DM38x and TMS320DM8127 DaVinci™ Digital Media Processor Imaging Subsystem (ISS)

User's Guide

Literature Number: SPRUHL6A
January 2013–Revised June 2016

Contents

List of Figures

List of Tables

Read This First

About This Manual

This document describes the operation of the Imaging Subsystem (ISS) in the DM38x and TMS320DM8127 DaVinci™ Digital Media Processor.

Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers may be shown with the suffix h or the prefix 0x. For example, the following number is 40 hexadecimal (decimal 64): 40h or 0x40
- Registers in this document are shown in figures and described in tables.
  - Each register figure shows a rectangle divided into fields that represent the fields of the register. Each field is labeled with its bit name, its beginning and ending bit numbers above, and its read/write properties with default reset value below. A legend explains the notation used for the properties.
  - Reserved bits in a register figure can have one of multiple meanings:
    - Not implemented on the device
    - Reserved for future device expansion
    - Reserved for TI testing
    - Reserved configurations of the device that are not supported
  - Writing non-default values to the Reserved bits could cause unexpected behavior and should be avoided.

Glossary

SLYZ022 — *TI Glossary*.

This glossary lists and explains terms, acronyms, and definitions.

Related Documentation From Texas Instruments

For product information, visit the Texas Instruments website at http://www.ti.com.

SPRS821 — DM385 and DM388 DaVinci™ Digital Media Processor

SPRS712 — TMS320DM8127 DaVinci™ Video Processor

Community Resources

The following links connect to TI community resources. Linked contents are provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

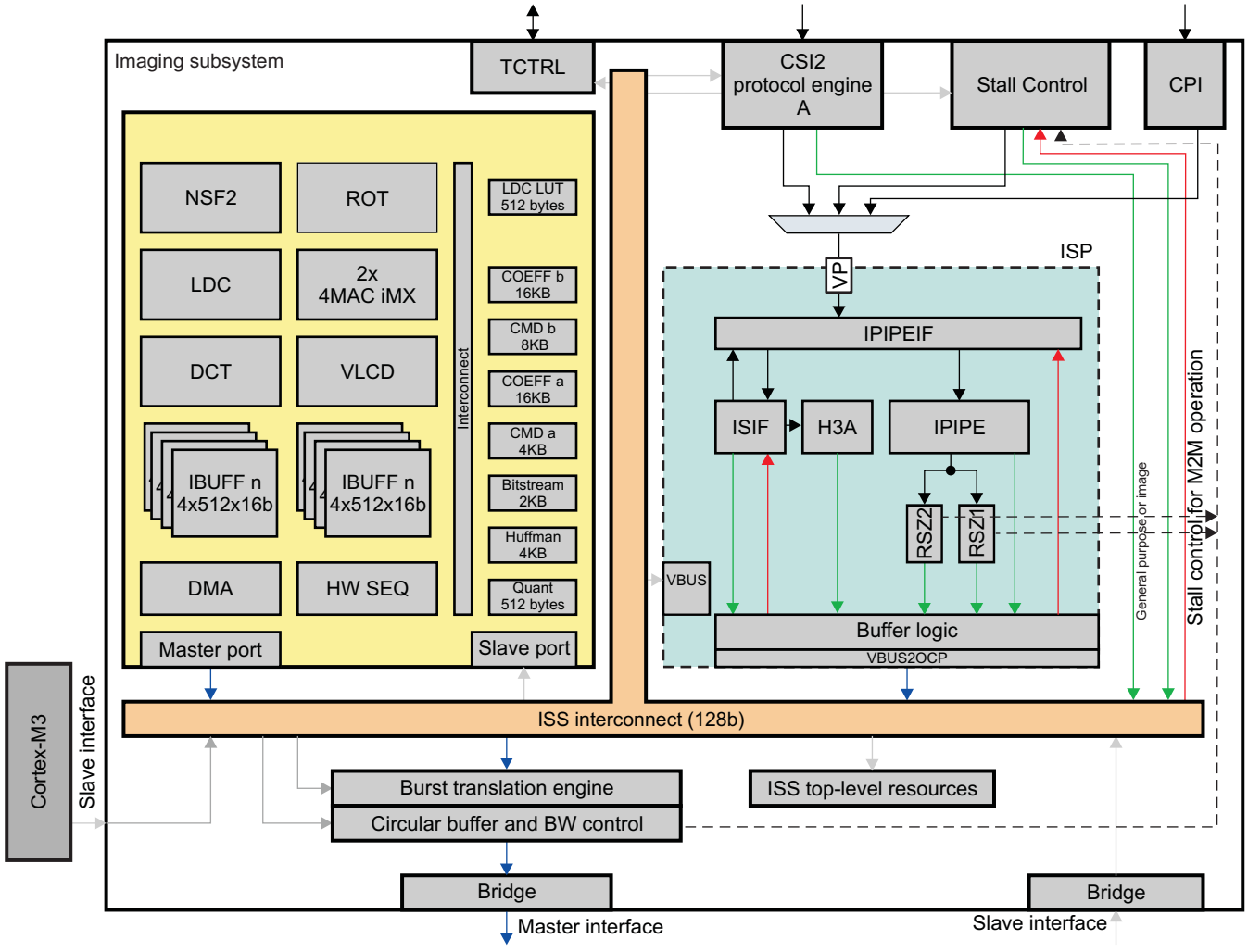
TI E2E™ Online Community— *TI's Engineer-to-Engineer (E2E) Community.* Created to foster collaboration among engineers. At e2e.ti.com, you can ask questions, share knowledge, explore ideas and help solve problems with fellow engineers.

TI Embedded Processors Wiki— *Texas Instruments Embedded Processors Wiki.* Established to help developers get started with Embedded Processors from Texas Instruments and to foster innovation and growth of general knowledge about the hardware and software surrounding these devices.

1 ISS Overview

The imaging subsystem (ISS) deals with the processing of the pixel data coming from an external image sensor, data from memory (image format encoding and decoding can be done to and from memory), or data from SL2 in IVA-HD for hardware encoding. With its subparts, such as interfaces and interconnects, image signal processor (ISP), and still image coprocessor (SIMCOP), the ISS is a key component for the following multimedia applications: camera viewfinder, video record, and still image capture. Figure 1 shows an overview of the ISS.

Figure 1. ISS Overview



The direction of the arrows shows the command flow direction from the master (initiator) to the slave (target). The following color conventions are used for the connections:

- Blue: Bidirectional, 128-bit-wide interface data connection
- Green: Write (ISS → system memory) data connection. Either 64-bit interface, 128-bit interface, or 32-bit MTC (inside ISP5).
- Red: Read (system memory → ISS) data connection. 128-bit interface port or 32-bit MTC (inside ISP5).
- Gray: 32-bit interface configuration connection
- Solid black: Video port and camera interface related signals
- Dotted black: Data flow stall control signal. Used to slow down ISP for memory-to-memory operation.

The ISS is mainly composed of a CSI2-A camera interface, a parallel interface (CPI), an ISP, and a block-based imaging accelerator (SIMCOP).

The ISS is designed to reach high throughput and low latency with large image sensors. In high-performance mode, the ISS supports a pixel throughput of 200 MPix/s.

The ISS is tightly coupled with a low-interrupt latency microprocessor unit (MPU) subsystem (Cortex™-M3 MPU) that runs a real-time operating system (OS) to reach optimal performance. Mainly, the Cortex-M3 MPU can quickly change the ISS configuration during frame blanking periods and run some sequencing tasks.

The ISS targets the following major use cases:

- Viewfinder with digital zoom, video stabilization, and rotations
- Up to 1080p video record at 30 fps with digital zoom, video stabilization, and rotation
- Up to 16 MPix still image capture with digital zoom and rotation
  - High performance mode: Up to 200 MPix/s throughput
  - High quality and low light modes: Up to 50 MPix/s throughput
- Still image capture during video record

NOTE: The ISS is not limited to 16 Mpix. Higher resolution can be achieved through multiple passes.

NOTE: For a detailed list of features of a certain submodule, see the related subsection.

The ISS offers the following features:

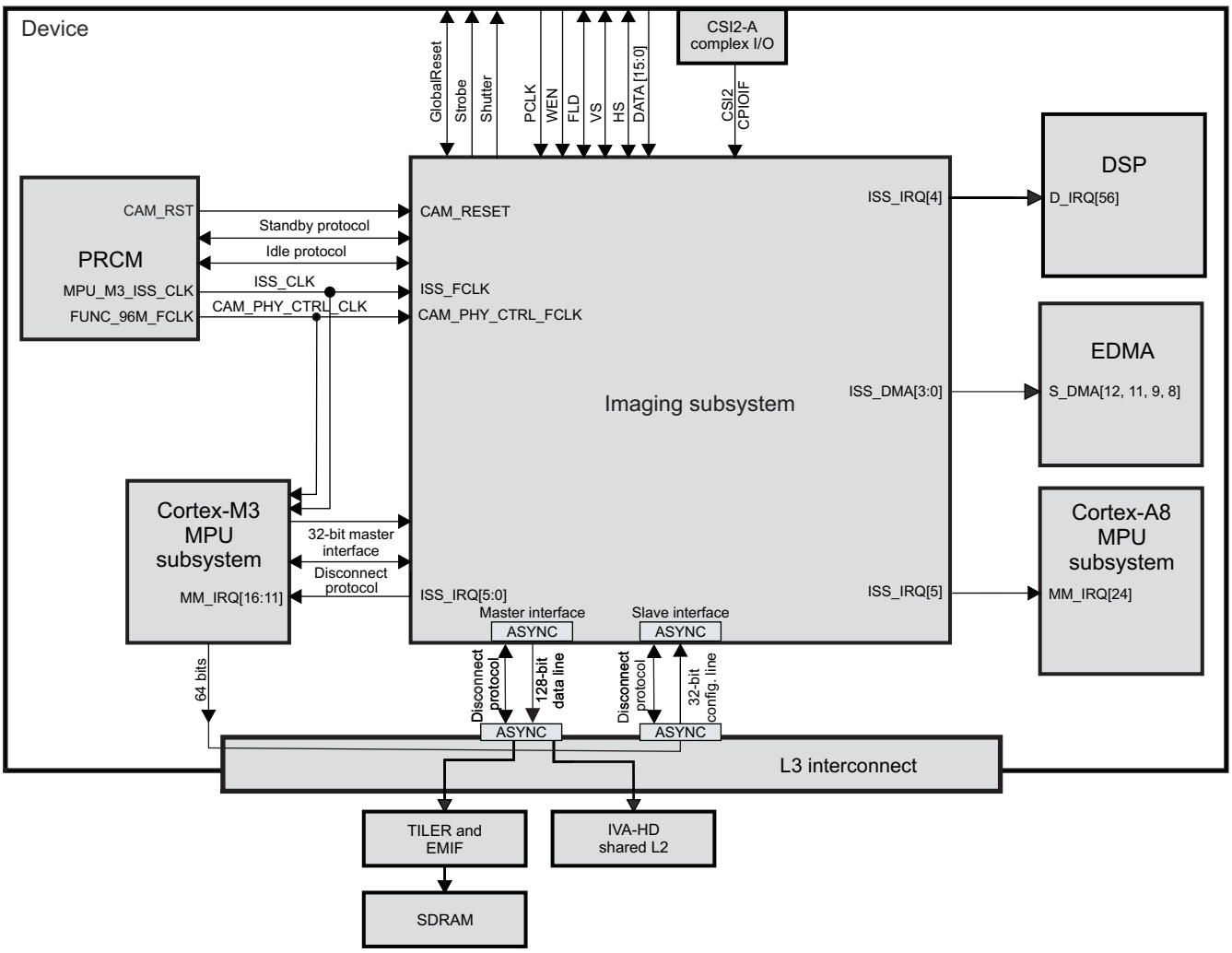
- SIMCOP:
  - Memory-to-memory operation
  - JPEG encode and decode hardware acceleration
  - High-ISO noise filtering (NSF 2.0)
  - Rotation accelerator
  - Warping accelerator
  - Lens distortion correction (YUV space)
- ISP:
  - On-the-fly or memory-to-memory processing
  - Up to 200-MHz pixel throughput
  - Statistic data collection
  - Image pipe interface front-end raw data processing
  - RGB and YUV data processing through ISIF and IPIPE
  - Hardware 3A (H3A) statistics block for real-time auto focus (AF), auto exposure (AE), and auto white balance (AWB)

  - Two image continuous real-time resizers
  - Video port (VP) for interfacing with the receivers and directing data to the ISP
- ISS interfaces:
  - 128-bit-wide data interface to the level 3 (L3) interconnect
  - Burst translation engine (BTE) tightly coupled with the TILER to support efficient rotation
  - Circular buffer for linear space, physically located in memory
  - The ISS relies on the centralized memory management unit (MMU).
  - CSI2 camera interface
  - Parallel interface (CPI) (16-bit wide, with up to 162 MPix/s throughput, and supporting BT656, SYNC modes)
  - System memory data read-back port (supported by the stall controller (SC) module)
- ISS interconnect:
  - 128-bit-wide network for image data (full speed)
  - 32-bit-wide network for configuration (half speed)
  - Hard and soft real-time data flows
  - CSI2-A, CPI, ISP, and SIMCOP data flow management

1.1 ISS Integration

This section describes the integration of the module in the device, including information about clocks, resets, and hardware requests. Figure 2 shows the ISS integration.

Figure 2. ISS Integration



NOTE: For more information about the IDLE hardware handshake and wake-up request, see section Clock Management, in chapter, Power, Reset, and Clock Management.

This section gives an overview of typical uses of the module. For more information about the relationship of the power, reset, and clock management (PRCM) module to the ISS clocks and the reset settings, see the detailed functional description in the *Power, Reset, and Clock Management* chapter of the TRM.

The ISS is part of ISS hardware and power management, which comprises the Cortex-M3 MPU, ISS, and a clock generator. These are all independent power domains. The ISS is part of the CAM power domain.

The PRCM module provides a single clock (MPU_M3_ISS_CLK) to ISS power management. This clock generates the clocks for the Cortex-M3 MPU, ISS, and the bridges to L3.

The ISS also supports software reset. A software reset has the same function as a hardware reset except that it does not reset the power-management protocols.

When enabled, MPU_M3_ISS_CLK is gated to provide ISS_CLK, which, after entering the ISS boundary, is named ISS_FCLK. MPU_M3_ISS_CLK is provided as long as the Cortex-M3 MPU or ISS modules within the ISS require it.

CAM_PHY_CTRL_CLK is gated from FUNC_96M_FCLK, which comes from the PRCM module. When enabled and inside the ISS boundary, it is called CAM_PHY_CTRL_FCLK.

NOTE: For more information about the device clocks and how they are handled by the PRCM module before going into the ISS, see *CM2 Clock Generator* in the *Power, Reset, and Clock Management* chapter of the TRM.

Table 1. ISS Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
ISS	PD_CAM	No	

Table 2. ISS Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
ISS	ISS_FLCK	ISS_CLK	PRCM	ISS global functional clock from the PRCM module
	CAM_PHY_CTRL_FCLK	CAM_PHY_CTRL_CLK	PRCM	Physical layer functional clock from the PRCM module
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
ISS	CAM_RESET	CAM_RST	PRCM	ISS global reset

Table 3. ISS Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination Signal Name**	**Destination**	**Description**
ISS	ISS_IRQ4	D_IRQ_56	DSP INTC	Interrupt generated by ISS to DSP
	ISS_IRQ5	MM_IRQ_24	Cortex-A8 INTC	Interrupt generated by ISS to Cortex-A8
	ISS_IRQ0	MM_IRQ_11	Cortex-M3 INTC	Interrupt generated by ISS to Cortex-M3
	ISS_IRQ1	MM_IRQ_12	Cortex-M3 INTC	Interrupt generated by ISS to Cortex-M3
	ISS_IRQ2	MM_IRQ_13	Cortex-M3 INTC	Interrupt generated by ISS to Cortex-M3
	ISS_IRQ3	MM_IRQ_14	Cortex-M3 INTC	Interrupt generated by ISS to Cortex-M3
	ISS_IRQ4	MM_IRQ_15	Cortex-M3 INTC	Interrupt generated by ISS to Cortex-M3
	ISS_IRQ5	MM_IRQ_16	Cortex-M3 INTC	Interrupt generated by ISS to Cortex-M3
DMA Requests				
Module Instance	**Source Signal Name**	**Destination Signal Name**	**Destination**	**Description**
ISS	ISS_DMA0	S_DMA_8	EDMA	Signal connected to the EDMA directly provided by ISP.
	ISS_DMA1	S_DMA_9	EDMA	Signal connected to the EDMA directly provided by ISP
	ISS_DMA2	S_DMA_11	EDMA	Signal connected to the EDMA directly provided by ISP
	ISS_DMA3	S_DMA_12	EDMA	Signal connected to the EDMA directly provided by ISP

1.1.1 ISS PRCM Interface Integration

1.1.1.1 ISS Clock Domains

The ISS has five asynchronous clock domains. Most of the logic uses ISS_FCLK; the other clock domains are used for interfaces.

Table 4 provides a high-level view of clocks.

Table 4. ISS Local Clock Domains

Name	Description
PCLK	Parallel interface (CPI) pixel clock provided by the camera external sensor. The CSI2-A and SC modules also have video port outputs, each having its own pixel clock. These pixel clocks are generated from the functional clock (ISS_FCLK). Then all three pixel clock sources are multiplexed into one clock provided to the ISP.
CSI2_A_BC	Byte clock provided by the CSI2-A complex I/O. Used by the CSI2-A receiver.
ISS_FCLK	Functional clock provided by ISS_CLK from the PRCM module. It is used by all ISS submodules and ISS top-level resources.
CAM_PHY_CTRL_FCLK	Functional clock provided by the PRCM module. It is used by the CSI2-A complex I/Os.

To save power, the ISS can divide the received functional clock (ISS_CLK) by 2 or 4 from Figure 13, [4:5] ISS_CLK_DIV. The configuration clock is always half the functional clock.

The functional clock of some submodules can be cut by software to reduce power consumption by cutting off or turning on the modules from the Figure 14 . Also, the pixel clocks sent by submodules to ISP can be cut off from Figure 14, VPORT*x*_CLK.

1.2 ISS Functional Description

This section provides only a top-level overview of the ISS. The ISS submodules are described in: *ISS:SIMCOP*,*ISS:Interfaces*, and *ISS:ISP*.

Section 1.2.1 describes the ISS power-management mechanisms and gives an introduction to and a functional description of the ISP submodules.

1.2.1 ISS Interrupts

Section 1.2.1.2 lists the events generated by the submodules and the top level of the ISS.

Each event that generates an interrupt can be individually enabled by setting the appropriate bit in the ISS_HL_IRQENABLE_SET_i register. The interrupt is disabled by setting the appropriate bit in the ISS_HL_IRQENABLE_CLR_i register.

When an event occurs, the corresponding bit in the ISS_HL_IRQSTATUS_RAW_i register is set regardless of whether or not the event is enabled. Bits in the ISS_HL_IRQSTATUS_i registers are set only when an enabled event occurs.

Software can clear a pending HS_VS_IRQ event by setting the ISS_HL_IRQSTATUS_i [17] HS_VS_IRQ bit. Events generated by submodules are automatically cleared at the ISS level when they are cleared at the submodule level.

1.2.1.1 ISS Interrupt Merger

The ISS merges the following eight interrupt sources into six physical interrupt lines. All six lines support level and pulse modes.

Table 5. ISS Interrupts

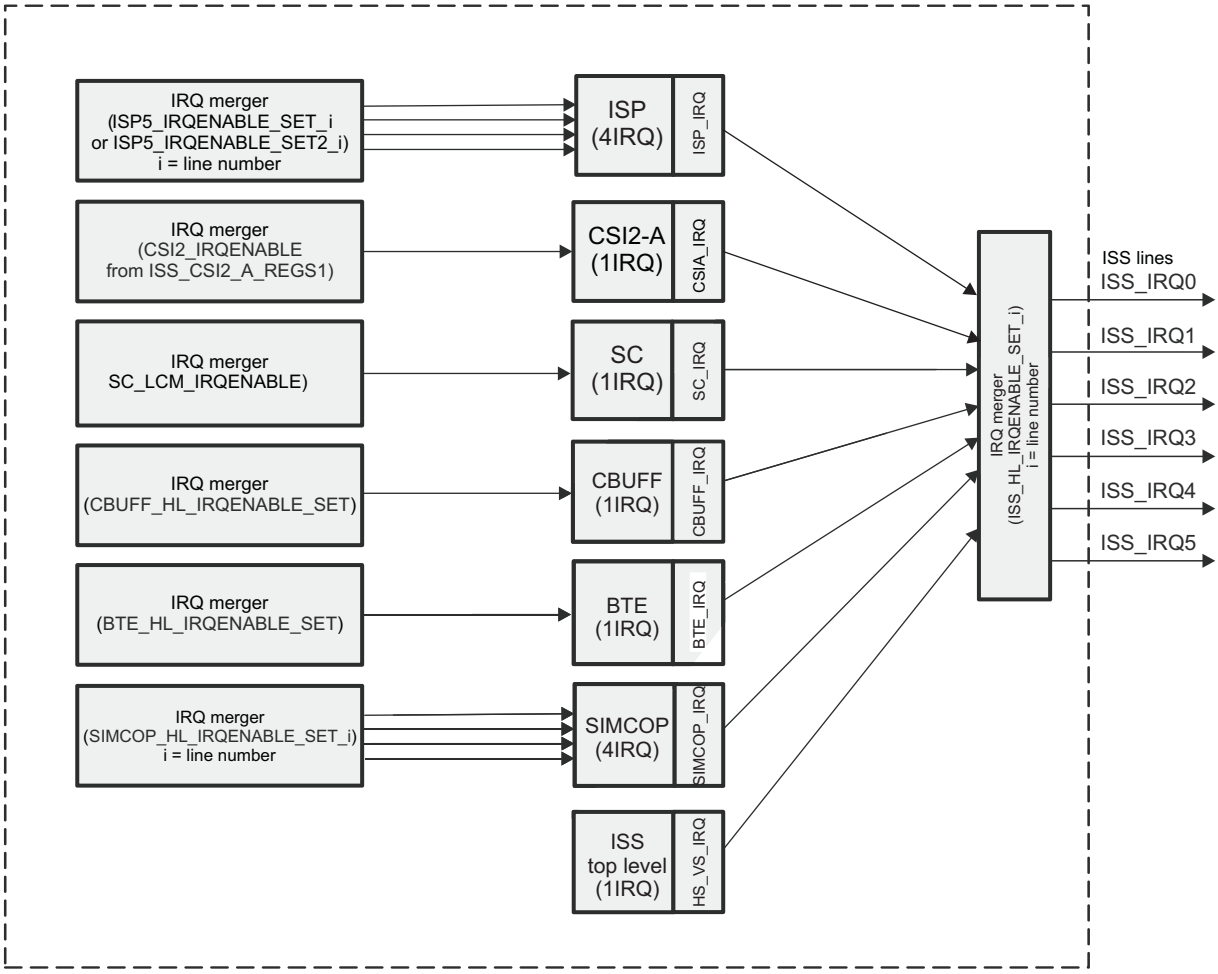
Interrupt	Source	Description
ISP_IRQ[3:0]	ISP	Interrupt generated by the ISP (1)
CSIA_IRQ	CSI2-A	Interrupt generated by the CSI2-A receiver(1)
SC_IRQ[8]	SC	Interrupt generated by the SC receiver(1)
CBUFF_IRQ	CBUFF	Interrupt generated by the circular buffer(1)
BTE_IRQ	BTE	Interrupt generated by the BTE (1)
SIMCOP_IRQ[3:0]	SIMCOP	Interrupt generated by SIMCOP
HS_VS_IRQ	ISS	HS or VS synchronization event. This event is triggered if a rising or falling edge is detected on the HS or VS signal. The rising or falling edge and the HS or VS signal selection are chosen using the ISP_CTRL[0:1] SYNC_DETECT bit field.

(1) For more information, see Section 1.2.1.2.

Software can select which interrupt sources are routed to each output line (ISS_IRQ[0:5]). All six physical interrupt outputs have equivalent functions. Software can use different interrupt lines to group events together by type and therefore reduce interrupt latencies. Typically, one interrupt line is used for low-priority events (errors), and the other interrupt lines are used for high-priority events (for example, SIMCOP sequencing and end-of-frame events to trigger configuration load). The ISS internal interrupt request (IRQ) merger (see Figure 3) relies only on level interrupts provided by submodules. Pulse interrupts provided by submodules are ignored. Typically, all of the interrupts are routed to the Cortex-M3 running the camera driver. In addition, one interrupt is routed to Cortex-A8, which may run the imaging software. Software does not need to clear events from submodules at the ISS level: it clears only the events at the submodule level; the IRQ merger automatically clears the IRQ at this level.

NOTE: Only the HS_VS_IRQ top-level event is cleared at the ISS level.

Figure 3. ISS Interrupt Merger



NOTE: For more information about mapping the six lines to the outside ISS device modules, see Table 5. Only the ISP, SIMCOP, and HL ISS merger IRQ lines are configurable. Each interrupt can be mapped to the required line.

1.2.1.2 ISS Submodule Interrupts

The ISS shown in Figure 3 and listed in Table 3 can generate six interrupts.

1.2.1.2.1 ISS ISP Interrupts

Table 6 summarizes events that cause ISP interrupts.

Table 6. ISS ISP Interrupts

Event and Register[(1)]	Description
ISP5_IRQENABLE_SET_i[[31] OCP_ERR_IRQ	An interface port error has been received on the ISP master port.
ISP5_IRQENABLE_SET_i[29] IPIPE_INT_DPC_RNEW1	HD interrupt signal to indicate the need to renew the defect pixel correction (DPC) table with new entries. The second 128 entries in the DPC table must be updated when this event triggers. This event is triggered when the 255th entry in the look-up table (LUT) is used. This interrupt is not synchronous to the HD signal.

(1) i = 0 to 3

Table 6. ISS ISP Interrupts (continued)

Event and Register[1]	Description
ISP5_IRQENABLE_SET_i[28] IPIPE_INT_DPC_RNEW0	VD interrupt signal to indicate the need to renew the DPC table with new entries. The first 128 entries in the DPC table must be updated when the event triggers. This event is triggered when the 127th entry in the LUT is used. This interrupt is not synchronous to the HD signal.
ISP5_IRQENABLE_SET_i[27] IPIPE_INT_DPC_INI	Interrupt to signal the need to initialize the DPC table. The DPC table contains two tables of 128 entries. When this signal is used, software must ensure that the 256 table entries are updated with the DPC information.
ISP5_IRQENABLE_SET_i[25] IPIPE_INT_EOF	End of frame interrupt signal
ISP5_IRQENABLE_SET_i[24] H3A_INT_EOF	End of frame interrupt signal
ISP5_IRQENABLE_SET_i[22] RSZ_INT_EOF1	See Section 1.3 and ISS ISP Register Manual.
ISP5_IRQENABLE_SET_i[23] RSZ_INT_EOF0	
ISP5_IRQENABLE_SET_i[19] RSZ_FIFO_IN_BLK_ERR	
ISP5_IRQENABLE_SET_i[18] RSZ_FIFO_IN_OVF	
ISP5_IRQENABLE_SET_i[17] RSZ_INT_CYC_RZB	
ISP5_IRQENABLE_SET_i[16] RSZ_INT_CYC_RZA	
ISP5_IRQENABLE_SET_i[15] RSZ_INT_DMA	
ISP5_IRQENABLE_SET_i[14] RSZ_INT_LAST_PIX	
ISP5_IRQENABLE_SET_i[13] RSZ_INT_REG	
ISP5_IRQENABLE_SET_i[12] H3A_INT	Interrupt generated by the AF and AE/AWB blocks inside the H3A module. It indicates the end of processing a frame and is active high for one configuration bus clock cycle.
ISP5_IRQENABLE_SET_i[11] AF_INT	AF inside generates an interrupt at the end of processing frame; a third interrupt is generated at the same time as the last process to finish.
ISP5_IRQENABLE_SET_i[10] AEW_INT	AEW inside generates an interrupt at the end of processing frame; a third interrupt is generated at the same time as the last process to finish.
ISP5_IRQENABLE_SET_i[9] IPIPEIF_IRQ	IPIPEIF module interrupt is generated at the start position of a frame and is active high for one configuration bus clock cycle.
ISP5_IRQENABLE_SET_i[8] IPIPE_INT_HST	IPIPE module interrupt is generated when histogram is done.
ISP5_IRQENABLE_SET_i[7] IPIPE_INT_BSC	IPIPE module interrupt is generated when boundary signal calculation is done.
ISP5_IRQENABLE_SET_i[6] IPIPE_INT_DMA	IPIPE module interrupt is issued when the SDRAM transfer of boxcar is done. At this time, IPIPE EOF is sent to buffer logic.
ISP5_IRQENABLE_SET_i[5] IPIPE_INT_LAST_PIX	IPIPE module interrupt is issued when the last pixel of a frame comes into IPIPE.
ISP5_IRQENABLE_SET_i[4] IPIPE_INT_REG	IPIPE module interrupt is issued when the register update of the module is allowed.
ISP5_IRQENABLE_SET_i[0] ISIF_INT_0	See Section 1.3 and ISS ISP Register Manual.
ISP5_IRQENABLE_SET_i[1] ISIF_INT_1	
ISP5_IRQENABLE_SET_i[2] ISIF_INT_2	
ISP5_IRQENABLE_SET_i[3] ISIF_INT_3	
ISP5_IRQENABLE_SET2_i[0] H3A_OVF	
ISP5_IRQENABLE_SET2_i[1] IPIPEIF_UDF	
ISP5_IRQENABLE_SET2_i[2] IPIPE_BOXCAR_OVF	
ISP5_IRQENABLE_SET2_i[3] ISIF_OVF	
ISP5_IRQENABLE_SET2_i[4] IPIPE_HST_ERR	
ISP5_IRQENABLE_SET2_i[5] IPIPE_BSC_ERR	

1.2.1.2.2 ISS CSI2-A Complex I/O Interrupts

Table 7 lists the event generation of the CSI2-A receiver through the CSI2 interrupt status and interrupt enable registers. The events are checked for status using the CSI2_IRQSTATUS and CSI2_IRQENABLE registers.

Table 7. ISS CSI2-A Interrupts

Event and Register	Description
CSI2_IRQENABLE[14] OCP_ERR_IRQ	Interface port error
CSI2_IRQENABLE[13] SHORT_PACKET_IRQ	Short packet reception (other than sync events: line start, line end, frame start, and frame end; only data types from 0x8 to 0xF are considered)
CSI2_IRQENABLE[12] ECC_CORRECTION_IRQ	ECC was used to correct a 1-bit error (short packet only).
CSI2_IRQENABLE[11] ECC_NO_CORRECTION_IRQ	ECC was not used to correct the header because the error is larger than 1 bit (short and long packets).
CSI2_IRQENABLE[9] COMPLEXIO_ERR_IRQ	Error signaling from complex I/O: This interrupt is triggered when any error is received from the complex I/O (events are defined in CSI2_COMPLEXIO_IRQSTATUS [see Table 8]).
CSI2_IRQENABLE[8] FIFO_OVF_IRQ	FIFO overflow error: This interrupt is triggered when a FIFO overflow is detected. An overflow can occur if there is a mismatch between the data input and output rates. A reset of the module is required to restart correctly.
CSI2_IRQENABLE[7] CONTEXT7	At least one interrupt event enabled from Context 7 occurred (see Table 9).
CSI2_IRQENABLE[6] CONTEXT6	At least one interrupt event enabled from Context 6 occurred (see Table 9).
CSI2_IRQENABLE[5] CONTEXT5	At least one interrupt event enabled from Context 5 occurred (see Table 9).
CSI2_IRQENABLE[4] CONTEXT4	At least one interrupt event enabled from Context 4 occurred (see Table 9).
CSI2_IRQENABLE[3] CONTEXT3	At least one interrupt event enabled from Context 3 occurred (see Table 9).
CSI2_IRQENABLE[2] CONTEXT2	At least one interrupt event enabled from Context 2 occurred (see Table 9).
CSI2_IRQENABLE[1] CONTEXT1	At least one interrupt event enabled from Context 1 occurred (see Table 9).
CSI2_IRQENABLE[0] CONTEXT0	At least one interrupt event enabled from Context 0 occurred (see Table 9).

Table 8 lists CSI2 receiver event generation through the CSI2-A complex I/O interrupt status and interrupt enable registers. The events are checked and controlled from the CSI2_COMPLEXIO_IRQSTATUS and CSI2_COMPLEXIO_IRQENABLE registers.

Table 8. ISS CSI2-A Receivers Complex I/O Interrupts

Event and Register	Description
CSI2_COMPLEXIO_IRQENABLE[0] ERRSOTHS1	Start of transmission error for lane 1
CSI2_COMPLEXIO_IRQENABLE[1] ERRSOTHS2	Start of transmission error for lane 2
CSI2_COMPLEXIO_IRQENABLE[2] ERRSOTHS3	Start of transmission error for lane 3
CSI2_COMPLEXIO_IRQENABLE[5] ERRSOTSYNCHS1	Start of transmission sync error for lane 1
CSI2_COMPLEXIO_IRQENABLE[6] ERRSOTSYNCHS2	Start of transmission sync error for lane 2
CSI2_COMPLEXIO_IRQENABLE[7] ERRSOTSYNCHS3	Start of transmission sync error for lane 3
CSI2_COMPLEXIO_IRQENABLE[10] ERRESC1	Escape entry error for lane 1
CSI2_COMPLEXIO_IRQENABLE[11] ERRESC2	Escape entry error for lane 2
CSI2_COMPLEXIO_IRQENABLE[12] ERRESC3	Escape entry error for lane 3
CSI2_COMPLEXIO_IRQENABLE[15] ERRCONTROL1	Control error for lane 1
CSI2_COMPLEXIO_IRQENABLE[16] ERRCONTROL2	Control error for lane 2
CSI2_COMPLEXIO_IRQENABLE[17] ERRCONTROL3	Control error for lane 3
CSI2_COMPLEXIO_IRQENABLE[20] STATEULPM1	Lane 1 in ULPM
CSI2_COMPLEXIO_IRQENABLE[21] STATEULPM2	Lane 2 in ULPM
CSI2_COMPLEXIO_IRQENABLE[22] STATEULPM3	Lane 3 in ULPM
CSI2_COMPLEXIO_IRQENABLE[25] STATEALLULPMENTER	All active lanes are entering the ULPM.
CSI2_COMPLEXIO_IRQENABLE[26] STATEALLULPMEXIT	At least one active lane exited the ULPM.

Because the CSI2-A receiver supports eight contexts, the CSI2_CTX_IRQSTATUS_i and CSI2_CTX_IRQENABLE_i registers are present eight times (one time per context).

The events are generated only for the enabled context(s). Table 9 describes the generation of the CS12 receiver event through the CSI2_CTX_IRQSTATUS_i and CSI2_CTX_IRQENABLE_i registers.

Table 9. ISS CSI2-A Receiver CONTEXT Interrupts

Event[1]	Description
CSI2_CTX_IRQENABLE_i[0] FS_IRQ	Frame start: This interrupt is triggered when a frame-start synchronization code is detected in the CSI2 data stream.
CSI2_CTX_IRQENABLE_i[1] FE_IRQ	Frame end: This interrupt is triggered when a frame-end synchronization code is detected in the CSI2 data stream.
CSI2_CTX_IRQENABLE_i[2] LS_IRQ	Line start: This interrupt is triggered when a line-start synchronization code is detected in the CSI2 data stream.
CSI2_CTX_IRQENABLE_i[3] LE_IRQ	Line end: This interrupt is triggered when a line-end synchronization code is detected in the CSI2 data stream.
CSI2_CTX_IRQENABLE_i[5] CS_IRQ	CS error: This interrupt is triggered when a mismatch between the transmitter and receiver checksums (payload) is detected.
CSI2_CTX_IRQENABLE_i[6] FRAME_NUMBER_IRQ	Frame counter reached: This interrupt is triggered when the frame counter reaches its programmable target value.
CSI2_CTX_IRQENABLE_i[7] LINE_NUMBER_IRQ	Line number reached: The programmable line number is received. The modulo feature can be selected (CSI2_CTX_CTRL1_i.LINE_MODULO). When selected, the interrupt is generated for each line number multiple of the programmed line number (CSI2_CTX_CTRL3_i.LINE_NUMBER); otherwise, the interrupt is generated only for the line number.
CSI2_CTX_IRQENABLE_i[8] ECC_CORRECTION_IRQ	ECC was used to correct a 1-bit error (long packets only).

(1) i = 0 to 7

1.2.1.2.3 ISS Stall Controller (SC) Interrupts

Table 10 summarizes the stall controller (SC) interrupts.

Table 10. ISS Stall Controller Interrupts

Event and Register	Description
SC_LCM_IRQENABLE[1] LCM_OCPERROR	SC an interface error occurred on the master read port. This interrupt is triggered when an OCP error is detected on the master read port.
SC_LCM_IRQENABLE[0] LCM_EOF	Memory read channel – end of frame: This interrupt is triggered when a frame is read completely from memory.

1.2.1.2.4 ISS CBUFF Interrupts

Table 11 summarizes the CBUFF interrupts.

Table 11. ISS CBUFF Interrupts

Event and Register	Description
CBUFF_HL_IRQENABLE_SET[31] IRQ_CTX7_OVR	CBUFF overflow
CBUFF_HL_IRQENABLE_SET[30] IRQ_CTX6_OVR	CBUFF overflow
CBUFF_HL_IRQENABLE_SET[29] IRQ_CTX5_OVR	CBUFF overflow
CBUFF_HL_IRQENABLE_SET[28] IRQ_CTX4_OVR	CBUFF overflow
CBUFF_HL_IRQENABLE_SET[27] IRQ_CTX3_OVR	CBUFF overflow
CBUFF_HL_IRQENABLE_SET[26] IRQ_CTX2_OVR	CBUFF overflow
CBUFF_HL_IRQENABLE_SET[25] IRQ_CTX1_OVR	CBUFF overflow
CBUFF_HL_IRQENABLE_SET[24] IRQ_CTX0_OVR	CBUFF overflow
CBUFF_HL_IRQENABLE_SET[23] IRQ_CTX7_INVALID	CBUFF invalid access
CBUFF_HL_IRQENABLE_SET[22] IRQ_CTX6_INVALID	CBUFF invalid access
CBUFF_HL_IRQENABLE_SET[21] IRQ_CTX5_INVALID	CBUFF invalid access
CBUFF_HL_IRQENABLE_SET[20] IRQ_CTX4_INVALID	CBUFF invalid access
CBUFF_HL_IRQENABLE_SET[19] IRQ_CTX3_INVALID	CBUFF invalid access
CBUFF_HL_IRQENABLE_SET[18] IRQ_CTX2_INVALID	CBUFF invalid access
CBUFF_HL_IRQENABLE_SET[17] IRQ_CTX1_INVALID	CBUFF invalid access
CBUFF_HL_IRQENABLE_SET[16] IRQ_CTX0_INVALID	CBUFF invalid access
CBUFF_HL_IRQENABLE_SET[15] IRQ_CTX7_READY	CBUFF WB physical window ready for access by the CPU
CBUFF_HL_IRQENABLE_SET[14] IRQ_CTX6_READY	CBUFF WB physical window ready for access by the CPU
CBUFF_HL_IRQENABLE_SET[13] IRQ_CTX5_READY	CBUFF WB physical window ready for access by the CPU
CBUFF_HL_IRQENABLE_SET[12] IRQ_CTX4_READY	CBUFF WB physical window ready for access by the CPU
CBUFF_HL_IRQENABLE_SET[11] IRQ_CTX3_READY	CBUFF WB physical window ready for access by the CPU
CBUFF_HL_IRQENABLE_SET[10] IRQ_CTX2_READY	CBUFF WB physical window ready for access by the CPU
CBUFF_HL_IRQENABLE_SET[9] IRQ_CTX1_READY	CBUFF WB physical window ready for access by the CPU
CBUFF_HL_IRQENABLE_SET[8] IRQ_CTX0_READY	CBUFF WB physical window ready for access by the CPU
CBUFF_HL_IRQENABLE_SET[0] IRQ_OCP_ERR	CBUFF master interface port error

1.2.1.2.5 ISS BTE Interrupts

Table 12 summarizes the BTE interrupts.

Table 12. ISS BTE Interrupts

Event and Register	Description
BTE_HL_IRQENABLE_SET[31] IRQ_CTX7_ERR	Writes enable notification for read request before enough data is prefetched. Reads notify when a read request that requires translation on Context 7 is received, but not enough frame lines have been prefetched in the buffer. See I*SS BTE Buffer Prefetch*.
BTE_HL_IRQENABLE_SET[30] IRQ_CTX6_ERR	Writes enable notification for read request before enough data is prefetched. Reads notify when a read request that requires translation on Context 6 is received, but not enough frame lines have been prefetched in the buffer. See *ISS BTE Buffer Prefetch*.
BTE_HL_IRQENABLE_SET[29] IRQ_CTX5_ERR	Writes enable notification for read request before enough data is prefetched. Reads notify when a read request that requires translation on Context 5 is received, but not enough frame lines have been prefetched in the buffer. See *ISS BTE Buffer Prefetch*.
BTE_HL_IRQENABLE_SET[28] IRQ_CTX4_ERR	Writes enable notification for read request before enough data is prefetched. Reads notify when a read request that requires translation on Context 4 is received, but not enough frame lines have been prefetched in the buffer. See *ISS BTE Buffer Prefetch*.
BTE_HL_IRQENABLE_SET[27] IRQ_CTX3_ERR	Writes enable notification for read request before enough data is prefetched. Reads notify when a read request that requires translation on Context 3 is received, but not enough frame lines have been prefetched in the buffer. See *ISS BTE Buffer Prefetch*.
BTE_HL_IRQENABLE_SET[26] IRQ_CTX2_ERR	Writes enable notification for read request before enough data is prefetched. Reads notify when a read request that requires translation on Context 2 is received, but not enough frame lines have been prefetched in the buffer. See *ISS BTE Buffer Prefetch*.
BTE_HL_IRQENABLE_SET[25] IRQ_CTX1_ERR	Writes enable notification for read request before enough data is prefetched. Reads notify when a read request that requires translation on Context 1 is received, but not enough frame lines have been prefetched in the buffer. See *ISS BTE Buffer Prefetch*.
BTE_HL_IRQENABLE_SET[24] IRQ_CTX0_ERR	Writes enable notification for read request before enough data is prefetched. Reads notify when a read request that requires translation on Context 0 is received, but not enough frame lines have been prefetched in the buffer. See *ISS BTE Buffer Prefetch*.
BTE_HL_IRQENABLE_SET[23] IRQ_CTX7_INVALID	Writes enable invalid access to Context 7. Reads notify when access to an unexpected location in Context 7 is requested, or the start context location access is valid, but the burst length exceeds the Context 7 end. See *ISS BTE Virtual Address Space and Context Mapping*.
BTE_HL_IRQENABLE_SET[22] IRQ_CTX6_INVALID	Writes enable invalid access to Context 6. Reads notify when access to an unexpected location in Context 6 is requested, or the start context location access is valid, but the burst length exceeds the Context 6 end. See *ISS BTE Virtual Address Space and Context Mapping*.
BTE_HL_IRQENABLE_SET[21] IRQ_CTX5_INVALID	Writes enable invalid access to Context 5. Reads notify when access to an unexpected location in Context 5 is requested, or the start context location access is valid, but the burst length exceeds the Context 5 end. See *ISS BTE Virtual Address Space and Context Mapping*
BTE_HL_IRQENABLE_SET[20] IRQ_CTX4_INVALID	Writes enable invalid access to Context 4. Reads notify when access to an unexpected location in Context 4 is requested, or the start context location access is valid, but the burst length exceeds the Context 4 end. See *ISS BTE Virtual Address Space and Context Mapping*.
BTE_HL_IRQENABLE_SET[19] IRQ_CTX3_INVALID	Writes enable invalid access to Context 3. Reads notify when access to an unexpected location in Context 3 is requested, or the start context location access is valid, but the burst length exceeds the Context 3 end. See *ISS BTE Virtual Address Space and Context Mapping*.

Table 12. ISS BTE Interrupts (continued)

Event and Register	Description
BTE_HL_IRQENABLE_SET[18] IRQ_CTX2_INVALID	Writes enable invalid access to Context 2. Reads notify when access to an unexpected location in Context 2 is requested, or the start context location access is valid, but the burst length exceeds the Context 2 end. See *ISS BTE Virtual Address Space and Context Mapping*.
BTE_HL_IRQENABLE_SET[17] IRQ_CTX1_INVALID	Writes enable invalid access to Context 1. Reads notify when access to an unexpected location in Context 1 is requested, or the start context location access is valid, but the burst length exceeds the Context 1 end. See *ISS BTE Virtual Address Space and Context Mapping*.
BTE_HL_IRQENABLE_SET[16] IRQ_CTX0_INVALID	Writes enable invalid access to Context 0. Reads notify when access to an unexpected location in Context 0 is requested, or the start context location access is valid, but the burst length exceeds the Context 0 end. See *ISS BTE Virtual Address Space and Context Mapping*.
BTE_HL_IRQENABLE_SET[15] IRQ_CTX7_DONE	Writes enable notification for context that has been fully and successfully transferred to TILER. Reads notify when enough frame lines have been prefetched into the buffer and Context 7 is translated to TILER. This interrupt is triggered when flushing completes (if enabled) in one-shot mode. It is triggered once per frame in continuous mode. See *ISS BTE Buffer Prefetch*.
BTE_HL_IRQENABLE_SET[14] IRQ_CTX6_DONE	Writes enable notification for context that has been fully and successfully transferred to TILER. Reads notify when enough frame lines have been prefetched into the buffer and Context 6 is translated to TILER. This interrupt is triggered when flushing completes (if enabled) in one-shot mode. It is triggered once per frame in continuous mode. See *ISS BTE Buffer Prefetch*.
BTE_HL_IRQENABLE_SET[13] IRQ_CTX5_DONE	Writes enable notification for context that has been fully and successfully transferred to TILER. Reads notify when enough frame lines have been prefetched into the buffer and Context 5 is translated to TILER. This interrupt is triggered when flushing completes (if enabled) in one-shot mode. It is triggered once per frame in continuous mode. See *ISS BTE Buffer Prefetch*.
BTE_HL_IRQENABLE_SET[12] IRQ_CTX4_DONE	Writes enable notification for context that has been fully and successfully transferred to TILER. Reads notify when enough frame lines have been prefetched into the buffer and Context 4 is translated to TILER. This interrupt is triggered when flushing completes (if enabled) in one-shot mode. It is triggered once per frame in continuous mode. See *ISS BTE Buffer Prefetch*.
BTE_HL_IRQENABLE_SET[11] IRQ_CTX3_DONE	Writes enable notification for context that has been fully and successfully transferred to TILER. Reads notify when enough frame lines have been prefetched into the buffer and Context 3 is translated to TILER. This interrupt is triggered when flushing completes (if enabled) in one-shot mode. It is triggered once per frame in continuous mode. See *ISS BTE Buffer Prefetch*.
BTE_HL_IRQENABLE_SET[10] IRQ_CTX2_DONE	Writes enable notification for context that has been fully and successfully transferred to TILER. Reads notify when enough frame lines have been prefetched into the buffer and Context 2 is translated to TILER. This interrupt is triggered when flushing completes (if enabled) in one-shot mode. It is triggered once per frame in continuous mode. See *ISS BTE Buffer Prefetch*.
BTE_HL_IRQENABLE_SET[9] IRQ_CTX1_DONE	Writes enable notification for context that has been fully and successfully transferred to TILER. Reads notify when enough frame lines have been prefetched into the buffer and Context 1 is translated to TILER. This interrupt is triggered when flushing completes (if enabled) in one-shot mode. It is triggered once per frame in continuous mode. See *ISS BTE Buffer Prefetch*.
BTE_HL_IRQENABLE_SET[8] IRQ_CTX0_DONE	Writes enable notification for context that has been fully and successfully transferred to TILER. Reads notify when enough frame lines have been prefetched into the buffer and Context 0 is translated to TILER. This interrupt is triggered when flushing completes (if enabled) in one-shot mode. It is triggered once per frame in continuous mode. See *ISS BTE Buffer Prefetch*.

Table 12. ISS BTE Interrupts (continued)

Event and Register	Description
BTE_HL_IRQENABLE_SET[1] IRQ_INVALID	Writes enable Invalid virtual space access notification. Reads notify when access falls into a translated from the BTE region, but it is 2D access or it does not map to an active context. See *ISS BTE Virtual Address Space and Context Mapping*.
BTE_HL_IRQENABLE_SET[1] IRQ_OCP_ERR	Writes enable notification for error on the master output interface. Reads notify when an error has occurred on the master output interface.

1.2.1.2.6 ISS SIMCOP Interrupts

Table 13 summarizes the SIMCOP high-level interrupts mapped to the outer boundaries of the SIMCOP. For more information about interrupts generated from inside the SIMCOP modules, see the SIMCOP section.

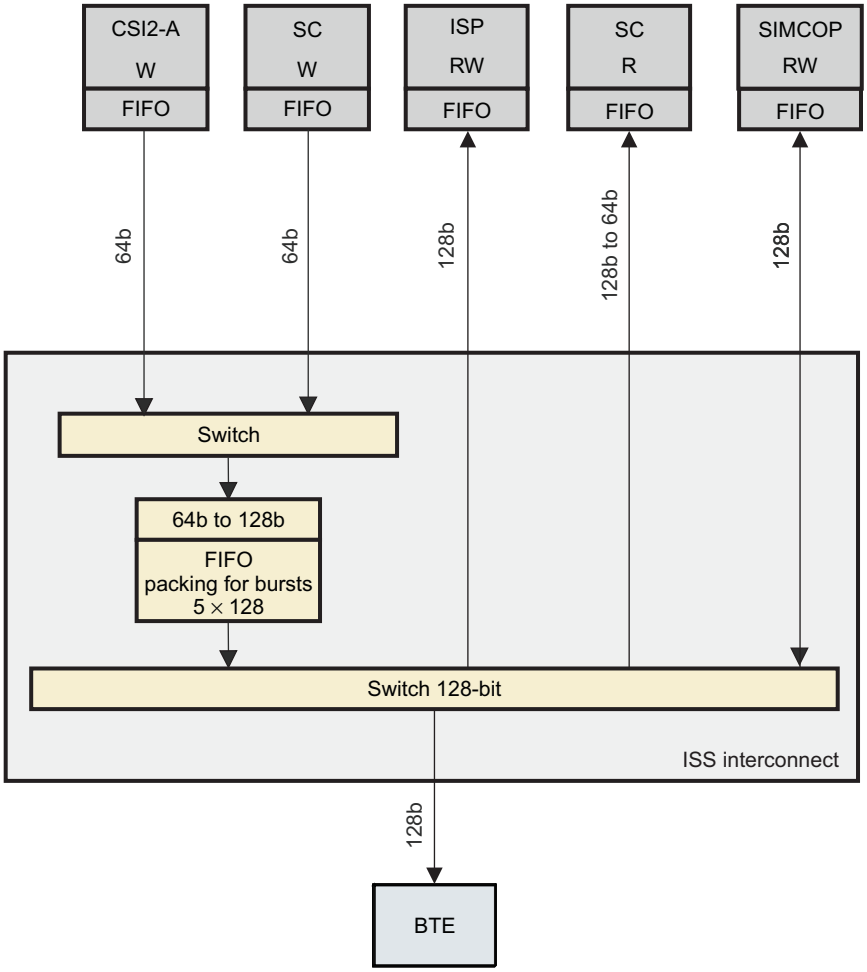
Table 13. ISS SIMCOP High-Level Interrupts

Event and Register	Description
SIMCOP_HL_IRQENABLE_SET_i[19] CPU_PROC_START_IRQ	Interrupt used when CPU data processing is used in a macroblock processing pipeline. When the CPU receives this IRQ, data is ready to be processed. When the CPU finishes processing the data, it acknowledges by setting the SIMCOP_HWSEQ_CTRL.CPU_PROC_DONE bit.
SIMCOP_HL_IRQENABLE_SET_i[18] SIMCOP_DMA_IRQ1	Interrupt triggered by SIMCOP DMA
SIMCOP_HL_IRQENABLE_SET_i[16] OCP_ERR_IRQ	SIMCOP master port interface error
SIMCOP_HL_IRQENABLE_SET_i[15] VLCDJ_DECODE_ERR_IRQ	A decode error has been signaled by the VLCDJ module.
SIMCOP_HL_IRQENABLE_SET_i[14] DONE_IRQ	Event triggered when hardware sequencer finishes the sequence: • The sequence step counter has reached the limit. • All accelerator and DMA events for the last sequence step have been received.
SIMCOP_HL_IRQENABLE_SET_i[13] STEP3_IRQ	Event triggered when a SIMCOP Context 3 is activated by the hardware sequencer.
SIMCOP_HL_IRQENABLE_SET_i[12] STEP2_IRQ	Event triggered when a SIMCOP Context 2 is activated by the hardware sequencer.
SIMCOP_HL_IRQENABLE_SET_i[11] STEP1_IRQ	Event triggered when a SIMCOP Context 1 is activated by the hardware sequencer.
SIMCOP_HL_IRQENABLE_SET_i[10] STEP0_IRQ	Event triggered when a SIMCOP Context 0 is activated by the hardware sequencer.
SIMCOP_HL_IRQENABLE_SET_i[9] LDC_BLOCK_IRQ	A macroblock has been processed.
SIMCOP_HL_IRQENABLE_SET_i[7] ROT_A	Rotational engine interrupt
SIMCOP_HL_IRQENABLE_SET_i[6] IMX_B_IRQ	Event triggered when iMX has executed a SLEEP instruction
SIMCOP_HL_IRQENABLE_SET_i[5] IMX_A_IRQ	Event triggered when iMX has executed a SLEEP instruction
SIMCOP_HL_IRQENABLE_SET_i[4] NSF_IRQ	Event triggered by the NSF2 imaging accelerator when processing of a block is done
SIMCOP_HL_IRQENABLE_SET_i[3] VLCDJ_BLOC_IRQ	A macroblock has been processed (that is, encode and decode).
SIMCOP_HL_IRQENABLE_SET_i[2] DCT_IRQ	DCT operating is complete (configured number of MCUs for YUV4:2:0/4:2:2 mode, or number of blocks for sequential block mode).
SIMCOP_HL_IRQENABLE_SET_i[1] LDC_FRAME_IRQ	A full frame has been processed.
SIMCOP_HL_IRQENABLE_SET_i[0] SIMCOP_DMA_IRQ0	Interrupt triggered by SIMCOP DMA

1.2.2 ISS Interconnect

Figure 4 shows the ISS local interconnect data network.

Figure 4. ISS Local Interconnect Data Network



The data network collects requests from CSI2-A, SC, ISP, and SIMCOP and forwards them to the BTE.

A first level of arbitration occurs between 64-bit initiators using a round-robin algorithm. Arbitration is done at the interface transaction level (up to 16 × 64-bit burst). The ISS interconnect packs bursts into up to 8 × 128-bit bursts before forwarding them to the second level of arbitration. The ISS interconnect uses a 5 × 128-bit first in first out (FIFO) for burst packing. Burst packing avoids inserting IDLE cycles into requests sent to the CBUFF.

A second level of arbitration occurs between 128-bit initiators. Again, a round-robin algorithm is used. Arbitration is done at bus transaction level (up to 8 × 128-bit burst). The ISS interconnect has no data FIFOs on the request or response paths. This implies the following for the ISP master, SC master, and SIMCOP master ports:

- The request and data phases can be stalled by the ISS interconnect when the port connected to the CBUFF becomes stalled.
- Read masters must be able to accept the data they have requested without stalling the response flow.

CSI2, SC, ISP, and SIMCOP can generate up to 16 tags per master port. The number of tags that can be used per initiator is set by software (the ISS_CTRL.TAG_CNT register). The ISS interconnect compresses the 6 × 16 internal ISS tags into 16 tags. Tag compression occurs after the 128-bit switch. When all 16 tags are used, the switch output is stalled. This stalls requests from ISS internal initiators.

NOTE: There is another internal 32-bit configuration interconnect within the ISS. There is no physical link between the data and the configuration interconnect.

1.2.3 ISS Clocks

The clocks of ISS submodules can be cut individually using the ISS_CLKCTRL register. Software can poll the module status reading the appropriate bit in the ISS_CLKSTAT register.

When software wants to enable a submodule:

- Software sets the appropriate bit in the ISS_CLKCTRL register.
- Hardware enables the submodule functional and interface clocks (expected to take a few cycles).
- Hardware sets the appropriate bit in the ISS_CLKSTAT register.

Software must enable the modules in the correct order. The hardware imposes no particular constraint. For example, when data must be provided by the CSI2-A receiver and processed by the ISP, both modules must be enabled and correctly configured before data arrives. An example of configuration order is: enabling the CSI2-A receiver powers up the complex I/O connected to the external sensor. Additionally, the ISP must be configured and the source interface must be selected. For details and the order of configuration, see the programming module of the particular submodule.

When software wants to shut down a submodule:

- Software ensures that the submodule is idle. Mainly:
  - The submodule must not generate new events.
  - The submodule must not have any pending events.
  - For initiators: The submodule must stop the generation of an interface bridge transaction.
- Software clears the appropriate bit in the ISS_CLKCTRL register.
- For modules having only a master port: Hardware waits until the submodule to be disconnected asserts the MStandBy signal on its master port. It asserts MWait of the submodule.

NOTE: The ISS does not assert the MWait signal when a shutdown of the module is not requested by software.

- Hardware cuts the submodule clocks.
- Hardware clears the appropriate bit in the ISS_CLKSTAT register.

Table 14 describes the clock gating of the ISS submodule.

Table 14. ISS Submodule Clock Gating

ISS Resource	Feature On/Off Control
ISS top-level resources	Not applicable. ISS top-level resources cannot be cut. However, top-level resources support the autogating feature.
SIMCOP	ISS_CLKCTRL[0] SIMCOP
ISP	ISS_CLKCTRL[1] ISP
CSI2-A	ISS_CLKCTRL[2] CSI2_A
SC	ISS_CLKCTRL[4] SC
ISS interconnect BTE CBUFF TCTRL	These modules cannot be switched off individually. They are required for any processing performed by SIMCOP because they are on the main data path. However, they support autogating to reduce power consumption when activity is low.

When the clock of a submodule is cut and an interface bridge request for this module is received from the ISS configuration interconnect, the ISS clock manager temporarily enables the module clock to handle the access properly.

All ISS submodules are off after reset; software must enable them before they can be used.

1.2.4 ISS Reset

The ISS can accept a general software reset, propagated through all the hierarchy. This reset can be done to initialize the module and has the same effect as the hardware reset.

1. Set the ISS_HL_SYSCONFIG[0] SOFTRESET bit to 1.
2. Read the ISS_HL_SYSCONFIG[0] SOFTRESET bit to check whether it equals 0, which means the reset occurred.

If after five reads, ISS_HL_SYSCONFIG[0] SOFTRESET still returns 1, it can be assumed that an error occurred during the reset stage.

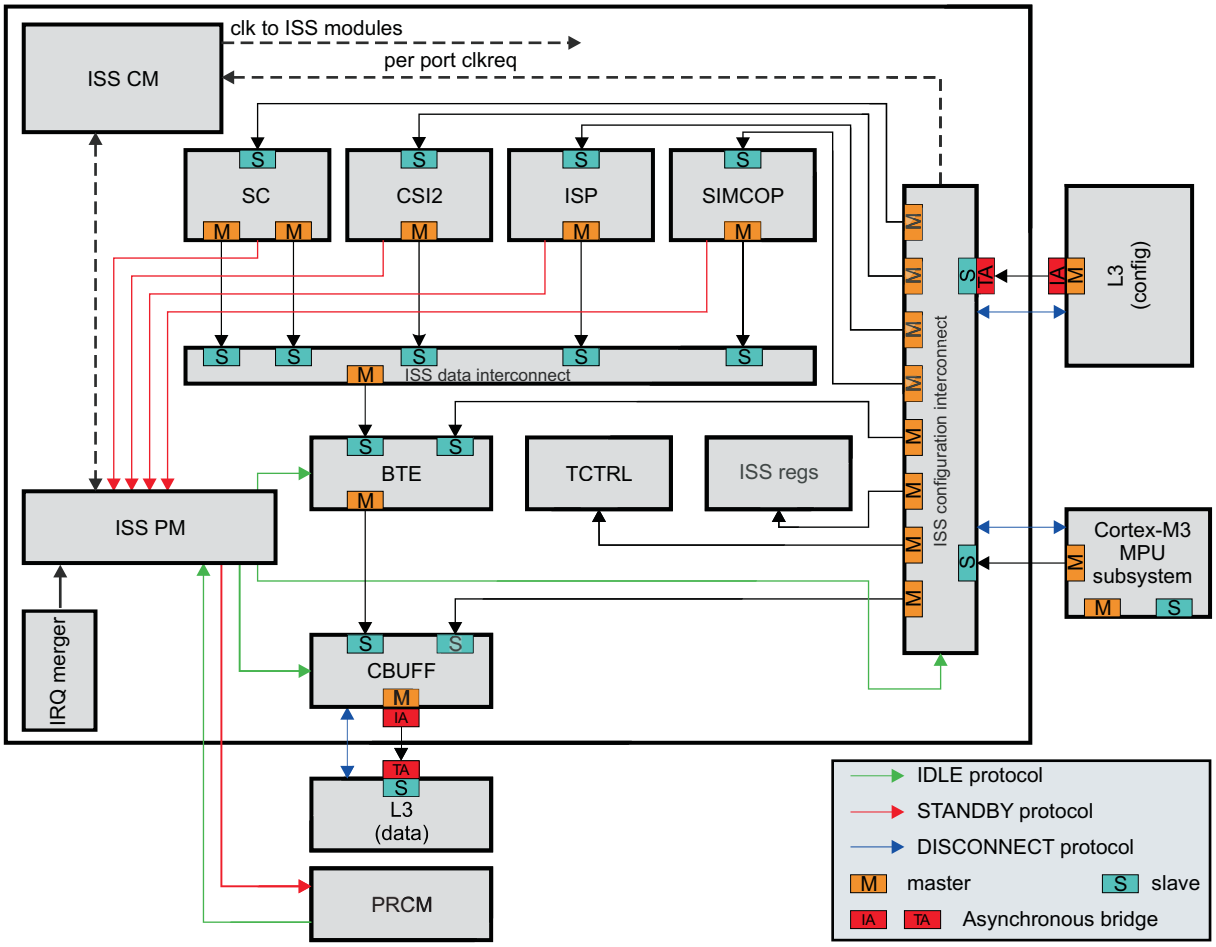
A software reset must not reset the power manager protocols (must not reset the IDLE and STANDBY generic IPs).

1.2.5 ISS Power Management

1.2.5.1 ISS Power-Management Infrastructure Overview

Figure 5 provides an overview of the ISS power management.

Figure 5. ISS Power Management



NOTE: For power savings, the PRCM module can request idle mode from the ISS. When the ISS is not functional, software must decide when the PRCM module can send the request. For more information, see *Module Level Clock Management* in the *Power, Reset, and Clock Management* chapter of the TRM.

1.2.5.2 ISS STANDBY Mechanism

The power manager receives STANDBY information from the SC, CSI2-A, ISP, and SIMCOP modules. These modules assert a standby signal when they have no more transactions to perform. The ISS power manager acknowledges by asserting a wait signal.

When those modules are in standby mode, the ISS power manager initiates a STANDBY sequence for the ISS:

1. The ISS waits while the SC, CSI2-A, ISP, and SIMCOP MStandby is asserted.
2. The ISS power manager acknowledges by asserting the SC, CSI2-A, ISP, and SIMCOP MWait signal.
3. The ISS initiators assert only the MStandBy signal when they receive responses to all sent requests. Therefore, when all initiators have asserted MStandBy, the ISS interconnect has no more pending traffic (although configurable, only nonposted writes must be used for error reporting).

4. The ISS power manager sends an IDLE request to the BTE. This IDLE request is used to drain BTE data. It does not affect the configuration port.
5. The BTE drains all transactions.
6. The BTE acknowledges the IDLE request.
7. The ISS power manager sends an IDLE request to the CBUFF. This IDLE request is used to drain CBUFF data. It does not affect the configuration port.
8. CBUFF drains all transactions.
9. CBUFF disconnects the interface master port connected to the L3 interconnect.
10. CBUFF acknowledges the IDLE request.
11. The ISS power manager asserts the MStandBy signals connected to the system PRCM module.
12. The PRCM module acknowledges by asserting the MWait signal.

A functional standby transition can be aborted when one of the SC, CSI2-A, ISP, or SIMCOP modules deasserts the MStandBy signal. The corresponding MWait signal is deasserted only when the ISS interconnect, BTE, CBUFF, and ISS interface master port are ready to receive requests.

When one of the SC, CSI2-A, ISP, or SIMCOP modules must perform accesses to the ISS interface master port, it deasserts the MStandBy signal. The ISS power manager executes the following sequence to leave the STANDBY state:

1. The ISS power manager deasserts the MStandby signal.
2. The ISS waits until the PRCM module deasserts the MWait signal.
3. The ISS power manager requests CBUFF to go into functional mode.
4. CBUFF connects the interface port.
5. The ISS power manager requests the BTE to go into functional mode.
6. The ISS power manager waits until CBUFF and BTE acknowledge functional mode.
7. The ISS deasserts the MWait signal of the module requesting access.

Abort of the standby-to-functional mode transition is not supported. The ISS power manager completes the standby-to-functional transition and then allows a new functional-to-standby transition.

Typically, the MStandby signal is used for two purposes. Software chooses one of the following behaviors through PRCM configuration:

- During blanking periods: The ISS asserts the MStandBy signal between frames when it has no more data to send. The PRCM module can use this information to switch off the L3 interconnect and save some dynamic power. However, the PRCM module is not allowed to cut the ISS functional clock in that case, because it is needed to receive the next frame.
- For ISS shutdown: The ISS asserts MStandBy when it has no more transactions to perform. The PRCM module then initiates an IDLE sequence. Once the ISS acknowledges the transition into idle mode, the PRCM module can cut the ISS clock and power.

The internal standby mode can be reached only when SC, CSI2-A, SIMCOP,ISP, ISS data interconnect, BTE, and CBUFF are in IDLE or STANDBY state. Choosing no-idle or no-standby mode for any of these modules prevents the ISS from going into STANDBY state.

Four modes for standby control are supported:

- Smart-standby-wakeup mode: This is the mode normally used. When in this mode, the ISS asserts the MStandBy signal when the MStandBy of all ISS internal initiators is asserted and the ISS data interconnect, BTE, and CBUFF are in IDLE state.
- Smart-standby mode: The ISS has no wake-up event. This mode is equivalent to smart-standby-wakeup mode.
- Force-standby mode: This is a backup mode. When in this mode, the ISS asserts MStandBy unconditionally. Software must ensure that the ISS is in a correct quiet state before programming this mode.
- No-standby mode: This is a backup mode. When in this mode, the ISS never asserts the MStandBy signal.

1.2.5.3 ISS IDLE Mechanism

The PRCM module can request the ISS to go into IDLE state when the ISS has asserted its MStandBy output.

In a normal case, software must ensure that the ISS is in a quiet state before allowing the PRCM module to send an IDLE request to the ISS:

- The ISS has no more traffic to generate.
- The ISS cannot generate any new interrupts.
- The ISS has no pending interrupts.

When an IDLE request is received from the PRCM module, the ISS power manager verifies that the ISS MWait input has been asserted and that all ISS interrupt outputs are deasserted. It then starts the STANDBY-to-IDLE transition:

1. Send IDLE request to the ISS configuration interconnect.
2. The ISS configuration interconnect requests disconnection of both ISS interface slave ports. Disconnection is done by the master:
   - It stops accepting new requests and drains currently ongoing ones.
   - It waits for completion of all ongoing transactions.
3. The ISS configuration interconnect acknowledges the IDLE transition.
4. The ISS power manager acknowledges the IDLE request from the PRCM module.
5. The PRCM module can cut the ISS clock and power.

The PRCM module first enables the ISS power and clock before requesting the ISS to go into functional state by deasserting the SIdleReq signal. The ISS power manager then executes the wake-up sequence:

1. Request the ISS interconnect to go into functional state.
2. The ISS interconnect connects the ISS slave ports.
3. The ISS interconnect acknowledges transition into functional mode.
4. The ISS power manager acknowledges transition into functional mode (ISS output SIdleAck = 00).

Four modes for IDLE control are supported:

- Smart-idle-wakeup mode [b11]: This is the mode normally used. When in this mode, the ISS acknowledges a request to go idle from the power manager after having performed all hardware operations necessary for the IAF to be in a correct quiet state.
- Smart-idle mode [b10]: This is equivalent to smart-idle-wakeup mode.
- Force-idle mode [b00]: This is a backup mode intended to be used only if smart-idle mode is bugged. When in this mode, the ISS acknowledges a request to go idle from the power manager with no hardware condition. Software must ensure that the ISS is in a correct quiet state before requesting a force-idle transition.
- No-idle mode [b01]: When in this mode, the ISS disregards any request to go idle from the power manager.

1.3 ISS Registers

Table 15 summarizes the ISS TOP register mapping.

Table 15. ISS TOP Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ISS_TOP Base Address Cortex-M3 Private Access	ISS_TOP Base Address L3 Interconnect
ISS_HL_REVISION	R	32	0x0000 0000	0x5504 0000	0x5C00 0000
ISS_HL_HWINFO	R	32	0x0000 0004	0x5504 0004	0x5C00 0004
ISS_HL_SYSCONFIG	RW	32	0x0000 0010	0x5504 0010	0x5C00 0010
RESERVED	RW	32	0x0000 001C	0x5504 001C	0x5C00 001C
ISS_HL_IRQSTATUS_RAW_i[(1)]	RW	32	0x0000 0020 + (0x10 * i)	0x5504 0020 + (0x10 * i)	0x5C00 0020 + (0x10 * i)
ISS_HL_IRQSTATUS_i[(1)]	RW	32	0x0000 0024 + (0x10 * i)	0x5504 0024 + (0x10 * i)	0x5C00 0024 + (0x10 * i)
ISS_HL_IRQENABLE_SET_i[(1)]	RW	32	0x0000 0028 + (0x10 * i)	0x5504 0028 + (0x10 * i)	0x5C00 0028 + (0x10 * i)
ISS_HL_IRQENABLE_CLR_i[(1)]	RW	32	0x0000 002C + (0x10 * i)	0x5504 002C + (0x10 * i)	0x5C00 002C + (0x10 * i)
ISS_CTRL	RW	32	0x0000 0080	0x5504 0080	0x5C00 0080
ISS_CLKCTRL	W	32	0x0000 0084	0x5504 0084	0x5C00 0084
ISS_CLKSTAT	R	32	0x0000 0088	0x5504 0088	0x5C00 0088
ISS_PM_STATUS	R	32	0x0000 008C	0x5504 008C	0x5C00 008C

(1) i = 0 to 5

1.3.1 ISS_HL_REVISION Register

Figure 6. ISS_HL_REVISION Register

31	0
REVISION	
R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 16. ISS_HL_REVISION_Register Field Description

Bit	Field	Value	Description
31-0	REVISION		IP Revision[(1)]

(1) TI internal data.

1.3.2 ISS_HL_HWINFO

Figure 7. ISS_HL_HWINFO Register

31	3	2	0
RESERVED		BRIDGE_BUFF	
R-0		R-3h	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 17. ISS_HL_HWINFO Field Descriptions

Bit	Field	Value	Description
31-3	Reserved		Reserved
2-0	BRIDGE_BUFF		Size of the reordering buffer in the SC read bridge.
		0h	8x128-bits
		1h	16x128-bits
		2h	32x128-bits
		3h	64x128-bits
		4h	128x128-bits

1.3.3 ISS_HL_SYSCONFIG

Figure 8. ISS_HL_SYSCONFIG Register

31	16
Reserved	
R-0	

15	6	5	4	3	2	1	0
Reserved		STANDBYMODE		IDLEMODE		Reserved	SOFTRESET
R-0		R/W-2h		R/W-2h		R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 18. ISS_HL_SYSCONFIG Field Descriptions

Bit	Field	Value	Description
31-6	Reserved		Reserved
5-4	STANDBYMODE		Master interface power management, standby/Wait control
		0h	Force-standby. MStandby is asserted unconditionally.
		1h	No-standby. MStandby is never asserted.
		2h	Smart-standby
		3h	Smart-standby
3-2	IDLEMODE		IDLE protocol configuration
		0h	Force-idle
		1h	No-idle
		2h	Smart-idle
		3h	Smart-idle
1	Reserved		Resrved
0	SOFTRESET		Software reset.
		W-0h	No action
		W-1h	Initiate software reset
		R-1h	Reset (software or other) ongoing
		R-0h	Reset done, no pending action

1.3.4 ISS_HL_IRQSTATUS_RAW_i

Figure 9. ISS_HL_IRQSTATUS_RAW_i_Register

31							24
Reserved							
R-0							
23						18	17	16
Reserved							HS_VS_IRQ	SC_IRQ8
R-0							R/W-0	R-0
15	14	13	12	11	10	9	8
SIMCOP_IRQ3	SIMCOP_IRQ2	SIMCOP_IRQ1	SIMCOP_IRQ0	BTE_IRQ	CBUFF_IRQ	Reserved	
R-0	R-0	R-0	R-0	R-0	R-0	R-0	
7			5	4	3	2	1	0
Reserved				CSIA_IRQ	ISP_IRQ3	ISP_IRQ2	ISP_IRQ1	ISP_IRQ0
R-0				R-0	R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19. ISS_HL_IRQSTATUS_RAW_i_Register Field Descriptions

Bit	Field	Value	Description
31-18	Reserved		Reserved
17	HS_VS_IRQ		HS or VS synchronization event. RW 0 This event is triggered if a rising or falling edge is detected on the HS or VS signal (after the video port mux). The rising or falling edge and the HS or VS signal selection is chosen with the ISS_CTRL.SYNC_DETECT bit field.
		W-0h	No action
		W-1h	Set event (debug)
		R-1h	Event pending
		R-0h	No event pending
16	SC_IRQ8		Event generated by the Stall controller
		1h	Event pending
		0h	No event pending
15	SIMCOP_IRQ3		Event generated by SIMCOP
		1h	Event pending
		0h	No event pending
14	SIMCOP_IRQ2		Event generated by SIMCOP
		1h	Event pending
		0h	No event pending
13	SIMCOP_IRQ1		Event generated by SIMCOP
		1h	Event pending
		0h	No event pending
12	SIMCOP_IRQ0 Event		Event generated by SIMCOP
		1h	Event pending
		0h	No event pending
11	BTE_IRQ		Event generated by the burst translation engine
		1h	Event pending
		0h	No event pending

Table 19. ISS_HL_IRQSTATUS_RAW_i_Register Field Descriptions (continued)

Bit	Field	Value	Description
10	CBUFF_IRQ		Event generated by the circular buffer
		1h	Event pending
		0h	No event pending
9-5	Reserved		Reserved
4	CSIA_IRQ		Event generated by the CSI2 receiver #a
		1h	Event pending
		0h	No event pending
3	ISP_IRQ3		Combined interrupt event provided by the ISP.
		1h	Event pending
		0h	No event pending
2	ISP_IRQ2		Combined interrupt event provided by the ISP.
		1h	Event pending
		0h	No event pending
1	ISP_IRQ1		Combined interrupt event provided by the ISP.
		1h	Event pending
		0h	No event pending
0	ISP_IRQ0		Combined interrupt event provided by the ISP.
		1h	Event pending
		0h	No event pending

1.3.5 ISS_HL_IRQSTATUS_i

Figure 10. ISS_HL_IRQSTATUS_i_Register

31	30	29	28	27	26	25	24
Reserved							
R-0							

23	22	21	20	19	18	17	16
Reserved						HS_VS_IRQ	SC_IRQ8
R-0						R/W1S-0	R-0

15	14	13	12	11	10	9	8
SIMCOP_IRQ3	SIMCOP_IRQ2	SIMCOP_IRQ1	SIMCOP_IRQ0	BTE_IRQ	CBUFF_IRQ	Reserved	
R-0	R-0	R-0	R-0	R-0	R-0	R-0	

7	6	5	4	3	2	1	0
Reserved			CSIA_IRQ	ISP_IRQ3	ISP_IRQ2	ISP_IRQ1	ISP_IRQ0
R-0			R-0	R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20. ISS_HL_IRQSTATUS_i_Register Field Descriptions

Bit	Field	Value	Description
31-18	Reserved		Reserved

Table 20. ISS_HL_IRQSTATUS_i_Register Field Descriptions (continued)

Bit	Field	Value	Description
17	HS_VS_IRQ		HS or VS synchronization event. This event is triggered if a rising or falling edge is detected on the HS or VS signal (after the video port mux). The rising or falling edge and the HS or VS signal selection is chosen with the ISS_CTRL.SYNC_DETECT bit field.
		W-0h	No action
		W-1h	Clear (raw) event
		R-0h	No (enabled) event pending
		R-1h	Event pending
16	SC_IRQ8		Event generated by the Stall controller
		0h	No (enabled) event pending
		1h	Event pending
15	SIMCOP_IRQ3		Event generated by SIMCOP
		0h	No (enabled) event pending
		1h	Event pending
14	SIMCOP_IRQ2		Event generated by SIMCOP
		0h	No (enabled) event pending
		1h	Event pending
13	SIMCOP_IRQ1		Event generated by SIMCOP
		0h	No (enabled) event pending
		1h	Event pending
12	SIMCOP_IRQ0 Event		Event generated by SIMCOP
		0h	No (enabled) event pending
		1h	Event pending
11	BTE_IRQ		Event generated by the burst translation engine
		0h	No (enabled) event pending
		1h	Event pending
10	CBUFF_IRQ		Event generated by the circular buffer
		0h	No (enabled) event pending
		1h	Event pending
9-5	Reserved		Reserved
4	CSIA_IRQ		Event generated by the CSI2 receiver #a
		0h	No (enabled) event pending
		1h	Event pending
3	ISP_IRQ3		Combined interrupt event provided by the ISP.
		0h	No (enabled) event pending
		1h	Event pending
2	ISP_IRQ2		Combined interrupt event provided by the ISP.
		0h	No (enabled) event pending
		1h	Event pending
1	ISP_IRQ1		Combined interrupt event provided by the ISP.
		0h	No (enabled) event pending
		1h	Event pending
0	ISP_IRQ0		Combined interrupt event provided by the ISP.
		0h	No (enabled) event pending
		1h	Event pending

1.3.6 ISS_HL_IRQENABLE_SET_i

Figure 11. ISS_HL_IRQENABLE_SET_i Register

31	30	29	28	27	26	25	24
Reserved							
R-0							

23	22	21	20	19	18	17	16
Reserved						HS_VS_IRQ	SC_IRQ8
R-0						R/W1S-0	R/W1S-0

15	14	13	12	11	10	9	8
SIMCOP_IRQ3	SIMCOP_IRQ2	SIMCOP_IRQ1	SIMCOP_IRQ0	BTE_IRQ	CBUFF_IRQ	Reserved	
R/W1S-0	R/W1S-0	R/W1S-0	R/W1S-0	R/W1S-0	R/W1S-0	R-0	

7	6	5	4	3	2	1	0
Reserved			CSIA_IRQ	ISP_IRQ3	ISP_IRQ2	ISP_IRQ1	ISP_IRQ0
R-0			R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21. ISS_HL_IRQENABLE_SET_i Field Descriptions

Bit	Field	Value	Description
31-18	Reserved		Reserved
17	HS_VS_IRQ		HS or VS synchronization event. This event is triggered if a rising or falling edge is detected on the HS or VS signal (after the video port mux). The rising or falling edge and the HS or VS signal selection is chosen with the ISS_CTRL.SYNC_DETECT bit field.
		W-0h	No action
		W-1h	Enable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled
16	SC_IRQ8		Event generated by the Stall controller
		W-0h	No action
		W-1h	Enable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled
15	SIMCOP_IRQ3		Event generated by SIMCOP
		W-0h	No action
		W-1h	Enable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled
14	SIMCOP_IRQ2		Event generated by SIMCOP
		W-0h	No action
		W-1h	Enable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled
13	SIMCOP_IRQ1		Event generated by SIMCOP
		W-0h	No action
		W-1h	Enable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled

Table 21. ISS_HL_IRQENABLE_SET_i Field Descriptions (continued)

Bit	Field	Value	Description
12	SIMCOP_IRQ0		Event generated by SIMCOP
		W-0h	No action
		W-1h	Enable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled
11	BTE_IRQ		Event generated by the burst translation engine
		W-0h	No action
		W-1h	Enable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled
10	CBUFF_IRQ		Event generated by the circular buffer
		W-0h	No action
		W-1h	Enable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled
9-5	Reserved		Reserved
4	CSIA_IRQ		Event generated by the CSI2 receiver #a RW
		W-0h	No action
		W-1h	Enable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled
3	ISP_IRQ3		Combined interrupt event provided by the ISP.
		W-0h	No action
		W-1h	Enable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled
2	ISP_IRQ2		Combined interrupt event provided by the ISP.
		W-0h	No action
		W-1h	Enable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled
1	ISP_IRQ1		Combined interrupt event provided by the ISP.
		W-0h	No action
		W-1h	Enable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled
0	ISP_IRQ0		Combined interrupt event provided by the ISP.
		W-0h	No action
		W-1h	Enable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled

1.3.7 ISS_HL_IRQENABLE_CLR_i

Figure 12. ISS_HL_IRQENABLE_CLR_i Register

31							24
Reserved							
R-0							
23						18	17	16
Reserved							HS_VS_IRQ	SC_IRQ8
R-0							R/W1C-0	R/W1S-0
15	14	13	12	11	10	9	8
SIMCOP_IRQ3	SIMCOP_IRQ2	SIMCOP_IRQ1	SIMCOP_IRQ0	BTE_IRQ	CBUFF_IRQ	Reserved	
R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R-0	
7	6	5	4	3	2	1	0
Reserved			CSIA_IRQ	ISP_IRQ3	ISP_IRQ2	ISP_IRQ1	ISP_IRQ0
R-0			R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset; R/W1 to set

Table 22. ISS_HL_IRQENABLE_CLR_i Register Field Descriptions

Bit	Field	Value	Description
31-18	Reserved		Reserved
17	HS_VS_IRQ		HS or VS synchronization event. This event is triggered if a rising or falling edge is detected on the HS or VS signal (after the video port mux). The rising or falling edge and the HS or VS signal selection is chosen with the ISS_CTRL.SYNC_DETECT bit field.
		W-0h	No action
		W-1h	Disable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled
16	SC_IRQ8		Event generated by the Stall controller
		W-0h	No action
		W-1h	Enable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled
15	SIMCOP_IRQ3		Event generated by SIMCOP
		W-0h	No action
		W-1h	Disable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled
14	SIMCOP_IRQ2		Event generated by SIMCOP
		W-0h	No action
		W-1h	Disable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled
13	SIMCOP_IRQ1		Event generated by SIMCOP
		W-0h	No action
		W-1h	Disable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled

Table 22. ISS_HL_IRQENABLE_CLR_i Register Field Descriptions (continued)

Bit	Field	Value	Description
12	SIMCOP_IRQ0		Event generated by SIMCOP
		W-0h	No action
		W-1h	Disable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled
11	BTE_IRQ		Event generated by the BTE
		W-0h	No action
		W-1h	Disable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled
10	CBUFF_IRQ		Event generated by the CBUFF
		W-0h	No action
		W-1h	Disable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled
9-5	Reserved		Reserved
4	CSIA_IRQ		Event generated by the CSI2 receiver a
		W-0h	No action
		W-1h	Disable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled
3	ISP_IRQ3		Combined interrupt event provided by the ISP.
		W-0h	No action
		W-1h	Disable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled
2	ISP_IRQ2		Combined interrupt event provided by the ISP.
		W-0h	No action
		W-1h	Disable interrupt
		R-0h	Interrupt disabled(masked)
		R-1h	Interrupt enabled
1	ISP_IRQ1		Combined interrupt event provided by the ISP.
		W-0h	No action
		W-1h	Disable interrupt
		R-0h	Interrupt disabled (masked)
		R-1h	Interrupt enabled
0	ISP_IRQ0		Combined interrupt event provided by the ISP.
		W-0h	No action
		W-1h	Disable interrupt
		R-0h	Interrupt enabled
		R-1h	Interrupt disabled (masked)

1.3.8 ISS_CTRL

Figure 13. ISS_CTRL Register

31 28	27 24	23 20	19 16
Reserved	CSI2_A_TAG_CNT	SCW_TAG_CNT	SCR_TAG_CNT
R-0	R/W-0	R/W-0	R/W-0

15 6	5 4	3 2	1 0
Reserved	ISS_CLK_DIV	INPUT_SEL	SYNC_ DETECT
R-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 23. ISS_CTRL Register Field Descriptions

Bit	Field	Value	Description
31-28	Reserved		Reserved
27-24	CSI2_A_TAG_CNT	0h	Defines the maximum number of tags that could be used by the CSI2 a write bridge. **Note:** Tag count must be set to 16 for best performance
23-20	SCW_TAG_CNT	0h	Defines the maximum number of tags that could be used by the SC write bridge **Note:** Tag count must be set to 16 for best performance.
19-16	SCR_TAG_CNT	0h	Defines the maximum number of tags that could be used by the SC read bridge **Note:** Tag count must be set to 16 for best performance.
15-6	Reserved		Reserved
5-4	ISS_CLK_DIV		ISS functional clock division CLK refers to the input clock provided to the ISS. FCLK is the functional clock provided to ISS top level and submodules. CFGCLK is the clock used for the configuration network
		0h	FCLK=CLK CFGCLK=CLK/2
		1h	FCLK=CLK/2 CFGCLK=CLK/4
		2h	FCLK=CLK/4 CFGCLK=CLK/8
		3h	Reserved
3-2	INPUT_SEL		Selects ISP input
		0h	CSI2-A
		2h	SC
		3h	Parallel interface
1-0	SYNC_DETECT		Chooses among rising and falling edge for the HS_VS_IRQ synchronization even
		0h	HS falling edge
		1h	HS raising edge
		2h	VS falling edge
		3h	VS raising edge

1.3.9 ISS_CLKCTRL

Figure 14. ISS_CLKCTRL Register

31	30	29	28	27	16
VPORT3_CLK	VPORT2_CLK	Reserved	VPORT0_CLK	Reserved	
R-1	R-1	R-1	R-1	R-0	

15	5	4	3	2	1	0
Reserved		SC	Reserved	CSI2_A	ISP	SIMCOP
R-0		W-0	R-0	W-0	W-0	W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 24. ISS_CLKCTRL Register Field Descriptions

Bit	Field	Value	Description
31	VPORT3_CLK		Enables the pixel clock from the parallel interface
		0h	Disabled
		1h	Enabled
30	VPORT2_CLK		Enables the pixel clock from the Stall controller
		0h	Disabled
		1h	Enabled
29	Reserved		Reserved
28	VPORT0_CLK		Enables the pixel clock from the CSI2_A protocol engine
		0h	Disabled
		1h	Enabled
27-5	Reserved		Reserved
4	SC		Stall controller
		0h	Request shutdown of the submodule. No effect if the submodule clock is already off.
		1h	Request enable of the submodule. No effect if the submodule clock is already off.
3	Reserved		Reserved
2	CSI2_A		CSI2_A
		0h	Request shutdown of the submodule. No effect if the submodule clock is already off.
		1h	Request enable of the submodule. No effect if the submodule clock is already off.
1	ISP		ISP
		0h	Request shutdown of the submodule. No effect if the submodule clock is already off.
		1h	Request enable of the submodule. No effect if the submodule clock is already off.
0	SIMCOP		SIMCOP
		0h	Request shutdown of the submodule. No effect if the submodule clock is already off.
		1h	Request enable of the submodule. No effect if the submodule clock is already off.

1.3.10 ISS_CLKSTAT

Figure 15. ISS_CLKSTAT Register

31	30	29	28	27	16
VPORT3_CLK	VPORT2_CLK	Reserved	VPORT0_CLK	Reserved	
R-1	R-1	R-1	R-1	R-0	

15	5	4	3	2	1	0
Reserved		SC	Reserved	CSI2_A	ISP	SIMCOP
R-0		R-0	R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 25. ISS_CLKSTAT Register Field Descriptions

Bit	Field	Value	Description
31	VPORT3_CLK		Status of the pixel clock from the parallel interface
		0h	Disabled
		1h	Enabled
30	VPORT2_CLK		Status of the pixel clock from the Stall controller
		0h	Disabled
		1h	Enabled
29	Reserved		Reserved
28	VPORT0_CLK		Status of the pixel clock from the CSI2_A protocol engine.
		0h	Disabled
		1h	Enabled
27-5	Reserved		Reserved
4	SC		Stall Controller
		0h	The submodule is off.
		1h	The submodule is on.
3	Reserved		Reserved
2	CSI2_A		CSI2_A
		0h	The submodule is off.
		1h	The submodule is on.
1	ISP		ISP
		0h	The submodule is off.
		1h	The submodule is on.
0	SIMCOP		SIMCOP
		0h	The submodule is off.
		1h	The submodule is on.

1.3.11 ISS_PM_STATUS

Figure 16. ISS_PM_STATUS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R-0															

15	14	13	12	11	10	9	8
Reserved		CBUFF_PM		BTE_PM		SIMCOP_PM	
R-0		R-0		R-0		R-0	

7	6	5	4	3	2	1	0
ISP_PM		SC_PM		Reserved		CSI2_A_PM	
R-0		R-0		R-0		R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 26. ISS_PM_STATUS Register Field Descriptions

Bit	Field	Value	Description
31-14	Reserved		Reserved
13-12	CBUFF_PM		Power status of the CBUFF.
		0h	Idle
		1h	Transition
		2h	Functional
11-10	BTE_PM		Power status of the BTE.
		0h	Idle
		1h	Transition
		2h	Funcitonal
9-8	SIMCOP_PM		Power status of the SIMCOP.
		0h	Standby
		1h	Transition
		2h	Functional
7-6	ISP_PM		Power status of the ISP.
		0h	Standby
		1h	Transition
		2h	Functional
5-4	SC_PM		Power status of the CS.
		0h	Standby
		1h	Transition
		2h	Functional
3-2	Reserved		Reserved
1-0	CSI2_A_PM		Power status of the CSI2 module A
		0h	Standby
		1h	Transition
		2h	Functional

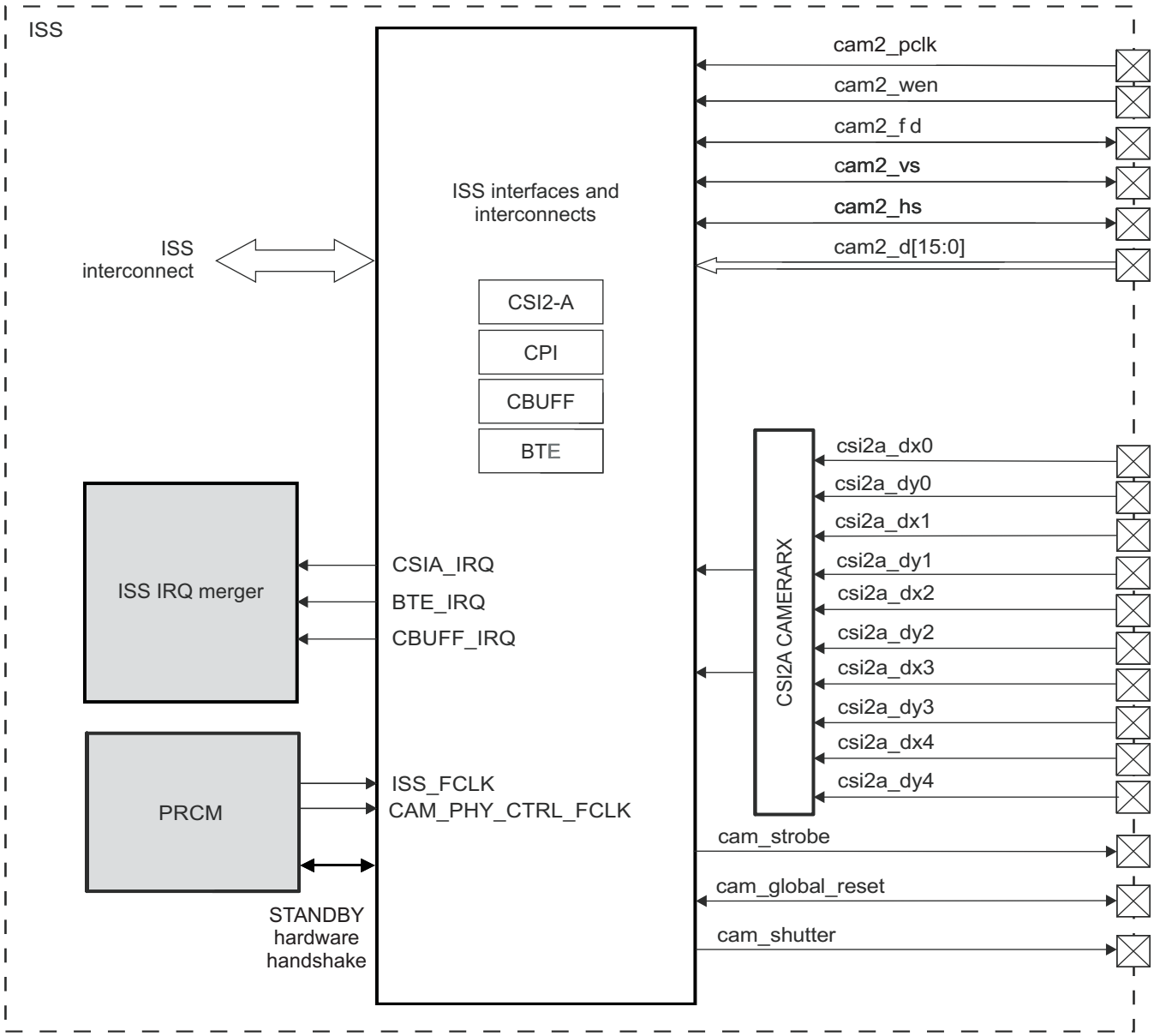
2 ISS Interfaces

2.1 ISS Interfaces Overview

Along with the submodules, the ISS has a serial camera interface and a parallel interface. The serial interface (CSI2-A) supports MIPI® CSI2 protocol with four data lanes. The parallel interface (CPI) supports up to 16 data lanes. All interfaces can use the image signal processor (ISP), but not concurrently. When one interface uses the ISP, the other must send data to memory. However, the ISP can still be used to process this data in memory-to-memory. Time multiplex processing is also possible.

Figure 17 shows the ISS interfaces and interconnects.

Figure 17. ISS Interfaces and Interconnects Highlights



NOTE: In the device, the number 2 in the cam2 interface name does not mean there are two camera parallel interfaces available in the device. The cam2 interface name defines the camera parallel interface available in the device versus the legacy device.

2.1.1 ISS Interface Features

The camera subsystem supports the following features:

- System interfaces and interconnects:
  - Two 32-bit-wide configuration interfaces:
    - Interface to Cortex™-M3 microprocessor unit (MPU): Synchronous to the functional clock
    - Interface to Cortex™-A8 MPU and system direct memory access (EDMA): Asynchronous from the functional clock
  - 128-bit-wide data interface to level 3 (L3) interconnect: Asynchronous from the functional clock
    - Shared interface level 2 (L2) (256KB) in IVA-HD module for hardware encoding
    - Outside connection using the L3 interconnect through the TILER to the synchronous dynamic random access memory (SDRAM) controller (SDRC), which acts as the primary interface between the SDRAM and the ISS functional block
  - The ISS has a local interconnect that connects all modules inside the ISS.
  - BTE:
    - Tightly coupled with the TILER to support efficient rotation
  - CBUFF:
    - Maps a linear space into a circular buffer
    - The buffer is physically located in system memory.
  - TCTRL:
    - Control signal generation for flash prestrobe and strobe
    - Camera global reset control
- Camera interfaces:
  - CSI2 camera interfaces: CSI2-A
    - Transfer pixels and data received by the CSI2 digital physical layer receiver to the system memory or to the ISP
    - Use unidirectional data link
    - CSI2-A supports four configurable data links in addition to the clock signaling.
    - Maximum data rate of 1 Gbps per data lane
    - Data merger for 2-, 3-, or 4-data lane configuration
    - Maximum data rate of 1 Gbps per data lane, possible configurations are:
      - One data lane: 1000 Mbps
      - Two data lanes: 2 × 1000 Mbps
      - Three data lanes: 3 × 1000 Mbps
      - Four data lanes: 4 × 824 Mbps
    - Error detection and correction by the protocol engine
    - Direct memory access (DMA) engine integrated with dedicated first in first out (FIFO)
    - One-dimensional (1D) and two-dimensional (2D) addressing mode
    - Burst support
    - Streaming burst support (64- or 32-bit)
    - Eight contexts to support eight dedicated configurations of virtual channel ID and data types
    - Ping-pong mechanism for double-buffering
    - All primary and secondary MIPI-defined formats are supported.
    - Conversion of the RGB formats
    - On-the-fly differential pulse code modulation (DPCM) decompression
    - On-the-fly image cropping and A-law/DPCM compression

- Parallel interface (CPI)
  - 16 bits wide
  - up to 162 MPix/s
  - BT656 and SYNC mode (HS, VS, FIELD, WEN)
- System memory data read back port (supported by the Stall controller)
  - RAW 6, 7, 8, 10, 12, 14, 16 formats supported
  - DPCM and A-law decompression
  - Supports image cropping for compressed or uncompressed data

NOTE: Rotated DPCM data is not supported.

2.2 ISS Interfaces Environment

2.2.1 ISS Interfaces Signal Descriptions

Table 27 summarizes the I/O signals.

Table 27. ISS I/O Description

Signal Name	I/O[1]	Description	Serial Mode CSI2	Parallel Mode CPI
cam_strobe	O	Flash strobe control signal	x	
cam_shutter	O	Mechanical shutter control signal	x	
cam_global_reset	I/O	Global reset release shutter signal	x	
csi2a_dx0	I	Serial CSI2-A mode: Differential clock positive input	x	
csi2a_dy0	I	Serial CSI2-A mode: Differential clock negative input	x	
csi2a_dx1	I	Serial CSI2-A mode: Differential data lane positive input	x	
csi2a_dy1	I	Serial CSI2-A mode: Differential data lane negative input	x	
csi2a_dx2	I	Serial CSI2-A mode: Differential data lane positive input	x	
csi2a_dy2	I	Serial CSI2-A mode: Differential data lane negative input	x	
csi2a_dx3	I	Serial CSI2-A mode: Differential data lane positive input	x	
csi2a_dy3	I	Serial CSI2-A mode: Differential data lane negative input	x	
csi2a_dx4	I	Serial CSI2-A mode: Differential data lane positive input	x	
csi2a_dy4	I	Serial CSI2-A mode: Differential data lane negative input	x	
cam2_pclk	I	Parallel mode pixel clock input		x
cam2_wen	I	Parallel mode write enable signal input		x
cam2_fld	I/O	Parallel mode pixel clock field signal		x
cam2_vs	I/O	Parallel mode vertical frame synchronization		x
cam2_hs	I/O	Parallel mode horizontal frame synchronization		x
cam2_d[15:0]	I	Parallel mode data lanes (16 signals)		x

(1) I = Input; O = Output

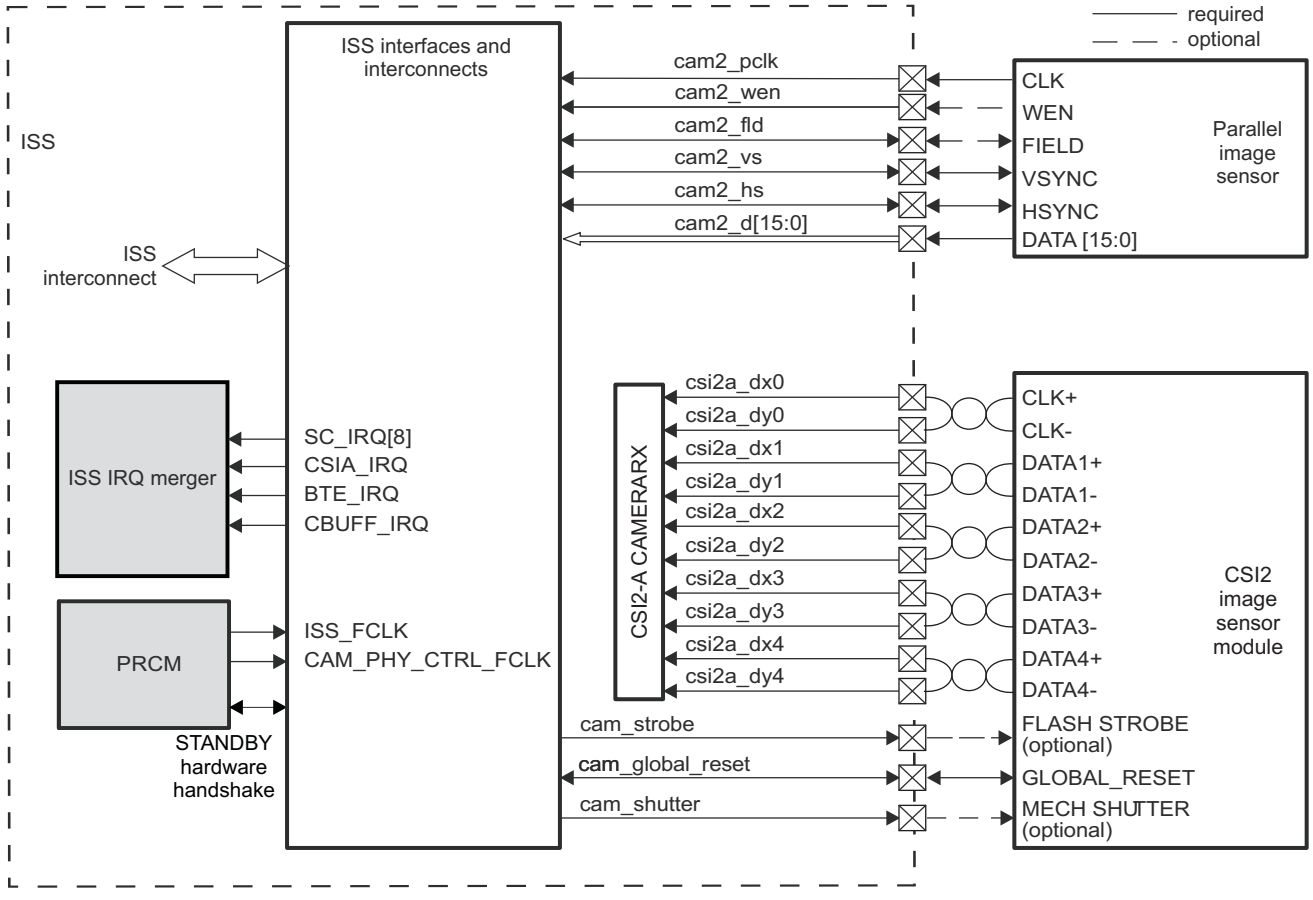
NOTE: Lane polarity can be changed in complex I/O. For more information, see Section 2.2.2, *ISS Interface Modes*.

NOTE: At least one data lane must be configured for using the CSI2-A interface. The signals are also configurable from the control module. Thus, they are not required to be at a certain location to act as clock or data; this can be configured.

2.2.2 ISS Interface Modes

The camera subsystem can manage a parallel and serial interface. Depending on the configuration of the shared pins, two of the interfaces can be active at the same time. However, only one data flow can use the ISP. Moreover, if the parallel interface is used data from it goes to ISP and the other used interface must send it to memory. Figure 18 shows an example block diagram of the interface configuration. Each serial port clock lane is configurable. The MIPI CSI2 protocol requires only a clock lane setup. The data lane configuration is optional. For more information, see Section 2.6.1.1, *ISS CSI2 Protocol and Data Format*.

Figure 18. ISS CSI2-A Serial Interface and Parallel Interface Configuration



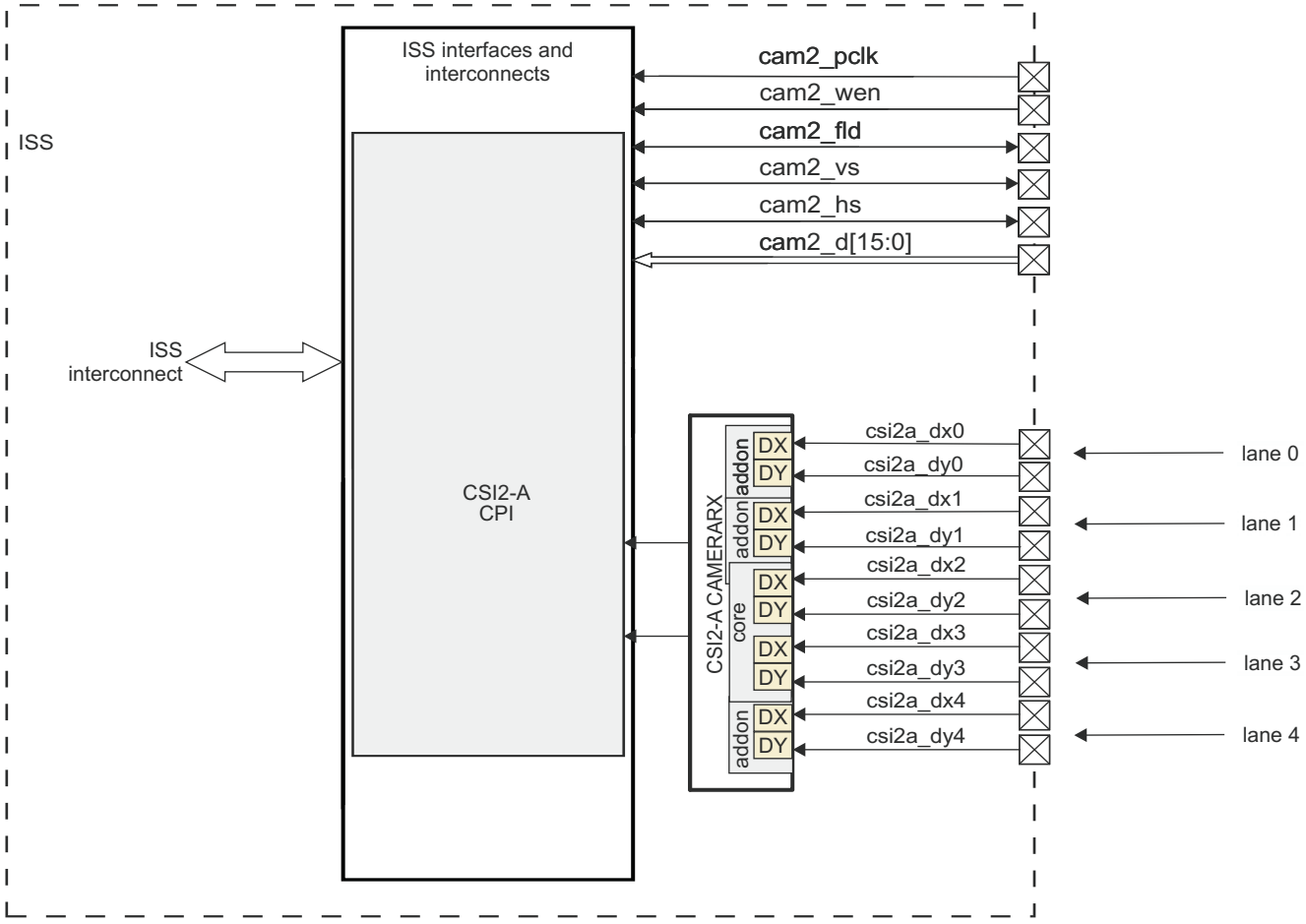
2.3 ISS CSI2 PHY

2.3.1 ISS CSI2 PHY Overview

A MIPI D-PHY-compliant CAMERARX PHY receiver immediately before the ISS interfaces acts as a physical connection and configuration of clock/data lanes with external sensors. CAMERARX PHY supports up to four configurations, depending on the required number of D-PHY data lane external sensors. The receivers are compatible with the *MIPI D-PHY Specification v0.92*. The selection of CAMERARX in D-PHY mode or parallel mode must be done before reset and not on the fly.

The PHY is controlled and must be configured first from the control module for pad configuration. The PHY is developed as two modules: CAMERARX_CORE and CAMERARX_ADDON. The CSI2-A CAMERARX contains four data lanes, as shown in Figure 19.

Figure 19. ISS Interfaces CSI2-A PHY Diagram Four D-PHY Data Lane Configuration



NOTE: LANE 4 can be used only as a data lane, never as a clock lane. All other configurations are possible. Also, a speed restriction is present when lane 4 is used; then all data lanes perform at up to 824 Mbps instead of 1000 Mbps.

CSI2-A CAMERARX represents the overall PHY solution for connecting external sensors to feed the ISS. The MIPI D-PHY function can support up to four data lane modules and one clock lane module. Reverse direction escape mode is not supported. The lane module polarity and positions are configurable; that is, any lane module can be chosen as the clock lane module, and the DX/DY data pad for each lane module can be configured as DP or DN pins defined. The configuration and the selection of D-PHY mode, data/clock, or data/strobes are done through the control module. The only exception is the four-data-lane use case, in which one corner lane is allowed to be only a data lane.

2.3.2 ISS CSI2 PHY Functional Description

2.3.2.1 ISS CSI2 PHY Functional Configuration

The CSI2 PHY converts the bitstream, divided into 1 to 4 serial data lanes, into a bitstream compatible with the CSI2 receiver and one clock lane.

The CSI2_COMPLEXIO_IRQSTATUS register logs complex I/O events of the following types:

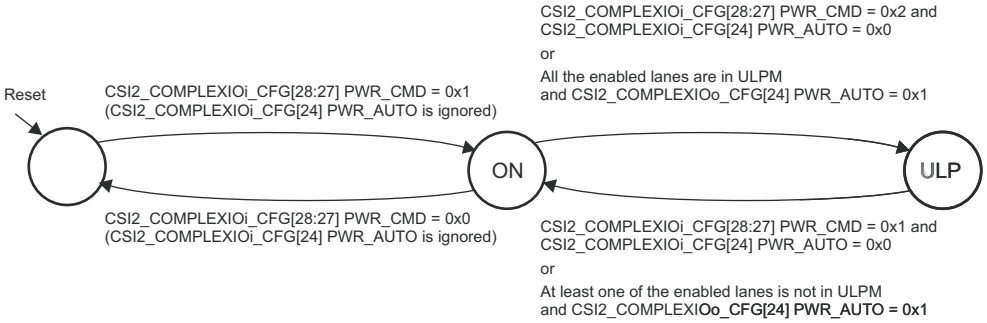
- Line power-state change (all lanes in ultralow-power mode [ULPM], at least one lane exits ULPM, etc.)
- Error on one lane

The CSI2 receiver embeds two registers to configure/read some complex I/O parameters:

- The REGISTER0 register detects clock miss with respect to the *MIPI D-PHY Specification v0.92* and control timing.
- The REGISTERO register reports completion of reset on the different parts of the module and configures the timing parameters.
- The CS12_COMPLEXIO_CFG registers contain the PWR_AUTO and PWR_CMD bit fields, which affect the power management of the complex I/O.

The complex I/O has three power modes: on, off, and ultralow power (ULP). These modes can reflect the ON or ULP states of the five differential lines if the CS12_COMPLEXIO_CFG [24] PWR_AUTO bit is set to 1. If the PWR_AUTO bit is at reset value (0), the complex I/O power state is controlled by the CS12_COMPLEXIO_CFG [28:27] PWR_CMD bit field, which directly defines the power state. Figure 20 shows the complex I/O power finite state-machine (FSM).

Figure 20. ISS CSI2 Complex I/O Power FSM



Another register, CS12_TIMING is used to control the power state of the complex I/O modules with regard to the differential line state. This register controls the mode of the complex I/O (RxMode and NoRxMode) and the delay between the differential lanes in STOP state and the complex I/O on NoRxMode. The CS12_TIMING[15] FORCE_RX_MODE_IO1 bit sets the complex I/O in RxMode or NoRxMode (stopped mode). The FORCE_RX_MODE_IO bit is automatically reset to 0 by hardware when the counter ends and the FSM returns to NoRxMode. Three bits (CS12_TIMING[14] STOP_STATE_X16_IO1, CS12_TIMING[13] STOP_STATE_X4_IO1, and the CS12_TIMING[12:0] STOP_STATE_COUNTER_IO1 bit field) configure the delay between line stop mode and complex I/O stop mode. The delay represents the number of functional clock (ISS_FCLK) cycles and can be calculated as follows:

Total delay in ISS_FCLK cycle = CS12_TIMING.STOP_STATE_COUNTER_IO x (1+CS12_TIMING.STOP_STATE_X16_IO x 15) x (1+CS12_TIMING.STOP_STATE_X4_IO x 3).

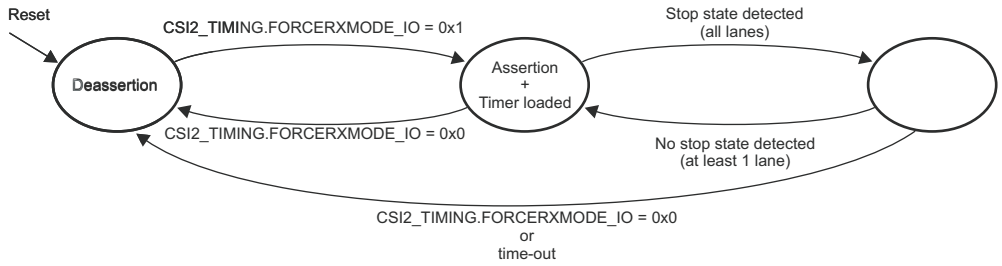
Table 28 lists the possible values of the delay, in terms of the ISS_FCLK cycles, depending on the values of the STOP_STATE_X16_IO and STOP_STATE_X4_IO bits.

Table 28. ISS CSI2 Possible Time-Out Value for RxMode Counter

STOP_STATE_X16_IO	STOP_STATE_X4_IO	Possible Delay Value (in Functional Clock Cycles)
0x0	0x0	8191 (with step of 1)
0x0	0x1	32764 (with step of 4)
0x1	0x0	131056 (with step of 16)
0x1	0x1	524224 (with step of 64)

The FORCERXMODE signal is used at initialization time (complex I/O). Figure 21 describes the ForceRxMode and StopState FSM to assert and deassert the FORCERXMODE signal and to monitor STOPSTATE from the complex I/O.

Figure 21. ISS CSI2 RxMode and StopState FSM



2.3.2.2 ISS CSI2 PHY and Link Initialization Sequence

The MIPI D-PHY initialization sequence is not implemented within CAMERARX. The CSI2-A receiver is expected to coordinate the PHY initialization. The controller must ensure that the PHY is held in RESET/WAIT for RX mode until the D-PHY transmitter is powered up and the link comes to the defined state. The controller can use the STOPSTATE and FORCERXMODE signals of CAMERARX for this purpose. STOPSTATE indicates the line states, while FORCERXMODE forces the receiver state-machine into "wait for stop state." One possible initialization sequence is:

To fully initialize the CSIPHY, perform the following steps:

1. Configure all CSI2 receiver registers to be ready to receive signals/data from the CSIPHY:
   (a) Configure all needed CSI2 registers:
      (i) Set CSI2_COMPLEXIO_CFG[18:16] DATA4_POSITION.
      (ii) Set CSI2_COMPLEXIO_CFG[14:12] DATA3_POSITION.
      (iii) Set CSI2_COMPLEXIO_CFG[10:8] DATA2_POSITION.
      (iv) Set CSI2_COMPLEXIO_CFG[6:4] DATA1_POSITION.
      (v) Set CSI2_COMPLEXIO_CFG[2:0] CLOCK_POSITION.
      (vi) Set the CONTROL_CAMERA_RX[17:16] CAMERARX_CSI21_CAMMODE.

CAUTION

This must be done before the CSIPHY is active.

2. CSIPHY and link initialization sequence:
   (a) Deassert the CSIPHY reset.
      (i) Set CSI2_COMPLEXIO_CFG[30] RESET_CTRL to 0x1.

      The following registers can be set only after deasserting the CSIPHY reset and before asserting the FORCERXMODE signal:
      - REGISTER0
      - REGISTER1
      - REGISTER2

   (b) Assert the FORCERXMODE signal:
      (i) Set CSI2_TIMING[15] FORCE_RX_MODE_IO1 to 0x1.

   (c) Connect pulldown on link (UP/DN) by asserting the respective PIPD* signals (PIPD* = 0):

      For CSI2-A CAMERARX pulldown on signals through padconf registers:
      - csi21_dx4:
        – CONTROL_CORE_PAD0_CSI21_DX4_PAD1_CSI21_DY4[8] CSI21_DX4_INPUTENABLE = 0x1
        – CONTROL_CORE_PAD0_CSI21_DX4_PAD1_CSI21_DY4[4] CSI21_DX4_PULLTYPESELECT = 0x0
        – CONTROL_CORE_PAD0_CSI21_DX4_PAD1_CSI21_DY4[3] CSI21_DX4_PULLUDENABLE = 0x1
      - csi21_dy4:
        – CONTROL_CORE_PAD0_CSI21_DX4_PAD1_CSI21_DY4[24] CSI21_DY4_INPUTENABLE = 0x1
        – CONTROL_CORE_PAD0_CSI21_DX4_PAD1_CSI21_DY4[20] CSI21_DY4_PULLTYPESELECT = 0x0
        – CONTROL_CORE_PAD0_CSI21_DX4_PAD1_CSI21_DY4[19] CSI21_DY4_PULLUDENABLE = 0x1
      - csi21_dx3:
        – CONTROL_CORE_PAD0_CSI21_DX3_PAD1_CSI21_DY3[8] CSI21_DX3_INPUTENABLE = 0x1
        – CONTROL_CORE_PAD0_CSI21_DX3_PAD1_CSI21_DY3[4] CSI21_DX3_PULLTYPESELECT= 0x0
        – CONTROL_CORE_PAD0_CSI21_DX3_PAD1_CSI21_DY3[3] CSI21_DX3_PULLUDENABLE = 0x1
      - csi21_dy3:
        – CONTROL_CORE_PAD0_CSI21_DX3_PAD1_CSI21_DY3[24] CSI21_DY3_INPUTENABLE = 0x1
        – CONTROL_CORE_PAD0_CSI21_DX3_PAD1_CSI21_DY3[20] CSI21_DY3_PULLTYPESELECT= 0x0
        – CONTROL_CORE_PAD0_CSI21_DX3_PAD1_CSI21_DY3[19] CSI21_DY3_PULLUDENABLE = 0x1
      - csi21_dx2:
        – CONTROL_CORE_PAD0_CSI21_DX2_PAD1_CSI21_DY2[8] CSI21_DX2_INPUTENABLE = 0x1
        – CONTROL_CORE_PAD0_CSI21_DX2_PAD1_CSI21_DY2[4] CSI21_DX2_PULLTYPESELECT= 0x0
        – CONTROL_CORE_PAD0_CSI21_DX2_PAD1_CSI21_DY2[3] CSI21_DX2_PULLUDENABLE = 0x1
      - csi21_dy2:
        – CONTROL_CORE_PAD0_CSI21_DX2_PAD1_CSI21_DY2[24] CSI21_DY2_INPUTENABLE = 0x1

- CONTROL_CORE_PAD0_CSI21_DX2_PAD1_CSI21_DY2[20] CSI21_DY2_PULLTYPESELECT= 0x0
- CONTROL_CORE_PAD0_CSI21_DX2_PAD1_CSI21_DY3[19] CSI21_DY2_PULLUDENABLE = 0x1

- csi21_dx1:
  - CONTROL_CORE_PAD0_CSI21_DX1_PAD1_CSI21_DY1[8] CSI21_DX1_INPUTENABLE = 0x1
  - CONTROL_CORE_PAD0_CSI21_DX1_PAD1_CSI21_DY1[4] CSI21_DX1_PULLTYPESELECT= 0x0
  - CONTROL_CORE_PAD0_CSI21_DX1_PAD1_CSI21_DY1[3] CSI21_DX1_PULLUDENABLE = 0x1
- csi21_dy1:
  - CONTROL_CORE_PAD0_CSI21_DX1_PAD1_CSI21_DY1[24] CSI21_DY1_INPUTENABLE = 0x1
  - CONTROL_CORE_PAD0_CSI21_DX1_PAD1_CSI21_DY1[20] CSI21_DY1_PULLTYPESELECT= 0x0
  - CONTROL_CORE_PAD0_CSI21_DX1_PAD1_CSI21_DY1[19] CSI21_DY1_PULLUDENABLE = 0x1
- csi21_dx0:
  - CONTROL_CORE_PAD0_CSI21_DX0_PAD1_CSI21_DY0[8] CSI21_DX0_INPUTENABLE = 0x1
  - CONTROL_CORE_PAD0_CSI21_DX0_PAD1_CSI21_DY0[4] CSI21_DX0_PULLTYPESELECT= 0x0
  - CONTROL_CORE_PAD0_CSI21_DX0_PAD1_CSI21_DY0[3] CSI21_DX0_PULLUDENABLE = 0x1
- csi21_dy0:
  - CONTROL_CORE_PAD0_CSI21_DX0_PAD1_CSI21_DY0[24] CSI21_DY0_INPUTENABLE = 0x1
  - CONTROL_CORE_PAD0_CSI21_DX0_PAD1_CSI21_DY0[20] CSI21_DY0_PULLTYPESELECT= 0x0
  - CONTROL_CORE_PAD0_CSI21_DX0_PAD1_CSI21_DY0[19] CSI21_DY0_PULLUDENABLE = 0x1

(d) Power up the CSIPHY:

(i) Set CSI2_COMPLEXIO_CFG[28:27] PWR_CMD to 0x1.

(e) Check whether the state status reaches the ON state:

- CSI2_COMPLEXIO_CFG[26:25] PWR_STATUS = 0x1

(f) Wait for STOPSTATE = 1 (for all enabled lane modules):

(i) The timer is set through the CSI2_TIMING[14:0] bit field. The reset value can be kept.

(ii) Wait until CSI2_TIMING[15] FORCE_RX_MODE_IO1 = 0x0. It is automatically put at 0 when all enabled lanes are in STOPSTATE and the timer is finished.

(g) Release PIPD* (= 1).

For CSI2-A CAMERARX pullup on signals through padconf registers:

- csi21_dx4:
  - CONTROL_CORE_PAD0_CSI21_DX4_PAD1_CSI21_DY4[4] CSI21_DX4_PULLTYPESELECT = 0x1
- csi21_dy4:
  - CONTROL_CORE_PAD0_CSI21_DX4_PAD1_CSI21_DY4[20] CSI21_DY4_PULLTYPESELECT = 0x1
- csi21_dx3:
  - CONTROL_CORE_PAD0_CSI21_DX3_PAD1_CSI21_DY3[4]

CSI21_DX3_PULLTYPESELECT = 0x1

- csi21_dy3:
  - CONTROL_CORE_PAD0_CSI21_DX3_PAD1_CSI21_DY3[20] CSI21_DY3_PULLTYPESELECT = 0x1
- csi21_dx2:
  - CONTROL_CORE_PAD0_CSI21_DX2_PAD1_CSI21_DY2[4] CSI21_DX2_PULLTYPESELECT = 0x1
- csi21_dy2:
  - CONTROL_CORE_PAD0_CSI21_DX2_PAD1_CSI21_DY2[20] CSI21_DY2_PULLTYPESELECT = 0x1
- csi21_dx1:
  - CONTROL_CORE_PAD0_CSI21_DX1_PAD1_CSI21_DY1[4] CSI21_DX1_PULLTYPESELECT = 0x1
- csi21_dy1:
  - CONTROL_CORE_PAD0_CSI21_DX1_PAD1_CSI21_DY1[20] CSI21_DY1_PULLTYPESELECT = 0x1
- csi21_dx0:
  - CONTROL_CORE_PAD0_CSI21_DX0_PAD1_CSI21_DY0[4] CSI21_DX0_PULLTYPESELECT = 0x1
- csi21_dy0:
  - CONTROL_CORE_PAD0_CSI21_DX0_PAD1_CSI21_DY0[20] CSI21_DY0_PULLTYPESELECT = 0x1

3. The CSIPHY is initialized and ready/active in CSI2 mode.

2.3.2.3 ISS CSI PHY Error Signals

In D-PHY mode, the CSIPHY supports the following error detection and signaling to the associated receiver:

- ERRSOTHS: Flags 1-bit errors in the HS start of transmission synchronization pattern. In this error scenario, the CSIPHY continues to receive the data and pass it to the receiver, but confidence in the data may be low, because of the 1-bit error seen in sync. This signal, if asserted, is high for one cycle of RXBYTECLKHS.
- ERRSOTSYNCHS: Flags multiple bit errors in the HS start of transmission synchronization pattern. In this case, the CSIPHY cannot achieve proper synchronization and does not pass the received data to the receiver. This signal, if asserted, is high for one cycle of RXBYTECLKHS.
- ERRCONTROL: Flags the control sequence error; that is, when the LP sequence observed on line is not recognized as a valid control sequence. This signal, if asserted, is high until the next change in the state of the LP line.
- ERRESC: Flags the escape entry error; that is, when the escape entry sequence is unrecognized. This signal, if asserted, is high until the next change in the state of the LP line.
- ERRSYNCESC: Flags the low-power data transmission synchronization error. This error is flagged if the number of bits received during a low-power data transmission is not a multiple of 8 bits. This signal, if asserted, is high until the next change in the state of the LP line. In case the number of received bits is 1 less than a multiple of 8, RXVALIDESC is also asserted together with ERRSYNCESC, and an erroneous data byte is output on RXDATAESC. In other cases of this error, RXVALIDESC is not asserted and an erroneous data byte is not sent out.

2.3.3 ISS CSI2 PHY Registers

Table 29 lists the CSI2 PHY instance.

Table 29. ISS CSI2 PHY Instance Summary

Module Name	Base Address Cortex-M3 Private Access	Base Address L3 Interconnect	Size
ISS_CAMERARX_CORE1	0x5504 1170	0x5C00 1170	32 bytes

Table 30 summarizes the ISS CSI2 PHY register mapping.

Table 30. ISS CSI2 PHY Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ISS_CAMERARX_CORE1 Base Address Cortex-M3 Private Access	ISS_CAMERARX_CORE1 Base Address L3 Interconnect
REGISTER0	RW	32	0x0000 0000	0x5504 1170	0x5C00 1170
REGISTER1	RW	32	0x0000 0004	0x5504 1174	0x5C00 1174
REGISTER2	RW	32	0x0000 0008	0x5504 1178	0x5C00 1178

2.3.3.1 ISS_CSI2_PHY_REGISTER0

Figure 22. ISS_CSI2_PHY_REGISTER0

31 25	24	23 16	15 8	7 0
Reserved	HSCLOCKCONFIG	Reserved	THS_TERM	THS_SETTLE
R-0	R/W-0	R-0	R/W-0x24	R/W-0x27

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 31. ISS_CSI2_PHY_REGISTER0 Field Descriptions

Bit	Field	Value	Description
31-25	Reserved		Reserved
24	HSCLOCKCONFIG	0h	Disable clock missing detector
23-16	Reserved		Reserved
15-8	THS_TERM	4h	THS_TERM timing parameter in multiples of DDR clock Effective time for enabling of termination = synchronizer delay + timer delay + LPRX delay + combinational routing delay ~ (1–2)* DDRCLK + THS-TERM + ~ (1 –15) ns Programmed value = ceil(12.5 / DDR clock period) –1
7-0	THS_SETTLE	27h	THS_SETTLE timing parameter in multiples of DDR clock frequency Effective THS_SETTLE seen on line (starting to look for sync pattern) = synchronizer delay + timer delay + LPRX delay + combinational routing delay – pipeline delay in HS data path. ~ (1–2)* DDRCLK + THS-SETTLE + ~ (1–15) ns –1*DDRCLK Programmed value = ceil(90 ns / DDR clock period) + 3

2.3.3.2 ISS_CSI2_PHY_REGISTER1

Figure 23. ISS_CSI2_PHY_REGISTER1

31 30	29 28	27 26	25	24 18	17 16
Reserved	RESET_DONE_ STATUS	Reserved	CLOCK_MISS_ DETECTOR_STATUS	TCLK_TERM	DPHY_HS_SYNC_ PATTERN
R-0	R-0x0	R-0	R-0	R/W-0x00	R/W-0xB8

15 10	9 8	7 0
DPHY_HS_SYNC_PATTERN	CTRLCLK_DIV_ FACTOR	TCLK_SETTLE
R/W-0xB8	R/W-0x1	R/W-0x0E

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 32. ISS_CSI2_PHY_REGISTER1 Field Descriptions

Bit	Field	Value	Description
31-30	Reserved		Reserved
29-28	RESET_DONE_ STATUS		Reset done read bits. 28: RESETDONERXBYTECLK Note: BYTECLK is provided to the ISS CSI2-A 29: RESETDONECTRLCLK Note: This is the CAM_PHY_CTRL_FCLK provided to the PHY from the PRCM module.
27-26	Reserved		Reserved
25	CLOCK_MISS_ DETECTOR_ STATUSW	 0h 1h	Clock missing detector Clock missing detector successful Error in clock missing detector.
24-18	TCLK_TERM	0h	TCLK_TERM timing parameter in multiples of CTRLCLK Effective time for enabling of termination = synchronizer delay + timer delay + LPRX delay + combinational routing delay ~ (1–2)* CTRLCLK + TCLK_TERM + ~ (1–15) ns Programmed value = ceil(9.5 / CTRLCLK period) – 1
17-10	DPHY_HS_SYNC_ PATTERN	B8h	DPHY mode HS sync pattern in byte order (reverse of received order) See ISS CSI PHY Error Signals.
9-8	CTRLCLK_DIV_ FACTOR	1h	Divide factor for CTRLCLK for CLKMISS detector
7-0	TCLK_SETTLE	Eh	TCLK_SETTLE timing parameter in multiples of CTRLCLK Clock Effective TCLK_SETTLE = synchronizer delay + timer delay + LPRX delay + combinational routing delay ~ (1–2)* CTRLCLK + Tclk-settle + ~ (1 –15) ns Programmed value = max[3, ceil(155 ns/CTRLCLK period) –1]

2.3.3.3 ISS_CSI2_PHY_REGISTER2

Figure 24. ISS_CSI2_PHY_REGISTER2

31 30	29 28	27 26	25 24	23 16
TRIGGER_CMD_ RXTRIGESC0	TRIGGER_CMD_ RXTRIGESC1	TRIGGER_CMD_ RXTRIGESC2	TRIGGER_CMD_ RXTRIGESC3	Reserved
R/W-0x0	R/W-0x0	R/W-0x0	R/W-0x0	R-0

15 0
Reserved
R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

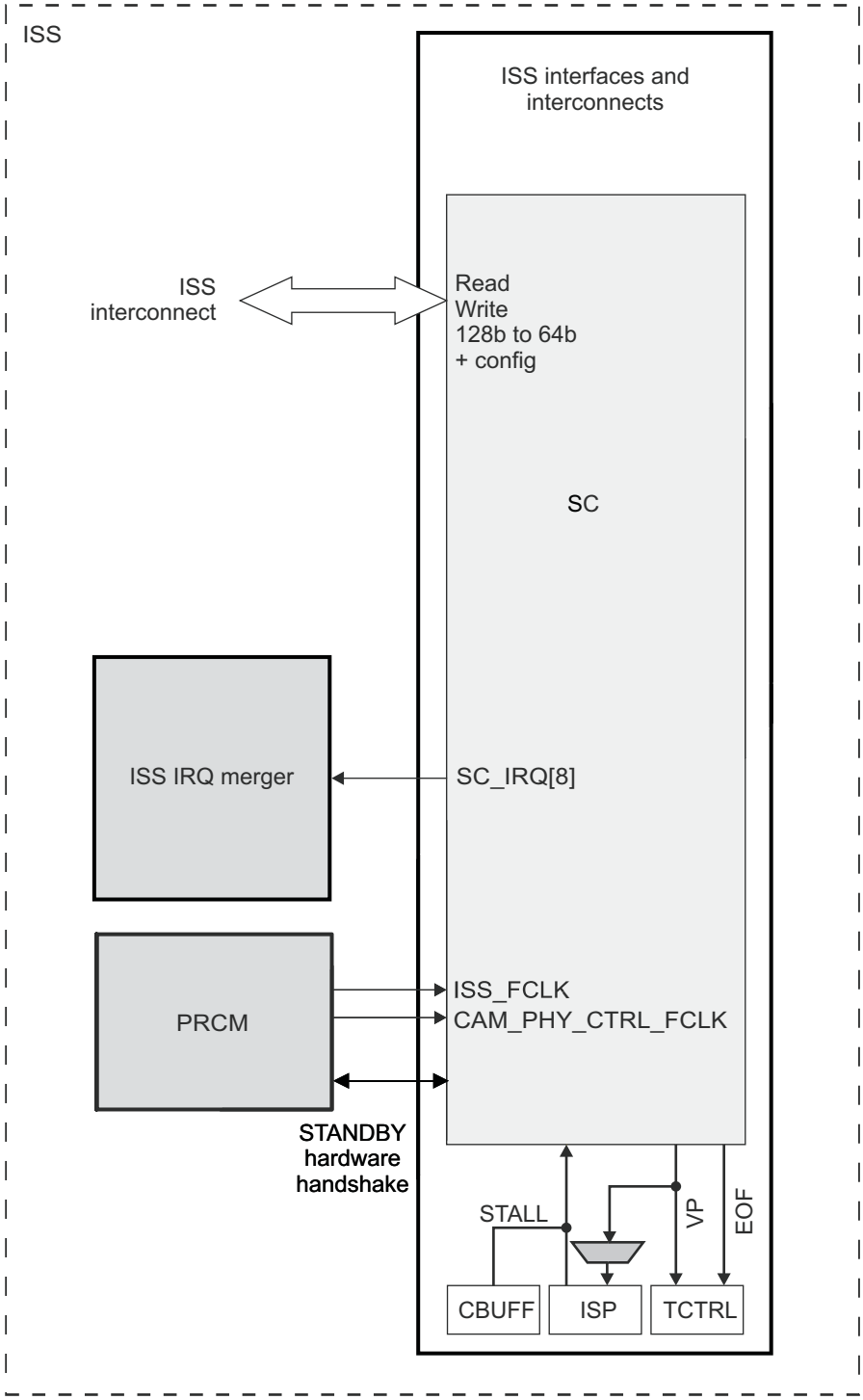
Table 33. ISS_CSI2_PHY_REGISTER2 Field Descriptions

Bit	Field	Value	Description
31-30	TRIGGER_CMD_ RXTRIGESC0	0h	Mapping of Trigger escape entry command to PPI output RXTRIGGERESC0
29-28	TRIGGER_CMD_ RXTRIGESC1	0h	Mapping of Trigger escape entry command to PPI output RXTRIGGERESC1
27-26	TRIGGER_CMD_ RXTRIGESC2	0h	Mapping of Trigger escape entry command to PPI output RXTRIGGERESC2
25-24	TRIGGER_CMD_ RXTRIGESC3	0h	Mapping of Trigger escape entry command to PPI output RW 0x0 RXTRIGGERESC3
23-0	Reserved		Reserved

2.4 ISS Stall Controller Integration

Figure 25 is a top-level block diagram of the Stall controller interface. The stall controller can get its data from memory using a dedicated interface read master port.

Figure 25. ISS Stall Controller Integration

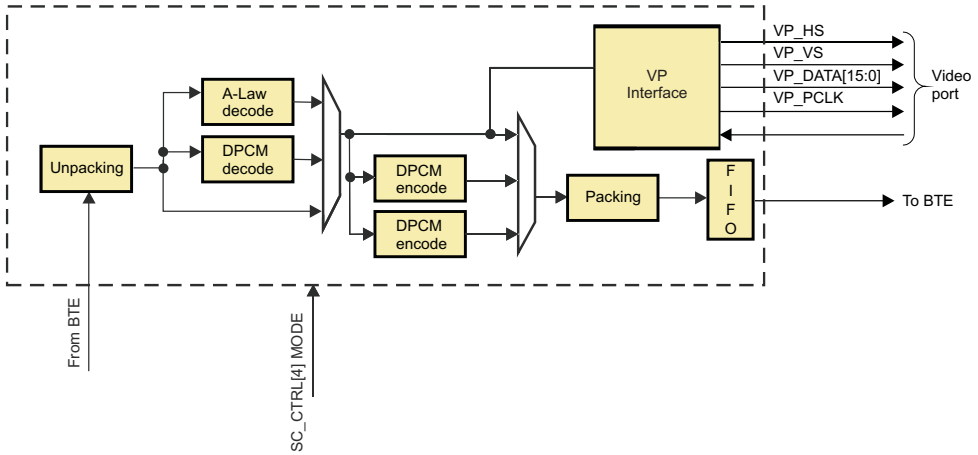


2.5 ISS SC Functional Description

Figure 26 is the top-level block diagram of the Stall controller. The SC can also read data from memory getting burst packets from the BTE. For read data from memory, see Section 2.5.3, *ISS SC Memory Read Channel*.

The Staller controller video port interface is connected to the video preprocessing hardware.

Figure 26. ISS SC Receiver Block Diagram



2.5.1 ISS SC VP Interface

Table 34 summarizes the video interface signals. The video interface connects the Stall controller module to the video preprocessing hardware (ISP). The interface is connected to a 16-bit video port. On the other side of the video port is the ISIF inside of the ISP. The ISIF also uses the signals listed in Table 34 to synchronize pixel data sent to it by the Stall controller.

Table 34. ISS SC Video interface Signals

Pin	Type[1]	Description
VP_HS	O	Line trigger output signal
VP_VS	O	Frame trigger output signal
VP_DATA[15:0]	O	Parallel output data: bits 0 to 15
VP_PCLK	O	Video port pixel clock. The frequency can be configured.
VP_STALL	I	Stalls data flow

(1) I = Input; O = Output

When data is sent to the video port, the data flow can be stalled by asserting the VP_STALL signal. Doing so does not overflow internal FIFOs: the SC module adapts its read rate automatically.

The response time to the VP_STALL signal must not exceed two cycles: when VP_STALL is asserted, the SC module can send 0, 1, or 2 pixels to the video port.

VP_STALL is asserted and deasserted synchronous to the functional clock.

The pixel clock is generated from the functional clock. Clock pulses are gated based on the selected clock division factor and pixel availability. In other words, software must set the Stall controller to ensure that the pixel clock:

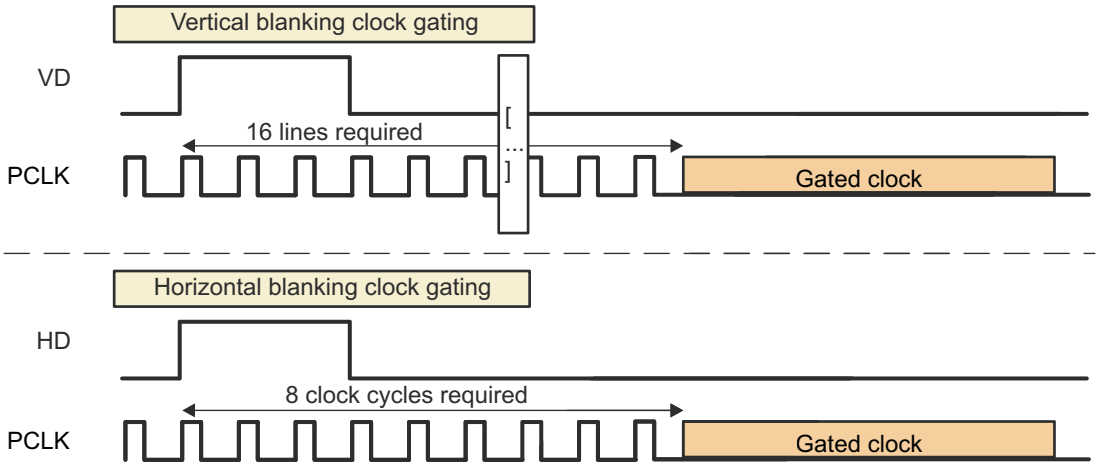
- Never exceeds what the ISP can support: the top value set in the SC_CTRL[31:15] FRACDIV bit field
- Is sent only when valid pixels or blanking data must be sent

Also, software must set the number of clock pulses during horizontal blanking periods using the SC_CTRL1[1:0] BLANKING bit field.

- The Stall controller stops generating horizontal blanking clock pulses when the number of pulses defined in the SC_CTRL1[1:0] BLANKING bit field are generated.
- The Stall controller ensures that the number of horizontal blanking pulses defined in the SC_CTRL1[1:0] BLANKING bit field is received. The Stall controller stops generating horizontal blanking clock pulses when then number of pulses defined in the SC_CTRL1[1:0] BLANKING bit field is generated.

NOTE: To work properly, the ISP requires a minimum of eight clock cycles in the horizontal blanking period and a minimum of 16 lines in the vertical blanking period. The pixel clock can be gated only when these intervals are respected. This is required to flush the pipeline of the different ISP modules. Figure 27 shows VP_PCLK gating during blanking periods.

Figure 27. ISS SC VP_PCLK Gating During Blanking Periods



Vertical blanking generation is controlled through the SC_CTRL[9] VP_CLK_FORCE_ON bit. The VP_PCLK clock is enabled during vertical blanking periods when this bit is set. This pushes pixels through the ISP processing pipe. It is needed, for example, when the ISP resizer uses the averager. Otherwise, hardware ensures only that at least four clock pulses are generated before the first pixel of each frame. It may be necessary for the clock to keep running after the frame end to flush internal pipelines. In that case, an interrupt request (IRQ) or status bit is typically present in the attached hardware that indicates when the VP_PCLK clock is no longer needed (for example, an end of processing interrupt). The module leaves the vertical blanking state when new data is received from the memory read channel.

The configured pixel clock is used for active and blanking periods.

Table 35 shows how RAW and YUV data is sent over the video port. The data is sent to the ISIF if ISP is used. For the ISIF details about video port data, see Section 3 *ISS: ISP*.

Table 35. ISS SC Video Port Data Mapping

Format	Video Port DATA[15:0]															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW6	0	0	0	0	0	0	0	0	0	0	R5	R4	R3	R2	R1	R0
RAW7	0	0	0	0	0	0	0	0	0	R6	R5	R4	R3	R2	R1	R0
RAW8	0	0	0	0	0	0	0	0	R7	R6	R5	R4	R3	R2	R1	R0
RAW10	0	0	0	0	0	0	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
RAW12	0	0	0	0	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
RAW14	0	0	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
RAW16	R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0

Table 35. ISS SC Video Port Data Mapping (continued)

Format	Video Port DATA[15:0]															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
YUV4:2:2	0	0	0	0	0	0	0	0	U7	U6	U5	U4	U3	U2	U1	U0
	0	0	0	0	0	0	0	0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
	0	0	0	0	0	0	0	0	V7	V6	V5	V4	V3	V2	V1	V0
	0	0	0	0	0	0	0	0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

2.5.2 ISS SC Data Compression

The data compression technique used is DPCM and A-Law.

The Stall controller performs on-the-fly compression and decompression. The compressed/decompressed data is passed to the video preprocessing hardware or stored in memory.

The data compression method is lossy and does not require any information outside the current encoded/decoded line. This means that all the image lines can be encoded and decoded separately.

Two different predictors are used:

- The simple predictor

  This predictor uses only the previous same color component value as a prediction value. Therefore, only 2-pixel memory is required.
- The advanced predictor

  This predictor uses four previous pixel values, when the prediction value is evaluated. This means that also the other color component values are used, when the prediction value is defined.

The preferable use is that the simple predictor is used with 10 bits to 8 bits or 12 bits to 8 bits conversion (10810 or 12812) and the advanced predictor is used with 10 bits to 7 bits and 10 bits to 6 bits conversions (10710 and 10610). The advanced predictor gives slightly better prediction for pixel value; thus, the image quality can be improved with it. Because the simple predictor is very simple, the processing power and memory requirements are reduced with it, when the image quality is already high enough.

Select the predictor with the SC_LCx_CTRL[10] DPCM_PRED bit.

2.5.3 ISS SC Memory Read Channel

The memory channel can perform the following operations:

- Reads data from memory. It is unpacked and DPCM or A-Law decompressed if necessary.
- Sends data to the video preprocessing hardware
- Sends data back to memory. It can be DPCM or A-Law compressed and packed before it is sent to memory.

Table 36 summarizes supported modes for memory-to-memory operations.

NOTE: Video port and memory destinations are mutually exclusive.

Table 36. ISS SC Memory-to-Memory Supported Operations

Memory Input	Memory Output																					
	RAW 6	RAW 6+PACK	RAW 6+DPCM	RAW 6+PACK+DPCM	RAW 6+DPCM_ADV	RAW 6+PACK+DPCM_ADV	RAW 7	RAW 7+PACK	RAW 7+DPCM	RAW 7+PACK+DPCM	RAW 7+DPCM_ADV	RAW 7+PACK+DPCM_ADV	RAW 8	RAW 8+DPCM	RAW 8+DPCM12	RAW 8+ALAW10	RAW 10	RAW 10+PACK	RAW 12	RAW 12+PACK	RAW 14	RAW 16
RAW6																						
RAW6 + PACK																						
RAW6 + DPCM																	X	X				
RAW6 + PACK + DPCM																	X	X				
RAW6 + DPCM_ADV																	X	X				
RAW6 + PACK + DPMC_ADV																	X	X				
RAW7																						
RAW7 + PACK																						
RAW7 + DPCM																	X	X				
RAW7 + PACK + DPCM																	X	X				
RAW7 + DPCM_ADV																	X	X				
RAW7 + PACK + DPMC_ADV																	X	X				
RAW8																						
RAW8 + DPCM																	X	X				
RAW8 + DPCM12																			X	X		

Table 36. ISS SC Memory-to-Memory Supported Operations (continued)

Memory Input	Memory Output																					
	RAW 6	RAW 6+PACK	RAW 6+DPCM	RAW 6+PACK+DPCM	RAW 6+DPCM_ADV	RAW 6+PACK+DPCM_ADV	RAW 7	RAW 7+PACK	RAW 7+DPCM	RAW 7+PACK+DPCM	RAW 7+DPCM_ADV	RAW 7+PACK+DPCM_ADV	RAW 8	RAW 8+DPCM	RAW 8+DPCM12	RAW 8+ALAW10	RAW 10	RAW 10+PACK	RAW 12	RAW 12+PACK	RAW 14	RAW 16
RAW8 + ALAW10																	X	X				
RAW10			X	X	X	X			X	X	X	X		X								
RAW10 + PACK			X	X	X	X			X	X	X	X		X								
RAW12																						
RAW12 + PACK																						
RAW14																						
RAW16																						

Table 37 summarizes supported modes for memory-to-video port operations.

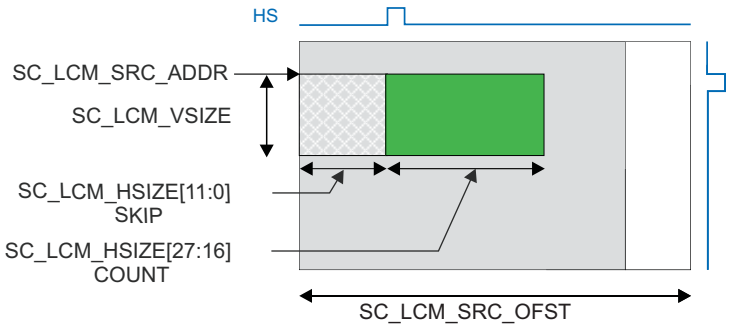
Table 37. ISS SC Memory-to-Video Port Supported Formats

Memory Input	Video Port Output						
	RAW6	RAW7	RAW8	RAW10	RAW12	RAW14	RAW16
RAW6				X			
RAW6 + PACK				X			
RAW6 +DPCM				X			
RAW6 + PACK + DPCM				X			
RAW6 + DPCM_ADV				X			
RAW6 + DPCM_ADV + PACK				X			
RAW7				X			
RAW7 + PACK				X			
RAW7 + DPCM				X			
RAW7 + PACK + DPCM				X			
RAW7 + DPCM_ADV				X			
RAW7 + DPCM_ADV + PACK				X			
RAW8				X			
RAW8 + DPCM				X			
RAW8 + DPCM12					X		
RAW10				X			
RAW10 + PACK				X			
RAW12					X		
RAW12 + PACK					X		
RAW14						X	
RAW16							X

2.5.3.1 ISS SC Read Data From Memory

Figure 28 shows the data organization in memory.

Figure 28. ISS SC Data Organization in Memory



The user chooses the start address and the line length using the SC_LCM_SRC_ADDR and SC_LCM_SRC_OFST registers. The image start address normally must point to the beginning of a line because of packing constraints. However, it does not necessarily point to the first line of the frame in memory. The SC_LCM_VSIZE[27:16] COUNT bit field specifies the total line count to be read from memory.

It is also possible to skip a certain pixel count SC_LCM_HSIZE[11:0] SKIP) from the start of the line. Thus, they are not sent to the video port or back to memory. The SC_LCM_HSIZE[27:16] COUNT bit field specifies the horizontal size of the image. The pixels after the right boundary of the image are not read from memory.

When data are sent to the video port, throughput is imposed by the selected VP_PCLK. Otherwise, it is imposed by the selected interconnect read port clock. The interconnect read rate can be throttled (limiting the maximum data read speed for memory-to-memory operation) using the SC_LCM_CTRL[4:3] READ_THROTTLE bit field. Therefore, it is possible to read the unused data at a higher rate than the used video port data rate. This provides better performance than framing the image in the video preprocessing hardware.

The data storage format in memory is defined by the SC_LCM_CTRL[18:16] SRC_FORMAT and SC_LCM_CTRL[23] SRC_PACK bit fields.

Not all I/O format combinations are valid. For more information, see Table 36 and Table 37.

Figure 29 shows how data are packed in memory. Pixel order (left to right in the image) is alphabetical (a, b, c). Therefore, data storage is little endian.

Figure 29. ISS SC Data Organization Packing in Memory

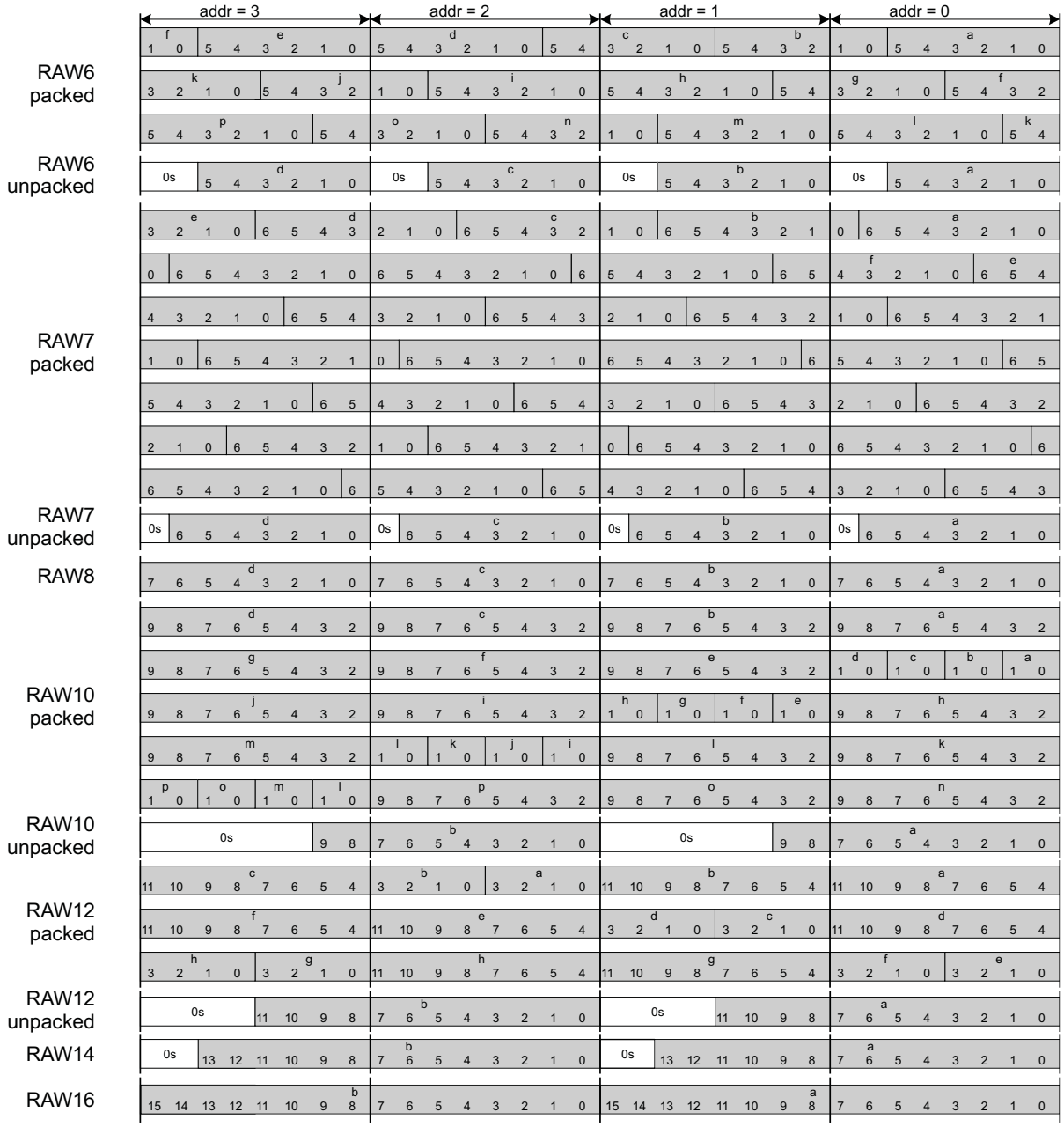


Table 38 summarizes the storage reduction versus unpacked format and image width restrictions when data packing is used. The image width applies to the data width multiple in pixels that must be stored to have storage reduction. Moreover, because each address is 8 bits long, the percentage shows how many bits out of 8 are to be packed in the empty space from another pixel address. A pixel is selected and split. One of the parts is put into another pixel address empty space. When unpacked, each pixel bit is stored continuously again.

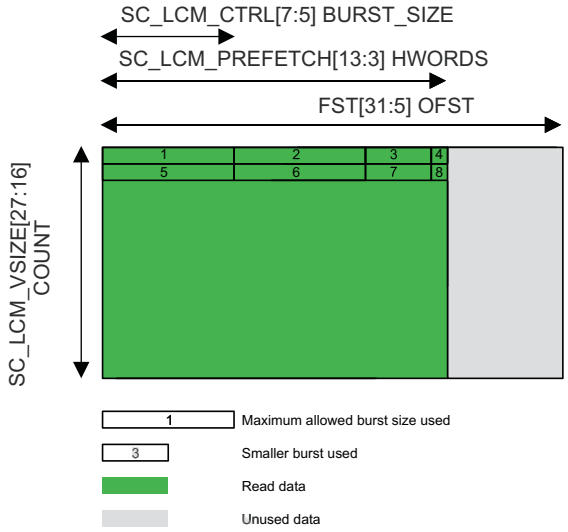
Table 38. ISS SC Data Packing Benefit and Constraints

	Bits Per Pixel		Storage Reduction	Width Multiple (Pixels)
	Packed	Unpacked		
RAW6	6	8	25%	16
RAW7	7	8	13%	3
RAW8	8	8	0%	4
RAW10	10	16	38%	16
RAW12	12	16	25%	8

2.5.3.2 ISS SC Memory Read Port Burst Generation

Hardware always uses the largest possible burst size according to the setup. The amount of data read from memory can be higher than what is actually used by the Stall controller. Only full 64-bit burst words are read back from memory. Figure 30 shows the data organization and the relationship between the different parameters controlling the burst generation.

Figure 30. ISS SC Data Organization in Memory



NOTE:

- A minimum burst size of 2 must be selected for correct operation.
- The HWORDS bit field must be even for correct operation.

The SC_LCM_SRC_ADDR register address of the first data to read is aligned to a 32-byte boundary. The read port fetches SC_LCM_PREFETCH[13:3] HWORDS of 64-bit words per line using the longest possible burst computed from the SC_LCM_CTRL[7:5] BURST_SIZE bit field and the remaining data to be fetched. Burst size of 128 bytes is preferred. When the Stall controller is configured to fetch more data than required, extra data are dropped internally.

2.5.3.3 ISS SC Video Port

The video port always receives unpacked data. It can be enabled using the SC_LCM_CTRL[2] DST_PORT bit. Its clock can be selected with the SC_LCM_CTRL[31:15] FRACDIV bit field and gated or not during frame blanking periods using the SC_LCM_CTRL[9] VP_CLK_FORCE_ON bit.

The data format used by the video port is defined by the SC_LCM_CTRL[26:24] DST_FORMAT bit field. For a list of supported modes, see Table 37.

2.5.3.4 ISS SC Encode, Pack, and Store Data

This stage is used only when data are sent to memory. Memory destination is selected using the SC_LCM_CTRL[2] DST_PORT bit. The output data format is defined by the SC_LCM_CTRL[26:24] DST_FORMAT bit field and the SC_LCM_CTRL[31] DST_PACK bit. Not all possible combinations are supported; see Table 39 for details.

The destination address and offset for the output data of the memory channel are set by the SC_LCM_DST_ADDR and SC_LCM_DST_OFSTC registers.

Because of alignment constraints on the interconnect port, the output image width restrictions in Table 39 apply.

Table 39. ISS SC Output Width Restrictions in Memory-to-Memory Operation

Format	Bits per Pixel	Width Multiple of(1)	Note
RAW6	8	1	Full 32-bit words are written at the end of the line. This last word can eventually include 0s.
RAW6 packed	6	1	
RAW7	8	1	
RAW7 packed	7	1	
RAW8	8	1	
RAW10	16	1	
RAW10 packed	10	16	
RAW12	16	1	Same constraints as RAW8
RAW12 packed	12	8	

(1) In continuous mode, lines must be multiples of 128 bits. In 2D mode, lines must start on 128-bit boundaries.

For example, when RAW6 packed data are written to memory, any output width is allowed. However, only full 32-bit words are written to memory. This eventually overwrites some data in memory at the end of a line.

The supported output width is restricted for packed RAW10 and RAW12 data because of the particular bit ordering in those formats (see Figure 29).

When the DST_OFST bit is set to 0, start of lines are aligned on 4-byte boundaries. When DST_OFST is not set to 0, data are aligned on 32-byte boundaries.

2.5.3.5 ISS SC DPCM Decompression History

The DPCM compression algorithm can encode the difference between consecutive samples in a line instead of the actual samples value to reduce the amount of data to store. The drawback is that lines must always be decoded from the beginning (the first samples of a line are always encoded as PCM).

The Stall controller has a mechanism to preserve the DPCM decode history for each line to avoid decoding the same samples multiple times when vertical frame division mode is used.

The typical use case (also known as vertical frame division mode) is when an image is wider than the ISP can process on the fly, but the image must be processed. Therefore, the image is cut into multiple vertical slices that are processed sequentially by the ISP. The slices are stitched together in the SDRAM through proper address generation in the ISP. Because of various alignment constraints in the ISP, the slices must overlap. Without preserving the DPCM history, all lines from the beginning of the second (and other consecutive) vertical slice would have to be recoded, which would lead to performance degradation.

Writing DPCM history information into the system memory is enabled by setting the SC_LCM_HISTORY[16] EN_HIST_WR bit. The SC_LCM_HISTORY[15:0] HIST_EXPORT bit field defines the position at which history data is written to memory. The position is counted from the beginning line. The first decoded pixel has position 0. The last decoded pixel has position SC_LCM_HSIZE[14:0] SKIP + SC_LCM_HSIZE [30:16] COUNT. The SC_LCM_HISTORY [15:0] HIST_EXPORT bit field is used to choose the resume position and allows support of overlapping vertical slices. History data is written to the SDRAM at ADDR = SC_LCM_DST_ADDR[31:5] ADDR + Y × 8 bytes (where Y is the line number). The Stall controller always writes 8 bytes of history data per line to the SDRAM regardless of the chosen DPCM format. The Stall controller receiver uses the interface bursts to send history data to memory.

DPCM history data holds the decoded value of four samples. The Stall controller exports samples of positions:

- SC_LCM_HISTORY[15:0] HIST_EXPORT – 3
- SC_LCM_HISTORY[15:0] HIST_EXPORT – 2
- SC_LCM_HISTORY[15:0] HIST_EXPORT – 1
- SC_LCM_HISTORY[15:0] HIST_EXPORT

Every sample is coded on 16 bits, and several MSBs are unused. The valid range for SC_LCM_HISTORY [15:0] HIST_EXPORT is [3.. SC_LCM_HSIZE[14:0] SKIP + SC_LCM_HSIZE[30:16] COUNT – 1]. History data can be exported even when data is not sent to the video port (SC_LCM_HISTORY[15:0] HIST_EXPORT SC_LCM_HSIZE14:0] SKIP is valid).

2.5.3.6 ISS SC Programming Model

This section describes the programming model of the Stall controller

2.5.3.7 ISS SC Reset Behavior

On hardware or software reset of the ISS, all registers in the Stall controller are reset to their reset value.

2.5.3.8 ISS SC Video Port

Table 40 lists the procedure to configure the video port.

Table 40. ISS SC Configure Video Port

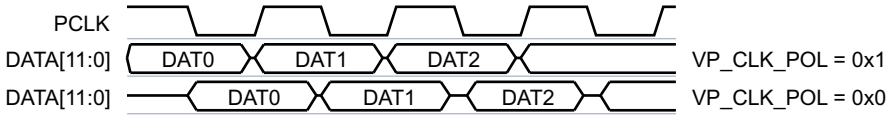
Step	Bit Field	Value
Set the video port output frequency. It varies from ISS_FLCK down to ISS_FLCK/65,536 MHz.	SC_CTRL[31:15] FRACDIV	From 1 to 65,536
Enable video port clock during vertical blanking periods or not.	SC_CTRL[9] VP_CLK_FORCE_ON	0x0: The video port clock is gated during vertical blanking periods. 0x1: The video port clock is free-running during vertical blanking periods.
Controls whether the video-port output is the only output interface enabled and applies for all channels. When SC_CTRL[11] VP_ONLY_EN = 0x1, the data are output only to the video port; the interface master port is not used.	SC_CTRL[11] VP_ONLY_EN	

Table 40. ISS SC Configure Video Port (continued)

Step	Bit Field	Value
Control the video port pixel clock polarity: (Recommended setting: rising edge)	SC_CTRL[12] VP_CLK_POL	0x0: The Stall controller writes the data on the video port on the pixel-clock falling edge. The module connected to the video port samples the data on the pixel clock rising edge. 0x1: The Stall controllerr writes the data on the video port on the pixel-clock rising edge. The module connected to the video port samples the data on the pixel clock falling edge. Figure 31 shows the settings for SC_CTRL.VP_CLK_POL.

Figure 31 shows the settings for SC_CTRL.VP_CLK_POL.

Figure 31. SC_CTRL.VP_CLK_POL Settings



2.5.3.9 ISS SC Memory Read Channel

2.5.3.9.1 ISS SC Read Data From Memory

Table 41 lists the procedure to configure read data from memory.

Table 41. ISS SC Configure Read Data From Memory

Step	Bit Field	Value
	SC_CTRL[0] IF_EN	0x0
	SC_CTRL[3] FRAME	0x1
Configure the burst size to 16 × 64-bit bursts.	SC_LCM_CTRL[7:5] BURST_SIZE	0x4
Configure the source data format, location, and framing. In addition to the SC_LCM_HSIZE[11:0] SKIP and SC_LCM_HSIZE[27:16] COUNT bit fields, firmware must specify the amount of data to be fetched from memory. This value is set in 64-bit word steps and must be a multiple of 32 bytes (four words of 64 bits). The value is computed with the following formula: HWORDS = 4 × ceil(((SKIP + COUNT) x bits_per_pixel)/(8 × 32)) (1) The SC_LCM_SRC_ADDR and SC_LCM_SRC_OFST registers must be aligned on 32-byte boundaries for correct operation. For best performance, both registers must be aligned on 256-byte boundaries. See the example following this table.	SC_LCM_CTRL, SC_LCM_HSIZE, SC_LCM_VSIZE, SC_LCM_PREFETCH, SC_LCM_SRC_ADDR, SC_LCM_SRC_OFST	
Select destination.	SC_LCM_CTRL[2] DST_PORT	0x0: Video port 0x1: Memory
If destination = video port, configure clock frequency and its gating during frame blanking periods.	SC_CTRL[31:15] FRACDIV, SC_CTRL[9] VP_CLK_FORCE_ON	
If needed, configure READ_THROTTLE to reduce the bandwidth in memory-to-memory operation to prevent system overload. It has no effect when data are sent to the video port (controlled by video port clock in this case).	SC_LCM_CTRL[4:3] READ_THROTTLE	

Table 41. ISS SC Configure Read Data From Memory (continued)

Step	Bit Field	Value
If the memory write port is used, the destination format and address must be configured.	SC_LCM_DST_ADDR, SC_LCM_DST_OFST	
Enable memory read channel. After processing a full frame, this bit is automatically cleared by hardware and an EOF event is triggered.	SC_LCM_CTRL[0] CHAN_EN	0x1

Example:

- SC_LCM_CTRL[7:5] BURST_SIZE is set to 16 × 64 bits
- SC_LCM_HSIZE[11:0] SKIP = 0
- SC_LCM_HSIZE[27:16] COUNT = 1000
- SC_LCM_CTRL[23] SRC_PACK = YES
- SC_LCM_CTRL[18:16] SRC_FORMAT = RAW6
- SC_LCM_PREFETCH[13:3] HWORDS = 96 (=94)
  Setting the size to 94 produces the following burst sequence: 16, 16, 16, 16, 16, 8, and 4 (7 interconnect requests). However, when it is set to 96, the burst sequence is 6 × 16 (6 interconnect requests).

2.5.4 ISS SC Registers

Table 42 lists the SC instance.

Table 42. ISS SC Instance Summary

Module Name	Base Address Cortex-M3 Private Access	Base Address L3 Interconnect	Size
ISS_SC	0x5504 1C00	0x5C00 1C00	512 bytes

Table 43 lists the ISS SC registers.

Table 43. ISS SC Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ISS_SC Base Address Cortex-M3 Private Access	ISS_SC Base Address L3 Interconnect
SC_REVISION	R	32	0x0000 0000	0x5504 1C00	0x5C00 1C00
SC_SYSCONFIG	RW	32	0x0000 0004	0x5504 1C04	0x5C00 1C04
SC_SYSSTATUS	R	32	0x0000 0008	0x5504 1C08	0x5C00 1C08
Reserved			0x0000 000C - 0x0000 0018	0x5504 1C0C - 0x5504 1C18	0x5C00 1C0C - 0x5C00 1C18
SC_LCM_IRQENABLE	RW	32	0x0000 002C	0x5504 1C2C	0x5C00 1C2C
SC_LCM_IRQSTATUS	RW	32	0x0000 0030	0x5504 1C30	0x5C00 1C30
SC_CTRL	RW	32	0x0000 0040	0x5504 1C40	0x5C00 1C40
Reserved	W	32	0x0000 0044	0x5504 1C44	0x5C00 1C44
SC_GNQ	R	32	0x0000 0048	0x5504 1C48	0x5C00 1C48
SC_CTRL1	RW	32	0x0000 004C	0x5504 1C4C	0x5C00 1C4C
Reserved	RW	32	0x0000 0050 - 0x0000 0078	0x5504 1C50 - 0x5504 1C78 + (x * 0x30)	0x5C00 1C50 - 0x5C00 1C78 + (x * 0x30)
SC_LCM_CTRL	RW	32	0x0000 01D0	0x5504 1DD0	0x5C00 1DD0
SC_LCM_VSIZE	RW	32	0x0000 01D4	0x5504 1DD4	0x5C00 1DD4
SC_LCM_HSIZE	RW	32	0x0000 01D8	0x5504 1DD8	0x5C00 1DD8
SC_LCM_PREFETCH	RW	32	0x0000 01DC	0x5504 1DDC	0x5C00 1DDC
SC_LCM_SRC_ADDR	RW	32	0x0000 01E0	0x5504 1DE0	0x5C00 1DE0
SC_LCM_SRC_OFST	RW	32	0x0000 01E4	0x5504 1DE4	0x5C00 1DE4
SC_LCM_DST_ADDR	RW	32	0x0000 01E8	0x5504 1DE8	0x5C00 1DE8
SC_LCM_DST_OFST	RW	32	0x0000 01EC	0x5504 1DEC	0x5C00 1DEC
SC_LCM_HISTORY	RW	32	0x0000 01F0	0x5504 1DF0	0x5C00 1DF0

2.5.4.1 SC_Revision

Figure 32. SC_REVISION Register[(1)]

31	0
REVISION	
R-0	

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

(1) TI internal data

Table 44. SC_REVISION Register Field Descriptions

Bit	Field	Value	Description
31-0	REVISION		IP Revision

2.5.4.2 SC SYSCONFIG

Table 45. SC_SYSCONFIG

Address Offset	0x0000 0004		
Physical Address	0x5504 1C04 0x5C00 1C04	**Instance**	ISS_SC_CORTEX-M3 ISS_SC_L3
Description	SYSCONFIG REGISTER		
Type	RW		

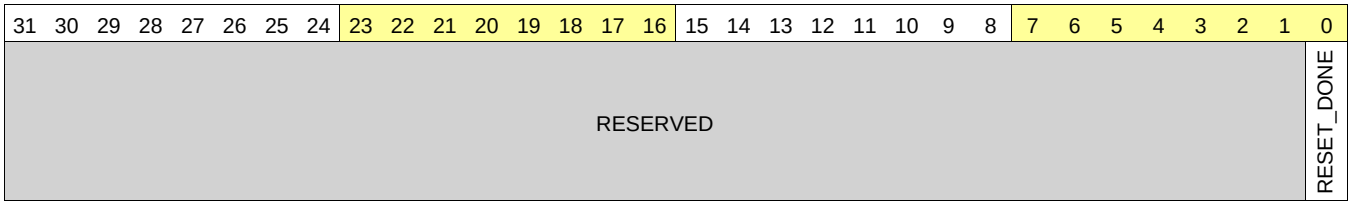
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		MSTANDBY_MODE		RESERVED										SOFT_RESET	AUTO_IDLE

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:12	MSTANDBY_MODE	Sets the behavior of the master port power management signals. 0x0: Force-standby. MStandby is only asserted when the module is disabled. 0x1: No-standby. MStandby is never asserted. 0x2: Smart-standby: MStandby is asserted based on the activity of the module. The module will try to go to standby during the vertical blanking period.	RW	0x0
11:2	RESERVED		R	0x000
1	SOFT_RESET	Software reset. Set the bit to 1 to trigger a module reset. The bit is automatically reset by the hardware. During reads return 0. 0x0: Normal mode. 0x1: The module is reset	RW	0
0	AUTO_IDLE	Internal OCP clock gating strategy. 0x0: OCP clock is free-running. 0x1: Automatic OCP clock gating strategy is applied based on the OCP interface activity.	RW	1

2.5.4.3 SC SYSSTATUS

Table 46. SC_SYSSTATUS

Address Offset	0x0000 0008		
Physical Address	0x5504 1C08 0x5C00 1C08	**Instance**	ISS_SC_CORTEX-M3 ISS_SC_L3
Description	SYSSTATUS REGISTER		
Type	R		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0's for future compatibility. Reads returns 0.	R	0x0000 0000
0	RESET_DONE	Internal reset monitoring Read 0x1: Reset completed. Read 0x0: Internal module reset is on going.	R	1

2.5.4.4 SC LCM IRQENABLE

Table 47. SC_LCM_IRQENABLE

Address Offset	0x0000 002C		
Physical Address	0x5504 1C2C 0x5C00 1C2C	**Instance**	ISS_SC_CORTEX-M3 ISS_SC_L3
Description	INTERRUPT ENABLE REGISTER - Memory channel This register regroups all the events related to the memory channel 2. The events related to memory channel trigger SINTERRUPTN[8]. The channel shall be enabled for events to be generated on that channel.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LCM_OCPERROR	LCM_EOF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LCM_OCPERROR	An interconnect error has been returned for a read (interconnect read master) or write (interconnect write master) transaction related to LCM operation 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
0	LCM_EOF	Memory read channel - End of frame 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

2.5.4.5 SC LCM IRQSTATUS

Table 48. SC_LCM_IRQSTATUS

Address Offset	0x0000 0030		
Physical Address	0x5504 1C30 0x5C00 1C30	**Instance**	ISS_SC_CORTEX-M3 ISS_SC_L3
Description	INTERRUPT STATUS REGISTER - Memory channel This register regroups all the events related to memory channel. The events related to memory channel trigger SINTERRUPTN[8]. The channel shall be enabled for events to be generated on that channel.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LCM_OCPERROR	LCM_EOF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LCM_OCPERROR	An interconnect error has been returned for a read (interconnect read master) or write (interconnect write master) transaction related to LCM operation 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
0	LCM_EOF	Memory read channel - End of frame 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0

2.5.4.6 SC CTRL

Table 49. SC_CTRL

Address Offset	0x0000 0040		
Physical Address	0x5504 1C40 0x5C00 1C40	**Instance**	ISS_SC_CORTEX-M3 ISS_SC_L3
Description	GLOBAL CONTROL REGISTER This register controls the Stall controller. This register shall not be modified dynamically (except IF_EN bit field).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRACDIV																POSTED	DBG_EN	VP_CLK_POL	VP_ONLY_EN	RESERVED	VP_CLK_FORCE_ON	RESERVED	BURST			MODE	FRAME	RESERVED	RESERVED	Reserved	

Bits	Field Name	Description	Type	Reset
31:15	FRACDIV	Fractional clock divider control for the video port. The mean video port clock is VPBASECLOCK * FRACDIV/65536. Valid range: 1-65536	RW	0x10000
14	POSTED	Selects between posted and non posted writes. 0x0: Non posted 0x1: Posted	RW	0
13	DBG_EN	Enables the debug mode. 0x0: Disable 0x1: Enable	RW	0
12	VP_CLK_POL	VP clock polarity 0x0: The Stall controller writes the data on the VP on the L3 falling edge before the next falling PCLK edge. 0x1: The Stall controller writes the data on the VP on the L3 rising edge before the next rising PCLK edge.	RW	0
11	VP_ONLY_EN	VP only enable. 0x0: The VP is enabled and the OCP master port are enabled. 0x1: The VP is enabled and the OCP master port is disabled. The embedded data and pixel data are output on the VP.	RW	0
10	RESERVED	Read returns reset value	RW	0
9	VP_CLK_FORCE_ON	Controls VP_PCLK gating during frame blanking periods. 0x0: The VP_PCLK is gated during vertical blanking periods. 0x1: The VP_PCLK is free-running during vertical blanking periods.	RW	0
8	RESERVED		R	0
7:5	BURST	Forces the write burst size used by the module. The write burst size shall never exceed the output FIFO size. The output FIFO size can be read with the SC_GNQ.FIFODEPTH bit field. 0x0: 1x 64-bit burst = single request. 0x1: 2x 64-bit bursts. 0x3: 8x 64-bit bursts. 0x4: 16x 64-bit bursts. 0x2: 4x 64-bit bursts.	RW	0x0
4	MODE	0x1: SC compatible mode	RW	0
3	FRAME	Set the modality in which IF_EN works. 0x0: When software writes IF_EN = 0 the interface is disabled immediately. 0x1: When software writes IF_EN = 0 the interface is disabled after the next FEC sync code.	RW	0
2	RESERVED	Read returns reset value	RW	0
1	RESERVED	Read returns reset value	RW	0
0	Reserved		R	0

2.5.4.7 SC GNQ

Table 50. SC_GNQ

Address Offset	0x0000 0048		
Physical Address	0x5504 1C48 0x5C00 1C48	**Instance**	ISS_SC_CORTEX-M3 ISS_SC_L3
Description	GENERIC PARAMETER REGISTER This register provide a way to read the generic parameters used in the design.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										OCPREADPORT	FIFODEPTH			NBCHANNELS	

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0000000
5	OCPREADPORT	The OCP master read port, the DPCM encoder and ALAW decompression are only present when this bit is set.	R	1
4:2	FIFODEPTH	Output FIFO size in multiple of 64 bits. Read 0x0: 2 x 64 bits Read 0x1: 4 x 64 bits Read 0x2: 8 x 64 bits Read 0x3: 16 x 64 bits Read 0x4: 32 x 64 bits Read 0x5: 64 x 64 bits	R	0x5
1:0	NBCHANNELS	Number of logical channels supported by the module. Read 0x0: 1 logical channel Read 0x1: 2 logical channels Read 0x2: 4 logical channels Read 0x3: 8 logical channels	R	0x2

2.5.4.8 SC CTRL1

Table 51. SC_CTRL1

Address Offset	0x0000 004C		
Physical Address	0x5504 1C4C 0x5C00 1C4C	**Instance**	ISS_SC_CORTEX-M3 ISS_SC_L3
Description	GLOBAL CONTROL REGISTER (2) This register controls the Stall controller.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				LEVH				RESERVED				LEVL				RESERVED															BLANKING

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	RO Rreturns 0s	0
30:24	LEVH	Controls generation of MFlag[1:0]: 00: FIFO_LEV=LEVL 01: Unused 10: LEVLFIFO_LEV and FIFO_LEV=LEVH 11: LEVHFIFO_LEV Allowed values 0..FIFO_SIZE.	RW	0x00
23	RESERVED	Reserved	RO Rreturns 0s	0
22:16	LEVL	Controls generation of MFlag[1:0]: 01: Unused 10: LEVLFIFO_LEV and FIFO_LEV=LEVH 11: LEVHFIFO_LEV Allowed values 0..FIFO_SIZE.	RW	00:FIFO_LEV=LEVL
15:2	RESERVED	Reserved	RO Rreturns 0s	0x0000
1:0	BLANKING	Controls the number of clock pulses provided during vertical and horizontal clock periods. When the blanking period provided by the camera is lower than the value set here, the blanking period is shortened by the Stall controller to prevent internal FIFO overflow. Software must increase the sensor blanking period in that case. 0x0: 4 video port clock cycles 0x1: 16 video port clock cycles 0x2: 64 video port clock cycles 0x3: Free running	RW	0x0

2.5.4.9 SC LCM_CTRL

Table 52. SC_LCM_CTRL

Address Offset	0x0000 01D0		
Physical Address	0x5504 1DD0 0x5C00 1DD0	**Instance**	ISS_SC_CORTEX-M3 ISS_SC_L3
Description	Control register for the memory channel. It defines the data format of the source frame stored in memory and how this frame is processed.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DST_PACK	DST_DPCM_PRED	DST_COMPR		RESERVED	DST_FORMAT			SRC_PACK	SRC_DPCM_PRED	SRC_DECOMPR		SRC_FORMAT				RESERVED								BURST_SIZE			READ_THROTTLE			DST_PORT	RESERVED	CHAN_EN

Bits	Field Name	Description	Type	Reset
31	DST_PACK	Data is packed before it is sent to memory. Applies to RAW6, RAW7, RAW10, and RAW12 only. 0x0: Disabled 0x1: Enabled	RW	0
30	DST_DPCM_PRED	Selects the DPCM predictor to be used for the RAW6+DPCM10 and RAW7+DPCM10 data formats. The RAW8+DPCM10 data format always use the simple predictor. 0x0: The advanced predictor is used 0x1: The simple predictor is used.	RW	0
29:28	DST_COMPR	Enables data compression of data sent to memory 0x0: No compression 0x1: A-Law compression RAW10 - RAW8 A-Law when DST_FORMAT=RAW8 other destination formats are invalid. 0x2: DPCM compression RAW10 - RAW6 DPCM when DST_FORMAT=RAW6 RAW10 - RAW7 DPCM when DST_FORMAT=RAW7 RAW10 - RAW8 DPCM when DST_FORMAT=RAW8 other destination formats are invalid.	RW	0x0
27	RESERVED		R	0
26:24	DST_FORMAT	Output format selection. Not every combination between input and output formats are possible. 0x0: RAW6 0x1: RAW7 0x2: RAW8 0x3: RAW10 0x4: RAW12 0x5: RAW14 0x6: RAW16	RW	0x0
23	SRC_PACK	Data stored in memory is packed and must be unpacked. 0x0: Disabled 0x1: Enabled	RW	0

Bits	Field Name	Description	Type	Reset
22	SRC_DPCM_PRED	Selects the DPCM predictor to be used for the RAW6+DPCM10, RAW7+DPCM10 and RAW8+DPCM12 data formats. The RAW8+DPCM10 and RAW6 + DPCM12 data format always use the simple predictor. 0x0: The advanced predictor is used 0x1: The simple predictor is used.	RW	0
21:20	SRC_DECOMPR	Enable decompression of incoming data 0x0: No decompression 0x1: A-Law decompression RAW8 A-Law - RAW10 when SRC_FORMAT=RAW8 other source formats are invalid. 0x3: DPCM decompression RAW6 DPCM - RAW12 when SRC_FORMAT=RAW6 RAW8 DPCM - RAW12 when SRC_FORMAT=RAW8 other source formats are invalid. 0x2: DPCM decompression RAW6 DPCM - RAW10 when SRC_FORMAT=RAW6 RAW7 DPCM - RAW10 when SRC_FORMAT=RAW7 RAW8 DPCM - RAW10 when SRC_FORMAT=RAW8 other source formats are invalid.	RW	0x0
19:16	SRC_FORMAT	Data format of the data stored in memory. As there is no header embedded in the data sent to memory the user is responsible of choosing the adequate format. 0x0: RAW6. 0x1: RAW7 0x2: RAW8 0x3: RAW10 0x4: RAW12 0x5: RAW14 0x6: RAW16 0x7: Reserved 0x8: Reserved 0x9: Reserved 0xA: Reserved 0xB: Reserved 0xC: Reserved 0xD: Reserved 0xE: Reserved 0xF: Reserved	RW	0x0
15:8	RESERVED		R	0x00
7:5	BURST_SIZE	Defines the burst size of the master read port 0x0: 1x 64-bit burst = single request. 0x1: 2x 64-bit bursts. 0x3: 8x 64-bit bursts. 0x4: 16x 64-bit bursts. 0x2: 4x 64-bit bursts.	RW	0x0
4:3	READ_THROTTLE	Limit maximum data read speed for memory to memory operation 0x0: Full speed. Throughput is limited by internal processing capabilities. 0x1: 1/2 speed 0x3: 1/8 speed 0x2: 1/4 speed	RW	0x0

Bits	Field Name	Description	Type	Reset
2	DST_PORT	Select the destination port 0x0: Data is send to video port, it is always send without compression or packing. The DST_COMPR, DST_DPCM_PRED, DST_PACK, SC_LCM_DST_WRITE, and SC_LCM_DST_OFST registers have no effect. 0x1: Data is send to memory.	RW	0
1	RESERVED		R	0
0	CHAN_EN	Enables the read from memory channel. Before enabling the memory read channel software shall: Read from memory starts as soon as this bit is set, therefore all SC_LCM_x registers must be configured correctly before. This bit is cleared by hardware at the end of the frame. 0x0: Disabled 0x1: Enabled	RW	0

2.5.4.10 SC LCM VSIZE

Table 53. SC_LCM_VSIZE

Address Offset	0x0000 01D4		
Physical Address	0x5504 1DD4 0x5C00 1DD4	**Instance**	ISS_SC_CORTEX-M3 ISS_SC_L3
Description	Memory channel vertical framing register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COUNT								RESERVED															

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	COUNT	Defines the line count to be read from memory. From 1 to 8191 lines.	RW	0x0001
15:0	RESERVED		R	0x0000

2.5.4.11 SC LCM HSIZE

Table 54. SC_LCM_HSIZE

Address Offset	0x0000 01D8		
Physical Address	0x5504 1DD8 0x5C00 1DD8	**Instance**	ISS_SC_CORTEX-M3 ISS_SC_L3
Description	Memory read channel horizontal framing register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COUNT								RESERVED								SKIP							

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:16	COUNT	Horizontal count of samples to output after the skipped pixels. Valid values: 1 to 32767.	RW	0x0001
15	RESERVED		R	0x0
14:0	SKIP	Horizontal count of samples to skip after the start of the line. When DPCM compressed data is read from memory using this feature is the only valid way to set a horizontal starting position. Valid values: 0 to 32767. 0 disables pixel skipping	RW	0x0000

2.5.4.12 SC LCM_PREFETCH

Table 55. SC_LCM_PREFETCH

Address Offset	0x0000 01DC		
Physical Address	0x5504 1DDC 0x5C00 1DDC	**Instance**	ISS_SC_CORTEX-M3 ISS_SC_L3
Description	This register defines the amount of data to be fetched from memory. It must be consistent with the SC_LCM_HSIZE register (check programming model).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															HWORDS														RESERVED	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:3	HWORDS	64 bit words to read from memory for each line of the image. Possible values 1..8191	RW	0x0001
2:0	RESERVED		R	0x0

2.5.4.13 SC_LCM_SRC_ADDR

Table 56. SC_LCM_SRC_ADDR

Address Offset	0x0000 01E0		
Physical Address	0x5504 1DE0 0x5C00 1DE0	**Instance**	ISS_SC_CORTEX-M3 ISS_SC_L3
Description	Memory channel source address register This register sets the 32-bit memory address where the pixel data are stored. The 5 LSBs are ignored: the address shall be aligned on a 32-byte boundary.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																											RESERVED			

Bits	Field Name	Description	Type	Reset
31:5	ADDR	27 most-significant bits of the 32-bit address	RW	0x0000000
4:0	RESERVED		R	0x00

2.5.4.14 SC_LCM_SRC_OFST

Table 57. SC_LCM_SRC_OFST

Address Offset	0x0000 01E4		
Physical Address	0x5504 1DE4 0x5C00 1DE4	**Instance**	ISS_SC_CORTEX-M3 ISS_SC_L3
Description	Memory channel source offset register. This register sets the offset, which is applied on the source address after each line is read from memory. For example, it enables to perform 2D data transfers of the pixel data from a frame buffer. In such case, the pixel data and frame buffer data shall have the same data format. The 5 LSBs are ignored: the offset shall be a multiple of 32 bytes.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFST																											RESERVED			

Bits	Field Name	Description	Type	Reset
31:5	OFST	Line offset programmed in bytes. If OFST = 0, the data is read contiguously from memory. Otherwise, OFST sets the source offset between the first pixel of the previous line and the first pixel of the current line.	RW	0x0000000
4:0	RESERVED		R	0x00

2.5.4.15 SC_LCM_DST_ADDR

Table 58. SC_LCM_DST_ADDR

Address Offset	0x0000 01E8		
Physical Address	0x5504 1DE8 0x5C00 1DE8	**Instance**	ISS_SC_CORTEX-M3 ISS_SC_L3
Description	Memory channel destination address. This register sets the 32-bit memory address where the pixel data are stored. The 5 LSBs are ignored: the address shall be aligned on a 32-byte boundary.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																											RESERVED			

Bits	Field Name	Description	Type	Reset
31:5	ADDR	27 most significant bits of the 32-bit address.	RW	0x0000000
4:0	RESERVED		R	0x00

2.5.4.16 SC_LCM_DST_OFST

Table 59. SC_LCM_DST_OFST

Address Offset	0x0000 01EC		
Physical Address	0x5504 1DEC 0x5C00 1DEC	**Instance**	ISS_SC_CORTEX-M3 ISS_SC_L3
Description	Memory channel destination offset register. This register sets the offset, which is applied on the destination address after each line is written to memory. For example, it enables to perform 2D data transfers of the pixel data into a frame buffer. In such case, the pixel data and frame buffer data shall have the same data format. The 5 LSBs are ignored: the offset shall be a multiple of 32 bytes.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFST																								RESERVED							

Bits	Field Name	Description	Type	Reset
31:5	OFST	Line offset programmed in bytes. If OFST = 0, the data is written contiguously to memory if possible. At the end of a line only full 32 bit words will be written, creating eventually gaps at the end of lines. Otherwise, OFST sets the destination offset between the first pixel of the previous line and the first pixel of the current line.	RW	0x0000000
4:0	RESERVED		R	0x00

2.5.4.17 SC_LCM_HISTORY

Table 60. SC_LCM_HISTORY

Address Offset	0x0000 01F0		
Physical Address	0x5504 1DF0 0x5C00 1DF0	**Instance**	ISS_SC_CORTEX-M3 ISS_SC_L3
Description	Controls operation of the DPCM history read/write feature		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														EN_HIST_RD	EN_HIST_WR	HIST_EXPORT															

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17	EN_HIST_RD	Enable DPCM history read 0x0: Disable 0x1: Enable	RW	0
16	EN_HIST_WR	Enable DPCM history write 0x0: Disable 0x1: Enable	RW	0
15:0	HIST_EXPORT	Defines the horizontal position at which DPCM history information is written. The first decoded sample of a line has position 0 The last decoded sample has position SKIP+COUNT–1 Valid range [3..SKIP+COUNT–1]	RW	0x0000

2.6 ISS CSI2

2.6.1 ISS CSI2 Environment

2.6.1.1 ISS CSI2 Protocol and Data Format

The CSI2 supports MIPI CSI2 multiple data type formats. This section describes MIPI CSI2 protocol and data formats. The CSI2 is compatible with the *MIPI CSI2 Specification v1.0-01-00 r0.03*. Table 61 lists the MIPI CSI2 supported by CSI2 formats in addition to JPEG8. Shading in the MIPI CSI2-defined formats indicates special format extensions of the CSI2 receiver.

Table 61 summarizes the pixel formats supported by the CSI2 receiver interface.

Table 61. ISS CSI2 Pixel Format Modes

CSI2_CTX_CTRL2_i[9:0] Format	CSI2 Data Format	Bits per Pixel (BPP)	Data Size Increases in Memory	2D Mode Availability	Comments
0x18	YUV4:2:0 8 bit	12	0%	Yes	
0x19	YUV4:2:0 10 bit	12	0%	Yes	
0x1E	YUV4:2:2 8 bit	16	0%	Yes	
0x1F	YUV4:2:2 10 bit	16	0%	Yes	
0x22	RGB565	16	0%	Yes	
0x24	RGB888	24	0%	Yes	
0x29	RAW7	7	0%	Yes	
0x2A	RAW8	8	8%	Yes	
0x2B	RAW10	10	0%	Yes	
0x2C	RAW12	12	0%	Yes	
0x2D	RAW14	14	0%	Yes	
0xA3	RGB666 + EXP32	32	77%	Yes	
0x68	RAW6 + EXP8	8	33%	Yes	
0x69	RAW7 + EXP8	8	14%	Yes	
0xA0	RGB444 + EXP16	16	33%	Yes	
0xA1	RGB555 + EXP16	16	6%	Yes	
0xAB	RAW10 + EXP16	16	60%	Yes	
0xAC	RAW12 + EXP16	16	33%	Yes	
0xAD	RAW14 + EXP16	16	14%	Yes	
0xE3	RGB666 + EXP32	32	77%	Yes	
0xE4	RGB888 + EXP32	32	33%	Yes	
0x2A8	RAW6 + DPCM10 + EXP16	16	166%	Yes	DPCM decompression
0x229	RAW7 + DPCM10 + EXP16	16	128%	Yes	DPCM decompression
0x2AA	RAW8 + DPCM10 + EXP16	16	100%	Yes	DPCM decompression
0x369	RAW7 + DPCM12 + EXP16	16	128%	Yes	DPCM decompression
0x36A	RAW8 + DPCM12 + EXP16	16	100%	Yes	DPCM decompression
0x3A8	RAW6 + DPCM12 + EXP16	16	166%	Yes	DPCM decompression
0x12	JPEG8	8	0%		

For more information about how the data formats are transmitted and how the data are stored in memory, see Section 2.6.1.1.4, *CSI2 Operating Modes*.

NOTE: The VP formats are not included in Table 61, because they are not sent to memory; instead they are sent to the ISP for further processing.

NOTE: Data written by CSI2 can be read back by the SC read channel. For more information about supported memory operations, see Table 36.

2.6.1.1.1 ISS CSI2 Physical Layer

The CSI2-A receiver is tightly connected to a PHY layer (for more information about the PHY, see Section 2.3, *ISS CSI2 PHY*). Table 62 lists the CSI2-A receiver I/O. The CSI2_RECEIVER provides access to the complex I/O register through a serial configuration port interface. The CSI2_RECEIVER can access up to eight 32-bit registers in complex I/O.

Table 62. ISS CSI2-A I/O Description

Signal Name		I/O[1]	Description
csi2a_dx0	lane 0 (position 1)	I	Serial data/clock input
csi2a_dy0			
csi2a_dx1	lane 1 (position 2)	I	Serial data/clock input
csi2a_dy1			
csi2a_dx2	lane 2 (position 3)	I	Serial data/clock input
csi2a_dy2			
csi2a_dx3	lane 3 (position 4)	I	Serial data/clock input
csi2a_dy3			
csi2a_dx4	lane 4 (position 5)	I	Serial data input only
csi2a_dy4			

(1) I = Input

NOTE: The serial lane can be used as clock lane or data lane (excluding lane 4 on the CSI2-A I/O). The MIPI CSI2 protocol requires one clock lane (others are data lane or unused lane).

Lanes support the two operating modes:

- HS mode: High-speed transmit mode
- Off mode: Lane is off.

2.6.1.1.2 ISS CSI2 Lane Merger

The layer consists of lane merger logic to merge the incoming serial stream into a byte stream. The lane merger can merge up to four lanes (CSI2-A) into a single byte stream. The bits are sent with the LSB first. The order of the lanes at the CSI2-A receiver core depends on the lane configuration. The merger is not used for a single lane.

The number of lanes and their configuration can be changed only in ULPM or when all data lanes are in off mode.

Figure 33 to Figure 36 show the byte position into each serial link for one to four data lane configurations. The byte stream always starts from lane 1. It finishes on one of the lanes, depending on the number of bytes to receive and the number of lanes.

Figure 33. ISS CSI2 One Data-Lane Configuration

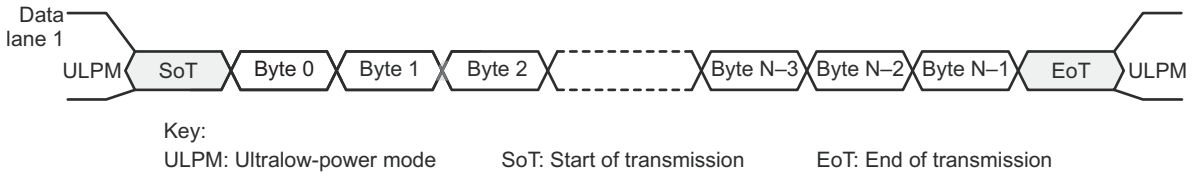


Figure 34. ISS CSI2 Two Data-Lane Merger Configuration

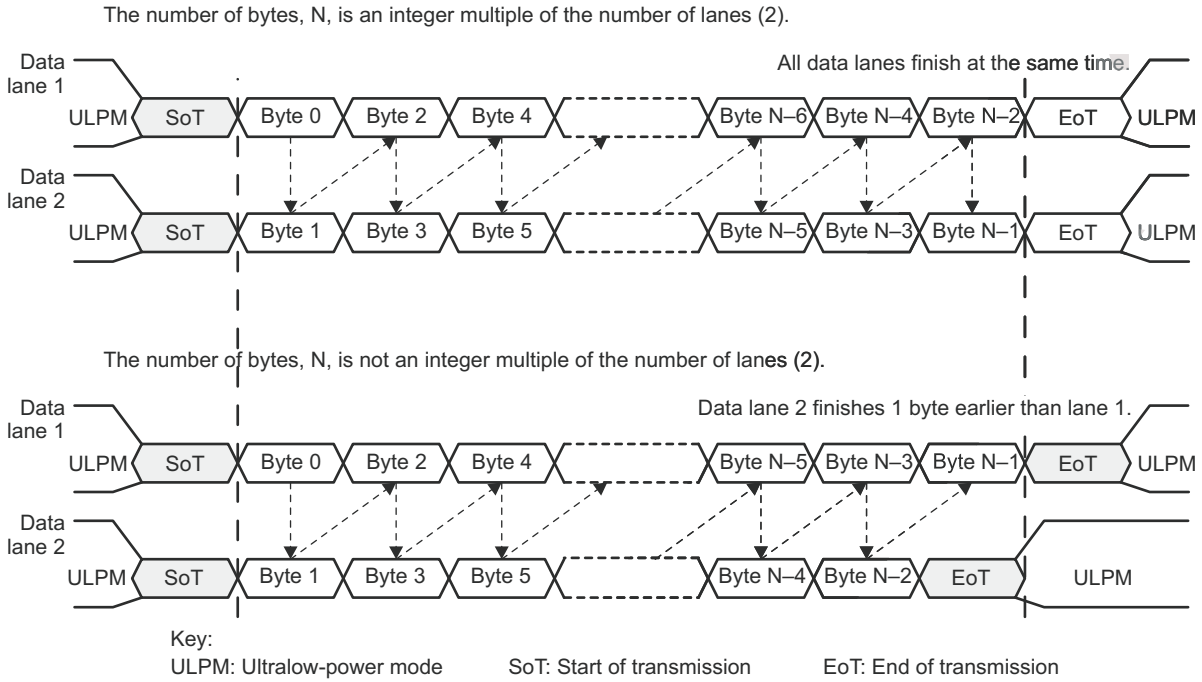


Figure 35. ISS CSI2 Three Data-Lane Merger Configuration

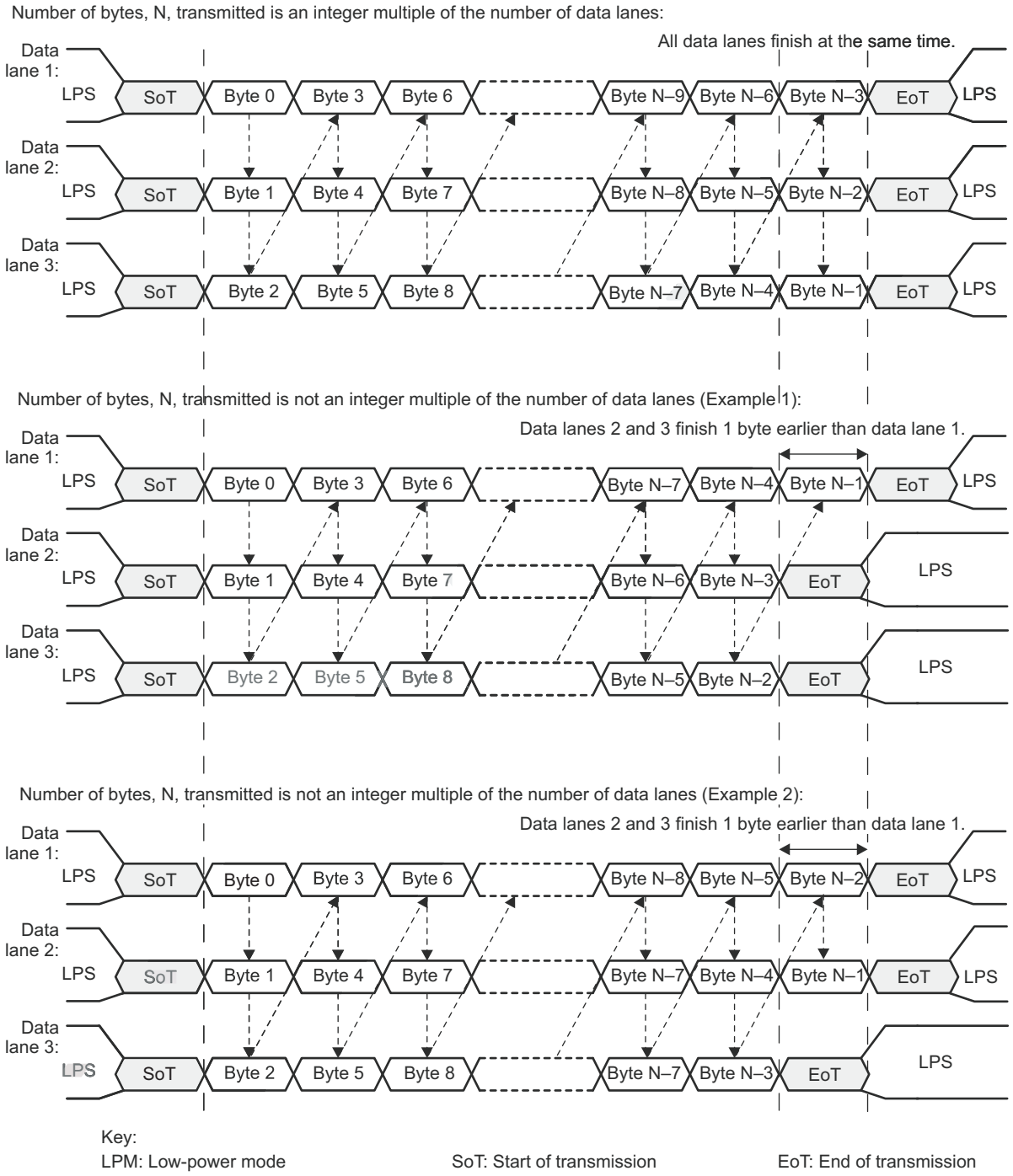
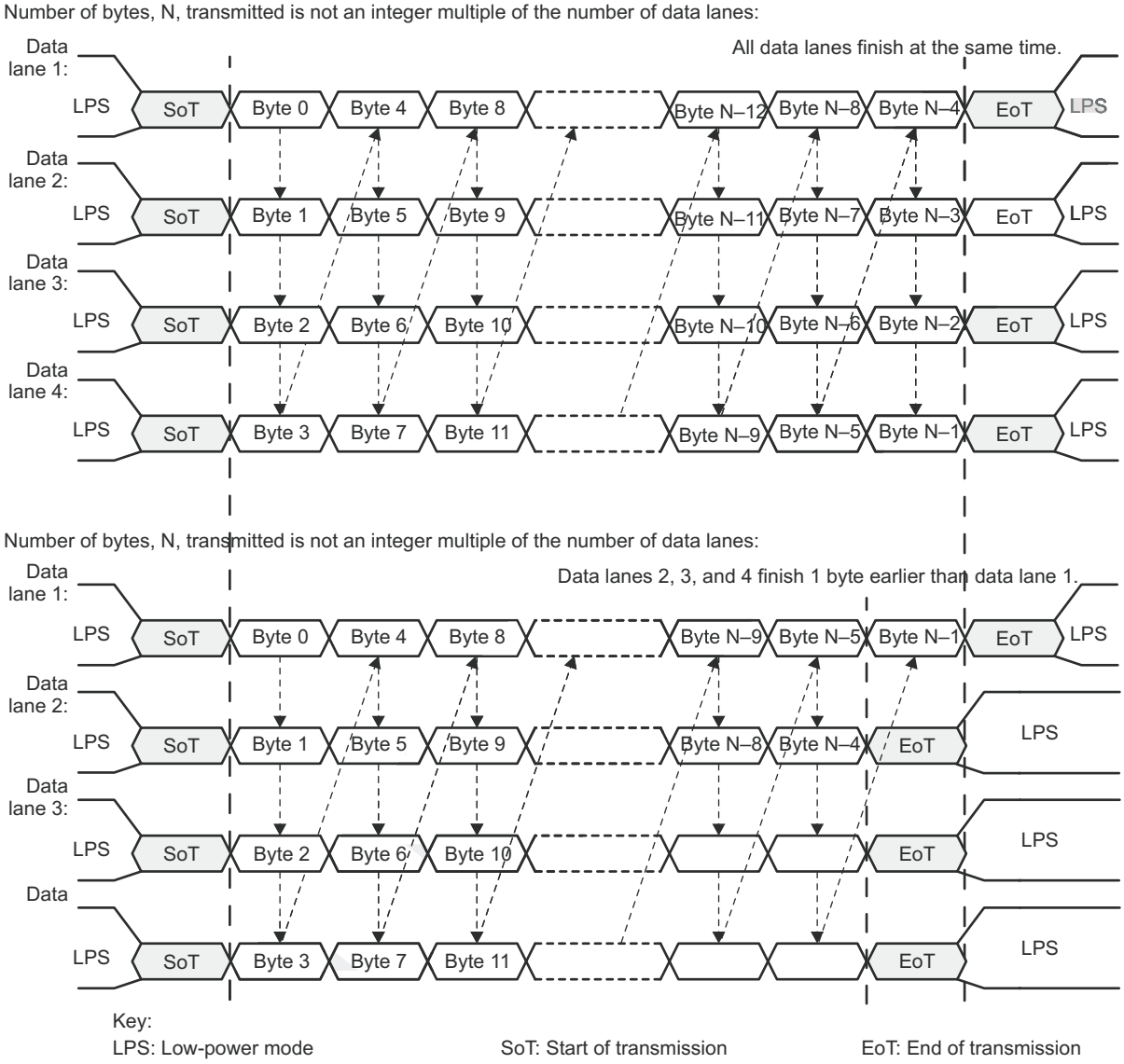


Figure 36. ISS CSI2 Four Data-Lane Merger Configuration



2.6.1.1.3 ISS CSI2 Protocol Layer

The low-level protocol (LLP) is a byte-oriented protocol from the lane merger layer. It supports short and long packet formats.

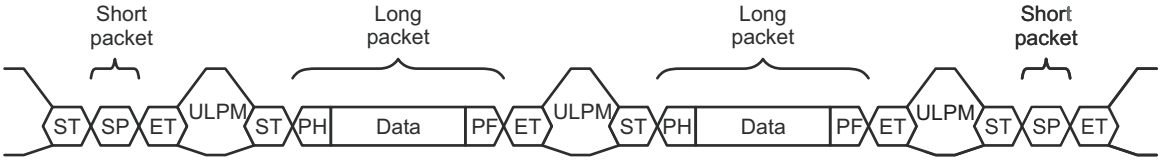
The CSI2 protocol layer defines how image-sensor data is transported onto the physical layer.

The feature set of the protocol layer implemented by the CSI2 receiver is:

- Transport of arbitrary data (payload-independent)
- 8-bit word size
- Support for up to four interleaved virtual channels on the same link
- Special packets for frame-start, frame-end, line-start, and line-end information
- Descriptor for the type, pixel depth, and format of application-specific payload data
- Error-correction code (ECC) for 1-bit error correction or 2-bit error detection in the header
- 16-bit checksum code for payload error detection

Figure 37 shows the CSI2 protocol layer with short and long packets.

Figure 37. ISS CSI2 Protocol Layer With Short and Long Packets



Key:

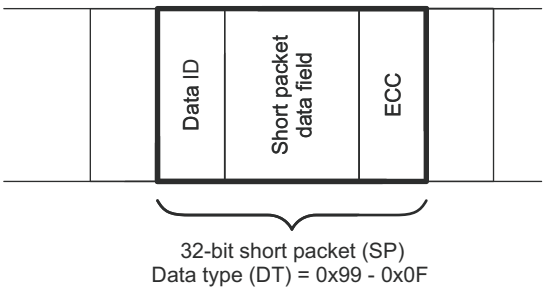
ST: Start of transmission	ET: End of transmission
PH: Packet header	PF: Packet footer
ULPM: Ultralow-power mode	SP: Short packet

Two packets are always separated from each other with a sequence of ET, ULPM, and ST.

2.6.1.1.3.1 ISS CSI2 Short Packet

A short packet is identified by data types 0x00 to 0x0F. A short packet can be used for frame or line synchronization or for generic data. Figure 38 shows the structure of a short packet.

Figure 38. ISS CSI2 Short Packet Structure



For frame-synchronization data types, the short packet data field is the frame number. For line-synchronization data types, the short packet data field is the line number. For generic short packet data types, the content of the short packet data field is user-defined.

The 16-bit frame number, when used, is always nonzero to distinguish it from the use case where the frame number is inoperative and remains set to 0. The behavior of the 16-bit frame number is one of the following:

- The frame number is always 0 and is inoperative.
- The frame number increments by 1 for every FS packet within the same virtual channel and is periodically reset to 1 (1, 2, 1, 2, 1, 2, 1, 2 or 1, 2, 3, 4, 1, 2, 3, 4).

For LSC and LEC synchronization packets, the short packet data field contains a 16-bit line number. This line number is the same for the LS and LE packets corresponding to a given line. Line numbers are logical line numbers and do not necessarily equal physical line numbers. The 16-bit line number, when used, is always nonzero to distinguish it from the use case where the line number is inoperative and remains set to 0.

The behavior of the 16-bit line number is one of the following:

- The line number is always 0 and is inoperative.
- The line number increments by 1 for every LS packet within the same virtual channel and the same data type. The line number is periodically reset to 1 for the first LS packet after an FS packet. The intended use is for progressive scan (noninterlaced) video data streams. The line number must be a nonzero value.
- The line number increments by the same arbitrary step value greater than 1 for every LS packet within the same virtual channel and the same data type. The line number is periodically reset to a nonzero arbitrary start value for the first LS packet after an FS packet. The arbitrary start value can be different between successive frames. The intended use is for interlaced video data streams.

The ECC byte allows single-bit errors to be corrected and 2-bit errors to be detected in the short packet.

Short packets apply to all contexts using the same virtual channel ID (up to eight contexts support eight dedicated configurations of virtual channel ID and data types). The data type associated with the context is not used to distinguish which context is used when receiving short packets.

2.6.1.1.3.2 ISS CSI2 Long Packet

A long packet is identified by data types 0x10 to 0x37. A long packet consists of three elements:

- A 32-bit packet header (PH)
- An application-specific data payload with a variable number of 8-bit data words
- A 16-bit packet footer (PF)

The packet header is composed of three elements:

- An 8-bit data identifier
- A 16-bit word count field
- An 8-bit ECC

The packet footer has one element, a 16-bit checksum.

Figure 39 and Table 63 show the structure of a long packet.

Figure 39. ISS CSI2 Long Packet Structure

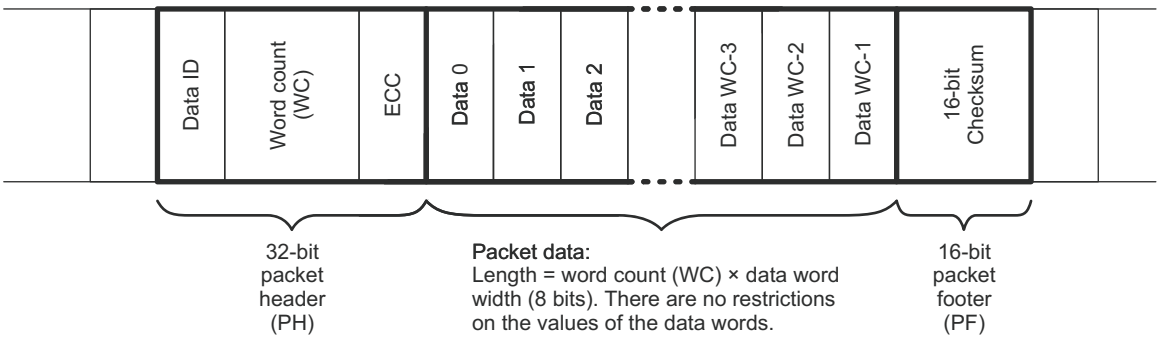


Table 63. ISS CSI2 Long Packet Structure Description

Packet Part	Field Name	Size (Bits)	Description
Header	Data ID	8	Contains the virtual channel identifier and the data-type information
	Word count	16	Number of data words in the packet data. A word is 8 bits.
	ECC	8	ECC for data ID and WC field. Allows 1-bit error recovery and 2-bit error detection.
Data	Data	WC-8	Application-specific payload (WC words of 8 bits)
Footer	Checksum	16	16-bit CRC for packet data

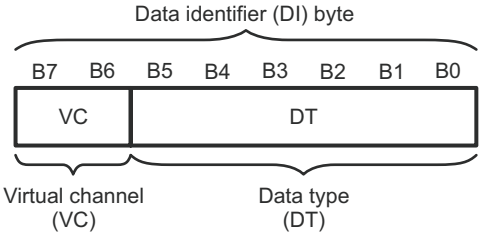
There are no restrictions on the size of the packet data, but each data format can impose additional restrictions on the length of the payload data (for example, a multiple of 4 bytes).

2.6.1.1.3.3 ISS CSI2 Data Identifier

The data identifier byte contains the virtual channel (VC) value and the data-type (DT) value, as shown in Figure 40. The VC value is in the 2 MSBs of the data identifier byte. The DT value is in the 6 LSBs of the data identifier byte.

Figure 40 shows the data identifier structure.

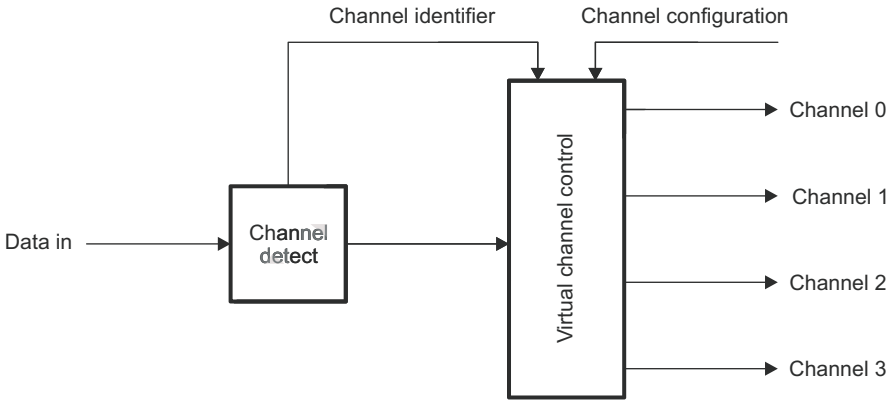
Figure 40. ISS CSI2 Data Identifier Structure



Virtual Channel

The CSI2 protocol layer transports virtual channels. Virtual channels are built of frames. A frame can comprise embedded data and image-sensor data. Two contexts are used to send the two types of data separately. Each frame is identified by unique mandatory synchronization codes: frame start and frame end. Line start and line end synchronization codes are optional for the transmitter. A set of registers is associated with each context defined by the virtual channel ID and the data type. Figure 41 shows a virtual channel.

Figure 41. ISS CSI2 Virtual Channel



2.6.1.1.3.4 ISS CSI2 Synchronization Codes

Data reception from the image-sensor module uses four synchronization codes embedded in the serial bitstream:

- FSC: Identifies the start of a new frame
- LSC: Identifies the start of a new line; received for every line
- LEC: Identifies the end of a line; received for every line
- FEC: Identifies the end of the last line and the end of the current frame

Table 64 summarizes the synchronization code values.

Table 64. ISS CSI2 Synchronization Codes

Synchronization Code	Value	Comments
FSC	0x0	Mandatory
FEC	0x1	Mandatory
LSC	0x2	Optional
LEC	0x3	Optional
Reserved	0x4 to 0x7	Not used

2.6.1.1.3.5 ISS CSI2 Generic Short Packet Codes

When the synchronization code value is from 0x8 to 0xF, the short packet is called a generic short packet. Short packets are not processed by the camera interface hardware. A generic short packet is stored in a register without the ECC and an interrupt can be generated. Therefore, generic short packets must be handled by software.

2.6.1.1.3.6 ISS CSI2 Generic Long Packet Codes

The code value 0x10 indicates null packets, which can be received at any time. They are discarded by the protocol engine.

The code value 0x11 indicates blanking packets, which can be received at any time. They are discarded by the protocol engine.

The code value 0x12 indicates embedded 8-bit nonimage data typically used for JPEG.

Code values from 0x13 to 0x17 are reserved.

2.6.1.1.3.7 ISS CSI2 Frame Structure

Each frame consists of short packets to indicate SOF and EOF. Optional short packets for start of line and end of line can be sent by the image sensor.

Some information before and after the picture data can be sent as SOF and EOF information by the image sensor to the memory through the L3 port.

For each frame, the pixel data (arbitrary data or user-defined byte data) are valid only after an SOF short packet. If the data are invalid, they are discarded by the protocol engine.

A frame contains embedded data and image-sensor data. Figure 42 shows where the embedded data and image-sensor data are in the frame. The frame is scanned in raster order starting from the top-left corner, as shown in Figure 42 and Figure 43. The following definitions for a frame apply:

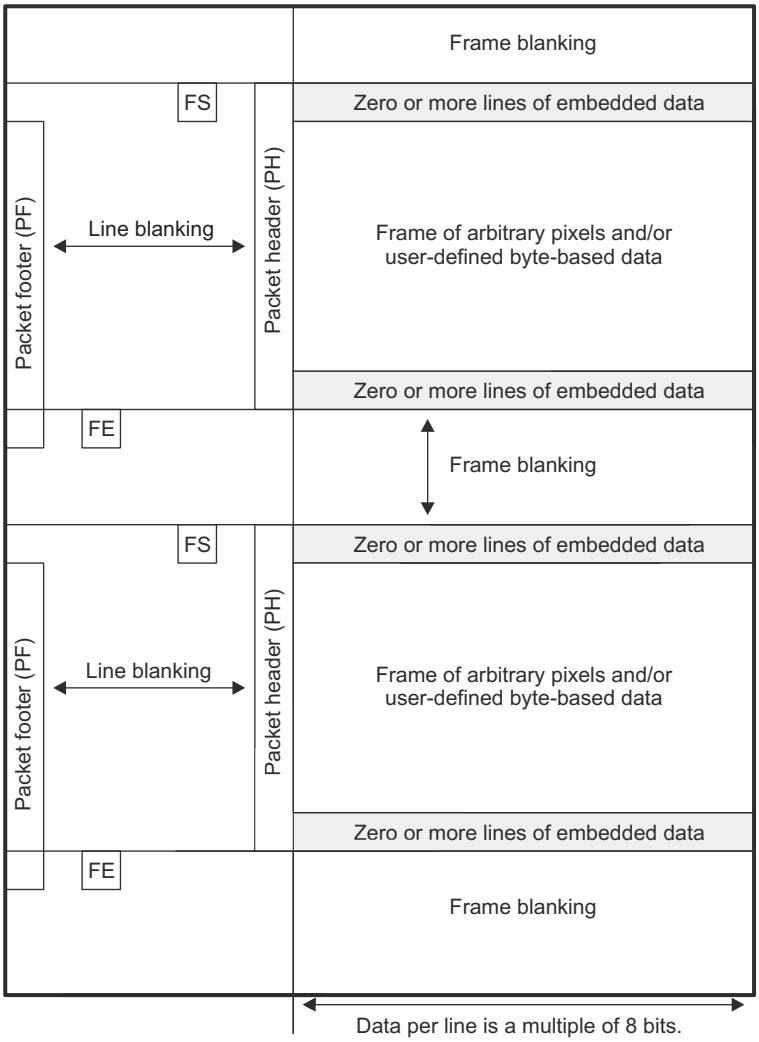
- Zero or more SOF status lines (SOF lines) can be embedded at the beginning of a CSI2 frame.
- The image embedded data is carried using separate data types and virtual channels (see Section 2.6.3.3.4, *ISS CSI2 Virtual Channel and Context*).
- Zero or more EOF status lines (EOF lines) can be embedded at the end of a CSI2 frame.
- The SOF lines, pixel data, and EOF lines do not overlap.

The CSI2 receiver does not use the information in the status lines. However, it extracts it and stores it in memory for use by software.

Because the data types are different, the data is carried using separate data types called virtual channels. Those must be mapped to the adequate context. The CSI2 receiver uses a different context for embedded data and image-sensor data. See Section 2.6.3.3.4, *ISS CSI2 Virtual Channel and Context*.

Embedded data is supported as a context by the CSI2 receiver; therefore, there is no specific hardware support for embedded data.

Figure 42. ISS CSI2 General Frame Structure (Informative)

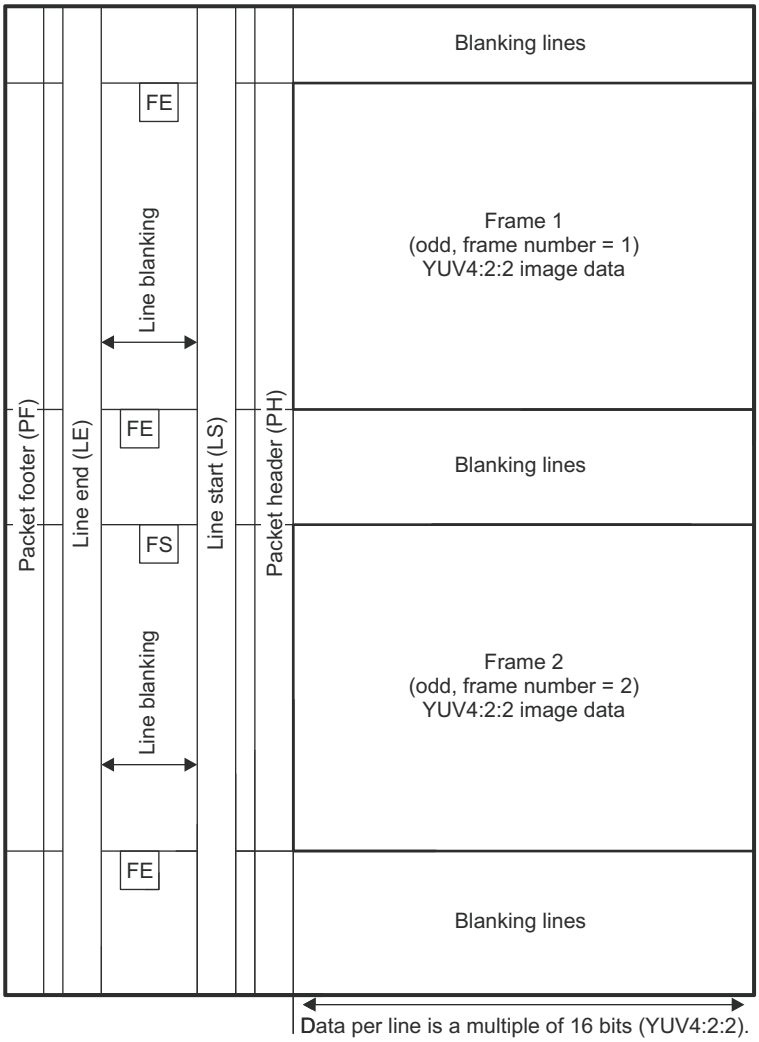


Key:

PH: Packet header
PF: Packet footer
FS: Frame start
FE: Frame end

Figure 43 shows the frame structure of a YUV4:2:2 interlaced video frame without embedded data.

Figure 43. ISS CSI2 Digital Interlaced Video Frame (Informative)



Key:

PH: Packet header	PF: Packet footer
FS: Frame start	FE: Frame end
LS: Line start	LE: Line end

The period between the LEC and the new LSC is the line blanking period. The time between the FEC and the new FSC is the frame blanking period. The receiver works with the line blanking period set to 0. The image data is stored in memory by selecting one of the various operating modes. Section 2.6.1.1.4, *ISS CSI2 Operating Modes*, explains storing image data frames into memory.

2.6.1.1.4 ISS CSI2 Operating Modes

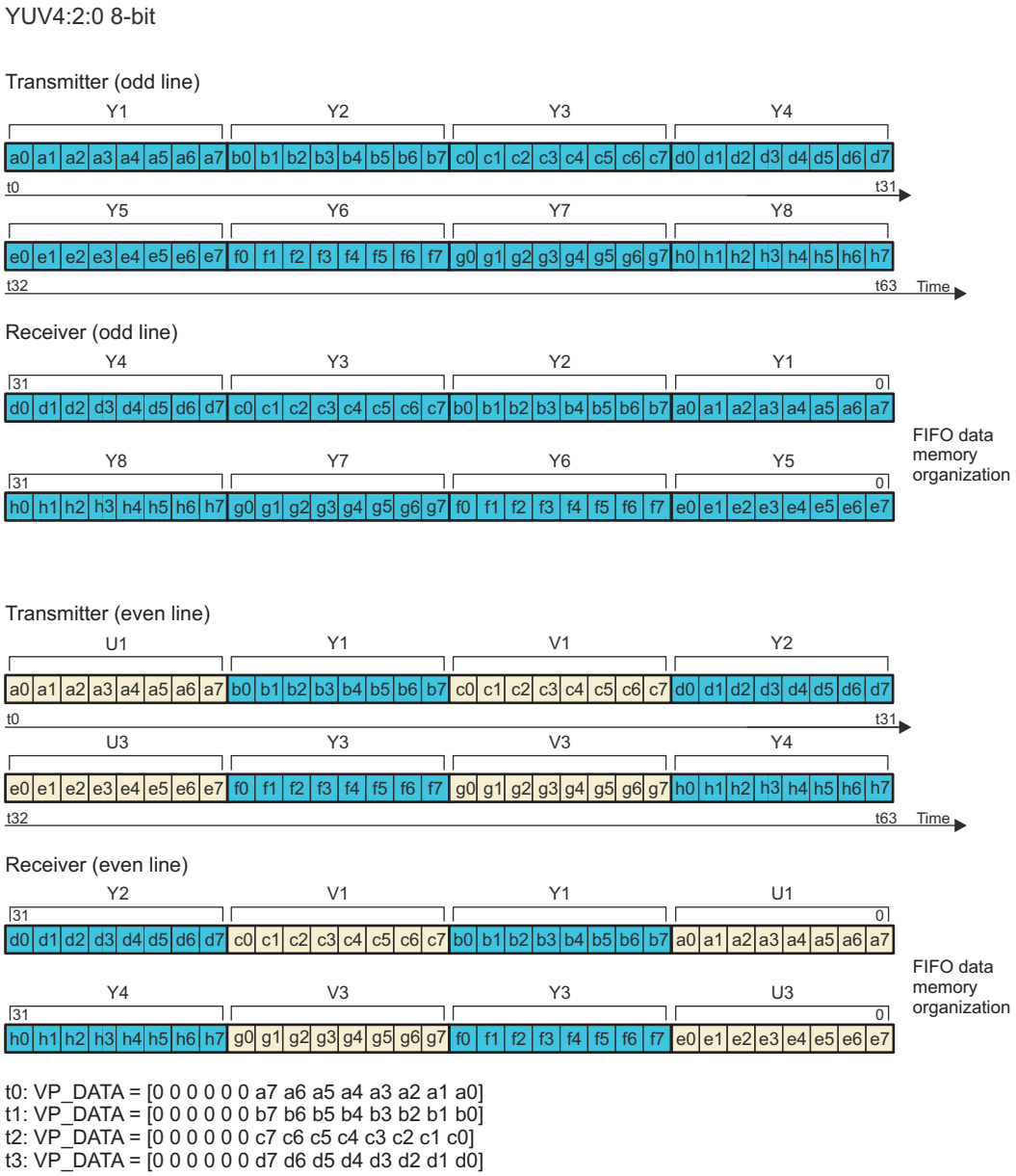
2.6.1.1.4.1 ISS CSI2 YUV Operating Modes

2.6.1.1.4.1.1 ISS CSI2 YUV4:2:0 8-Bit

YUV4:2:0 8-bit data can be stored to memory in little-endian and big-endian format. To set the endianness for YUV4:2:0 use the CSI2_CTRL[4] ENDIANNESS bit. The line length sent through the CSI2 physical protocol is a multiple of 16 bits for odd lines and 32 bits for even lines.

For correct pixel reconstruction, the line length must be a multiple of 32 bits and the number of lines must be even. Figure 44 shows the storage format for YUV4:2:0 8-bit data. It is shown as little endian. If the data format is big endian, the figure changes accordingly. Set the CSI2_CTX_CTRL2_i[9:0] FORMAT bit field to 0x18 to select YUV4:2:0 8-bit mode. Even and odd lines do not have the same length. Offset must be set accordingly with the CSI2_CTX_DAT_OFST_i[16:5] OFST bit field; for example, if the offset is 0, the data is written in a contiguous way (bit-to-bit of odd and even lines). If the data has an offset, set the destination offset between the first pixel of the previous line and the first pixel of the current line being written to memory.

Figure 44. ISS CSI2 YUV4:2:0 8-Bit

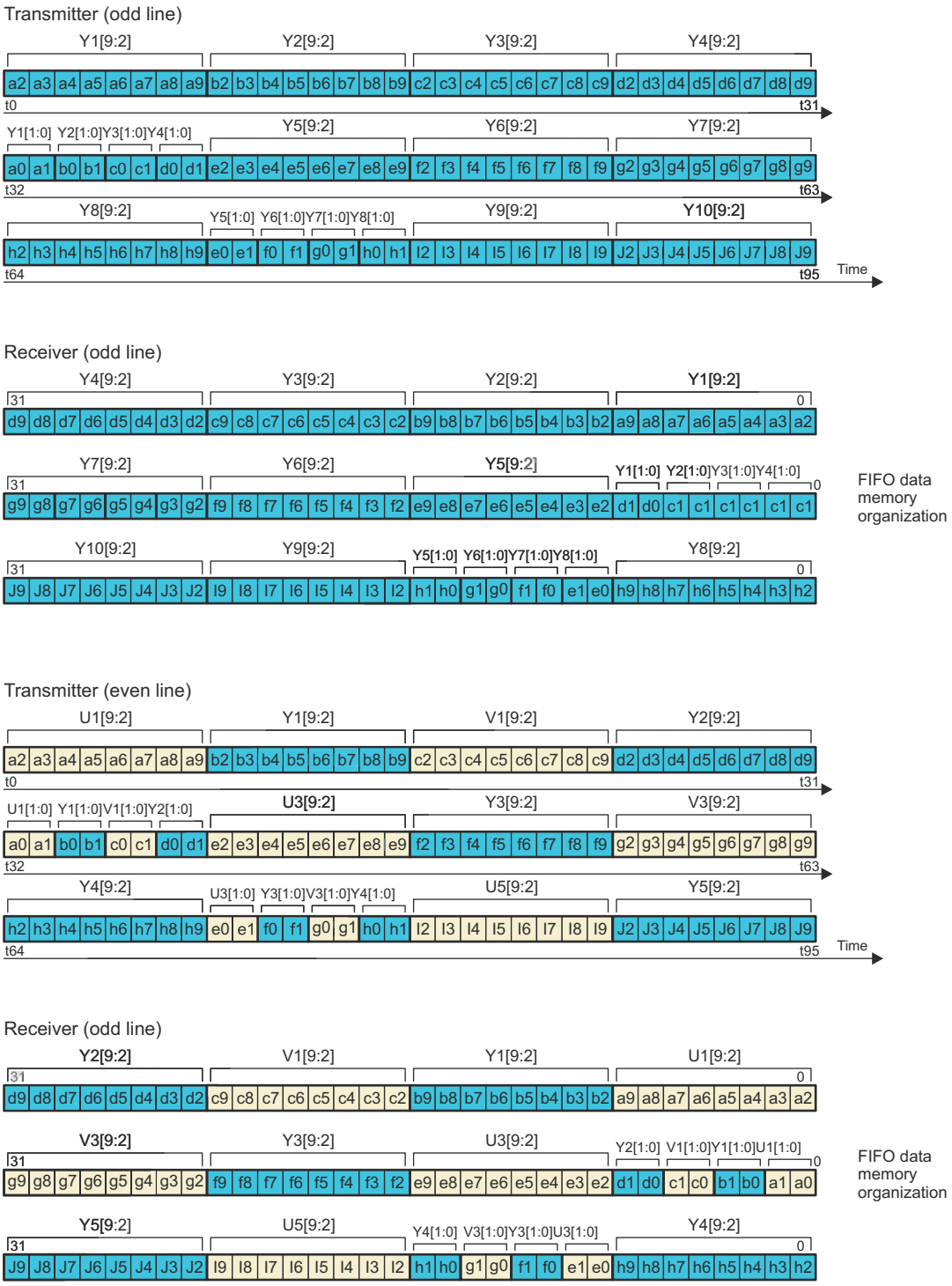


t0: VP_DATA = [0 0 0 0 0 0 a7 a6 a5 a4 a3 a2 a1 a0]
t1: VP_DATA = [0 0 0 0 0 0 b7 b6 b5 b4 b3 b2 b1 b0]
t2: VP_DATA = [0 0 0 0 0 0 c7 c6 c5 c4 c3 c2 c1 c0]
t3: VP_DATA = [0 0 0 0 0 0 d7 d6 d5 d4 d3 d2 d1 d0]

2.6.1.1.4.1.2 ISS CSI2 YUV4:2:0 10-Bit

YUV4:2:0 10-bit data can be stored to memory in little-endian and big-endian format. The line length sent through the CSI2 physical protocol is a multiple of 40 bits for odd lines and 80 bits for even lines. Figure 45 shows the storage format for YUV4:2:0 10-bit data. Set the CSI2_CTX_CTRL2_i[9:0] FORMAT bit field to 0x19 to select YUV4:2:0 10-bit mode. Even and odd lines do not have the same length. Offset must be set accordingly with the CSI2_CTX_DAT_OFST_i[16:5] OFST bit field.

Figure 45. ISS CSI2 YUV4:2:0 10-Bit

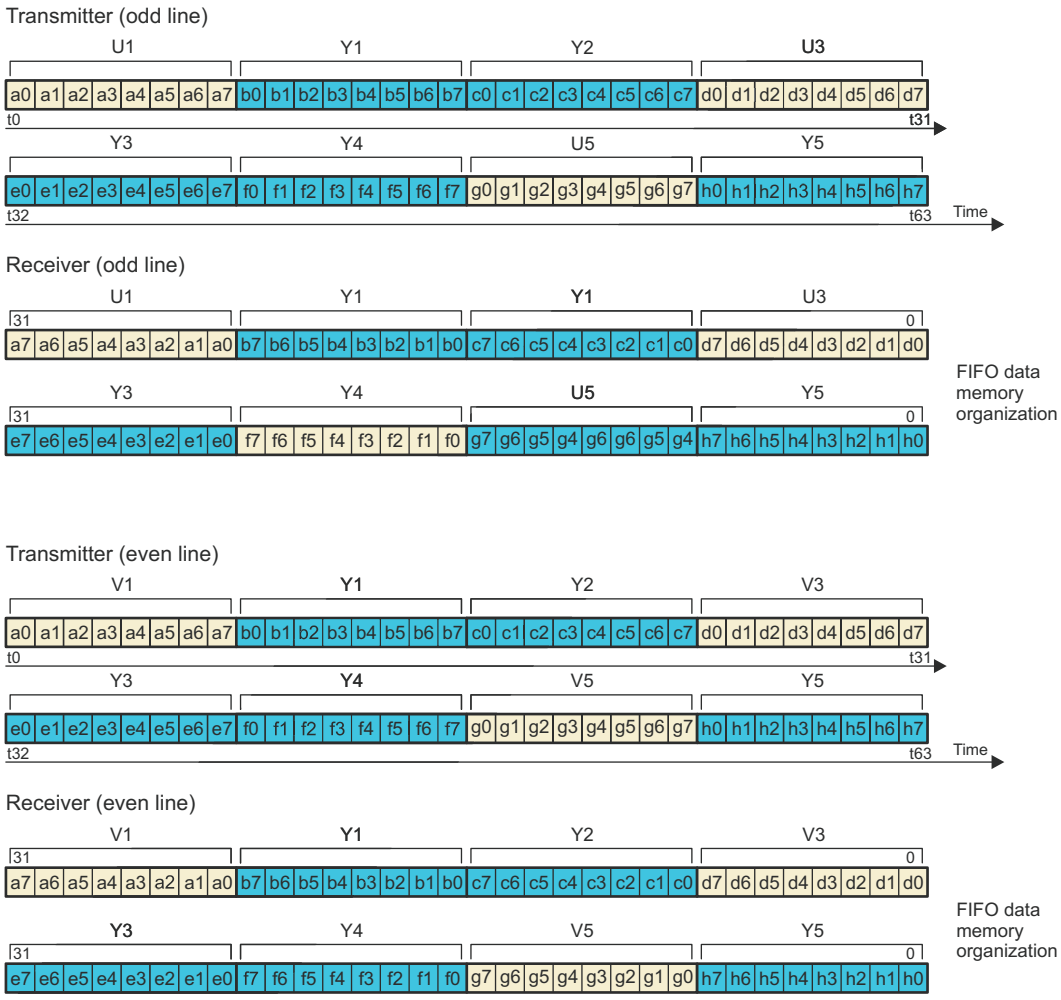


2.6.1.1.4.1.3 CSI2 YUV4:2:0 8-Bit Legacy

YUV4:2:0 8-bit legacy data can be stored to memory in little-endian and big-endian format. The line length sent through the CSI2 physical protocol is a multiple of 4 bytes. Figure 46 shows the storage format for YUV4:2:0 8-bit legacy data. Set the CSI2_CTX_CTRL2_i[9:0] FORMAT bit field to 0x1A to select YUV4:2:0 8-bit legacy mode.

Figure 46. ISS CSI2 YUV4:2:0 8-Bit Legacy

YUV4:2:0 8-bit legacy

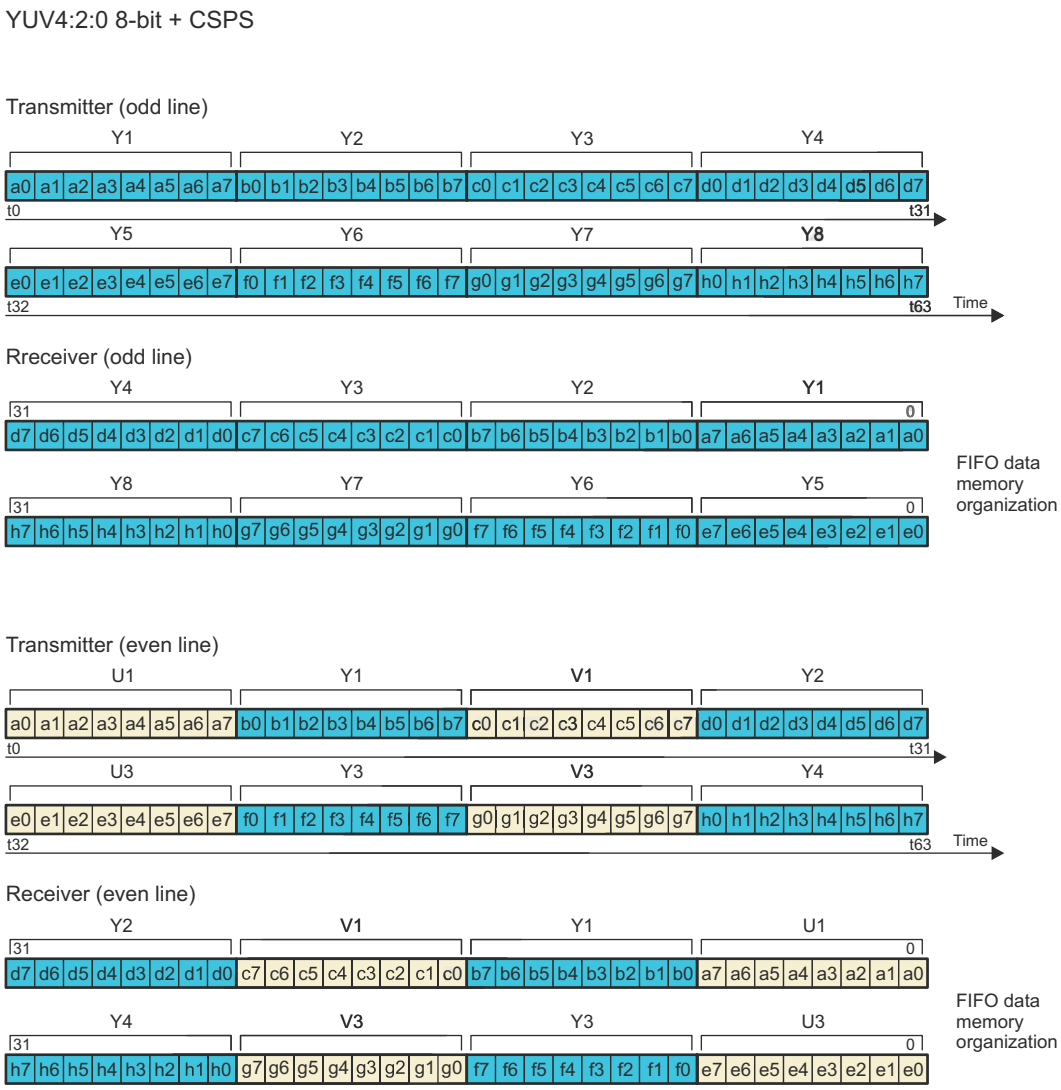


2.6.1.1.4.1.4 ISS CSI2 YUV4:2:0 8-Bit + CSPS

YUV4:2:0 8-bit CSPS data can be stored to memory in little-endian and big-endian format. The line length sent through the CSI2 physical protocol is a multiple of 16 bits for odd lines and 32 bits for even lines.

For correct pixel reconstruction, the line length must be a multiple of 3 32 bits and the number of lines must be even. Figure 47 shows the storage format for YUV4:2:0 8-bit + CSPS data. Set the CSI2_CTX_CTRL2_i[9:0] FORMAT bit field to 0x1C to select YUV4:2:0 8-bit + CSPS mode.

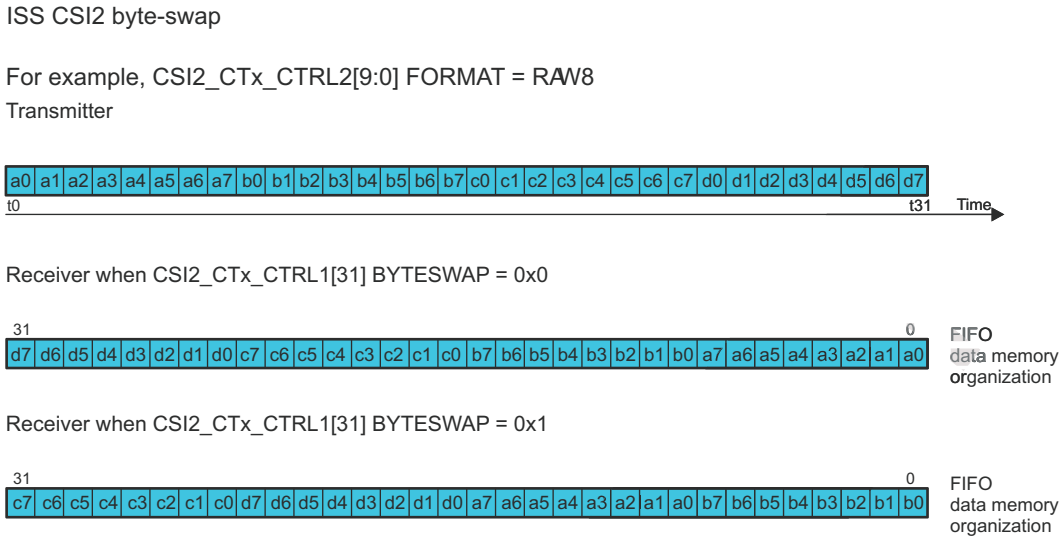
Figure 47. ISS CSI2 YUV4:2:0 8-Bit + CSPS



2.6.1.1.4.1.5 Camera ISP CSI2 Byte Swap

The CSI2 receiver incorporates a byte-swapping function. Software can optionally enable byte-swapping of the payload data by setting the CSI2_CTx_CTRL1[31] BYTESWAP bit. This feature must be used only when the amount of payload data per packet is a multiple of 16 bits. The byte-swapping is performed before pixel reconstruction.

Figure 48. Camera ISP CSI2 Byte Swap

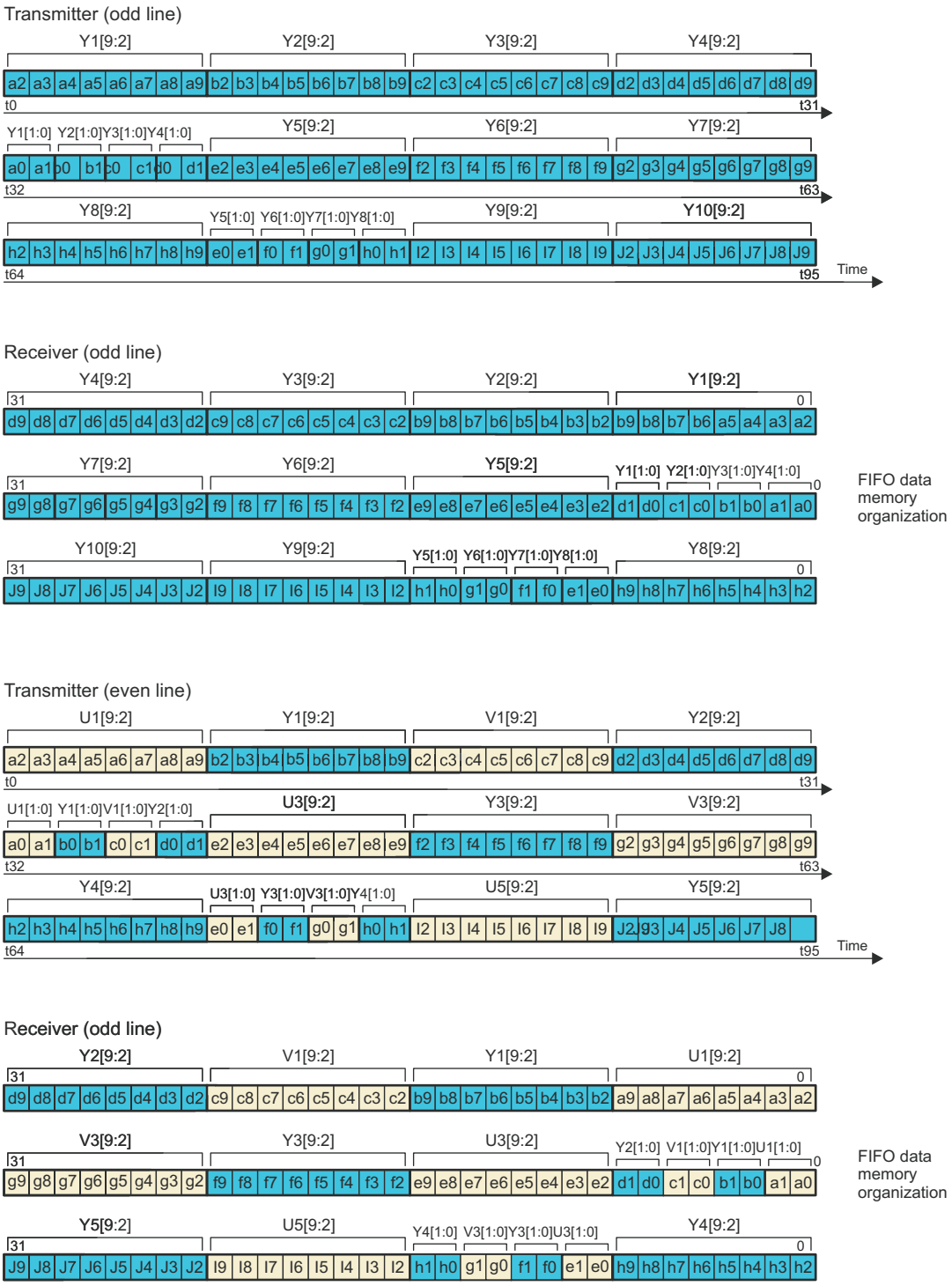


2.6.1.1.4.1.6 ISS CSI2 YUV4:2:0 10-Bit + CSPS

YUV4:2:0 10-bit CSPS data can be stored to memory in little-endian and big-endian format. The line length sent through the CSI2 physical protocol is a multiple of 40 bits for odd lines and 80 bits for even lines.

For correct pixel reconstruction, the line length must be a multiple of 3 32 bits and the number of lines must be even. Figure 49 shows the storage format for YUV4:2:0 10-bit + CSPS data. Set the CSI2_CTX_CTRL2_i [9:0] FORMAT bit field to 0x1D to select YUV4:2:0 10-bit + CSPS mode.

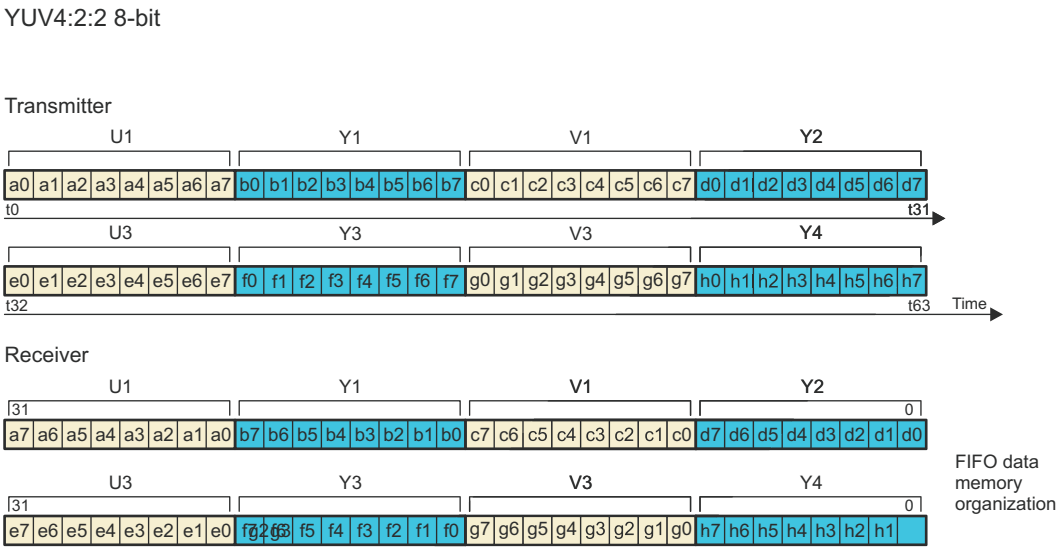
Figure 49. ISS CSI2 YUV4:2:0 10-Bit + CSPS



2.6.1.1.4.1.7 ISS CSI2 YUV4:2:2 8-Bit

YUV4:2:2 data can be stored to memory in little-endian and big-endian format. To set the endianness for YUV4:2:2 use the CSI2_CTRL[4] ENDIANNESS bit. The line length sent through the CSI2 physical protocol is a multiple of 32 bits. Figure 50 shows the storage format for YUV4:2:2 8-bit data. Set the CSI2_CTX_CTRL2_i[9:0] FORMAT bit field to 0x1E to select YUV4:2:2 8-bit mode.

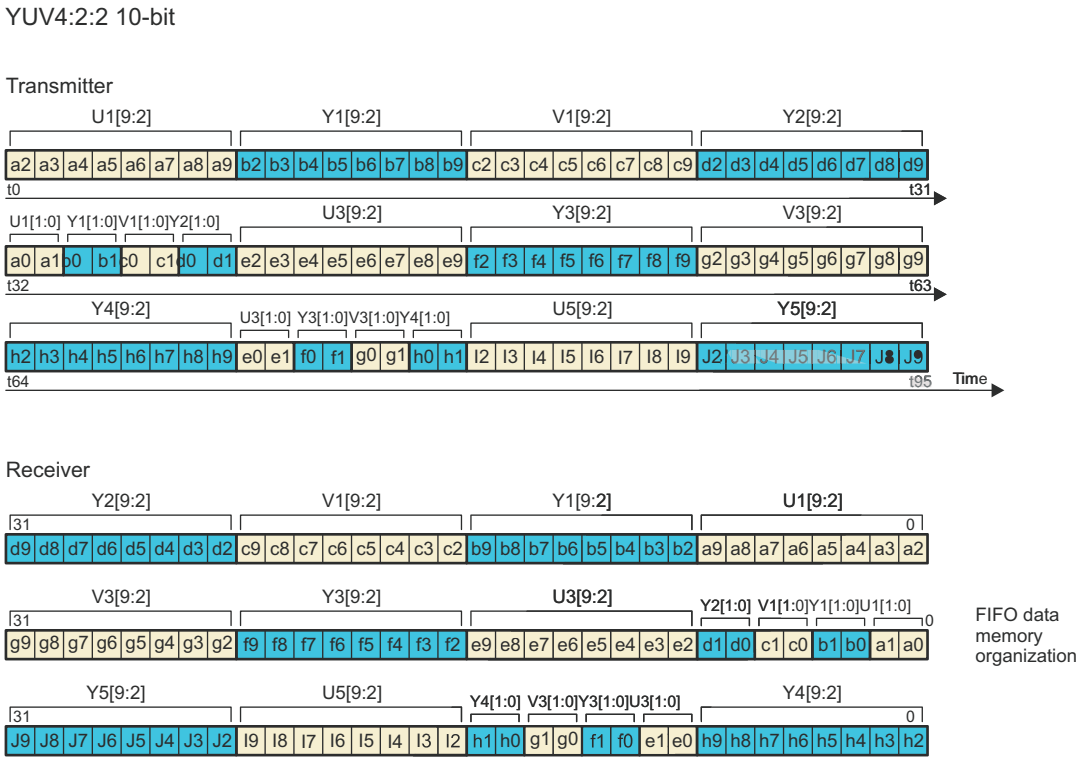
Figure 50. ISS CSI2 YUV4:2:2 8-Bit



2.6.1.1.4.1.8 CSI2 YUV4:2:2 10-Bit

YUV4:2:2 data can be stored to memory in little-endian and big-endian format. The line length sent through the CSI2 physical protocol is a multiple of 40 bits. Figure 51 shows the storage format for YUV4:2:2 10-bit data. Set the CSI2_CTX_CTRL2_i[9:0] FORMAT bit field to 0x1F to select YUV4:2:2 10-bit mode.

Figure 51. ISS CSI2 YUV4:2:2 10-Bit

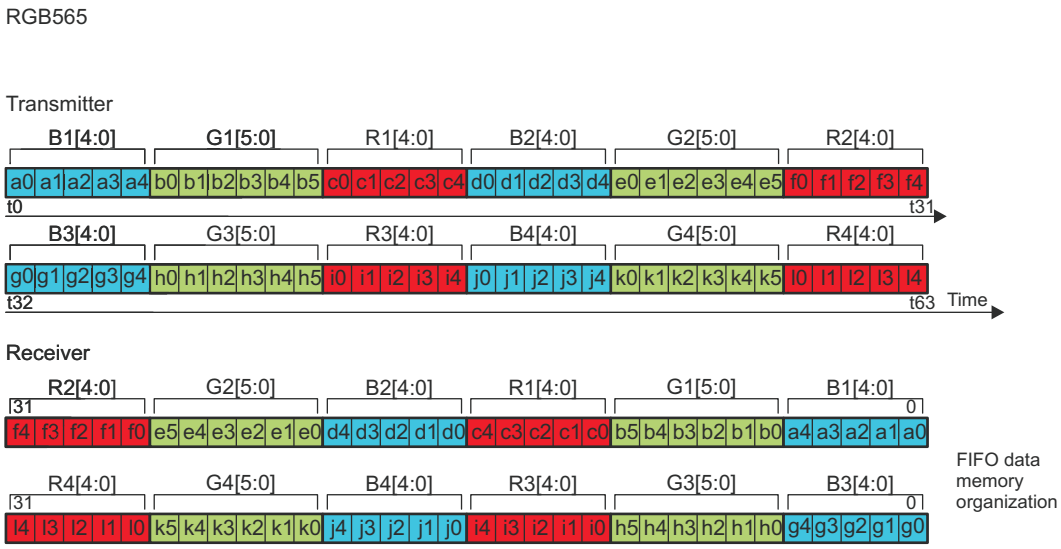


2.6.1.1.4.2 ISS CSI2 RGB Operating Modes

2.6.1.1.4.2.1 ISS CSI2 RGB565

RGB565 data is output to memory without data expansion. The line length sent through the CSI2 physical layer is always a multiple of 16 bits. Figure 52 shows the storage format for RGB565 data. Set the CSI2_CTX_CTRL2_i[9:0] FORMAT bit field to 0x22 to select RGB565 mode.

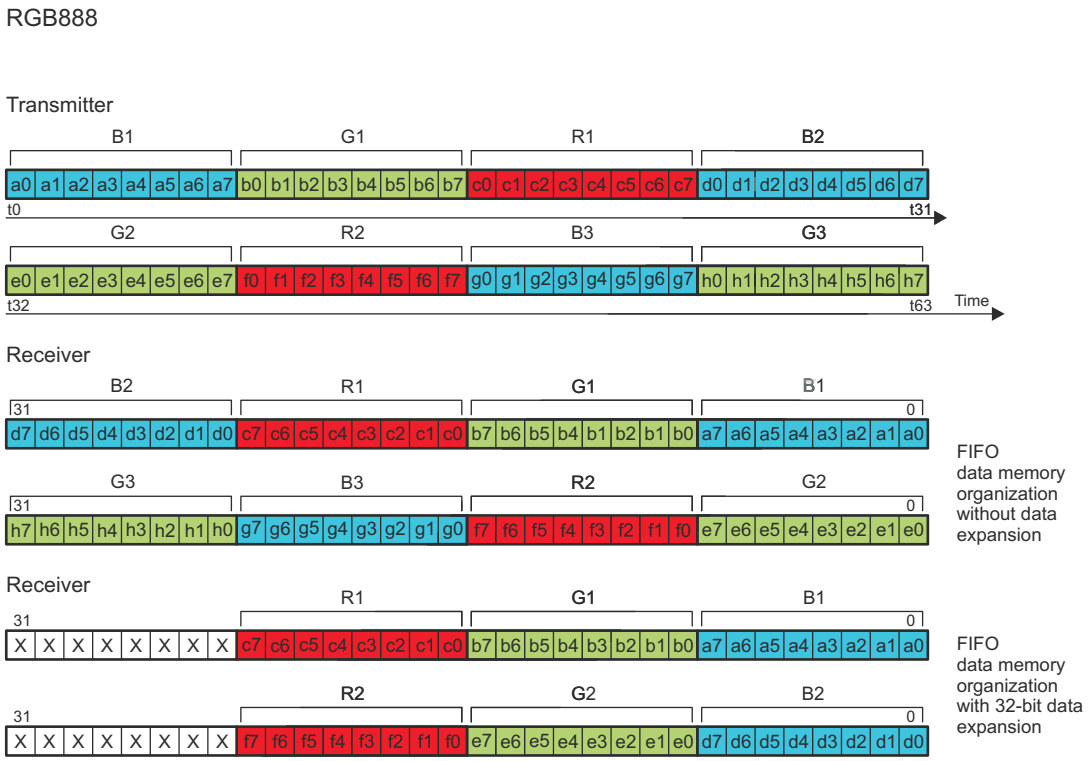
Figure 52. ISS CSI2 RGB565



2.6.1.1.4.2.2 ISS CSI2 RGB888

RGB888 data can be output to memory in two formats: with or without data expansion. If data expansion is used, the value of the 8 upper bits is programmable and can be set with an alpha value for computer graphics applications (the CSI2_CTX_CTRL3_i[29:16] ALPHA bit field). Figure 53 shows the storage format for RGB888 data. Set the CSI2_CTX_CTRL2_i [9:0] FORMAT bit field to 0x24 to select RGB888 mode.

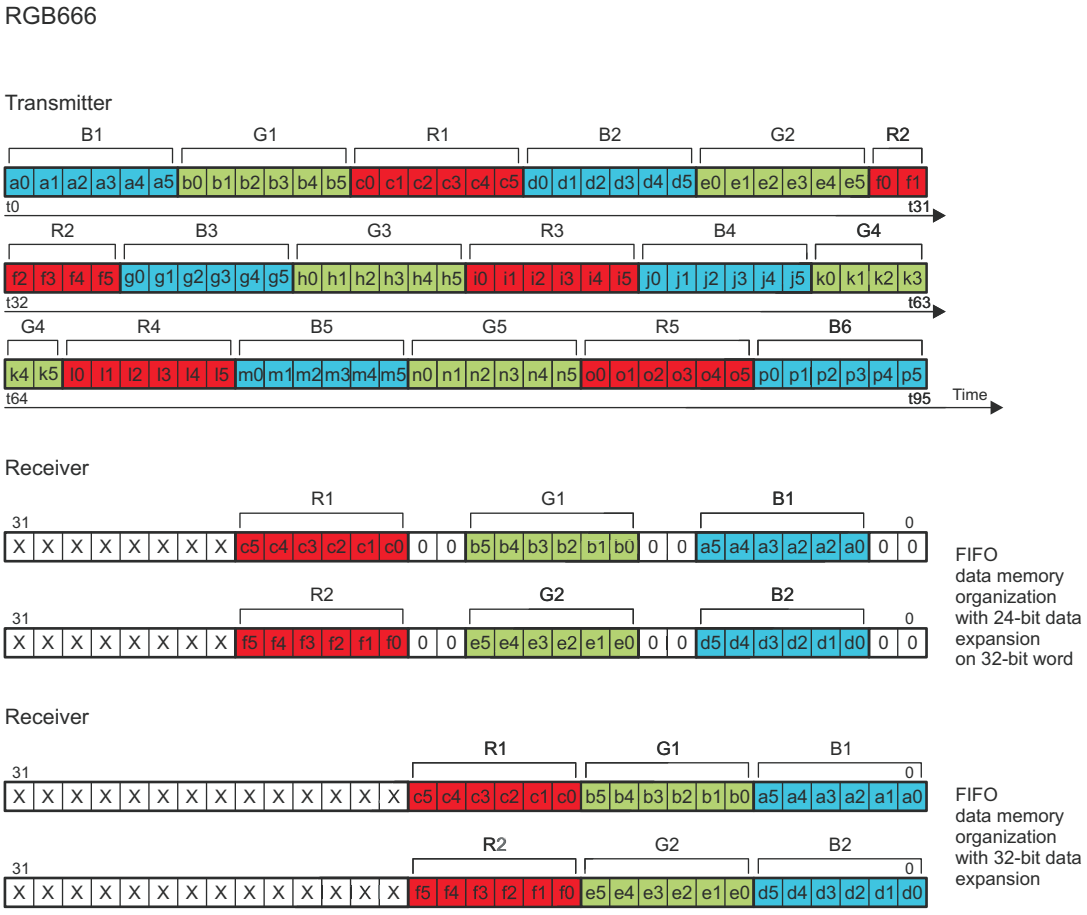
Figure 53. ISS CSI2 RGB888



2.6.1.1.4.2.3 ISS CSI2 RGB666

RGB666 data is always output to memory with data expansion. The value of the 14 upper bits is programmable and can be set with an alpha value for computer graphics applications (the CSI2_CTX_CTRL3_i[29:16] ALPHA bit field). The line length sent through the CSI2 physical protocol is a multiple of 8 bits. Furthermore, the line length is a multiple of 9 × 8 bits to finish the pixel reconstruction correctly. Figure 54 shows the storage format for RGB666 data. Set the CSI2_CTX_CTRL2_i[9:0] FORMAT bit field to 0x33 to select RGB666 mode.

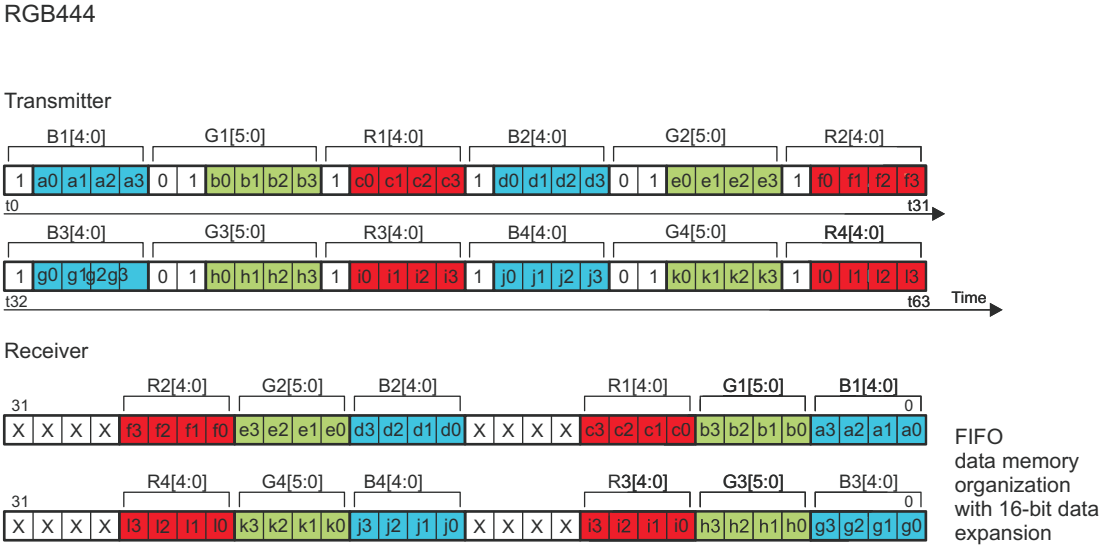
Figure 54. ISS CSI2 RGB666



2.6.1.1.4.2.4 ISS CSI2 RGB444

RGB444 data is output to memory with data expansion. When data expansion is used, the value of the 4 upper bits is programmable and can be set with an alpha value for computer graphics applications (the CSI2_CTX_CTRL3_i[29:16] ALPHA bit field). Figure 55 shows the storage format for RGB444 data. Set the CSI2_CTX_CTRL2_i[9:0] FORMAT bit field to 0xA0 to select RGB444 mode.

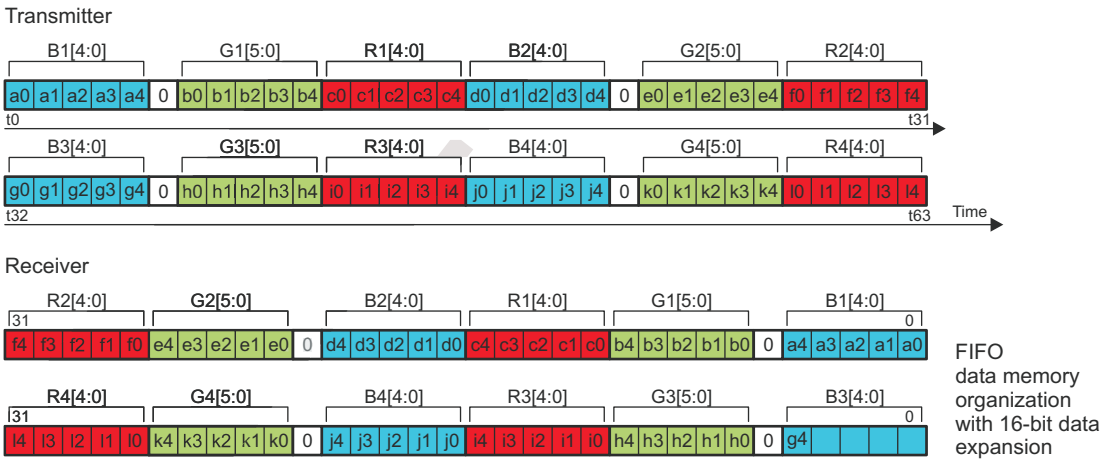
Figure 55. ISS CSI2 RGB444



2.6.1.1.4.2.5 ISS CSI2 RGB555

RGB555 data is output to memory with data expansion. Figure 56 shows the storage format for RGB555 data. Set the CSI2_CTX_CTRL2_i[9:0] FORMAT bit field to 0xA1 to select RGB555 mode.

Figure 56. ISS CSI2 RGB555



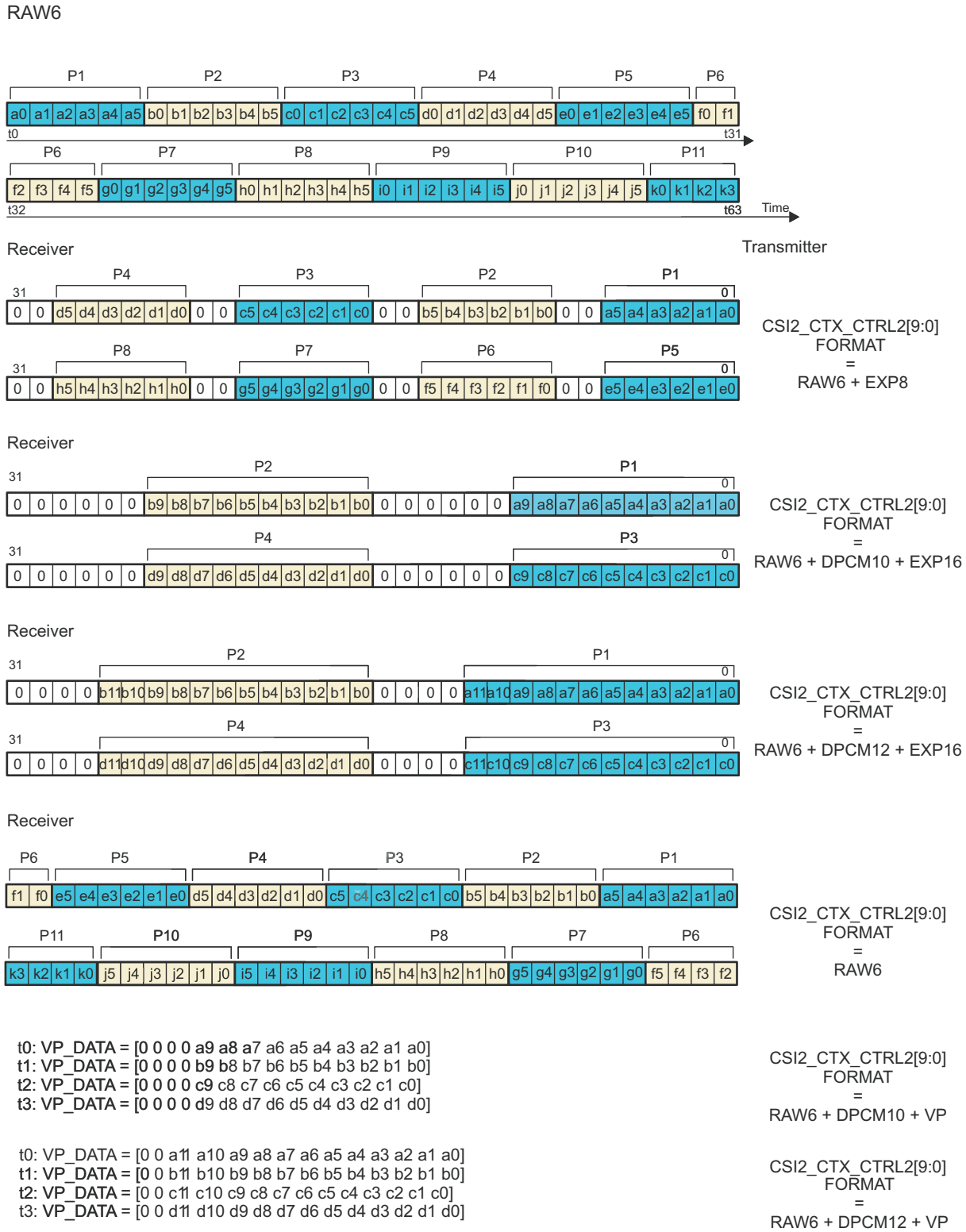
2.6.1.1.4.3 ISS CSI2 RAW Bayer RGB Operating Modes

2.6.1.1.4.3.1 ISS CSI2 RAW6

RAW6 data can be output to memory with or without data expansion. The line length sent through the CSI2 physical layer is a multiple of 8 bits (6-bit image data + 2-bit expansion). Furthermore, the line length is a multiple of 3 × 8 bits to complete the pixel reconstruction correctly (the lowest common multiple of 8 and 6 is 24, so 3 × 8 bits). Figure 57 shows the storage format for RAW6 data. Set the CSI2_CTX_CTRL2_i[9:0] FORMAT bit field as follows:

- To 0x28 to select RAW6 mode
- To 0x68 for RAW6 + 8-bit expansion
- To 0xE8 for RAW6 + DPCM decompression to 10-bit to video port
- To 0x2A8 for RAW6 + DPCM decompression to 10-bit expanded to 16-bit
- To 0x3A8 for RAW6 + DPCM decompression to 12-bit expanded to 16-bit
- To 0x368 for RAW6 + DPCM decompression to 12-bit to video port

Figure 57. ISS CSI2 RAW6

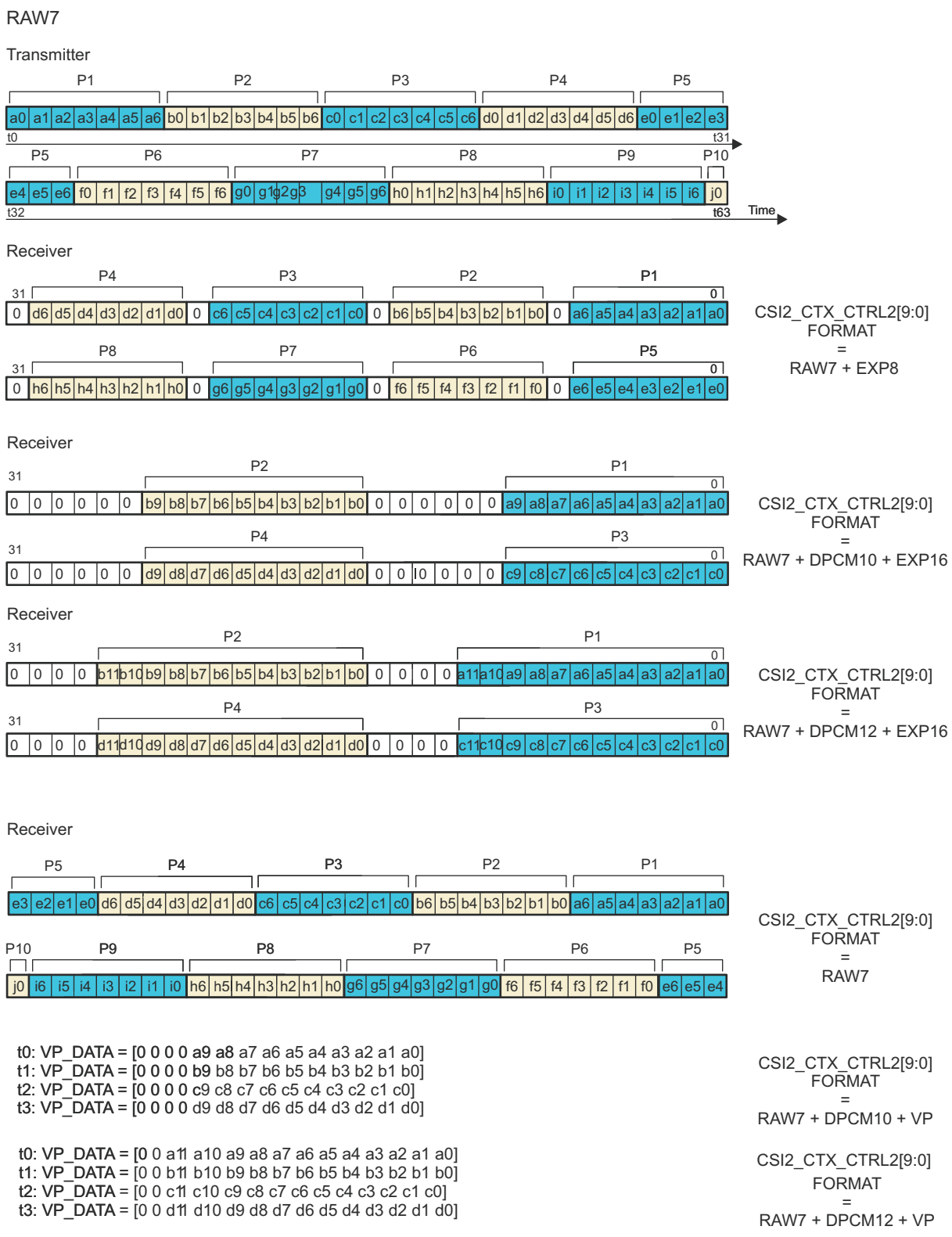


2.6.1.1.4.3.2 ISS CSI2 RAW7

RAW7 data can be output to memory with or without data expansion. The line length sent through the CSI2 physical layer is a multiple of 8 bits. Furthermore, the line length is a multiple of 7 × 8 bits to complete the pixel reconstruction correctly (the lowest common multiple of 8 and 7 is 56, so 7 × 8 bits). Figure 58 shows the storage format for RAW7 data. Set the CSI2_CTX_CTRL2_i [9:0] FORMAT bit field as follows:

- To 0x29 to select RAW7 mode
- To 0x69 for RAW7 + 8-bit expansion
- To 0x329 for RAW7 + DPCM decompression to 10-bit to video port
- To 0x229 for RAW7 + DPCM decompression to 10-bit expanded to 16-bit
- To 0x369 for RAW7 + DPCM decompression to 12-bit expanded to 16-bit
- To 0x3A9 for RAW7 + DPCM decompression to 12-bit to video port

Figure 58. ISS CSI2 RAW7



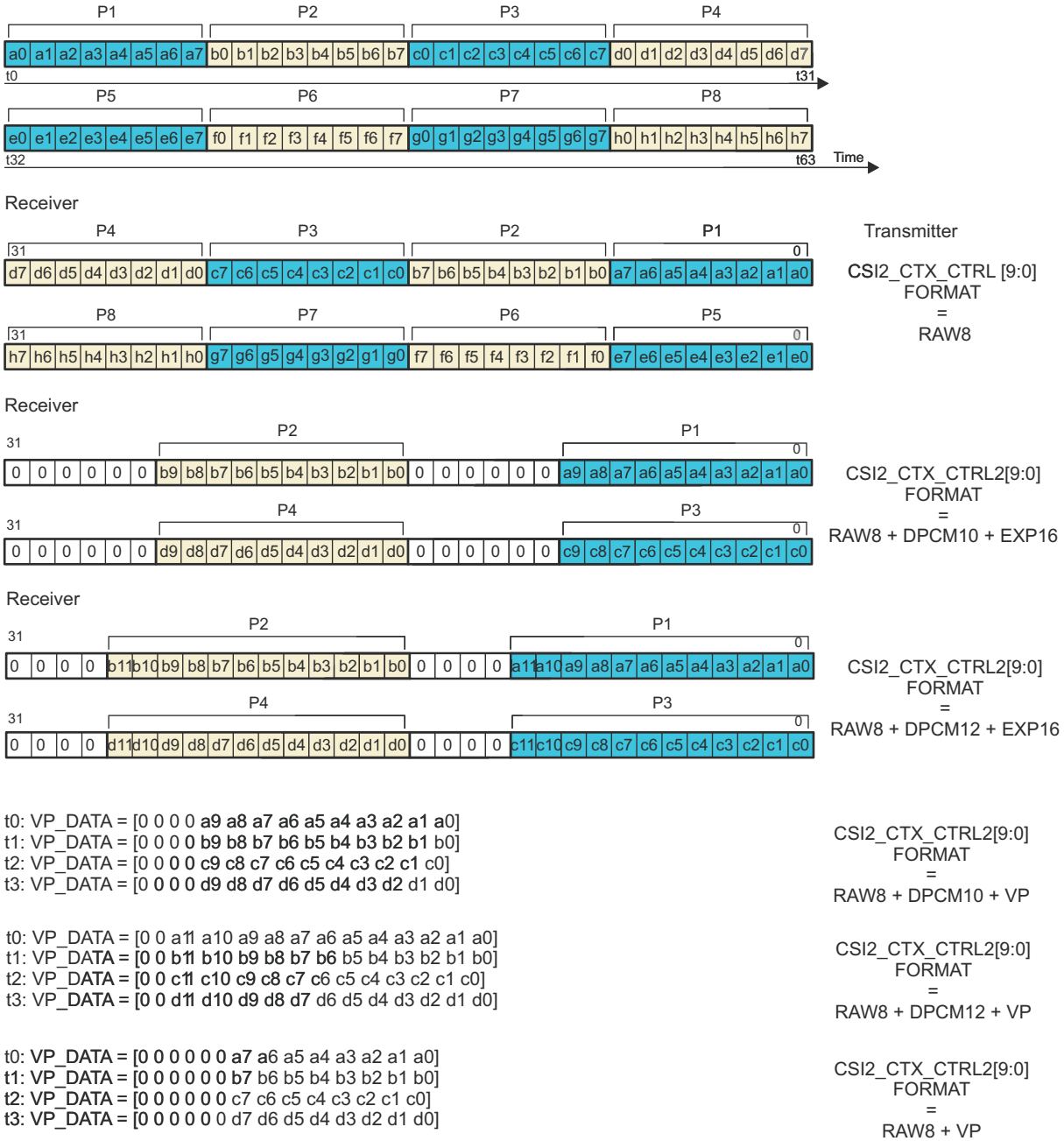
2.6.1.1.4.3.3 ISS CSI2 RAW8

RAW8 data can be output to memory with or without data expansion. The line length sent through the CSI2 physical layer is always a multiple of 8 bits. Figure 59 shows the storage format for RAW8 data. Set the CSI2_CTX_CTRL2_i[9:0] FORMAT bit field as follows:

- To 0x2A to select RAW8 mode
- To 0x12A for RAW8 to video port
- To 0x32A for RAW8 + DPCM decompression to 10-bit to video port
- To 0x2AA for RAW8 + DPCM decompression to 10-bit expanded to 16-bit
- To 0x36A for RAW8 + DPCM decompression to 12-bit expanded to 16-bit
- To 0x3AA for RAW8 + DPCM decompression to 12-bit to video port

Figure 59. ISS CSI2 RAW8

RAW8



t0: VP_DATA = [0 0 0 0 **a9 a8 a7 a6 a5 a4 a3 a2 a1** a0]
t1: VP_DATA = [0 0 0 **0 b9 b8 b7 b6 b5 b4 b3 b2 b1** b0]
t2: VP_DATA = [0 0 **0 0 c9 c8 c7 c6 c5 c4 c3 c2 c1** c0]
t3: VP_DATA = [0 **0 0 0 d9 d8 d7 d6 d5 d4 d3 d2** d1 d0]

CSI2_CTX_CTRL2[9:0]
FORMAT
=
RAW8 + DPCM10 + VP

t0: VP_DATA = [0 0 a11 a10 a9 a8 a7 a6 a5 a4 a3 a2 a1 a0]
t1: VP_DATA = **[0 0 b11 b10 b9 b8 b7 b6** b5 b4 b3 b2 b1 b0]
t2: VP_DATA = **[0 0 c11 c10 c9 c8 c7** c6 c5 c4 c3 c2 c1 c0]
t3: VP_DATA = **[0 0 d11 d10 d9 d8 d7** d6 d5 d4 d3 d2 d1 d0]

CSI2_CTX_CTRL2[9:0]
FORMAT
=
RAW8 + DPCM12 + VP

t0: VP_DATA = **[0 0 0 0 0 0 a7 a**6 a5 a4 a3 a2 a1 a0]
t1: VP_DATA = **[0 0 0 0 0 0 b7** b6 b5 b4 b3 b2 b1 b0]
t2: VP_DATA = **[0 0 0 0 0 0** c7 c6 c5 c4 c3 c2 c1 c0]
t3: VP_DATA = **[0 0 0 0 0** 0 d7 d6 d5 d4 d3 d2 d1 d0]

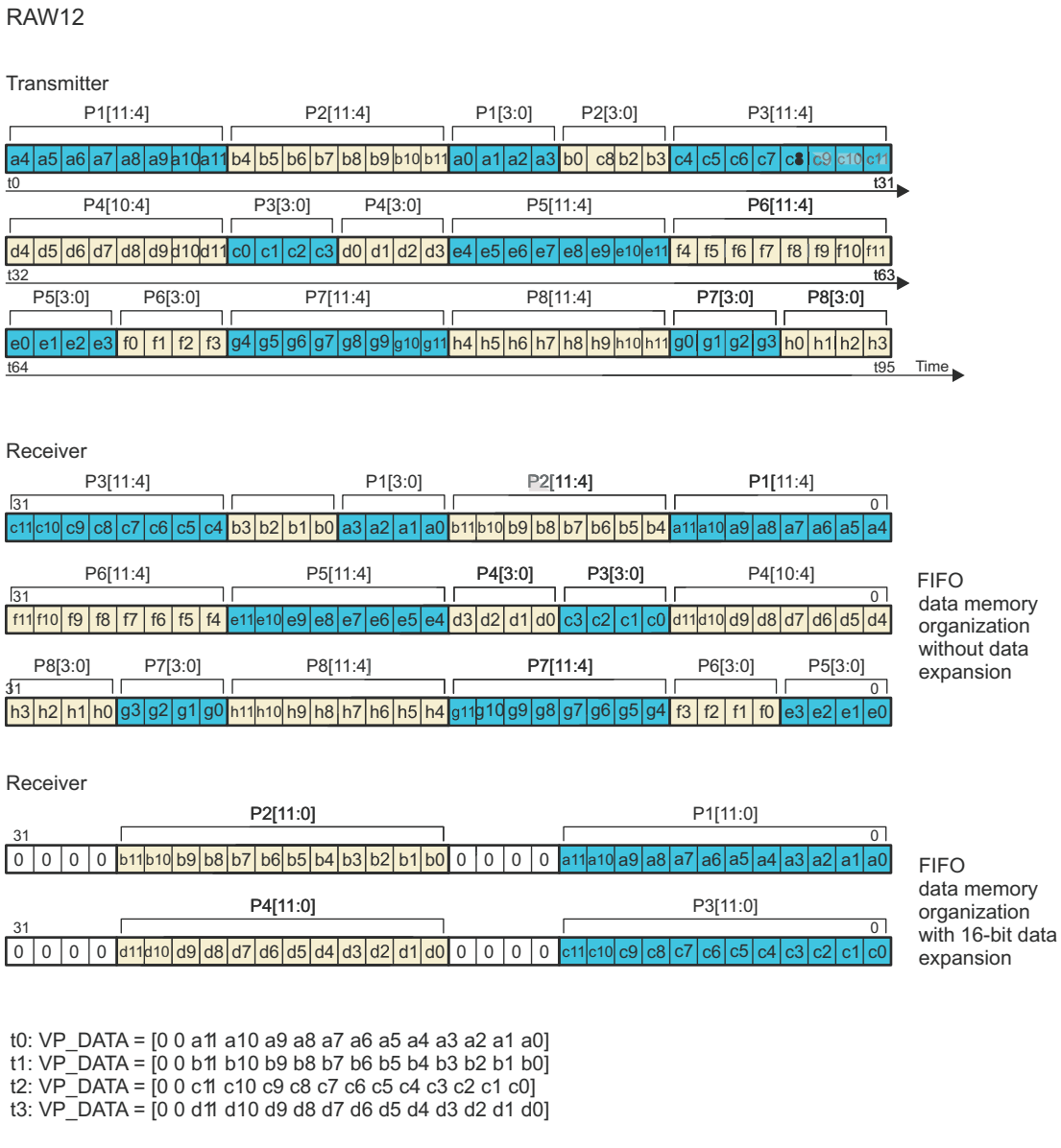
CSI2_CTX_CTRL2[9:0]
FORMAT
=
RAW8 + VP

2.6.1.1.4.3.4 ISS CSI2 RAW10

RAW10 data can be output memory in two formats: with or without data expansion. It can also be sent to the video port. If data expansion is used, the 10-bit data are padded with 0s on a 16-bit word. The line length sent through the CSI2 physical layer is a multiple of 8 bits. Furthermore, the line length is a multiple of 5 × 8 bits to complete the pixel reconstruction correctly (the lowest common multiple of 8 and 10 is 40, so 5 × 8 bits). Figure 60 shows the storage format for RAW10 data. Set the CSI2_CTX_CTRL2_i[9:0] FORMAT bit field as follows:

- To 0x2B to select RAW10 mode
- To 0xAB for RAW10 + 16-bit expansion
- To 0x12F for RAW10 to video port

Figure 60. ISS CSI2 RAW10



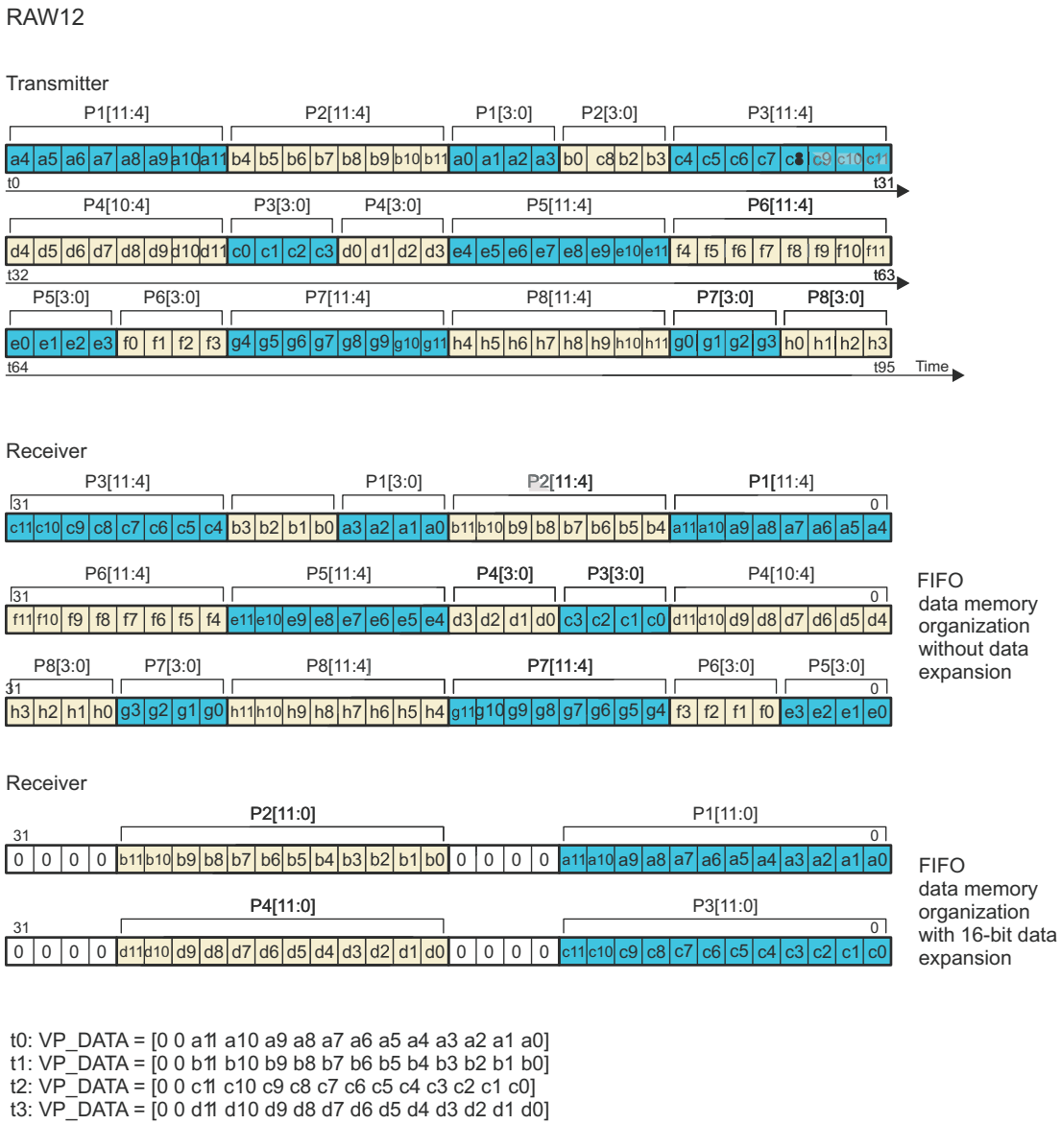
t0: VP_DATA = [0 0 a11 a10 a9 a8 a7 a6 a5 a4 a3 a2 a1 a0]
t1: VP_DATA = [0 0 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0]
t2: VP_DATA = [0 0 c11 c10 c9 c8 c7 c6 c5 c4 c3 c2 c1 c0]
t3: VP_DATA = [0 0 d11 d10 d9 d8 d7 d6 d5 d4 d3 d2 d1 d0]

2.6.1.1.4.3.5 ISS CSI2 RAW12

RAW12 data can be output to memory in two formats: with or without data expansion. It can also be sent to the video port. If data expansion is used, the 12-bit data are padded with 0s on a 16-bit word. The line length sent through the CSI2 physical layer is a multiple of 8 bits. Furthermore, the line length is a multiple of 3 × 8 bits to complete the pixel reconstruction correctly (the lowest common multiple of 8 and 12 is 24, so 3 × 8 bits). Figure 61 shows the storage format for RAW12 data. Set the CSI2_CTX_CTRL2_i[9:0] FORMAT bit field as follows:

- To 0x2C to select RAW12 mode
- To 0xAC for RAW12 + 16-bit expansion
- To 0x12C for RAW12 to video port

Figure 61. ISS CSI2 RAW12



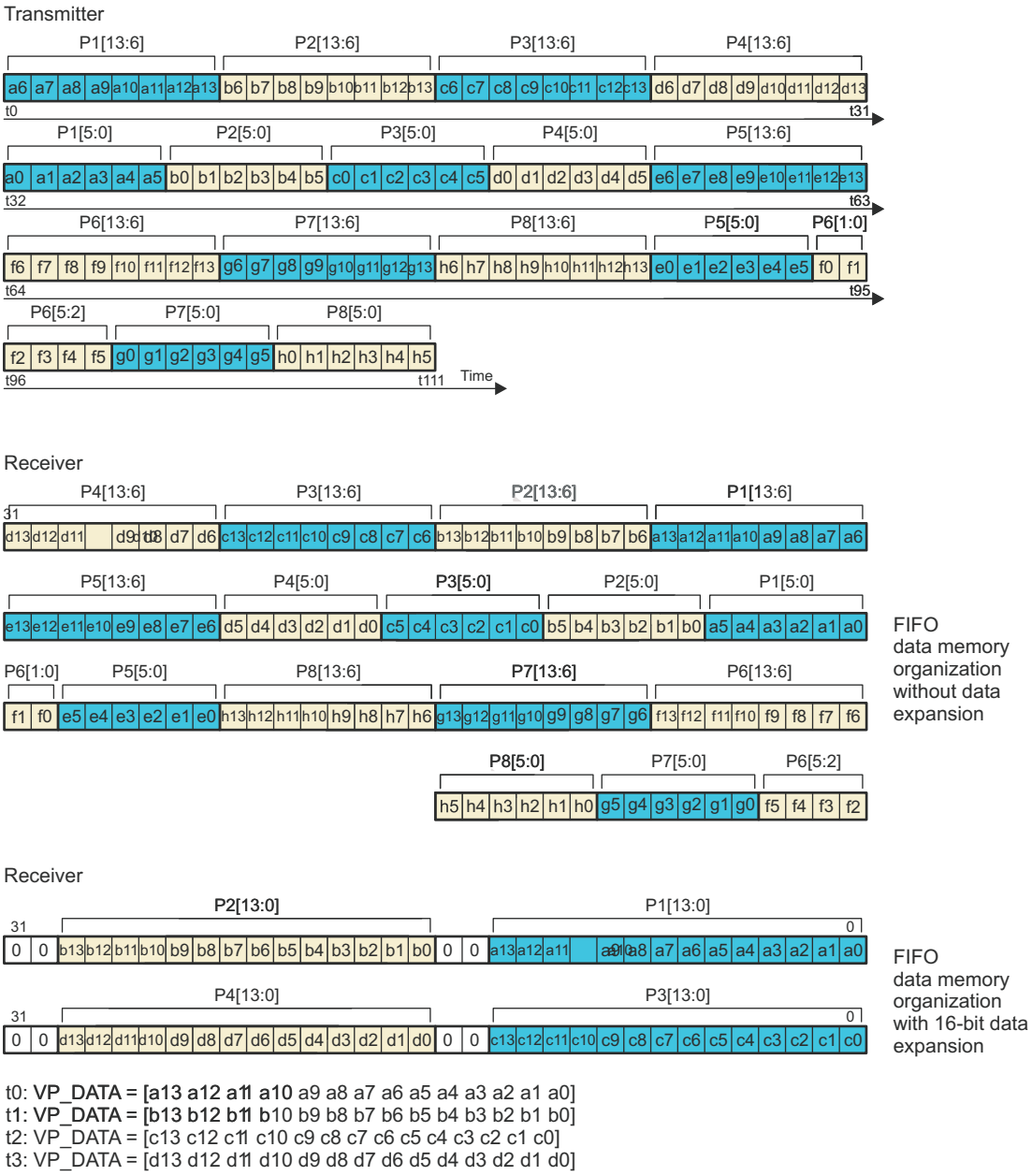
t0: VP_DATA = [0 0 a11 a10 a9 a8 a7 a6 a5 a4 a3 a2 a1 a0]
t1: VP_DATA = [0 0 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0]
t2: VP_DATA = [0 0 c11 c10 c9 c8 c7 c6 c5 c4 c3 c2 c1 c0]
t3: VP_DATA = [0 0 d11 d10 d9 d8 d7 d6 d5 d4 d3 d2 d1 d0]

2.6.1.1.4.3.6 ISS CSI2 RAW14

RAW14 data can be output to memory in two formats: with or without data expansion. It can also be sent to the video port. If data expansion is used, the 14-bit data are padded with 0s on a 16-bit word. The line length sent through the CSI2 physical layer is a multiple of 8 bits. Furthermore, the line length is a multiple of 7 × 8 bits to complete the pixel reconstruction correctly (the lowest common multiple of 8 and 14 is 56, so 7 × 8 bits). Figure 62 shows the storage format for RAW14 data. Set the CSI2_CTX_CTRL2_i[9:0] FORMAT bit field as follows:

- To 0x2D to select RAW14 mode
- To 0xAD for RAW14 + 16-bit expansion
- To 0x12D for RAW12 to video port

Figure 62. ISS CSI2 RAW14

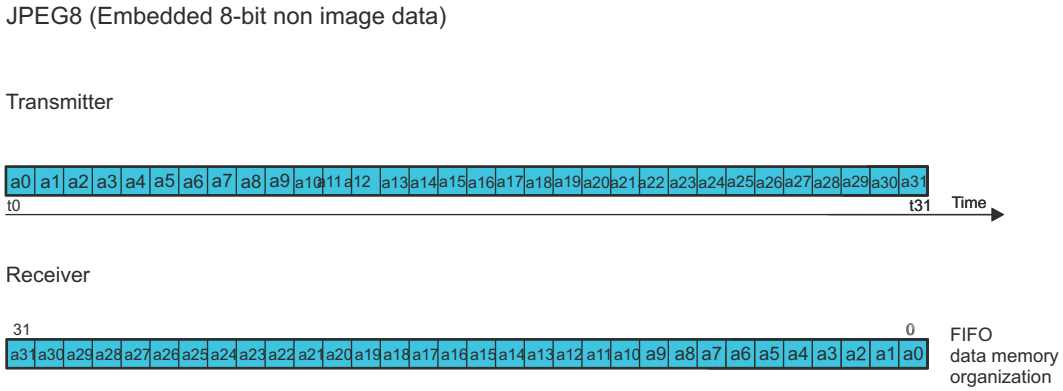


t0: VP_DATA = [a13 a12 a11 a10 a9 a8 a7 a6 a5 a4 a3 a2 a1 a0]
t1: VP_DATA = [b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0]
t2: VP_DATA = [c13 c12 c11 c10 c9 c8 c7 c6 c5 c4 c3 c2 c1 c0]
t3: VP_DATA = [d13 d12 d11 d10 d9 d8 d7 d6 d5 d4 d3 d2 d1 d0]

2.6.1.1.4.4 ISS CSI2 JPEG8 Operating Modes

The size of a compressed stream can be known in advance. Figure 63 shows the format for storing JPEG8 data.

Figure 63. ISS CSI2 JPEG8

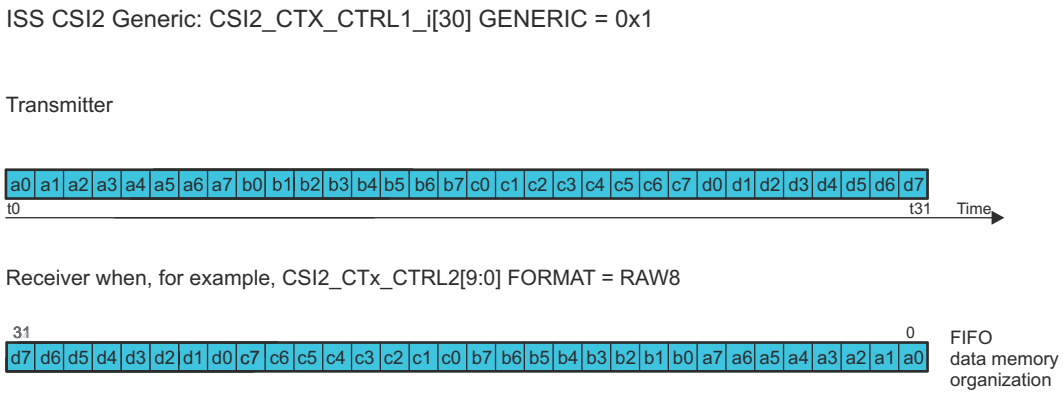


2.6.1.1.4.5 ISS CSI2 Generic Format

The CSI2 receiver supports a generic format to send data to memory and/or the video port. The generic mode is entered by setting the CSI2_CTX_CTRL1_i[30] GENERIC bit. The CSI2_CTX_CTRL2_i [9:0] FORMAT bit field defines how the data stream is decoded. When generic mode is enabled (GENERIC = 1), the MIPI data type code is ignored and data is decoded using the FORMAT bit. Whatever the MIPI data type code, it is ignored (the data stream is processed even if the FORMAT bit does not match the MIPI data type code.) When generic mode is not used (GENERIC = 0), the data stream is processed only when the MIPI data type code matches the FORMAT setting of the enabled context. If not matched, the data stream is not processed by the CSI2 engine. Only the virtual channel information is used to map a received data stream to a context. Software must ensure that a MIPI virtual channel used in generic mode is mapped only to a single context.

Figure 64 shows the ISS CSI2 generic format.

Figure 64. ISS CSI2 Generic Format



2.6.1.1.4.6 ISS CSI2 MIPI Format Supported Summary

Table 65 summarizes the CSI2 MIPI-supported formats and their output category. By setting the CSI2_CTX_CTRL2_i register format, the CSI2 outputs certain types of pixel packet data.

Table 65. ISS CSI2 MIPI Format Supported by the Protocol Engine

MIPI			CSI2 Protocol Engine Support
Category	**Abbreviation**	**Register Setting Format Description**	**Configuration Value for CSI2_CTX_CTRL2_i[9:0] FORMAT**
Sync short packet data types[1]	Short packet sync code	Mandatory FSC	0x000
	Short packet sync code	Mandatory FEC	0x001
	Short packet sync code	Optional LSC	0x002
	Short packet sync code	Optional LEC	0x003
			0x004
			0x005
			0x006
			0x007
Generic short packet data types[1]	Short packet	32-bit without ECC is stored in a register with code value 0x008.	0x008
	Short packet	32-bit without ECC is stored in a register with code value 0x009.	0x009
	Short packet	32-bit without ECC is stored in a register with code value 0x00A.	0x00A
	Short packet	32-bit without ECC is stored in a register with code value 0x00B.	0x00B
	Short packet	32-bit without ECC is stored in a register with code value 0x00C.	0x00C
	Short packet	32-bit without ECC is stored in a register with code value 0x00D.	0x00D
	Short packet	32-bit without ECC is stored in a register with code value 0x00E.	0x00E
	Short packet	32-bit without ECC is stored in a register with code value 0x00.	0x00F
Generic Long packet data types[2]	Null	Discarded	0x010
	Blanking data	Discarded	0x011
	Embedded 8-bit nonimage data (for example, JPEG)	0x12: Embedded 8-bit nonimage data (for example, JPEG)	0x012
		Send to memory when FORMAT = 0	0x013
		Send to memory when FORMAT = 0	0x014
		Send to memory when FORMAT = 0	0x015
		Send to memory when FORMAT = 0	0x016
		Send to memory when FORMAT = 0	0x017
YUV data	YUV4:2:0 8-bit	YUV4:2:0 8-bit	0x018
	YUV4:2:0 10-bit	YUV4:2:0 10-bit	0x019
	YUV4:2:0 8-bit legacy	YUV4:2:0 8-bit legacy	0x01A
	Reserved	Send to memory when FORMAT = 0	0x01B

(1) To understand ISS synchronization codes and short packets, see Section 2.6.3.3.3, *ISS CSI2 Short Packet*.
(2) To understand ISS synchronization codes and long packets, see Section 2.6.1.1.3.2, *ISS CSI2 Long Packet*.

Table 65. ISS CSI2 MIPI Format Supported by the Protocol Engine (continued)

MIPI			CSI2 Protocol Engine Support
Category	**Abbreviation**	**Register Setting Format Description**	**Configuration Value for CSI2_CTX_CTRL2_i[9:0] FORMAT**
	YUV4:2:0 8-bit + CSPS	YUV4:2:0 8 bit + CSPS	0x01C
	YUV4:2:0 10-bit + CSPS	YUV4:2:0 10 bit + CSPS	0x01D
	YUV4:2:2 8-bit	YUV4:2:2 8-bit	0x01E
		YUV4:2:2 8-bit + VP	0x09E
		YUV4:2:2 8-bit + VP16	0x0DE
	YUV4:2:2 10-bit	YUV4:2:2 10-bit	0x01F
RGB data	RGB444	RGB444 + EXP16	0x0A0
	RGB555	RGB555 + EXP16	0x0A1
	RGB565	RGB565	0x022
	RGB666	RGB666 + EXP32	0x0E3
		RGB666 + EXP32_24	0x033
	RGB888	RGB888	0x024
		RGB888 + EXP32	0x0E4
	Reserved	Send to memory when FORMAT = 0	0x025
	Reserved	Send to memory when FORMAT = 0	0x026
	Reserved	Send to memory when FORMAT = 0	0x027
RAW data	RAW6	RAW6	0x028
		RAW6 + EXP8	0x068
		RAW6 + DPCM10 + VP	0x0E8
		RAW6 + DPCM10 + EXP16	0x2A8
		RAW6 + DPCM12 + VP	0x368
		RAW6 + DPCM12 + EXP16	0x3A8
	RAW7	RAW7	0x029
		RAW7 + EXP8	0x069
		RAW7 + DPCM10 + EXP16	0x229
		RAW7 + DPCM10 + VP	0x329
		RAW7 + DPCM12 + EXP16	0x369
		RAW7 + DPCM12 + VP	0x3A9
	RAW8	RAW8	0x02A
		RAW8 + VP	0x12A
		RAW8 + DPCM10 + EXP16	0x2AA
		RAW8 + DPCM10 + VP	0x32A
		RAW8 + DPCM12 + EXP16	0x36A
		RAW8 + DPCM12 + VP	0x3AA
	RAW10	RAW10	0x02B
		RAW10 + EXP16	0x0AB
		RAW10 + VP	0x12F
	RAW12	RAW12	0x02C
		RAW12 + EXP16	0x0AC
		RAW12 + VP	0x12C
	RAW14	RAW14	0x02D
		RAW14 + EXP16	0x0AD
		RAW14 + VP	0x12D
	Reserved	Send to memory when FORMAT = 0	0x02E

Table 65. ISS CSI2 MIPI Format Supported by the Protocol Engine (continued)

MIPI			CSI2 Protocol Engine Support
Category	**Abbreviation**	**Register Setting Format Description**	**Configuration Value for CSI2_CTX_CTRL2_i[9:0] FORMAT**
	Reserved	Send to memory when FORMAT = 0	0x02F
User-defined byte-based data		USER_DEFINED_BYTE_DATA	0x040
		USER_DEFINED_BYTE_DATA + EXP8	0x080
		USER_DEFINED_BYTE_DATA + DPCM10 + EXP16	0x2C0
		USER_DEFINED_BYTE_DATA + DPCM10 + VP	0x340
		USER_DEFINED_BYTE_DATA + DPCM12 + EXP16	0x1C0
		USER_DEFINED_BYTE_DATA + DPCM12 + VP	0x140
		USER_DEFINED_BYTE_DATA	0x041
		USER_DEFINED_BYTE_DATA + EXP8	0x081
		USER_DEFINED_BYTE_DATA + DPCM10 + EXP16	0x2C1
		USER_DEFINED_BYTE_DATA + DPCM10 + VP	0x341
		USER_DEFINED_BYTE_DATA + DPCM12 + EXP16	0x1C1
		USER_DEFINED_BYTE_DATA + DPCM12 + VP	0x141
		USER_DEFINED_BYTE_DATA	0x042
		USER_DEFINED_BYTE_DATA + EXP8	0x082
		USER_DEFINED_BYTE_DATA + DPCM10 + EXP16	0x2C2
		USER_DEFINED_BYTE_DATA + DPCM10 + VP	0x342
		USER_DEFINED_BYTE_DATA + DPCM12 + EXP16	0x1C2
		USER_DEFINED_BYTE_DATA + DPCM12 + VP	0x142
		USER_DEFINED_BYTE_DATA	0x043
		USER_DEFINED_BYTE_DATA + EXP8	0x083
		USER_DEFINED_BYTE_DATA + DPCM10 + EXP16	0x2C3
		USER_DEFINED_BYTE_DATA + DPCM10 + VP	0x343
		USER_DEFINED_BYTE_DATA + DPCM12 + EXP16	0x1C3
		USER_DEFINED_BYTE_DATA + DPCM12 + VP	0x143
		USER_DEFINED_BYTE_DATA	0x044
		USER_DEFINED_BYTE_DATA + EXP8	0x084
		USER_DEFINED_BYTE_DATA + DPCM10 + EXP16	0x2C4
		USER_DEFINED_BYTE_DATA + DPCM10 + VP	0x344

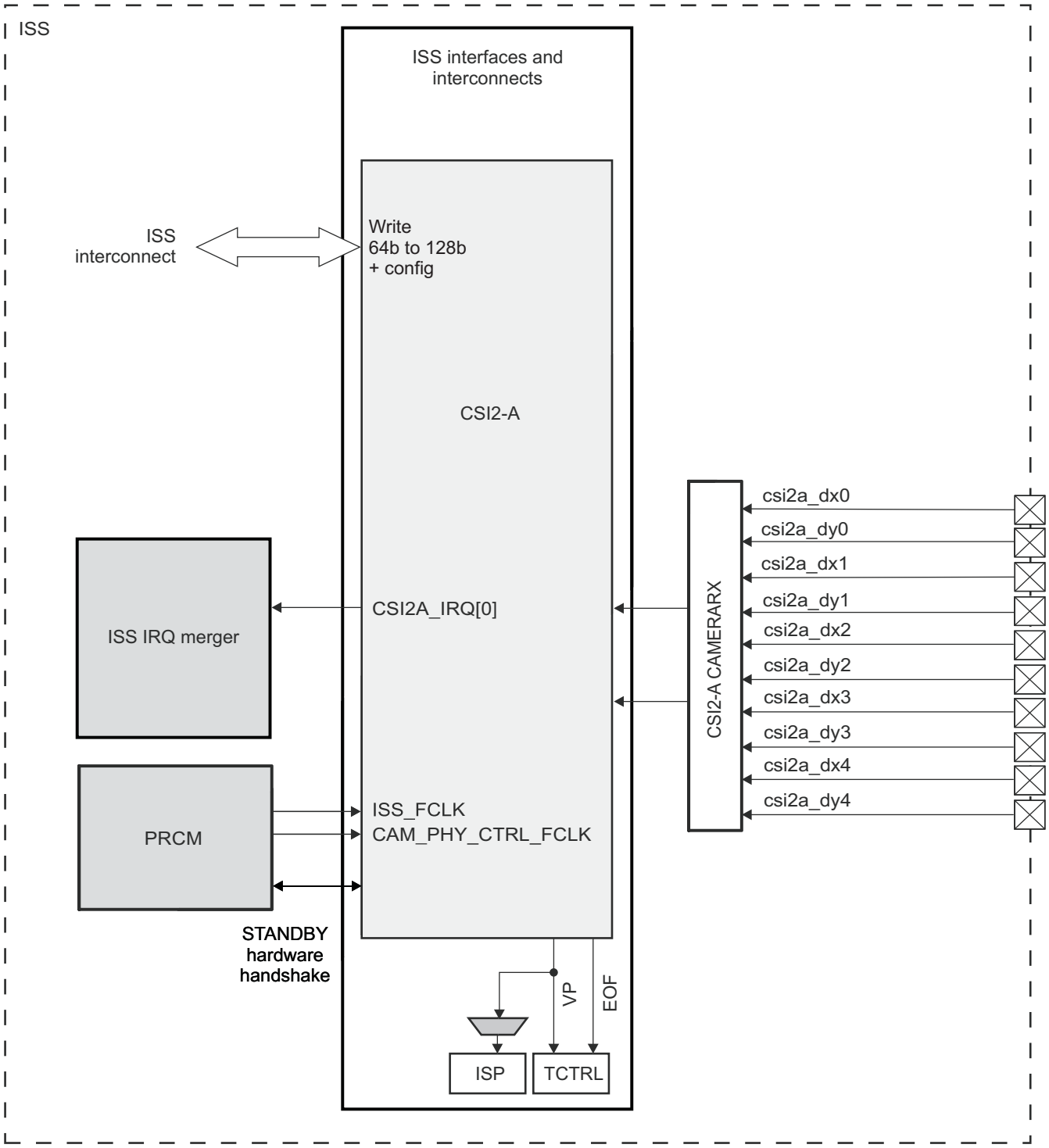
Table 65. ISS CSI2 MIPI Format Supported by the Protocol Engine (continued)

MIPI			CSI2 Protocol Engine Support
Category	**Abbreviation**	**Register Setting Format Description**	**Configuration Value for CSI2_CTX_CTRL2_i[9:0] FORMAT**
		USER_DEFINED_BYTE_DATA + DPCM12 + EXP16	0x1C4
		USER_DEFINED_BYTE_DATA + DPCM12 + VP	0x144
		USER_DEFINED_BYTE_DATA	0x045
		USER_DEFINED_BYTE_DATA + EXP8	0x085
		USER_DEFINED_BYTE_DATA + DPCM10 + EXP16	0x2C5
		USER_DEFINED_BYTE_DATA + DPCM10 + VP	0x345
		USER_DEFINED_BYTE_DATA + DPCM12 + EXP16	0x1C5
		USER_DEFINED_BYTE_DATA + DPCM12 + VP	0x145
		USER_DEFINED_BYTE_DATA	0x046
		USER_DEFINED_BYTE_DATA + EXP8	0x086
		USER_DEFINED_BYTE_DATA + DPCM10 + EXP16	0x2C6
		USER_DEFINED_BYTE_DATA + DPCM10 + VP	0x346
		USER_DEFINED_BYTE_DATA + DPCM12 + EXP16	0x1C6
		USER_DEFINED_BYTE_DATA + DPCM12 + VP	0x146
		USER_DEFINED_BYTE_DATA	0x047
		USER_DEFINED_BYTE_DATA + EXP8	0x087
		USER_DEFINED_BYTE_DATA + DPCM10 + EXP16	0x2C7
		USER_DEFINED_BYTE_DATA + DPCM10 + VP	0x347
		USER_DEFINED_BYTE_DATA + DPCM12 + EXP16	0x1C7
		USER_DEFINED_BYTE_DATA + DPCM12 + VP	0x147
Reserved		Send to memory when FORMAT = 0	0x038
		Send to memory when FORMAT = 0	0x039
		Send to memory when FORMAT = 0	0x03A
		Send to memory when FORMAT = 0	0x03B
		Send to memory when FORMAT = 0	0x03C
		Send to memory when FORMAT = 0	0x03D
		Send to memory when FORMAT = 0	0x03E
		Send to memory when FORMAT = 0	0x03F

2.6.2 ISS CSI2 Integration

Figure 65 is an overview of the integration of the CSI2-A interface in the device. The figure is the top-level block diagram of the CSI2-A receiver. The receiver receives the serial data coming from a CSI2 compatible image sensor, converts it to parallel data, extracts the logical channels, detects and extracts the synchronization codes, reformats the data, and outputs it through the video port or the ISS interconnect interface.

Figure 65. ISS CSI2-A Integration



The CSI2-A receiver can send data directly to system memory using the master port or send it to the camera ISP using the video port.

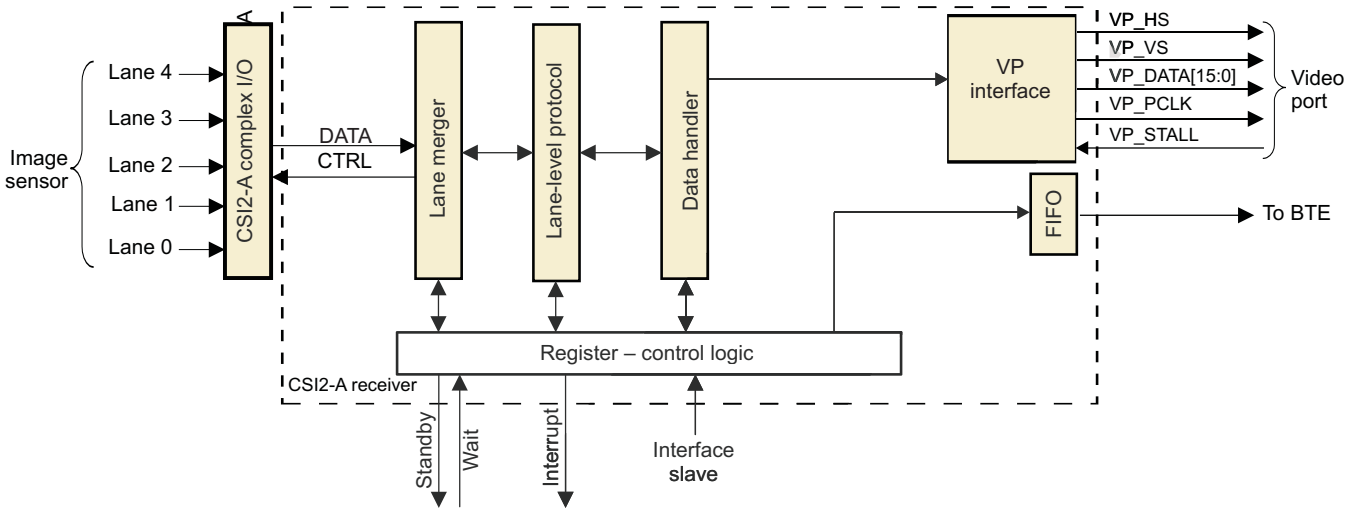
For power domain, clocks, reset, and hardware requests, see Section 1.2.5, *ISS Power Management*.

2.6.3 ISS CSI2 Functional Description

2.6.3.1 ISS CSI2 Overview

Figure 66 is the CSI2-A receiver block diagram (it assumes there are four CSI2 image sensor data lines). The CSI2 receiver receives the byte data coming from a CSI2 D-PHY receiver (up to four data pairs), converts it to byte stream, detects and corrects errors, extracts the virtual channel ID, detects and extracts the synchronization codes, reformats the data, and outputs it through the video port or the ISS interconnect interface. Data communication between CSI2 and ISP is done through the video port.

Figure 66. ISS CSI2-A Receiver Block Diagram



2.6.3.2 ISS CSI2 Features

The CSI2 receiver is a master on the L3 interconnect for storing data in memory and a slave on the level 4 (L4) interconnect for register access.

The main features of the CSI2 receiver are:

- Transfer pixels and data received by the CSI2 PHY to the system memory or video processor
- Unidirectional data link
- Supports up to four data-configurable links in addition to the clock signaling (minimum of one data link and maximum of four depending on the speed)
- Data merger for two, three, or four data lane configurations
- Error detection and correction by the protocol engine
- DMA engine integrated with dedicated FIFO
- 1D and 2D addressing modes
- Up to eight contexts to support eight dedicated configurations of virtual channel ID and data types
- Ping-pong mechanism for double-buffering
- JPEG support for unknown length transfer (no extraction of the thumbnail)
- Supports all primary and secondary MIPI-defined formats (RGB, RAW, and YUV)
- Storage in progressive mode for interlaced stream (using line numbering)
- Conversion to the RGB formats
- Decompressions of the RAW formats

- RAW frame transcoding, including DPCM and A-Law compression
- Fully configurable interface of the complex PHY I/O: position of the clock and data and order of ± differential signals for each pair

2.6.3.3 ISS CSI2 Functional Description

2.6.3.3.1 ISS CSI2 Physical Layer Lane Configuration

The CSI2 serial interface is a unidirectional differential serial interface with data/clock for the physical layer.

The maximum CSI2 receiver data transfer capacity is 1000 Mbps per data lane.

Data-clock signaling consists of two to five differential signal pairs: from one to four data lanes and one clock lane:

- The data signal carries the bit-serial data. The CSI2 transmitter in the image sensor sends the data in-quadrature with the dual-data rate (DDR) clock in HS mode; otherwise, the clock is extracted from the received data in LS mode. Data is transmitted byte-wise, LSB first. The CSI2 complex I/O receives the data and sends the byte stream to the CSI2 receiver.
- The clock signal carries the DDR clock signal.

Each physical lane can be a data or clock lane with a restriction to the fourth line, which can only be data (see Section 2.3.1, *ISS CSI2 PHY Overview*). The clock/data lane must be configured before transmission to indicate the byte order, while merging the received bytes into a byte stream shows the reachable speed per data lane function of data lane numbers.

Lanes are configured through the CSI2_COMPLEXIO_CFG registers for CSI2-A and PHY. The CSI2_COMPLEXIO_CFG[2:0] CLOCK_POSITION bit field and the CSI2_COMPLEXIO_CFG[3] CLOCK_POL bit configure which lane transmits the clock and define its polarity. DATAl_POSITION and DATAl_POL configure the data lanes and their polarity, where l is the number of the data lane (l = 1 to 4). When the DATAl_POSITION field is set to 0, data lane l is not used.

CAUTION

Lane 4 (position 5) supports only data. The CLOCK_POSITION must not be set at position 5.

2.6.3.3.2 ISS CSI2 ECC and Checksum Generation

The CSI2 receiver includes an ECC in the packet header and a checksum in the packet footer for long-packet transmission. These two fields can be used to detect and/or correct errors in the received packet.

2.6.3.3.2.1 ISS CSI2 ECC

To detect and correct transmission errors of the header of short and long packets, an 8-bit ECC is included in the header of packets (short and long packet).

The ECC concerns all the fields for a short packet (data ID and short-packet data field) and the packet header for a long packet (data ID and word count). The ECC can only correct one error. Additional errors cannot be repaired, but they are flagged.

The CSI2 receiver ECC is compared against the CSI2 transmitter ECC embedded in the bitstream. If the ECC does not match, an interrupt is triggered to the host central processing unit (CPU).

For long and short packets, the correction is always done if there is only one error per packet header.

An ECC error with or without correction can be reported at two levels, depending on the type of packet. Table 66 describes the field in which events are logged. Logging cannot be disabled, but users can set the corresponding bit in the CSI2_IRQENABLE and CSI2_CTX_IRQENABLE registers to prevent event generation at a higher level.

The ECC check can be disabled (short and long packet) by setting the CSI2_CTRL[2] ECC_EN bit to 0. Setting the bit to 1 enables the ECC check.

Table 66. ISS CSI2 ECC Event Logging

	Short Packet	Long Packet
With correction	Global CSI2_IRQSTATUS[12] ECC_CORRECTION_IRQ	Context CSI2_CTX_IRQSTATUSi[8] ECC_CORRECTION_IRQ
Without correction	Global CSI2_IRQSTATUS[11] ECC_NO_CORRECTION_IRQ	Global CSI2_IRQSTATUS[11] ECC_NO_CORRECTION_IRQ

2.6.3.3.2.2 ISS CSI2 Checksum

To detect errors in transmission of the payload of long packets, a 16-bit CRC checksum is computed on the payload of the long packets in the transmitter. This CRC is stored in the packet footer. A CRC is also computed in the CSI2 receiver. If the checksums do not match, an interrupt is triggered to the host CPU.

CRC errors are logged in the CS_IRQ field of the corresponding context register, CSI2_CTX_IRQSTATUS_i. Logging cannot be disabled, but users can set the corresponding bit in the CSI2_CTX_IRQSTATUS_i register to prevent event generation at a higher level.

The CRC can be disabled for a specific context by setting the CSI2_CTRL[5] CS_EN bit to 0. Setting the bit to 1 enables the CRC.

2.6.3.3.3 ISS CSI2 Short Packet

There are two types of short packets in the CSI2 receiver:

- Synchronization short packet: Used by the protocol engine to synchronize frame and line (data ID from 0x0 to 0x7)
- Generic short packet: User-dependent; not treated by the protocol engine (data ID from 0x8 to 0xF)

When a generic short packet is received by the CSI2 receiver, the ECC check is performed if it is enabled. Then, the short packet is written in the CSI2_SHORT_PACKET[23:0] SHORT_PACKET bit field. The ECC field is deleted from the short packet. Figure 67 shows the SHORT_PACKET bit field format.

Figure 67. ISS CSI2 SHORT_PACKET Bit Field Format

When a short packet is stored, an event is logged in the CSI2_IRQSTATUS [13] SHORT_PACKET_IRQ bit. Logging cannot be disabled, but users can set the corresponding bit in the CSI2_IRQENABLE register to prevent event generation at a higher level.

The application reads the CSI2_SHORT_PACKET register before the next short packet with a code from 0x8 to 0xF. There is a single register for capturing the generic short packet, because no data type in it is associated with context.

2.6.3.3.4 ISS CSI2 Virtual Channel and Context

The CSI2 protocol layer transports virtual channels. The virtual channels separate different data flows interleaved in the same data stream. Each virtual channel is identified by a unique channel identification number in the packet header. This channel identification number is encoded in the 2-bit code.

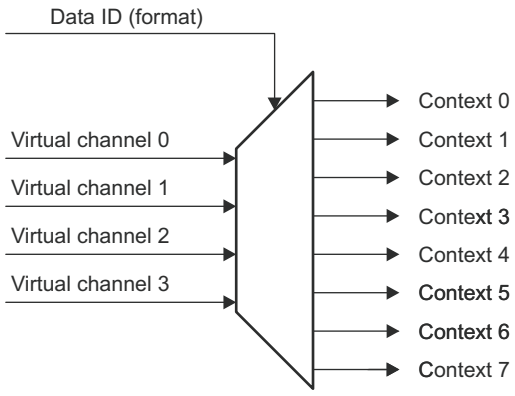
The CSI2 receiver monitors the channel identifier number and demultiplexes the interleaved data streams. The CSI2 receiver supports up to four concurrent virtual channels.

The CSI2 receiver supports eight contexts with their events to control the four possible virtual channels and the different data transmitted through them. A context is linked to a specific data type transported by a given virtual channel. The following bit fields permit configuration of a context:

- CSI2_CTX_CTRL2_i[12:11] VIRTUAL_ID: Configures the virtual ID linked to the current context
- CSI2_CTX_CTRL2_i[9:0] FORMAT: Configures the data format linked to the current context

Figure 68 shows the relationships between virtual channels and contexts.

Figure 68. ISS CSI2 Virtual Channel to Context



Each context consists of eight registers: six registers to control the corresponding context and two to log and enable events from the context. All registers in a context can be modified at any time; however, modifications apply only from the start of the following frame.

A context can be enabled independently by setting the CSI2_CTX_CTRL1_i[0] CTX_EN bit to 1; setting this bit to 0 disables the corresponding context.

When acquiring frames on a context, users can write the number of frames to capture in the CSI2_CTX_CTRL1_i[15:8] COUNT bit field. Acceptable values are 0 to 255; 0 stands for infinite capture (no count). After each frame is acquired, the count value is decremented by 1. When the count value reaches 0, the CSI2_CTX_IRQSTATUS_i[6] FRAME_NUMBER_IRQ event is set and the CTX_EN bit is set to 0. To write a value in the COUNT bit field, the CSI2_CTX_CTRL1_i[4] COUNT_UNLOCK bit must be set to 1. If the value of the COUNT_UNLOCK bit is 0, a write in the COUNT bit field has no effect.

The CSI2_CTX_CTRL3_i [15:0] LINE_NUMBER bit field configures the generation of the CSI2_CTX_IRQSTATUS_i [7] LINE_NUMBER_IRQ event. The CSI2_CTX_CTRL1_i [1] LINE_MODULO bit configures how the LINE_NUMBER event is generated:

- 0: The event is generated one time by frame.
- 1: The event is generated modulo LINE_NUMBER (the event can be generated more than once in a frame).

During a frame capture, the CSI2_CTX_CTRL2_i[31:16] FRAME_NUMBER bit field shows the number that identifies the frame received.

2.6.3.3.5 ISS CSI2 DMA Engine

The CSI2 receiver integrates its own DMA engine with dedicated FIFO.

Global DMA configuration is common to the eight channels and is defined in the CSI2_CTRL register. Configuration of the ping-pong address and the offset between lines is specific for a given context; therefore, each context has its own DMA configuration registers.

The DMA engine supports:

- 1D addressing mode (no address line offset, CSI2_CTX_DAT_OFST_i = 0)
- 2D addressing mode (address line offset different than 0, CSI2_CTX_DAT_OFST_i = 0)

The burst size is defined in the CSI2_CTRL [6:5] BURST_SIZE bit field and the CSI2_CTRL[16] BURST_SIZE_EXPAND bit. The DMA uses the burst size or smaller sizes down to single open-core protocol (OCP) writes depending on the alignment at the end of lines. The DMA engine can handle burst requests. When the burst requests can be used, as soon as one burst of data is present in the FIFO, the DMA engine initiates a burst write. The burst size is defined in the CSI2_CTRL [6:5] BURST_SIZE bit field and the CSI2_CTRL [16] BURST_SIZE_EXPAND bit.

NOTE: Unless there are specific requirements, CSI2 (also applies to all other ISS initiators) must be configured to use only a burst size of 128 bytes and nonposted writes.

When single requests must be used, as soon as one element (the size depends on the data type and the post-processing: DPCM, EXT, etc.) is present in the FIFO, the DMA engine initiates a single write.

Interleave mode is dedicated by the CSI2 receiver only when the line numbers are received (short packets). The line number is used to calculate the start address of the line.

The DMA starts to write in memory using the CSI2_CTX_DAT_PING_ADDR_i[31:5] ADDR bit field for the first frame to be transferred, and then uses the CSI2_CTX_DAT_PONG_ADDR_i [31:5] ADDR bit field and the ping address alternately. Thus, the first frame uses the ping address, the second frame uses the pong address, the third frame uses the ping address, and so on.

The CSI2_CTX_CTRL_i [3] PING_PONG status bit indicates whether the ping address (CSI2_CTX_DAT_PING_ADDR_i) or the pong address (CSI2_CTX_DAT_PONG_ADDR_i) was used to store the pixel data of the last frame. After reset or after a 0-to-1 edge transition in the CSI2_CTRL[0] IF_EN bit, the pixel data is written in the ping buffer and the CSI2_CTX_CTRL1_i[3] PING_PONG bit = PONG. When the number of FECs received equals the value programmed in the CSI2_CTX_CTRL1_i[23:16] FEC_NUMBER bit field, the pixel data are written in the pong buffer and CSI2_CTX_CTRL1_i [3] PING_PONG = PING. CSI2_CTX_CTRL1_i[3] PING_PONG toggles after the CSI2_CTX_CTRL1_i[23:16] FEC_NUMBER FEC sync code with the virtual channel ID defined is received in the CSI2_CTX_CTRL2_i [12:11] VIRTUAL_ID bit field.

The CSI2_CTX_CTRL1_i[23:16] FEC_NUMBER bit field must be set as follows:

- In progressive mode, set to 1.
- In interlaced mode, set to the number of interlaced frames to recreate a progressive image in the PING_PONG buffer.

2.6.3.3.5.1 ISS CSI2 Progressive Frame to Progressive Storage

After each line, a new start line address is computed, depending on the value of the CSI2_CTX_DAT_OFST_i[31:5] OFST bit field:

- If OFST = 0, the new line starts immediately after the last pixel (data are written contiguously in memory).
- Otherwise, the value of OFST sets the offset between the first pixel of the previous line and the first pixel of the current line in memory.

For the ping frame:

```
@Line0 = CSI2_CTX_DAT_PING_ADDR_i@Line1 = @Line0 + CSI2_CTX_DAT_OFST_i@Line2 = @Line1 +
CSI2_CTX_DAT_OFST_i
```

For the pong frame:

```
@Line0 = CSI2_CTX_DAT_PONG_ADDR_i@Line1 = @Line0 + CSI2_CTX_DAT_OFST_i@Line2 = @Line1 +
CSI2_CTX_DAT_OFST_i
```

2.6.3.3.5.2 ISS CSI2 Interlaced Frame to Progressive Storage

The mode is functional only when the line numbers are transmitted. It is automatically enabled without setting.

For the ping frame:

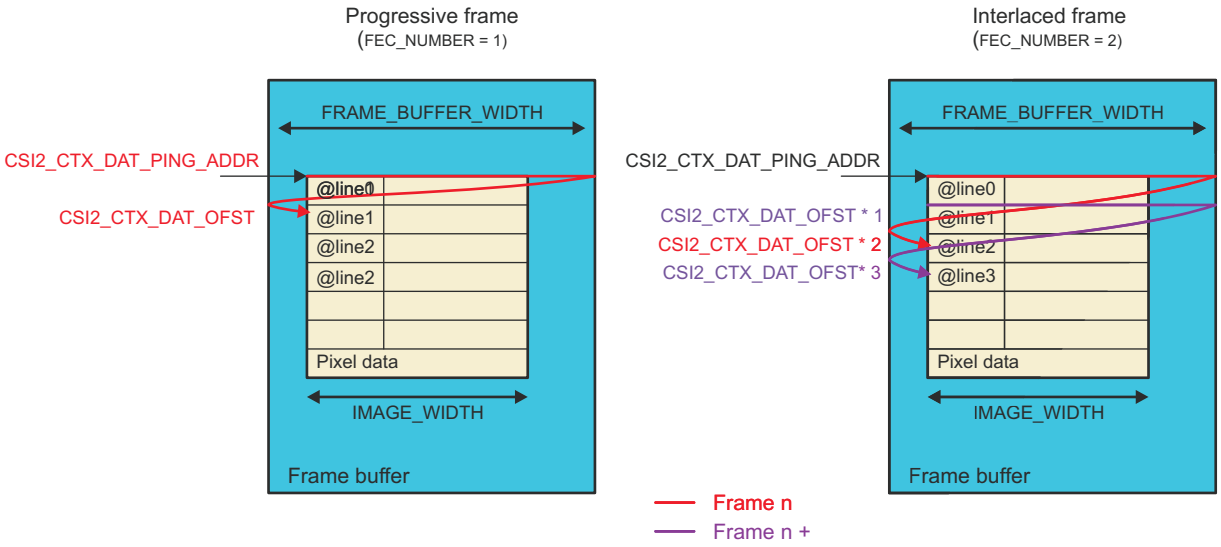
```
@LineX = CSI2_CTX_DAT_PING_ADDR_i + CSI2_CTX_DAT_OFST_i * Line_Number
```

For the pong frame:

```
@LineX = CSI2_CTX_DAT_PONG_ADDR_i + CSI2_CTX_DAT_OFST_i * Line_Number
```

Figure 69 shows how data are stored in memory regarding the DMA configuration.

Figure 69. ISS CSI2 Pixel Data Destination Setting in Progressive and Interlaced Mode



The burst size is defined in the CSI2_CTRL[6:5] BURST_SIZE bit field for bursts up to 16 × 64 bits or the CSI2_CTRL[16] BURST_SIZE_EXPAND bit for 16 × 128-bit bursts. It can be changed only while the CSI2_CTRL[0] IF_EN bit is reset to 0. The recommended value is the CSI2_CTRL[16] BURST_SIZE_EXPAND bit set to 1, which defines a burst of 16 × 64 bits (the maximum value); otherwise, by default it is set to 8 × 64 bits. When the BURST_SIZE_EXPAND bit is set, the BURST_SIZE setting has no effect. The DMA uses nonposted writes by default. The CSI2_CTRL[13] NON_POSTED_WRITE bit must be set to 1 to match DMA default configuration. It can be changed only while the CSI2_CTRL[0] IF_EN bit is reset to 0.

2.6.3.3.6 ISS CSI2 Transcoding

Image transcoding is used mainly to reduce memory footprint and bandwidth when:

- The sensor does not support DPCM compression. In fact, A-Law and DPCM compressed pixels occupy only 6, 7, or 8 BPP of storage.
- Digital zoom is used
  - Data that is not going to be used by further processing does not need to the stored in system memory.
  - Pixels cannot be accessed from random locations in a DPCM-compressed frame. Transcoding avoids memory-to-memory processing of unused pixels.

Figure 70 shows the logical representation of the image transcoding operation.

- Data is extracted from the CSI2 stream by the protocol engine.
- It is DPCM decompressed if necessary. That is the case when the received stream is DPCM-compressed and transcoding has been enabled using the CSI2_CTRL1_i[27:24] TRANSCODE bit field.
- Data sent to the video port cannot be compressed: it is intended to be processed by the ISS ISP. Data sent to system memory can be optionally compressed.
- Internal data are aligned on MSB when they enter the cropping stage. For example:
  - 4 LSBs are 0s when RAW10 data are handled.
  - 2 LSBs are 0s when RAW12 data are handled.

Figure 70. ISS CSI2 Frame Processing

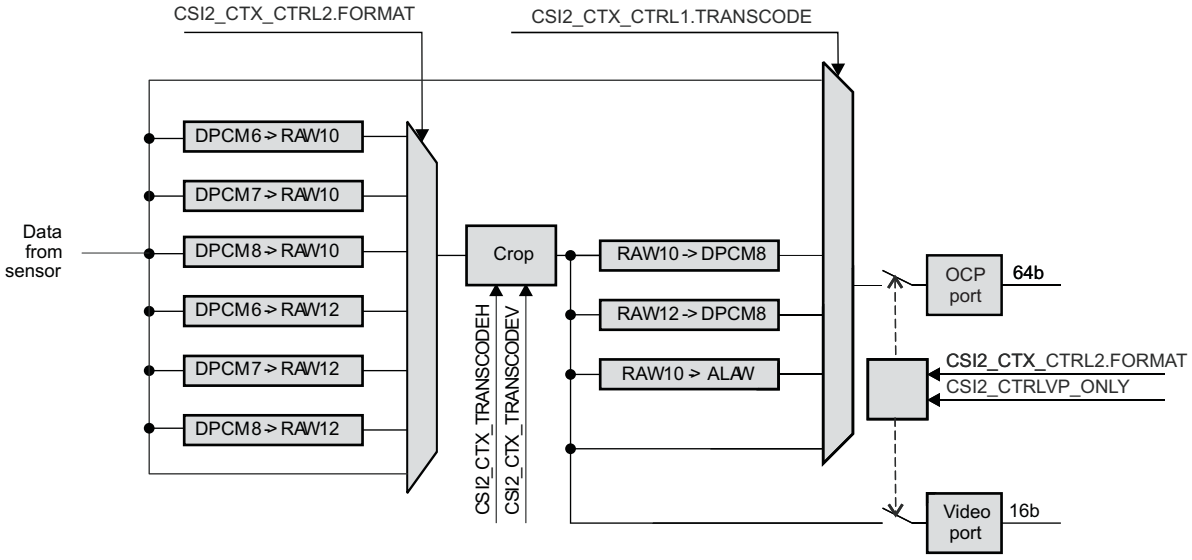


Table 67 shows the input format provided to the cropping engine for a given pixel format provided by the sensor. Formats not listed in the table are not supported for transcoding. The FORMAT and Corresponding Setting Value column corresponds to the value set in the CSI2_CTX_CTRL2_i[9:0] FORMAT register.

Table 67. ISS CSI2 Supported Transcoding Input Formats

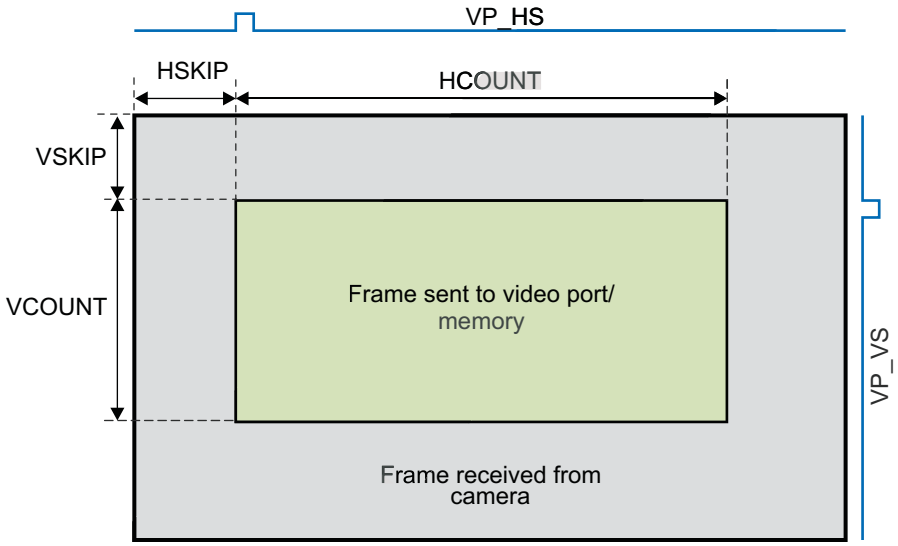
CSI2_CTX_CTRL2_i[9:0] FORMAT and Corresponding Setting Value		Cropping Engine Input Format	DPCM Decomposition Enabled	Video Port Enabled
0x028	RAW6	RAW6		
0x058	RAW6 + EXP8			
0x029	RAW7	RAW7		
0x059	RAW7 + EXP8			
0x02A	RAW8	RAW8		
0x12A	RAW8 + VP			Yes
0x02B	RAW10	RAW10		
0x0AB	RAW10 + EXP16			
0x0E8	RAW6 + DPCM10 + VP		Yes	Yes
0x12F	RAW10 + VP			Yes
0x229	RAW7 + DPCM10 + EXP16		Yes	
0x2A8	RAW6 + DPCM10 + EXP16		Yes	
0x2AA	RAW8 + DPCM10 + EXP16		Yes	
0x329	RAW7 + DPCM10 + VP		Yes	Yes
0x32A	RAW8 + DPCM10 + VP		Yes	Yes
0x2Cn	USER_DEFINED_BYTE_DATA + DPCM10 + EXP16		Yes	
0x34n	USER_DEFINED_BYTE_DATA + DPCM10 + VP		Yes	Yes
0x02C	RAW12	RAW12		
0x0AC	RAW12 + EXP16			
0x12C	RAW12 + VP			Yes
0x35A	RAW8 DPCM12 + EXP16		Yes	
0x3AA	RAW8 DPCM12 + VP		Yes	Yes
0x1Cn	USER_DEFINED_BYTE_DATA + DPCM12 + EXP16		Yes	
0x14n	USER_DEFINED_BYTE_DATA + DPCM12 + VP		Yes	Yes

Table 67. ISS CSI2 Supported Transcoding Input Formats (continued)

CSI2_CTX_CTRL2_i[9:0] FORMAT and Corresponding Setting Value		Cropping Engine Input Format	DPCM Decomposition Enabled	Video Port Enabled
0x3A8	RAW6 + DPCM12 + EXP16		Yes	
0x358	RAW6 + DPCM12 + VP		Yes	Yes
0x359	RAW7 + DPCM12 + EXP16		Yes	
0x3A9	RAW7 + DPCM12 + VP		Yes	Yes
0x02D	RAW14	RAW14		
0x0AD	RAW14 + EXP16			
0x12D	RAW14 + VP			Yes

Image cropping parameters are controlled by software. Figure 71 shows the cropping operation.

Figure 71. ISS CSI2 Frame Cropping



CAUTION

Hardware does not check for validity of the settings. The following rules must be respected:

- CSI2_CTX_TRANSCODEH_i[12:0] HSKIP + CSI2_CTX_TRANSCODEH_i[28:16] HCOUNT = image width
- CSI2_CTX_TRANSCODEV_i[12:0] VSKIP + CSI2_CTX_TRANSCODEV_i[28:16] VCOUNT = image height

Furthermore, the CSI2_CTX_TRANSCODEH_i[28:16] HCOUNT bit field must comply with the following alignment constraints; otherwise, undefined behavior occurs. Table 68 shows the transcode alignment constraints

Table 68. ISS CSI2 Transcode Alignment Constraints

CSI2_CTX_CTRLi[27:24] TRANSCODE Value	Transcode	HCOUNT Must Be Multiple of
0x0	Disabled	1
0x1	DPCM10 RAW8	1
0x2	DPCM12 RAW8	1

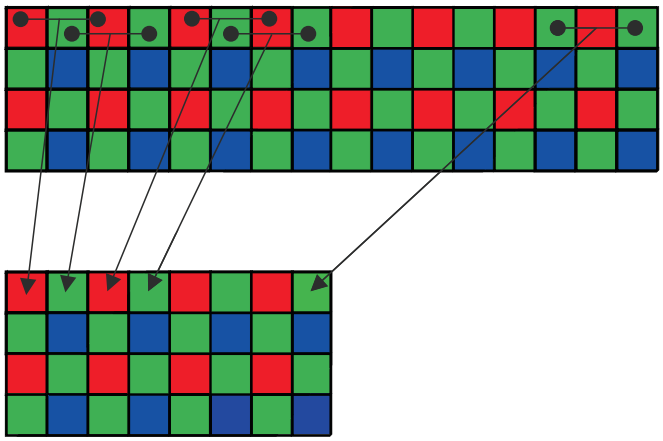
Table 68. ISS CSI2 Transcode Alignment Constraints (continued)

CSI2_CTX_CTRLi[27:24] TRANSCODE Value	Transcode	HCOUNT Must Be Multiple of
0x3	ALAW10 RAW8	1
0x4	RAW8	1
0x5	RAW10 + EXP16	1
0x6	RAW10	4
0x7	RAW12 + EXP16	1
0x8	RAW12	2
0x9	RAW10 + EXP16	4

The CSI2_CTX_CTRL1_i[28] HSCALE configuration register enables horizontal downscaling of RAW data. It reduces the horizontal size and pixel clock by a factor of 2. The scaler uses a 2-tap horizontal filter operating on samples of the same color plane. The coefficients are: [1/2 ; 0 ; 1/2]

Figure 72 shows the scaler operation.

Figure 72. ISS CSI2 Horizontal Scaler



The scaler can send data to the video port or the interface port. When data goes to the video port, no additional alignment constraints apply. But when data goes to the interface port, HCOUNT/2 must comply with the constraints from Table 68 (for example, for RAW10, HCOUNT must be a multiple of 8).

Table 69 lists possible combinations of input and output formats supported by the transcoding engine. The Transcode column corresponds to the CSI2_CTX_CTRL1_i[27:24] TRANSCODE bit field of a context.

Table 69. ISS CSI2-Supported Transcoding Output Formats

Cropping Engine Output	Transcode		Supported	Cropping Engine Output	Transcode		Supported
RAW6	0	Disabled	Yes	RAW10	0	Disabled	Yes
	1	DPCM10 RAW8			1	DPCM10 RAW8	Yes
	2	DPCM12 RAW8			2	DPCM12 RAW8	
	3	ALAW10 RAW8			3	ALAW10 RAW8	Yes
	4	RAW8			4	RAW8	
	5	RAW10 + EXP16			5	RAW10 + EXP16	Yes
	6	RAW10			6	RAW10	Yes
	7	RAW12 + EXP16			7	RAW12 + EXP16	
	8	RAW12			8	RAW12	
	9	RAW14			9	RAW14	

Table 69. ISS CSI2-Supported Transcoding Output Formats (continued)

Cropping Engine Output	Transcode		Supported	Cropping Engine Output	Transcode		Supported
RAW7	0	Disabled	Yes	RAW12	0	Disabled	Yes
	1	DPCM10 RAW8			1	DPCM10 RAW8	
	2	DPCM12 RAW8			2	DPCM12 RAW8	Yes
	3	ALAW10 RAW8			3	ALAW10 RAW8	
	4	RAW8			4	RAW8	
	5	RAW10 + EXP16			5	RAW10 + EXP16	
	6	RAW10			6	RAW10	
	7	RAW12 + EXP16			7	RAW12 + EXP16	Yes
	8	RAW12			8	RAW12	Yes
	9	RAW14			9	RAW14	
RAW8	0	Disabled	Yes	RAW14	0	Disabled	Yes
	1	DPCM10 RAW8			1	DPCM10 RAW8	
	2	DPCM12 RAW8			2	DPCM12 RAW8	
	3	ALAW10 RAW8			3	ALAW10 RAW8	
	4	RAW8	Yes		4	RAW8	
	5	RAW10 + EXP16			5	RAW10 + EXP16	
	6	RAW10			6	RAW10	
	7	RAW12 + EXP16			7	RAW12 + EXP16	
	8	RAW12			8	RAW12	
	9	RAW14			9	RAW14	Yes

RAW pixels are packed into 64-bit words sent to the OCP master port, as defined in:

- Section 2.6.1.1.4.3.3, *ISS CSI2 RAW8*
- Section 2.6.1.1.4.3.4, *ISS CSI2 RAW10*
- Section 2.6.1.1.4.3.5, *ISS CSI2 RAW12*
- Section 2.6.1.1.4.3.6, *ISS CSI2 RAW14*

For RAW10 and RAW12, software can choose among packed and nonpacked storage. A-Law and DPCM-compressed pixels are stored as RAW8 data: each RAW8 container holds a compressed data point.

Similarly, RAW data is sent over the video port, as described in:

- Section 2.6.1.1.4.3.3, *ISS CSI2 RAW8*
- Section 2.6.1.1.4.3.4, *ISS CSI2 RAW10*
- Section 2.6.1.1.4.3.5, *ISS CSI2 RAW12*
- Section 2.6.1.1.4.3.6, *ISS CSI2 RAW14*

Enabling of the OCP/video port is controlled by the CSI2_CTX_CTRL2_i[9:0] FORMAT bit field and the CSI2_CTRL[11] VP_ONLY_EN and CSI2_CTX_CTRL1_i[2] VPFORCE bits.

To enable transcoding, software configures the context normally and also configures the framing using the CSI2_CTX_TRANSCODEV_i and CSI2_CTX_TRANSCODEH_i registers. Software defines the after transcoding with the CSI2_CTX_CTRL1_i[27:24] TRANSCODE bit field.

2.6.3.3.7 ISS CSI2 EndOfFrame and EndOfLine (EOF and EOL) Pulses

The CSI2 receiver generates two signals to qualify the last pixel of a frame and the last pixel of a line to the TCTRL. It is active during or after the adequate interface bridge transaction and becomes inactive before the first transaction of the next line. Software can enable/disable generation of those signals for each context using the CSI2_CTX_CTRL1_i[7] EOF_EN and CSI2_CTX_CTRL1_i [6] EOL_EN bits. When data is sent to both OCP and video ports, the EOL/EOF timing defined for the OCP port is used.

2.6.3.3.8 ISS CSI2 Data Decompression

The data compression technique used is DPCM and PCM.

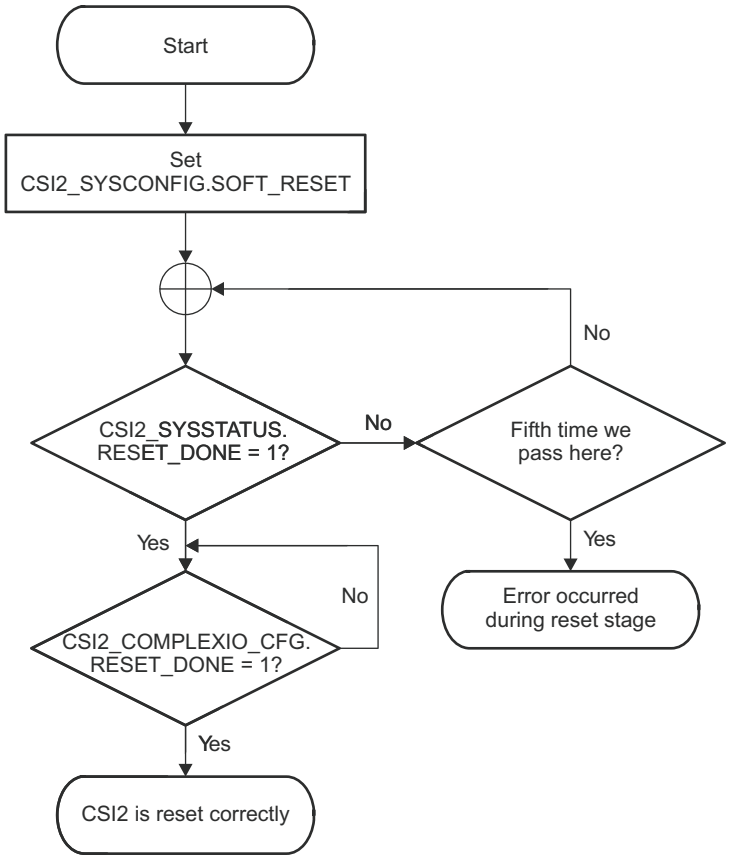
To select the DPCM decompression predictor for the CSI2 Interface, set the CSI2_CTX_CTRL2_i[10] DPCM_PRED bit to 1 for simple predictor or to 0 for advanced predictor.

2.6.4 ISS CSI2 Programming Model

2.6.4.1 ISS CSI2 Reset Management

The CSI2 receiver accepts a general software reset, propagated throughout the hierarchy. This reset can be done to initialize the CSI2 receiver and the complex I/O and has the same effect as a hardware reset. Figure 73 shows how to reset CSI2 globally.

Figure 73. ISS CSI2 Receiver Global Reset Flow Chart



NOTE: Before setting the software reset bit to 1 in the CSI2_SYSCONFIG register, the user must have access to a CSI2 receiver register.

NOTE: The CSI2_COMPLEXIO_CFG[29] RESET_DONE bit is set to 1 only after the initialization of the CSI2 receiver, CSI2 complex I/O, and external camera completes.

2.6.4.2 ISS CSI2 Enable Video/Picture Acquisition

Before using the receiver, a CSIPHY initialization in CSI2 mode must be made for CSI2-A CAMERARX, which is associated with the CSI2 receiver. See Section 2.3.2.2, *ISS CSI2 PHY and Link Initialization Sequence.* To start a video/picture acquisition, perform the steps listed in Table 70.

Table 70. ISS CSI2 Global Initialization

Step	Register/Bit Field/Programming Model	Value
Reset the CSI2 receiver.	See Section 2.6.4.1, *Reset Management.*	
Configure the module power management. The module tries to enter smart-standby mode during the vertical blanking period. The CSI2_SYSCONFIG[0] AUTO_IDLE bit keeps its reset value; by default, an automatic port clock gating strategy is applied based on port interface activity.	CSI2_SYSCONFIG[13:12] MSTANDBY_MODE	0x2
Configure the interrupt generation as required. To enable context and/or complex I/O event reporting, enable the corresponding bit field in the CSI2_IRQENABLE register. If the enable bit is at 0, logging is still effective if an event occurs, but is not reported to a higher level.	CSI2_IRQSTATUS and CSI2_IRQENABLE	
Configure the complex I/O interrupt generation as required. If the enable bit is at 0, logging is still effective if an event occurs, but is not reported to a higher level.	CSI2_COMPLEXIO_IRQSTATUS and CSI2_COMPLEXIO_IRQENABLE)	
Start complex I/O: Set the CSI2_COMPLEXIO_CFG[28:27] PWR_CMD bit field to 0x1 to pass the complex I/O to the ON state, and then check that the state status reaches the ON state (CSI2_COMPLEXIO_CFG[26:25] PWR_STATUS = 0x1) (for complex I/O A).	CSI2_COMPLEXIO_CFG[28:27] PWR_CMD	0x1
Configure the complex I/O: • The complex I/O is fully functional with CSI2_COMPLEX_CFG set at its reset value. • CSI2_COMPLEX_CFG must be changed according to the data rate being used.	CSI2_COMPLEXIO_CFG	
Set RXMODE and STOPSTATE FSM to RXMODE state. Users can also configure the delay for the FSM to return from RXMODE to NORXMODE when all lines reach STOPSTATE.	CSI2_TIMING[15] FORCE_RX_MODE_IO1	0x1
Activate ECC correction and error detection on short packets and packet headers. The ECC check corrects the packet if there is one error and generates an error if there is more than one error (unrecoverable error).	CSI2_CTRL[2] ECC_EN	0x1
Start the CSI2 receiver.	CSI2_CTRL[0] IF_EN	0x1
Configure the different contexts to be used.		
Link the context to a virtual channel and a data type.	See Section 2.6.4.6, *Linking a Context to a Virtual Channel and a Data Type.*	
Set the FEC_NUMBER bit field to 0x1 for a progressive video and to 0x2 for an interlaced video. For more information, see Section 2.6.3.3.5, *DMA Engine.*	CSI2_CTX_CTRL1_i[26:23] FEC_NUMBER	0x1 or 0x2
Capture an infinite number of frames (until the interface or the context is disabled).	CSI2_CTX_CTRL1_i[15:8] COUNT and CSI2_CTX_CTRL1_i[4] COUNT_UNLOCK	0x0
Enable the CRC checksum on long packet payload. This allows detection of errors, but cannot correct errors like the ECC for header and short packet. On error detection, an event is triggered (the CSI2_CTX_IRQSTATUS_i[5] CS_IRQ bit).	CSI2_CTX_CTRL1_i[5] CS_EN	
Configure the DMA engine for the current channel: Configure the ping and pong addresses.	CSI2_CTX_DAT_PING_ADDR_i[31:5] ADDR and CSI2_CTX_DAT_PING_ADDR_i[31:5] ADDR	

Table 70. ISS CSI2 Global Initialization (continued)

Step	Register/Bit Field/Programming Model	Value
Set the CSI2_CTx_DAT_OFST_i[15:5] OFST bit field to 0x0 so consecutive lines are stored consecutively in memory (image width and frame-buffer width are equal).	CSI2_CTX_DAT_OFST_i[15:5] OFST	
Keep the ALPHA setting at its reset value (0x0) for RGB padding.	CSI2_CTX_CTRL3_i[29:16] ALPHA	
Enable the contexts.	CSI2_CTX_CTRL1_i[0] CTX_EN	0x1

2.6.4.3 ISS CSI2 Disable Video/Picture Acquisition

There are two ways to end picture acquisition:

- Disable the corresponding context by setting the CSI2_CTX_CTRL1_i[0] CTX_EN bit to 0. This stops the acquisition for the current context. Other enabled contexts are still capturing frames and writing them in memory.
- Disable the CSI2 receiver interface by setting the CSI2_CTRL[0] IF_EN bit to 0. This can have an immediate effect if the CSI2_CTRL[3] FRAME bit is set to 0, or it can be effective after all the enabled contexts receive the FEC if the CSI2_CTRL[3] FRAME bit is set to 1.

2.6.4.4 ISS CSI2 Capture a Finite Number of Frames

The CSI2 receiver can be configured to capture a finite number of frames. To configure the CSI2 receiver in this mode, perform the steps listed in Table 71.

Table 71. ISS CSI2 Capture a Finite Number of Frames

Step	Bit Field	Value
Enable a write to the COUNT bit field.	CSI2_CTX_CTRL1_i[4] COUNT_UNLOCK	0x1
Set the bit field to the number of frames the CSI2 receiver must capture.	CSI2_CTX_CTRL1_i[15:8] COUNT	Valid values are 0 to 255; 0 is infinite capture and 1 to 255 defines the number of frames to capture.
Disable a write to the COUNT bit field.	CSI2_CTX_CTRL1_i[4] COUNT_UNLOCK	0x0

During frame capture, the COUNT bit field is decremented by 1 at each frame capture. Software reads the COUNT bit field to know how many frames must still be captured.

The COUNT bit can be updated during capture if the COUNT_UNLOCK bit is set to 1.

2.6.4.5 ISS CSI2 Configure a Periodic Event During Frame Acquisition

The CSI2 receiver can generate a periodic event. This line number is defined in the CSI2_CTX_CTRL3_i[15:0] LINE_NUMBER bit field. The event can be generated once or multiple times per frame, depending on the value of the CSI2_CTX_CTRL1_i[1] LINE_MODULO bit:

- If the LINE_MODULO bit = 0, the event is generated when the line number corresponding to the LINE_NUMBER bit field is received.
- If the LINE_MODULO bit = 1, the event is generated when the line number received corresponds to a multiple of the LINE_NUMBER value (LINE_NUMBER is used as a modulo).

2.6.4.6 ISS CSI2 Linking a Context to a Virtual Channel and a Data Type

The CSI2 receiver supports eight contexts and the CSI2 protocol defines four virtual channels. Therefore, a CSI2 receiver context can be associated with a virtual channel and a data type. Virtual channels are defined by a 2-bit field. Valid data types for the CSI2 receiver with their associated values are described in the CSI2_CTX_CTRL2_i[9:0] FORMAT bit field.

For each context, a CSI2_CTX_CTRL2_i register defines with which channel and data type the context is associated:

- The VIRTUAL_ID bit field defines the associated virtual ID transported by the CSI2 protocol from the camera sensor.
- The FORMAT bit field defines the associated data type. The data type is a combination of the data type transported by the CSI2 protocol and the type of storage in memory. A given data type (RGB888) can be stored in memory in different ways (RGB888 or RGB888 + EXP32). Therefore, the FORMAT bit field also defines how DMA stores data in memory.

For example, for the current context to capture a frame from virtual channel 2 and data type RAW12 with data expansion (RAW12 + EXP16), write the value 0x10AC (0x2 11 + 0xAC) in the 16 LSBs of the CSI2_CTX_CTRL2_i register.

2.6.4.7 ISS CSI2 Progressive and Interleaved Frame Configuration

The CSI2 receiver can treat progressive and interlaced frames. There is no progressive or interleaved mode, but the CSI2_CTX_CTRL1_i[23:16] FEC_NUMBER bit field controls the number of FECs before swapping to the other (ping or pong) buffer. Therefore, two modes are possible:

- FEC_NUMBER = 1: This is equivalent to progressive mode. After a FEC on the context, the current buffer is switched (ping to pong or pong to ping). The image in the memory buffer consists of one transmitted frame.
- FEC_NUMBER 1: The current buffer is switched (ping to pong or pong to ping) after the FEC_NUMBER FEC is received for the context. The image in the memory buffer consists of the FEC_NUMBER transmitted frame.

For more information about how data is stored in memory through the DMA, see Section 2.6.3.3.5, *DMA Engine*.

NOTE: If FEC_NUMBER 1, the camera sensor must send the line number information with the current line. Otherwise, the CSI2 receiver cannot calculate each line address.

2.6.4.8 ISS CSI2 Progressive and Interleaved Frame Configuration

Table 72 lists the procedure to enable debug mode.

Table 72. ISS CSI2 Enable Debug Mode

Step	Bit	Value
Enable debug mode.	CSI2_CTRL[7] DBG_EN	0x1

- During debug mode the input does not come from the CSI2 receiver interface but from the CSI2_DBG_H and CSI2_DBG_P registers. The full CSI2 receiver function can be debugged in debug mode. Full 32-bit values must always be written to the CSI2_DBG_H register. The CSI2_CTRL[0] IF_EN bit has no affect during debug mode. To reset the FIFO in case of overflow, the CSI2_CTRL[7] DBG_EN bit must be reset to 0, and the interface must be enabled by setting the CSI2_CTRL[0] IF_EN bit to 0x1.
- The CSI2_DBG_H register is used to provide short packet and long packet headers.
- The CSI2_DBG_P register is used to provide long packet payload.

The following examples apply to the CSI2_DBG_H register:

- The sync codes for virtual channel 0 are written as CSI2_DBG_H = 0xFF00 0000 or 0xFF00 0001, or 0xFF00 0002 or 0xFF00 0003. To send the RAW12 pixels 0x673, 0x452, 0x01d, 0xefc, 0xab0, 0x891, 0x326, 0x547, write CSI2_DBG_H = 0x0123 4567, followed by CSI2_DBG_H = 0x89abcdef, and CSI2_DBG_H = 0x7654 3210.

2.6.5 ISS CSI2 Registers

Table 73 lists the CSI2 instances.

Table 73. ISS CSI2 Instance Summary

Module Name	Base Address Cortex-M3 Private Access	Base Address L3 Interconnect	Size
ISS_CSI2_A_REGS1	0x5504 1000	0x5C00 1000	368 bytes
ISS_CSI2_A_REGS2	0x5504 11C0	0x5C00 11C0	64 bytes

2.6.5.1 ISS CSI2 REGS1 Registers

Table 74 summarizes the CSI2 REGS1 registers.

Table 74. ISS CSI2 REGS1 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ISS_CSI2_A_REGS1 Base Address Cortex-M3 Private Access	ISS_CSI2_A_REGS1 Base Address L3 Interconnect
CSI2_REVISION	R	32	0x0000 0000	0x5504 1000	0x5C00 1000
CSI2_SYSCONFIG	RW	32	0x0000 0010	0x5504 1010	0x5C00 1010
CSI2_SYSSTATUS	R	32	0x0000 0014	0x5504 1014	0x5C00 1014
CSI2_IRQSTATUS	RW	32	0x0000 0018	0x5504 1018	0x5C00 1018
CSI2_IRQENABLE	RW	32	0x0000 001C	0x5504 101C	0x5C00 101C
CSI2_CTRL	RW	32	0x0000 0040	0x5504 1040	0x5C00 1040
CSI2_DBG_H	W	32	0x0000 0044	0x5504 1044	0x5C00 1044
RESERVED	R	32	0x0000 0048	0x5504 1048	0x5C00 1048
RESERVED	RW	32	0x0000 004C	0x5504 104C	0x5C00 104C
CSI2_COMPLEXIO_CFG	RW	32	0x0000 0050	0x5504 1050	0x5C00 1050
CSI2_COMPLEXIO_IRQSTATUS	RW	32	0x0000 0054	0x5504 1054	0x5C00 1054
RESERVED	RW	32	0x0000 0058	0x5504 1058	0x5C00 1058
CSI2_SHORT_PACKET	R	32	0x0000 005C	0x5504 105C	0x5C00 105C
CSI2_COMPLEXIO_IRQENABLE	RW	32	0x0000 0060	0x5504 1060	0x5C00 1060
RESERVED	RW	32	0x0000 0064	0x5504 1064	0x5C00 1064
CSI2_DBG_P	W	32	0x0000 0068	0x5504 1068	0x5C00 1068
CSI2_TIMING	RW	32	0x0000 006C	0x5504 106C	0x5C00 106C
CSI2_CTX_CTRL1_i (1)	RW	32	0x0000 0070 + (0x20 * i)	0x5504 1070 + (0x20 * i)	0x5C00 1070 + (0x20 * i)
CSI2_CTX_CTRL2_i (1)	RW	32	0x0000 0074 + (0x20 * i)	0x5504 1074 + (0x20 * i)	0x5C00 1074 + (0x20 * i)
CSI2_CTX_DAT_OFST_i (1)	RW	32	0x0000 0078 + (0x20 * i)	0x5504 1078 + (0x20 * i)	0x5C00 1078 + (0x20 * i)
CSI2_CTX_DAT_PING_ADDR_i (1)	RW	32	0x0000 007C + (0x20 * i)	0x5504 107C + (0x20 * i)	0x5C00 107C + (0x20 * i)
CSI2_CTX_DAT_PONG_ADDR_i (1)	RW	32	0x0000 0080 + (0x20 * i)	0x5504 1080 + (0x20 * i)	0x5C00 1080 + (0x20 * i)
CSI2_CTX_IRQENABLE_i (1)	RW	32	0x0000 0084 + (0x20 * i)	0x5504 1084 + (0x20 * i)	0x5C00 1084 + (0x20 * i)
CSI2_CTX_IRQSTATUS_i (1)	RW	32	0x0000 0088 + (0x20 * i)	0x5504 1088 + (0x20 * i)	0x5C00 1088 + (0x20 * i)
CSI2_CTX_CTRL3_i (1)	RW	32	0x0000 008C + (0x20 * i)	0x5504 108C + (0x20 * i)	0x5C00 108C + (0x20 * i)

(1) i = 0 to 7

2.6.5.1.1 CSI2_REVISION

Table 75. CSI2_REVISION

Address Offset	0x0000 0000		
Physical Address	0x5504 1000 0x5C00 1000	**Instance**	ISS_CSI2_A_REGS1_CORTEX-M3 ISS_CSI2_A_REGS1_L3
Description	IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	See[1]

(1) TI internal data

2.6.5.1.2 CSI2_SYSCONFIG

Table 76. CSI2_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x5504 1010 0x5C00 1010	**Instance**	ISS_CSI2_A_REGS1_CORTEX-M3 ISS_CSI2_A_REGS1_L3
Description	SYSTEM CONFIGURATION REGISTER This register is the OCP-socket system configuration register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		MSTANDBY_MODE		RESERVED										SOFT_RESET	AUTO_IDLE

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Reserved	R	0x00000
13:12	MSTANDBY_MODE	Sets the behavior of the master port power management signals. 0x0: Force-standby. MStandby is only asserted when the module is disabled. 0x1: No-standby. MStandby is never asserted. 0x2: Smart-standby: MStandby is asserted based on the activity of the module. The module will try to go to standby during the vertical blanking period.	RW	0x0
11:2	RESERVED	Reserved	R	0x000
1	SOFT_RESET	Software reset. Set the bit to 1 to trigger a module reset. The bit is automatically reset by the hardware. During reads return 0. 0x0: Normal mode. 0x1: The module is reset Note: Before setting the software reset bit to 1 in CSI2_SYSCONFIG register, the user must have access to a CSI2 receiver register.	RW	0
0	AUTO_IDLE	Internal OCP gating strategy 0x0: OCP clock is free-running. 0x1: Automatic OCP clock gating strategy is applied based on the OCP interface activity.	RW	1

2.6.5.1.3 CSI2_SYSSTATUS

Table 77. CSI2_SYSSTATUS

Address Offset	0x0000 0014		
Physical Address	0x5504 1014 0x5C00 1014	**Instance**	ISS_CSI2_A_REGS1_CORTEX-M3 ISS_CSI2_A_REGS1_L3
Description	SYSTEM STATUS REGISTER This register provides status information about the module, excluding the interrupt status register.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															RESET_DONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads returns 0.	R	0x0000 0000
0	RESET_DONE	Internal reset monitoring Read 0x1: Reset completed. Read 0x0: Internal module reset is on going.	R	1

2.6.5.1.4 CSI2_IRQSTATUS

Table 78. CSI2_IRQSTATUS

Address Offset	0x0000 0018		
Physical Address	0x5504 1018 0x5C00 1018	**Instance**	ISS_CSI2_A_REGS1_CORTEX-M3 ISS_CSI2_A_REGS1_L3
Description	INTERRUPT STATUS REGISTER - All contexts This register associates one bit for each context in order to determine which context has generated the interrupt. The context shall be enabled for events to be generated on that context. If the context is disabled, the interrupt is not generated.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	OCP_ERR_IRQ	SHORT_PACKET_IRQ	ECC_CORRECTION_IRQ	ECC_NO_CORRECTION_IRQ	RESERVED	COMPLEXIO_ERR_IRQ	FIFO_OVF_IRQ	CONTEXT7	CONTEXT6	CONTEXT5	CONTEXT4	CONTEXT3	CONTEXT2	CONTEXT1	CONTEXT0

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Reserved	R	0x00000
14	OCP_ERR_IRQ	OCP Error Interrupt 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
13	SHORT_PACKET_IRQ	Short packet reception status (other than synch events: Line Start, Line End, Frame Start, and Frame End: data type between 0x8 and x0F only shall be considered). 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
12	ECC_CORRECTION_IRQ	ECC has been used to do the correction of the only 1-bit error status (short packet only). 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
11	ECC_NO_CORRECTION_IRQ	ECC error status (short and long packets). No correction of the header because of more than 1-bit error. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
10	RESERVED	Reserved	R	0
9	COMPLEXIO_ERR_IRQ	Error signaling from complex I/O: status of the PHY errors received from the complex I/O (events are defined in CSI2_COMPLEXIO_IRQSTATUS for the complex I/O). Read 0x1: READS: Event is true (pending). Read 0x0: READS: Event is false.	R	0

Bits	Field Name	Description	Type	Reset
8	FIFO_OVF_IRQ	FIFO overflow error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
7	CONTEXT7	Context 7 Read 0x1: READS: Event is true (pending). Read 0x0: READS: Event is false.	R	0
6	CONTEXT6	Context 6 Read 0x1: READS: Event is true (pending). Read 0x0: READS: Event is false.	R	0
5	CONTEXT5	Context 5 Read 0x1: READS: Event is true (pending). Read 0x0: READS: Event is false.	R	0
4	CONTEXT4	Context 4 Read 0x1: READS: Event is true (pending). Read 0x0: READS: Event is false.	R	0
3	CONTEXT3	Context 3 Read 0x1: READS: Event is true (pending). Read 0x0: READS: Event is false.	R	0
2	CONTEXT2	Context 2 Read 0x1: READS: Event is true (pending). Read 0x0: READS: Event is false.	R	0
1	CONTEXT1	Context 1 Read 0x1: READS: Event is true (pending). Read 0x0: READS: Event is false.	R	0
0	CONTEXT0	Context 0 Read 0x1: READS: Event is true (pending). Read 0x0: READS: Event is false.	R	0

2.6.5.1.5 CSI2_IRQENABLE

Table 79. CSI2_IRQENABLE

Address Offset	0x0000 001C		
Physical Address	0x5504 101C #INTFC_CSI2_IRQENABLE_Call_4	**Instance**	ISS_CSI2_A_REGS1_CORTEX-M3 ISS_CSI2_A_REGS1_L3
Description	INTERRUPT ENABLE REGISTER - All contexts This register associates one bit for each context in order to enable/disable each context individually.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	OCP_ERR_IRQ	SHORT_PACKET_IRQ	ECC_CORRECTION_IRQ	ECC_NO_CORRECTION_IRQ	RESERVED	COMPLEXIO_ERR_IRQ	FIFO_OVF_IRQ	CONTEXT7	CONTEXT6	CONTEXT5	CONTEXT4	CONTEXT3	CONTEXT2	CONTEXT1	CONTEXT0

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Reserved	R	0x00000
14	OCP_ERR_IRQ	OCP Error Interrupt 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
13	SHORT_PACKET_IRQ	Short packet reception (other than synch events: Line Start, Line End, Frame Start, and Frame End: data type between 0x8 and x0F only shall be considered). 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
12	ECC_CORRECTION_IRQ	ECC has been used to correct the only 1-bit error (short packet only). 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
11	ECC_NO_CORRECTION_IRQ	ECC error (short and long packets). No correction of the header because of more than 1-bit error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
10	RESERVED	Reserved	RW	0
9	COMPLEXIO_ERR_IRQ	Error signaling from complex I/O: the interrupt is triggered when any error is received from the complex I/O (events are defined in CSI2_COMPLEXIO_IRQSTATUS for the complex I/O). 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
8	FIFO_OVF_IRQ	FIFO overflow enable 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
7	CONTEXT7	Context 7 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

Bits	Field Name	Description	Type	Reset
6	CONTEXT6	Context 6 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
5	CONTEXT5	Context 5 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
4	CONTEXT4	Context 4 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
3	CONTEXT3	Context 3 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
2	CONTEXT2	Context 2 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
1	CONTEXT1	Context 1 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
0	CONTEXT0	Context 0 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

2.6.5.1.6 CSI2_CTRL

Table 80. CSI2_CTRL

Address Offset	0x0000 0040		
Physical Address	0x5504 1040 0x5C00 1040	**Instance**	ISS_CSI2_A_REGS1_CORTEX-M3 ISS_CSI2_A_REGS1_L3
Description	GLOBAL CONTROL REGISTER This register controls the CSI2 RECEIVER module. This register shall not be modified dynamically (except IF_EN bit field).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								MFLAG_LEVH				MFLAG_LEVL				BURST_SIZE_EXPAND	VP_CLK_EN	RESERVED	NON_POSTED_WRITE	RESERVED	VP_ONLY_EN	STREAMING_32_BIT	VP_OUT_CTRL	DBG_EN	BURST_SIZE		ENDIANNESS	FRAME	ECC_EN	RESERVED	IF_EN

Bits	Field Name	Description	Type	Reset
31:23	RESERVED	Reserved	R	0x000
22:20	MFLAG_LEVH	Controls assertion of the MFlag[1:0] OCP sideband signal. Check the OCP master port definition for details. 0x0: 8/8 of the FIFO size 0x1: 7/8 of the FIFO size 0x2: 6/8 of the FIFO size 0x3: 5/8 of the FIFO size 0x4: 4/8 of the FIFO size 0x5: 3/8 of the FIFO size 0x6: 2/8 of the FIFO size 0x7: 1/8 of the FIFO size	RW	0x0
19:17	MFLAG_LEVL	Controls assertion of the MFlag[1:0] OCP sideband signal. Check the OCP master port definition for details 0x0: 8/8 of the FIFO size 0x1: 7/8 of the FIFO size 0x2: 6/8 of the FIFO size 0x3: 5/8 of the FIFO size 0x4: 4/8 of the FIFO size 0x5: 3/8 of the FIFO size 0x6: 2/8 of the FIFO size 0x7: 1/8 of the FIFO size	RW	0x0
16	BURST_SIZE_EXPAND	Sets the DMA burst size on the L3 interconnect. 0x0: Use the burst size defined in the BURST_SIZE register 0x1: Allow generation of 16x64-bit bursts	RW	0
15	VP_CLK_EN	VP clock enable. 0x0: The VP clock is disabled. 0x1: The VP clock is enabled.	RW	0
14	RESERVED	Read returns reset value	RW	0

Bits	Field Name	Description	Type	Reset
13	NON_POSTED_WRITE	Not posted writes 0x0: Disable 0x1: Enable	RW	0
12	RESERVED	Reserved	R	0
11	VP_ONLY_EN	VP only enable. 0x0: The VP is enabled and the OCP master port is enabled. 0x1: The VP is enabled and the OCP master port is disabled.	RW	0
10	STREAMING_32_BIT	Indicates if 64-bit or 32-bit streaming burst is used. Valid only if CSI2_CTRL.STREAMING=1 0x0: 64-bit streaming burst is used; byte enable pattern is 0xFF 0x1: 32-bit streaming burst is used; byte enable pattern is 0x0F	RW	0
9:8	VP_OUT_CTRL	VP_PCLK control. Sets the VP_PCLK as a function of the ISS interconnect interface clock (OCPCLK). 0x0: No division: VP_PCLK = OCPCLK. 0x1: Division by 2: VP_PCLK = OCPCLK / 2. 0x3: Division by 4: VP_PCLK = OCPCLK / 4. 0x2: Division by 3: VP_PCLK = OCPCLK / 3. Example scenarios: - Low VP_PCLK, Memory - VP: Same as typical memory - VP, but VP_PCLK = OCPCLK/2 - Typical sensor - VP: Autoidle enabled, FCLK at optimal rate, sensor provides DPCM compressed RAW12 data at 650 Mbps. Image timings VP_PCLK = (OCPCLK/2) and SC_CTRL[31:15] FRACDIV = 0xD000 2600 active pixels/line, 128 blanking pixels, no vertical blanking. (This scenario corresponds to the OTF operation at maximum SC speed.)	RW	0x0
7	DBG_EN	Enables the debug mode. 0x0: Disable 0x1: Enable	RW	0
6:5	BURST_SIZE	Sets the DMA burst size on the L3 interconnect. 0x0: 1x64 OCP writes 0x1: 2x64 OCP writes 0x2: 4x64 OCP writes 0x3: 8x64 OCP writes	RW	0x0
4	ENDIANNESS	Select endianness for YUV4:2:2 8 bit and YUV4:2:0 legacy formats. 0x0: Use native MIPI CSI2 endianness: Little endian for all formats except for YUV4:2:2 8b and YUV4:2:0 Legacy which a big endian. 0x1: Store all pixel formats little endian.	RW	0
3	FRAME	Set the modality in which IF_EN works. 0x0: If IF_EN = 0 the interface is disabled immediately. 0x1: If IF_EN = 1 the interface is disabled after all FEC sync code have been received for the active contexts.	RW	0
2	ECC_EN	Enables the Error Correction Code check for the received header (short and long packets for all virtual channel ids). 0x0: Disabled 0x1: Enabled	RW	0
1	RESERVED	Read returns reset value	RW	0

Bits	Field Name	Description	Type	Reset
0	IF_EN	Enables the physical interface to the module. 0x0: The interface is disabled. If FRAME = 0, it is disabled immediately. If FRAME = 1, it is disabled when each context has received the FEC sync code. 0x1: The interface is enabled immediately, the data acquisition starts on the next FSC sync code. Writing 1 to this register when the current value is 0 has the effect to clear the output FIFO. The pixel data of the following frame will be written in the PING buffer, that is, the CSI2_CTX_CTRL.PING_PONG bits are reset to 0 as well.	RW	0

2.6.5.1.7 CSI2_DBG_H

Table 81. CSI2_DBG_H

Address Offset	0x0000 0044		
Physical Address	0x5504 1044 0x5C00 1044	**Instance**	ISS_CSI2_A_REGS1_CORTEX-M3 ISS_CSI2_A_REGS1_L3
Description	DEBUG REGISTER (Header) This register provides a way to debug the CSI2 RECEIVER module with no image sensor connected to the module. The debug mode is enabled by CSI2_CTRL.DBG_EN. Only full 32-bit values shall be written. The register is used to write short packets and header of long packets.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBG																															

Bits	Field Name	Description	Type	Reset
31:0	DBG	32-bit input value.	W	0x0000 0000

2.6.5.1.8 CSI2_COMPLEXIO_CFG

Table 82. CSI2_COMPLEXIO_CFG

Address Offset	0x0000 0050		
Physical Address	0x5504 1050 0x5C00 1050	**Instance**	ISS_CSI2_A_REGS1_CORTEX-M3 ISS_CSI2_A_REGS1_L3
Description	COMPLEXIO CONFIGURATION REGISTER for the complex I/O This register contains the lane configuration for the order and position of the lanes (clock and data) and the polarity order for the control of the PHY differential signals in addition to the control bit for the power FSM.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESET_CTRL	RESET_DONE	PWR_CMD		PWR_STATUS		PWR_AUTO	RESERVED				DATA4_POL	DATA4_POSITION			DATA3_POL	DATA3_POSITION			DATA2_POL	DATA2_POSITION			DATA1_POL	DATA1_POSITION			CLOCK_POL	CLOCK_POSITION		

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30	RESET_CTRL	Controls the reset of the complex I/O 0x0: Complex I/O reset active. 0x1: Complex I/O reset deasserted.	RW	0
29	RESET_DONE	Internal reset monitoring of the power domain using the byte clock provided by the associated CSIPHY (see Section 1.1.1.1, *ISS Clock Domains*. Read 0x1: Reset completed. Read 0x0: Internal module reset is on going.	R	0
28:27	PWR_CMD	Command for power control of the complex I/O 0x0: Command to change to OFF state 0x1: Command to change to ON state 0x2: Command to change to Ultralow-Power state	RW	0x0
26:25	PWR_STATUS	Status of the power control of the complex I/O Read 0x0: Complex I/O in OFF state Read 0x1: Complex I/O in ON state Read 0x2: Complex I/O in Ultralow-Power state	R	0x0
24	PWR_AUTO	Automatic switch between ULP and ON states based on ULPM signals from complex I/O 0x0: Disable 0x1: Enable	RW	0
23:20	RESERVED	Reserved	R	0x0
19	DATA4_POL	+/- differential pin order of data lane 4. 0x0: +/- pin order 0x1: -/+ pin order	RW	0

Bits	Field Name	Description	Type	Reset
18:16	DATA4_POSITION	Position and order of the data lane 4. The values 6 and 7 are reserved. 0x0: Not used/connected 0x1: Data lane 4 is at the position 1. 0x2: Data lane 4 is at the position 2. 0x3: Data lane 4 is at the position 3. 0x4: Data lane 4 is at the position 4. 0x5: Data lane 4 is at the position 5.	RW	0x0
15	DATA3_POL	+/- differential pin order of data lane 3. 0x0: +/- pin order 0x1: -/+ pin order	RW	0
14:12	DATA3_POSITION	Position and order of the data lane 3. The values 6 and 7 are reserved. 0x0: Not used/connected 0x1: Data lane 3 is at the position 1. 0x2: Data lane 3 is at the position 2. 0x3: Data lane 3 is at the position 3. 0x4: Data lane 3 is at the position 4. 0x5: Data lane 3 is at the position 5.	RW	0x0
11	DATA2_POL	+/- differential pin order of DATA lane 2. 0x0: +/- pin order (csi2_dx=+ and csi2_dy=-) 0x1: -/+ pin order (csi2_dx=- and csi2_dy=+)	RW	0
10:8	DATA2_POSITION	Position and order of the data lane 2. The values 6 and 7 are reserved. 0x0: Not used/connected 0x1: Data lane 2 is at the position 1. 0x2: Data lane 2 is at the position 2. 0x3: Data lane 2 is at the position 3. 0x4: Data lane 2 is at the position 4. 0x5: Data lane 2 is at the position 5.	RW	0x0
7	DATA1_POL	+/- differential pin order of data lane 1. 0x0: +/- pin order (csi2_dx=+ and csi2_dy=-) 0x1: -/+ pin order (csi2_dx=- and csi2_dy=+)	RW	0
6:4	DATA1_POSITION	Position and order of the DATA lane 1. 0, 6 and 7 are reserved. The data lane 1 is always present. 0x0: Not used/connected 0x1: Data lane 1 is at the position 1. 0x2: Data lane 1 is at the position 2. 0x3: Data lane 1 is at the position 3. 0x4: Data lane 1 is at the position 4. 0x5: Data lane 1 is at the position 5.	RW	0x0
3	CLOCK_POL	+/- differential pin order of clock lane. 0x0: +/- pin order (csi2_dx=+ and csi2_dy=-) 0x1: -/+ pin order (csi2_dx=- and csi2_dy=+)	RW	0

Bits	Field Name	Description	Type	Reset
2:0	CLOCK_POSITION	Position and order of the clock lane. 0, 6 and 7 are reserved. The clock lane is always present. 0x0: Not used/connected 0x1: Clock lane is at the position 1. 0x2: Clock lane is at the position 2. 0x3: Clock lane is at the position 3. 0x4: Clock lane is at the position 4. 0x5: Reserved	RW	0x0

2.6.5.1.9 CSI2_COMPLEXIO_IRQSTATUS

Table 83. CSI2_COMPLEXIO_IRQSTATUS

Address Offset	0x0000 0054		
Physical Address	0x5504 1054 0x5C00 1054	**Instance**	ISS_CSI2_A_REGS1_CORTEX-M3 ISS_CSI2_A_REGS1_L3
Description	INTERRUPT STATUS REGISTER - All errors from complex I/O #1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					STATEALLULPMEXIT	STATEALLULPMENTER	STATEULPM5	STATEULPM4	STATEULPM3	STATEULPM2	STATEULPM1	ERRCONTROL5	ERRCONTROL4	ERRCONTROL3	ERRCONTROL2	ERRCONTROL1	ERRESC5	ERRESC4	ERRESC3	ERRESC2	ERRESC1	ERRSOTSYNCHS5	ERRSOTSYNCHS4	ERRSOTSYNCHS3	ERRSOTSYNCHS2	ERRSOTSYNCHS1	ERRSOTHS5	ERRSOTHS4	ERRSOTHS3	ERRSOTHS2	ERRSOTHS1

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Reserved	R	0x00
26	STATEALLULPMEXIT	At least one of the active lanes has exit the ULPM 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
25	STATEALLULPMENTER	All active lanes are entering in ULPM. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
24	STATEULPM5	Lane 5 in ULPM 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
23	STATEULPM4	Lane 4 in ULPM 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
22	STATEULPM3	Lane 3 in ULPM 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
21	STATEULPM2	Lane 2 in ULPM 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
20	STATEULPM1	Lane 1 in ULPM 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
19	ERRCONTROL5	Control error for lane 5 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
18	ERRCONTROL4	Control error for lane 4 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
17	ERRCONTROL3	Control error for lane 3 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
16	ERRCONTROL2	Control error for lane 2 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
15	ERRCONTROL1	Control error for lane 1 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
14	ERRESC5	Escape entry error for lane 5 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
13	ERRESC4	Escape entry error for lane 4 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
12	ERRESC3	Escape entry error for lane 3 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
11	ERRESC2	Escape entry error for lane 2 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
10	ERRESC1	Escape entry error for lane 1 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
9	ERRSOTSYNCHS5	Start of transmission sync error for lane 5 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
8	ERRSOTSYNCHS4	Start of transmission sync error for lane 4 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
7	ERRSOTSYNCHS3	Start of transmission sync error for lane 3 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
6	ERRSOTSYNCHS2	Start of transmission sync error for lane 2 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
5	ERRSOTSYNCHS1	Start of transmission sync error for lane 1 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
4	ERRSOTHS5	Start of transmission error for lane 5 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
3	ERRSOTHS4	Start of transmission error for lane 4 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
2	ERRSOTHS3	Start of transmission error for lane 3 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
1	ERRSOTHS2	Start of transmission error for lane 2 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
0	ERRSOTHS1	Start of transmission error for lane 1 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0

2.6.5.1.10 CSI2_SHORT_PACKET

Table 84. CSI2_SHORT_PACKET

Address Offset	0x0000 005C		
Physical Address	0x5504 105C 0x5C00 105C	**Instance**	ISS_CSI2_A_REGS1_CORTEX-M3 ISS_CSI2_A_REGS1_L3
Description	SHORT PACKET INFORMATION - This register sets the 24-bit DATA_ID + Short Packet Data Field when the data type is between 0x8 and x0F		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SHORT_PACKET																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reads returns 0.	R	0x00
23:0	SHORT_PACKET	Short Packet information: DATA ID + DATA FIELD	R	0x000000

2.6.5.1.11 CSI2_COMPLEXIO_IRQENABLE

Table 85. CSI2_COMPLEXIO_IRQENABLE

Address Offset	0x0000 0060		
Physical Address	0x5504 1060 0x5C00 1060	**Instance**	ISS_CSI2_A_REGS1_CORTEX-M3 ISS_CSI2_A_REGS1_L3
Description	INTERRUPT ENABLE REGISTER - All errors from complex I/O		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					STATEALLULPMEXIT	STATEALLULPMENTER	STATEULPM5	STATEULPM4	STATEULPM3	STATEULPM2	STATEULPM1	ERRCONTROL5	ERRCONTROL4	ERRCONTROL3	ERRCONTROL2	ERRCONTROL1	ERRESC5	ERRESC4	ERRESC3	ERRESC2	ERRESC1	ERRSOTSYNCHS5	ERRSOTSYNCHS4	ERRSOTSYNCHS3	ERRSOTSYNCHS2	ERRSOTSYNCHS1	ERRSOTHS5	ERRSOTHS4	ERRSOTHS3	ERRSOTHS2	ERRSOTHS1

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Reserved	R	0x00
26	STATEALLULPMEXIT	At least one of the active lanes has exit the ULPM 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
25	STATEALLULPMENTER	All active lanes are entering in ULPM. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
24	STATEULPM5	Lane 5 in ULPM 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
23	STATEULPM4	Lane 4 in ULPM 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
22	STATEULPM3	Lane 3 in ULPM 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
21	STATEULPM2	Lane 2 in ULPM 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
20	STATEULPM1	Lane 1 in ULPM 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
19	ERRCONTROL5	Control error for lane 5 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
18	ERRCONTROL4	Control error for lane 4 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

Bits	Field Name	Description	Type	Reset
17	ERRCONTROL3	Control error for lane 3 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
16	ERRCONTROL2	Control error for lane 2 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
15	ERRCONTROL1	Control error for lane 1 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
14	ERRESC5	Escape entry error for lane 5 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
13	ERRESC4	Escape entry error for lane 4 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
12	ERRESC3	Escape entry error for lane 3 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
11	ERRESC2	Escape entry error for lane 2 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
10	ERRESC1	Escape entry error for lane 1 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
9	ERRSOTSYNCHS5	Start of transmission sync error for lane 5 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
8	ERRSOTSYNCHS4	Start of transmission sync error for lane 4 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
7	ERRSOTSYNCHS3	Start of transmission sync error for lane 3 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
6	ERRSOTSYNCHS2	Start of transmission sync error for lane 2 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
5	ERRSOTSYNCHS1	Start of transmission sync error for lane 1 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
4	ERRSOTHS5	Start of transmission error for lane 5 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
3	ERRSOTHS4	Start of transmission error for lane 4 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
2	ERRSOTHS3	Start of transmission error for lane 3 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

Bits	Field Name	Description	Type	Reset
1	ERRSOTHS2	Start of transmission error for lane 2 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
0	ERRSOTHS1	Start of transmission error for lane 1 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

2.6.5.1.12 CSI2_DBG_P

Table 86. CSI2_DBG_P

Address Offset	0x0000 0068		
Physical Address	0x5504 1068 0x5C00 1068	**Instance**	ISS_CSI2_A_REGS1_CORTEX-M3 ISS_CSI2_A_REGS1_L3
Description	DEBUG REGISTER (Payload) This register provides a way to debug the CSI2 RECEIVER module with no image sensor connected to the module. The debug mode is enabled by CSI2_CTRL.DBG_EN. Only full 32-bit values shall be written. The register is used to write payload of long packets.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBG																															

Bits	Field Name	Description	Type	Reset
31:0	DBG	32-bit input value.	W	0x0000 0000

2.6.5.1.13 CSI2_TIMING

Table 87. CSI2_TIMING

Address Offset	0x0000 006C		
Physical Address	0x5504 106C 0x5C00 106C	**Instance**	ISS_CSI2_A_REGS1_CORTEX-M3 ISS_CSI2_A_REGS1_L3
Description	TIMING REGISTER This register controls the CSI2 RECEIVER module. This register shall not be modified while CSI2_CTRL.IF_EN is set to 1. It is used to indicate the number of L3 cycles for the Stop State monitoring.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			RESERVED													FORCE_RX_MODE_IO1			STOP_STATE_X16_IO1					STOP_STATE_X4_IO1							
RESERVED			RESERVED																STOP_STATE_COUNTER_IO1												
RESERVED			RESERVED																												

Bits	Field Name	Description	Type	Reset
31	RESERVED	Read returns reset value	RW	0
30	RESERVED	Read returns reset value	RW	1
29	RESERVED	Read returns reset value	RW	1
28:16	RESERVED	Read returns reset value	RW	0x1FFF
15	FORCE_RX_MODE_IO1	Control of ForceRxMode signal 0x0: Deassertion of ForceRxMode. The hardware reset the bit at the end of the Force RX Mode assertion. The software can reset the bit in order to stop the assertion of the ForceRXMode signal prior to the completion of the period. 0x1: Assertion of ForceRxMode	RW	0
14	STOP_STATE_X16_IO1	Multiplication factor for the number of L3 cycles defined in STOP_STATE_COUNTER bit field 0x0: The number of L3 cycles defined in STOP_STATE _COUNTER is multiplied by 1x 0x1: The number of L3 cycles defined in STOP_STATE _COUNTER is multiplied by 16x	RW	1
13	STOP_STATE_X4_IO1	Multiplication factor for the number of L3 cycles defined in STOP_STATE_COUNTER bit field 0x0: The number of L3 cycles defined in STOP_STATE _COUNTER is multiplied by 1x 0x1: The number of L3 cycles defined in STOP_STATE _COUNTER is multiplied by 4x	RW	1
12:0	STOP_STATE_COUNTER_IO1	Stop State counter for monitoring. It indicates the number of L3 to monitor for Stop State before deasserting ForceRxMode (complex I/O 1). The value is from 0 to 8191.	RW	0x1FFF

2.6.5.1.14 CSI2_CTX_CTRL1_i

Table 88. CSI2_CTX_CTRL1_i

Address Offset	0x0000 0070 + (0x20 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 1070 + (0x20 * i) 0x5C00 1070 + (0x20 * i)	**Instance**	ISS_CSI2_A_REGS1_CORTEX-M3 ISS_CSI2_A_REGS1_L3
Description	CONTROL REGISTER - Context This register controls the Context. This register is shadowed: modifications are taken into account after the next FSC sync code.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BYTESWAP		GENERIC		RESERVED		HSCALE		TRANSCODE								FEC_NUMBER								COUNT							
																								EOF_EN	EOL_EN	CS_EN	COUNT_UNLOCK	PING_PONG	VP_FORCE	LINE_MODULO	CTX_EN

Bits	Field Name	Description	Type	Reset
31	BYTESWAP	Allows swapping bytes two by two in the payload data. It does not affect: - short packets - long packet header or footers - CRC calculation The purpose is to by swap data send to the OCP port and/or video port 0x0: Disabled 0x1: Enabled	RW	0
30	GENERIC	Enables the generic mode. 0x0: Disabled. Data is received according to CSI2_CTX_CTRL1.FORMAT and the long packet code transmitted in the MIPI stream is used. 0x1: Enabled. Data is received according to CSI2_CTX_CTRL1.FORMAT and the long packet code transmitted in the MIPI stream is ignored.	RW	0
29	RESERVED	Reserved	R	0x0
28	HSCALE	Enable horizontal downscaling by a factor of two. Applies to RAW data when transcoding is enabled. 0x0: Disable 0x1: Enable	RW	0x0

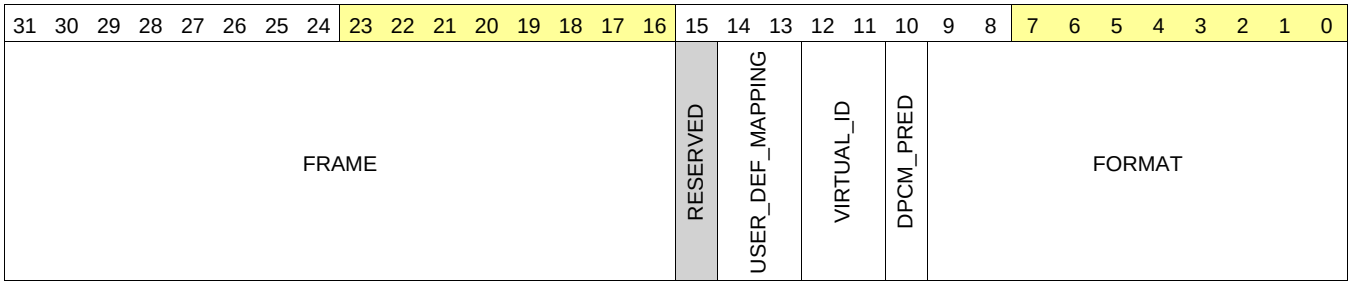
Bits	Field Name	Description	Type	Reset
27:24	TRANSCODE	Enables image transcoding. When this features is enabled: - the data format from the camera is defined by the FORMAT register - the format after transcode is defined by the TRANSCODE register. The memory storage / video port formats is defined by the TRANSCODE register 0x0: Feature disabled. 0x1: Outputs DPCM compressed RAW10 data. After compression, pixels are coded on 8 bits. Data in memory is organized as regular RAW8 data 0x2: Outputs DPCM compressed RAW12 data. After compression, pixels are coded on 8 bits. Data in memory is organized as regular RAW8 data 0x3: Outputs A-Law compressed RAW10 data. After compression, pixels are coded on 8 bits. Data in memory is organized as regular RAW8 data. 0x4: Outputs uncompressed RAW8 data. Data in memory is organized as regular RAW8 data 0x5: Outputs uncompressed RAW10 data. Data in memory is organized as regular RAW10+EXP16 data 0x6: Outputs uncompressed RAW10 data. Data in memory is organized as regular packed RAW10 data 0x7: Outputs uncompressed RAW12 data. Data in memory is organized as regular RAW12+EXP16 data 0x8: Outputs uncompressed RAW12 data. Data in memory is organized as regular packed RAW12 data 0x9: Outputs uncompressed RAW14 data.	RW	0x0
23:16	FEC_NUMBER	Number of FEC to receive between using swap of CSI2_CTX_DAT_PING_ADDR and CSI2_CTX_DAT_PONG_ADDR for the calculation of the address in memory (must be used only in interlace mode, otherwise set to 1).	RW	0x01
15:8	COUNT	Sets the number of frame to acquire. Once the frame acquisition starts, the COUNT value is decremented after every frame. When COUNT reaches 0, the FRAME_NUMBER_IRQ interrupt is triggered and CTX_EN is set to 0. Writes to this bit field are controlled by the COUNT_UNLOCK bit. During the same OCP write access, the bit field COUNT_UNLOCK shall be written in addition to COUNT bit field in order to change the COUNT value. COUNT can be overwritten dynamically with a new count value. 0: Infinite number of frames (no count). 1: 1 frame to acquire ... 255: 255 frames to acquire	RW	0x00
7	EOF_EN	Indicates if the end of frame signal shall be asserted at the end of the line. Read 0x1: The end of frame signal is asserted at the end of each frame. Read 0x0: The end of frame signal is not asserted at the end of each frame.	RW	0

Bits	Field Name	Description	Type	Reset
6	EOL_EN	Indicates if the end of line signal shall be asserted at the end of the line. Read 0x1: The end of line signal is asserted at the end of each frame. Read 0x0: The end of line signal is not asserted at the end of each frame.	RW	0
5	CS_EN	Enables the checksum check for the received payload (long packet only). 0x0: Disabled 0x1: Enabled	RW	0
4	COUNT_UNLOCK	Unlock writes to the COUNT bit field. Write 0x0: COUNT bit field is locked. Writes have no effect Write 0x1: COUNT bit field is unlocked. Writes are possible.	W	0
3	PING_PONG	Indicates whether the PING or PONG destination address (CSI2_CTX_DAT_PING_ADDR or CSI2_CTX_DAT_PONG_ADDR) was used to write the last frame. This bit field toggles after every FEC_NUMBER FEC sync code received for the current context. Read 0x1: PONG buffer Read 0x0: PING buffer	R	1
2	VP_FORCE	Forces sending of the data to both VPORT and OCP. Only applies to formats that existing in two versions: - One sending data to OCP port only - One sending data to VPORT only (tagged with the +VP extension) The format version sending data only to OCP should be chosen. 0x0: Disabled 0x1: Enabled	RW	0
1	LINE_MODULO	Line modulo configuration 0x0: CSI2_CTX_CTRL3.LINE_NUMBER is used once per frame for the generation of the LINE_NUMBER_IRQ. 0x1: CSI2_CTX_CTRL3.LINE_NUMBER is used as a modulo number for the generation of the LINE_NUMBER_IRQ (multiple times the interrupt can be generated for each frame)	RW	0
0	CTX_EN	Enables the context 0x0: Disabled 0x1: Enabled	RW	0

2.6.5.1.15 CSI2_CTX_CTRL2_i

Table 89. CSI2_CTX_CTRL2_i

Address Offset	0x0000 0074 + (0x20 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 1074 + (0x20 * i) 0x5C00 1074 + (0x20 * i)	**Instance**	ISS_CSI2_A_REGS1_CORTEX-M3 ISS_CSI2_A_REGS1_L3
Description	CONTROL REGISTER - Context This register controls the Context. This register is shadowed: modifications are taken into account after the next FSC sync code (except for VIRTUAL_ID and FORMAT fields). The change of VIRTUIAL_ID and FORMAT has to occur only when the context is disabled (CSI2_CTX_CTRL1.CTX_EN).		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:16	FRAME	Frame number received	R	0x0000
15	RESERVED	Reserved	R	0
14:13	USER_DEF_MAPPING	Selects the pixel format of USER_DEFINED in FORMAT 0x0: RAW6 0x1: RAW7 0x2: RAW8 (not valid if FORMAT is USER_DEFINED_8_BIT_DATA_TYPE_x_EXP8 with x from 1 to 8)	RW	0x0
12:11	VIRTUAL_ID	Virtual channel ID 0x0: Virtual Channel ID 0 0x1: Virtual Channel ID 1 0x2: Virtual Channel ID 2 0x3: Virtual Channel ID 3	RW	0x0
10	DPCM_PRED	Selects the DPCM predictor. 0x0: The advanced predictor is used. Not supported for 10- 8- 10 algorithm. Performance limited to 1 pixel/cycle. 0x1: The simple predictor is used.	RW	0
9:0	FORMAT	Data format selection. 0x0: OTHERS (except NULL and BLANKING packets) 0x12: Embedded 8-bit nonimage data (that is, JPEG) 0x18: YUV4:2:0 8 bit 0x19: YUV4:2:0 10 bit 0x1A: YUV4:2:0 8bit legacy 0x1C: YUV4:2:0 8 bit + CSPS 0x1D: YUV4:2:0 10bit + CSPS 0x1E: YUV4:2:2 8 bit 0x1F: YUV4:2:2 10 bit 0x22: RGB565 0x24: RGB888	RW	0x000

Bits	Field Name	Description	Type	Reset
		0x28: RAW6		
		0x29: RAW7		
		0x2A: RAW8		
		0x2B: RAW10		
		0x2C: RAW12		
		0x2D: RAW14		
		0x33: RGB666 + EXP32_24		
		0x40: USER_DEFINED_8_BIT_DATA_TYPE_1		
		0x41: USER_DEFINED_8_BIT_DATA_TYPE_2		
		0x42: USER_DEFINED_8_BIT_DATA_TYPE_3		
		0x43: USER_DEFINED_8_BIT_DATA_TYPE_4		
		0x44: USER_DEFINED_8_BIT_DATA_TYPE_5		
		0x45: USER_DEFINED_8_BIT_DATA_TYPE_6		
		0x46: USER_DEFINED_8_BIT_DATA_TYPE_7		
		0x47: USER_DEFINED_8_BIT_DATA_TYPE_8		
		0x68: RAW6 + EXP8		
		0x69: RAW7 + EXP8		
		0x80: USER_DEFINED_8_BIT_DATA_TYPE_1 + EXP8		
		0x81: USER_DEFINED_8_BIT_DATA_TYPE_2 + EXP8		
		0x82: USER_DEFINED_8_BIT_DATA_TYPE_3 + EXP8		
		0x83: USER_DEFINED_8_BIT_DATA_TYPE_4 + EXP8		
		0x84: USER_DEFINED_8_BIT_DATA_TYPE_5 + EXP8		
		0x85: USER_DEFINED_8_BIT_DATA_TYPE_6 + EXP8		
		0x86: USER_DEFINED_8_BIT_DATA_TYPE_7 + EXP8		
		0x87: USER_DEFINED_8_BIT_DATA_TYPE_8 + EXP8		
		0x9E: YUV4:2:2 8bit + VP		
		0xA0: RGB444 + EXP16		
		0xA1: RGB555 + EXP16		
		0xAB: RAW10 + EXP16		
		0xAC: RAW12 + EXP16		
		0xAD: RAW14 + EXP16		
		0xDE: Same as YUV4:2:2 8bit + VP, but data is sent as 16-bit wide words to video port. Could be used together with the GENERIC and BYTESWAP features.		
		0xE3: RGB666 + EXP32		
		0xE4: RGB888 + EXP32		
		0xE8: RAW6 + DPCM10 + VP		
		0x12A: RAW8 + VP		
		0x12C: RAW12 + VP		
		0x12D: RAW14 + VP		
		0x12F: RAW10 + VP		
		0x140: USER_DEFINED_8_BIT_DATA_TYPE_1_DPCM12_VP		
		0x141: USER_DEFINED_8_BIT_DATA_TYPE_2_DPCM12_VP		
		0x142: USER_DEFINED_8_BIT_DATA_TYPE_3_DPCM12_VP		
		0x143: USER_DEFINED_8_BIT_DATA_TYPE_4_DPCM12_VP		
		0x144: USER_DEFINED_8_BIT_DATA_TYPE_5_DPCM12_VP		

Bits	Field Name	Description	Type	Reset
		0x145: USER_DEFINED_8_BIT_DATA_TYPE_6_DPCM12_VP		
		0x146: USER_DEFINED_8_BIT_DATA_TYPE_7_DPCM12_VP		
		0x147: USER_DEFINED_8_BIT_DATA_TYPE_8_DPCM12_VP		
		0x1C0: USER_DEFINED_8_BIT_DATA_TYPE_1_DPCM12_EXP16		
		0x1C1: USER_DEFINED_8_BIT_DATA_TYPE_2_DPCM12_EXP16		
		0x1C2: USER_DEFINED_8_BIT_DATA_TYPE_3_DPCM12_EXP16		
		0x1C3: USER_DEFINED_8_BIT_DATA_TYPE_4_DPCM12_EXP16		
		0x1C4: USER_DEFINED_8_BIT_DATA_TYPE_5_DPCM12_EXP16		
		0x1C5: USER_DEFINED_8_BIT_DATA_TYPE_6_DPCM12_EXP16		
		0x1C6: USER_DEFINED_8_BIT_DATA_TYPE_7_DPCM12_EXP16		
		0x1C7: USER_DEFINED_8_BIT_DATA_TYPE_8_DPCM12_EXP16		
		0x229: RAW7 + DPCM10 + EXP16		
		0x2A8: RAW6 + DPCM10 + EXP16		
		0x2AA: RAW8 + DPCM10 + EXP16		
		0x2C0: USER_DEFINED_8_BIT_DATA_TYPE_1 + DPCM10 + EXP16		
		0x2C1: USER_DEFINED_8_BIT_DATA_TYPE_2 + DPCM10 + EXP16		
		0x2C2: USER_DEFINED_8_BIT_DATA_TYPE_3 + DPCM10 + EXP16		
		0x2C3: USER_DEFINED_8_BIT_DATA_TYPE_4 + DPCM10 + EXP16		
		0x2C4: USER_DEFINED_8_BIT_DATA_TYPE_5 + DPCM10 + EXP16		
		0x2C5: USER_DEFINED_8_BIT_DATA_TYPE_6 + DPCM10 + EXP16		
		0x2C6: USER_DEFINED_8_BIT_DATA_TYPE_7 + DPCM10 + EXP16		
		0x2C7: USER_DEFINED_8_BIT_DATA_TYPE_8 + DPCM10 + EXP16		
		0x329: RAW7 + DPCM10 + VP		
		0x32A: RAW8 + DPCM10 + VP		
		0x340: USER_DEFINED_8_BIT_DATA_TYPE_1 + DPCM10 + VP		
		0x341: USER_DEFINED_8_BIT_DATA_TYPE_2 + DPCM10 + VP		
		0x342: USER_DEFINED_8_BIT_DATA_TYPE_3 + DPCM10 + VP		
		0x343: USER_DEFINED_8_BIT_DATA_TYPE_4 + DPCM10 + VP		

Bits	Field Name	Description	Type	Reset
		0x344: USER_DEFINED_8_BIT_DATA_TYPE_5 + DPCM10 + VP 0x345: USER_DEFINED_8_BIT_DATA_TYPE_6 + DPCM10 + VP 0x346: USER_DEFINED_8_BIT_DATA_TYPE_7 + DPCM10 + VP 0x347: USER_DEFINED_8_BIT_DATA_TYPE_8 + DPCM10 + VP 0x368: RAW6 DPCM12 + VP 0x369: RAW7 DPCM12 + EXP16 0x36A: RAW8 DPCM12 + EXP16 0x3A8: RAW6 DPCM12 + EXP16 0x3A9: RAW7 DPCM12 + VP 0x3AA: RAW8 DPCM12 + VP		

2.6.5.1.16 CSI2_CTX_DAT_OFST_i

Table 90. CSI2_CTX_DAT_OFST_i

Address Offset	0x0000 0078 + (0x20 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 1078 + (0x20 * i) 0x5C00 1078 + (0x20 * i)	**Instance**	ISS_CSI2_A_REGS1_CORTEX-M3 ISS_CSI2_A_REGS1_L3
Description	DATA MEM ADDRESS OFFSET REGISTER - Context This register sets the offset, which is applied on the destination address after each line is written to memory. This register applies for both CSI2_CTX_DAT_PING_ADDR and CSI2_CTX_DAT_PONG_ADDR. For example, it enables to perform 2D data transfers of the pixel data into a frame buffer. In such case, the pixel data and frame buffer data shall have the same data format. The 5 LSBs are ignored: the offset shall be a multiple of 32 bytes. This register is shadowed: modifications are taken into account after the next FSC sync code. Only full 32-bit values shall be written.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								OFST																RESERVED							

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Reserved	R	0x0000
16:5	OFST	Line offset programmed in bytes (signed value 2s complement). If OFST = 0, the data is written contiguously in memory. Otherwise, OFST sets the destination offset between the first pixel of the previous line and the first pixel of the current line.	RW	0x000
4:0	RESERVED	Reserved	R	0x00

2.6.5.1.17 CSI2_CTX_DAT_PING_ADDR_i

Table 91. CSI2_CTX_DAT_PING_ADDR_i

Address Offset	0x0000 007C + (0x20 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 107C + (0x20 * i) 0x5C00 107C + (0x20 * i)	**Instance**	ISS_CSI2_A_REGS1_CORTEX-M3 ISS_CSI2_A_REGS1_L3
Description	DATA MEM PING ADDRESS REGISTER - Context This register sets the 32-bit memory address where the pixel data are stored. The destination is double buffered: this register sets the PING address. Double buffering is enabled when the addresses CSI2_CTX_DAT_PING_ADDR and CSI2_CTX_DAT_PONG_ADDR are different. The 5 LSBs are ignored: the address shall be aligned on a 32-byte boundary. This register is shadowed: modifications are taken into account after the next FSC sync code. Only full 32-bit values shall be written.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																											RESERVED			

Bits	Field Name	Description	Type	Reset
31:5	ADDR	27 most-significant bits of the 32-bit address.	RW	0x0000000
4:0	RESERVED	Reserved	R	0x00

2.6.5.1.18 CSI2_CTX_DAT_PONG_ADDR_i

Table 92. CSI2_CTX_DAT_PONG_ADDR_i

Address Offset	0x0000 0080 + (0x20 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 1080 + (0x20 * i) 0x5C00 1080 + (0x20 * i)	**Instance**	ISS_CSI2_A_REGS1_CORTEX-M3 ISS_CSI2_A_REGS1_L3
Description	DATA MEM PONG ADDRESS REGISTER - Context This register sets the 32-bit memory address where the pixel data are stored. The destination is double-buffered: this register sets the PONG address. Double-buffering is enabled when the addresses CSI2_CTX_DAT_PING_ADDR and CSI2_CTX_DAT_PONG_ADDR are different. The 5 LSBs are ignored: the address shall be aligned on a 32-byte boundary. This register is shadowed: modifications are taken into account after the next FSC sync code. Only full 32-bit values shall be written.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																											RESERVED			

Bits	Field Name	Description	Type	Reset
31:5	ADDR	27 most-significant bits of the 32-bit address.	RW	0x0000000
4:0	RESERVED	Reserved	R	0x00

2.6.5.1.19 CSI2_CTX_IRQENABLE_i

Table 93. CSI2_CTX_IRQENABLE_i

Address Offset	0x0000 0084 + (0x20 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 1084 + (0x20 * i) 0x5C00 1084 + (0x20 * i)	**Instance**	ISS_CSI2_A_REGS1_CORTEX-M3 ISS_CSI2_A_REGS1_L3
Description	INTERRUPT ENABLE REGISTER - Context This register regroups all the events related to context.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							ECC_CORRECTION_IRQ	LINE_NUMBER_IRQ	FRAME_NUMBER_IRQ	CS_IRQ	RESERVED	LE_IRQ	LS_IRQ	FE_IRQ	FS_IRQ

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8	ECC_CORRECTION_IRQ	Context - ECC has been used to correct the only 1-bit error (long packet only). 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
7	LINE_NUMBER_IRQ	Context - Line number is reached. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
6	FRAME_NUMBER_IRQ	Context - Frame counter reached. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
5	CS_IRQ	Context - Check-Sum of the payload mismatch detection 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
4	RESERVED	Reserved	R	0
3	LE_IRQ	Context - Line end sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
2	LS_IRQ	Context - Line start sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
1	FE_IRQ	Context - Frame end sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
0	FS_IRQ	Context - Frame start sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

2.6.5.1.20 CSI2_CTX_IRQSTATUS_i

Table 94. CSI2_CTX_IRQSTATUS_i

Address Offset	0x0000 0088 + (0x20 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 1088 + (0x20 * i) 0x5C00 1088 + (0x20 * i)	**Instance**	ISS_CSI2_A_REGS1_CORTEX-M3 ISS_CSI2_A_REGS1_L3
Description	INTERRUPT STATUS REGISTER - Context This register regroups all the events related to Context.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							ECC_CORRECTION_IRQ	LINE_NUMBER_IRQ	FRAME_NUMBER_IRQ	CS_IRQ	RESERVED	LE_IRQ	LS_IRQ	FE_IRQ	FS_IRQ

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8	ECC_CORRECTION_IRQ	Context - ECC has been used to do the correction of the only 1-bit error status (long packet only). 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
7	LINE_NUMBER_IRQ	Context - Line number reached status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
6	FRAME_NUMBER_IRQ	Context - Frame counter reached status 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
5	CS_IRQ	Context - Check-Sum mismatch status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
4	RESERVED	Reserved	R	0
3	LE_IRQ	Context - Line end sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
2	LS_IRQ	Context - Line start sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
1	FE_IRQ	Context - Frame end sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
0	FS_IRQ	Context - Frame start sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0

2.6.5.1.21 CSI2_CTX_CTRL3_i

Table 95. CSI2_CTX_CTRL3_i

Address Offset	0x0000 008C + (0x20 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 108C + (0x20 * i) 0x5C00 108C + (0x20 * i)	**Instance**	ISS_CSI2_A_REGS1_CORTEX-M3 ISS_CSI2_A_REGS1_L3
Description	CONTROL REGISTER - Context This register controls the Context. This register is shadowed: modifications are taken into account after the next FSC sync code.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ALPHA								LINE_NUMBER															

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Reserved	R	0x0
29:16	ALPHA	Alpha value for RGB888, RGB666 and RBG444.	RW	0x0000
15:0	LINE_NUMBER	Line number for the interrupt generation	RW	0x0000

2.6.5.2 ISS CSI2 REGS2 Registers

Table 96 summarizes the CSI2 REGS2 registers.

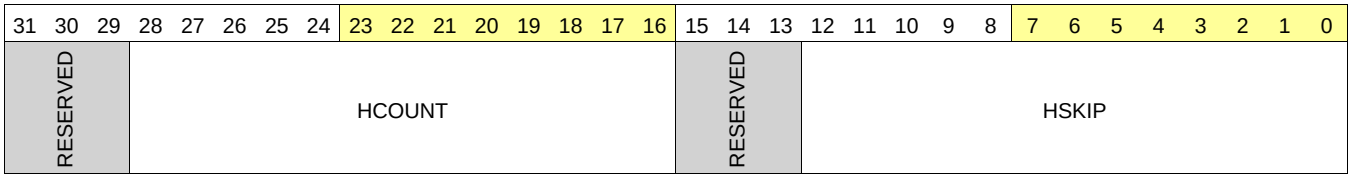
Table 96. ISS CSI2 REGS2 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ISS_CSI2_A_REGS2 Base Address Cortex-M3 Private Access	ISS_CSI2_A_REGS2 Base Address L3 Interconnect
CSI2_CTX_TRANSCODEH_i	RW	32	0x0000 0000 + (0x8 * i)	0x5504 11C0 + (0x8 * i)	0x5C00 11C0 + (0x8 * i)
CSI2_CTX_TRANSCODEV_i	RW	32	0x0000 0004 + (0x8 * i)	0x5504 11C4 + (0x8 * i)	0x5C00 11C4 + (0x8 * i)

2.6.5.2.1 CSI2_CTX_TRANSCODEH_i

Table 97. CSI2_CTX_TRANSCODEH_i

Address Offset	0x0000 0000 + (0x8 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 11C0 + (0x8 * i) 0x5C00 11C0 + (0x8 * i)	**Instance**	ISS_CSI2_A_REGS2_CORTEX-M3 ISS_CSI2_A_REGS2_L3
Description	Transcode configuration register: defines horizontal frame cropping		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Reserved	R	0x0
28:16	HCOUNT	Pixels to output per line when the values is between 1 and 8191. Pixels HSKIP-WIDTH pixels are output when HCOUNT=0. WIDTH corresponds to the image width provided by the sensor.	RW	0x0000
15:13	RESERVED	Reserved	R	0x0
12:0	HSKIP	Pixel to skip horizontally. Valid values: 0-8191	RW	0x0000

2.6.5.2.2 CSI2_CTX_TRANSCODEV_i

Table 98. CSI2_CTX_TRANSCODEV_i

Address Offset	0x0000 0004 + (0x8 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 11C4 + (0x8 * i) 0x5C00 11C4 + (0x8 * i)	**Instance**	ISS_CSI2_A_REGS2_CORTEX-M3 ISS_CSI2_A_REGS2_L3
Description	Transcode configuration register: defines vertical frame cropping		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			VCOUNT													RESERVED			VSKIP												

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Reserved	R	0x0
28:16	VCOUNT	Lines to output per frame when the values is between 1 and 8191. Pixels VSKIP-HEIGHT pixels are output when VCOUNT=0. HEIGHT corresponds to the image height provided by the sensor.	RW	0x0000
15:13	RESERVED	Reserved	R	0x0
12:0	VSKIP	Pixel to skip vertically Valid values: 0-8191	RW	0x0000

2.7 ISS TCTRL

2.7.1 ISS TCTRL Environment

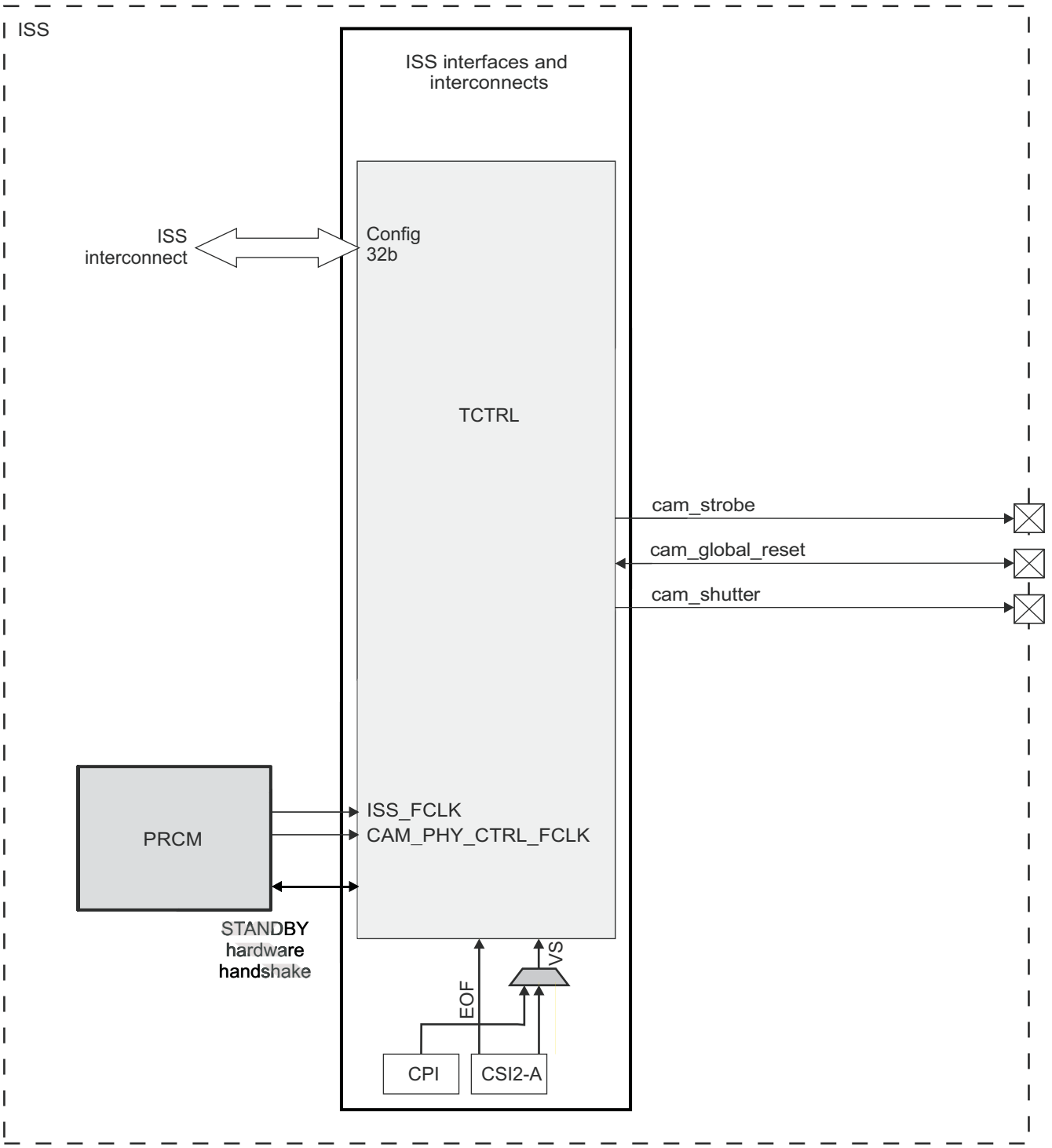
There are no particular environment attributes. See Section 2.2, *ISS Interfaces Environment*.

2.7.2 ISS TCTRL Integration

Figure 74 shows the integration of the TCTRL.

For information about the power domain, clocks, reset, and hardware requests, see Section 1.2.5, *ISS Power Management*.

Figure 74. ISS TCTRL Integration



2.7.3 ISS TCTRL Functional Description

2.7.3.1 ISS TCTRL Features

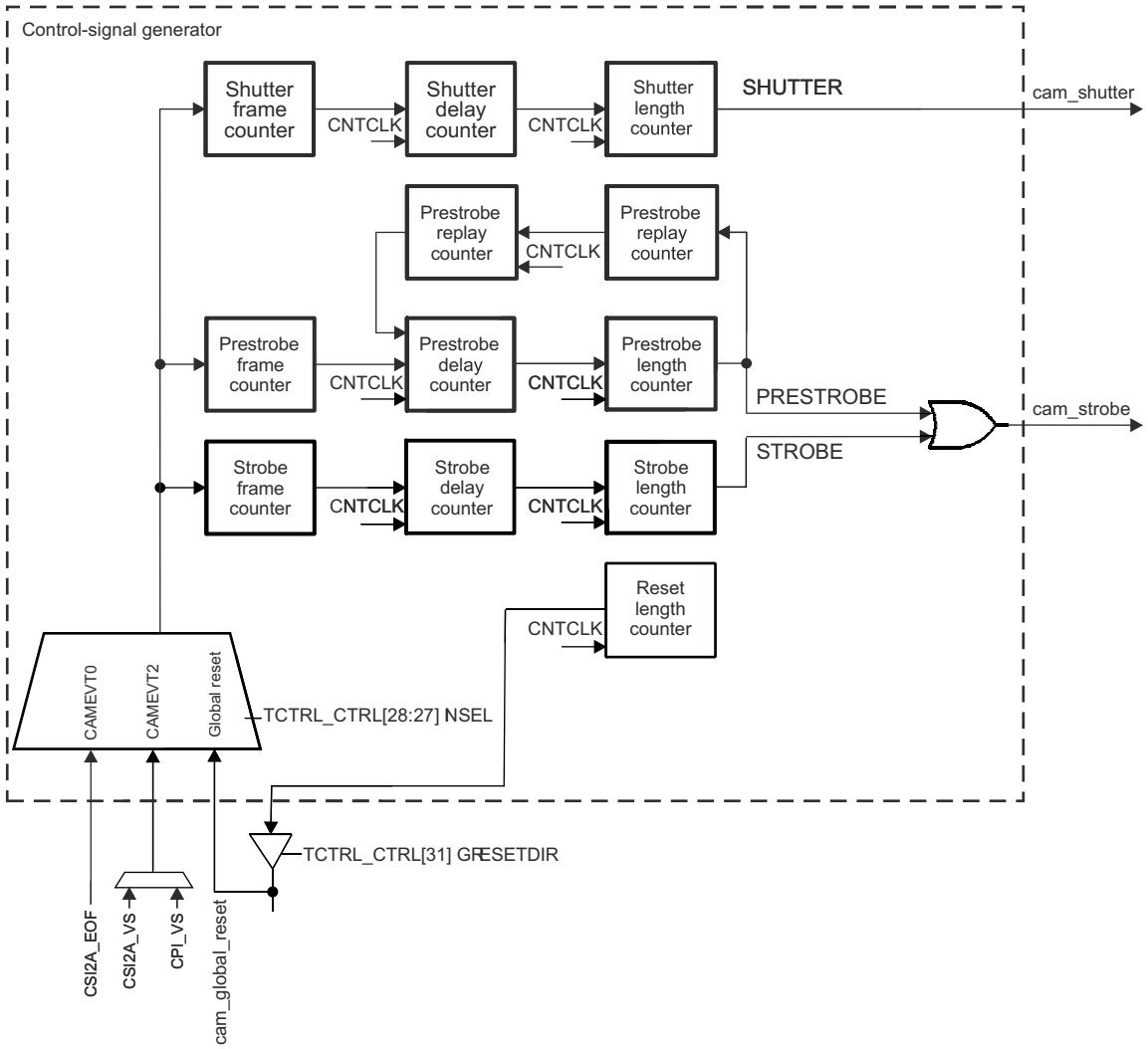
The TCTRL generates the control signals (cam_strobe and cam_shutter) for the flash prestrobe, flash strobe, and mechanical shutters.

The TCTRL includes a timing generator and a control-signal generator.

2.7.3.2 ISS TCTRL Control-Signal Generator

The control-signal generator generates the prestrobe, strobe, and shutter signals: cam_strobe and cam_shutter. Figure 75 shows the principle of control-signal generation.

Figure 75. TCTRL Control-Signal Generation



The control-signal generator gathers precise timings for the cam_strobe and cam_shutter signals, to assert and deassert the signals at known times. The timing control-signal generator can be synchronized on the vertical synchronization signal coming from the CSI2-A, parallel interface (CPI), or on an externally generated cam_global_reset signal.

This multiplexer can also select the externally generated cam_global_reset signal as the trigger event. The TCTRL_CTRL[31] GRESETDIR bit defines the direction of cam_global_reset.

- The externally generated cam_global_reset is used as a trigger when TCTRL_CTRL[31] GRESETDIR = 0 and TCTRL_CTRL[28:27] INSEL = 3.

- The internally generated cam_global_reset is used as a trigger when TCTRL_CTRL[31] GRESETDIR = 1 and TCTRL_CTRL[28:27] INSEL = 3.

The cam_global_reset signal can also be generated internally by the control-signal generator under software control. In this case, the prestrobe and shutter signals are synchronized on internally generated cam_global_reset . The multiplexer controls whether control-signal generation must be triggered by the internal or external cam_global_reset.

The prestrobe-, strobe-, and shutter-control signals can be individually enabled at any time. These signals must not be disabled by software.

The clock divider generates the CNTCLK clock based on the functional clock. The clock divider is programmable. Table 99 lists the possible frequencies as a function of the divisor values.

Table 99. ISS TCTRL Control-Signal Generator: CNTCLK Frequencies

Divisor Value TCTRL_CTRL[18:10] DIVC	CNTCLK Clock	CNTCLK Precision (ns)
0 (default)	Clock gated. No clock.	N/A
1	200 MHz	5
2	100 MHz	10
3	66,667 MHz	15
4	50 MHz	20
...	...	...
510	0.392 MHz	2550
511	0.391 MHz	2555

There are three counters per control signal, for a total of nine counters. Each counter is programmable.

- The frame counter is decreased each time a full new frame is received.
  - A new frame is detected by the TCTRL module when CAMEVTx is received.
  - The frame counter determines how many whole frames must be ignored before the delay counter is triggered. The frame counters can be set to 0 to bypass the frame counters.
- The delay counter determines the control-signal activation delay. The counter is decreased at every CNTCLK clock cycle. When the counter reaches 0, the control signal is asserted. If the delay counter is set to 0, the control signal is asserted immediately.
- The activation-length counter determines the length of control-signal assertion. The counter is decreased at every CNTCLK clock cycle. When the counter reaches 0, the signal is deasserted and the control-signal enable bit is disabled. If the activation length is set to 0, the control signal is not asserted and the control-signal enable bit is disabled.

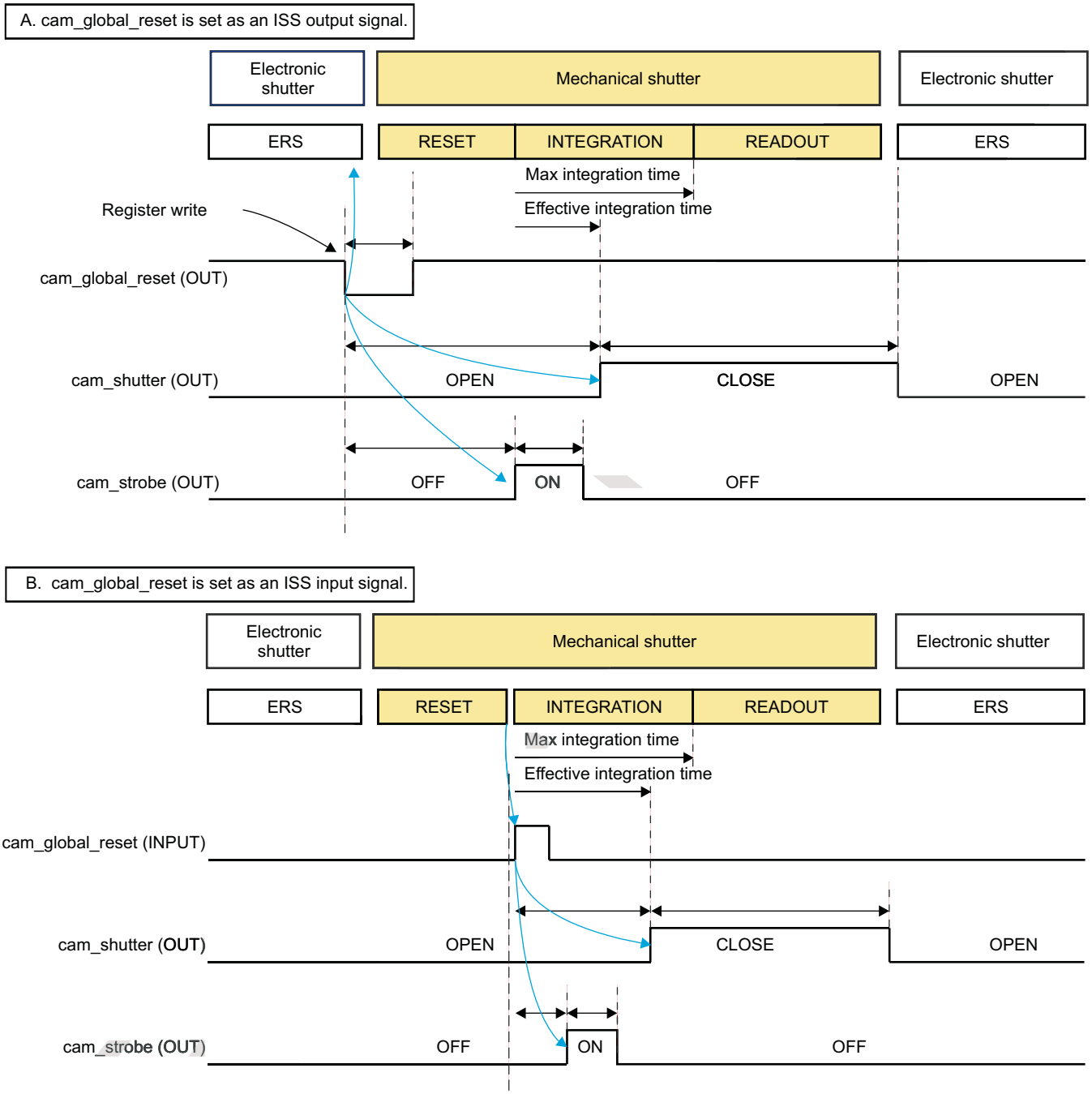
The polarity of the following signals can be individually selected:

- TCTRL_CTRL[26] STRBPSTRBPOL for the prestrobe and strobe signals
- TCTRL_CTRL[24] SHUTPOL for the shutter signal
- TCTRL_CTRL[30] GRESETPOL for cam_global_reset

Software can trigger the generation of cam_global_reset to the camera module. The length of signal-activation is programmable. The counter is decreased at every CNTCLK clock cycle. When the counter reaches 0, the signal is deasserted and the global reset enable bit is disabled (the TCTRL_CTRL[29] GRESETEN bit). If the activation length is set to 0, the control signal is not asserted and the control-signal enable bit is disabled. The polarity of cam_global_reset can be selected (the TCTRL_CTRL[30] GRESETPOL bit).

Figure 76 shows the use of cam_global_reset set as an input or output signal. cam_global_reset is asynchronous, edge-sensitive, and asserted for at least one interconnect clock cycle.

Figure 76. ISS TCTRL Use of cam_global_reset With Global Reset Release Camera Modules



There are two types of shutter mechanisms: mechanical and electronic. A mechanical shutter is used only for high-resolution sensors. The three control signals (cam_global_reset, cam_shutter, and cam_strobe) are useful with a mechanical shutter. High frame rates can be achieved only with an electronic shutter. When an electronic shutter is used, none of the three control signals is used.

- Mechanical shutter mechanism:
  - Reset: All pixels of the sensor are reset to their black value. When the sensor has a global reset feature, the mechanical shutter can be open during reset.
  - Integration: The light received by the sensor is transformed into electrical charges that are stored inside pixels. At the end of the integration time, the shutter must be closed. Exposure time is defined by the time between reset release and shutter close.

- Readout: The charges accumulated in pixels are converted to digital values that are sent to the camera receiver.

- Electronic rolling shutter (ERS) mechanism:
  - Each line of the sensor is reset separately and read after a fixed amount of time. Exposure time is defined by the time between reset and read.

2.7.4 ISS TCTRL Programming Model

The following settings must be done before enabling the TCTRL.

2.7.4.1 ISS TCTRL Camera-Control Signal Generator

Two configurations apply:

- First configuration: The control signals are based on the vertical synchronization information coming from the camera module or from externally generated cam_global_reset.
- Second configuration: The control signals are based on internally generated cam_global_reset.

2.7.4.1.1 ISS TCTRL Vertical Sync-Based Control-Signal Generation or Externally Generated cam_global_reset

To enable control-signal generation in the first configuration, follow the procedure listed in Table 100.

Table 100. ISS TCTRL Enabling the Control-Signal Generation in First Configuration

Step	Bit Field	Value
Select the input that triggers the control signals. The trigger signal can come from the CSI2-A, Parallel interface (CPI) or the externally generated cam_global_reset signal.	TCTRL_CTRL[28:27] INSEL	0x0: CSI2A 0x2: VS signal 0x3: Global reset
Set the global reset as input signal. Writes to the TCTRL_CTRL[29] GRESETEN bit do not trigger the CAM_STROBE and CAM_SHUTTER signals and do not generate the CAM_GLOBAL_RESET signal.	TCTRL_CTRL[31] GRESETDIR	0x0
The following bits are cleared automatically to 0 after the signal assertion:	• TCTRL_CTRL[21] SHUTEN • TCTRL_CTRL[22] PSTRBEN • TCTRL_CTRL[23] STRBEN	
Set the polarity of the SHUTTER, STROBE/PRESTROBE, and cam_global_reset signals.	• TCTRL_CTRL[24] SHUTPOL • TCTRL_CTRL[26] STRBPSTRBPOL • TCTRL_CTRL[30] GRESETPOL	
Set the clock divisor value, which generates the CNTCLK clock The clock is set by CNTCLK = ISS_FCLK/TCTRL_CTRL[18:10] DIVC.	TCTRL_CTRL[18:10] DIVC	0x0: Disable CNTCLK 0 to 511: Divider
Set the frame counters.	• TCTRL_FRAME[5:0] SHUT • TCTRL_FRAME[11:6] PSTRB • TCTRL_FRAME[17:12] STRB	0: TCTRL does not delay any frame in input. 1 to 63
Set the delay counters.	• TCTRL_SHUT_DELAY • TCTRL_PSTRB_DELAY • TCTRL_STRB_DELAY	The possible values are 0 to 2^{25} 1 cycles. The cycles are at the CNTCLK clock frequency. The maximum signal duration is (2^{25} 1) x 511/200 MHz = 85.73157376 s (TCTRL_CTRL[18:10] DIVC = 511).

Table 100. ISS TCTRL Enabling the Control-Signal Generation in First Configuration (continued)

Step	Bit Field	Value
Set the signal durations.	• TCTRL_SHUT_LENGTH • TCTRL_PSTRB_LENGTH • TCTRL_STRB_LENGTH	The possible values are 0 to 2^{24} 1 cycles. The cycles are at the CNTCLK clock frequency. The maximum signal duration for a 200 MHz input clock is (2^{24} 1) x 511/200 MHz = 42.865784325 s (TCTRL_CTRL.DIVC = 511).
Enable the SHUTTER signal.	TCTRL_CTRL[21] SHUTEN	0x1
Enable the PRESTROBE signal.	TCTRL_CTRL[22] PSTRBEN	0x1
Enable the STROBE signal.	TCTRL_CTRL[23] STRBEN	0x1

2.7.4.1.2 ISS TCTRL Internally Generated cam_global_reset-Based Control-Signal Generation

To enable control-signal generation in the second configuration, follow the procedure listed in Table 101.

Table 101. ISS TCTRL Enabling the Control-Signal Generation in Second Configuration

Step	Bit Field	Value
Select the input to global reset, to loop back the internally generated GLOBAL_RESET. Vertical synchronization events do not trigger the CAM_STROBE and CAM_SHUTTER signals.	TCTRL_CTRL[28:27] INSEL	0x3: Global reset
Set the global reset as output signal.	TCTRL_CTRL[31] GRESETDIR	0x1
The following bits are cleared automatically to 0 after the signal assertion:	• TCTRL_CTRL[21] SHUTEN • TCTRL_CTRL[22] PSTRBEN • TCTRL_CTRL[23] STRBEN • TCTRL_CTRL[29] GRESETEN	0x0
Set the polarity of the SHUTTER, STROBE/PRESTROBE, and cam_global_reset signals.	• TCTRL_CTRL[24] SHUTPOL • TCTRL_CTRL[26] STRBPSTRBPOL • TCTRL_CTRL[30] GRESETPOL	
Set the clock divisor value, which generates the CNTCLK clock The clock is set by CNTCLK = ISS_FCLK/TCTRL_CTRL[18:10] DIVC.	TCTRL_CTRL[18:10] DIVC	0x0: Disable CNTCLK 0 to 511: Divider
Set the frame counters.	• TCTRL_FRAME[5:0] SHUT • TCTRL_FRAME[11:6] PSTRB • TCTRL_FRAME[17:12] STRB	0: TCTRL does not delay any frame in input. 1 to 63
Set the delay counters.	• TCTRL_SHUT_DELAY • TCTRL_PSTRB_DELAY • TCTRL_STRB_DELAY	The possible values are 0 to 2^{25} 1 cycles. The cycles are at the CNTCLK clock frequency. The maximum signal duration is (2^{25} 1) x 511/200 MHz = 85.73157376 s (TCTRL_CTRL[18:10] DIVC = 511).

Table 101. ISS TCTRL Enabling the Control-Signal Generation in Second Configuration (continued)

Step	Bit Field	Value
Set the signal durations.	• TCTRL_SHUT_LENGTH • TCTRL_PSTRB_LENGTH • TCTRL_STRB_LENGTH	The possible values are 0 to 2^{24} 1 cycles. The cycles are at the CNTCLK clock frequency. The maximum signal duration for a 200 MHz input clock is (2^{24} 1) x 511/200 MHz = 42.865784325 s (TCTRL_CTRL.DIVC = 511).
Set the cam_global_reset assertion time.	TCTRL_GRESET_LENGTH	The possible values are 0 to 2^{24} 1 cycles. The cycles are at the CNTCLK clock frequency. The maximum signal duration for a 200 MHz input clock is (2^{24} 1) x 511/200 MHz = 42.865784325 s (TCTRL_CTRL.DIVC = 511).
Enable the SHUTTER signal.	TCTRL_CTRL[21] SHUTEN	0x1
Enable the PRESTROBE signal.	TCTRL_CTRL[22] PSTRBEN	0x1
Enable the STROBE signal.	TCTRL_CTRL[23] STRBEN	0x1
Enable the cam_global_reset control-signal generation.	TCTRL_CTRL[29] GRESETEN	0x1

NOTE: Setting the following bits to 1 simultaneously leads to unpredictable behavior:

- TCTRL_CTRL[21] SHUTEN
- TCTRL_CTRL[22] PSTRBEN
- TCTRL_CTRL[23] STRBEN
- TCTRL_CTRL [29] GRESETEN

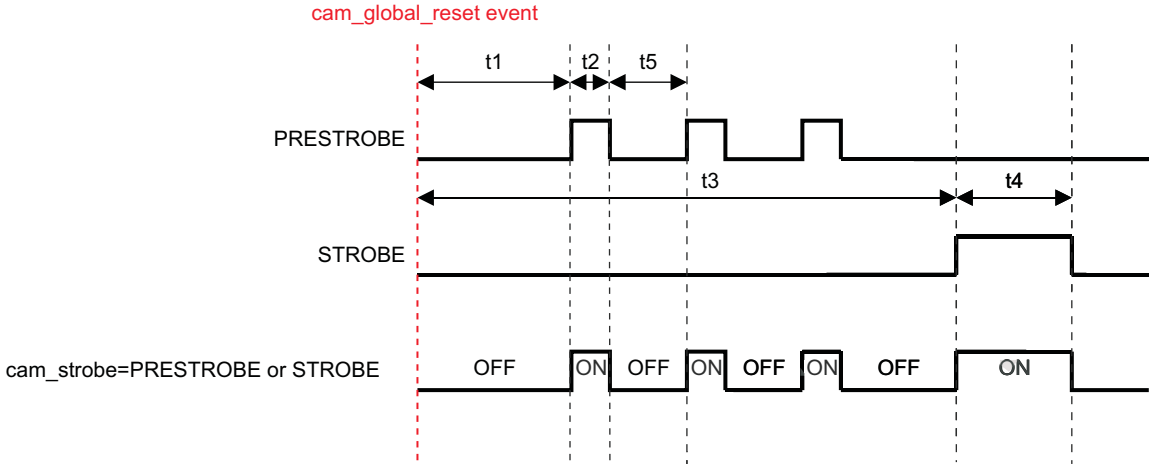
The following bits must be set before TCTRL_CTRL[29] GRESETEN is enabled:

- TCTRL_CTRL[21] SHUTEN
- TCTRL_CTRL[22] PSTRBEN
- TCTRL_CTRL[23] STRBEN

2.7.4.1.3 ISS TCTRL STROBE and PRESTROBE Signal Generation for Red-Eye Removal

The STROBE and PRESTROBE signal generation enables a strobe flash for red-eye removal. The process is shown in Figure 77. The dotted line corresponds to known timings from which the delay counters start decreasing: cam_global_reset event.

Figure 77. cam_strobe Signal-Generation for Red-Eye Removal



- t1: Set by the TCTRL_PSTRB_DELAY register
- t2: Set by the TCTRL_PSTRB_LENGTH register
- t5: Set by the TCTRL_PSTRB_REPLAY[24:0] DELAY bit field. The number of times the pulse is repeated is controlled by the TCTRL_PSTRB_REPLAY[31:25] COUNTER bit field.

  In the previous example, TCTRL_PSTRB_REPLAY[31:25] COUNTER = 2.

  – The possible delay values are 0 to 2^{25} 1 cycle. The cycles are at the CNTCLK clock frequency. The maximum signal duration is (2^{25} 1) x 511/200 MHz = 85.73157376 s (TCTRL_CTRL[18:11] DIVC = 511).

  – The possible count values are 0 to 127 additional pulses.
- t3: Set by the TCTRL_STRB_DELAY register
- t4: Set by the TCTRL_STRB_LENGTH register

2.7.5 ISS TCTRL Registers

Table 102 lists the TCTRL instance.

Table 102. ISS TCTRL Instance Summary

Module Name	Base Address Cortex-M3 Private Access	Base Address L3 Interconnect	Size
ISS_TCTRL	0x5504 0400	0x5C00 0400	256 bytes

Table 103 summarizes the TCTRL registers.

Table 103. ISS TCTRL Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ISS_TCTRL Base Address Cortex-M3 Private Access	ISS_TCTRL Base Address L3 Interconnect
TCTRL_REVISION	R	32	0x0000 0000	0x5504 0400	0x5C00 0400
TCTRL_SYSCONFIG	RW	32	0x0000 0004	0x5504 0404	0x5C00 0404
TCTRL_SYSSTATUS	R	32	0x0000 0008	0x5504 0408	0x5C00 0408
TCTRL_STRB_LENGTH	RW	32	0x0000 0010	0x5504 0410	0x5C00 0410
TCTRL_PSTRB_LENGTH	RW	32	0x0000 0014	0x5504 0414	0x5C00 0414
TCTRL_SHUT_LENGTH	RW	32	0x0000 0018	0x5504 0418	0x5C00 0418
TCTRL_GRESET_LENGTH	RW	32	0x0000 001C	0x5504 041C	0x5C00 041C

Table 103. ISS TCTRL Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	ISS_TCTRL Base Address Cortex-M3 Private Access	ISS_TCTRL Base Address L3 Interconnect
TCTRL_STRB_DELAY	RW	32	0x0000 0020	0x5504 0420	0x5C00 0420
TCTRL_PSTRB_DELAY	RW	32	0x0000 0024	0x5504 0424	0x5C00 0424
TCTRL_SHUT_DELAY	RW	32	0x0000 0028	0x5504 0428	0x5C00 0428
TCTRL_CTRL	RW	32	0x0000 0030	0x5504 0430	0x5C00 0430
TCTRL_PSTRB_REPLAY	RW	32	0x0000 0034	0x5504 0434	0x5C00 0434
TCTRL_FRAME	RW	32	0x0000 0038	0x5504 0438	0x5C00 0438

2.7.5.1 TCTRL_REVISION

Table 104. TCTRL_REVISION

Address Offset	0x0000 0000		
Physical Address	0x5504 0400 0x5C00 0400	**Instance**	ISS_TCTRL_CORTEX-M3 ISS_TCTRL_L3
Description	IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	See(1)

(1) TI internal data

2.7.5.2 TCTRL_SYSCONFIG

Table 105. TCTRL_SYSCONFIG

Address Offset	0x0000 0004		
Physical Address	0x5504 0404 0x5C00 0404	**Instance**	ISS_TCTRL_CORTEX-M3 ISS_TCTRL_L3
Description	OCP-SOCKET SYSTEM CONFIGURATION REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														SOFT_RESET	AUTO_IDLE

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1	SOFT_RESET	Software reset. Set the bit to 1 to trigger the module reset. The bit is automatically reset be the hardware. During reads return 0. 0x0: Normal mode. 0x1: The module is reset.	RW	0
0	AUTO_IDLE	Internal OCP and functional clock gating strategy 0x0: OCP and functional clocks are free-running 0x1: Automatic clock gating strategy is applied, based on the OCP interface activity for interface clock and on the functional activity for functional clocks.	RW	1

2.7.5.3 TCTRL_SYSSTATUS

Table 106. TCTRL_SYSSTATUS

Address Offset	0x0000 0008		
Physical Address	0x5504 0408 0x5C00 0408	**Instance**	ISS_TCTRL_CORTEX-M3 ISS_TCTRL_L3
Description	OCP-SOCKET SYSTEM STATUS REGISTER		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															RESET_DONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	RESET_DONE	Internal reset monitoring Read 0x1: Reset completed. Read 0x0: Internal module reset is ongoing.	R	0

2.7.5.4 TCTRL_STRB_LENGTH

Table 107. TCTRL_STRB_LENGTH

Address Offset	0x0000 0010		
Physical Address	0x5504 0410 0x5C00 0410	**Instance**	ISS_TCTRL_CORTEX-M3 ISS_TCTRL_L3
Description	TIMING CONTROL - STROBE LENGTH REGISTER This register is used by the TIMING CTRL module to generate the STROBE signal.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LENGTH																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x00
23:0	LENGTH	Sets the length of the CAM_STROBE signal assertion in cycles of the CNTCLK clock. The CNTCLK frequency is generated with the TCTRL_CTRL.DIVC bit field. After signal assertion, the TCTRL_CTRL.STRBEN bit is automatically cleared. The possible values are 0 to 2^{24}-1 cycles.	RW	0x000000

2.7.5.5 TCTRL_PSTRB_LENGTH

Table 108. TCTRL_PSTRB_LENGTH

Address Offset	0x0000 0014		
Physical Address	0x5504 0414 0x5C00 0414	**Instance**	ISS_TCTRL_CORTEX-M3 ISS_TCTRL_L3
Description	TIMING CONTROL - PRESTROBE LENGTH REGISTER This register is used by the TIMING CTRL module to generate the PRESTROBE signal.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LENGTH																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x00
23:0	LENGTH	Sets the length of the CAM_PRESTROBE signal assertion in cycles of the CNTCLK clock. The CNTCLK frequency is generated with the TCTRL_CTRL.DIVC bit field. After signal assertion, the TCTRL_CTRL.PSTRBEN bit is automatically cleared. The possible values are 0 to 2^{24}-1 cycles.	RW	0x000000

2.7.5.6 TCTRL_SHUT_LENGTH

Table 109. TCTRL_SHUT_LENGTH

Address Offset	0x0000 0018		
Physical Address	0x5504 0418 0x5C00 0418	**Instance**	ISS_TCTRL_CORTEX-M3 ISS_TCTRL_L3
Description	TIMING CONTROL - SHUTTER LENGTH REGISTER This register is used by the TIMING CTRL module to generate the SHUTTER signal.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LENGTH																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x00
23:0	LENGTH	Sets the length of the CAM_SHUTTER signal assertion in cycles of the CNTCLK clock. The CNTCLK frequency is generated with the TCTRL_CTRL.DIVC bit field. After signal assertion, the TCTRL_CTRL.SHUTEN bit is automatically cleared. The possible values are 0 to 2^{24}-1 cycles.	RW	0x000000

2.7.5.7 TCTRL_GRESET_LENGTH

Table 110. TCTRL_GRESET_LENGTH

Address Offset	0x0000 001C		
Physical Address	0x5504 041C 0x5C00 041C	**Instance**	ISS_TCTRL_CORTEX-M3 ISS_TCTRL_L3
Description	TIMING CONTROL - GLOBAL SHUTTER LENGTH REGISTER This register is used by the TIMING CTRL module to generate the CAM.GRESET signal.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LENGTH																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x00
23:0	LENGTH	Sets the length of the CAM_GLOBAL_RESET signal assertion in cycles of the CNTCLK clock. The CNTCLK frequency is generated with the TCTRL_CTRL.DIVC bit field. After signal assertion, the TCTRL_CTRL.GRESETEN bit is automatically cleared. The possible values are 0 to 2^{24}-1 cycles. The polarity of the CAM_GLOBAL_RESET signal is set by the TCTRL_CTRL.GRESETPOL bit.	RW	0x000000

2.7.5.8 TCTRL_STRB_DELAY

Table 111. TCTRL_STRB_DELAY

Address Offset	0x0000 0020		
Physical Address	0x5504 0420 0x5C00 0420	**Instance**	ISS_TCTRL_CORTEX-M3 ISS_TCTRL_L3
Description	TIMING CONTROL - STROBE DELAY REGISTER This register is used by the TIMING CTRL module to generate the STROBE signal.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DELAY																							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Reserved	R	0x00
24:0	DELAY	Sets the delay for the CAM_STROBE signal assertion in cycles of the CNTCLK clock. The CNTCLK frequency is generated with the TCTRL_CTRL.DIVC bit field. The possible values are 0 to 2^{25}-1 cycles.	RW	0x0000000

2.7.5.9 TCTRL_PSTRB_DELAY

Table 112. TCTRL_PSTRB_DELAY

Address Offset	0x0000 0024		
Physical Address	0x5504 0424 0x5C00 0424	**Instance**	ISS_TCTRL_CORTEX-M3 ISS_TCTRL_L3
Description	TIMING CONTROL - PRE STROBE DELAY REGISTER This register is used by the TIMING CTRL module to generate the PRESTROBE signal.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DELAY																							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Reserved	R	0x00
24:0	DELAY	Sets the delay for the CAM_PSTROBE signal assertion in cycles of the CNTCLK clock. The CNTCLK frequency is generated with the TCTRL_CTRL.DIVC bit field. The possible values are 0 to 2^{25}-1 cycles.	RW	0x0000000

2.7.5.10 TCTRL_SHUT_DELAY

Table 113. TCTRL_SHUT_DELAY

Address Offset	0x0000 0028		
Physical Address	0x5504 0428 0x5C00 0428	**Instance**	ISS_TCTRL_CORTEX-M3 ISS_TCTRL_L3
Description	TIMING CONTROL - SHUTTER DELAY REGISTER This register is used by the TIMING CTRL module to generate the SHUTTER signal.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DELAY																							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Reserved	R	0x00
24:0	DELAY	Sets the delay for the CAM_SHUTTER signal assertion in cycles of the CNTCLK clock. The CNTCLK frequency is generated with the TCTRL_CTRL.DIVC bit field. The possible values are 0 to 2^{25}-1 cycles.	RW	0x0000000

2.7.5.11 TCTRL_CTRL

Table 114. TCTRL_CTRL

Address Offset	0x0000 0030		
Physical Address	0x5504 0430 0x5C00 0430	**Instance**	ISS_TCTRL_CORTEX-M3 ISS_TCTRL_L3
Description	TIMING CONTROL - CONTROL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GRESETDIR	GRESETPOL	GRESETEN	INSEL		STRBPSTRBPOL	RESERVED	SHUTPOL	STRBEN	PSTRBEN	SHUTEN	RESERVED					DIVC								RESERVED							

Bits	Field Name	Description	Type	Reset
31	GRESETDIR	Sets the direction of the GLOBAL_RESET signal. 0x0: INPUT – GLOBAL_RESET is an input to the TIMING CONTROL module. GLOBAL_RESET is externally generated. 0x1: OUTPUT – GLOBAL_RESET is an output of the TIMING CONTROL module. GLOBAL_RESET is internally generated. If GRESETEN is set to 1, the internally generated GLOBAL_RESET will trigger the generation of the PRESTROBE, STROBE and SHUTTER signals. The frame counters are ignored.	RW	0
30	GRESETPOL	Sets the polarity of the global reset signal: CAM_GLOBAL_RESET. It applies whatever the direction of the GLOBAL_RESET signal: input or output. 0x0: active high 0x1: active low	RW	0

Bits	Field Name	Description	Type	Reset
29	GRESETEN	Triggers the generation of the CAM_GLOBAL_RESET signal. The signal is asserted immediately. If enabled, the CAM_GLOBAL_RESET signal will be asserted for TCTRL_GRESET_LENGTH cycles. After the signal assertion, the enable bit is automatically cleared to 0. The polarity of the GLOBAL_RESET signal is set with TCTRL_CTRL.GRESETPOL. Enabling this bit triggers the generation of the CAM_SHUTTER and CAM_STROBE signals (if previously enabled). The frame counters shall be set to 0 when this bit is set to 1 and GRESETDIR is set a OUTPUT.	RW	0
28:27	INSEL	Sets the mode that will trigger the SHUTTER, PRESTROBE and STROBE signals. 0x0: Synchronization event from camera 0 0x1: Synchronization event from camera 1 0x3: GRESET – The CAM_GLOBAL_RESET input signal will trigger the SHUTTER, PRESTROBE and STROBE signals. In this mode, there are no frame counters. The delay counters start decrementing as soon as the GLOBAL_RESET signal is asserted. The polarity of the GLOBAL_RESET signal is set with TCTRL_CTRL.GRESETPOL. 0x2: Synchronization event from camera 2 (serial interfaces muxed with the camera Parallel interface (CPI))	RW	0x0
26	STRBPSTRBPOL	Sets the polarity of the strobe and prestrobe signals. 0x0: Active high 0x1: Active low	RW	0
25	RESERVED	Reserved	R	0
24	SHUTPOL	Sets the polarity of the mechanical shutter signal: CAM_SHUTTER 0x0: Active high 0x1: Active low	RW	0
23	STRBEN	Flash strobe signal enable. If enabled, the STROBE signal will be asserted after TCTRL_FRAME.STRB frames have been received and a delay of TCTRL_STRB_DELAY cycles have passed. The STROBE signal is asserted for TCTRL_STRB_LENGTH cycles. After the signal assertion, the enable bit is automatically cleared to 0. This signal shall not be disabled by software.	RW	0
22	PSTRBEN	Flash prestrobe signal enable. If enabled, the PRESTROBE signal will be asserted after TCTRL_FRAME.PSTRB frames have been received and a delay of TCTRL_PSTRB_DELAY cycles have passed. The PRESTROBE signal is asserted for TCTRL_PSTRB_LENGTH cycles. After the signal assertion, the enable bit is automatically cleared to 0. This signal shall not be disabled by software.	RW	0
21	SHUTEN	Mechanical shutter signal enable. If enabled, the SHUTTER signal will be asserted after TCTRL_FRAME.SHUT frames have been received and a delay of TCTRL_SHUT_DELAY cycles have passed. The SHUTTER signal is asserted for TCTRL_SHUT_LENGTH cycles. After the signal assertion, the enable bit is automatically cleared to 0. This signal shall not be disabled by software.	RW	0
20:19	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
18:10	DIVC	Sets the clock divisor value for the CNTCLK clock generation based on the CLK input clock. CNTCLK is an internal clock used by the TIMING CTRL module counters. Usually, CNTCLK = CLK / DIVC, except for some particular values shown hereafter. 0x0: No clock. CNTCLK is gated.	RW	0x000
9:0	RESERVED	Reserved	R	0x000

2.7.5.12 TCTRL_PSTRB_REPLAY

Table 115. TCTRL_PSTRB_REPLAY

Address Offset	0x0000 0034		
Physical Address	0x5504 0434 0x5C00 0434	**Instance**	ISS_TCTRL_CORTEX-M3 ISS_TCTRL_L3
Description	TIMING CONTROL - PRESTROBE REPLAY REGISTER This register is used by the TIMING CTRL module to generate the prestrobe signal.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER								DELAY																							

Bits	Field Name	Description	Type	Reset
31:25	COUNTER	Sets the number of PRESTROBE pulses after the original pulse. If this bit is set to 0, the PRESTROBE signal behavior is only controlled by TCTRL_FRAME.STRB, TCTRL_PSTRB_DELAY, and TCTRL_PSTRB_LENGTH. If TCTRL_PSTRB_LENGTH=0, there is no replay. This bit is useful when one wants to enable red-eye removal.	RW	0x00
24:0	DELAY	Sets the delay for the PRESTROBE signal reassertion in cycles of the CNTCLK clock. The CNTCLK frequency is generated with the TCTRL_CTRL.DIVC bit field. The possible values are 0 to 2^{25}-1 cycles. If TCTRL_PSTRB_LENGTH=0, there is no replay. This bit field shall not be set to 0 if the COUNTER is set to a value different of 0. This bit is useful when one wants to enable red-eye removal.	RW	0x0000000

2.7.5.13 TCTRL_FRAME

Table 116. TCTRL_FRAME

Address Offset	0x0000 0038		
Physical Address	0x5504 0438 0x5C00 0438	**Instance**	ISS_TCTRL_CORTEX-M3 ISS_TCTRL_L3
Description	TIMING CONTROL - FRAME REGISTER This register is used by the TIMING CTRL module to generate the SHUTTER, PRESTROBE, and STROBE signals.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STRB								PSTRB								SHUT							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	R	0x0000
17:12	STRB	Frame counter for the STROBE signal generation. From 0 to 63 frames. This bit field is ignored if TCTRL.INSEL=GRESET.	RW	0x00
11:6	PSTRB	Frame counter for the PRESTROBE signal generation. From 0 to 63 frames. This bit field is ignored if TCTRL.INSEL=GRESET.	RW	0x00
5:0	SHUT	Frame counter for the SHUTTER signal generation. From 0 to 63 frames. This bit field is ignored if TCTRL.INSEL=GRESET.	RW	0x00

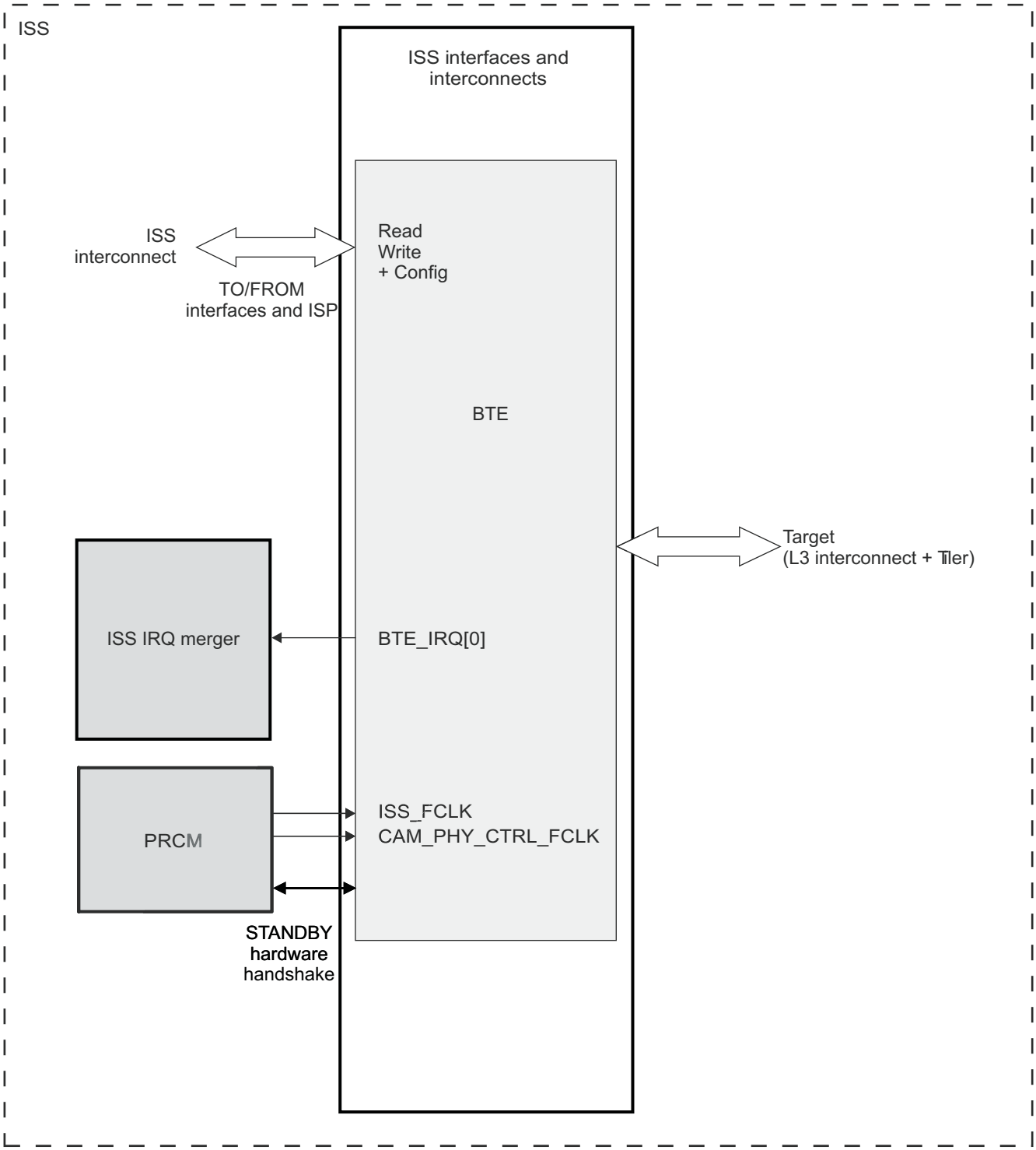
2.8 ISS BTE

2.8.1 ISS BTE Environment

There are no particular environment attributes (see Section 2.2, *ISS Interfaces Environment*).

2.8.2 ISS BTE Integration

Figure 78. ISS BTE Integration



For power domain, clocks, reset, and hardware requests, see Section 1.2.5, *ISS Power Management*.

2.8.2.1 ISS BTE PRCM Interface

2.8.2.1.1 ISS BTE PRCM Handshake

The BTE supports the IDLE protocol to flush outstanding transactions. When an IDLE request is received, the BTE:

- Completes ongoing requests on OCPI (request, data and response phases) and stalls the port (SCmdAccept = 0)
- Flushes all contexts (same behavior than when the BTE_CONTEXT_CTRL_i[2] FLUSH bit is set by software) when the BTE_CONTEXT_CTRL_i[11] AUTOFLUSH bit is set. Otherwise, no context flushing is triggered by an IDLE request.
- Completes ongoing requests on OCPO (request, data and response phases)
- Acknowledges the IDLE request.

2.8.3 ISS BTE Functional Description

2.8.3.1 ISS BTE Features

The BTE increases access efficiency of raster initiators to tiled SDRAM. In fact, the TILER expects 2D-bursts corresponding to a row or column of subtiles for maximal efficiency.

The BTE is connected between one or multiple raster initiators. It can translate reads and writes. For reads, BTE prefetches sufficient data from tiled memory to translate raster requests. For writes, BTE buffers raster requests until is has sufficient data to generate requests to tiled memory. The features of the BTE are:

- Interfaces:
  - 32-bit-wide configuration interface (OCPC)
  - 128-bit-wide slave data port (OCPI)
  - 128-bit-wide master data port (OCPO)
- Incrementing to 2D burst translation for read and writes:
  - Four contexts. A context is a virtual frame buffer attached to a data flow requiring translation.
  - One-shot and continuous mode
- Local memories for temporal storage:
  - Cannot use external memories for temporal data storage
- Transparent for accesses that do not require translation. Requests are forwarded from OCPI to OCPO without modification.
- Local buffer

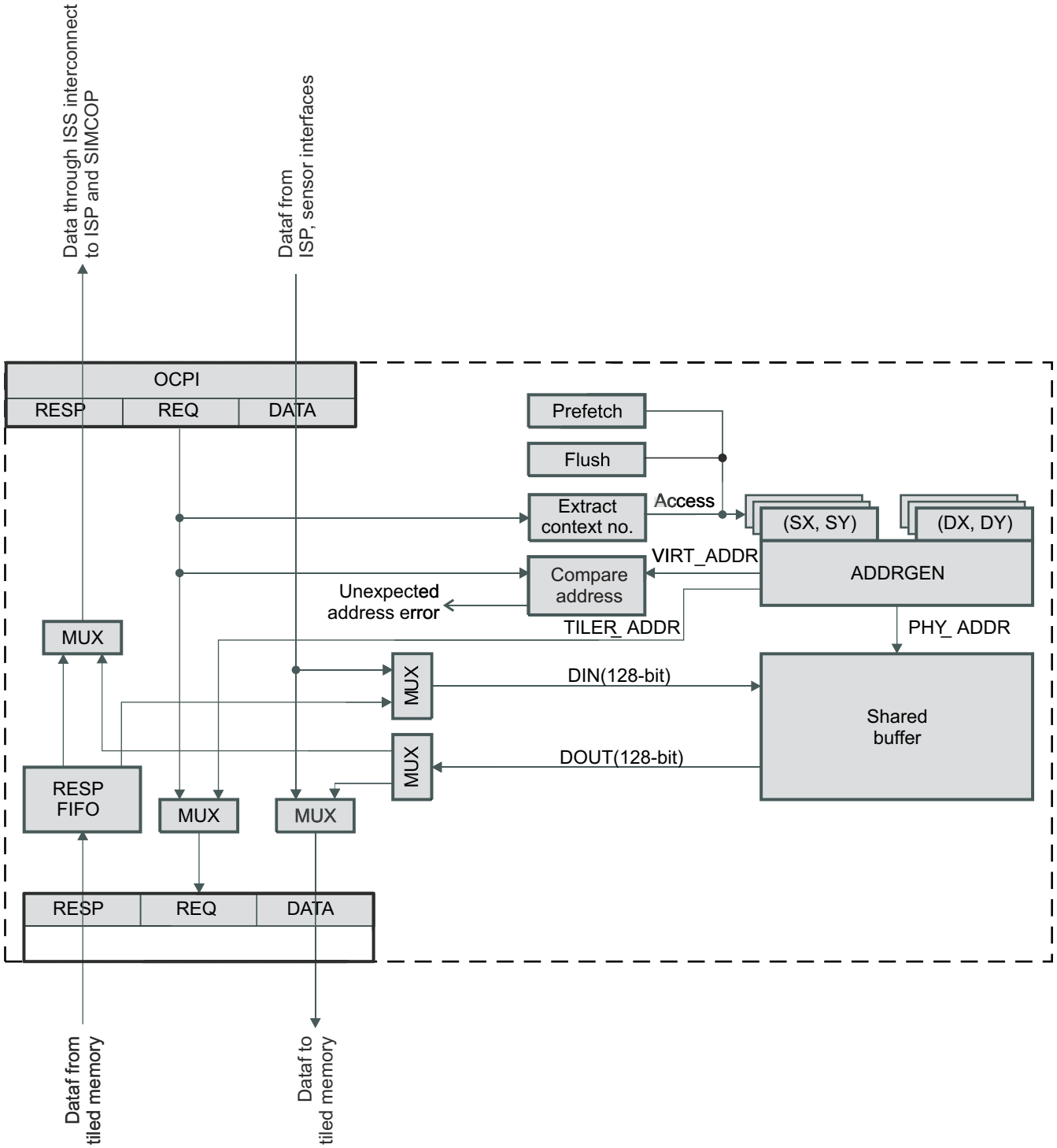
2.8.3.2 ISS BTE Functional Description Details

The main tasks and capabilities of the BTE are:

- Forward OCP transactions that do not need translation.
- For OCP writes requiring translation:
  - Store data received from OCPI to local buffers.
  - Read data from local buffers and send it to OCPO.
- For OCP read requiring translation:
  - Store data received from OCPO to local buffers.
  - Read data from local buffers and send it to OCPI.
- BTE DMA capability

Figure 79 is a logical overview of the BTE.

Figure 79. ISS BTE Logical Overview

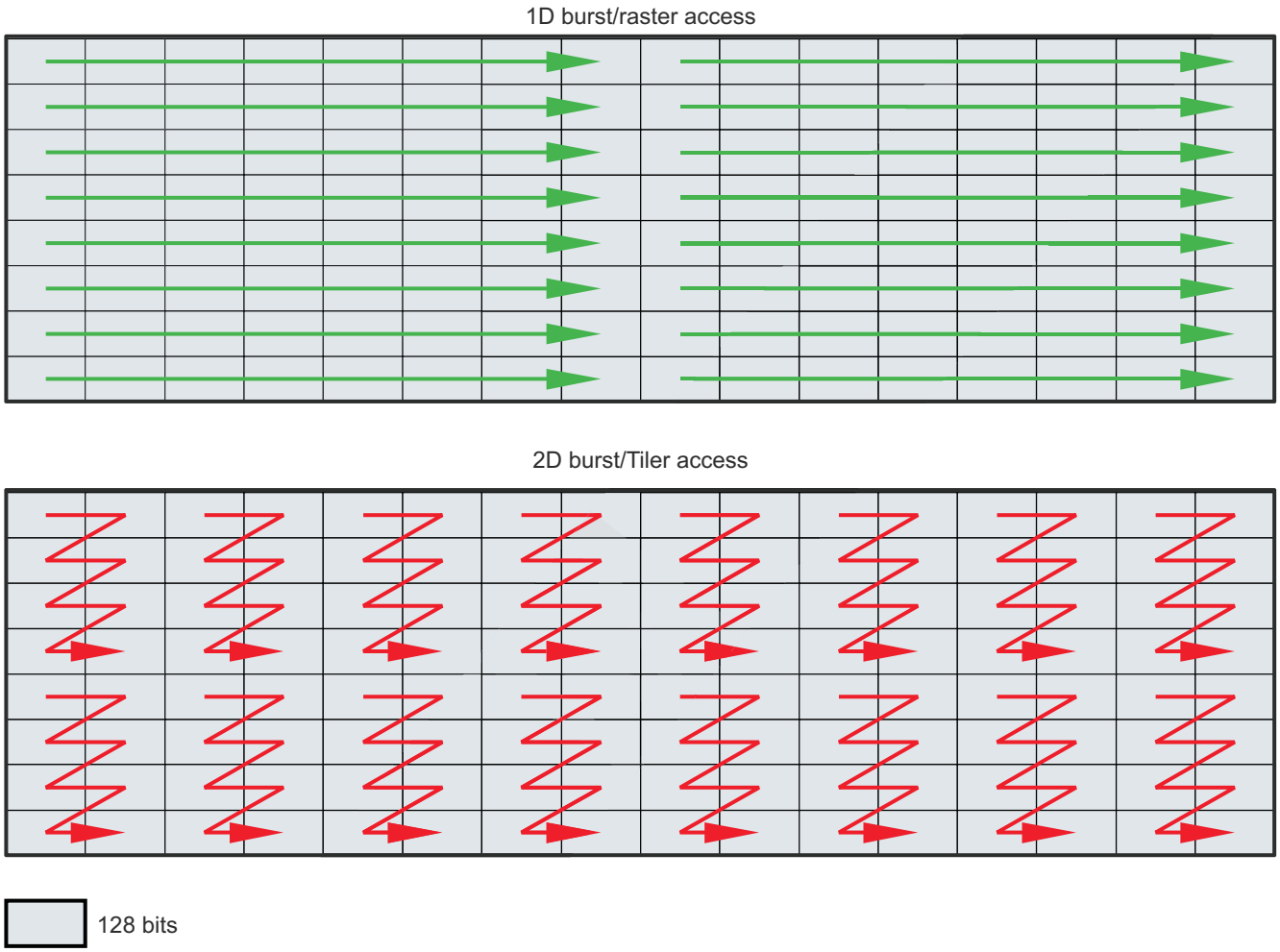


2.8.3.2.1 ISS BTE Burst Translation Principle

The BTE receives raster accesses from the OCPI port and generates TILER accesses on OCPO.

The BTE keeps track of the amount of data written into the local memories. For translated writes, 2D bursts are sent to OCPO when there is sufficient data in the local memory to generate TILER accesses. For translated reads, 2D burst are generated to fill the local memory with data that is returned when raster accesses are received on OCPI. Figure 80 shows the BTE burst translation principle.

Figure 80. ISS BTE Burst Translation Principle



In the following sections, three address spaces are described:

- Virtual: Corresponds to the translated address region on OCPI. Accesses performed to this address space are translated by the BTE. It can be seen as a 64KB x 8 k lines frame buffer. Locations in this space are expressed as byte addresses for OCP compliance. However, the access granularity is 128-bit (4 LSBs expected to be zeros). For more information about this space, see Section 2.8.3.2.2, *ISS BTE Virtual Address Space and Context Mapping*.
- Physical: Corresponds to the addresses used to access the physical buffer of the BTE. The physical space is addressed in 128-bit-wide words. However, this document refers to byte addresses to preserve homogeneity.
- Tiler: Corresponds to addresses used by translated accesses. Locations in this space are expressed as byte addresses for OCP compliance. However, the access granularity is 128-bit (4 LSBs expected to be zeros). For more information about this space, see Section 2.8.3.2.5, *ISS BTE TILER Space Accesses: 2D Burst Generation*.

2.8.3.2.2 ISS BTE Virtual Address Space and Context Mapping

The location of the virtual space in the address map of OCPI is set by the BTE_CTRL[11:8] BASE bit field. It always occupies 64KB × 8k lines = 512MB. Software must map a virtual space into an unused region (a region, for example, that the ISS top level cannot access).

The virtual space is decomposed into contexts. A context corresponds to a 2D region in the virtual space that requires burst translation. It can also be seen as a virtual frame buffer.

Accesses to different contexts can be interleaved at OCP transaction level. OCP transactions spanning multiple contexts are not allowed.

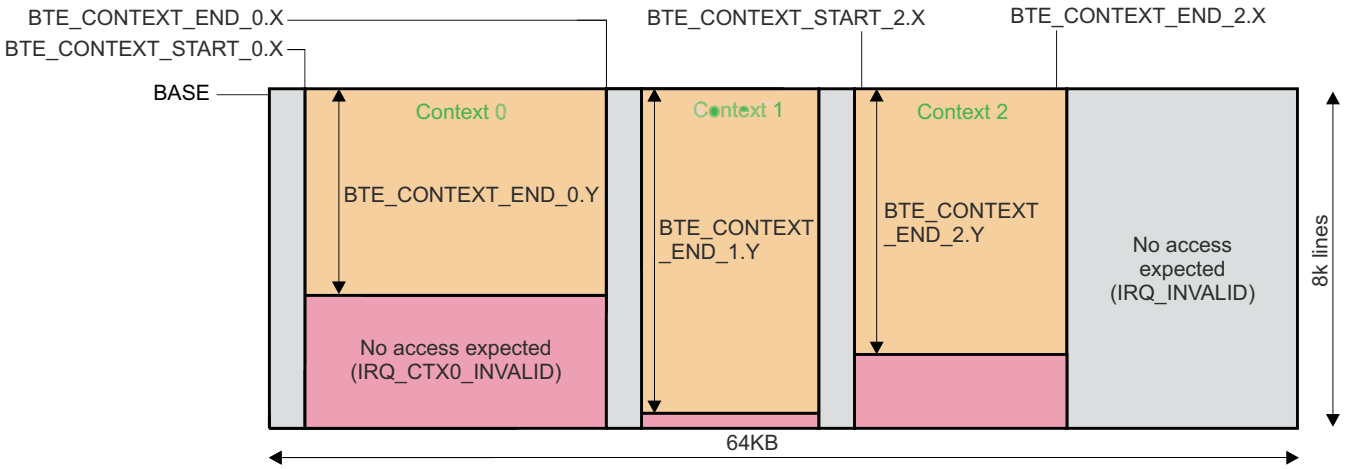
When the BTE receives an access from the OCPI port, it performs the following checks:

- If the access address is not between BASE 512MB and (BASE + 1) 512MB 1, it is handled as a transparent: request and data phases are simply forwarded to the OCPO without modification.
- If the access falls into the translated region:
  - If the access is a 2D burst, an IRQ_INVALID event is generated.
  - If the access does not map to an active context, an IRQ_INVALID event is generated. The BTE ignores bits [28:16] for this test.
  - Otherwise, context mapping is performed (see Figure 81).

Invalid requests are not forwarded to the OCPO.

Software must ensure that contexts do not overlap. The BTE hardware does not check for this condition. Wrong setup is likely to lead to corrupted data.

Figure 81. ISS BTE Context Mapping



The BTE internally keeps track of every context where the next access is expected. It internally maintains a 2D pointer, referred to as (SX_i, SY_i) in the remainder of this document. The expected byte address for an access into context x is:

ADDR = BTE_CTRL[11:8] BASE 512MB +
SX_i 16 bytes + SY_i 64KB

If an access to an unexpected location in a given context is received from OCPI, the BTE generates an IRQ_CTXx_INVALID event. The BTE provides a valid response on OCPI but does not store any data into the internal buffer. Subsequent accesses are handled normally: in other words, the BTE does not enter any specific error mode. When this happens, typically an initiator configuration is not aligned with the BTE context configuration.

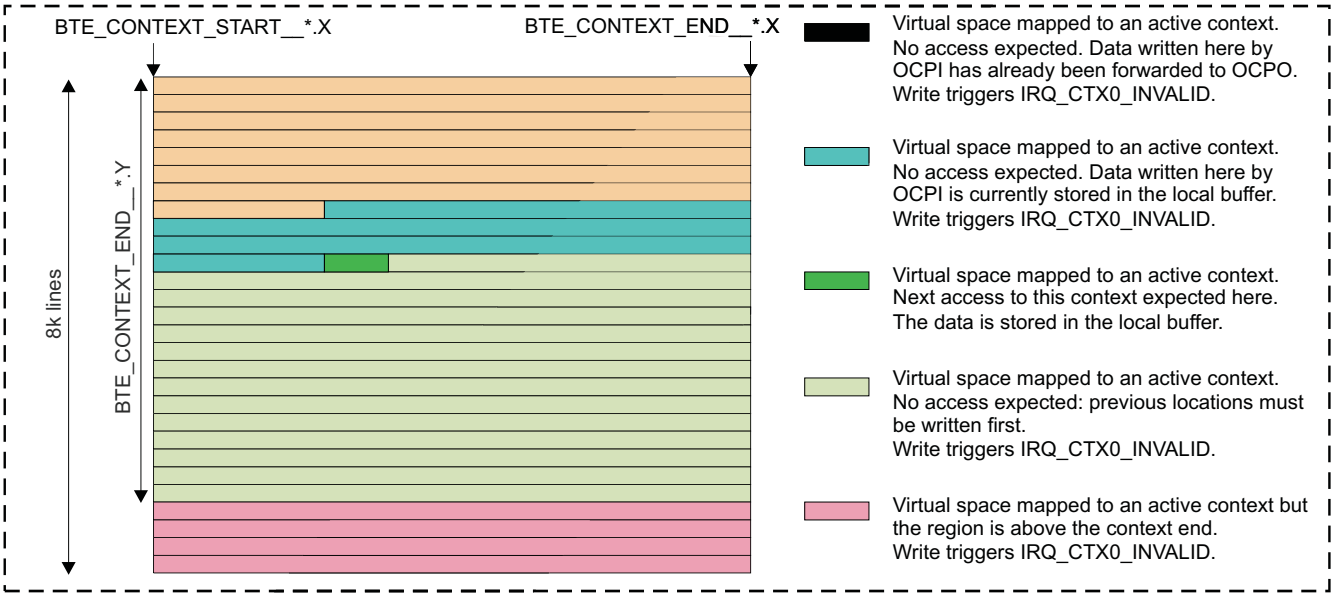
The (SX_i, SY_i) pointer of a context is updated on every access to this context. It is reset to SX_i = BTE_CONTEXT_START_i[15:7] X and SY_x = 0 when a context is enabled. SX_i is incremented by 1 for every received 16-byte word, except when it equals BTE_CONTEXT_END_i[15:4] X. In that case it is reset to BTE_CONTEXT_START_i[15:7] X and SY_i is incremented. SY_i is reset when it reaches the bottom-right corner of the context.

Therefore, the BTE can only translate raster accesses:

- Image data must be provided line by line, starting from the top-left corner of the context.
- A line stride of 64KB is expected.
- The maximum supported image height is 8k lines.

Figure 82 is an example of an active context. There is only one location in an active context where an access from OCPI is expected (green). Accesses to any other location trigger an IRQ_CTXx_INVALID event. Reads to contexts in write mode (BTE_CONTEXT_CTRL_i[7:6] MODE = 0) or writes to contexts in read mode (BTE_CONTEXT_CTRL_i[7:6] MODE = 1) trigger an IRQ_CTXx_INVALID event. When an access to an expected location is received but the burst length exceeds the context end, an IRQ_CTXx_INVALID event is triggered.

Figure 82. ISS BTE Expected Access Locations in the Virtual Space



Some alignment constraints are summarized as follows and are shown in Figure 83. They apply only to translated accesses.

- The context start addresses are aligned on 16-byte boundaries.
- The context width is aligned on 16-byte boundaries.
- The OCPI initiator must not use BYTEEN to qualify subwords in the middle of lines. The BTE interprets each BYTEEN as 0xFFFF. Access of partial words triggers an IRQ_CTXx_INVALID error.
- The line length sent by the initiator can be 1-byte aligned. The initiator can use BYTEEN to qualify valid data for the last access of the line. The BTE treats those accesses as full 16-byte writes and writes 0 data into its local buffers to complete the access. Those 0s are forwarded to OCPO when burst translation is performed. Software must ensure this data is discarded when the buffer is read back.
- The BTE does not impose any specific constraint on vertical alignment on the TILER tile and subtile grids. However, it is strongly recommended to configure the burst generation in such a way the requests performed to the TILER do not cross tile or subtile boundaries. For more information, see Section 2.8.3.2.5, *ISS BTE TILER Space Accesses: 2D Burst Generation*.
- The BTE accepts only one outstanding transaction per OCP tag on the OCPI port. When an initiator tries to generate an OCP request on a tag ID that is already used (no response has been returned on

OCPI), the OCPI port is stalled.

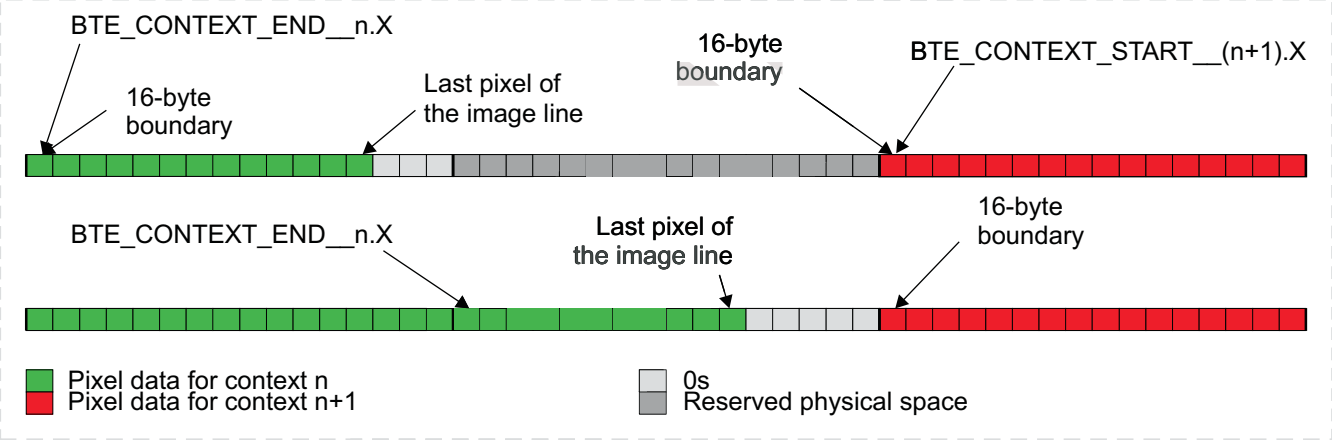
- The amount of memory allocated for each context must be a multiple of 512 bytes. Software must leave mod(BTE_CONTEXT_END_i BTE_CONTEXT_START_i + 1, 8) unused locations between the end of the context and the start of the next one. No accesses to this space from OCPI are allowed, but the BTE can use it to store data.
- While programming a BTE context, the BTE context End X value of all contexts must be less than MEMORY/64 in 128-bit address (equals MEMORY/4 in byte address)

NOTE: The generic BTE MEMORY size can be found by reading the BTE_HL_HWINFO[18:0] MEMORY bit field.

- While reading from a BTE-translated location, the SC read burst size must be greater than or equal to (=) 16 bytes (128 bits) for correct operation. Bus width translation is performed in the ISS. Therefore, 64-bit accesses lead to 128-bit accesses with BYTEEN = 0x00FF. On the BTE side, BYTEEN is always processed as 0xFFFF; thus, access to a BTE-translated 128-bit location is not correct if software uses 64-bit access.

Figure 83 shows the BTE context alignment constraints.

Figure 83. ISS BTE Context Alignment Constraints



2.8.3.2.3 ISS BTE TILER Context Configuration Example

Global configuration and context configuration must be done before traffic from the master can be enabled.

This section provides a configuration example used to write YUV4:2:0 data into a 90-degree rotated buffer (S = 1,/Y = 1,/X = 0). The YUV frame is made up of two objects: Y data (8 bits) and UV data (16 bits).

The BTE provides data orientation and format information to the TILER using a 33-bit address. It can be controlled by software using the BTE_CONTEXT_BASE_i and BTE_CONTEXT_CTRL_i[12] ADDR32 bits. ADDR32 must be set to 1. Bits [31:27] of the address control the accessed view format. Table 117 is an example of a BTE TILER context configuration.

Figure 84 shows BTE TILER mode addressing in 90- or 270-degree orientation.

Table 117. ISS BTE TILER Context Configuration Example

32	31	30	29	28	27	26 ... 4	3 ... 0
T	Orientation			Mode		Virtual Address	
1	S	/Y	/X	M1	M0	A26 ... A4	0

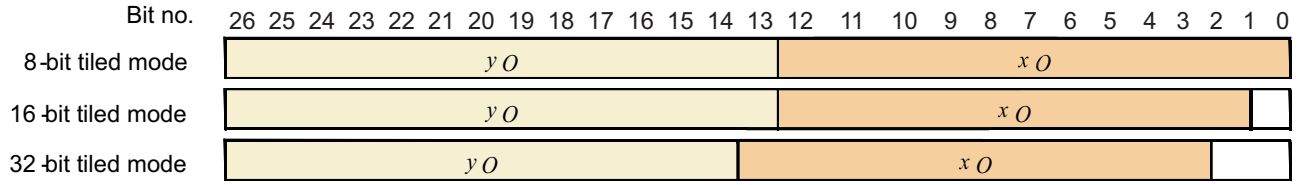
For the example described here, the following settings are used:

90-degree view, 8-bit data	ADDR[32:27] = 0b1 110 00
90-degree view, 16-bit data	ADDR[32:27] = 0b1 110 01

Software must also configure the BTE_CONTEXT_CTRL_i[9:8] GRID bit field to match the view and format set by the base address:

90-degree view, 8-bit data	GRID = 1 Stride = 8k
90-degree view, 16-bit data	GRID = 1 Stride = 8k

Figure 84. ISS BTE TILER Mode Addressing in 90- or 270-Degree Orientation



Bits [26:0] of the BTE_CONTEXT_BASE_i register are used to address a pixel in the virtual space of TILER. It must point to the top-left corner of the 2D object.

Software must also configure the other context register before it can start it by setting the BTE_CONTEXT_CTRL_i[0] START bit.

2.8.3.2.4 ISS BTE Local Memory Management

2.8.3.2.4.1 ISS BTE Introduction

The amount of data buffered is defined by the BTE_CONTEXT_CTRL_i[29:16] TRIGGER bit field. In write mode, translated 2D writes to OCPO are issued when the buffer fill level is greater than or equal to the value of the BTE_CONTEXT_CTRL_i[29:16] TRIGGER bit field. In read mode, translated 2D reads are sent to OCPO when the buffer level is less than the value of the BTE_CONTEXT_CTRL_i[29:16] TRIGGER bit field.

2.8.3.2.4.2 ISS BTE Buffer Flushing

2.8.3.2.4.2.1 ISS BTE One-Shot Mode

One-shot mode is enabled by setting the BTE_CONTEXT_CTRL_i[10] ONESHOT bit.

During normal operation, a data transfer from local memory to the TILER is automatically triggered when sufficient data is available in the buffer.

Three lines of data remain in the buffer once the OCPI initiator stops sending data into the context.

The BTE supports two ways to flush data remaining in the buffer:

- The last data (bottom-right corner) in a context is written. The last data is defined by the BTE_CONTEXT_END_i[15:4] X and BTE_CONTEXT_END_i[28:16] Y bit fields.
- Software writes the BTE_CONTEXT_CTRL_i[2] FLUSH bit. This is typically done when the context has been stopped before the full frame has been written.

An autoflush mode controls automatic context flushing when an IDLE request is received. It can be activated by setting the BTE_CONTEXT_CTRL_i[11] AUTOFLUSH bit to 1.

During a context flush, all remaining data in the buffer are written to the TILER.

Buffer flushing has lower priority than burst translation active contexts or forwarding transparent accesses. This prevents performance degradation.

If software wants to abort context operation at a random location, it must disable the context.

2.8.3.2.4.2.2 ISS BTE Continuous Mode

When continuous mode is selected (the BTE_CONTEXT_CTRL_i[10] ONESHOT bit set to 0), reception of frame n pushes the data remaining from frame 1 out of the memory to OCPO. This mode is particularly useful when vertical blanking periods are too short to perform a buffer flush at the end of the frame. Also, it avoids creating traffic peaks due to buffer flushing.

The frame height must be a multiple of eight lines and the BTE_CONTEXT_CTRL_i[14:13] INITSY bit field must be 0x0 in continuous mode.

2.8.3.2.4.3 ISS BTE Buffer Prefetch

The buffer must be prefilled before read requests requiring translation can be accepted. Buffer prefetch starts with the top-left corner of the frame when a context is enabled by setting the BTE_CONTEXT_CTRL_i[0] START bit. When enough lines have been prefetched, an IRQ_CTXx_DONE event is triggered to inform software that the context is ready to perform request translation. Typically, software enables the data flow requiring translation in response to this event. Context ready for transaction and last transaction can be controlled by BTE_CONTEXT_START and BTE_CONTEXT_END, respectively.

When a read request requiring translation is received while prefetch is ongoing, an IRQ_CTXx_ERR event is triggered. It informs software that the read traffic was enabled too early. The BTE returns 0s to OCPI for the failing request (it does not hold the response until real data is available).

After prefetch completion, when a request requiring translation is received but the required data is missing (the BTE is waiting for a response from OCPO), the BTE delays the response on OCPI until the missing data is received on OCPI.

This behavior avoids stalling the OCPI port for too long (that is, a prefetch of up to 3.25 lines of data). However, it avoids getting errors because of slow OCPI responses.

2.8.3.2.4.4 ISS BTE Bandwidth Limiter

Translated and transparent traffic has higher priority than prefetch and flushing traffic. However, overall system bandwidth is limited. Requesting too much bandwidth for prefetch and flushing traffic may increase latencies for higher priority traffic. That could affect higher priority traffic.

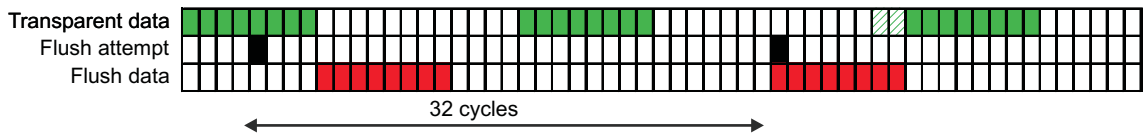
Software can limit the speed of prefetching and buffer flushing by using the BTE_CTRL[31:22] BW_LIMITER bit field. Typically, this register is used to avoid the buffer prefetch and flush traffic using all the available system bandwidth. This register does not slow down the translated or transparent traffic.

The example in Figure 85 assumes:

- 200-MHz functional clock
- 800 Mbps of transparent traffic = one 8 × 128-bit burst every 32 cycles

Without the bandwidth limiter, prefetch and flush traffic may use up to 3.2 0.8 = 2.4Gbps. Using BTE_CTRL[31:22] BW_LIMITER = 24 ensures that at a maximum one flush/prefetch request is issued every 32 cycles. A prefetch/flush request may be delayed by higher priority traffic of OCP port stalls. To avoid excessive traffic slowdown, the BTE tries to catch up by requesting the next flush/prefetch transaction earlier.

Figure 85. BTE Bandwidth Limiter Example



2.8.3.2.4.5 ISS BTE Direct Buffer Access

The local buffer can be directly accessed without any burst translation when the BTE_CONTEXT_CTRL_i[7:6] MODE bit field is set to 2.

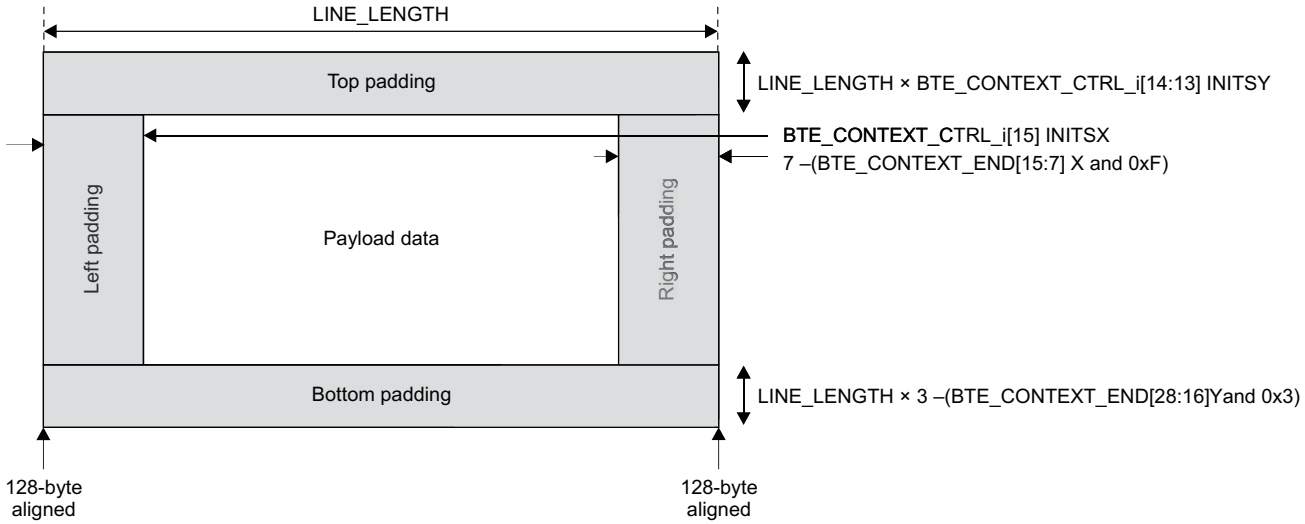
The BTE local buffer can be used as general-purpose SRAM accessed from OCPI in this mode. It can also be used for debug purposes.

2.8.3.2.5 ISS BTE TILER Space Accesses: 2D Burst Generation

2.8.3.2.5.1 ISS BTE Buffer Fill Level

The BTE maintains an internal counter to keep track of the buffer fill level. This internal counter is used to detect when a translated OCP request is sent to the OCPO port. The counter also accounts for padding data. Figure 86 shows the BTE buffer fill-level padding.

Figure 86. ISS BTE Buffer Fill-Level Padding



LINE_LENGTH = BTE_CONTEXT_END_i[15:4] × 0xFF8 – (BTE_CONTEXT_START_i[15:7] × 0xFF8) + 8 words of 16 bytes.

Figure 86 is a visual representation of the following explanation about how read or write events are triggered determined by the internal buffer size.

When context is started, the buffer fill level is initialized to LINE_LENGTH × BTE_CONTEXT_CTRL_i[14:13] INITSY + BTE_CONTEXT_CTRL_i[15] INITSX (see the previous equation for the LINE_LENGTH calculation). If the BTE reads or writes, the level of the internal buffer is incremented or decremented, respectively, by the burst size. Here, only full 16-byte accesses are performed. Other OCP BYTEEN patterns are forced to 0xFF.

The BTE_CONTEXT_CTRL_i [29:16] TRIGGER bit field triggers a buffer level read or write even. In write mode, if flushing, the data level is greater than 0; if not flushing, the BTE translates to OCPO when the level is greater than or equal to the value set by the register. If the level is smaller than the BTE_CONTEXT_CTRL_i[29:16] TRIGGER software setup, then the BTE reads data.

2.8.3.2.5.2 ISS BTE OCP Request Generation

Except on borders:

- TILER bursts are 32 bytes × 4 lines of data blocks.
- Bursts are aligned on subtile boundaries.

The BTE maintains an internal 2D pointer (DX_i, DY_i) that corresponds to the top-left corner of the next access to be issued to the TILER. It is initialized to (0, 0) when a context is enabled or wraps around. It is updated each time an access to the TILER is performed.

Table 118 lists all supported TILER formats and views.

Table 118. ISS BTE Supported TILER Formats and Views

Modes	View			Description	OFST	Subtile Grid		1KB Tile Grid		X		Y		GRID	Subtile Aligned		Tile Aligned	
	S	/Y	/X			X	Y	X	Y	LSB	MSB	LSB	MSB		X	Y	X	Y
8-bit	0	0	0	0-degree view	16384	4	4	32	32	0	13	14	26	0	[1:0]	[15:14]	[4:0]	[18:14]
	0	0	1	0-degree view with vertical mirror														
	0	1	0	0-degree view with horizontal mirror														
	0	1	1	180-degree view														
	1	0	0	90-degree view with vertical mirror	8192	4	4	32	32	0	12	13	26	1	[1:0]	[14:13]	[4:0]	[17:13]
	1	0	1	270-degree view														
	1	1	0	90-degree view														
	1	1	1	90-degree view with horizontal mirror														
16-bit	0	0	0	0-degree view	32768	8	2	64	16	0	14	15	26	2	[2:0]	[15]	[5:0]	[18:15]
	0	0	1	0-degree view with vertical mirror														
	0	1	0	0-degree view with horizontal mirror														
	0	1	1	180-degree view														
	1	0	0	90-degree view with vertical mirror	8192	4	4	32	32	0	12	13	26	1	[1:0]	[14:13]	[4:0]	[17:13]
	1	0	1	270-degree view														
	1	1	0	90-degree view														
	1	1	1	90-degree view with horizontal mirror														
32-bit	0	0	0	0-degree view	32768	8	2	64	16	0	14	15	26	2	[2:0]	[15]	[5:0]	[18:15]
	0	0	1	0-degree view with vertical mirror														
	0	1	0	0-degree view with horizontal mirror														
	0	1	1	180-degree view														
	1	0	0	90-degree view with vertical mirror	16384	8	2	64	16	0	13	14	26	3	[2:0]	[14]	[5:0]	[17:14]
	1	0	1	270-degree view														
	1	1	0	90-degree view														
	1	1	1	90-degree view with horizontal mirror														

The BTE_CONTEXT_BASE_i and BTE_CONTEXT_CTRL_i[9:8] GRID registers must the configured by software to choose the format and view. The format must match the data stored into the virtual space. The choice of the view depends on the desired behavior.

The GRID bit field controls the used OCP stride and OCP address generation. The OCP address is generated using the following formula:

OCP_ADDR = BTE_CONTEXT_BASE_n + DX_x + DY_x Y_LSB

Y_LSB corresponds to the Y LSB column of Table 118.

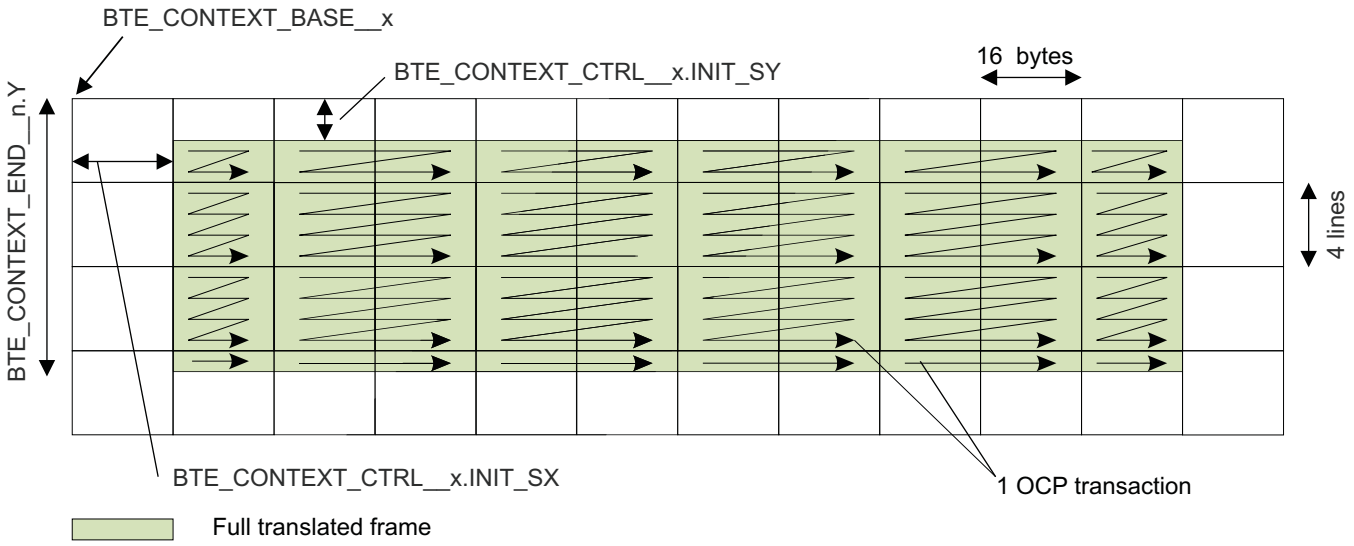
The OCP stride corresponds to the OFST column of Table 118. The tile and subtile grid information is currently not used by hardware.

The BTE does not automatically realign 2D bursts on tile or subtile boundaries. However, software can force the (SX_i, SY_i) (see Section 2.8.3.2.4, *ISS BTE Local Memory Management*, for the definition of SY_i) reset value to be used. It is set through the BTE_CONTEXT_CTRL_i[15] INITSX bit and the BTE_CONTEXT_CTRL_i[14:13] INITSY bit field.

The length and height of the 2D burst is adapted by the BTE to avoid sending dummy data to the TILER.

Figure 87 is an example of BTE 2D burst generation. The vertical start and end of the full 2D frame are not vertically aligned on the grid.

Figure 87. ISS BTE 2D Burst Generation



2.8.3.2.6 ISS BTE Posted and Nonposted Write Support

The BTE can handle posted and nonposted writes received on OCPI. Normally, only nonposted writes should be used.

For transparent accesses, posted and nonposted writes are forwarded to OCPO. The response provided by OCPO is returned to OCPI.

For translated accesses, the response to posted and nonposted writes is provided by the BTE. It does not wait for the response of the translated request that is sent to the TILER. In other words, the BTE has no true nonposted write support for translated accesses. The BTE does not ensure that the data has effectively been written to the destination memory when it returns the response to a nonposted translated write.

True nonposted write support is ensured for transparent accesses. Only nonposted writes must be used. Select nonposted write mode through the BTE_CTRL[5] POSTED bit.

2.8.3.2.7 ISS BTE Error Reporting

Unexpected accesses are flagged using interrupts. Also, when an SResp = ERR is received on OCPO, an interrupt is triggered. If the response corresponds to a transparent access, it is forwarded to OCPI.

The BTE is not an OCP checker: It expects only valid and supported transactions from the external world.

Also, it is not intended to detect all types of software errors; few cases are detected. Those cases are described in the functional description sections.

2.8.3.2.8 ISS BTE Interrupts

All events generated by the module are merged into a single event at ISS level. This event can be enabled from ISS level by the ISS_HL_IRQENABLE_SET_i[11] BTE_IRQ bit. Table 119 lists the BTE interrupt events.

Table 119. ISS BTE Interrupt Events

Event	Description
IRQ_OCP_ERR	OCP error received from OCPO master port
IRQ_INVALID	An access to a location that is not mapped to any context has been performed. For more information, see Section 2.8.3.2.2, *ISS BTE Virtual Address Space and Context Mapping*.
IRQ_CTXx_DONE	Context has been fully transferred to the TILER. This interrupt is triggered when flushing completes in one-shot mode. It is triggered once per frame in continuous mode.
IRQ_CTXx_INVALID	Unexpected address sequence or access direction (read of a context in write mode or write of a context in read mode). For more information, see Section 2.8.3.2.2, *ISS BTE Virtual Address Space and Context Mapping*.
IRQ_CTXx_ERR	Can occur only when a context is configured in read mode. This request triggers when a read request is received but insufficient data is buffered to perform the translation. For more information, see Section 2.8.3.2.4.3, *ISS BTE Buffer Prefetch*.

2.8.3.2.9 ISS BTE Debug Support

The BTE has no specific debug support.

2.8.4 ISS BTE Programming Model

2.8.4.1 ISS BTE Reset

The BTE can accept a general software reset, propagated through all the hierarchy. This reset can be done to initialize the module and has the same effect as the hardware reset.

1. Set the BTE_HL_SYSCONFIG[0] SOFTRESET bit to 1.
2. Read the BTE_HL_SYSCONFIG[0] SOFTRESET bit to check whether it is set to 1, which means the reset occurred.

If after five reads, the BTE_HL_SYSCONFIG[0] SOFTRESET bit still returns 0, assume that an error occurred during the reset stage.

Programmers must not set the BTE_HL_SYSCONFIG[0] SOFTRESET bit to 1 if the BTE is integrated in a subsystem; it is safer to use the software reset at subsystem level.

NOTE: A software reset does not reset the IDLE protocol signals.

2.8.4.2 ISS BTE Interrupts

All events are mapped to a single interrupt output, BTE_IRQ. Table 120 lists the procedure to configure or manage the BTE interrupts.

Table 120. ISS BTE Configure/Manage Interrupts

Step	Bit Field	Value
Each event that generates an interrupt can be individually enabled by setting the appropriate bit.	BTE_HL_IRQENABLE_SET	
Each event that generates an interrupt can be individually disabled by setting the adequate bit.	BTE_HL_IRQENABLE_CLR	
When an event occurs, the corresponding bit in the BTE_HL_IRQSTATUS_RAW register is set, regardless of whether or not the event is enabled. Bits in the BTE_HL_IRQSTATUS registers are set only when an enabled event occurs.	BTE_HL_IRQSTATUS_RAW and BTE_HL_IRQSTATUS	
Software can clear a pending event by setting the adequate bit in the BTE_HL_IRQSTATUS register.	BTE_HL_IRQSTATUS	

2.8.4.3 ISS BTE Context Configuration

Global configuration and context configuration must be done before traffic from the master can be enabled.

This section provides a configuration example used to write YUV4:2:0 data into a 90-degree rotated buffer (S = 1,/Y = 1,/X = 0). The YUV frame consists of two objects: Y data (8 bits) and UV data (16 bits).

The BTE provides data orientation and format information to the TILER using a 33-bit address. It can be controlled by software using the BTE_CONTEXT_BASE_i and BTE_CONTEXT_CTRL_i[12] ADDR32 registers.

ADDR32 must be set to 1. Bits [31:27] of the address control the accessed view and data format. Table 121 gives the format of the TILER address. Figure 88 shows BTE tiled mode addressing in 90- or 270-degree orientation.

Table 121. TILER Address Format

32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T	Orientation			Mode		Virtual Address																										
1	S			M1	M0	A26 ... A4																									0	

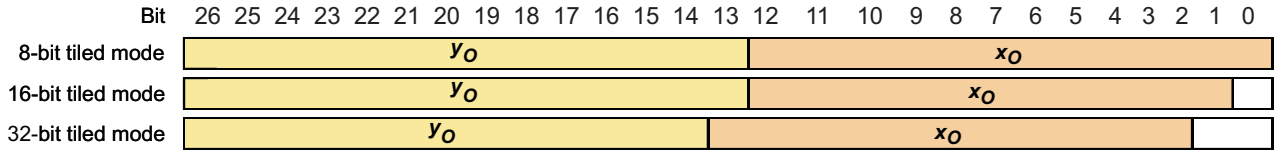
For the example described here, the following settings are used.

90-degree view, 8-bit data	ADDR[32:27] = 0b1 110 00
90-degree view, 16-bit data	ADDR[32:27] = 0b1 110 01

Software must also configure the BTE_CONTEXT_CTRL_i[9:8] GRID bit field to match the view and format set by the base address:

90-degree view, 8-bit data	GRID = 1	Stride = 8k
90-degree view, 16-bit data	GRID = 1	Stride = 8k

Figure 88. ISS BTE Tiler Mode Addressing in 90- or 270-Degree Orientation (S = 1)



Bits [26:0] of the BTE_CONTEXT_BASE_i register are used to address a pixel in the virtual space of the TILER. It must point to the top-left corner of the 2D object.

Software must also configure the other context registers before it can start it by setting the BTE_CONTEXT_CTRL_i[0] START bit.

2.8.4.4 ISS BTE Change Context Configuration

All contexts operate independently. Software can change the configuration of an inactive context while other contexts are active and perform request translation.

When software must change the configuration of an active context, it must follow the sequence in Table 122.

Table 122. ISS BTE Change Context Configuration

Step	Bit Field	Value
Ensure that the initiator does not send any more data to this context until it is re-enabled.		
Disable the context. • The STOP condition is considered on a valid OCP boundary. • It preserves the internal states so that buffer flushing can be done. • The BTE no longer translates requests received for this context. Any transactions received for an inactive context trigger error interrupts. • The BTE completes all outstanding transactions on OCPO.	BTE_CONTEXT_CTRL_i[1] STOP	0x1
Flush all remaining data for the context, if needed. If software simply wants to abort the transfer to recover from some error condition, flushing is not needed. The context is automatically reset when it is enabled again.	BTE_CONTEXT_CTRL_i[2] FLUSH	0x1
Wait until the context completes pending OCP transaction and buffer flush (if enabled). It sets the IRQ_CTXx_DONE when it becomes idle.	BTE_HL_IRQSTATUS. IRQ_CTXx_DONE	Read 0x0
Change the context configuration.		
Enable the context by setting the BTE_CONTEXT_CTRL_i[0] START bit. Setting the START bit resets the internal state-machine of the context.	BTE_CONTEXT_CTRL_i[0] START	0x1

Alternatively, software can change the context mode to one-shot and wait until the CTXx_DONE_IRQ is triggered.

NOTE: Once a context is disabled it cannot be resumed simply by writing the START bit. In fact, doing so resets the internal FSM. If data is lost in the buffer, it will be lost.

2.8.5 ISS BTE Registers

Table 123 lists the BTE instance.

Table 123. ISS BTE Instance Summary

Module Name	Base Address Cortex-M3 Private Access	Base Address L3 Interconnect	Size
ISS_BTE	0x5504 2000	0x5C00 2000	512 bytes

Table 124 summarizes the BTE registers.

Table 124. ISS BTE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ISS_BTE Base Address Cortex-M3 Private Access	ISS_BTE Base Address L3 Interconnect
BTE_HL_REVISION	R	32	0x0000 0000	0x5504 2000	0x5C00 2000
BTE_HL_HWINFO	R	32	0x0000 0004	0x5504 2004	0x5C00 2004
BTE_HL_SYSCONFIG	RW	32	0x0000 0010	0x5504 2010	0x5C00 2010
RESERVED	RW	32	0x0000 001C	0x5504 201C	0x5C00 201C
BTE_HL_IRQSTATUS_RAW	RW	32	0x0000 0020	0x5504 2020	0x5C00 2020
BTE_HL_IRQSTATUS	RW	32	0x0000 0024	0x5504 2024	0x5C00 2024
BTE_HL_IRQENABLE_SET	RW	32	0x0000 0028	0x5504 2028	0x5C00 2028
BTE_HL_IRQENABLE_CLR	RW	32	0x0000 002C	0x5504 202C	0x5C00 202C
BTE_CTRL	RW	32	0x0000 0030	0x5504 2030	0x5C00 2030
BTE_CTRL1	RW	32	0x0000 0034	0x5504 2034	0x5C00 2034
BTE_CONTEXT_CTRL_i (1)	RW	32	0x0000 0040 + (0x20 * i)	0x5504 2040 + (0x20 * i)	0x5C00 2040 + (0x20 * i)
BTE_CONTEXT_BASE_i (1)	RW	32	0x0000 0044 + (0x20 * i)	0x5504 2044 + (0x20 * i)	0x5C00 2044 + (0x20 * i)
BTE_CONTEXT_START_i (1)	RW	32	0x0000 0048 + (0x20 * i)	0x5504 2048 + (0x20 * i)	0x5C00 2048 + (0x20 * i)
BTE_CONTEXT_END_i (1)	RW	32	0x0000 004C + (0x20 * i)	0x5504 204C + (0x20 * i)	0x5C00 204C + (0x20 * i)

(1) i = 0 to 7

2.8.5.1 BTE_HL_REVISION

Table 125. BTE_HL_REVISION

Address Offset	0x0000 0000		
Physical Address	0x5504 2000 0x5C00 2000	**Instance**	ISS_BTE_CORTEX-M3 ISS_BTE_L3
Description	IP revision identifier (X.Y.R) Used by software to track features, bugs, and compatibility		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	See (1)

(1) TI internal data

2.8.5.2 BTE_HL_HWINFO

Table 126. BTE_HL_HWINFO

Address Offset	0x0000 0004		
Physical Address	0x5504 2004 0x5C00 2004	**Instance**	ISS_BTE_CORTEX-M3 ISS_BTE_L3
Description	Information about the hardware configuration of the IP module; that is, typically, the HDL generics (if any) of the module.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESPFIFO			CONTEXTS					MEMORY															

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x00
23:21	RESPFIFO	Response FIFO size Read 0x0: Reserved Read 0x1: 16 x 128 bits Read 0x2: 32 x 128 bits Read 0x3: 64 x 128 bits Read 0x4: 128 x 128 bits Read 0x5: Reserved Read 0x6: Reserved Read 0x7: Reserved	R	0x2
20:19	CONTEXTS	Number of contexts Read 0x0: 2 contexts Read 0x1: 4 contexts Read 0x2: 8 contexts Read 0x3: Reserved	R	0x1
18:0	MEMORY	Memory size, in bytes	R	0x05800

2.8.5.3 BTE_HL_SYSCONFIG

Table 127. BTE_HL_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x5504 2010 0x5C00 2010	**Instance**	ISS_BTE_CORTEX-M3 ISS_BTE_L3
Description	Clock management configuration		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												IDLEMODE		RESERVED	SOFTRESET

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000000
3:2	IDLEMODE	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state. 0x0: An IDLE request is acknowledged unconditionally 0x1: An IDLE request is never acknowledged 0x2: Smart-idle mode. Acknowledgment to an IDLE request is given based on the internal activity of the module. 0x3: Reserved. Do not use	RW	0x2
1	RESERVED	Reserved	R	0
0	SOFTRESET	Software reset. Write 0x0: No action Write 0x1: Initiate software reset Read 0x1: Reset (software or other) ongoing Read 0x0: Reset done, no pending action	RW	0

2.8.5.4 BTE_HL_IRQSTATUS_RAW

Table 128. BTE_HL_IRQSTATUS_RAW

Address Offset	0x0000 0020		
Physical Address	0x5504 2020 0x5C00 2020	**Instance**	ISS_BTE_CORTEX-M3 ISS_BTE_L3
Description	Per-event raw interrupt status vector Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_CTX7_ERR	IRQ_CTX6_ERR	IRQ_CTX5_ERR	IRQ_CTX4_ERR	IRQ_CTX3_ERR	IRQ_CTX2_ERR	IRQ_CTX1_ERR	IRQ_CTX0_ERR	IRQ_CTX7_INVALID	IRQ_CTX6_INVALID	IRQ_CTX5_INVALID	IRQ_CTX4_INVALID	IRQ_CTX3_INVALID	IRQ_CTX2_INVALID	IRQ_CTX1_INVALID	IRQ_CTX0_INVALID	IRQ_CTX7_DONE	IRQ_CTX6_DONE	IRQ_CTX5_DONE	IRQ_CTX4_DONE	IRQ_CTX3_DONE	IRQ_CTX2_DONE	IRQ_CTX1_DONE	IRQ_CTX0_DONE	RESERVED						IRQ_INVALID	IRQ_OCP_ERR

Bits	Field Name	Description	Type	Reset
31	IRQ_CTX7_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
30	IRQ_CTX6_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
29	IRQ_CTX5_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
28	IRQ_CTX4_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
27	IRQ_CTX3_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
26	IRQ_CTX2_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
25	IRQ_CTX1_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
24	IRQ_CTX0_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
23	IRQ_CTX7_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
22	IRQ_CTX6_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
21	IRQ_CTX5_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
20	IRQ_CTX4_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
19	IRQ_CTX3_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
18	IRQ_CTX2_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
17	IRQ_CTX1_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
16	IRQ_CTX0_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
15	IRQ_CTX7_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
14	IRQ_CTX6_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
13	IRQ_CTX5_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
12	IRQ_CTX4_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
11	IRQ_CTX3_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
10	IRQ_CTX2_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
9	IRQ_CTX1_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
8	IRQ_CTX0_DONE	Write mode: Context has been fully transferred to the TILER Read mode: Context prefetch has completed. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
7:2	RESERVED	Reserved	R	0x00
1	IRQ_INVALID	Invalid access to the virtual space Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
0	IRQ_OCP_ERR	OCP error received from OCP master port. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

2.8.5.5 BTE_HL_IRQSTATUS

Table 129. BTE_HL_IRQSTATUS

Address Offset	0x0000 0024		
Physical Address	0x5504 2024 0x5C00 2024	**Instance**	ISS_BTE_CORTEX-M3 ISS_BTE_L3
Description	Per-event "enabled" interrupt status vector. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_CTX7_ERR	IRQ_CTX6_ERR	IRQ_CTX5_ERR	IRQ_CTX4_ERR	IRQ_CTX3_ERR	IRQ_CTX2_ERR	IRQ_CTX1_ERR	IRQ_CTX0_ERR	IRQ_CTX7_INVALID	IRQ_CTX6_INVALID	IRQ_CTX5_INVALID	IRQ_CTX4_INVALID	IRQ_CTX3_INVALID	IRQ_CTX2_INVALID	IRQ_CTX1_INVALID	IRQ_CTX0_INVALID	IRQ_CTX7_DONE	IRQ_CTX6_DONE	IRQ_CTX5_DONE	IRQ_CTX4_DONE	IRQ_CTX3_DONE	IRQ_CTX2_DONE	IRQ_CTX1_DONE	IRQ_CTX0_DONE	RESERVED						IRQ_INVALID	IRQ_OCP_ERR

Bits	Field Name	Description	Type	Reset
31	IRQ_CTX7_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
30	IRQ_CTX6_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
29	IRQ_CTX5_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
28	IRQ_CTX4_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
27	IRQ_CTX3_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
26	IRQ_CTX2_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
25	IRQ_CTX1_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
24	IRQ_CTX0_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
23	IRQ_CTX7_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
22	IRQ_CTX6_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
21	IRQ_CTX5_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
20	IRQ_CTX4_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
19	IRQ_CTX3_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
18	IRQ_CTX2_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
17	IRQ_CTX1_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
16	IRQ_CTX0_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
15	IRQ_CTX7_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
14	IRQ_CTX6_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
13	IRQ_CTX5_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
12	IRQ_CTX4_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
11	IRQ_CTX3_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
10	IRQ_CTX2_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
9	IRQ_CTX1_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
8	IRQ_CTX0_DONE	Write mode: Context has been fully transferred to the TILER Read mode: Context prefetch has completed. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
7:2	RESERVED	Reserved	R	0x00
1	IRQ_INVALID	Invalid access to the virtual space Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
0	IRQ_OCP_ERR	OCP error received from OCP master port. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

2.8.5.6 BTE_HL_IRQENABLE_SET

Table 130. BTE_HL_IRQENABLE_SET

Address Offset	0x0000 0028		
Physical Address	0x5504 2028 0x5C00 2028	**Instance**	ISS_BTE_CORTEX-M3 ISS_BTE_L3
Description	Per-event interrupt enable bit vector Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_CTX7_ERR	IRQ_CTX6_ERR	IRQ_CTX5_ERR	IRQ_CTX4_ERR	IRQ_CTX3_ERR	IRQ_CTX2_ERR	IRQ_CTX1_ERR	IRQ_CTX0_ERR	IRQ_CTX7_INVALID	IRQ_CTX6_INVALID	IRQ_CTX5_INVALID	IRQ_CTX4_INVALID	IRQ_CTX3_INVALID	IRQ_CTX2_INVALID	IRQ_CTX1_INVALID	IRQ_CTX0_INVALID	IRQ_CTX7_DONE	IRQ_CTX6_DONE	IRQ_CTX5_DONE	IRQ_CTX4_DONE	IRQ_CTX3_DONE	IRQ_CTX2_DONE	IRQ_CTX1_DONE	IRQ_CTX0_DONE	RESERVED						IRQ_INVALID	IRQ_OCP_ERR

Bits	Field Name	Description	Type	Reset
31	IRQ_CTX7_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
30	IRQ_CTX6_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
29	IRQ_CTX5_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
28	IRQ_CTX4_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
27	IRQ_CTX3_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
26	IRQ_CTX2_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
25	IRQ_CTX1_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
24	IRQ_CTX0_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
23	IRQ_CTX7_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
22	IRQ_CTX6_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
21	IRQ_CTX5_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
20	IRQ_CTX4_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
19	IRQ_CTX3_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
18	IRQ_CTX2_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
17	IRQ_CTX1_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
16	IRQ_CTX0_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
15	IRQ_CTX7_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
14	IRQ_CTX6_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
13	IRQ_CTX5_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
12	IRQ_CTX4_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
11	IRQ_CTX3_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
10	IRQ_CTX2_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
9	IRQ_CTX1_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
8	IRQ_CTX0_DONE	Write mode: Context has been fully transferred to the TILER Read mode: Context prefetch has completed. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
7:2	RESERVED	Reserved	R	0x00
1	IRQ_INVALID	Invalid access to the virtual space Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
0	IRQ_OCP_ERR	OCP error received from OCP master port. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

2.8.5.7 BTE_HL_IRQENABLE_CLR

Table 131. BTE_HL_IRQENABLE_CLR

Address Offset	0x0000 002C		
Physical Address	0x5504 202C 0x5C00 202C	**Instance**	ISS_BTE_CORTEX-M3 ISS_BTE_L3
Description	Per-event interrupt enable bit vector Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_CTX7_ERR	IRQ_CTX6_ERR	IRQ_CTX5_ERR	IRQ_CTX4_ERR	IRQ_CTX3_ERR	IRQ_CTX2_ERR	IRQ_CTX1_ERR	IRQ_CTX0_ERR	IRQ_CTX7_INVALID	IRQ_CTX6_INVALID	IRQ_CTX5_INVALID	IRQ_CTX4_INVALID	IRQ_CTX3_INVALID	IRQ_CTX2_INVALID	IRQ_CTX1_INVALID	IRQ_CTX0_INVALID	IRQ_CTX7_DONE	IRQ_CTX6_DONE	IRQ_CTX5_DONE	IRQ_CTX4_DONE	IRQ_CTX3_DONE	IRQ_CTX2_DONE	IRQ_CTX1_DONE	IRQ_CTX0_DONE	RESERVED						IRQ_INVALID	IRQ_OCP_ERR

Bits	Field Name	Description	Type	Reset
31	IRQ_CTX7_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
30	IRQ_CTX6_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
29	IRQ_CTX5_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
28	IRQ_CTX4_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
27	IRQ_CTX3_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
26	IRQ_CTX2_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
25	IRQ_CTX1_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
24	IRQ_CTX0_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
23	IRQ_CTX7_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
22	IRQ_CTX6_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
21	IRQ_CTX5_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
20	IRQ_CTX4_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
19	IRQ_CTX3_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
18	IRQ_CTX2_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
17	IRQ_CTX1_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
16	IRQ_CTX0_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
15	IRQ_CTX7_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
14	IRQ_CTX6_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
13	IRQ_CTX5_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
12	IRQ_CTX4_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
11	IRQ_CTX3_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
10	IRQ_CTX2_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
9	IRQ_CTX1_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
8	IRQ_CTX0_DONE	Write mode: Context has been fully transferred to the TILER Read mode: Context prefetch has completed. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
7:2	RESERVED	Reserved	R	0x00
1	IRQ_INVALID	Invalid access to the virtual space Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
0	IRQ_OCP_ERR	OCP error received from OCP master port. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

2.8.5.8 BTE_CTRL

Table 132. BTE_CTRL

Address Offset	0x0000 0030		
Physical Address	0x5504 2030 0x5C00 2030	**Instance**	ISS_BTE_CORTEX-M3 ISS_BTE_L3
Description	BTE control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BW_LIMITER								RESERVED								BASE								RESERVED		POSTED	RESERVED	TAG_CNT			

Bits	Field Name	Description	Type	Reset
31:22	BW_LIMITER	Minimum number of OCP cycles between two consecutive buffer flushing or prefetch requests. Used to limit the bandwidth used to fill/empty buffers. 0: Maximum speed. Up to 1 request every 8 cycles (3.2GB @ 200 MHz) 1: Up to 1 request every 9 cycles. 1023: Minimum speed. Up to 1 request every 1031 cycles (24MB @ 200 MHz)	RW	0x000
21:12	RESERVED	Reserved	R	0x000
11:8	BASE	Base address of the virtual space translated by the BTE. Start address = BASE*512MB End address = (BASE+1)*512MB 1 For example: BASE=3 = 0x 0 6000 0000 - 0x 0 7FFF FFFF	RW	0x0
7:6	RESERVED	Reserved	R	0x0
5	POSTED	Select among posted and nonposted writes for translated requests. 0x0: Use non posted writes 0x1: Use posted writes	RW	0
4	RESERVED	Reserved	R	0
3:0	TAG_CNT	BTE could use up to TAG_CNT+1 tags on OCPO. There could only be one outstanding request per tag. TAG_CNT does not control the number of requests it could handle on OCPI. This register is internally shadowed. Modifications are taken into account when there are no outstanding transactions on OCPO. TAG ID 0 to TAG_CNT are used on OCPO.	RW	0xF

2.8.5.9 BTE_CTRL1

Table 133. BTE_CTRL1

Address Offset	0x0000 0034		
Physical Address	0x5504 2034 0x5C00 2034	**Instance**	ISS_BTE_CORTEX-M3 ISS_BTE_L3
Description	BTE control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RESP_FIFO_THR							

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0000000
6:0	RESP_FIFO_THR	The BTE stops accepting new requests from OCPI (on a clean burst boundary) when the response FIFO contains more than RESP_FIFO_THR words. The reset value is FIFO_SIZE - 16 - 1. FIFO_SIZE = 8 * $2^{RESPFIFO}$	RW	0x0F

2.8.5.10 BTE_CONTEXT_CTRL_i

Table 134. BTE_CONTEXT_CTRL_i

Address Offset	0x0000 0040 + (0x20 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 2040 + (0x20 * i) 0x5C00 2040 + (0x20 * i)	**Instance**	ISS_BTE_CORTEX-M3 ISS_BTE_L3
Description	Context control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								TRIGGER								INITSX	INITSY	ADDR32	AUTOFLUSH	ONESHOT	GRID			MODE	RESERVED				FLUSH	STOP	START

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Reserved	R	0x0
29:16	TRIGGER	Threshold used to trigger translated requests to OCPO. Unit: words or 16 bytes Valid range: 3 lines + 2 ... 4 lines WRITE: a 2D write is issued to OCPO when the internal buffer level (including masked accesses) is superior or equal to TRIGGER READ: a 2D read is issued to OCPO when the internal buffer level (including masked accesses) is inferior to TRIGGER.	RW	0x0000
15	INITSX	Reset value to be used for SX__x. Check the section describing the local buffer management for details.	RW	0

Bits	Field Name	Description	Type	Reset
14:13	INITSY	Reset value to be used for SY__x. Check the section describing the local buffer management for details.	RW	0x0
12	ADDR32	Controls the value of the OCP address bit 32 to be used for translated accesses	RW	1
11	AUTOFLUSH	Controls automatic context flushing when an IDLE request is received 0x0: Disabled 0x1: Enabled	RW	0
10	ONESHOT	Selects one-shot or continuous mode 0x0: The context is automatically re-enabled when its end is reached. 0x1: The context is disabled when the end of a frame has been reached.	RW	0
9:8	GRID	Grid used to access the TILER 0x0: Stride = 16k Subtile = 4x4 bytes Tile = 32x32 bytes 0x1: Stride = 8k Subtile = 4x4 bytes Tile = 32x32 bytes 0x2: Stride = 32k Subtile = 8x2 bytes Tile = 64x16 bytes 0x3: Stride = 16k Subtile = 8x2 bytes Tile = 64x16 bytes	RW	0x0
7:6	MODE	Select the translation mode for the context 0x0: Write translation 0x1: Read translation 0x2: Direct access to local buffer 0x3: reserved	RW	0x0
5:3	RESERVED	Reserved	R	0x0
2	FLUSH	Flushes all remaining data of the context to the TILER. Write 0x0: No effect Write 0x1: Flush	W	0
1	STOP	Stops the context on a clean OCP transaction boundary. Write 0x0: No effect Write 0x1: Stop the context	W	0
0	START	Resets the contexts internal state and enables the context on a clean OCP transaction boundary. Write 0x0: No effect Write 0x1: Reset + Enable	W	0

2.8.5.11 BTE_CONTEXT_BASE_i

Table 135. BTE_CONTEXT_BASE_i

Address Offset	0x0000 0044 + (0x20 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 2044 + (0x20 * i) 0x5C00 2044 + (0x20 * i)	**Instance**	ISS_BTE_CORTEX-M3 ISS_BTE_L3
Description	Address of the frame buffer in the TILER address space.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																														RESERVED	

Bits	Field Name	Description	Type	Reset
31:5	ADDR	Address	RW	0x0000000
4:0	RESERVED	Reserved	R	0x00

2.8.5.12 BTE_CONTEXT_START_i

Table 136. BTE_CONTEXT_START_i

Address Offset	0x0000 0048 + (0x20 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 2048 + (0x20 * i) 0x5C00 2048 + (0x20 * i)	**Instance**	ISS_BTE_CORTEX-M3 ISS_BTE_L3
Description	Top-left corner of the context.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																X								RESERVED							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:7	X	Address, in 128-byte words	RW	0x000
6:0	RESERVED	Reserved	R	0x00

2.8.5.13 BTE_CONTEXT_END_i

Table 137. BTE_CONTEXT_END_i

Address Offset	0x0000 004C + (0x20 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 204C + (0x20 * i) 0x5C00 204C + (0x20 * i)	**Instance**	ISS_BTE_CORTEX-M3 ISS_BTE_L3
Description	Bottom-right corner of the context.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			Y													X								RESERVED							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Reserved	R	0x0
28:16	Y	Last line number for the context (0 corresponds to a context of 1 line)	RW	0x0000
15:4	X	Address, in 128-bit words, of the last column of the context	RW	0x000
3:0	RESERVED	Reserved	R	0x0

2.9 ISS CBUFF

2.9.1 ISS CBUFF Environment

There are no particular environment attributes. See Section 2.2, *ISS Interfaces Environment*.

2.9.2 ISS CBUFF Integration

Figure 89 shows the integration of the CBUFF in the ISS. Because the CBUFF maps a virtual memory space from the physical memory, it therefore communicates with the ISS ISP and ISS interface modules for data to and from memory. This figure shows the normal data flow for further processing by the ISP and/or still-image coprocessor (SIMCOP) (in green) and the stall functionality of CBUFF (in red), which stall the data for a certain amount of time. An example of stalling is when the ISP processes the data faster than the input from SC to memory. The CBUFF then must stall the data flow until a sufficient amount of data can be read from memory by the ISP.

For power domain, clocks, reset, and hardware requests, see Section 1.2.5, *ISS Power Management*.

2.9.2.1 ISS CBUFF Reset and Idle Mechanism

A reset signal is provided by the PRCM module to the top-level ISS power and clock-management module.

For standby, when none of the ISS modules require CBUFF execution actions, ISS PM executes the standby sequence, which reaches the CBUFF. When an IDLE request arrives, the CBUFF data is drained. After all transactions are drained, the CBUFF acknowledges the IDLE request to the ISS PM/CM. The IDLE request/acknowledge steps are:

When an IDLE request is received in smart-idle mode:

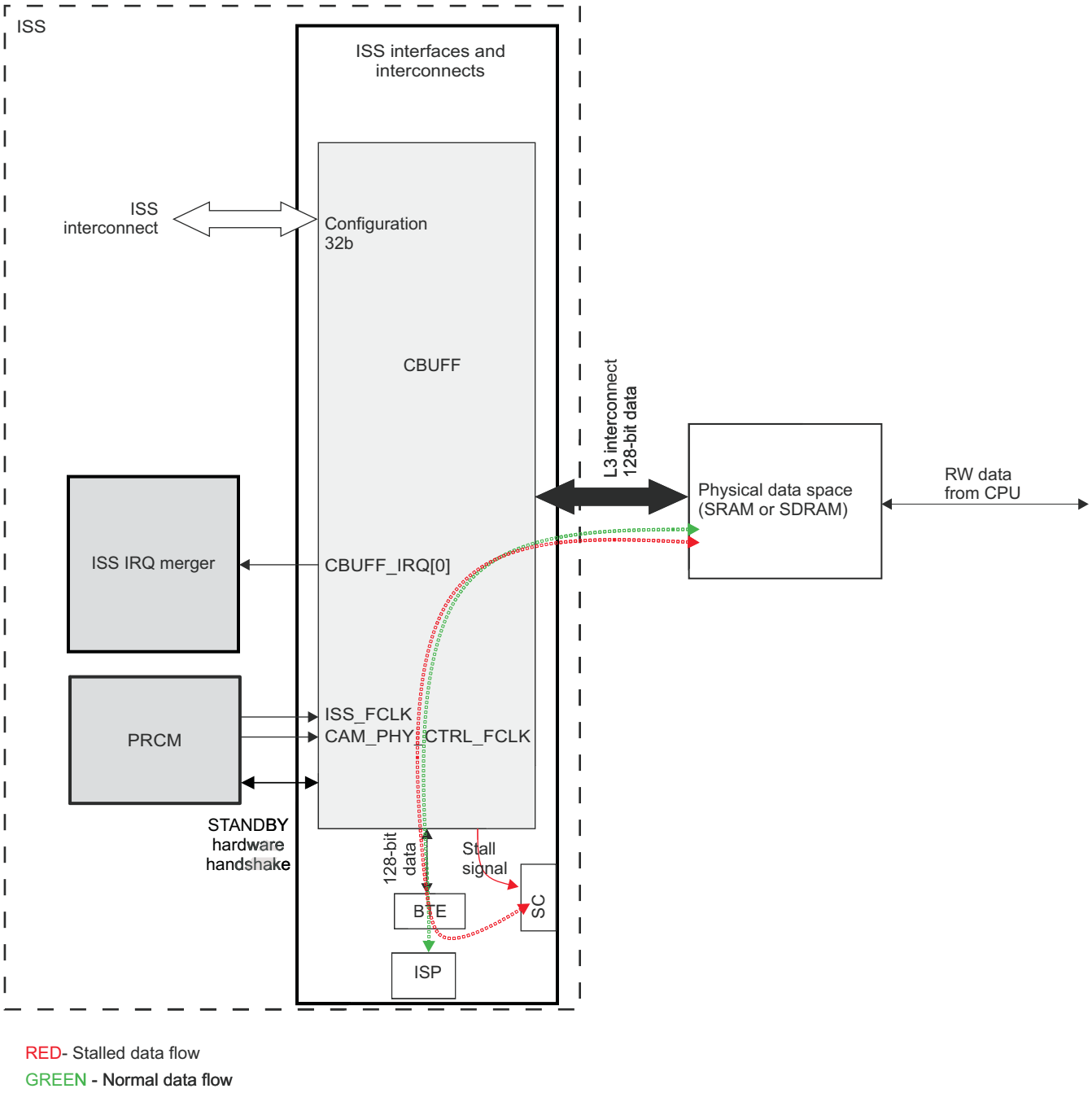
1. CBUFF stops accepting any new OCP requests on a clean OCP transaction boundary.
2. Waits until the interrupt output becomes inactive (no more enabled event pending)
3. Waits until all outstanding OCP transactions are complete on OCPO and OCPI. CBUFF keeps track of issued requests, data phases, and responses for that purpose.
4. CBUFF disconnects the OCPO port.
5. CBUFF acknowledges the IDLE request.

When a wake-up request is received in smart-idle mode:

1. CBUFF connects the OCPO port.
2. Starts accepting new requests from OCPI
3. Acknowledges the functional state

Idle mode is controlled through the CBUFF_HL_SYSCONFIG[3:2] IDLEMODE bit field. For software reset, it is recommended to use the global ISS reset; if a reset is required, it can be set from the CBUFF_HL_SYSCONFIG[0] SOFTRESET bit.

Figure 89. ISS CBUFF Integration



2.9.2.2 ISS CBUFF Interrupts

All events generated by the module are merged into a single event at ISS level. This event can be mapped to the MPU subsystem by enabling the ISS_HL_IRQENABLE_SET_i[10] CBUFF_IRQ bit. Table 138 lists the procedure to manage CBUFF interrupts.

Table 138. ISS CBUFF Interrupt Management

Description	Bit Field
Each event that generates an interrupt can be individually enabled by setting the appropriate bit.	CBUFF_HL_IRQENABLE_SET
Each event that generates an interrupt can be individually disabled by setting the appropriate bit.	CBUFF_HL_IRQENABLE_CLR
When an event occurs, the corresponding bit in the CBUFF_HL_IRQSTATUS_RAW register is set regardless of whether or not the event is enabled. Bits in the CBUFF_HL_IRQSTATUS registers are only set only when an enabled event occurs	CBUFF_HL_IRQSTATUS_RAW and CBUFF_HL_IRQSTATUS
Software can clear a pending event by setting the appropriate bit in the CBUFF_HL_IRQSTATUS register.	CBUFF_HL_IRQSTATUS

The CBUFF can generate three events per context and one global event. All events are merged into one physical interrupt line. Table 139 describes the CBUFF-generated events.

Table 139. ISS CBUFF-Generated Events

Event	Description
IRQ_CTXx_READY	Write mode: CPU can read data from the physical window pointed by CBUFF_CTX_STATUS_i[3:0] WB.
	mode: The OCPI initiator has completed writing a physical window.
IRQ_CTXx_INVALID	Invalid access
	OCPI writes the virtual space of context i in read mode.
	OCPI reads the virtual space of context i in write mode.
	OCPI writes the virtual space of context i outside the CBUFF_CTX_STATUS_i[11:8] WA window in write or read/write mode.
	OCPI reads the virtual space of context i outside the CBUFF_CTX_STATUS_i[11:8] WA window in read mode.
	OCPI reads the virtual space of context i outside the CBUFF_CTX_STATUS_i[3:0] WB window in read/write mode.
	CPU writes the DONE bit when physical windows are not ready for the CPU.
	This event indicates a wrong configuration of the CBUFF, the OCPI initiator or bogus software. When it happens, context i goes into an error state. In this state all accesses to the virtual space of context i are cancelled: they are not forwarded to the physical space. The purpose is to prevent corruption of the physical memory. Of course, the CBUFF still returns OCP responses to OCPI to ensure the integrity of the OCP.
	The error state can be left by disabling the context i and re-enabling it. Before doing so, software must ensure that there are no more outstanding requests to the virtual space of context i.
IRQ_CTXx_OVR	Physical space overflow or underflow event This event indicates a bandwidth mismatch between data producer and data consumer. When it happens, context i does not go into error state. However, the data in the physical space is likely to be corrupted.
IRQ_OCP_ERR	OCP error received in the OCPO master port. The OCP response is forwarded to OCPI normally.

2.9.3 ISS CBUFF Functional Description

The CBUFF maps a virtual space to a physical space by address translation. It does not change the data or store it locally.

2.9.3.1 ISS CBUFF Features

The ISS CBUFF features are:

- Fully transparent for accesses out of the configured virtual space
- Three functional modes :
  - Read mode: Read requests received from OCPI and forwarded after translation to OCPO. Writes are handled by an external process and acknowledged by the CPU.
  - Write mode: Write request received from OCPI and forwarded after translation to OCPO. Reads are handled by an external process and acknowledged by the CPU.
  - Read/write mode: Read and write requests received from OCPI and forwarded after translation to OCPO
- Four independent contexts
- Virtual address space (linear) mapped into a physical space (circular)
- Maximum physical buffer size of 16 × 16MB:
  - Physical space consists of 2, 4, 8, or 16 windows.
  - Maximum allowed window size is 16MB.
- Support of 2D addressing modes
- Strong error detection mechanisms to prevent data corruption caused by bogus configuration
- Addresses are 128-bit aligned, but window fill level managing is byte accurate
- Bandwidth control feedback loop to stall in initiator connected to OCPI

2.9.3.2 ISS CBUFF Functional Description

The CBUFF maps a virtual address space to a physical space also called circular buffer.

The CBUFF can handle up to eight contexts. For most data formats, primarily four contexts are used. In cases where YUV4:2:0 data is exchanged, two contexts are consumed by the SIMCOP (JPEG encode) coming from the resizer module, which is the ISP output module. Moreover, a context is a virtual full-frame buffer that maps to a configurable number of physical windows.

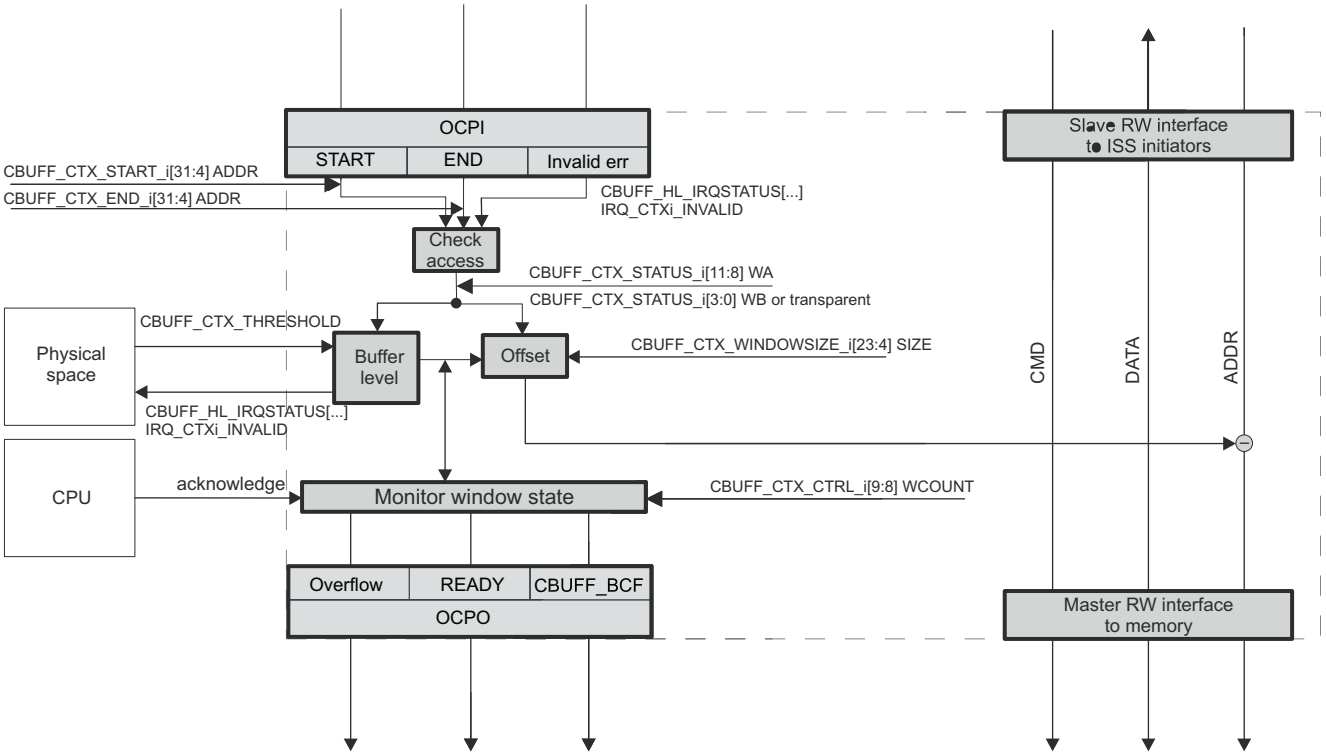
This section gives an overview of typical uses of the CBUFF.

2.9.3.2.1 ISS CBUFF Top-Level Diagram

Figure 90 shows the functional principle diagram. It does not include an exhaustive list of the interface signals or internal status registers.

A more detailed functional description is provided in the following sections.

Figure 90. ISS CBUFF Top-Level Diagram



2.9.3.2.2 ISS CBUFF Functional Modes

The CBUFF supports three functional modes (see Table 140).

Table 140. ISS CBUFF Functional Modes

Mode	Data Written by	Data Read by
Write mode	ISS initiator	CPU-controlled process
Read mode	CPU-controlled process	ISS initiator
Read/write mode	ISS initiator	ISS initiator

2.9.3.2.2.1 ISS CBUFF Write Mode

In write mode, the physical space is written by the CBUFF and it is read by the CPU. An IRQ_CTXx_READY event is set each time a physical window is available to be read by the CPU. This happens when the CBUFF_CTX_STATUS_i[11:8] WA pointer is moved by the CBUFF.

The CBUFF sets an IRQ_CTXx_READY event to inform the CPU that it can access the CBUFF_CTX_STATUS_i[3:0] WB physical window. The CBUFF cannot monitor CPU accesses to the physical window. The CPU must indicate when it has completed the processing of the CBUFF_CTX_STATUS_i[3:0] WB window by setting the CBUFF_CTX_CTRL_i[2:1] DONE bit field. This increments the window index CBUFF_CTX_STATUS_i[3:0] WB by one modulo the window count (defined by the CBUFF_CTX_CTRL_i[9:8] WCOUNT bit field).

The CBUFF ensures that one IRQ_CTXx_READY event is sent to the CPU per physical window to be read. In other words, when a new IRQ_CTXx_READY event occurs before the previous one(s) is acknowledged by the CPU, it is not lost. The CBUFF memorizes the event and triggers the interrupt line again when the CPU clears it.

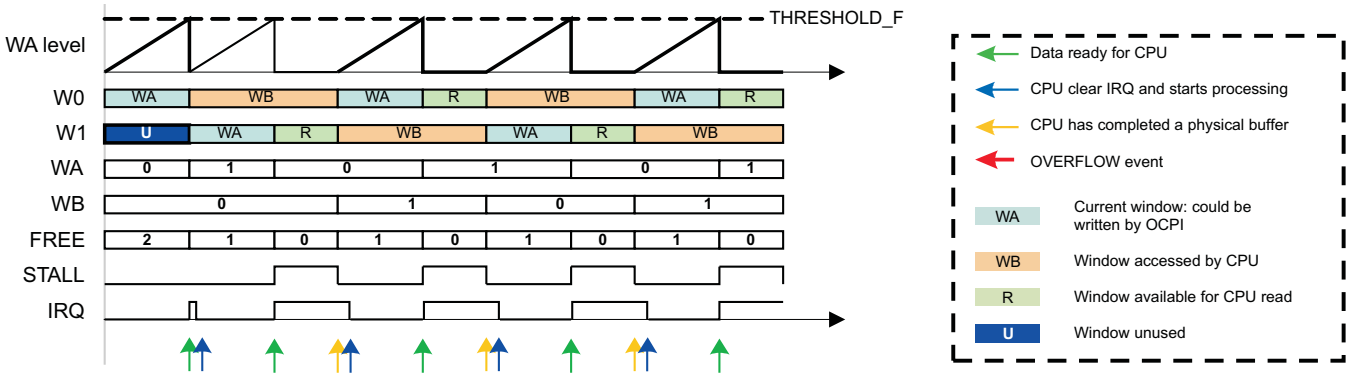
When the CPU reads the physical space too slowly, the CBUFF_CTX_STATUS_i[11:8] WA window pointer falls into the CBUFF_CTX_STATUS_i[3:0] WB window. This generates an IRQ_CTXx_OVR event when the OCPI initiator performs an access to that window. The CPU receives the IRQ_CTXx_READY event only if it is allowed to access a physical window. Therefore, it cannot generate an IRQ_CTXx_OVR event in a normal case. The OCPI initiator accesses are tracked based on activity on the OCP port.

When an IRQ_CTXx_OVR event occurs, the buffer content is likely to be corrupted. However, the CBUFF context continues processing data normally; it does not go into error state. Software must reset data generation and the CBUFF in a clean manner.

Figure 91 shows an example scenario with CBUFF_CTX_CTRL_i[9:8] WCOUNT = 1 (four windows). In normal operation, the CPU processes data at least at the same speed as it is written to the physical space. When this is not true, the number of windows to be read by the CPU increases. When no more physical windows are available for OCPI writes, an overflow occurs.

In the following example, the CPU takes more time than expected to read the third buffer. Therefore, physical windows are not freed up. The OPCI initiator continues to write data into the physical window. That leads to an overflow when the OCPI initiator writes into the physical window that is read by the CPU (CBUFF_CTX_STATUS_i[11:8] WA = CBUFF_CTX_STATUS_i[3:0] WB and writes into CBUFF_CTX_STATUS_i[11:8] WA detected).

Figure 91. ISS CBUFF Write Mode CPU Interaction Example



(1) When there is no physical window available to be read by the CPU and the CPU writes the DONE bit, an IRQ_CTXx_INVALID event occurs.

(2) The bandwidth control feedback (BCF) feature can be used to prevent overflow. It must be supported by the module writing data into the virtual space (typically, an ISS).

2.9.3.2.2.2 ISS CBUFF Read Mode

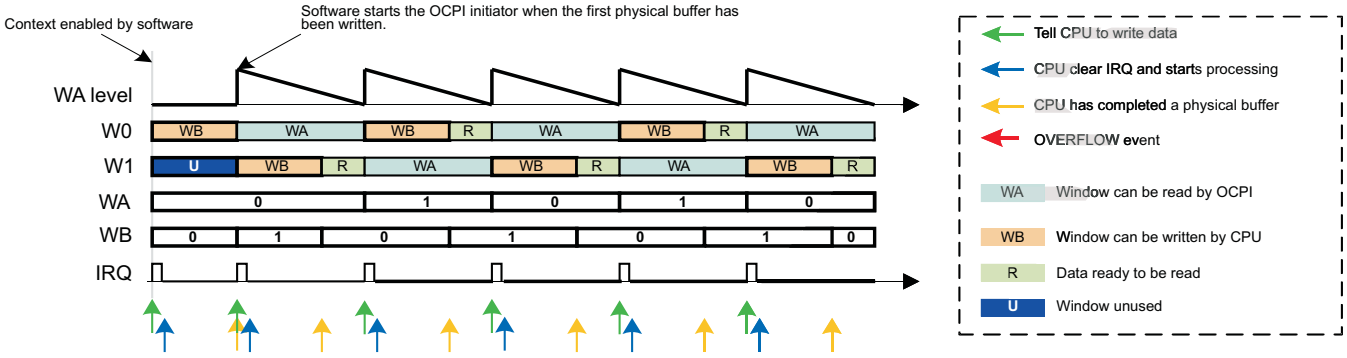
In read mode, the physical space is read by the CBUFF and it is written by the CPU. An IRQ_CTXx_READY event is set each time a physical window is available to be written by the CPU. This is true until all free buffers are used.

The CBUFF sets an IRQ_CTXx_READY event to inform the CPU that it can write the CBUFF_CTX_STATUS_i[3:0] WB window. The CBUFF cannot monitor CPU accesses to the physical space. It must indicate when it completes writing the CBUFF_CTX_STATUS_i[3:0] WB window by setting the CBUFF_CTX_CTRL_i[2:1] DONE bit field. This increments the CPU window index CBUFF_CTX_STATUS_i[3:0] WB by one modulo the window count (defined by the CBUFF_CTX_CTRL_i[9:8] WCOUNT bit field).

The CBUFF ensures that one IRQ_CTXx_READY event is sent to the CPU per physical window to be written. In other words, when a new IRQ_CTXx_READY event occurs before the previous one(s) is acknowledged by the CPU, it is not lost. The CBUFF memorizes the event and triggers the interrupt line again when the CPU clears it.

The following is an example of a normal operation in which the CPU writes data faster than it is read by the OCPI initiator. Figure 92 uses two physical windows (CBUFF_CTX_CTRL_i[9:8] WCOUNT = 0).

Figure 92. ISS CBUFF CPU Writes Data Faster Than it Is Read by the OCPI Initiator



Software must enable the ISS initiator only when at least one window is written by the CPU. Otherwise, an OVERFLOW event occurs when BCF is not used.

The CPU receives an interrupt each time a physical window is available to receive data. It clears the interrupt and then starts filling the physical window. When the buffer is completely written, it sets the CBUFF_CTX_CTRL_i[2:1] DONE bit field. This happens before the OCPI initiator has read all data from the previous physical window: the CPU must wait until the next IRQ_CTXx_READY event is received before it can write again to the physical space.

When the CPU writes buffers too slowly, the CBUFF_CTX_STATUS_i[11:8] WA window falls into the CBUFF_CTX_STATUS_i[3:0] WB window. This generates an IRQ_CTXx_OVR event only when the OCPI initiator performs reads to that window. The CPU receives the IRQ_CTXx_READY event only if it is allowed to access a physical window. Therefore, it does not generate an IRQ_CTXx_OVR event in a normal case. ISS accesses are tracked based on activity on the OCPI port.

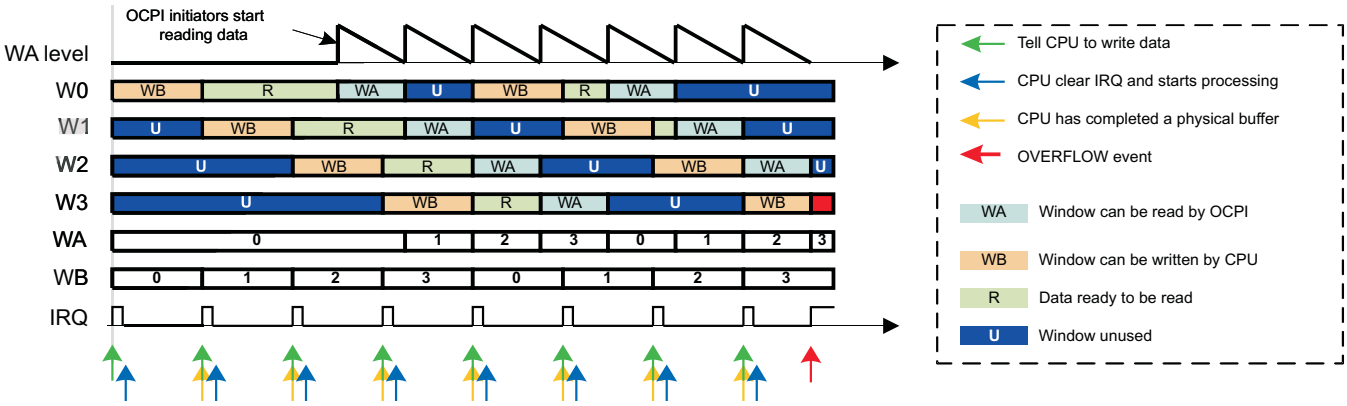
When no buffer is available to be written by the CPU and the CPU writes the DONE bit, an IRQ_CTXx_INVALID event occurs.

When an IRQ_CTXx_OVR event occurs, the OPCI initiator reading the virtual space is likely to receive dummy data. However, the CBUFF context continues to process data normally; it does not go into error state. Software must reset the data consumer (OCPI initiator) and CBUFF context in a clean manner.

Figure 93 is an example of CBUFF read mode CPU interaction.

The BCF feature can be used to prevent overflow. It must be supported by the module reading data from the virtual space (typically, an ISS).

Figure 93. CBUFF Read Mode CPU Interaction Example



2.9.3.2.2.3 ISS CBUFF Read/Write Mode

Reads and writes are performed by OCPI initiators such as a camera interface, ISP, or SIMCOP. Address translation is performed for read and write data flows. The WA pointer is used for the write data flow, and the WB pointer is used for the read data flow. Therefore, address translation for the write data flow is the same as for write mode.

In this mode, the OCPI read data flow is stalled when there is not enough data to read in the physical space. A typical application is to store data from the camera in a CBUFF and to read it back by the ISP. When the camera is slower than the ISP, the ISP stalls.

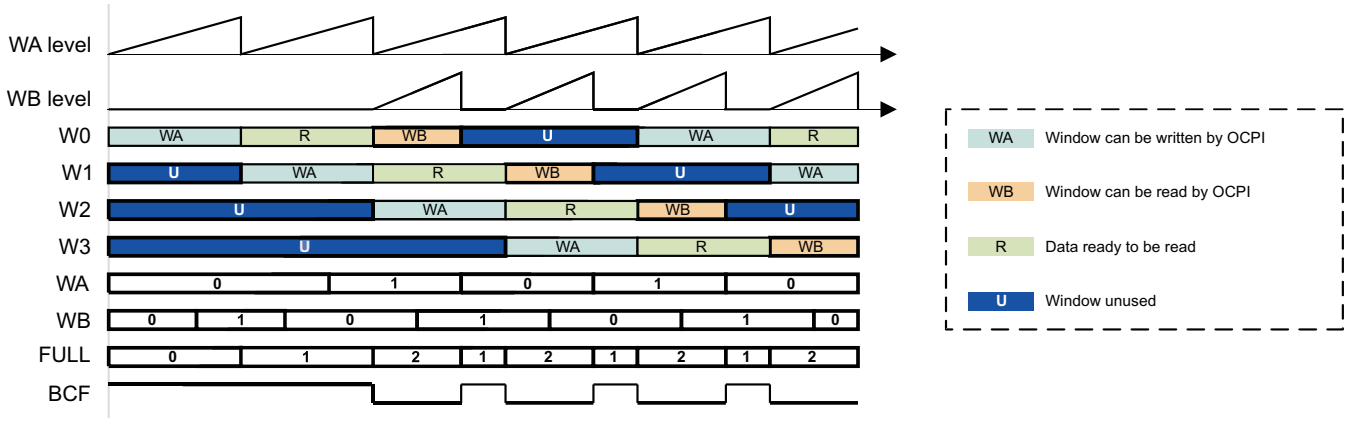
This mode does not rely on CPU synchronization. Synchronization is always performed using the BCF hardware mechanism. IRQ_CTXx_READY events are triggered in this mode when the OPCI write initiator has filled a physical window. Software can use this event for debug or performance bench marking purposes. Writes to the CBUFF_CTX_CTRL_i[2:1] DONE bit field are ignored in this mode.

The IRQ_CTXx_OVR event is triggered when an underflow occurs. In a normal case this should not occur, because the BCF signal is used to stall the read data flow when insufficient data is available in the physical space.

Figure 94 assumes:

- Read/write mode
- Four physical windows
- CBUFF_CTX_THRESHOLD_S_i = CBUFF_CTX_THRESHOLD_F_i
- The OCPI read initiator is faster than the OCPI write initiator. The BCF feature is used to stall the read initiator. CBUFF_CTX_CTRL_i[7:4] BCF = 2

Figure 94. ISS CBUFF Read/Write Mode Example



2.9.3.2.3 ISS CBUFF Events and Status Checking

2.9.3.2.3.1 ISS CBUFF Operations

A CBUFFx_READY_IRQ event is generated each time the CPU can read data from the CBUFF. The CPU can clear the event when it starts processing the data to avoid masking other events. The CPU can keep track of the location on the data internally or use the CBUFF registers to compute it.

The formula used is: ADDR = CBUFF_CTX_STATUS_i[3:0] WB CBUFF_CTX_WINDOWSIZE_i + CBUFF_CTX_PHY_i

When the CPU is done with processing, it must free the buffer by setting the CBUFF_CTX_CTRL_i[10] DONE bit. Otherwise, an overflow event may occur.

The CBUFF does not keep track of EOF events. They must be managed by the CPU using the EOF event of the module that writes into the CBUFF. At the EOF, data may remain in the current write windows. For example, when the window size is set to 8 lines and the image size is 20 lines, only two window-ready events are generated for a linear addressing scheme. The remaining four lines can be read after the EOF event.

No automatic reset of the CBUFF FSM occurs at the end of the image frame. Software must reset the context by clearing the CBUFF_CTX_CTRL_i[0] ENABLE bit when the frame is completely processed. A new frame can start only when the CBUFF_CTX_CTRL_i[0] ENABLE bit is set.

2.9.3.2.3.2 ISS CBUFF Status Checking

The CBUFF provides read-only access to the CBUFF_CTX_STATUS_i[11:8] WA and CBUFF_CTX_STATUS_i[3:0] WB pointers through the CBUFF_CTX_STATUS_i register. For example, the CBUFF_CTX_STATUS_i[3:0] WB index can be used by the CPU to compute the address of a physical window. Those indexes can also be used to evaluate latency margins.

2.9.3.2.3.3 ISS CBUFF Register Accessibility During Frame Processing

All registers are busy-writeable registers. These registers/fields can be read or written even if the module is busy. Changes to the underlying settings occur instantly. However, module behavior is unpredictable when registers are changed during processing.

For correct operation, software must:

1. Disable all accesses to the virtual space managed by the context.
2. Disable the context by clearing the CBUFF_CTX_CTRL_i[0] ENABLE bit.
3. Change the configuration.
4. Re-enable CBUFFx by setting the CBUFF_CTX_CTRL_i[0] ENABLE bit.

2.9.3.2.4 ISS CBUFF Memory-to-Memory Operation BCF

The BCF mechanism matches the bandwidth between two processes.

The BCF feature can be used in all three CBUFF modes:

- Read mode: A CPU-controlled process writes data into physical space. The BCF signal is deasserted when the physical space contains enough data to start the read initiator connected to OCPI.
- Write mode: The BCF signal controls the write initiator connected to OCPI. When the CPU-controlled process does not read the data fast enough from the physical space, the BCF signal is asserted to stall filling of the buffer. It is deasserted when enough space is available in the physical space.
- Read/write mode: The BCF signal controls the read initiator connected to OCPI. Another initiator connected to OCPI fills the physical space. The BCF signal is deasserted when the physical space contains data that can be read by the read initiator. It is deasserted when insufficient data is available in the physical space.

The CBUFF_BCF output is controlled based on two factors:

- The window count available for the OCPI initiator to write
- The amount of data in the last available window (pretrigger)

The CBUFF_BCF signal is enabled by the CBUFF_CTX_CTRL_i[7:4] BCF bit field. It defines:

- Write mode: The amount of required free windows to allow writing from the ISS. In other words, when less than the required amount of BCF windows is available for ISS writes, the stall mechanism is triggered. The number of free windows is initialized to the total window count of the context. It is decreased by 1 each time the OCPI initiator finishes writing a window. It is increased by 1 each time the CPU finishes reading a window.
- Read and read/write modes: The minimum amount of required full windows to allow reading from the OCPI. When fewer than CBUFF_CTX_CTRL_i[7:4] BCF windows are available for ISS read, the stall mechanism is triggered. The number of full windows is initialized to 0. It is decreased by 1 each time the OCPI initiator completes reading a window. It is increased by 1 each time the CPU/OCPI initiator finishes writing a window.

Figure 95 is an example of BCF use. It assumes:

- Write mode: OCPI writes data into physical space and the CPU reads it.
- Two physical windows
- CBUFF_CTX_THRESHOLD_S_i = CBUFF_CTX_THRESHOLD_F_i
- The OCPI write initiator is faster than the CPU.
- CBUFF_CTX_CTRL_i[7:4] BCF = 1

Figure 95. ISS CBUFF Write Mode CPU Interaction Example BCF Used

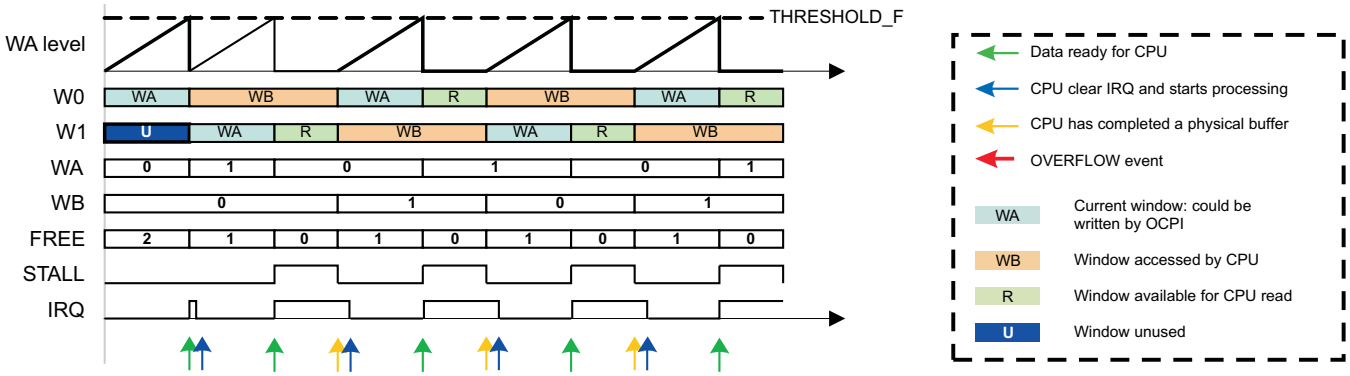


Figure 96 is another example. It assumes:

- Read mode: The CPU writes data into physical space and the OCPI reads it.
- Four physical windows
- CBUFF_CTX_THRESHOLD_S_i = CBUFF_CTX_THRESHOLD_F_i
- The OCPI read initiator is faster than the CPU. It starts after the CPU.
- CBUFF_CTX_CTRL_i[7:4] BCF = 2

Figure 96. ISS CBUFF Read Mode CPU Interaction Example (1)

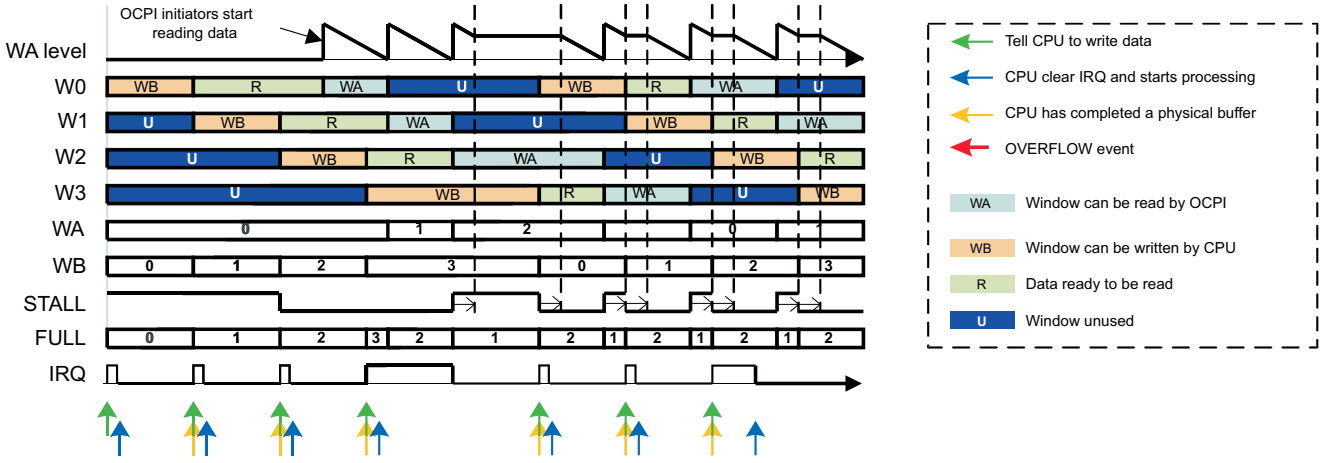


Figure 97 is another example. It assumes:

- Read mode: The CPU writes data into physical space and the OCPI reads it.
- Two physical windows
- CBUFF_CTX_THRESHOLD_S_i = CBUFF_CTX_THRESHOLD_F_i
- The OCPI read initiator is faster than the CPU. It starts after the CPU.
- CBUFF_CTX_CTRL_i[7:4] BCF = 1

Figure 97. ISS CBUFF Read Mode CPU Interaction Example (2)

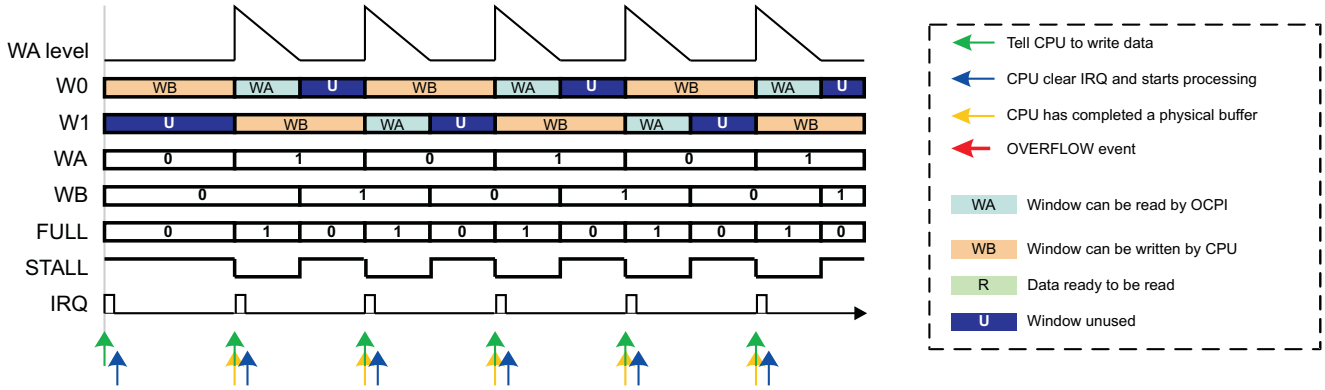


Figure 98 is another example. It assumes:

- Read mode: The CPU writes data into physical space and the OCPI reads it.
- Four physical windows
- CBUFF_CTX_THRESHOLD_S_i = CBUFF_CTX_THRESHOLD_F_i
- The OCPI read initiator is faster than the CPU. It starts after the CPU.
- CBUFF_CTX_CTRL_i[7:4] BCF = 1

CBUFF supports pretriggering the BCF signal for finer latency compensation control. This mechanism is active only for the last window available for OCPI access. It typically improves the physical space use, which is particularly useful when on-chip SRAM is used as a ping-pong buffer.

Pretriggering is controlled through the CBUFF_CTX_THRESHOLD_S_i register. It defines:

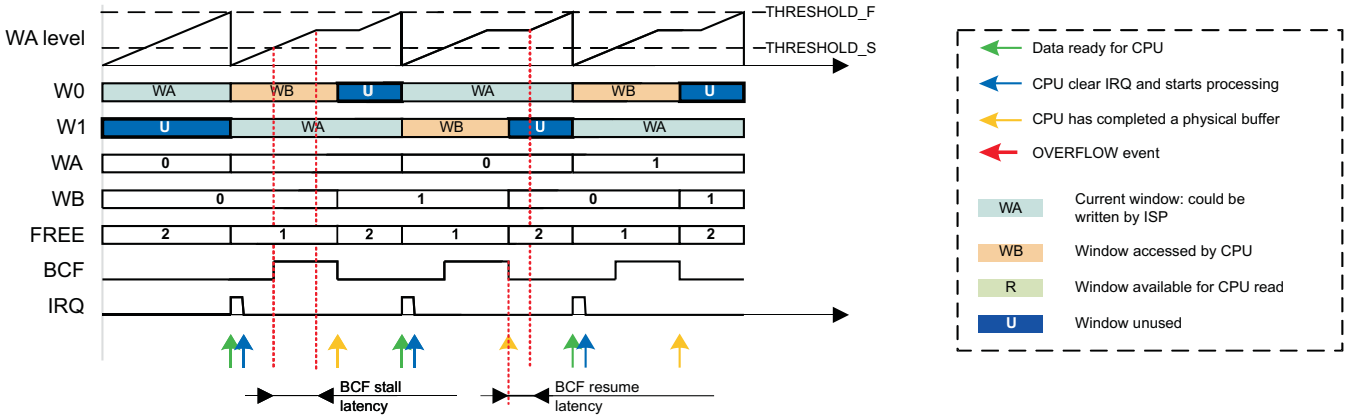
- Write mode: When the fill level of the last window available for ISS writes is greater than or equal to CBUFF_CTX_THRESHOLD_S_i, the CBUFF_BCF signal is asserted.
- Read mode: When the amount of data in the last window available for ISS reads is less than CBUFF_CTX_THRESHOLD_F_i/CBUFF_CTX_THRESHOLD_S_i, the CBUFF_BCF signal is asserted.

BCF pretriggering is disabled by setting CBUFF_CTX_THRESHOLD_S_i = CBUFF_CTX_THRESHOLD_F_i.

Figure 98 is an example of BCF pretriggering, assuming the following:

- Write mode: The CPU reads data from the physical space and the OCPI writes it.
- Two physical windows
- CBUFF_CTX_THRESHOLD_S_i = 1/3 CBUFF_CTX_THRESHOLD_F_i
- CBUFF_CTX_CTRL_i[7:4] BCF = 1

Figure 98. ISS CBUFF BCF Pretrigger Example: Write Mode

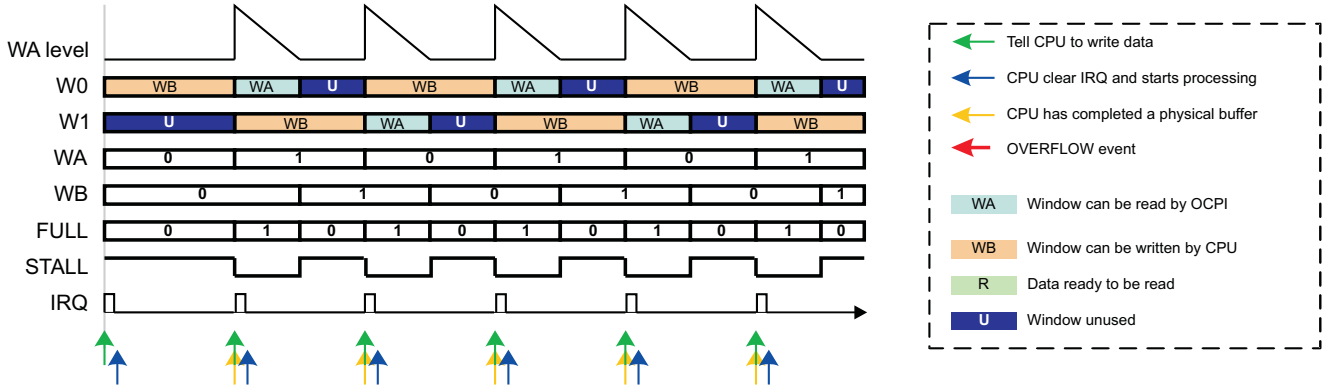


The CBUFF_BCF output signal is a logical OR of all internal BCF signals of CBUFF contexts. For example, when contexts 0 and 1 have enabled the BCF feature, both contexts can request a data flow stall.

Figure 99 is an example of pretriggering in read mode:

- Read mode: CPU writes data into the physical space and the OCPI reads it.
- Four physical windows
- CBUFF_CTX_THRESHOLD_S_i = 2/3 CBUFF_CTX_THRESHOLD_F_i
- The OCPI read initiator is faster than the CPU. It starts after the CPU.
- CBUFF_CTX_CTRL_i[7:4] BCF = 2

Figure 99. ISS CBUFF BCF Pretrigger Example: Read Mode



2.9.3.2.5 ISS CBUFF TILER Support

The CBUFF can translate 2D (BLOCK) bursts intended for the TILER. However, software must ensure that a given burst fits in a window.

The expected value for ADDR[32] is defined by the CBUFF_CTX_CTRL_i[11] TILERMODE bit.

- When ADDR[32] = CBUFF_CTX_CTRL_i[11] TILERMODE, the access is processed normally. ADDR[32] is not used for further processing.
- Otherwise, the access is treated as transparent.

2.9.3.2.6 ISS CBUFF Window Management and Address Remapping Details

This section explains the internal address remapping and windows management algorithm. Internally, the module maintains some variables in addition to the configuration registers. The module manages multiple contexts in parallel. Table 141 lists the CBUFF internal variables.

Table 141. ISS CBUFF Internal Variables

Variable	Description
WAx	Current window index for context x. Possible values are 0 to allowed window count. The current value can be read using the CBUFF_CTX_STATUS_i[11:8] WA bit field.
WBx	Window in the physical space that can be accessed by the CPU in read/write modes . Window that is read from OCPI in read/write mode. Possible values are 0 to allowed window count. The current value can be read using the CBUFF_CTX_STATUS_i[3:0] WB bit field.
VPAx	Start address, in the virtual space, of the CBUFF_CTX_STATUS_i[11:8] WA window Used as a base pointer for the read (read mode) or write data flow (write , read/write modes) This is an internal quantity that cannot be accessed by software.
VPBx	Start address, in the virtual space, of the CBUFF_CTX_STATUS_i[3:0] WB window Used as a base pointer for the read data flow. This is an internal quantity that cannot be accessed by software.
OFFSETAx	This is an internal quantity that cannot be accessed by software.
OFFSETBx	Address offset used when WAx or WBx is accessed
LEVELAx	This is an internal quantity that cannot be accessed by software.
LEVELBx	Address offset used when WAx or WBx is accessed

2.9.3.2.6.1 ISS CBUFF Startup

The status of a CBUFF context is reset when it is disabled. This does not affect the configuration registers or the CBUFF_IRQSTATUS register. Table 142 lists the internal state after reset.

Table 142. ISS CBUFF Internal State After Reset

Variable	Description
WAx	0
WBx	0
VPAx VPBx	CBUFF_CTX_START_i
OFFSETAx OFFSETBx	CBUFF_CTX_START_i – CBUFF_CTX_PHY_i
LEVELAx LEVELBx	0

2.9.3.2.6.2 ISS CBUFF Access Identification

For each access to the virtual space (OCPI slave port), the CBUFF first checks the address to classify the transaction into one of the categories listed in Table 143.

Table 143. ISS CBUFF Address identification

Address ID	Variable	Condition
0+2*x	WA_CBUFFx	CBUFF_CTX_CTRL_i[0] ENABLE = 1 and ADDR = VPAx and ADDR VPAx + CBUFF_CTX_WINDOWSIZE_i and ADDR = CBUFF_CTX_END_i and Access type = write when read/write mode is selected
1+2*x	WB_CBUFFx	CBUFF_CTX_CTRL_i[0] ENABLE = 1 and CBUFF_CTX_CTRL_i[2:1] MODE = 2 and Access type = read and ADDR=VPBx and ADDRVPBx + CBUFF_CTX_WINDOWSIZE_i and ADDR= CBUFF_CTX_END_i

Table 143. ISS CBUFF Address identification (continued)

Address ID	Variable	Condition
16+x	ERR_CBUFFx	CBUFF_CTX_CTRL_i[0] ENABLE = 1 and ADDR= CBUFF_CTX_START_i and ADDR= CBUFF_CTX_END_i
24	TRANSPARENT	Always true

Lower IDs correspond to higher priorities if multiple conditions are true. For example, when the current virtual window of CBUFF 0 is accessed, at least the tests for categories WA_CBUFFx and ERR_CBUFFx are true. The final category is WA_CBUFFx, because it has a higher priority.

NOTE: Tests must be performed in parallel to match the desired performance.

Further processing depends on the category:

- TRANSPARENT: Accesses flow though the module without changing its internal state or any translation.
- ERR_CBUFFx: The module goes into error state for the concerned context and sets the CBUFF_HL_IRQSTATUS.IRQ_CTXx_INVALID bit. When the module is in error state for CBUFFx, all accesses to that buffer are cancelled. In other words, any access that has an address between CBUFF_CTX_START_i and CBUFF_CTX_END_i is not transmitted to the OCPO master port. There are two ways to leave the error state:
  - Hardware reset
  - Disable and re-enable the context in error state.

    Accesses outside of the virtual space from the context in error state are not affected.
- WA_CBUFFx and WB_CBUFFx: The internal state is updated and address translation is performed when the performed access type (read or write) is compatible with the current mode (read, write, or read/write mode). Otherwise, a CBUFF_HL_IRQSTATUS.IRQ_CTXx_INVALID event is set and CBUFFx goes into the error state.

2.9.3.2.6.3 ISS CBUFF Address Translation

An offset is selected depending on the access category (see Section 2.9.3.2.6.1, *ISS CBUFF Startup*) and the internal state of the accessed buffer. Table 144 lists possible cases.

Table 144. ISS CBUFF Address Translation

Condition	Address Translation
CBUFF_CTX_CTRL_i[0] ENABLE = 1 and ADDR = CBUFF_CTX_START_i and ADDR = CBUFF_CTX_END_i and CBUFFx in error state	Access cancelled
Category = WA_CBUFFx	ADDROUT = ADDRIN-OFFSETAx
Category = WB_CBUFFx	Read/write mode only ADDROUT = ADDRIN-OFFSETBx
Category = ERR_CBUFFx	Access cancelled
Category = TRANSPARENT	ADDROUT = ADDRIN

2.9.3.2.6.4 ISS CBUFF Window Fill Level

Each time an access is performed into an active window (WA_CBUFFx when context x is enabled) the window level is updated. The corresponding LEVELy is incremented according to the BYTEEN input of the OCPI slave port. All possible BYTEEN patterns, including nonaligned ones, are supported. Table 145 shows some examples. The basic idea is to count the number of 1s in the BYTEEN input for each 128-bit word and to sum the values.

Table 145. ISS CBUFF Window-Level Increment

BYTEEN	LEVELy Increment	Comment
0x0000	0	No access
0x0001	1	8-bit access
0x0002	1	8-bit access
0x0003	2	16-bit access
...	...	...
0x0007	3	24-bit access
...	...	...
0x000F	4	32-bit access
...	...	...
0x00F0	4	32-bit access
...	...	...
0xFFFF	16	128-bit access

The window level is compared to CBUFFx_THRESHOLD. Table 146 lists the situations that may occur:

Table 146. ISS CBUFF Window-Level Comparison(1)

Condition	Description
LEVELAx = CBUFF_CTX_THRESHOLD_F_i	The CBUFF_CTX_STATUS_i[11:8] WA bit field of context x is full (write mode) or empty (read mode). Internal window indexes, levels, and offsets are updated.
LEVELBx = CBUFF_CTX_THRESHOLD_F_i	Read/write mode only Window used to handle the read flow of context x is empty. Internal window indexes, levels, and offsets are updated.

(1) All situations described in Table 146 are mutually exclusive because only one level is updated each cycle.

2.9.3.2.6.5 ISS CBUFF Window Pointer and Offset Update

The following description refers to the update of WA in write mode:

When the current window of a context is full:

- A new window is opened. The update is done in a circular manner: the first physical window in is reused after the last one.
  - WA (WA + 1) modulo the number defined by CBUFF_CTX_CTRL_i[9:8] WCOUNT
  - LEVELA - 0
  - VPA - VPA + CBUFF_CTX_WINDOWSIZE_i
- When the window is moved from the last buffer to the first:
  - OFFSETA - OFFSETA + CBUFF_CTX_WINDOWSIZE_i (CBUFF_CTX_CTRL_i[9:8] WCOUNT + 1)
- Otherwise, OFFSETA does not change.

The algorithm used to update WA in read mode is the same, except that *full* refers to *all data of the window has been read* or *window empty*.

For read/write mode, the same algorithm is used to translate reads to the WBx window. Make the following changes:

- Replace WA with WB.
- Replace LEVELA with LEVELB.
- Replace VPA with VPB.
- Replace OFFSETA with OFFSETB.

2.9.3.2.7 ISS CBUFF Error State

Contexts may go into error state when they receive unexpected accesses. In that case, the CBUFF does not send dummy transactions to the OCPO port. It does not forward the failing transactions to OCPO and returns valid responses (SResp = DVA) on OCPI to preserve the integrity of the OCP.

Responses may be received on OCPO while the CBUFF responds to failing transactions on OCPI. The OCPO response phase is stalled (MRespAccept = 0) during that time to prevent corruption. OCPO cannot be stalled longer than one full OCP transaction. Responses received on OCPO are handled before another internally generated response can be sent back to the OCPI. In that case, command and data phases are eventually stalled. Not stalling OCPO for too long is required to avoid affecting system performance.

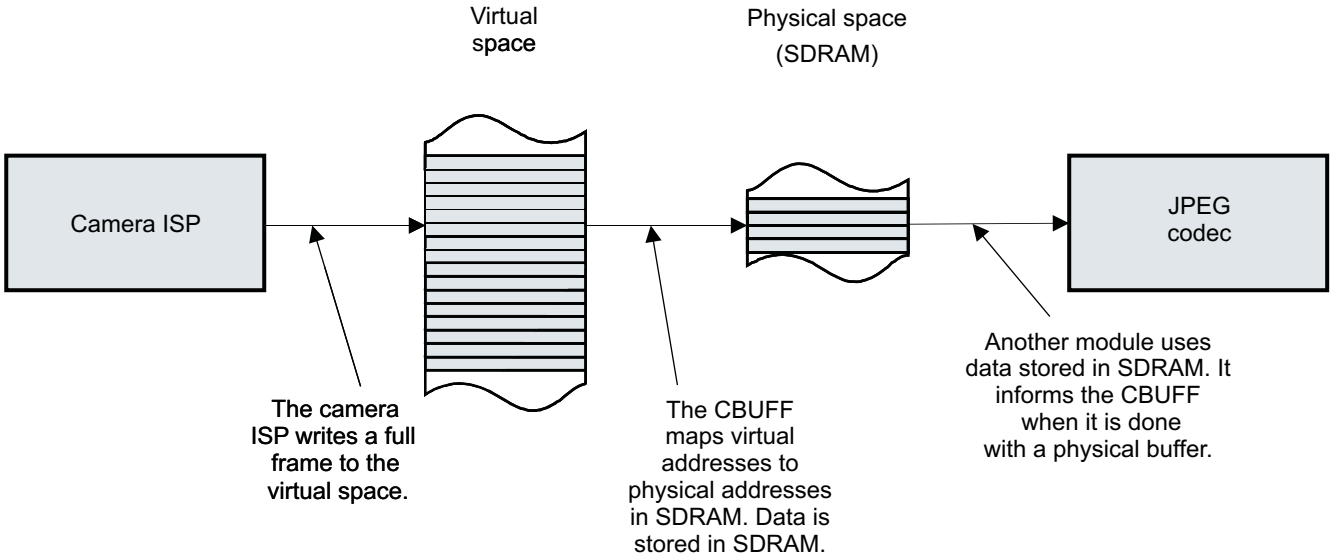
During normal operation (that is, no context is in error state), the OCPO response phase is never stalled.

2.9.3.2.8 ISS CBUFF Typical Configurations

2.9.3.2.8.1 ISS CBUFF Single-Slice Buffer

An OCP master (typically, the ISS) writes data with an incremental addressing scheme to the virtual space. The physical space is smaller than the virtual space. Therefore, physical space locations are read and written multiple times. Figure 100 shows the CBUFF single-slice buffer in write mode.

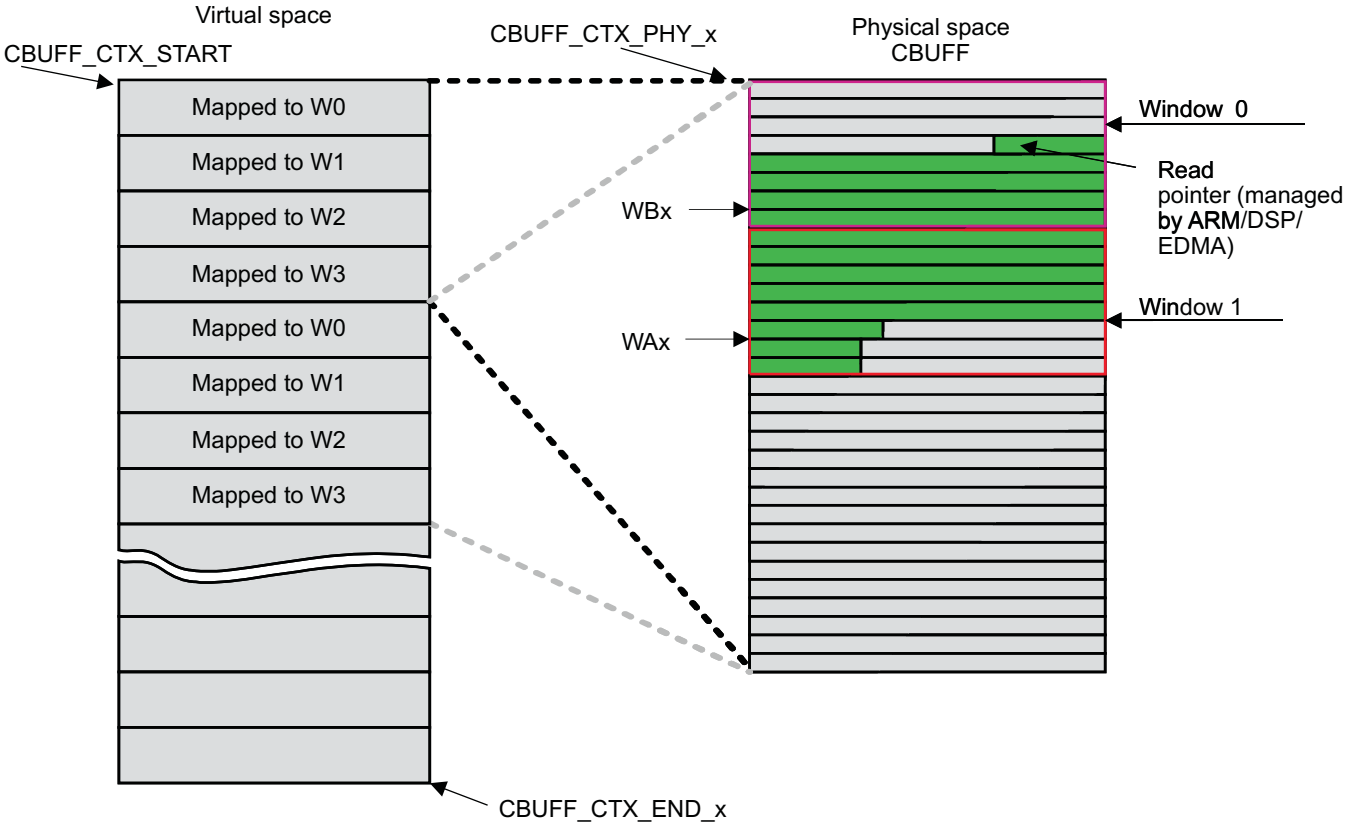
Figure 100. ISS CBUFF Single-Slice Buffer (Write Mode)



Physically, this CBUFF is in on-chip SRAM or SDRAM. The virtual space is defined by a start and end addresses. The physical space is defined by a start address, a window size, and a window count. It is contiguous in memory. When the CPU accesses physical memory for processing, it must know if the SDRAM is available for it to access and if the CBUFF is not using it. The CPU and CBUFF cannot track each other. For example, an interrupt must be triggered from the SIMCOP to the CBUFF to notify it when the processor is done working with SDRAM. For more information about the software configuration for these interrupts, see Section 2.9.3.2.2.1, *ISS CBUFF Write Mode*.

Figure 101 shows the buffer organization for a 4-window buffer.

Figure 101. ISS CBUFF Single-Slice Buffer Example (Write Mode)

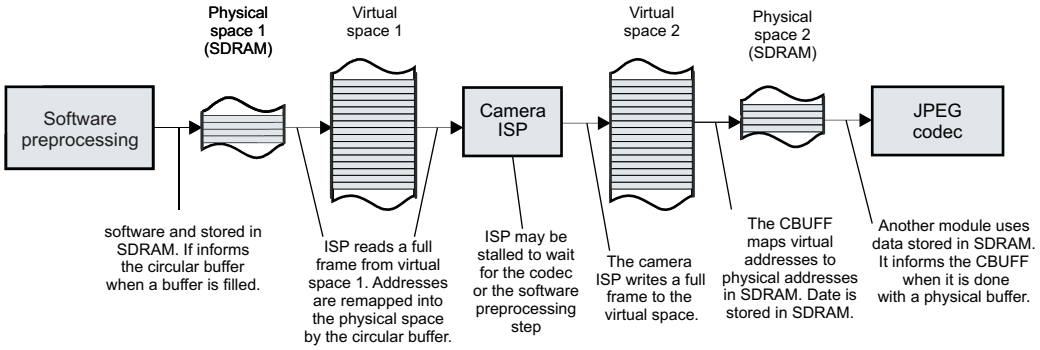


The CBUFF can manage multiple contexts in single-slice mode.

2.9.3.2.8.2 ISS CBUFF Extended-Slice Buffer

In extended-slice mode, at least two contexts managed by the CBUFF are used together. The two contexts provide address translation, one for the read data flow and the other for a write data flow. Figure 102 is an example of the CBUFF extended-slice buffer.

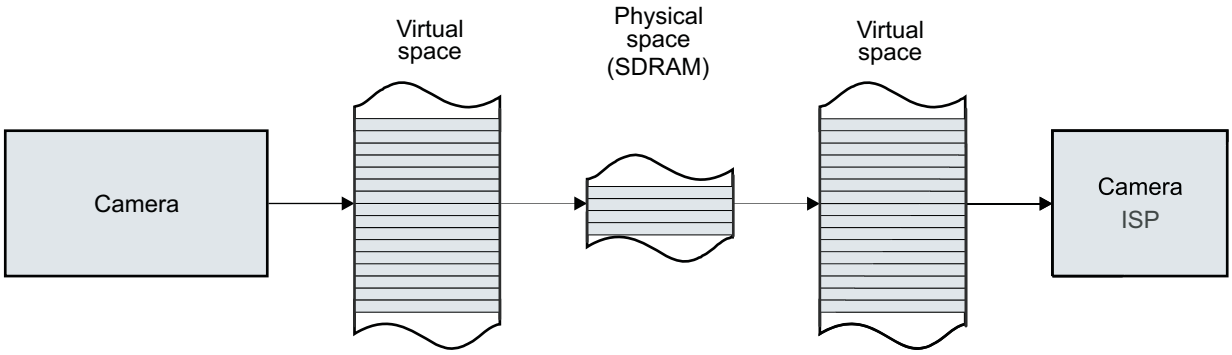
Figure 102. ISS CBUFF Extended-Slice Buffer Example



2.9.3.2.9 ISS CBUFF FIFO Mode

The CBUFF can behave like a FIFO to buffer data between two initiators connected to the OCPI. A typical use case is a camera interface writing data to the FIFO and an ISP reading data from the FIFO. Figure 103 shows the CBUFF FIFO mode.

Figure 103. ISS CBUFF FIFO Mode

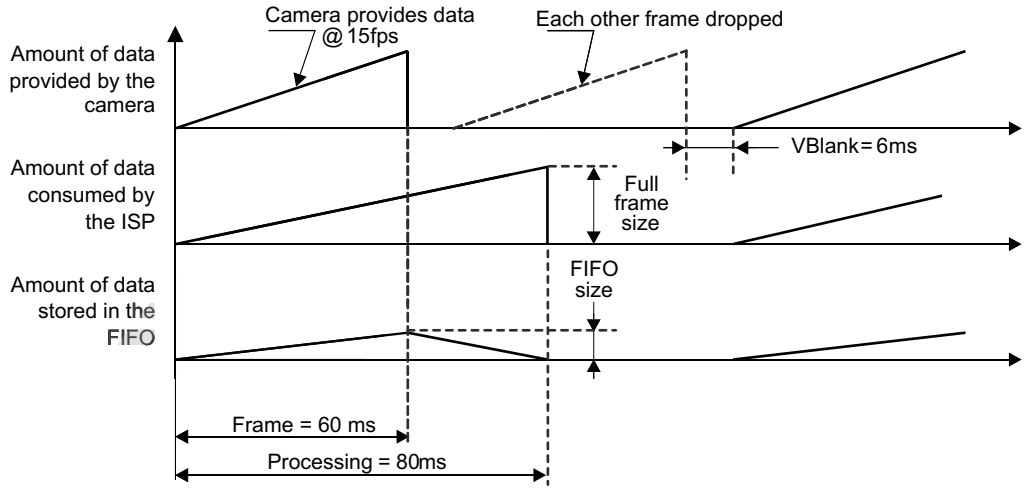


The CBUFF is used as a FIFO when the camera provides data at a higher rate than the ISP can process. To avoid storing one or multiple full-frame buffers, a FIFO buffer is used to store the data that cannot be processed immediately.

For this example, assume that the camera provides 16 MPix at 15 frames per second. A new frame is provided every 66 ms. The ISP needs 80 ms to process this frame. Therefore, the processing cannot be done on the fly, and each other frame must be dropped.

Figure 104 shows the accumulated amount of data provided by the camera since the start of the frame; the amount of data consumed by the ISP since it has started processing the frame; and the difference between the two, which must be stored in the FIFO.

Figure 104. ISS CBUFF FIFO Use Example



The FIFO is emulated as 16 windows (CBUFF_CTX_CTRL_i[9:8] WCOUNT = 0x3). The ISP is started by software when a few kilobytes of data are written to the physical space. Alternatively, the BCF mechanism can be used, but the minimum FIFO size is more complex to calculate.

The FIFO level reaches its maximum when the camera performs the last write of the frame. The camera provides the data at 1 byte/pixel 16 MPix/60 ms = 266 Mbps. The ISP reads data at 1 byte/pixel × 16 MPix/80 ms = 200 Mbps. Therefore, the minimum FIFO size is 60 ms × (266 Mbps – 200 Mbps) = 3.96 Mbps, which is four times less than a full-frame buffer.

Each window holds 256KB of data. The camera starts writing to a new physical window every 256KB/266 Mbps = 962 µs. The ISP starts reading from a new physical window every 1280 µs.

The FIFO mode can also be used when the ISP is faster than the camera. In that case, the ISP is stalled when insufficient data is available in the physical space.

2.9.4 ISS CBUFF Programming Model

2.9.4.1 ISS CBUFF Reset Behavior

Upon hardware or software reset of the CBUFF, all registers in the CBUFF are reset to their reset values

This software reset has the same effect as a hardware reset. ISS high-level software reset ensures that traffic is stopped on clear boundary because the CBUFF is sending data to and from other modules. ISS top-level reset is preferred. For more information about ISS software reset, see Section 1.2.4, *ISS Reset*. Submodule reset is not preferred but is available through the following registers:

1. Set the CBUFF_HL_SYSCONFIG[0] SOFTRESET bit to 1.
2. Read the CBUFF_HL_SYSCONFIG[0] SOFTRESET bit to check whether it is set to 1, which means the reset occurred.

The reset is performed without waiting until all OCP traffic stops. To avoid OCP corruption, software must ensure there is no more ongoing traffic before performing a reset.

2.9.4.2 ISS CBUFF Register Setup

All registers of the context to be used must be initialized for correct operation. Table 147 lists the procedure for the CBUFF register setup.

Table 147. ISS CBUFF Setup Register

Step	Bit Field	Value
Set operation mode.	CBUFF_CTX_CTRL_i[2:1] MODE	0x0: Write mode 0x1: Read mode 0x2: Read/write mode 0x3: Reserved
Define the virtual address range managed by the CBUFF. It usually corresponds to the address region where one image frame is written by the OCPI initiator.	CBUFF_CTX_START_i and CBUFF_CTX_END_i	
Define the start address of the physical buffer.	CBUFF_CTX_PHY_i	
Set the window count and size. The window size usually depends on the use of the buffer. 8 or 16 video lines correspond to a current size for JPEG video compression. A higher window count provides better latency-related overflow protection.	CBUFF_CTX_CTRL_i[9:8] WCOUNT and CBUFF_CTX_WINDOWSIZE_i	
When the 2D addressing capability is not used, set to the window size in CBUFF_CTX_THRESHOLD_F_i. Otherwise, it is set to a smaller value depending on the buffer organization. For example, when each window corresponds to lines by 4096 pixels, but the ISP sends lines of only 2560 pixels, CBUFF_CTX_WINDOWSIZE_i = 8 4096 and CBUFF_CTX_THRESHOLD_F_i = 8 2560.	CBUFF_CTX_THRESHOLD_F_i	
BCF signal-generation configuration is optional.	CBUFF_CTX_THRESHOLD_S_i and CBUFF_CTX_CTRL_i[7:4] BCF	
Enable the module. It can be disabled by clearing the ENABLE bit. This must be done only when there are no more outstanding requests to the virtual space managed by CBUFFx. All internal FSMs and counters of the CBUFF are reset when it is disabled. Pending interrupts are not affected.	CBUFF_CTX_CTRL_i[0] ENABLE	

2.9.5 ISS CBUFF Registers

Table 148 lists the CBUFF instance.

Table 148. ISS CBUFF Instance Summary

Module Name	Base Address Cortex-M3 Private Access	Base Address L3 Interconnect	Size
ISS_CBUFF	0x5504 1800	0x5C00 1800	512 bytes

Table 149 summarizes the CBUFF registers.

Table 149. ISS CBUFF Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ISS_CBUFF Base Address Cortex-M3 Private Access	ISS_CBUFF Base Address L3 Interconnect
CBUFF_HL_REVISION	R	32	0x0000 0000	0x5504 1800	0x5C00 1800
CBUFF_HL_HWINFO	R	32	0x0000 0004	0x5504 1804	0x5C00 1804
CBUFF_HL_SYSCONFIG	RW	32	0x0000 0010	0x5504 1810	0x5C00 1810
RESERVED	RW	32	0x0000 001C	0x5504 181C	0x5C00 181C
CBUFF_HL_IRQSTATUS_RAW	RW	32	0x0000 0020	0x5504 1820	0x5C00 1820
CBUFF_HL_IRQSTATUS	RW	32	0x0000 0024	0x5504 1824	0x5C00 1824
CBUFF_HL_IRQENABLE_SET	RW	32	0x0000 0028	0x5504 1828	0x5C00 1828
CBUFF_HL_IRQENABLE_CLR	RW	32	0x0000 002C	0x5504 182C	0x5C00 182C
CBUFF_FRAG_ADDR_j [1]	RW	32	0x0000 0080 + (0x4 * j)	0x5504 1880 + (0x4 * j)	0x5C00 1880 + (0x4 * j)
CBUFF_CTX_CTRL_i [2]	RW	32	0x0000 0100 + (0x20 * i)	0x5504 1900 + (0x20 * i)	0x5C00 1900 + (0x20 * i)
CBUFF_CTX_START_i [2]	RW	32	0x0000 0104 + (0x20 * i)	0x5504 1904 + (0x20 * i)	0x5C00 1904 + (0x20 * i)
CBUFF_CTX_END_i [2]	RW	32	0x0000 0108 + (0x20 * i)	0x5504 1908 + (0x20 * i)	0x5C00 1908 + (0x20 * i)
CBUFF_CTX_WINDOWSIZE_i [2]	RW	32	0x0000 010C + (0x20 * i)	0x5504 190C + (0x20 * i)	0x5C00 190C + (0x20 * i)
CBUFF_CTX_THRESHOLD_F_i [2]	RW	32	0x0000 0110 + (0x20 * i)	0x5504 1910 + (0x20 * i)	0x5C00 1910 + (0x20 * i)
CBUFF_CTX_THRESHOLD_S_i [2]	RW	32	0x0000 0114 + (0x20 * i)	0x5504 1914 + (0x20 * i)	0x5C00 1914 + (0x20 * i)
CBUFF_CTX_STATUS_i [2]	R	32	0x0000 0118 + (0x20 * i)	0x5504 1918 + (0x20 * i)	0x5C00 1918 + (0x20 * i)
CBUFF_CTX_PHY_i [2]	RW	32	0x0000 011C + (0x20 * i)	0x5504 191C + (0x20 * i)	0x5C00 191C + (0x20 * i)

(1) j = 0 to 15
(2) i = 0 to 7

2.9.5.1 CBUFF_HL_REVISION

Table 150. CBUFF_HL_REVISION

Address Offset	0x0000 0000		
Physical Address	0x5504 1800 0x5C00 1800	**Instance**	ISS_CBUFF_CORTEX-M3 ISS_CBUFF_L3
Description	IP revision identifier (X.Y.R) Used by software to track features, bugs, and compatibility		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	See(1)

(1) TI internal data

2.9.5.2 CBUFF_HL_HWINFO

Table 151. CBUFF_HL_HWINFO

Address Offset	0x0000 0004		
Physical Address	0x5504 1804 0x5C00 1804	**Instance**	ISS_CBUFF_CORTEX-M3 ISS_CBUFF_L3
Description	Information about the IP module's hardware configuration.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CONTEXTS		ENABLE_FRAGMENTATION

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:1	CONTEXTS	Number of contexts Read 0x0: 2 contexts Read 0x1: 4 contexts Read 0x2: 8 contexts Read 0x3: Reserved	R	0x1
0	ENABLE_FRAGMENTATION	Provides information to software if fragmentation support is available Read 0x1: Yes Read 0x0: No	R	0

2.9.5.3 CBUFF_HL_SYSCONFIG

Table 152. CBUFF_HL_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x5504 1810 0x5C00 1810	**Instance**	ISS_CBUFF_CORTEX-M3 ISS_CBUFF_L3
Description	Clock management configuration		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												IDLEMODE		RESERVED	SOFTRESET

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000000
3:2	IDLEMODE	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state. 0x0: Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, that is, regardless of the IP module's internal requirements. Backup mode, for debug only. 0x1: No-idle mode: local target never enters IDLE state. Backup mode, for debug only. 0x2: Smart-idle mode: local target's IDLE state eventually follows (acknowledges) the system's IDLE requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wake-up events. 0x3: Reserved	RW	0x2
1	RESERVED	Reserved	R	0
0	SOFTRESET	Software reset Write 0x0: No action Write 0x1: Initiate software reset Read 0x1: Reset (software or other) ongoing Read 0x0: Reset done, no pending action	RW	0

2.9.5.4 CBUFF_HL_IRQSTATUS_RAW

Table 153. CBUFF_HL_IRQSTATUS_RAW

Address Offset	0x0000 0020		
Physical Address	0x5504 1820 0x5C00 1820	**Instance**	ISS_CBUFF_CORTEX-M3 ISS_CBUFF_L3
Description	Per-event raw interrupt status vector. Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_CTX7_OVR	IRQ_CTX6_OVR	IRQ_CTX5_OVR	IRQ_CTX4_OVR	IRQ_CTX3_OVR	IRQ_CTX2_OVR	IRQ_CTX1_OVR	IRQ_CTX0_OVR	IRQ_CTX7_INVALID	IRQ_CTX6_INVALID	IRQ_CTX5_INVALID	IRQ_CTX4_INVALID	IRQ_CTX3_INVALID	IRQ_CTX2_INVALID	IRQ_CTX1_INVALID	IRQ_CTX0_INVALID	IRQ_CTX7_READY	IRQ_CTX6_READY	IRQ_CTX5_READY	IRQ_CTX4_READY	IRQ_CTX3_READY	IRQ_CTX2_READY	IRQ_CTX1_READY	IRQ_CTX0_READY	RESERVED							IRQ_OCP_ERR

Bits	Field Name	Description	Type	Reset
31	IRQ_CTX7_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
30	IRQ_CTX6_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
29	IRQ_CTX5_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
28	IRQ_CTX4_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
27	IRQ_CTX3_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
26	IRQ_CTX2_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
25	IRQ_CTX1_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
24	IRQ_CTX0_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
23	IRQ_CTX7_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
22	IRQ_CTX6_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
21	IRQ_CTX5_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
20	IRQ_CTX4_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
19	IRQ_CTX3_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
18	IRQ_CTX2_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
17	IRQ_CTX1_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
16	IRQ_CTX0_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
15	IRQ_CTX7_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
14	IRQ_CTX6_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
13	IRQ_CTX5_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
12	IRQ_CTX4_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
11	IRQ_CTX3_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
10	IRQ_CTX2_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
9	IRQ_CTX1_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
8	IRQ_CTX0_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
7:1	RESERVED	Reserved	R	0x00
0	IRQ_OCP_ERR	OCP error received in the master port. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

2.9.5.5 CBUFF_HL_IRQSTATUS

Table 154. CBUFF_HL_IRQSTATUS

Address Offset	0x0000 0024		
Physical Address	0x5504 1824 0x5C00 1824	**Instance**	ISS_CBUFF_CORTEX-M3 ISS_CBUFF_L3
Description	Per-event "enabled" interrupt status vector. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_CTX7_OVR	IRQ_CTX6_OVR	IRQ_CTX5_OVR	IRQ_CTX4_OVR	IRQ_CTX3_OVR	IRQ_CTX2_OVR	IRQ_CTX1_OVR	IRQ_CTX0_OVR	IRQ_CTX7_INVALID	IRQ_CTX6_INVALID	IRQ_CTX5_INVALID	IRQ_CTX4_INVALID	IRQ_CTX3_INVALID	IRQ_CTX2_INVALID	IRQ_CTX1_INVALID	IRQ_CTX0_INVALID	IRQ_CTX7_READY	IRQ_CTX6_READY	IRQ_CTX5_READY	IRQ_CTX4_READY	IRQ_CTX3_READY	IRQ_CTX2_READY	IRQ_CTX1_READY	IRQ_CTX0_READY	RESERVED							IRQ_OCP_ERR

Bits	Field Name	Description	Type	Reset
31	IRQ_CTX7_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
30	IRQ_CTX6_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
29	IRQ_CTX5_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
28	IRQ_CTX4_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
27	IRQ_CTX3_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
26	IRQ_CTX2_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
25	IRQ_CTX1_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
24	IRQ_CTX0_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
23	IRQ_CTX7_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
22	IRQ_CTX6_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
21	IRQ_CTX5_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
20	IRQ_CTX4_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
19	IRQ_CTX3_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
18	IRQ_CTX2_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
17	IRQ_CTX1_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
16	IRQ_CTX0_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
15	IRQ_CTX7_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
14	IRQ_CTX6_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
13	IRQ_CTX5_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
12	IRQ_CTX4_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
11	IRQ_CTX3_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
10	IRQ_CTX2_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
9	IRQ_CTX1_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
8	IRQ_CTX0_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
7:1	RESERVED	Reserved	R	0x00
0	IRQ_OCP_ERR	OCP error received in the master port. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

2.9.5.6 CBUFF_HL_IRQENABLE_SET

Table 155. CBUFF_HL_IRQENABLE_SET

Address Offset	0x0000 0028		
Physical Address	0x5504 1828 0x5C00 1828	**Instance**	ISS_CBUFF_CORTEX-M3 ISS_CBUFF_L3
Description	Per-event interrupt enable bit vector. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_CTX7_OVR	IRQ_CTX6_OVR	IRQ_CTX5_OVR	IRQ_CTX4_OVR	IRQ_CTX3_OVR	IRQ_CTX2_OVR	IRQ_CTX1_OVR	IRQ_CTX0_OVR	IRQ_CTX7_INVALID	IRQ_CTX6_INVALID	IRQ_CTX5_INVALID	IRQ_CTX4_INVALID	IRQ_CTX3_INVALID	IRQ_CTX2_INVALID	IRQ_CTX1_INVALID	IRQ_CTX0_INVALID	IRQ_CTX7_READY	IRQ_CTX6_READY	IRQ_CTX5_READY	IRQ_CTX4_READY	IRQ_CTX3_READY	IRQ_CTX2_READY	IRQ_CTX1_READY	IRQ_CTX0_READY	RESERVED							IRQ_OCP_ERR

Bits	Field Name	Description	Type	Reset
31	IRQ_CTX7_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
30	IRQ_CTX6_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
29	IRQ_CTX5_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
28	IRQ_CTX4_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
27	IRQ_CTX3_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
26	IRQ_CTX2_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
25	IRQ_CTX1_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
24	IRQ_CTX0_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
23	IRQ_CTX7_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
22	IRQ_CTX6_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
21	IRQ_CTX5_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
20	IRQ_CTX4_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
19	IRQ_CTX3_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
18	IRQ_CTX2_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
17	IRQ_CTX1_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
16	IRQ_CTX0_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
15	IRQ_CTX7_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
14	IRQ_CTX6_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
13	IRQ_CTX5_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
12	IRQ_CTX4_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
11	IRQ_CTX3_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
10	IRQ_CTX2_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
9	IRQ_CTX1_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
8	IRQ_CTX0_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
7:1	RESERVED	Reserved	R	0x00
0	IRQ_OCP_ERR	OCP error received in the master port. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

2.9.5.7 CBUFF_HL_IRQENABLE_CLR

Table 156. CBUFF_HL_IRQENABLE_CLR

Address Offset	0x0000 002C		
Physical Address	0x5504 182C 0x5C00 182C	**Instance**	ISS_CBUFF_CORTEX-M3 ISS_CBUFF_L3
Description	Per-event interrupt enable bit vector, line 0. Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_CTX7_OVR	IRQ_CTX6_OVR	IRQ_CTX5_OVR	IRQ_CTX4_OVR	IRQ_CTX3_OVR	IRQ_CTX2_OVR	IRQ_CTX1_OVR	IRQ_CTX0_OVR	IRQ_CTX7_INVALID	IRQ_CTX6_INVALID	IRQ_CTX5_INVALID	IRQ_CTX4_INVALID	IRQ_CTX3_INVALID	IRQ_CTX2_INVALID	IRQ_CTX1_INVALID	IRQ_CTX0_INVALID	IRQ_CTX7_READY	IRQ_CTX6_READY	IRQ_CTX5_READY	IRQ_CTX4_READY	IRQ_CTX3_READY	IRQ_CTX2_READY	IRQ_CTX1_READY	IRQ_CTX0_READY	RESERVED							IRQ_OCP_ERR

Bits	Field Name	Description	Type	Reset
31	IRQ_CTX7_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
30	IRQ_CTX6_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
29	IRQ_CTX5_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
28	IRQ_CTX4_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
27	IRQ_CTX3_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
26	IRQ_CTX2_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
25	IRQ_CTX1_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
24	IRQ_CTX0_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
23	IRQ_CTX7_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
22	IRQ_CTX6_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
21	IRQ_CTX5_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
20	IRQ_CTX4_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
19	IRQ_CTX3_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
18	IRQ_CTX2_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
17	IRQ_CTX1_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
16	IRQ_CTX0_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
15	IRQ_CTX7_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
14	IRQ_CTX6_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
13	IRQ_CTX5_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
12	IRQ_CTX4_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
11	IRQ_CTX3_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
10	IRQ_CTX2_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
9	IRQ_CTX1_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
8	IRQ_CTX0_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
7:1	RESERVED	Reserved	R	0x00
0	IRQ_OCP_ERR	OCP error received in the master port. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

2.9.5.8 CBUFF_FRAG_ADDR_j

Table 157. CBUFF_FRAG_ADDR_j

Address Offset	0x0000 0080 + (0x4 * j)	**Index**	j = 0 to 15
Physical Address	See Table 149.	**Instance**	ISS_CBUFF_CORTEX-M3 ISS_CBUFF_L3
Description	Start address of the physical buffer of the CBUFF context 0. This register only exists when fragmentation support is enabled.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																												RESERVED			

Bits	Field Name	Description	Type	Reset
31:4	ADDR	Address in 128-bit words	RW	0x0000000
3:0	RESERVED	Reserved	R	0x0

2.9.5.9 CBUFF_CTX_CTRL_i

Table 158. CBUFF_CTX_CTRL_i

Address Offset	0x0000 0100 + (0x20 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 1900 + (0x20 * i) 0x5C00 1900 + (0x20 * i)	**Instance**	ISS_CBUFF_CORTEX-M3 ISS_CBUFF_L3
Description	Context control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				TILERMODE	DONE	WCOUNT		BCF				RESERVED	MODE		ENABLE

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reserved	R	0x00000
11	TILERMODE	Sets the expected value for ADDR[32]. If ADDR[32]=TILERMODE, ADDR[31:4] is processed and eventually translated. Otherwise, the access is handled as transparent, regardless of the other address bits.	RW	0
10	DONE	Write this bit to 1 to indicate the CPU has finished processing its physical buffer. This bit is automatically cleared by hardware, reads always return 0. This bit has no effect when MODE=2 (read/write) Write 0x0: No effect. Write 0x1: The CPU has completely processed the WB physical buffer.	W	0
9:8	WCOUNT	Window count 0x0: 2 windows 0x1: 4 windows 0x2: 8 windows 0x3: 16 windows	RW	0x0
7:4	BCF	This register controls the bandwidth control feedback loop output. 0: Control loop disabled. 1-15: The control feedback loop enabled. Behavior depends on functional mode, see Section 2.9.3.2.4, *ISS CBUFF Memory-to-Memory Operation BCF*.	RW	0x0
3	RESERVED	Reserved	R	0

Bits	Field Name	Description	Type	Reset
2:1	MODE	Selects the functional mode of this context 0x0: Write mode. ISS writes and CPU reads the physical space. CPU accesses are out of the scope of the CBUFF module; therefore, only writes are permitted between CBUFF_CTX_START__x and CBUFF_CTX_END__x. 0x1: Read mode. Hardware reads and CPU writes the physical space. CPU accesses are out of the scope of the CBUFF module; therefore, only reads are permitted between CBUFF_CTX_START__x and CBUFF_CTX_END__x. 0x2: Read/Write mode. Read and writes are monitored by the CBUFF. WB is used to track current read positions WA is used to track current write position.	RW	0x0
0	ENABLE	Enable/disable 0x0: Disables the context. This resets the internal state of the context. All accesses received on OCPI are transmitted to OCPO without modification. Disabling the context takes effect immediately. Software must ensure that no more accesses to the context are outstanding before disabling it. Otherwise memory corruption may occur. 0x1: Enable the context. All accesses between CBUFF_CTX_START__x and CBUFF_CTX_END__x are processed by the CBUFF.	RW	0

2.9.5.10 CBUFF_CTX_START_i

Table 159. CBUFF_CTX_START_i

Address Offset	0x0000 0104 + (0x20 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 1904 + (0x20 * i) 0x5C00 1904 + (0x20 * i)	**Instance**	ISS_CBUFF_CORTEX-M3 ISS_CBUFF_L3
Description	Start address of the virtual space managed by the context		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																												RESERVED			

Bits	Field Name	Description	Type	Reset
31:4	ADDR	Address in 128-bit words	RW	0x0000000
3:0	RESERVED	Reserved	R	0x0

2.9.5.11 CBUFF_CTX_END_i

Table 160. CBUFF_CTX_END_i

Address Offset	0x0000 0108 + (0x20 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 1908 + (0x20 * i) 0x5C00 1908 + (0x20 * i)	**Instance**	ISS_CBUFF_CORTEX-M3 ISS_CBUFF_L3
Description	End address of the virtual space managed by the context		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																												RESERVED			

Bits	Field Name	Description	Type	Reset
31:4	ADDR	Address in 128-bit words	RW	0x0000000
3:0	RESERVED	Reserved	R	0x0

2.9.5.12 CBUFF_CTX_WINDOWSIZE_i

Table 161. CBUFF_CTX_WINDOWSIZE_i

Address Offset	0x0000 010C + (0x20 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 190C + (0x20 * i) 0x5C00 190C + (0x20 * i)	**Instance**	ISS_CBUFF_CORTEX-M3 ISS_CBUFF_L3
Description	Defines the size of a window		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SIZE																				RESERVED			

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x00
23:4	SIZE	Size in 128-bit words	RW	0x00000
3:0	RESERVED	Reserved	R	0x0

2.9.5.13 CBUFF_CTX_THRESHOLD_F_i

Table 162. CBUFF_CTX_THRESHOLD_F_i

Address Offset	0x0000 0110 + (0x20 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 1910 + (0x20 * i) 0x5C00 1910 + (0x20 * i)	**Instance**	ISS_CBUFF_CORTEX-M3 ISS_CBUFF_L3
Description	Threshold value used to check if a write window is full		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								THRESHOLD																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x00
23:0	THRESHOLD	Threshold value, in bytes	RW	0x000000

2.9.5.14 CBUFF_CTX_THRESHOLD_S_i

Table 163. CBUFF_CTX_THRESHOLD_S_i

Address Offset	0x0000 0114 + (0x20 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 1914 + (0x20 * i) 0x5C00 1914 + (0x20 * i)	**Instance**	ISS_CBUFF_CORTEX-M3 ISS_CBUFF_L3
Description	Threshold value used to control the BCF synchronization mechanism		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								THRESHOLD																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x00
23:0	THRESHOLD	Threshold value, in bytes	RW	0x000000

2.9.5.15 CBUFF_CTX_STATUS_i

Table 164. CBUFF_CTX_STATUS_i

Address Offset	0x0000 0118 + (0x20 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 1918 + (0x20 * i) 0x5C00 1918 + (0x20 * i)	**Instance**	ISS_CBUFF_CORTEX-M3 ISS_CBUFF_L3
Description	Status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				WA				RESERVED				WB			

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reserved	R	0x00000
11:8	WA	Valid values depend on the CBUFF_CTX_CTRL__x.WCOUNT register.	R	0x0
7:4	RESERVED	Reserved	R	0x0
3:0	WB	Valid values depend on the CBUFF_CTX_CTRL__x.WCOUNT register.	R	0x0

2.9.5.16 CBUFF_CTX_PHY_i

Table 165. CBUFF_CTX_PHY_i

Address Offset	0x0000 011C + (0x20 * i)	**Index**	i = 0 to 7
Physical Address	0x5504 191C + (0x20 * i) 0x5C00 191C + (0x20 * i)	**Instance**	ISS_CBUFF_CORTEX-M3 ISS_CBUFF_L3
Description	Start address of the first physical buffer managed by the context when fragmentation support is disabled.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																												RESERVED			

Bits	Field Name	Description	Type	Reset
31:4	ADDR	Address in 128-bit words.	RW	0x0000000
3:0	RESERVED	Reserved	R	0x0

3 ISS ISP

This section describes the image signal processor.

3.1 ISS ISP Overview

The image signal processor (ISP) is part of the image subsystem (ISS) of the device and is a key component for imaging and video applications. This section describes all ISP modules in the multimedia device; that is, the video port (VP), image pipe interface (IPIPEIF), image pipe module (IPIPE), resizer (RSZ), hardware 3A (H3A), image sensor interface (ISIF), and buffer logic (BL).For better understanding, see the first top-level ISS diagram and feature list in , *ISS Overview*.

3.1.1 ISS ISP Features

The video-processing hardware removes the need for expensive camera modules to perform processing functions. The ISS can support the following features:

- On-the-fly or memory-to-memory processing
- Up to 200-MHz pixel throughput.
- Statistic data collection
  - On-the-fly or memory-to-memory operation
  - Data collection for auto exposure
  - Data collection for auto white balance
  - Data collection for auto focus
  - Boundary signal calculation for video stabilization
- IPIPE front end: RAW data processing
  - On-the-fly or memory-to-memory processing
  - 16-bit-wide RAW BAYER data path between image and sensor linearization module
  - 12-bit-wide RAW BAYER data path between sensor linearization module and gamma correction module. Gamma correction module outputs 10-bit data.
  - Programmable Bayer RGB positions
  - Sensor data linearization for dynamic range extension
  - Programmable 2D lens shading compensation (LSC)
  - Per-pixel gain and offset control
  - Black level compensation
  - Boxcar filter
  - Data collection for histogram generation
  - Defect pixel correction (LUT_DPC) with look-up table (LUT)
  - Defect pixel correction (OTF_DPC) with on-the-fly detection and correction
  - 2D noise filtering
  - Green imbalance correction (GIC)
  - Digital gains and offset
  - 8- to 10-bit A-law decompression and 10- to 8-bit A-law compression
- IPIPE back end: RGB and YUV data processing
  - RGB-to-RGB color correction
  - Gamma correction (GC).
  - 3D LUT for color correction
  - 2D edge enhancement (EE)
  - RGB - YUV4:2:2: Color conversion, cosited chroma filtering and downsampling
  - False chroma suppression (FCS)

- Two resizers
  - Performance: input and output rates up to 200 MPix/s
  - YUV4:2:2 to RGB565, ARGB888, YUV4:2:2, and YUV4:2:0 data output
  - YUV4:2:0 to YUV4:2:0 data output
  - RAW to RAW data output
  - Range from x1/4096 to x20. Supports memory-to-memory rescaling.

The ISP comprises the following modules:

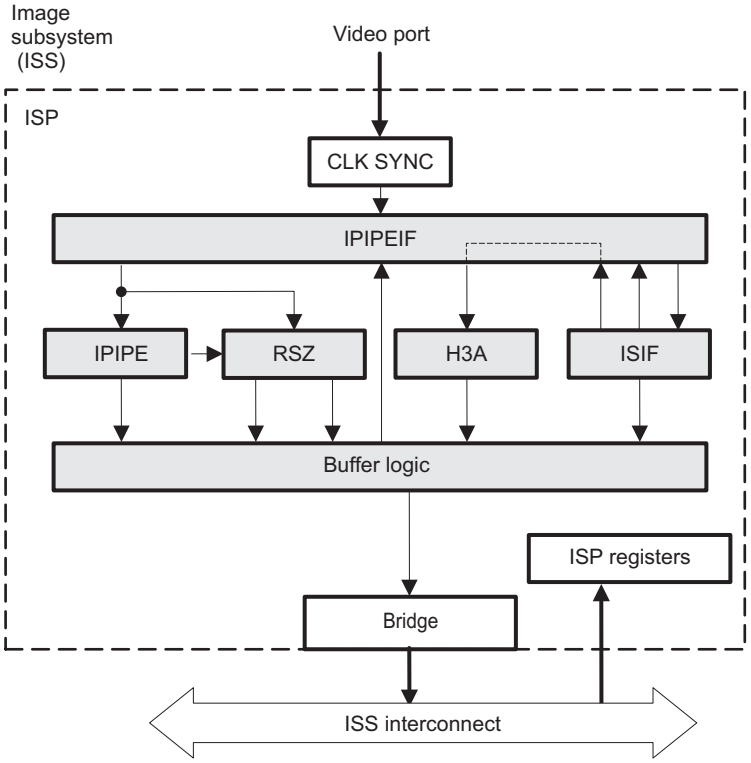
- IPIPE interface (IPIPEIF)
- Image sensor interface (ISIF) accelerator
- Auto exposure, auto white balance, and auto focus engine (H3A)
- Image pipe accelerator (IPIPE)
- Two resizer (RSZ) accelerators
- Buffer logic (BL): Receives module requests, performs arbitration, and creates read/write bursts to the memory subsystem

3.1.2 ISS ISP Block Diagram

Figure 105 is the ISP top-level block diagram. The ISS supports three simultaneous pixel flows (Stall controller, CSI2 A, and parallel interface [CPI]), but only one of them at a time can use the video-processing hardware; the others can go directly to memory (CPI must always use ISP).

The ISP master port is connected to the level 3 (L3) interconnect, and the slave port is connected to the level 4 (L4) interconnect.

Figure 105. ISS ISP Block Diagram



3.2 ISS ISP Integration

This section describes the integration of the ISP modules in the ISP and includes information about clocks, resets, and hardware requests.

The ISP is part of the ISS of the device and is a key component for imaging and video applications such as camera viewfinder, video record, and still image capture.

ISS is coupled with a low interrupt latency ISS microprocessor unit (MPU) subsystem running a real-time operating system to reach optimal performance. Primarily, the ISS MPU subsystem can quickly change the ISS configuration during frame blanking periods and run some sequencing tasks.

3.2.1 ISS ISP PRCM Interface

The ISP and its internal modules, as integrated in the ISS, use the same power and clock management details.

3.2.1.1 ISS ISP Clocks

The PRCM module, through the local power and clock management inside the ISS, provides a unique PCLK that can be enabled from the ISP5_CTRL register.

The modules inside the ISP require three clocks:

- PCLK: This clock is asynchronous to the other clocks. It is provided by the module sending the data to the IPIPEIF module on the VP.
- ISS_ FCLK: This clock is synchronous with the configuration clock domain. This is the clock used for the MTC interface.
- GCK_MMR: This is the clock for the configuration bus. It is created from the ISS_FCLK and runs at half the speed of the ISS_FCLK.

3.2.1.2 ISS ISP Reset

The ISP supports global software reset along with internal hardware reset, if needed.

Software reset is done through the ISP5_SYSCONFIG[1] SOFTRESET bit. Before issuing a software reset, the ISP must be in standby mode. The following must be done:

1. Ensure that the interfaces are stopped from sending data and/or the ISP modules are disabled. Before reset, the last interrupt triggered by the ISP when the frame processing completes is RSZ_INT_DMA. RSZ_INT_DMA must be used to enable clean termination of the processing. Software must wait a few hundred cycles to trigger a soft reset after RSZ_INT_DMA is asserted; this is to ensure that the BL is completely drained.
2. Ensure that ISP5_SYSCONFIG[5:4] STANDBYMODE = 2 (smart standby). Write the ISP5_CTRL[24] MSTANDBY bit to 1 and poll for ISP5_CTRL[20] MSTANDBY_WAIT = 1. Then, the soft reset can be applied (ISP5_SYSCONFIG[1] SOFTRESET = 1).

In case an ISP overflow or underflow event happens (for example, RSZ_FIFO_OVF, ISIF_OVF, etc.), it is not sufficient to reset the ISP. In that case a reset must take place at the ISS level.

3.2.2 ISS ISP Interrupt Tree

Table 166 summarizes the submodule interrupts that can be mapped to the four ISP interrupt output lines. After this, these output lines are mapped to the ISS top interrupt request (IRQ) merger (for more information, see). Software enables the explained before merger from the top-level ISS resources registers in .

Table 166. ISS ISP Interrupt Tree Table

Register	Module	Destination	Comments
ISP5_IRQENABLE_SET_i[9] IPIPEIF_IRQ	IPIPEIF	ISP to ISS merger four IRQ lines	See Section 3.2.3.1.
ISP5_IRQENABLE_SET2_i[1] IPIPEIF_UDF			

Table 166. ISS ISP Interrupt Tree Table (continued)

Register	Module	Destination	Comments
ISP5_IRQENABLE_SET_i[29] IPIPE_INT_DPC_RNEW1	IPIPE		See Section 3.2.4.1.
ISP5_IRQENABLE_SET_i[28] IPIPE_INT_DPC_RNEW0			
ISP5_IRQENABLE_SET_i[27] IPIPE_INT_DPC_INI			
ISP5_IRQENABLE_SET_i[8] IPIPE_INT_HST			
ISP5_IRQENABLE_SET_i[7] IPIPE_INT_BSC			
ISP5_IRQENABLE_SET_i[6] IPIPE_INT_DMA			
ISP5_IRQENABLE_SET_i[5] IPIPE_INT_LAST_PIX			
ISP5_IRQENABLE_SET_i[4] IPIPE_INT_REG			
ISP5_IRQENABLE_SET_i[25] IPIPE_INT_EOF			
ISP5_IRQENABLE_SET2_i[5] IPIPE_BSC_ERR			
ISP5_IRQENABLE_SET2_i[4] IPIPE_HST_ERR			
ISP5_IRQENABLE_SET2_i[2] IPIPE_BOXCAR_OVF			
ISP5_IRQENABLE_SET_i[12] H3A_INT	H3A		See Section 3.2.6.
ISP5_IRQENABLE_SET_i[24] H3A_INT_EOF			
ISP5_IRQENABLE_SET2_i[0] H3A_OVF			
ISP5_IRQENABLE_SET_i[23] RSZ_INT_EOF1	RSZ		See Section 3.2.5.2.
ISP5_IRQENABLE_SET_i[22] RSZ_INT_EOF0			
ISP5_IRQENABLE_SET_i[19] RSZ_FIFO_IN_BLK_ERR			
ISP5_IRQENABLE_SET_i[18] RSZ_FIFO_OVF			
ISP5_IRQENABLE_SET_i[17] RSZ_INT_CYC_RZB			
ISP5_IRQENABLE_SET_i[16] RSZ_INT_CYC_RZA			
ISP5_IRQENABLE_SET_i[15] RSZ_INT_DMA			
ISP5_IRQENABLE_SET_i[14] RSZ_INT_LAST_PIX			
ISP5_IRQENABLE_SET_i[13] RSZ_INT_REG			
ISP5_IRQENABLE_SET_i[3] ISIF_INT_3	ISIF		See Section 3.2.7.1.
ISP5_IRQENABLE_SET_i[2] ISIF_INT_2			
ISP5_IRQENABLE_SET_i[1] ISIF_INT_1			
ISP5_IRQENABLE_SET_i[0] ISIF_INT_0			
ISP5_IRQENABLE_SET2_i[3] ISIF_OVF			

3.2.3 ISS ISP IPIPEIF Integration

3.2.3.1 ISS ISP IPIPEIF Interrupts

The IPIPEIF module generates two interrupts:

- IPIPEIF_IRQ: This event is triggered to the BL module when a new frame starts (VS signal). The interrupt is active low and is asserted for one GCK_MMR clock cycle.
- IPIPEIF_UDF: Interrupt generated when an underflow happens in the IPIPEIF module. Underflow events are nonrecoverable at the ISP level and a soft reset is required at the ISS level.

The interrupts are enabled from the ISP5_IRQENABLE_SET_i[9] IPIPEIF_IRQ and ISP5_IRQENABLE_SET2_i[1] IPIPEIF_UDF bits (where i = 0 to 3 for the line that will be mapped to the four lines of the ISP). Then, each line from the ISP is sent to the ISS top level, where it is muxed with other ISS modules for a total output of six interrupt lines.

3.2.4 ISS ISP IPIPE Integration

3.2.4.1 ISS ISP IPIPE Interrupts

IPIPE can generate several interrupts:

- IPIPE_INT_DPC_RNEW: This event is triggered when there is permission to initialize LUT-DPC table lines 0 and 1.
- IPIPE_INT_DPC_INI: This event is triggered when the defect pixel correction (DPC) table is initialized.
- IPIPE_INT_HST: This event is triggered when the histogram is done.
- IPIPE_INT_BSC: This event is triggered when boundary signal calculation is done.
- IPIPE_INT_DMA: This event is triggered when the boxcar SDRAM transfer is done. On this timing, IPIPE_INT_EOF is sent to the BL. This event is active high for one GCK_MMR clock cycle.
- IPIPE_INT_LAST_PIX: This event is triggered when the last pixel of a frame comes into IPIPE. This event is active high for one GCK_MMR clock cycle.
- IPIPE_INT_EOF: This event is triggered for end of frame.
- IPIPE_BOXCAR_OVF: This event is generated when an overflow happens in the IPIPE-BOXCAR output buffer. The interrupt avoids polling the IPIPE_SRC_STA[0] VAL0 bit for errors. Overflow events are nonrecoverable at the ISP level and a soft reset is required at the ISS level.
- IPIPE_HST_ERR: This event is triggered when the MPU or system direct memory access (EDMA) is still reading the memory that is being used by the module. This is an indication that the read operation was not fast enough.
- IPIPE_BSC_ERR: This error happens when the BSC data is not read fast enough by the MPUs or the EDMA.

The interrupts are enabled from the ISP5_IRQENABLE_SET_i and ISP5_IRQENABLE_SET2_i registers (where i = 0 to 3 for the line that will be mapped to the four lines of the ISP). Then, each line from the ISP is sent to the ISS top level, where it is muxed with other ISS modules for a total output of six interrupt lines.

3.2.4.2 ISS ISP DMA Requests

The ISP generally outputs four direct memory access (DMA) requests, which can be used to read or write memories inside the IPIPE module. These memories are:

- BSC memory: This memory must be read during a vertical blanking period. It is used by the video stabilization application.
- HIST memory: This memory must be read during a vertical blanking period. It is used by the 3A application. The HIST data is double-buffered from frame-to-frame. Software must select the memory that is to be used by setting the IPIPE_HST_TBL[0] SEL bit.
- GAMMA memory: This memory must be set during a vertical blanking period. The imaging application typically uses multiple gamma tables.
- DPC memory: This memory must be set during a frame acquisition. The memory is not big enough to store all faulty pixels for a given frame.

To generate the DMA requests, the following events must be used:

- The IPIPE_INT_BSC event is used to generate the DMA request for the BSC memory. It maps on DMA line 0, ISS_DMA0.

NOTE: If the BSC memory is not read fast enough the ISP5_IRQSTATUS2_i[5] IPIPE_BSC_ERR event is set. This event is available whether memory read takes place with a DMA request or an MPU read.

- The IPIPE_INT_HST event is used to generate the DMA request for the HIST memory. It maps on DMA line 1, ISS_DMA1. The HIST data is double-buffered from frame-to-frame. Software must select the memory that is to be used by setting the IPIPE_HST_TBL[0] SEL bit. When the DMA request is set, it is required to read 4KB from the ping buffer address (0x2000) or the pong buffer address (0x3000). Software must ensure that when one buffer is selected no other accesses (for example, MPU) take place in the other buffer.

NOTE: If the HST memory is not read quickly enough the ISP5_IRQSTATUS2_i[5] IPIPE_HST_ERR event is set. This event is available whether memory read takes place with a DMA request or a CPU read.

- The IPIPE_INT_LAST_PIX event is used to generate the DMA request for the GAMMA memory. This same event can also be used to initialize the DPC LUT (not the preferred method; it is better to use IPIPE_INT_DPC_INI). Basically, when the last pixel is output from the IPIPE module, it is safe to modify the IPIPE memories. It maps on DMA line 3, ISS_DMA3.
- The IPIPE_INT_DPC_INI event signals that DPC table memory initialization can take place. IPIPE_INT_DPC_INI is used to generate two back-to-back DMA requests, the first one mapping on IPIPE_INT_DPC_RNEW0 and the second one mapping on IPIPE_INT_DPC_RNEW1. After initialization (steady state), the IPIPE_INT_DPC_RNEW0 and IPIPE_INT_DPC_RNEW1 events are used to generate the DMA request for the DPC LUT renewal. It maps on DMA line 2, ISS_DMA2. To select which event is used to initialize the DPC (IPIPE_INT_LAST_PIX or IPIPE_INT_DPC_INI), set the ISP5_CTRL[25] DPC_EVT_INI bit.

NOTE: The size of the DPC table in SDRAM must be a multiple of the number of DMA requests. It ensures that during burst capture mode the DMA always loads the correct data from frame-to-frame. The DMA automatically warps back to the start of the table after all expected DMA requests are received. There is a total of ceil(nb_faulty_pixel/128) + 2 DMA requests per frame when the IPIPE_INT_DPC_INI EVENT is used. Hence, the size of the DPC table in the SDRAM is (ceil(nb_faulty_pixel/128) + 2) × 128 × 32 bits.

NOTE: If DMA channels are not used but the MPU is used, an error check must be performed for IPIPE HST and IPIPE BSC. See ISP5_CTRL[26] HST_RD_CHK and ISP5_CTRL[27] BSC_RD_CHK for details.

The DMA request assertion and deassertion to the EDMA is automatic and no software intervention is needed. The ISP contains two registers (ISP5_DMAENABLE_SET and ISP5_DMAENABLE_CLR) that must be used to enable or disable generation of the DMA requests.

The DMA request deassertion is based on late deassertion; that is, the DMA request is disabled only when all the data corresponding to the transfer size have been read or written. To deassert the DMA request and generate the hw_eoi signal, hardware counts the number of 32-bit accesses that are done in the memory range of the corresponding DMA request through the ISP slave port. When the number of accesses corresponding to the DMA request is done, the DMA request is cleared. Multiple DMA requests can be active simultaneously.

Software must not attempt to read or write in the memory range of the DMA requests that are enabled because CPU accesses, instead of EDMA accesses, will be counted. Software can freely access ISP memories for which the DMA request is disabled, and can access registers while the EDMA performs the transfers.

3.2.5 ISS ISP RSZ Integration

3.2.5.1 ISS ISP RSZ PRCM Interface

3.2.5.1.1 ISS ISP RSZ Reset

The resizer module has no standalone software reset. RSZ must be reset at the ISP level. See Section 3.2.1.2, *ISS ISP Reset*.

3.2.5.2 ISS ISP RSZ Interrupts

RSZ can generate several interrupts:

- RSZ_INT_EOF0: This event is triggered for end of frame.
- RSZ_INT_EOF1: This event is triggered for end of frame.
- RSZ_FIFO_IN_BLK_ERR: This event is triggered when the minimum vertical blanking period has not been respected, thus causing errors in the input data buffering submodule. This event is triggered when the RSZ_INT_REG event of frame N is triggered before RSZ_INT_DMA of frame N + 1. This event typically happens at the transition between two frames because there is not enough vertical blanking between frames. Hardware cannot recover from this error. It requires a reset. It is a requirement that despite the error the RSZ module must finish correctly: ongoing requests are completed and further requests are blocked.
- RSZ_FIFO_OVF: This event is triggered when overflow happens in the input data buffering submodule. It typically occurs while processing a frame, because the VP pixel clock is too high. Because hardware cannot recover from this error, a reset is required. It is a requirement that despite the error the RSZ module must finish correctly: ongoing requests are completed and further requests are blocked.

  This event can be also triggered at the master write interface (MTC), where an MTC stall signal is generated.
- RSZ_INT_CYC_RZA/RSZ_INT_CYC_RZB: This event is triggered as circular interrupt every time that RSZ_IRQ_RZA/RSZ_IRQ_RZB output lines are written out to the RZA_SDR_Y/RZB_SDR_Y buffer. The range can go from 1 to 8192 lines. Usually, this value must be such that the circular buffer vertical size (set by the RZBA_SDR_Y_PTR_E/RZBB_SDR_Y_PTR_E register) is a multiple of RSZ_IRQ_RZA/RSZ_IRQ_RZA.
- RSZ_INT_DMA: This event is triggered when the last EOF (of the two MTC interfaces) is sent out to the BL and the RSZ core returns to idle. This event is active high for one GCK_MMR clock cycle.
- RSZ_INT_LAST_PIX: This event is triggered when the last pixel of the valid area is received. This event is active high for one GCK_MMR clock cycle.
- RSZ_INT_REG: This event is triggered when the new value of the shadowed registers, if updated, takes effect on the next RSZ_INT_REG event. Then again, shadowed registers can be updated for the next frame after the RSZ_INT_REG event is triggered. This event is active high for one GCK_MMR clock cycle.

The interrupts are enabled from the ISP5_IRQENABLE_SET_i register (where i = 0 to 3 for the line that will be mapped to the four lines of the ISP). Then, each line from the ISP is sent to the ISS top level, where it is muxed with other ISS modules for a total output of six interrupt lines.

3.2.6 ISS ISP H3A Integration

3.2.6.1 ISS ISP H3A Interrupts

H3A can generate the following interrupts:

- H3A_INT: This event is triggered at the end of the last window or last paxel, whichever completes last. This always triggers at the same time as H3A_INT_EOF.
- H3A_INT_EOF: This event is triggered and generated at the end of the last window. This event is active high for one GCK_MMR clock cycle.
- H3A_OVF: This interrupt is generated when an overflow happens in the H3A output buffer. The interrupt avoids polling the H3A_PCR[21] OVF bit for errors. Overflow events are nonrecoverable at the ISP level and a soft reset is required at the ISS level. The event is active high for one GCK_MMR clock cycle.

The interrupts are enabled from ISP5_IRQENABLE_SET_i and ISP5_IRQENABLE_SET2_i (where i = 0 to 3 for the line that will be mapped to the four lines of the ISP). Then, each line from the ISP is sent to the ISS top level, where it is muxed with other ISS modules for a total output of six interrupt lines.

3.2.7 ISS ISP ISIF Integration

3.2.7.1 ISS ISP ISIF Interrupts

The ISIF can generate several interrupts:

- ISIF_INT_0: This event is triggered when the VD0 interrupt on line 0 is configured. The VD0 interrupt can be configured based on the VD position. It is asserted after receiving the number of horizontal lines (horizontal pulse signals) set in VDINT0. See Section 3.3.6.19.1 for more information.
- ISIF_INT_1: This event is triggered when the VD1 interrupt on line 1 is configured. The VD1 interrupt can be configured based on the VD position. It is asserted after receiving the number of horizontal lines (horizontal pulse signals) set in VDINT1. See Section 3.3.6.19.1 for more information.
- ISIF_INT_2: This event is triggered when the VD2 interrupt on line 2 is configured. The VD2 interrupt can be configured based on the VD position. It is asserted after receiving the number of horizontal lines (horizontal pulse signals) set in VDINT2. See Section 3.3.6.19.1 for more information.
- ISIF_INT_3: This event is triggered LSC interrupt is an interrupt issued by the 2D-LSC block. See Section 3.3.6.10.1.5 for more information.
- ISIF_OVF: This Interrupt is generated when an underflow happens in the ISIF module. The interrupt avoids polling the ISIF_MODESET[11] OVF bit for errors.

The interrupts are enabled from the ISP5_IRQENABLE_SET_i and ISP5_IRQENABLE_SET2_i registers (where i = 0 to 3 for the line that will be mapped to the four lines of the ISP). Then, each line from ISP is sent to the ISS top level where it is muxed with other ISS modules for a total output of six interrupt lines.

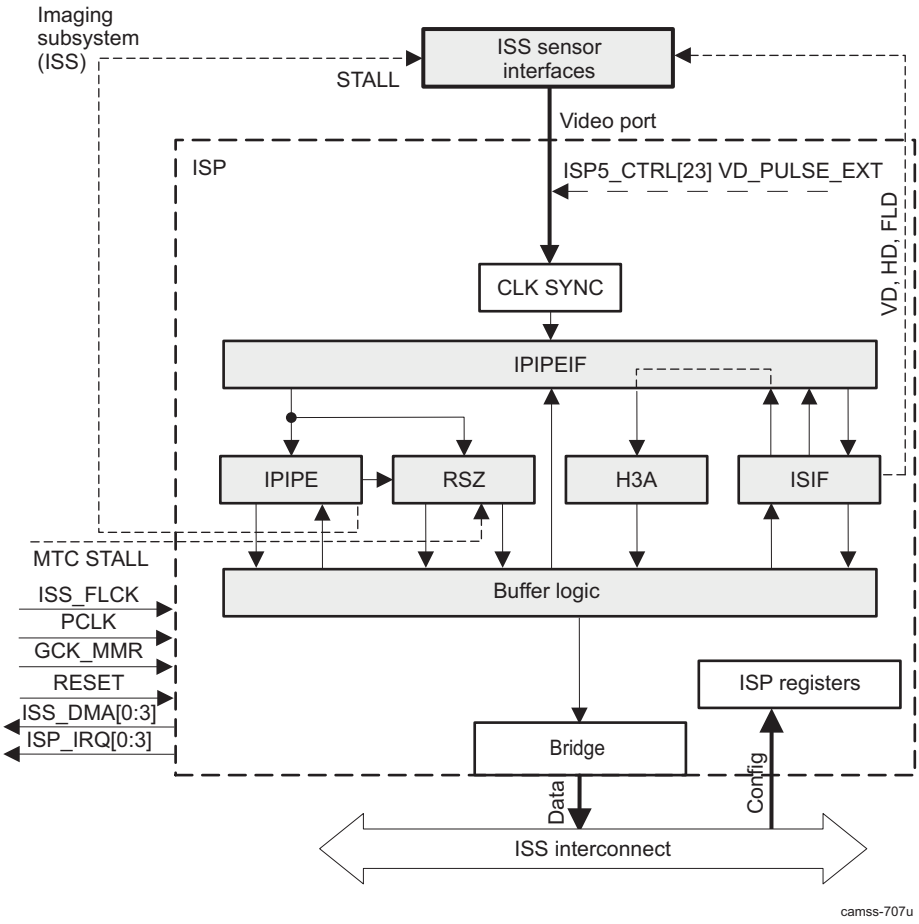
3.2.8 ISS ISP BL Integration

See Section 3.2, *ISS ISP Integration*.

3.3 ISS ISP Functional Description

The functionality of the ISP is part of the overall performance of the ISS. For the top-level ISS diagram with ISP inside and for a features list, see , *ISS Overview*. Figure 106 is an overview of the ISP module. It outputs DMA and interrupt requests, has clocks coming in, and a stall signal to the SC in the ISS sensor interfaces from the resizer. It also shows the top-level configuration for input to IPIPEIF. For the scaled-in functional details of each submodule inside the ISP, see its functional description section.

Figure 106. ISS ISP High-Level Diagram



3.3.1 ISS ISP VP Functional Description

3.3.1.1 ISS ISP VP Overview

The VP supports a parallel interface that is used for interfacing with image sensors. The ISP VP can transport 8- to 16-bit RAW data and 8-/16-bit YCbCr data.

Figure 107 shows the VP module connections to other submodules of the ISP.

3.3.1.2 ISS ISP VP Data Formats

The VP can be used to connect external camera receivers to the ISS. Data paths inside the ISP hardware depend on the image format sourced by the sensor (RAW RGB, YUV4:2:2, JPEG, etc.). Table 167 shows how the CSI2/SC modules are connected to the VP in function of the image format.

Figure 107. ISS ISP VP High-Level Diagram

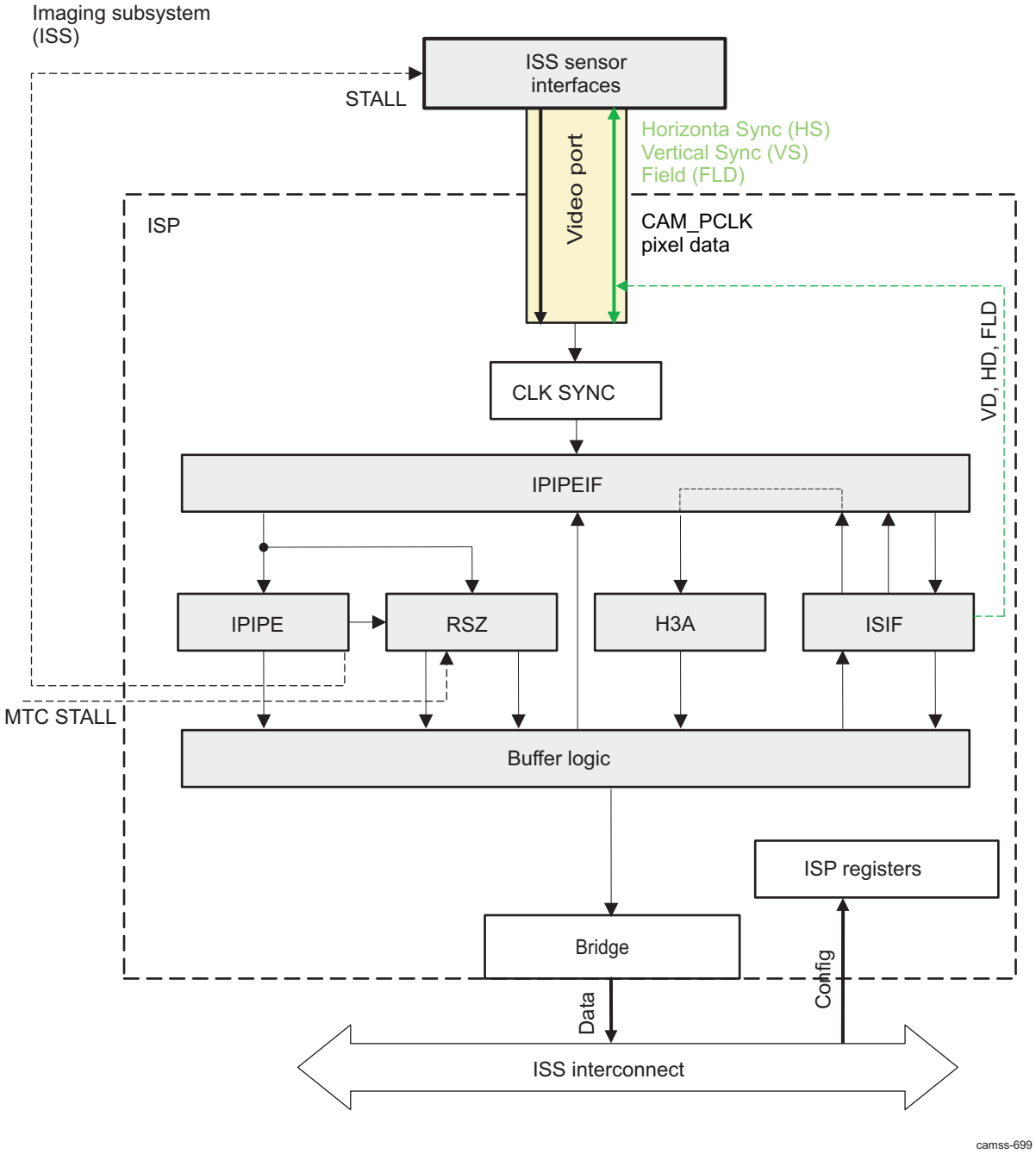


Table 167. ISS ISP VP Format Mapping

Source		Format	Connected to																ISIF	Data Provided to ISIF Linearization Engine																	
SC	CSI2		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	GW DI	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	RAW16	R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
X	X	RAW14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0
			0	0	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	1	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0
X	X	RAW12	0	0	0	0	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0
			0	0	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	1	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0
			R11	R10	R9	R8	R7	R5	R5	R4	R3	R2	R1	R0	0	0	0	0	2	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0
X	X	RAW10	0	0	0	0	0	0	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0	0	0
			0	0	0	0	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	1	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0	0	0
			0	0	R9	R8	R7	R5	R5	R4	R3	R2	R1	R0	0	0	0	0	2	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0	0	0
			R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0	0	0	3	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0	0	0
X	X	RAW8								R8	R7	R6	R5	R4	R3	R2	R1	R0	0	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0	0	0	0	0
									R7	R6	R5	R4	R3	R2	R1	R0			1	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0	0	0	0	0
							R7	R6	R5	R4	R3	R2	R1	R0					2	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0	0	0	0	0
					R7	R6	R5	R4	R3	R2	R1	R0							3	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0	0	0	0	0
			R7	R5	R5	R4	R3	R2	R1	R0									4	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0	0	0	0	0
		YUV 16-bit	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	C7	C6	C5	C4	C3	C2	C1	C0	0																	
		YUV 8-bit									YC7	YC6	YC5	YC4	YC3	YC2	YC1	YC0	0																	

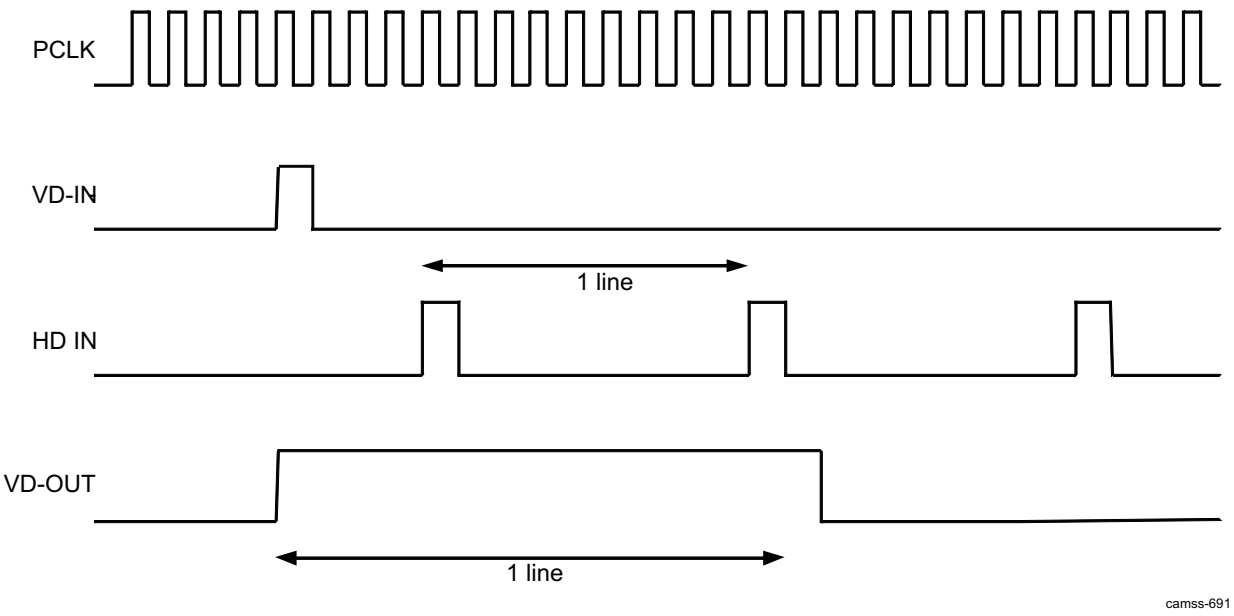
3.3.1.3 ISS ISP VP Top-Level Communication With SC and CSI2 RX

At the ISS level, the VP is connected to the VP of the SC and CSI2 RX modules. VP implementation differences force the introduction of a bridge between the SC/CIS2 RX modulesand the VP. The role of the bridge is to perform VD pulse extension. The SC module assumes that the VD signal is active for at least one pixel clock cycle, and the CSI2 RX module assumes that the VD signal is asserted for four pixel clock cycles. However, the ISP assumes that the VD pulse is active on at least one line.

Figure 108 shows how the VD pulse extension works. Assume that VD-IN is the VD signal at the input of the pulse extension bridge, and VD-OUT is the VD signal at the output of the pulse extension bridge.

The VD-OUT signal is asserted at the same time as VD-IN. The VD-OUT signal is kept high until one full line is received. A line is delimited by two rising edges of the HD signal. The VD-OUT pulse is deasserted on the next cycle after the falling edge of the HD signal.

Figure 108. ISS ISP VP VD Pulse



The ISP5_CTRL[23] VD_PULSE_EXT bit controls whether the VD extension bridge is enabled or disabled. By default, the bridge is enabled. When the bridge is disabled, the VD pulse must be unmodified: VD-OUT = VD-IN. At the ISS level, it is expected that ISP5_CTRL [23] VD_PULSE_EXT = 1 when the VP gets data from the CSI2 RX modules and ISP5_CTRL[23] VD_PULSE_EXT = 0 when the VP gets data from the parallel interface or SC module.

CAUTION

A minimum of four lines per frame is required on the VP when the VD pulse extension bridge is enabled; therefore, the VD extension bridge is not functional if a 1-/2-/3-line frame is sent to the VP.

3.3.1.4 ISS ISP VP Pixel Clock Inversion

The ISP always uses the rising edge of the pixel clock to sample the pixel data. The ISP provides the capability to invert the pixel clock so it can shift the resampling of a pixel clock period by half. This is controlled by the ISP5_CTRL[22] PCLK_INV bit. By default, the inversion is disabled. The 5 bits in Table 168 are resynchronized from the GCK_MMR clock domain to the PCLK clock domain. There must be at least three clock cycles between the time these bits are modified and the HD/VD pulse for start of frame comes.

Table 168. ISS ISP VP GCK_MMR to PCLK Clock Resynchronization

Module	Register	Bit Field
ISP	ISP5_CTRL	VD_PULSE_EXT
ISIF	ISIF_MODESET	HDVDD
ISIF	ISIF_MODESET	FIDD
ISP	ISP5_CTRL	ISIF_CLK_ENABLE

3.3.2 ISS ISP IPIPEIF Functional Description

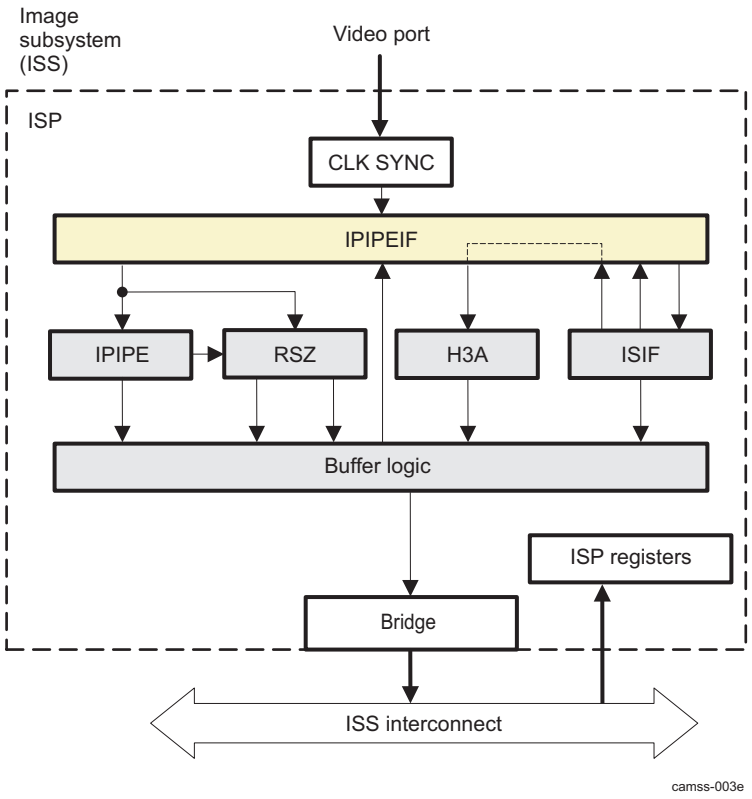
3.3.2.1 ISS ISP IPIPEIF Overview

The IPIPEIF module provides data and synchronization signals (HD, VD) for the ISIF, IPIPE, RSZ, and H3A modules. The data source of this module is the VP, ISIF, or SDRAM using BL, and the selected data is output to ISIF, IPIPE, H3A, and RSZ. This module supports:

- Up to 16-bpp data on the VP
- Up to 200-MHz pixel clock on the VP, up to 8K × 8K image resolution
- RAW and YUV data formats on the VP and BL ports
- Dark-frame subtract of 8-bit RAW image stored in SDRAM from image from VP
- Dark-frame subtract of 8-bit RAW image stored in SDRAM from image from ISIF
- Dark-frame subtract of 8-bit RAW image from VP from image read from SDRAM through the BL
- Simple defect correction to prevent the subtraction of defect pixels
- 8-10, 8-12 DPCM decompression of 10-8, 12-8 DPCM compressed data in SDRAM
- Simple and advanced DPCM predictor
- Inverse A-Law decompression of RAW data 10-8 A-Law compressed from SDRAM
- 8-bit, 12-bit unpacking of 8-bit,12-bit packed SDRAM data
- Gain multiply for output data to IPIPE module
- Horizontal Bayer rescaler in the data paths to the IPIPE and H3A modules: Supports (1, 2, 1) averaging filter and supports horizontal pixel decimation
- Data rate control when reading data from SDRAM: Fraction clock divider
- (1, 2, 1) averaging filter and supports horizontal pixel decimation in the data path to the IPIPE for YUV data.

  Figure 109 show the IPIPEIF module connections to other submodules of the ISP.

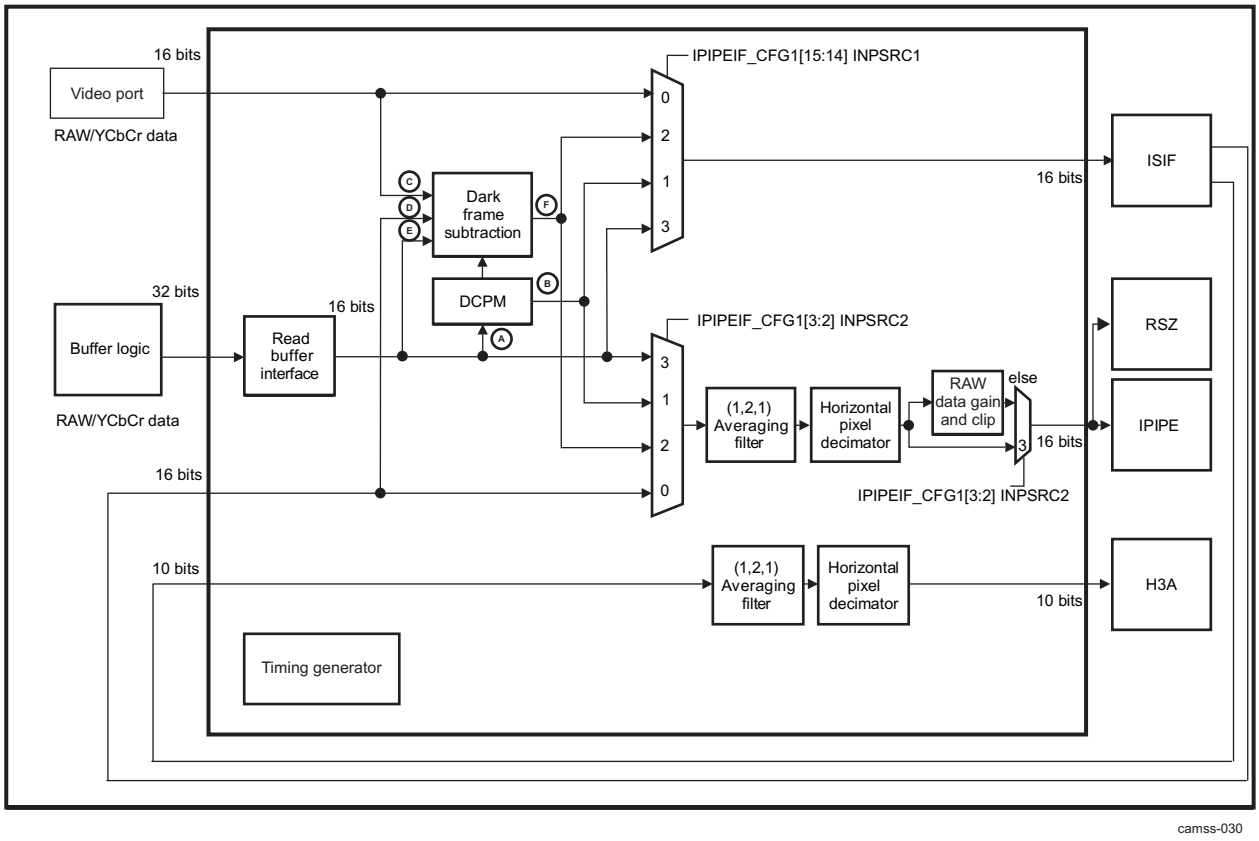
Figure 109. ISS ISP IPIPEIF High-Level Diagram



3.3.2.2 ISS ISP IPIPEIF Top-Level Block Diagram

The following sections describe the function of each subblock in the IPIPEIF, as shown in Figure 110.

Figure 110. ISS ISP IPIPEIF Top-Level Block Diagram



NOTE: When the IPIPEIF receives data from the VP, the timing generator must be configured for HD, VD, and WEN. For more information, see Section 3.3.2.5, *ISS ISP IPIPEIF Timing Generation.*

3.3.2.3 ISS ISP IPIPEIF Input Interface

The IPIPEIF module comprises two major interface blocks: VP and BL. The data types can be RAW or YUV.

3.3.2.3.1 ISS ISP IPIPEIF Input From VP

The VP typically receives data from the image sensor. At the ISS level, it is connected to the serial interface receivers.

3.3.2.3.2 ISS ISP IPIPEIF Input From BL

The BL is the interface with the memory (SDRAM). In that case, the SDRAM address and line offset registers must be programmed in units of 32 bytes.

- SDRAM start address (byte) = (IPIPEIF_ADDRU[10:0] ADDRU) <<16 + (IPIPEIF_ADDRL[15:0] ADDRL)
- SDRAM address offset (byte) = IPIPEIF_ADOFS[11:0] ADOFS

Two types of data can be stored in memory: pixel data and dark frame data.

For pixel data, the HD and VD signals are reconstructed with:

- IPIPEIF_HNUM
- IPIPEIF_VNUM
- IPIPEIF_LPFR
- IPIPEIF_PPLN

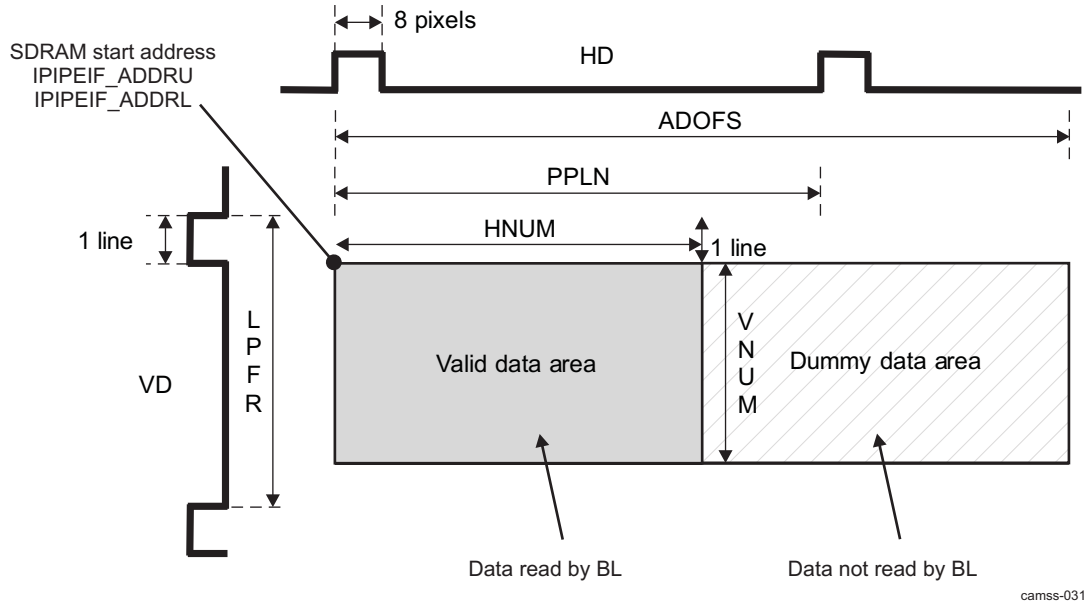
The IPIPEIF_HNUM and IPIPEIF_VNUM registers define the number of pixels per line and lines per frame to read from the SDRAM, and the IPIPEIF_LPFR and IPIPEIF_PPLN registers define the interval of VD and HD, respectively.

Vertical blanking for the frame is defined with the following equation: IPIPEIF_LPFR - IPIPEIF_VNUM - 1.

Horizontal blanking for the frame is defined with the following equation: IPIPEIF_PPLN - IPIPEIF_HNUM.

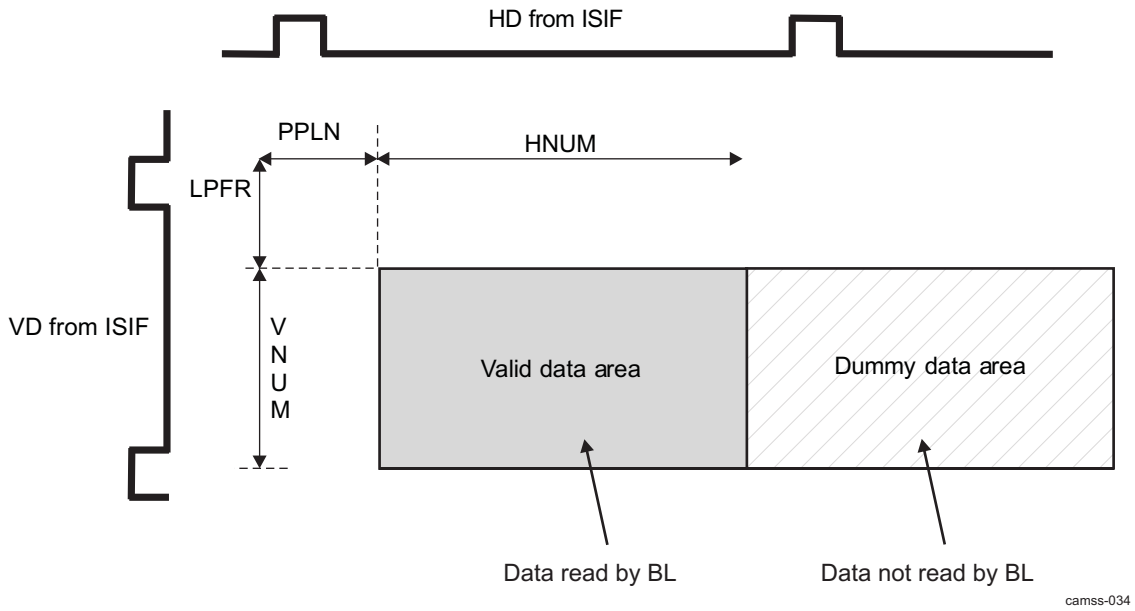
Figure 111 shows the global frame definition for all SDRAM input modes, except for dark frame subtract.

Figure 111. ISS ISP IPIPEIF Global Frame Definition in SDRAM Input Modes (Except Dark Frame)



For dark frame data, the HD and VD signals come from the VP through the ISIF. The IPIPEIF_PPLN and IPIPEIF_LPFR registers must be used to indicate the horizontal and vertical start position of the subtraction from the ISIF data, as shown in Figure 112. The value of the IPIPEIF_LPFR [12:0] LPFR bit field must be greater than 0 because the first line from the VP or ISIF cannot be subtracted from. The IPIPEIF_HNUM and IPIPEIF_VNUM registers must be used to set the number of valid pixels horizontally and the number of valid lines vertically.

Figure 112. ISS ISP IPIPEIF Global Frame Definition in Dark Frame Subtract Mode



3.3.2.3.2.1 ISS ISP IPIPEIF Double-Buffer Input Function When Reading From BL

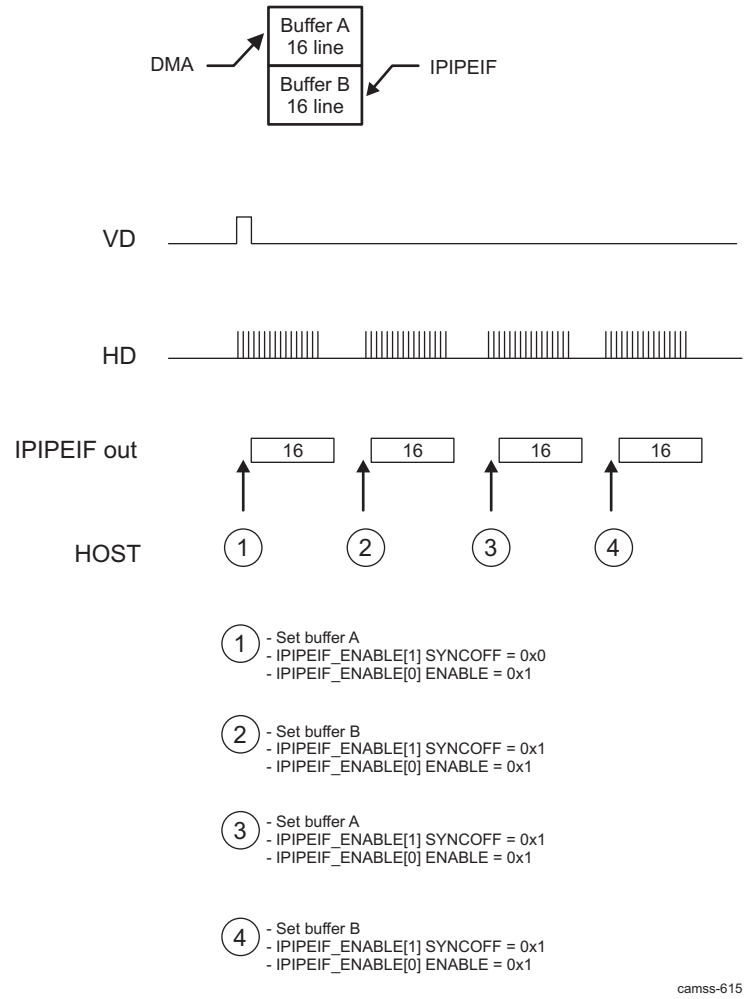
The IPIPEIF module supports a double-buffer input function. This feature is most useful when SDRAM space is limited, because it enables to read continuously from two buffers and to push data to the rest of the ISP (ISIF, H3A, etc.) for further processing.

Consider the following configuration where data are read from two buffers, A and B. The intent is not only to read continuously from these buffers but also to ensure that the ISP modules consider the data as being from the same frame; that is, VD is generated the first time buffer A is read, but it must not toggle until all the frames are read.

The IPIPEIF module can mask the VD sync signal by writing the IPIPEIF_ENABLE[1] SYNCOFF bit but such that the IPIPEIF module drives the data to the ISP modules as if it is a continuous frame data.

In the following example, there are 16 lines per trigger and input circular addressing. VNUM = 16 (see Figure 111), and the VD signal is generated only for the first frame (see Figure 113).

Figure 113. ISS ISP IPIPEIF Double-Buffer Functionality



3.3.2.4 ISS ISP IPIPEIF Data Path Selection

The data path configuration through the IPIPEIF module is set with the IPIPEIF_CFG1[15:14] INPSRC1 and IPIPEIF_CFG1[3:2] INPSRC2 bit fields. Table 169 lists the possible combinations for these two bit fields.

Table 169. ISS ISP IPIPEIF IPIPEIF_CFG1[15:14] INPSRC1 and IPIPEIF_CFG1[3:2] INPSRC2 Possible Combinations

IPIPEIF_CFG1[15:14] INPSRC1	IPIPEIF_CFG1[3:2] INPSRC2	Description	Common Use
0	0	This data path is described in Section 3.3.2.4.1, *ISS ISP IPIPEIF INPSRC1 = 0 and INPSRC2 = 0* .	Video record, view finder, on-the-fly still image capture applications
0	1	This data path is described in Section 3.3.2.4.2, *ISS ISP IPIPEIF INPSRC1 = 0 and INPSRC2 = 1*.	Memory-to IPIPE-to memory
0	2	This data path is described in Section 3.3.2.4.3, *ISS ISP IPIPEIF INPSRC1 = 0 and INPSRC2 = 2*.	Image capture with on-the-fly dark frame subtraction

Table 169. ISS ISP IPIPEIF IPIPEIF_CFG1[15:14] INPSRC1 and IPIPEIF_CFG1[3:2] INPSRC2 Possible Combinations (continued)

IPIPEIF_CFG1[15:14] INPSRC1	IPIPEIF_CFG1[3:2] INPSRC2	Description	Common Use
0	3	This data path is described in Section 3.3.2.4.4, *ISS ISP IPIPEIF INPSRC1 = 0 and INPSRC2 = 3*.	On-the-fly data acquisition done in VP, forwarded to the ISIF, and then to the H3A through IPIPEIF while data from memory is processed and forwarded to the IPIPE module and then stored in memory YUV4:2:2 or RAW data processing with the IPIPE and RESIZER modules from memory to memory fetched by IPIPEIF
1	0	This data path is described in Section 3.3.2.4.5, *ISS ISP IPIPEIF INPSRC1 = 1 and INPSRC2 = 0*.	Memory-to-ISIF- to memory operation
1	1	This data path is possible but there is no use case associated.	N/A
1	2	This data path is not supported.	N/A
1	3	This data path is not supported.	N/A
2	0	This data path is described in Section 3.3.2.4.6, *ISS ISP IPIPEIF INPSRC1 = 2 and INPSRC2 = 0*.	Dark frame subtraction is performed and data sent to the ISIF module for further processing, back to IPIPEIF, and then to IPIPE and RSZ.
2	1	This data path is not supported.	N/A
2	2	This data path is not supported.	N/A
2	3	This data path is not supported.	N/A
3	0	This data path is described in Section 3.3.2.4.7, *ISS ISP IPIPEIF INPSRC1 = 3 and INPSRC2 = 0*.	Memory-to-ISIF-to memory operation
3	1	This data path is not supported.	N/A
3	2	This data path is not supported.	N/A
3	3	This data path is possible but there is no use case associated.	N/A

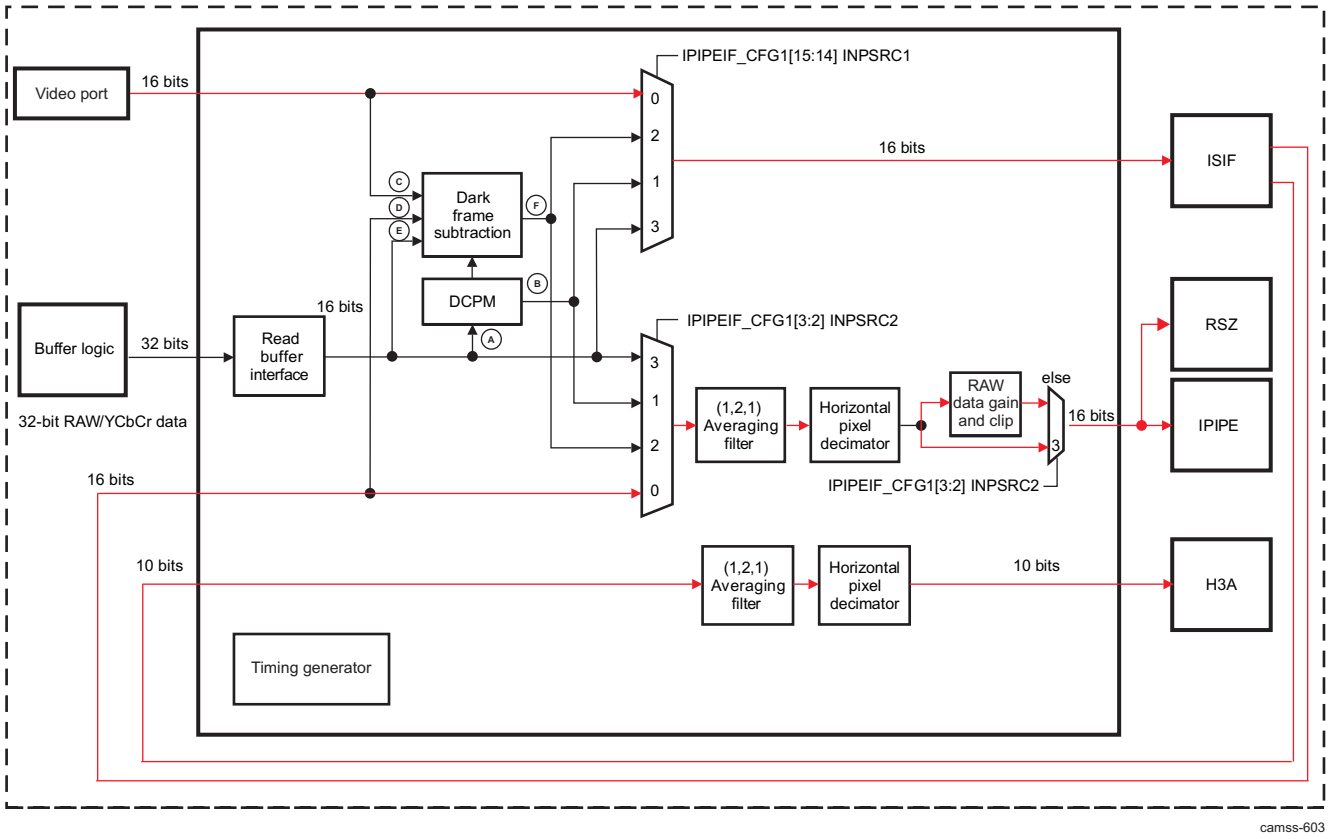
3.3.2.4.1 ISS ISP IPIPEIF INPSRC1 = 0 and INPSRC2 = 0

Set the IPIPEIF_CFG1[15:14] INPSRC1 bit field to 0 and the IPIPEIF_CFG1[3:2] INPSRC2 bit field to 0.

This configuration can be used for the video record, viewfinder, and on-the-fly still image capture applications. The full ISP processing capability is used in a single pass.

Figure 114 shows the data path.

Figure 114. ISS ISP IPIPEIF INPSRC1 = 0 and INPSRC2 = 0 Data Path



3.3.2.4.2 ISS ISP IPIPEIF INPSRC1 = 0 and INPSRC2 = 1

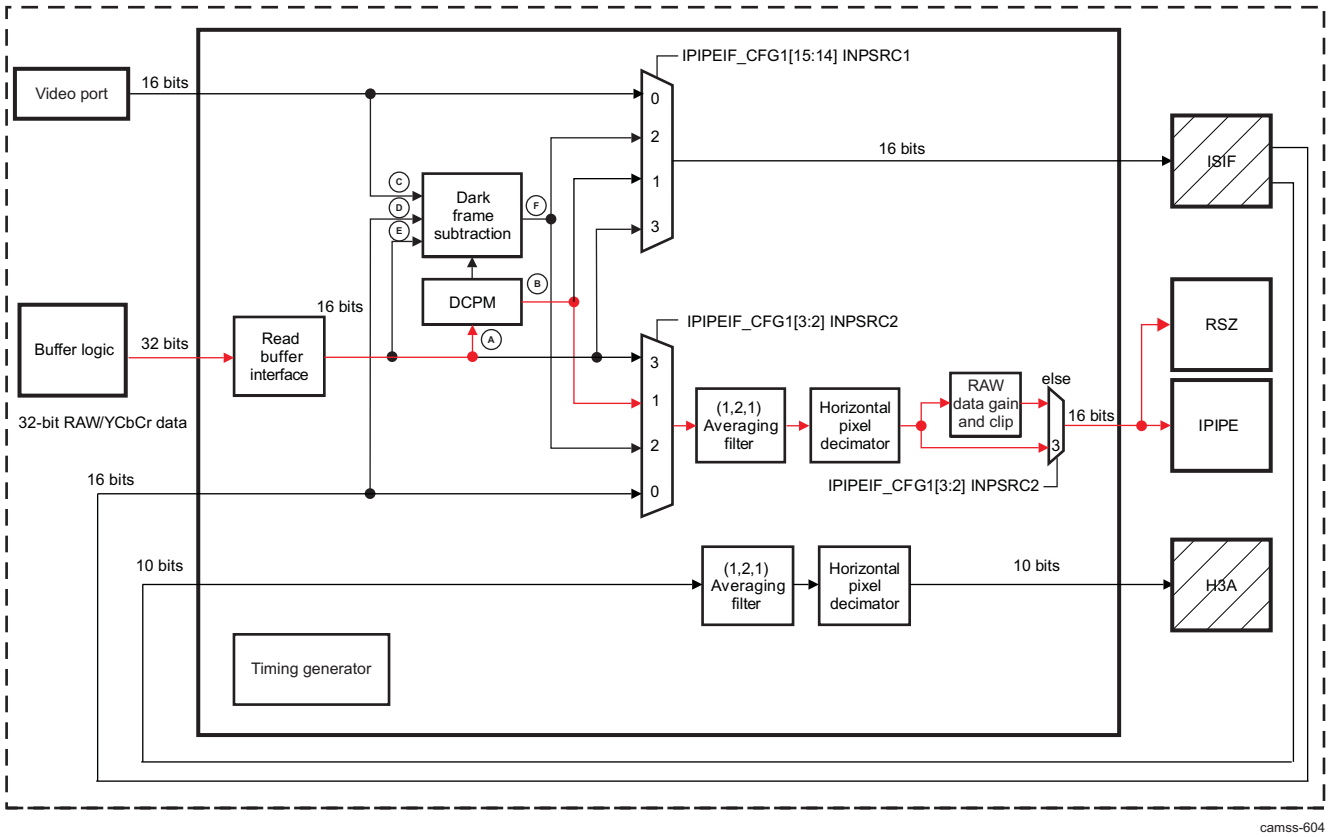
Set the IPIPEIF_CFG1[15:14] INPSRC1 bit field to 0 and the IPIPEIF_CFG1[3:2] INPSRC2 bit field to 1.

This configuration can be used to process data with the IPIPE module from memory to memory. The data stored in SDRAM can be decompressed (A-law or DPCM) before being forwarded to the IPIPE module.

NOTE: In this configuration, the ISIF and H3A modules are assumed to be disabled.

Figure 115 shows the data path.

Figure 115. ISS ISP IPIPEIF INPSRC1 = 0 and INPSRC2 = 1 Data Path



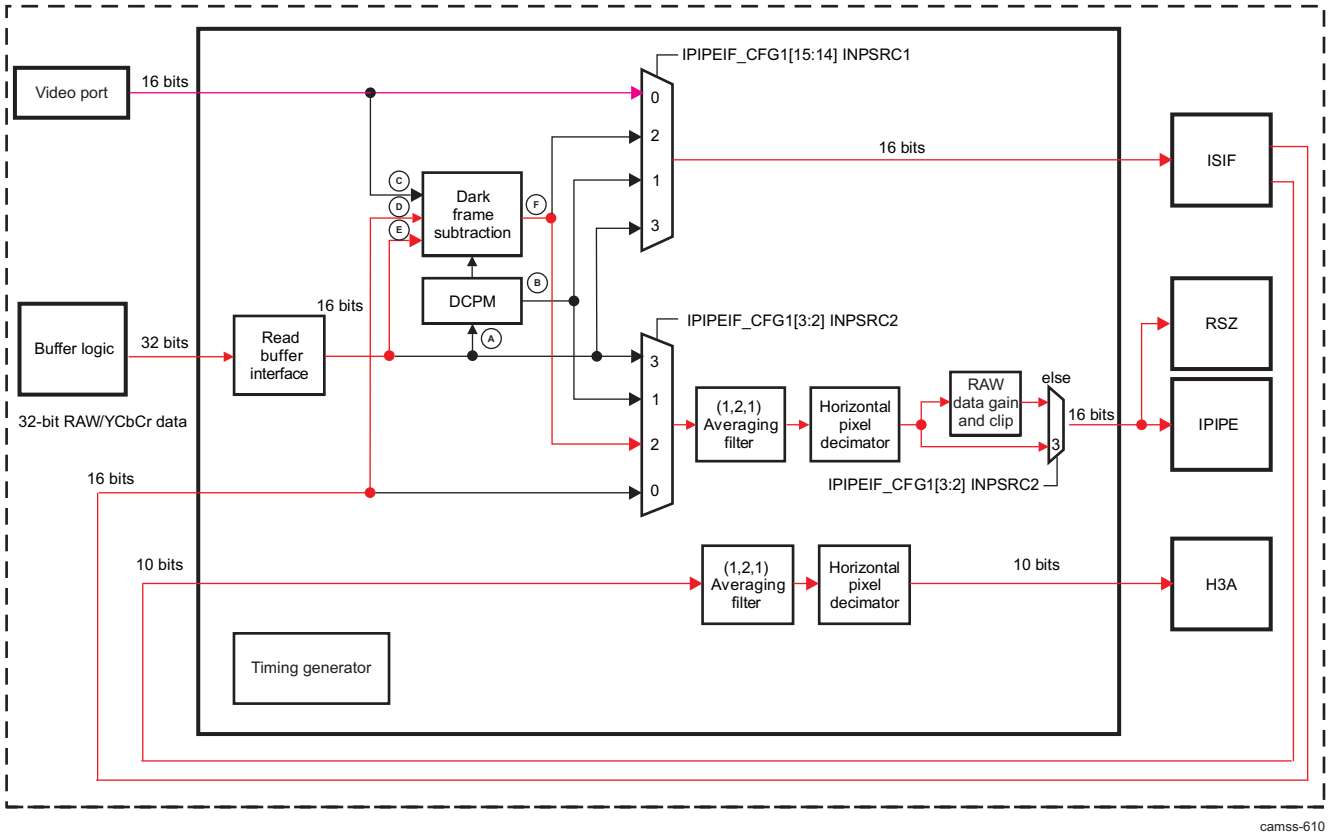
3.3.2.4.3 ISS ISP IPIPEIF INPSRC1 = 0 and INPSRC2 = 2

Set IPIPEIF_CFG1[15:14] INPSRC1 to 0 and IPIPEIF_CFG1[3:2] INPSRC2 to 2.

This configuration can be used for image capture with on-the-fly dark-frame subtraction. In the first case, the dark frame can come from BL and data from the ISIF. In the second case, the dark frame can come from the VP and data from BL.

Figure 116 shows the data path.

Figure 116. ISS ISP IPIPEIF INPSRC1 = 0 and INPSRC2 = 2 Data Paths



3.3.2.4.4 ISS ISP IPIPEIF INPSRC1 = 0 and INPSRC2 = 3

Set the IPIPEIF_CFG1[15:14] INPSRC1 bit field to 0 and the IPIPEIF_CFG1[3:2] INPSRC2 bit field to 3.

In the first case, on-the-fly data acquisition is done with the VP, forwarded to the ISIF, and then sent to the H3A through the IPIPEIF while data from memory is processed and forwarded to the IPIPE module and then stored in memory.

In the second case, the configuration can be used to process YUV4:2:2 or RAW data with the IPIPE and RESIZER modules from memory-to-memory. The YUV4:2:2 or RAW data stored in the SDRAM is fetched and forwarded to the IPIPE and RSZ modules. ISIF and H3A are assumed to be disabled in this configuration.

Figure 117 and Figure 118 show the two possible data paths.

Figure 117. ISS ISP IPIPEIF INPSRC1 = 0 and INPSRC2 = 3 Data Paths: First Case

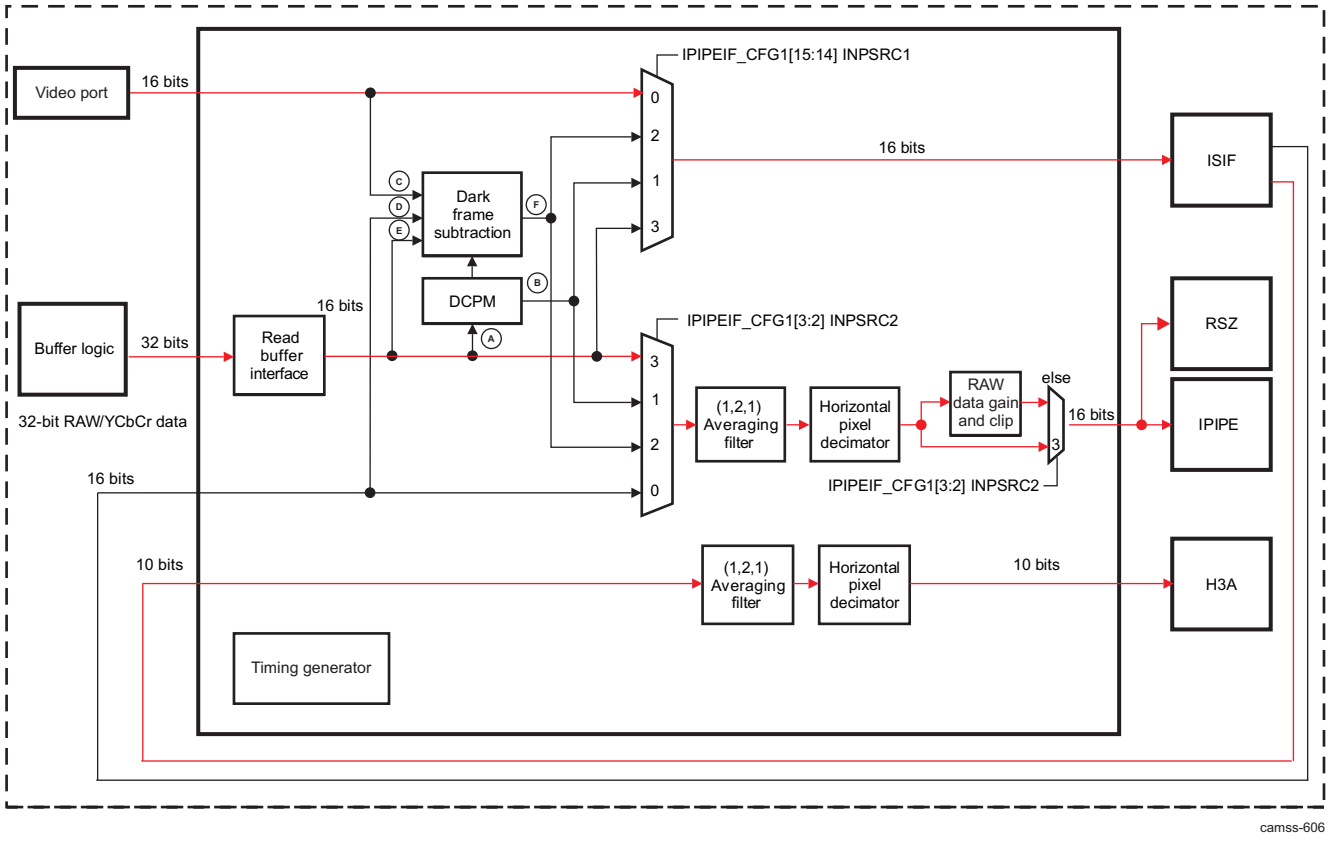
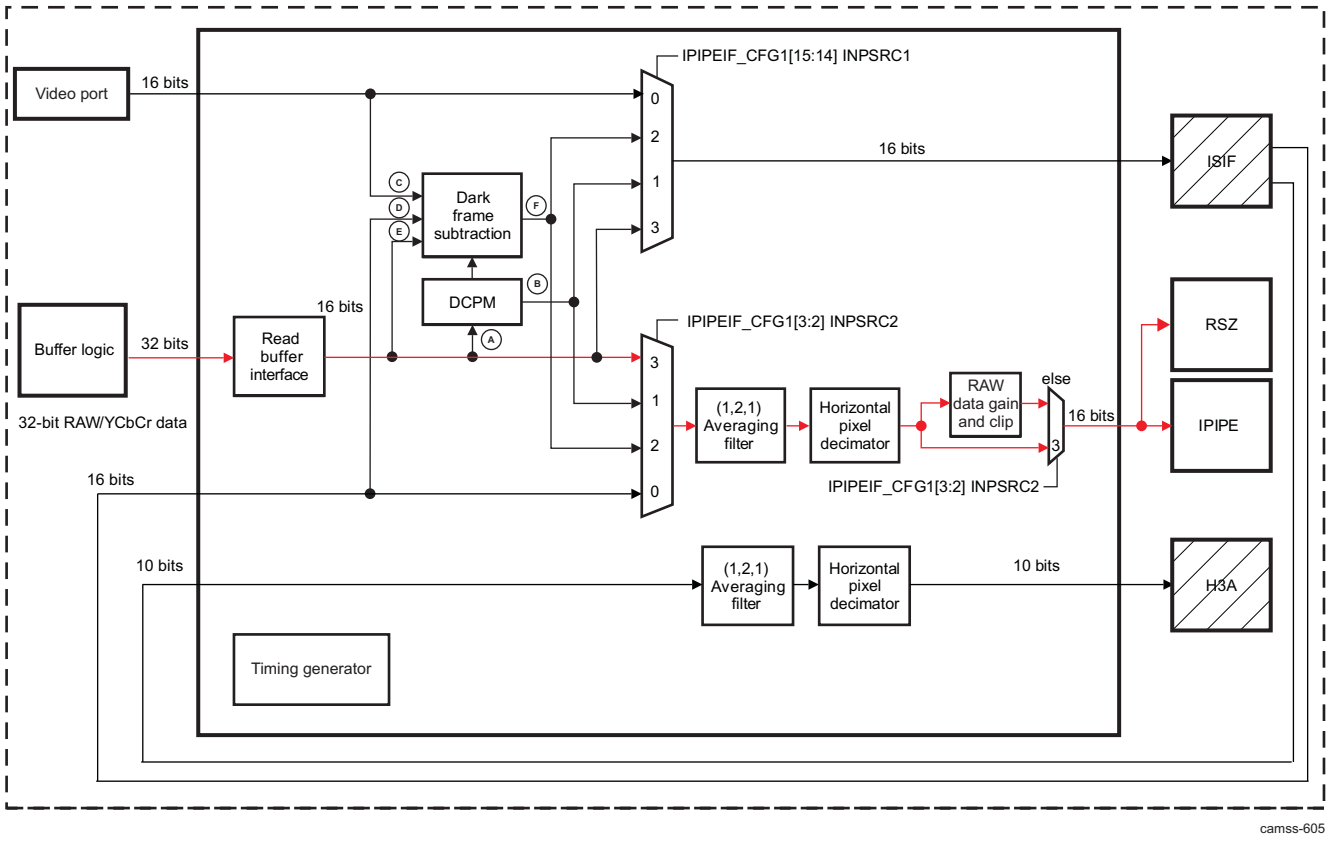


Figure 118. ISS ISP IPIPEIF INPSRC1 = 0 and INPSRC2 = 3 Data Paths: Second Case



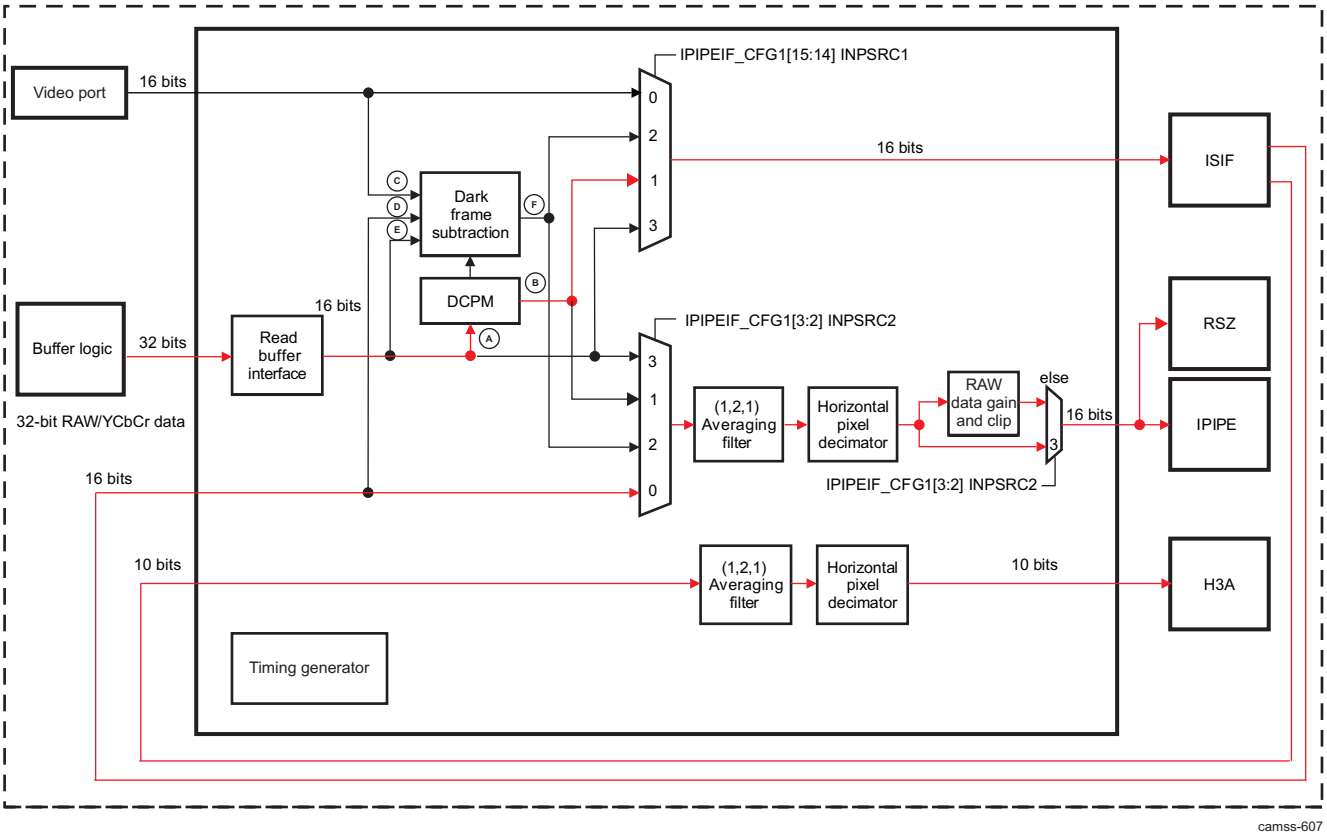
3.3.2.4.5 ISS ISP IPIPEIF INPSRC1 = 1 and INPSRC2 = 0

Set the IPIPEIF_CFG1[15:14] INPSRC1 bit field to 1 and the IPIPEIF_CFG1[3:2] INPSRC2 bit field to 0.

This configuration is a memory-to-memory operation. RAW data is read by the BL interface, decompressed, and pushed to the ISIF. The ISIF processes the data and sends it back to the IPIPEIF module before the data is pushed to the IPIPE and H3A modules.

Figure 119 shows the data path.

Figure 119. ISS ISP IPIPEIF INPSRC1 = 1 and INPSRC2 = 0 Data Path



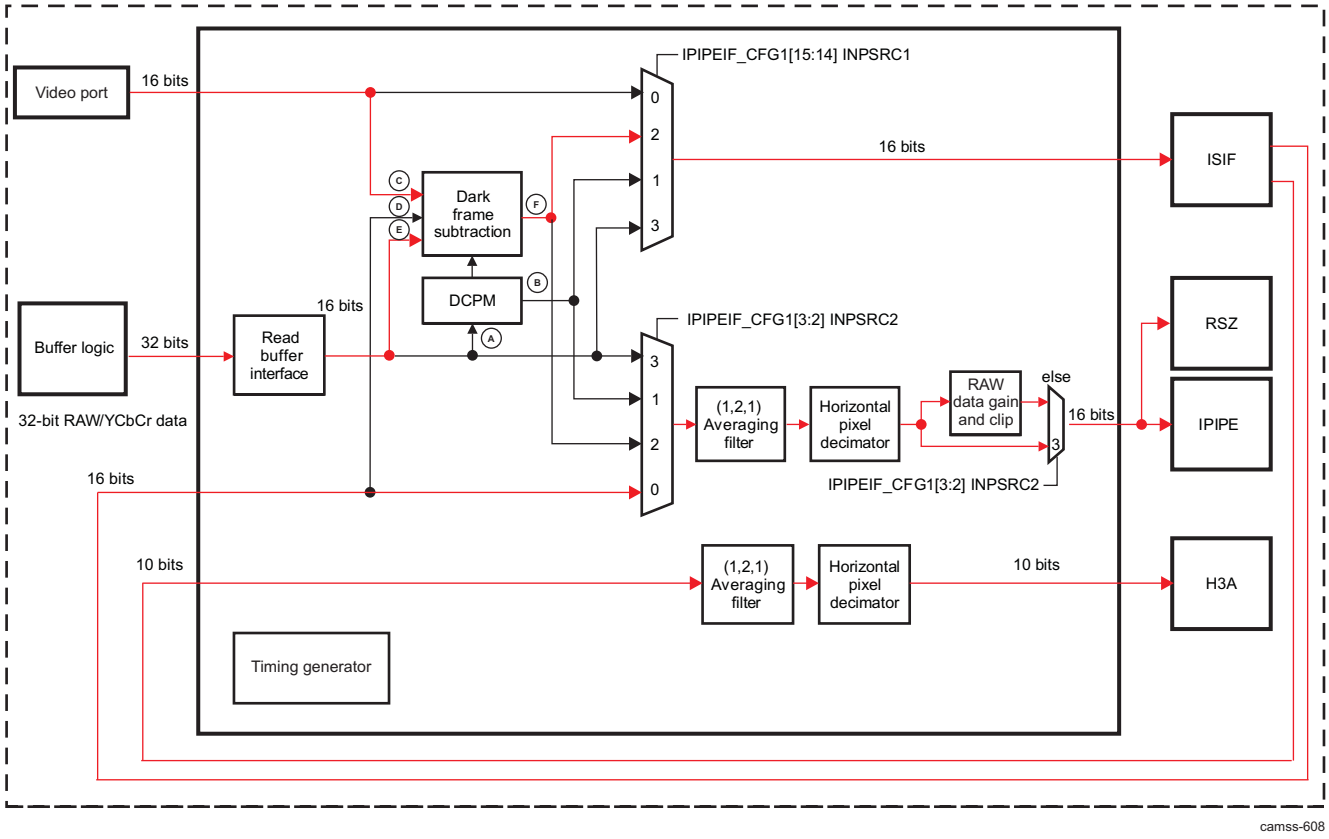
3.3.2.4.6 ISS ISP IPIPEIF INPSRC1 = 2 and INPSRC2 = 0

Set the IPIPEIF_CFG1[15:14] INPSRC1 bit field to 2 and the IPIPEIF_CFG1[3:2] INPSRC2 bit field to 0.

In this configuration, dark frame subtraction is performed and data is sent to the ISIF module. The ISIF processes the data and sends it back to the IPIPEIF module before the data is pushed to the IPIPE or RSZ modules. There are two possible dark frame subtractions: the first is with data coming from the VP and the dark frame coming from BL; the second is with data coming from BL and the dark frame coming from the VP.

Figure 120 shows the data path.

Figure 120. ISS ISP IPIPEIF INPSRC1 = 2 and INPSRC2 = 0 Data Path



3.3.2.4.7 ISS ISP IPIPEIF INPSRC1 = 3 and INPSRC2 = 0

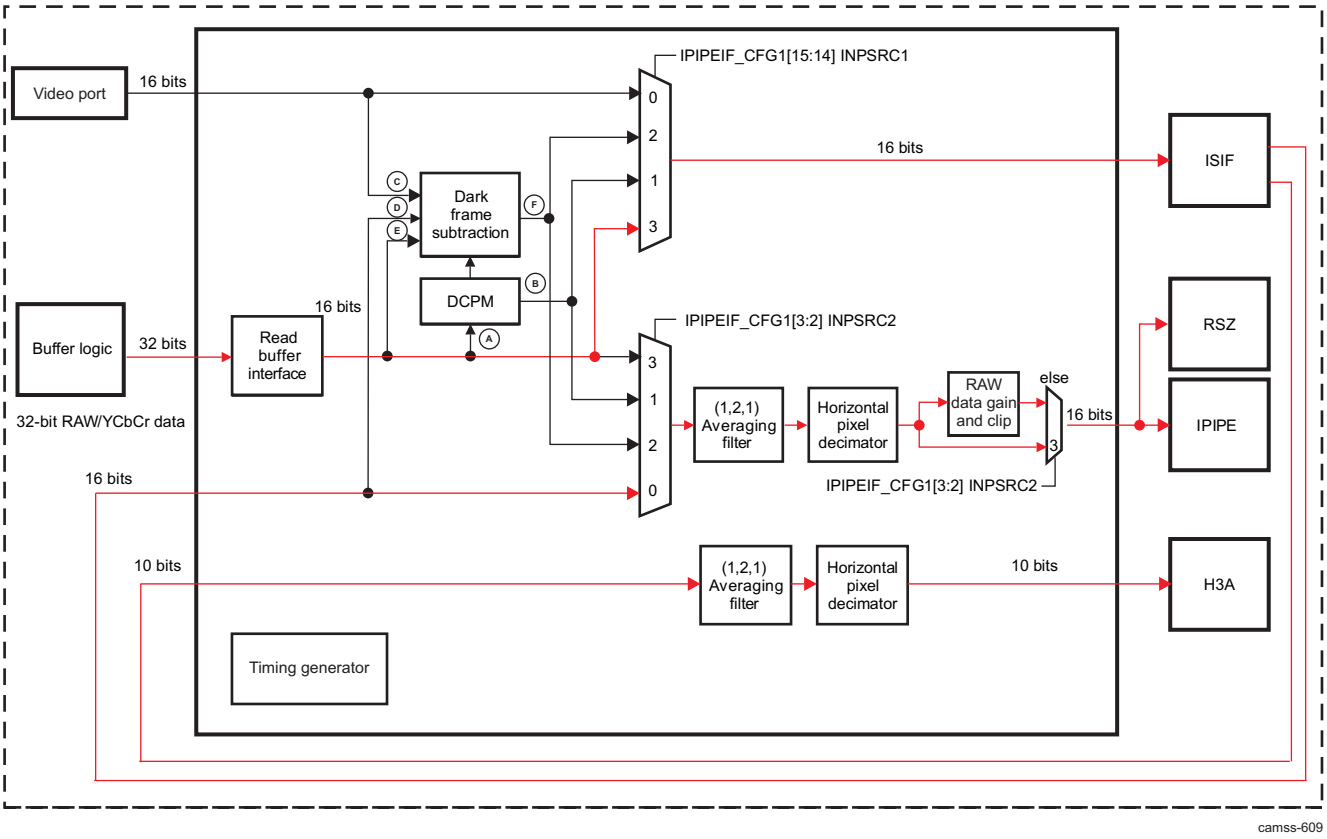
Set the IPIPEIF_CFG1[15:14] INPSRC1 bit field to 3 and the IPIPEIF_CFG1[3:2] INPSRC2 bit field to 0.

This configuration is a memory-to-memory operation. Data is loaded from the SDRAM. Input data is expected as 16 bpp. The ISIF processes the data and sends it back to the IPIPEIF module before the data is pushed to the IPIPE or RSZ module.

In this configuration data is assumed to be YUV only, and H3A and RAW data gain are assumed to be disabled.

Figure 121 shows the data path.

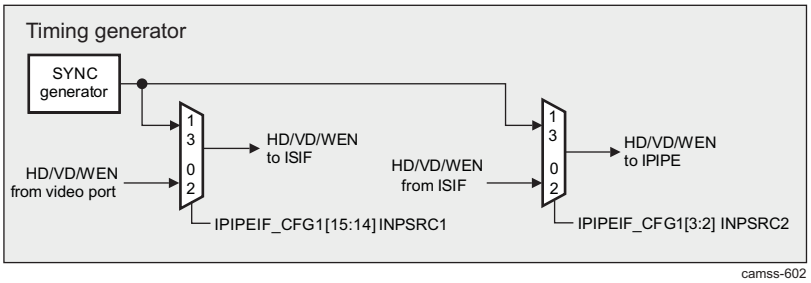
Figure 121. ISS ISP IPIPEIF INPSRC1 = 3 and INPSRC2 = 0 Data Path



3.3.2.5 ISS ISP IPIPEIF Timing Generation

Figure 122 shows the ISS ISP IPIPEIF timing generator submodule.

Figure 122. ISS ISP IPIPEIF Timing Generator Submodule



When the IPIPEIF module input source is from the VP (IPIPEIF_CFG1[15:14] INPSRC1 = 0 or 2) or the ISIF (IPIPEIF_CFG1[3:2] INPSRC2 = 0 or 2) , the IPIPEIF_CFG1[10] CLKSEL bit must be set to 0 so that data is latched using the PCLK, HD, and VD signals from the VP.

When the IPIPEIF module input source is not from the VP (IPIPEIF_CFG1[15:14] INPSRC1 = 1 or 3), the IPIPEIF_CFG1[10] CLKSEL bit must be set to 1 so that the IPIPEIF module generates its proper PCLK, HD, and VD signals (through the use of the SYNC generator). The IPIPEIF_CLKDIV register is then used to select a divide ratio of the SDRAM (DMA) clock for the pixel clock frequency, which is used to clock the data into the PCLK. See Section 3.3.2.5.1, *ISS ISP IPIPEIF Fractional Clock Divider*.

When the IPIPEIF_CFG1[15:14] INPSRC1 or IPIPEIF_CFG1[3:2] INPSRC2 bit field is not set to 0, the IPIPEIF SDRAM data reading and timing generation can be enabled (IPIPEIF_ENABLE[0] ENABLE) in one-shot mode or continuous mode (IPIPEIF_CFG1[0] ONESHOT).

3.3.2.5.1 ISS ISP IPIPEIF Fractional Clock Divider

When the input data of the IPIPEIF module does not come from the VP but from memory, it is useful to have control of the rate at which the data is fetched from memory to avoid overflow conditions or to avoid peak bandwidth requirements. The IPIPEIF_CFG1[10] CLKSEL bit is equal to 1 for fractional divider use.

The ISP clock ISS_FCLK is divided to generate the pixel clock, which goes to the ISIF and IPIPE modules when data is read from memory (IPIPEIF_CFG1[15:14] INPSRC1 = IPIPEIF_CFG1[3:2] INPSRC2 = 1 or 3). The IPIPEIF_CLKDIV register selects the divider ratio: M and N values in the IPIPEIF_CLKDIV[15:0] CLKDIV bit field.

Given an input clock of clock rate ISS_FCLK, the fractional clock divider generates an output clock with average clock rate f_out.

Where f_out = ISS_FCLK x M/N, and M = 1 through 256, and N = 1 through 256.

The fractional clock divider logic is synchronous and uses only the positive clock edge of the input clock.

3.3.2.6 ISS ISP IPIPEIF Decompression (DCPM) Subblock: Unpack and Decompression Function

The IPIPEIF module can read RAW data from memory. The RAW data can be previously packed/compressed into memory. Unpack, A-Law decompression, and DPCM decompression are available in the IPIPEIF module.

Figure 123 shows the DCPM subblock.

Figure 123. ISS ISP IPIPEIF DCPM Subblock

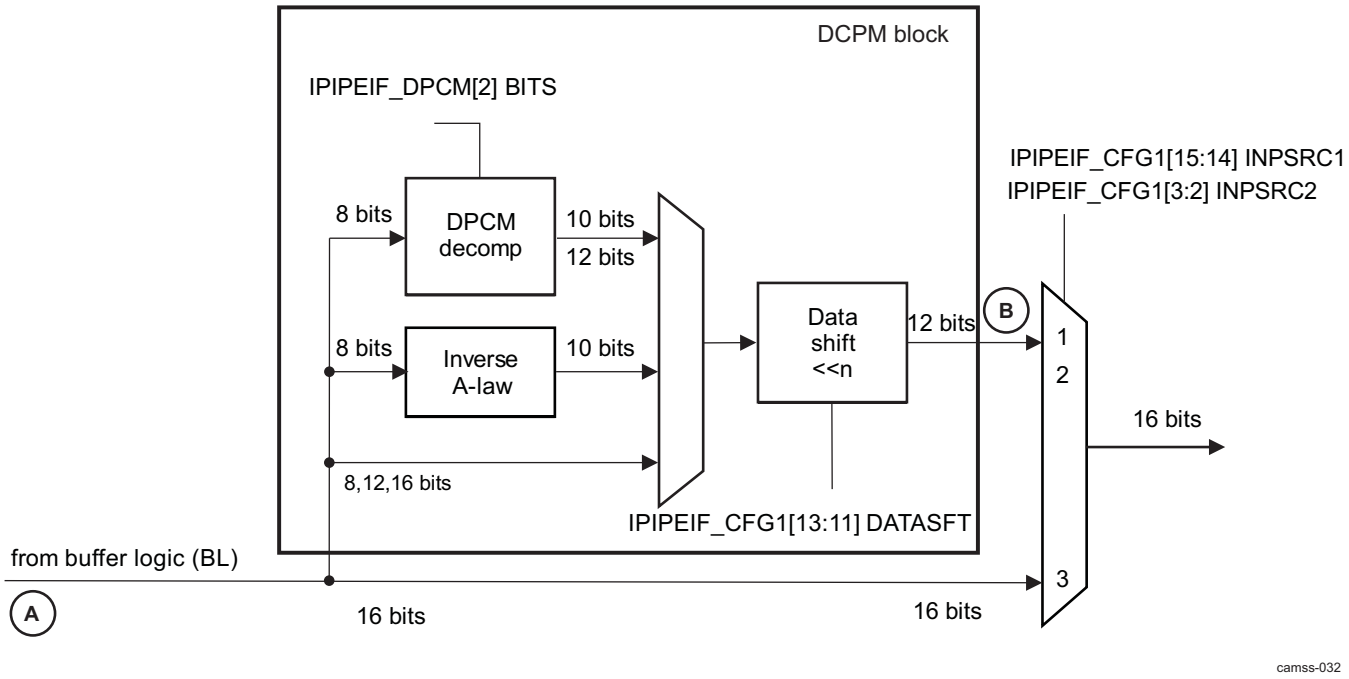


Table 170 provides the possible configuration of the DCPM block.

Table 170. ISS ISP IPIPEIF DCPM Block Possible Configuration

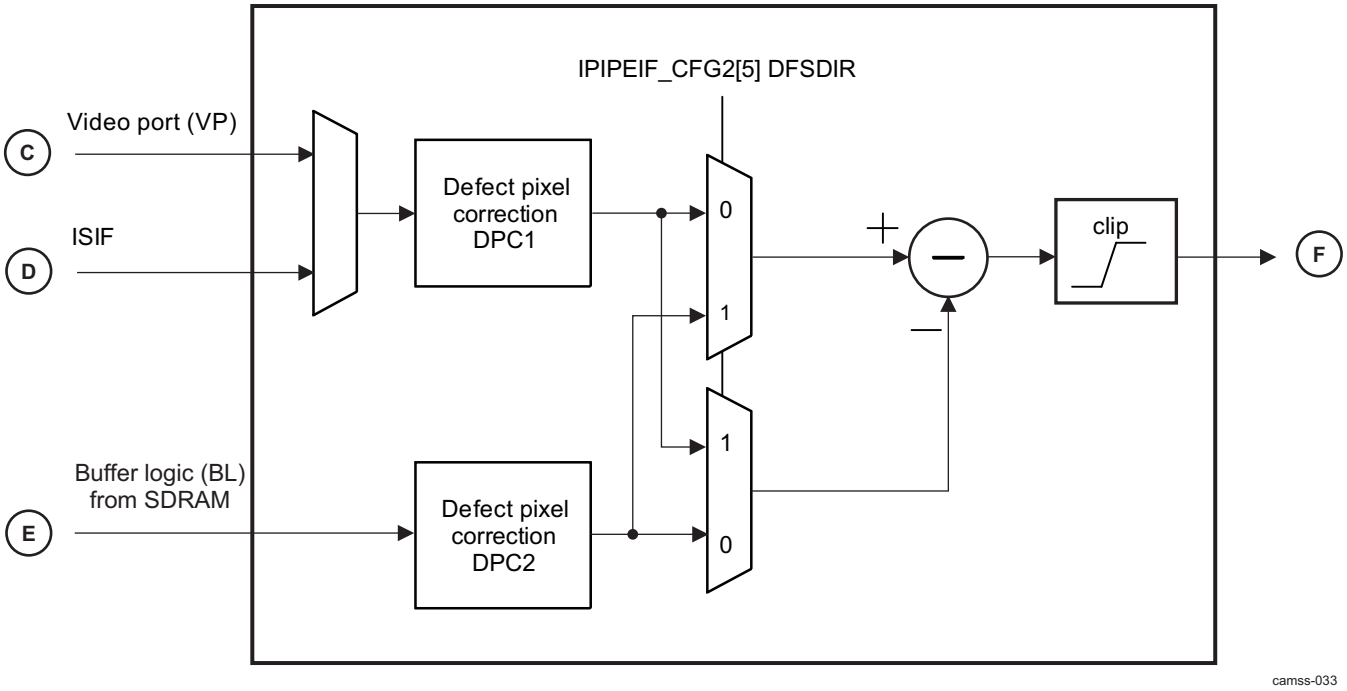
Number of Bits per Pixel at DCPM Block Input	Number of Bits per Pixel at DCPM Block Output	Description	Registers
16	16	It can correspond to YUV4:2:2 or RAW16 data.	IPIPEIF_CFG1[9:8] UNPACK = 0x0 In this configuration, IPIPEIF_CFG1[3:2] INPSRC2 = 0x3. In this configuration data bypasses the DCPM block.
8	8	PACK8: YUV	IPIPEIF_CFG1[9:8] UNPACK = 0x1 IPIPEIF_DPCM[0] ENA = 0x0 IPIPEIF_CFG1[13:11] DATASFT = 0x01. In this configuration an 8-bit packed RAW data is used in DFS mode or YUV4:2:0 pass through. (See Section 3.3.2.13, *ISS ISP IPIPEIF YUV4:2:2 8-bits Packed Data Input Coming From ISIF Module*.)
8	10	DPCM10: The 8-bit input data are DPCM-decompressed in 10 bits.	IPIPEIF_CFG1[9:8] UNPACK = 0x1 IPIPEIF_DPCM[0] ENA = 0x1 IPIPEIF_DPCM[2] BITS= 0x0 Set the IPIPEIF_DPCM[1] PRED(1) (2) IPIPEIF_CFG1[13:11] DATASFT = 0x2
8	12	DPCM12: The 8-bit input data are DPCM-decompressed in 12 bits.	IPIPEIF_CFG1[9:8] UNPACK = 0x1 IPIPEIF_DPCM[0] ENA = 0x1 IPIPEIF_DPCM[2] = 0x1 Set the IPIPEIF_DPCM[1] PRED(1) (2) IPIPEIF_CFG1[13:11] DATASFT = 0x0
8	10	ALAW10: The 8-bit input data were previously A-law compressed. The 8 bits are A-law decompressed and padded with 0.	IPIPEIF_CFG1[9:8] UNPACK = 0x2 IPIPEIF_CFG1[13:11] DATASFT = 0x2
12	12	PACK12: The input is coded on 12 bits and is packed in the SDRAM. This mode is compatible with the ISIF module packing capability. If inverse A-law and DPCM decompression are not enabled, the data read from the SDRAM can be shifted by the IPIPEIF_CFG1[13:11] DATASFT bit field to select which 12 bits to use. MSB input data bit must be shifted such that it corresponds to bit 11 after the shift.	IPIPEIF_CFG1[9:8] UNPACK = 0x3 Set the IPIPEIF_CFG1[13:11] DATASFT

(1) The simple predictor uses only the value of the previous same color component as a prediction value. Therefore, only 2-pixel memory is required. It is typically used for 10–8–10 or 12–8–12 bit conversions.

(2) The advanced predictor uses four previous pixel values, when the prediction value is evaluated. This means that the values of the other color components are also used, when the prediction value is defined. Therefore, the advanced predictor is slightly better than the simple predictor but consumes more power and memory. It can, however, improve image quality. It is typically used for 10–7–10 and 10–6–10 bit conversions.

3.3.2.7 ISS ISP IPIPEIF Dark-Frame Subtraction Functionality

Figure 124. ISS ISP IPIPEIF Dark-Frame Subtraction Subblock



The dark-frame subtract function is used to remove noise from the sensor. Typically, the ISIF module previously writes a dark frame (frame captured when the shutter is closed) to SDRAM using 8 bits of linear data packed into 2 pixels per 16 bits.

In this mode, RAW data from the ISIF and SDRAM is used. Data can also be read from SDRAM with the IPIPEIF_CFG1[9:8] UNPACK bit field set to 1. Each pixel read from SDRAM is subtracted from each pixel sent from the VP or ISIF.

The mux at the input of the dark-frame subtraction subblock is implicitly controlled by the selection of the IPIPEIF_CFG1[15:14] INPSRC1 and IPIPEIF_CFG1[3:2] INPSRC2 bit fields:

- When IPIPEIF_CFG1[15:14] INPSRC1 = 0x2, the selected input ports are VP and BL. Therefore, the dark frame operation can be:
  - Dark frame = VP – BL
  - Dark frame = BL – VP
- When IPIPEIF_CFG1[3:2] INPSRC2 = 0x2, the selected input ports are ISIF and BL. Therefore, the dark frame operation can be:
  - Dark frame = ISIF – BL
  - Dark frame = BL – ISIF

The output of the dark frame subtract operation is 12-bits wide (U12Q0). There must be adequate SDRAM bandwidth if this feature is enabled. If the data fetched from memory arrives late, an underflow bit (IPIPEIF_DTUF) must be triggered to know it .

3.3.2.7.1 ISS ISP IPIPEIF Defect Pixel Correction

NOTE: For DPC memory access locations, see Section 3.3.8.

A simple DPC can be applied to the ISIF or VP input data path and SDRAM input data path, respectively. This DPC algorithm is intended to correct hot pixels during RAW dark frame acquisition or dark frame readout from SDRAM before dark frame subtraction.

The following code describes DPC algorithm:

```
If (image(n)  TH) || ((image(2)  TH) (image(n+2) TH)
 image(n) = image(n)
Else if image(2)  TH
 image(n) = image(n+2)
Else if image(n+2)  TH
 image(n) = image(2)
Else
 image(n) = (image(2)+ image(n+2))/2

// Where TH is equal to IPIPEIF_DPC1[11:0] TH for DPC1
// Where TH is equal to IPIPEIF_DPC2[11:0] TH for DPC2
```

The IPIPEIF_DPC1[12] ENA bit enables DPC for the VP/ISIF input path, and the IPIPEIF_DPC2[12] ENA bit enables DPC for the SDRAM input path. The algorithm requires a threshold value that is set by the IPIPEIF_DPC1[11:0] TH or IPIPEIF_DPC2[11:0] TH bit field that is a 12-bit unsigned value.

3.3.2.7.2 ISS ISP IPIPEIF DFS Subtraction Direction

The IPIPEIF_CFG2[5] DFSDIR bit selects how the DFS subtraction is performed.

- Set theIPIPEIF_CFG2[5] DFSDIR bit to 0 when the RAW data is coming from the VP/ISIF and the dark frame is stored in SDRAM.
- Set the IPIPEIF_CFG2[5] DFSDIR bit to 1 when the RAW data is coming from SDRAM and the dark frame is coming from the VP/ISIF.

After subtraction, a clip ensures that the value is not negative.

Table 171 lists the different modes supported in DFS.

Table 171. ISS ISP IPIPEIF DFS Modes Supported

Description	DFDIR Value
Dark frame subtract of 8-bit RAW image stored in SDRAM from image from VP	IPIPEIF_CFG2[5] DFSDIR = 0x0
Dark frame subtract of 8-bit RAW image stored in SDRAM from image from ISIF	IPIPEIF_CFG2[5] DFSDIR = 0x0
Dark frame subtract of 8-bit RAW image from VP from image read from SDRAM through the BL	IPIPEIF_CFG2[5] DFSDIR = 0x1

NOTE: DFS input depends on the INPSRC1 and INPSRC2 settings.

3.3.2.8 ISS ISP IPIPEIF (1, 2, 1) Averaging Filter for IPIPE Data Path

The averaging filter acts as an anti-aliasing low-pass filter for the horizontal pixel decimator. Usually, when horizontal pixel decimation is enabled (IPIPEIF_CFG1[1] DECIM), the averaging filter must also be enabled to avoid aliasing artifacts. The averaging filter can be enabled by setting the IPIPEIF_CFG1[7] AVGFILT bit. It operates on every other pixel (same color) in RAW Bayer input or every Y component in YCbCr data in the following equation:

output = (input[i – 1] + 2 * input[i] + input[i + 1]) >>2

The averaging filter operates on every other pixel (same color) in RAW Bayer input or YUV data.

If the data is YUV4:2:2, the option to average and decimate is given under the conditions listed in Table 172.

Table 172. ISS ISP IPIPEIF Averaging Filter Conditions for YUV4:2:2 Data

IPIPEIF_CFG1[3:2] INPSRC2	IPIPEIF_CFG2[3] YUV16	Comments
0	1	YUV4:2:2 data is coming from the ISIF module. Averager and decimation is possible on the data path to the IPIPE module.
1	1	YUV4:2:2 data is read from SDRAM. Averager and decimation is possible on the data path to the IPIPE module.
3		YUV4:2:2 data is read from SDRAM. Averager and decimation is possible on the data path to the IPIPE module.
Other value	Other value	For YUV4:2:2 data, averager and decimation is not possible on the data path to the IPIPE module.

The averager implements a (1, 2, 1) FIR filter on Luma and Chroma. The following registers have a part in the behavior of the YUV data averaging and decimation:

- IPIPEIF_INIRSZ[12:0] INIRSZ
- IPIPEIF_CFG1[7] AVGFILT
- IPIPEIF_CFG1[1] DECIM

3.3.2.9 ISS ISP IPIPEIF Horizontal Pixel Decimator (Downsizer) for IPIPE Data Path

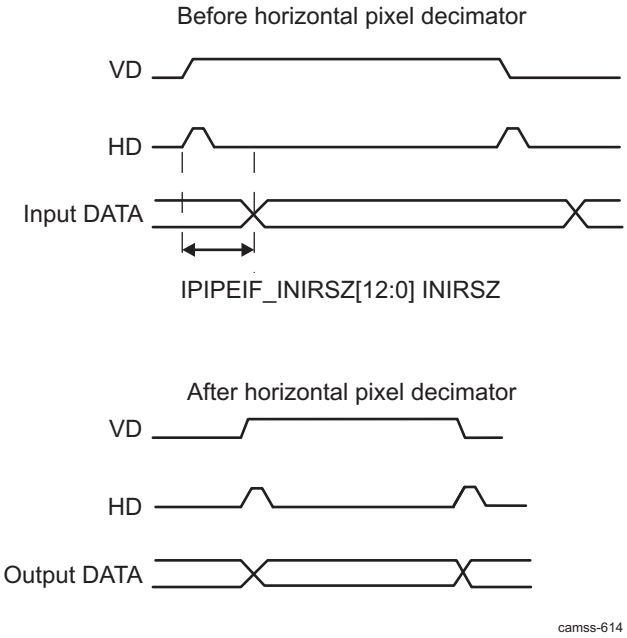
The IPIPE input is limited to 5376 pixels per horizontal line due to restrictions in the line memory width in the IPIPE.

To process image sensor resolutions with more than 5376 pixels per line with no resolution loss, vertical frame division mode (FDM) must be used; that is, the image must be divided into vertical chunks of less than 5376 pixels, and each chunk must be processed sequentially by the ISP. FDM is memory-to-memory processing and is not supported on the fly.

Alternatively, if a loss in resolution is acceptable, the line width decimator (the IPIPEIF_CFG1[1] DECIM bit) can be enabled to downsample the input lines to a width equal to or less than the 5376 pixel maximum. The resize ratio (16/RSZ) can be configured by programming the IPIPEIF_RSZ[6:0] RSZ bit field to be within the range from 16 to 112 to give a resampling range from 1x to 1/7x.

When ALNSYNC is enabled (IPIPEIF_INIRSZ[13] ALNSYNC = 0x1), the IPIPEIF_INIRSZ[12:0] INIRSZ pixels are skipped (from the HD position) before the horizontal pixel decimator, as shown in Figure 125.

Figure 125. ISS ISP IPIPEIF Resizer Offset Definition



3.3.2.10 ISS ISP IPIPEIF RAW Data Gain for IPIPE Data Path

A gain factor ranging from 0.00195(1/512) to 1.99805(1023/512) is multiplied to the RAW output of the IPIPEIF. The gain is not applied if the input data is YCbCr. The gain constant is set in the IPIPEIF_GAIN[9:0] GAIN bit field using U10Q9 format.

The output value is clipped after gain control through the value of the IPIPEIF_OCLIP[11:0] OCLIP bit field.

3.3.2.11 ISS ISP IPIPEIF (1, 2 ,1) Averaging Filter for H3A Data Path

The averaging filter acts as an anti-aliasing low-pass filter for the horizontal pixel decimator. Usually, when horizontal pixel decimation is enabled (the IPIPEIF_RSZ3A[9] DECIM bit), the averaging filter must also be enabled to avoid aliasing artifacts. The averaging filter can also be used to reduce noise before H3A statistics generation. It operates on every other pixel (the same color) in a RAW Bayer input or every Y component in YCbCr data. The averaging filter can be enabled by setting the IPIPEIF_RSZ3A[8] AVGFILT bit, and it operates with the following equation:

output = (input[i – 1] + 2 * input[i] + input[i + 1]) >>2

3.3.2.12 ISS ISP IPIPEIF Horizontal Pixel Decimator (Downsizer) for H3A Data Path

The H3A input is limited to 3008 pixels per horizontal line due to restrictions in the line memory width in the H3A.

To process image-sensor resolutions with more than 3008 pixels per line with no resolution loss, vertical frame division mode (FDM must be used); that is, the image must be divided into vertical chunks of less than 3008 pixels, and each chunk must be processed sequentially by the ISP. FDM is memory-to-memory processing and is not supported on the fly.

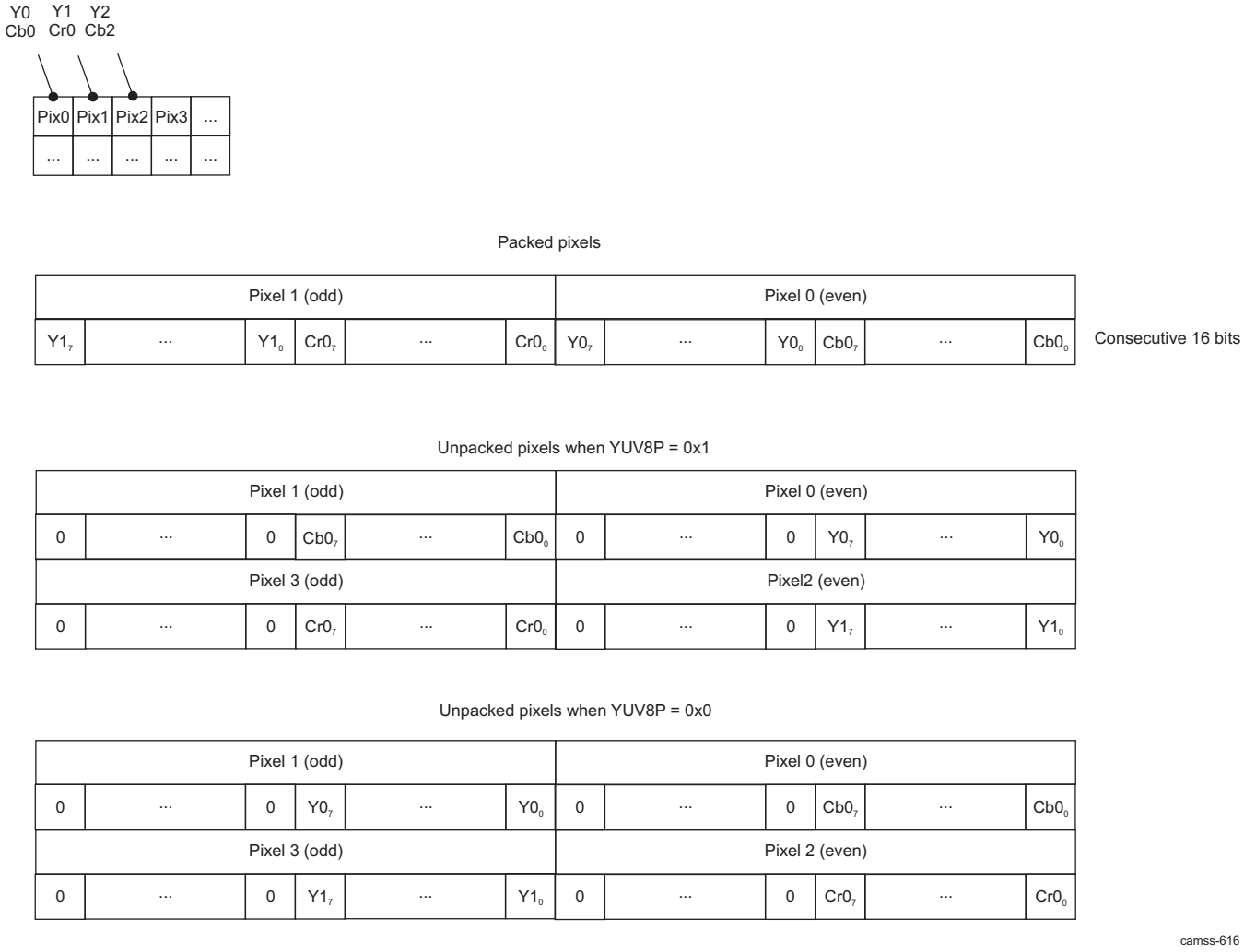
Alternatively, if loss of resolution is acceptable, the line width decimator (the IPIPEIF_RSZ3A[9] DECIM bit) can be enabled to downsample the input lines to a width equal to or less than the 3008 pixel maximum. The resize ratio (16/RSZ) can be configured by programming the IPIPEIF_RSZ3A[6:0] RSZ bit field to be within the range from 16 to 112 to give a resampling range from 1x to 1/7x.

When ALNSYNC is enabled (IPIPEIF_INIRSZ3A[13] ALNSYNC = 0x1), the IPIPEIF_INIRSZ3A[12:0] INIRSZ pixels are skipped (from the HD position) before the horizontal pixel decimator (see Figure 125).

3.3.2.13 ISS ISP IPIPEIF YUV4:2:2 8-bits Packed Data Input Coming From ISIF Module

This section applies when data coming from the ISIF is 8 bits (IPIPEIF_CFG2[6] YUV8 = 0x1). When IPIPEIF_CFG1[3:2] INPSRC2 = 0 and IPIPEIF_CFG2[3] YUV16 = 1, the 8-bit YUV data are transformed into 16-bit YUV data. The way the data are unpacked from 8 bits to 16 bits is controlled by the IPIPEIF_CFG2[7] YUV8P bit. See Figure 126.

Figure 126. ISS ISP IPIPEIF YUV8P Settings



3.3.2.14 ISS ISP IPIPEIF YUV4:2:0 Data Input for Memory-to-Memory Resize Operations

The ISP RSZ module can resize YUV4:2:0 data. The YUV4:2:0 data can come from the memory (through BL) or from the VP through the SC module, which reads data from SDRAM and pushes it to the IPIPEIF module VP. The possible data paths are:

- SDRAM - IPIPEIF (BL) - RESIZER - SDRAM
- SDRAM - SC - IPIPEIF (VP) - RESIZER - SDRAM
- SDRAM - IPIPEIF (BL) - IPIPE - RESIZER - SDRAM
- SDRAM - SC - IPIPEIF (BL) - IPIPE - RESIZER - SDRAM

When the data comes from BL, the IPIPEIF module must be set up to process the luminance data first, and then the chrominance data.

- For 420Y (first pass):
  - IPIPEIF_CFG1[15:14] INPSRC1 = 0x1 (SDRAM data input)
  - IPIPEIF_CFG1[3:2] INPSRC2 = 0x1 (SDRAM data input)
  - IPIPEIF_CFG1[13:11] DATASFT = 0x0 (no data shift)
  - IPIPEIF_CFG1[9:8] UNPACK = 0x1 (data packed on 8 bits)
  - IPIPEIF_CFG2[3] YUV16 = 0x1 (data_input [7:0] = 0 and data_input [15:8] = valid)
- For 420C (second pass):
  - IPIPEIF_CFG1[15:14] INPSRC1 = 0x1 (SDRAM data input)
  - IPIPEIF_CFG1[3:2] INPSRC2 = 0x1 (SDRAM data input)
  - IPIPEIF_CFG1[13:11] DATASFT = 0x0 (no data shift)
  - IPIPEIF_CFG1[9:8] UNPACK = 0x1 (data packed on 8 bits)
  - IPIPEIF_CFG2[3] YUV16 = 0x0 (data_input[7:0] = valid and data_input [15:8] = 0)

3.3.2.15 ISS ISP IPIPEIF Module Events and Status Checking

The IPIPEIF module generates an IPIPEIF event through the IPIPEIF_IRQ interrupt at the end of each frame. This interrupt is set through the ISP5_IRQENABLE_SET_i[9] IPIPEIF_IRQ bit. The input interrupt source generation is selected through the IPIPEIF_CFG2[0] INTSW bit in a certain configuration. The following pseudo code describes INTSW.

```
if (IPIPEIF_CFG2[0] INTSW==0) // Interrupt source from VP
 if (IPIPEIF_CFG1[15:14] INPSRC1==1,2 or 3)
 if (CFG1.ONESHOT==1) // In one shot mode
 Interrupt happens at the end of frame
 else // In continuous mode
 Interrupt is the start position of VD which is generated by IPIPEIF timing generator
 else // IPIPEIF_CFG1[15:14] INPSRC1==0, data is from VP
 Interrupt is the start position of VD from VP

else // Interrupt source from ISIF: IPIPEIF_CFG2[0] INTSW==1
 if (IPIPEIF_CFG1[3:2] INPSRC2==1,2 or 3)
 if (CFG1.ONESHOT==1) // In one shot mode
 Interrupt happens at the end of frame
 else
 Interrupt is the start position of VD which is generated by IPIPEIF timing generator
 else // IPIPEIF_CFG1[3:2] INPSRC2==0, data is from ISIF
 Interrupt is the start position of VD from ISIF
```

In addition to this interrupt, the host must check the IPIPEIF_DTUF status flag of the ISP5_IRQSTATUS_RAW2_i[1] IPIPEIF_UDF bit (if this is enabled and mapped to the ISP IRQ lines) to see if an underflow occurred. For more information, see Section 3.4.2.4.

3.3.3 ISS ISP IPIPE Functional Description

3.3.3.1 ISS ISP IPIPE Overview

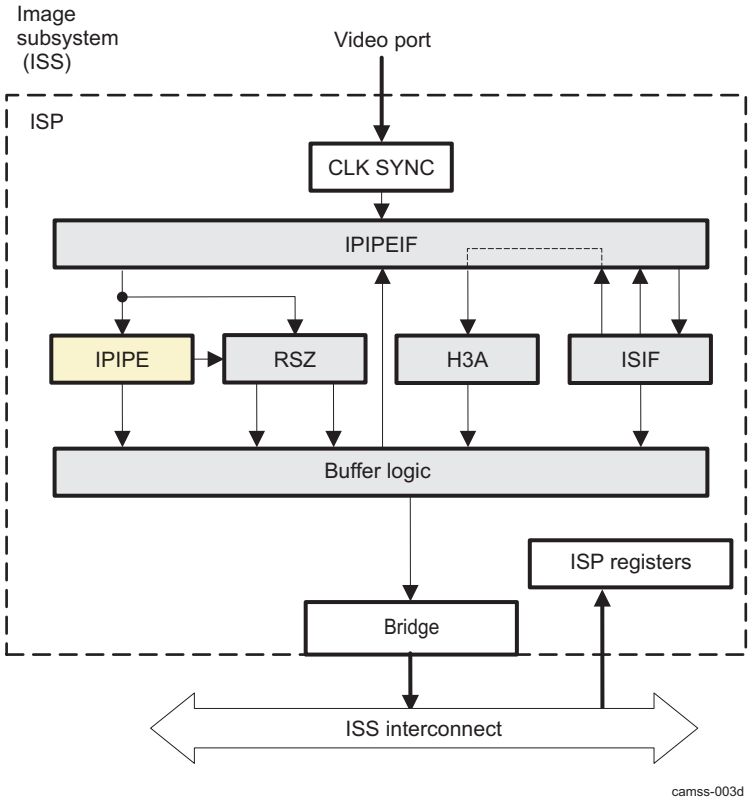
- The input interface extracts the valid region from the Bayer RAW data:
  - Up to 12-bit input pixel resolution
  - Requires at least 8 pixels for horizontal blanking and four lines for vertical blanking. In one-shot mode, 16 blanking lines after processing area are required.
  - The maximum horizontal and vertical offset of IPIPE processing area from synchronous signal is 65534.
  - Supports RGB Bayer pattern for input
- The DPC module fixes defect pixels using two methods: LUT-based and on-the-fly adaptive.
- The 2D noise-filter module reduces noise in RAW data.
- The green-imbalance-correction (GIC) module reduces Gb/Gr difference to remove line crawl noise.
- The white balance module applies offset and gain adjustments to each color.
- The color filter array (CFA) interpolation module implements CFA interpolation. The output from the CFA interpolation module is RGB-4:4:4 formatted data. CFA also reduces aliasing caused by undersampling by digital anti-aliasing (DAA).
- The RGB2RGB blending module applies a 3 × 3 matrix transform to the RGB data generated by the CFA interpolation module.
- The gamma correction module independently applies gamma correction to each RGB component. Gamma is implemented using a piece-wise linear interpolation approach with a 512-entry LUT for each color.
- The 2nd RGB2RGB blending module applies a 3 × 3 matrix transform to the RGB data after gamma correction.
- 3D-LUT converts RGB data to RGB data using 9 × 9 × 9 table and tetrahedral interpolation.
- The RGB2YCbCr conversion module applies 3 × 3 matrix transformation to the RGB data to convert it to YCbCr data. This module also implements offset. The global brightness and contrast enhancement module fixes brightness and contrast tone.
- The 4:2:2 conversion module applies the chroma low pass filter and downsampling to Cb and Cr to convert 4:4:4 data to 4:2:2 data.
- The 2D edge-enhancer module improves image clarity with a luminance nonlinear filter.
- The chroma artifact reduction module reduces color artifacts using gain control and a 2D median filter.
- The output interface module transfers data from IPIPE to SDRAM in the form of one YCbCr (4:2:2 or 4:2:0), RGB (32/16 bits), or Bayer data.
- The histogram function can record histograms of up to four distinct areas into up to 256 bins.
- The boxcar function makes 1/8 or 1/16 size (1/64 or 1/256 in area) images.
- The boundary signal calculator (BSC) makes vectors of row and column summations.

IPIPE has four different processing paths:

- Case 1: IPIPE reads BAYER RAW data and applies all IPIPE functions and stores the YCbCr (or RGB) data to SDRAM.
- Case 2: IPIPE reads BAYER RAW data and stores the Bayer data after white balance to SDRAM.
- Case 3: IPIPE reads YCbCr-4:2:2 data and applies edge enhance, chroma suppression, and resize to output YCbCr data to SDRAM.
- Case 4: IPIPE reads YCbCr-4:2:0 data and applies resize to output YCbCr data to SDRAM.

Figure 127 shows the connections from the IPIPE module to other submodules of the ISP.

Figure 127. ISS ISP IPIPE High-Level Diagram

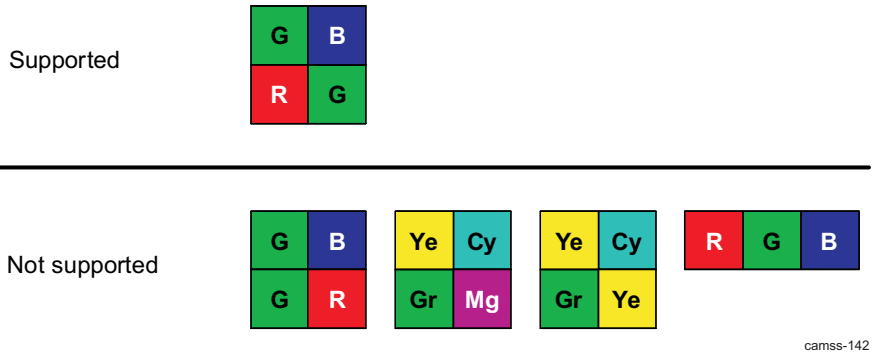


3.3.3.2 ISS ISP IPIPE Top-Level Block Diagram

The IPIPE is a programmable hardware image-processing module that generates image data in YCbCr-4:2:2 or YCbCr-4:2:0 format from RAW CCD/CMOS data. The IPIPE module supports output of Bayer data.

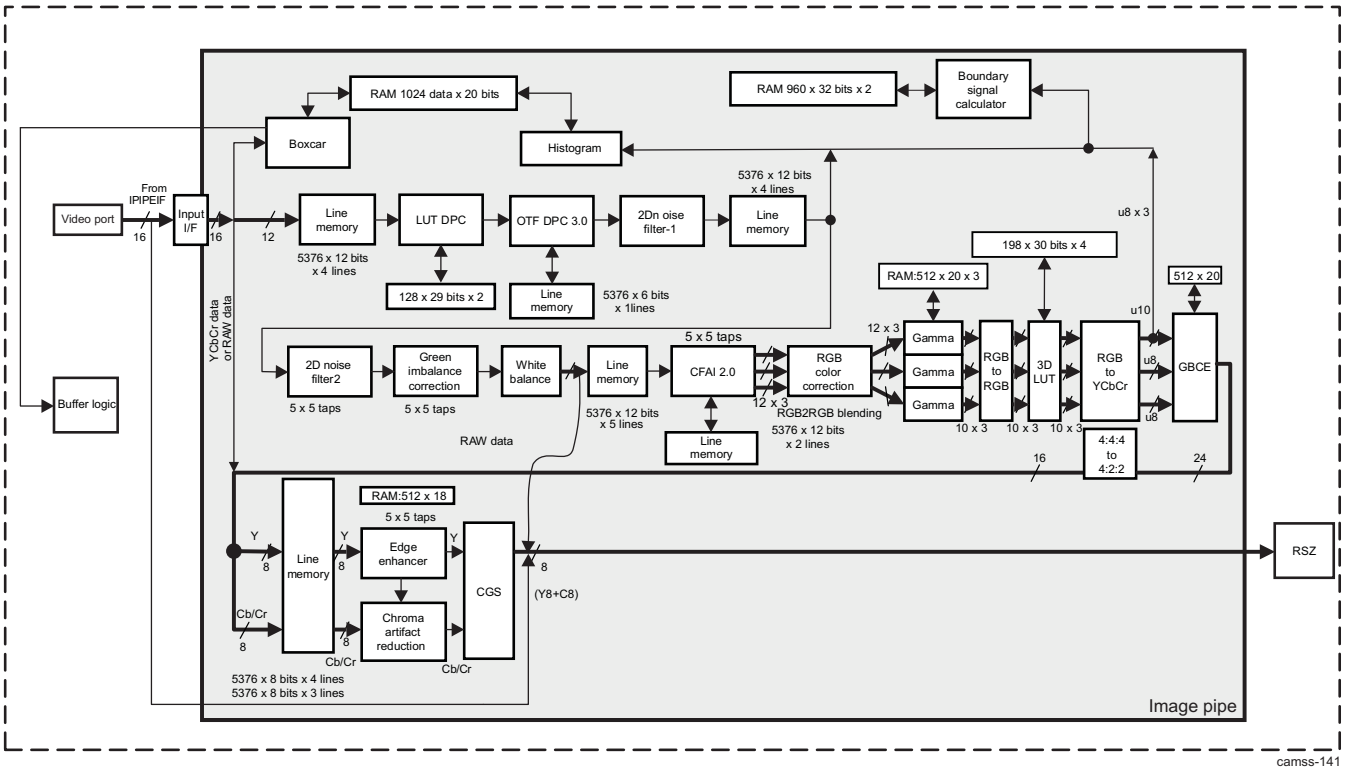
The IPIPE module supports RAW data in Bayer format, as shown in Figure 128. Other RGB formats or complimentary color formats are not supported.

Figure 128. ISS ISP IPIPE Supported CFA Format



As shown in Figure 129, many internal modules are used to process Bayer data into YCbCr data

Figure 129. ISS ISP IPIPE Module Block Diagram



3.3.3.3 ISS ISP IPIPE Input Interface

The IPIPE module receives 12-bit RAW image data or 16-bit YCbCr data through the IPIPEIF module. The IPIPE module can work with up to 5376 pixels in each horizontal line, except in RAW pass-through mode. If the image width is larger than 5376, the image must be scaled down at the IPIPEIF module level. Otherwise, the input image must be split into several blocks.

If the input data is YCbCr, all RGB processing modules are skipped, and only edge enhancer and chroma suppression are applied to the input data.

If the input data is YCbCr-4:2:0, only Y or C can be processed at a time, and only the resizer process can be applied. Because the resizer is outside the ISIF module, the data is passed to it directly by skipping the RGB and YCbCr processing modules.

In RAW pass-through mode, images up to 8190 pixels per line can be processed. In RAW pass-through mode, the input data is written out directly to SDRAM.

The IPIPE module is enabled through the IPIPE_SRC_EN[0] EN bit.

The IPIPEIF module must be selected as the IPIPE module source with the IPIPE_SRC_MODE[1] WRT bit set to 1 from the input port of the IPIPEIF. This is required to enable and transfer data properly from the interface to the IPIPE.

The IPIPE module has two processing modes, which can be selected through the IPIPE_SRC_MODE[0] OST bit:

- One-shot mode: IPIPE_SRC_MODE[0] OST = 0x1
- Free-run mode: IPIPE_SRC_MODE[0] OST = 0x0

The input and output formats are selected in the IPIPE_SRC_FMT[1:0] FMT bit field (see Table 173).

Table 173. ISS ISP IPIPE Input and Output Selections

IPIPE_SRC_FMT[1:0] FMT	IPIPE Module Input	IPIPE Module Output
0x0	RAW Bayer	YCbCr or RGB
0x1	RAW Bayer	RAW Bayer
0x2	RAW Bayer	Disabled
0x3	YCbCr 16 bits	YCbCr

The input to the IPIPE module is in the formats (YCbCr-8bit is not allowed) shown in Figure 130.

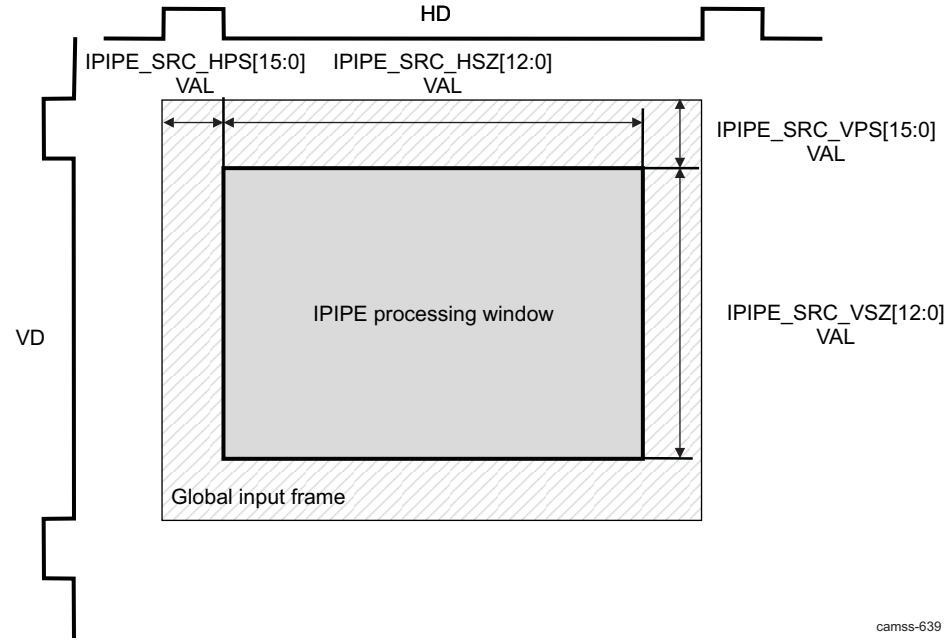
Figure 130. ISS ISP IPIPE Module Input Format

IPIPE input	RAW	LOW	LOW	LOW	LOW	RAW11	RAW10	RAW9	RAW8	RAW7	RAW6	RAW5	RAW4	RAW3	RAW2	RAW1	RAW0
	YCbCr 16b	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Cb/Cr7	Cb/Cr6	Cb/Cr5	Cb/Cr4	Cb/Cr3	Cb/Cr2	Cb/Cr1	Cb/Cr0
	Y 8bit	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	LOW	LOW	LOW	LOW	LOW	LOW	LOW	LOW
	CbCr 8bit	LOW	LOW	LOW	LOW	LOW	LOW	LOW	LOW	Cb/Cr7	Cb/Cr6	Cb/Cr5	Cb/Cr4	Cb/Cr3	Cb/Cr2	Cb/Cr1	Cb/Cr0

camss-143

The window to process can be defined by its vertical and horizontal start position (IPIPE_SRC_VPS and IPIPE_SRC_HPS) and vertical and horizontal size (IPIPE_SRC_VSZ and IPIPE_SRC_HSZ). Figure 131 shows the window settings for processing.

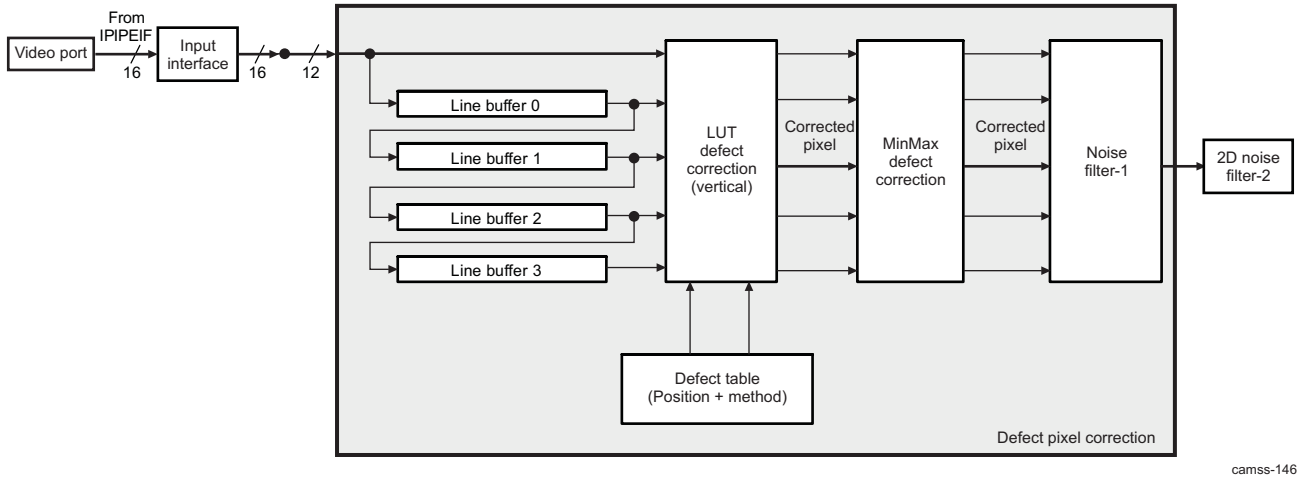
Figure 131. ISS ISP IPIPE Module Processing Window Settings



3.3.3.4 ISS ISP IPIPE Defect Pixel Correction

The DPC module corrects defect pixels using two methods: look-up-table-based method (LUT DPC) and on-the-fly adaptive method (OTF DPC). Figure 132 shows defect pixel correction.

Figure 132. ISS ISP IPIPE Defect Pixel Correction



3.3.3.4.1 ISS ISP IPIPE LUT Defect Pixel Correction (LUT DPC)

LUT DPC is the first stage of the IPIPE image-processing pipeline. The LUT DPC module corrects defects in input data. It supports up to a 256-defect point table. However, the table can be renewed as required during image processing. Therefore, the maximum amount of defect information is limited only by system-level performance.

The module uses two sets of 128 × 29 memories to hold defect information. The table contains the information of horizontal position (13 bits), vertical position (13 bits), and correction method (3 bits), as shown in Table 174. The LUT DPC is enabled through the IPIPE_DPC_LUT_EN[0] EN bit.

Table 174. ISS ISP IPIPE Defect Information Packing

Correction Method	Vertical Position	Horizontal Position
28…26	25…13	12…0

The information must be listed in the order of "from left to right" and "from the top to bottom." The first position in the defect information table and the number of defects that are used can be specified. The address of the table must be programmed in the IPIPE_DPC_LUT_ADR[9:0] ADR bit field. Thus, the address of the first valid data is stated.

The LUT type can be:

- With a finite number of entries:
  - IPIPE_DPC_LUT_SEL[1] TBL = 0x0
  - The size of the LUT is set in the IPIPE_DPC_LUT_SIZ[9:0] SIZ bit field.
- With an infinite number of entries:
  - IPIPE_DPC_LUT_SEL[1] TBL = 0x1

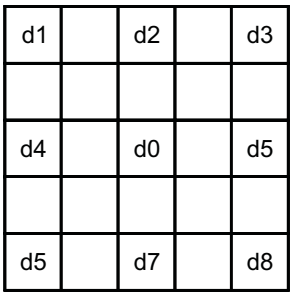
The correction methods, set in Table 174, are described in Table 175.

Table 175. ISS ISP IPIPE Correction Method Description

Correction Method	d0 =	Comment
0	Black or white dot	Replace with a black (or white) dot to force OTF-DPC to work on the pixel. White or black dot replacement can be selected through the IPIPE_DPC_LUT_SEL[0] DOT field.
1	d4	Copy from left
2	d5	Copy from right
3	(d4 + d5)/2	Horizontal interpolation
4	(d2 + d7)/2	Vertical interpolation
5	d2	Copy from top
6	d7	Copy from bottom
7	(d2 + d4 + d5 + d7)/2	2D interpolation

The pixels in the defect correction algorithm are numbered as shown in Figure 133.

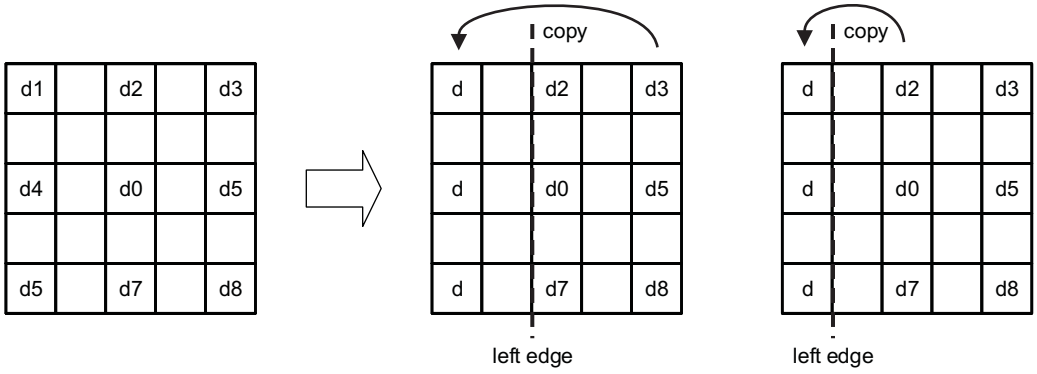
Figure 133. ISS ISP IPIPE Pixel Numbering in Defect Correction Algorithm



camss-145

The pixels at the edges are mirrored in the way described in Figure 134. The figure shows the typical correction by overwriting far-edge pixels and mirroring them with other edge pixels. The example shows how by using a noise filter the correct-by-definition pixels are copied over bad pixels.

Figure 134. ISS ISP IPIPE Mirroring in Defect Correction and Noise Filter



3.3.3.4.2 ISS ISP IPIPE OTF Defect Pixel Correction (OTF DPC)

OTF DPC is enabled through the IPIPE_DPC_OTF_EN[0] EN bit.

In OTF DPC correction, each pixel is compared with the surrounding 8 pixels in the same color plane. The defect pixel is detected and modified using one of the three correction methods:

- Adaptive OTF DPC method 1: IPIPE_DPC_OTF_TYP[1] TYP = 0x0 and IPIPE_DPC_OTF_TYP[0] ALG = 0x0
- Adaptive OTF DPC method 2: IPIPE_DPC_OTF_TYP[1] TYP = 0x1 and IPIPE_DPC_OTF_TYP[0] ALG = 0x0
- Adaptive OTF DPC method 3: IPIPE_DPC_OTF_TYP[1] TYP = 0x1 and IPIPE_DPC_OTF_TYP[0] ALG = 0x1

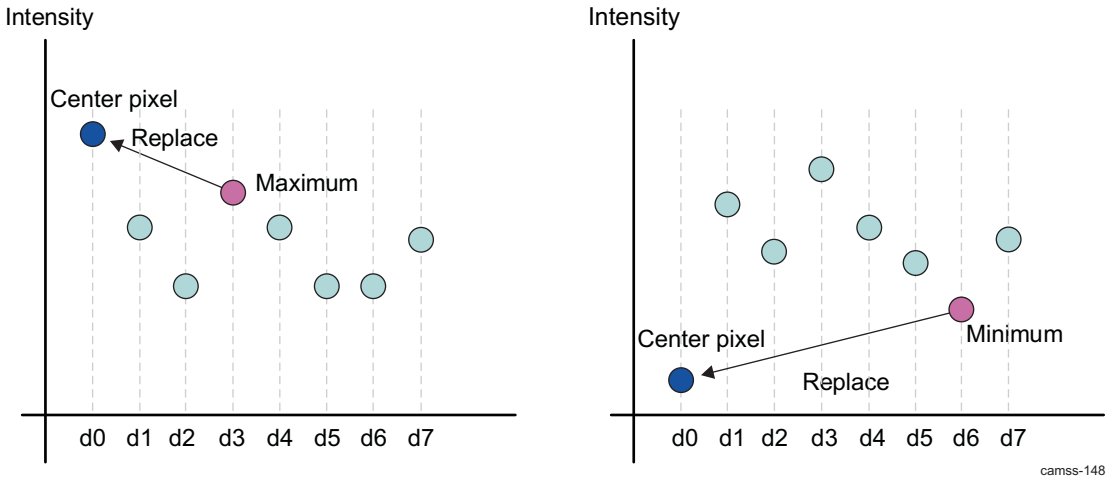
3.3.3.4.2.1 ISS ISP IPIPE Adaptive OTF DPC Method 1

In adaptive OTF DPC method 1, if the center pixel d0 is larger than the maximum among them, the output is the second largest value. If the center pixel d0 is the minimum, it is replaced by the second smallest value. If the center pixel is not the maximum or minimum, it is the output.

The parameters for the different thresholds shall be set as below:

- IPIPE_DPC_OTF_2_D_THR_R[11:0] VAL = 0
- IPIPE_DPC_OTF_2_D_THR_GR[11:0] VAL = 0
- IPIPE_DPC_OTF_2_D_THR_GB[11:0] VAL = 0
- IPIPE_DPC_OTF_2_D_THR_B[11:0] VAL = 0
- IPIPE_DPC_OTF_2_C_THR_R[11:0] VAL = 4095
- IPIPE_DPC_OTF_2_C_THR_GR[11:0] VAL = 4095
- IPIPE_DPC_OTF_2_C_THR_GB[11:0] VAL = 4095
- IPIPE_DPC_OTF_2_C_THR_B[11:0] VAL = 4095

Figure 135. ISS ISP IPIPE Adaptive OTF DPC Method 1 Defect Correction



3.3.3.4.2.2 ISS ISP IPIPE Adaptive OTF DPC Method 2

In adaptive OTF DPC method 2, the center pixel is compared with the second maximum and the second minimum pixels among the surrounding 8 pixels.

The parameters for the different thresholds can be set independently for each color component. Thresholds for each color are configured in this method:

- Detection threshold, thr_D:
  - IPIPE_DPC_OTF_2_D_THR_R[11:0] VAL
  - IPIPE_DPC_OTF_2_D_THR_GR[11:0] VAL
  - IPIPE_DPC_OTF_2_D_THR_GB[11:0] VAL
  - IPIPE_DPC_OTF_2_D_THR_B[11:0] VAL
- Correction threshold, thr_C:
  - IPIPE_DPC_OTF_2_C_THR_R[11:0] VAL
  - IPIPE_DPC_OTF_2_C_THR_GR[11:0] VAL
  - IPIPE_DPC_OTF_2_C_THR_GB[11:0] VAL
  - IPIPE_DPC_OTF_2_C_THR_B[11:0] VAL

After the comparison with thresholds, the center pixel d0 is replaced by d0' in the manner shown in Figure 136

Figure 136. ISS ISP IPIPE Center Pixel d0 Replacement in Adaptive OTF DPC Method 2 Formula

$$d_0' = \begin{cases} d_{max} & (d_0 > d_{max\,2} + thr_D) \text{ and } (d_{max} \le d_{max\,2} + thr_C) \\ d_{max\,2} & (d_0 > d_{max\,2} + thr_D) \text{ and } (d_{max} > d_{max\,2} + thr_C) \\ d_{min} & (d_0 < d_{min\,2} - thr_D) \text{ and } (d_{min} \ge d_{min\,2} - thr_C) \\ d_{min\,2} & (d_0 < d_{min\,2} - thr_D) \text{ and } (d_{min} < d_{min\,2} - thr_C) \\ d_0 & \text{otherwise} \end{cases}$$

camss-640

NOTE:

- dmax and dmax2 are the maximum and second maximum of the surrounding pixels, respectively.
- dmin and dmin2 are the minimum and second minimum, respectively.

Figure 137. ISS ISP IPIPE Adaptive OTF DPC Method 2 Method Center Pixel Replaced With dmax

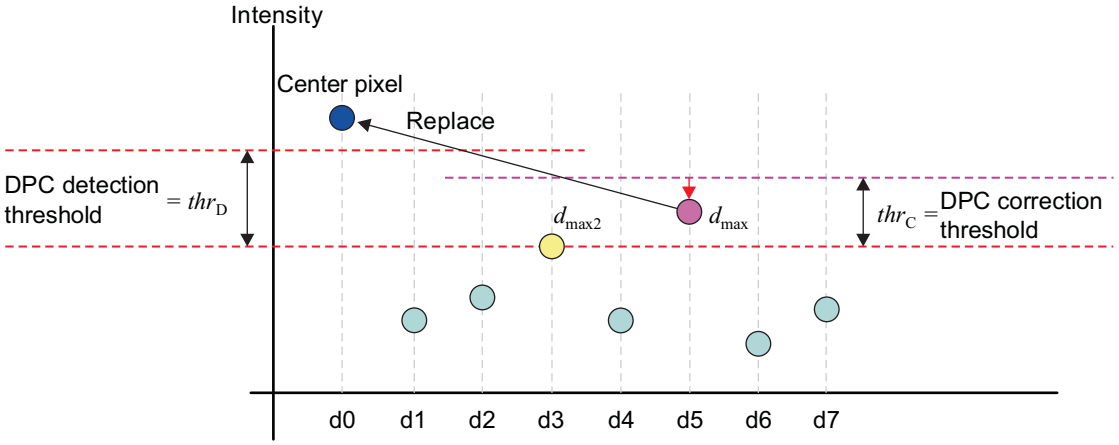
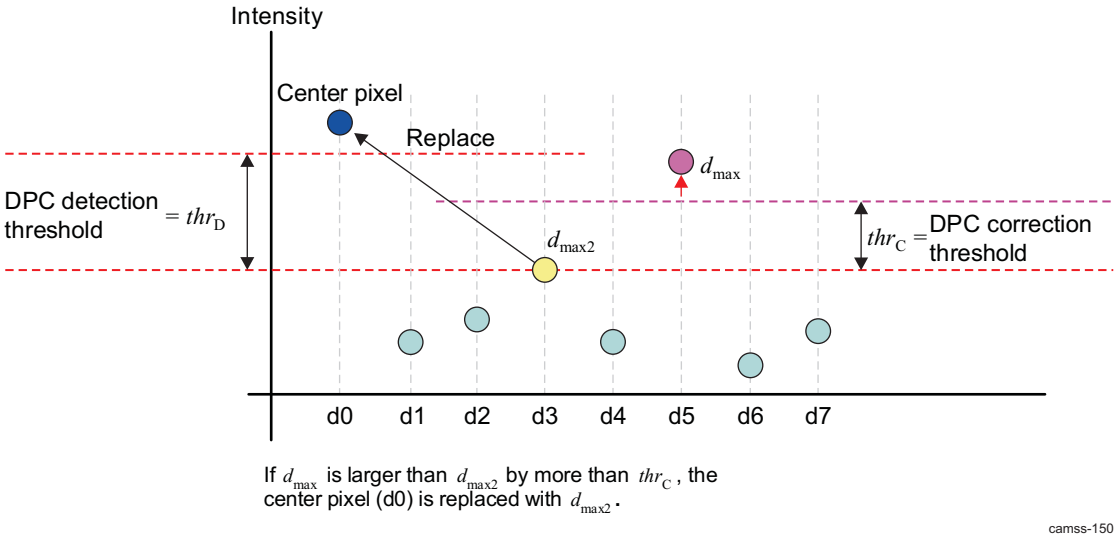


Figure 138. ISS ISP IPIPE Adaptive OTF DPC Method 2 Method Center Pixel Replaced With dmax2



3.3.3.4.2.3 ISS ISP IPIPE Adaptive OTF DPC Method 3

The adaptive OTF DPC method 3 algorithm is the same as adaptive OTF DPC method 2, but the threshold values (thr_D and thr_C) are calculated adaptively from the activity of the surrounding pixels.

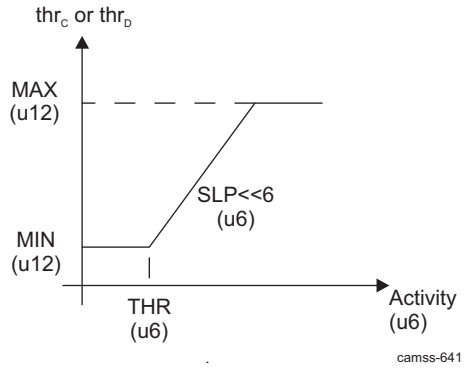
Thresholds are configured in this method:

- Detection threshold, thr_D:
  - THR = IPIPE_DPC_OTF_3_D_THR[11:6] VAL
  - SPL = IPIPE_DPC_OTF_3_D_SPL[5:0] VAL
  - MIN = IPIPE_DPC_OTF_3_D_MIN[11:0] VAL
  - MAX = IPIPE_DPC_OTF_3_D_MAX[11:0] VAL
- Correction threshold, thr_C:
  - THR = IPIPE_DPC_OTF_3_C_THR[11:6] VAL
  - SLP = IPIPE_DPC_OTF_3_C_SLP[5:0] VAL
  - MIN = IPIPE_DPC_OTF_3_C_MIN[11:0] VAL

– MAX = IPIPE_DPC_OTF_3_C_MAX[11:0] VAL

The threshold value depends on the set threshold minimum and maximum values, slope, and threshold. The threshold is calculated depending on the activity of the surrounding pixels. Figure 139 is a graphical representation of the activity definition. The activity value is specified by the following value.

Figure 139. ISS ISP IPIPE DPC Activity Pixel Definition



MAX, MIN, THR, and SLP are specified statically by registers and independently for thr$_C$ and thr$_D$. All four colors use the same set of values. THR is specified by the upper 6 bits of a register (IPIPE_DPC_OTF_3_D_THR or IPIPE_DPC_OTF_3_C_THR).

The suffix, current, top, left, and top_left indicate the position where the calculation takes place. For example, if the activity value at a coordinate of (x, y) is being calculated, (max3 – min3) left_top calculation is carried out at (x–1, y–1). The shift value can be set from the IPIPE_DPC_OTF_3_SHF[1:0] SHF bit field.

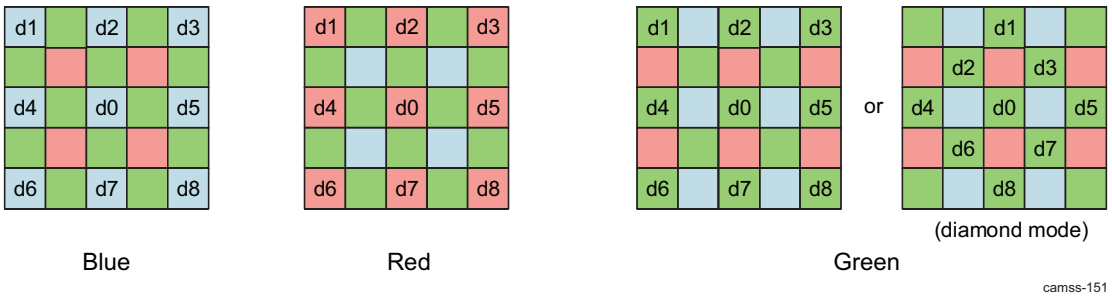
3.3.3.5 ISS ISP IPIPE 2D Noise Filter

The 2D noise filter reduces noise in RAW input data. In the device there are two 2D noise filters. Filter 1 and filter 2 can be enabled from the IPIPE_D2F_1ST_EN[0] EN and IPIPE_D2F_2ND_EN[0] EN bits, respectively. Data does not pass through the second 2D notice filter if is to be sent to the buffer logic module in RAW format. Moreover, if data is to be further processed in the IPIPE, it passes through the second 2D noise filter, which can be enabled and set accordingly if more filtering is needed.

NOTE: In this section only the first 2D noise filter is explained, because the second 2D noise filter is the same. The registers for both filters are also the same.

Figure 140 shows the data points used for this filtering. The pixel being filtered is d0; d0~d8 are same color pixels. For green, two numbering methods are available, and can be selected in the IPIPE_D2F_1ST_TYP[7] TYP bit.

Figure 140. ISS ISP IPIPE Noise Filter Array



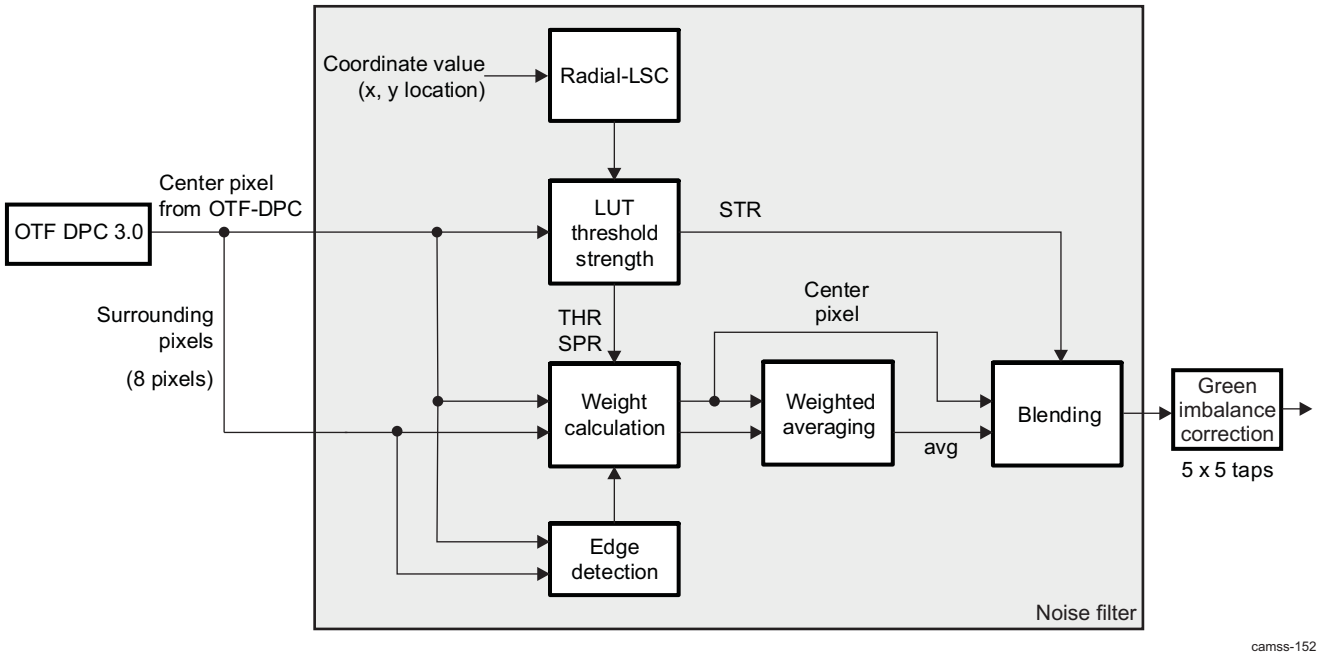
The basic concept of the 2D noise filter is area averaging by excluding far-apart-value neighbors. This algorithm uses the following tables:

- Threshold table (10 bits, 8 entries: IPIPE_D2F_1ST_THR_0x, where x = 0 to 7): Used in the filtering algorithm
- Intensity table (5 bits, 8 entries: IPIPE_D2F_1ST_STR_0x, where x = 0 to 7): Stores averaging weight

Threshold and intensity are read from the LUT and interpolated to make the intermediate value. The spread factor (SPR) is set through the IPIPE_D2F_1ST_TYP[4:0] SPR bit field.

Figure 141 is the block diagram of the noise filter module.

Figure 141. ISS ISP IPIPE Noise Filter Module Block Diagram



NOTE: Figure 141 is a generic visual representation of the 2D noise filter block. The actual 2D noise filter 1 outputs data to a line of memory block, while the second 2D noise filter 2 outputs data to the green imbalance correction block.

The threshold value is multiplied by radial-LSC gain to compensate for the noise enhancement caused by the LSC module in ISIF.The SPR value is a single value (IPIPE_D2F_1ST_TYP[4:0] SPR).

The LSC gain is applied to threshold values when IPIPE_D2F_1ST_TYP[8] LSC = 1.

Edge detection finds edges and controls the weight accordingly. If this function is enabled, the horizontal or vertical edge is detected through the equations shown in Figure 142. The numbering in this equation is always of box mode. Here, the minimum and maximum are specified in the 2D noise filter by the IPIPE_D2F_1ST_EDG_MIN and IPIPE_D2F_1ST_EDG_MAX registers, respectively.

Figure 142. ISS ISP IPIPE Edge Detection Formula

$$hedge = \text{abs}\left(\frac{(d_1 + 2d_2 + d_3) - (d_6 + 2d_7 + d_8)}{8}\right)$$

$$vedge = \text{abs}\left(\frac{(d_1 + 2d_4 + d_6) - (d_3 + 2d_5 + d_8)}{8}\right)$$

$$hedge > max, vedge < min \Rightarrow \text{horizontal edge}$$

$$hedge < min, vedge > max \Rightarrow \text{vertical edge}$$

camss-644

The threshold value is multiplied by radial-LSC gain *g* to compensate for the noise enhancement caused by the LSC module in the ISIF. The gain from radial-LSC is calculated in the manner shown in Figure 143. In the formula, x and y are the coordinates of the pixel of interest. The total gain (GAN) and offset (OFT) values are specified independently for each color (R, Gr, Gb, and B) by IPIPE_LSC_GAN_x and IPIPE_LSC_OFT_x (where x = R, GR, GB, or B). The other parameters are specified by the following corresponding registers:

- IPIPE_LSC_VOFT, for vertical offset
- IPIPE_LSC_VA2
- IPIPE_LSC_VA1
- IPIPE_LSC_VS
- IPIPE_LSC_HOFT, for horizontal offset
- IPIPE_LSC_HA2
- IPIPE_LSC_HA1
- IPIPE_LSC_HS
- IPIPE_LSC_SHF
- IPIPE_LSC_MAX

Here again, for the second noise filter, THR0-7, STR0-7, and SPR are specified by the following:

- IPIPE_D2F_2ND_THR_00 to 07
- IPIPE_D2F_2ND_STR_00 to 07
- IPIPE_D2F_2ND_TYP[4:0] SPR

Under attention are the minimum and maximum, which are specified by IPIPE_D2F_2ND_EDG_MIN and IPIPE_D2F_2ND_EDG_MAX, respectively.

Figure 143. ISS ISP IPIPE Edge Detection Formula 2

$$gain = limit\left(\left((H+V)\times GAN >> SHF + OFT\right)_{S14} >> 1,\ MAX\right)$$

$$H = \left(\frac{h}{2}\times\left(\left(\frac{h}{2}\times HA2 >> HS2\right)_{S13} + HA1\right)_{S13} >> HS1\right)_{S13}$$

$$V = \left(\frac{v}{2}\times\left(\left(\frac{v}{2}\times VA2 >> VS2\right)_{S13} + VA1\right)_{S13} >> VS1\right)_{S13}$$

$$h = x + HOFT$$

$$v = y + VOFT$$

$$limit\left(x, max\right) = \begin{cases} 0 & x < 0 \\ x & 0 \le x \le max \\ max & max < x \end{cases}$$

$$\left(x\right)_{S13} = \begin{cases} -4096 & x < -4096 \\ x & -4096 \le x \le 4095 \\ 4095 & 4095 < x \end{cases}$$

$$\left(x\right)_{S14} = \begin{cases} -8192 & x < -8192 \\ x & -8192 \le x \le 8191 \\ 8191 & 8191 < x \end{cases}$$

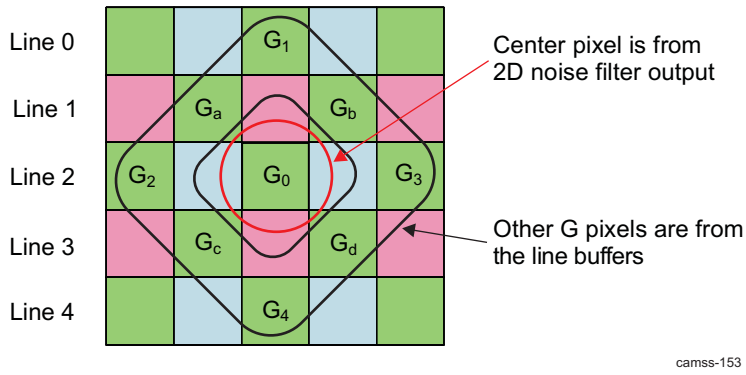
camss-645

3.3.3.6 ISS ISP IPIPE Green Imbalance Correction

GIC is implemented to reduce the effect of line-crawling (Gb-Gr difference), and it works only on Gb and Gr. IPIPE contains two algorithms: simple low filtering with constant gain, and adaptive gain control.

Here, G0 is the output of the 2D noise filter, and Ga through Gd are the surrounding pixels from line buffers, as shown in Figure 144. The GIC is enabled from the IPIPE_GIC_EN register.

Figure 144. ISS ISP IPIPE GIC Prefilter Pixels



In Figure 144, the simple averaging algorithm is applied in the following way. When the center site is green (Gb or Gr), the 2D prefilter mixes G0 and Gs2 = (Ga + Gb + Gc + Gd)/4 with a constant gain value set by the IPIPE_GIC_GAN register.

To select the simple averaging algorithm, set the IPIPE_GIC_TYP[0] TYP bit to 0, and set the gain value in the IPIPE_GIC_GAN register.

Another way to implement GIC Is to use the adaptive algorithm, which is applied in the following way. The output data *G*out is a weighted average of the center and the surrounding 4 pixels, as shown in Figure 144.

To select the adaptive algorithm, set the IPIPE_GIC_TYP[0] TYP bit to 1, and set the gain value in the IPIPE_GIC_GAN register.

The weight coefficients (w1, w2, w3, and w4) are determined by the function shown in Figure 147. The output mixing ratio wt is determined by the register.

The threshold values are calculated in the manner shown in Figure 145. (LSC g is the radial-LSC gain explained in Section 3.3.3.5, *ISS ISP IPIPE 2D Noise Filter*.) The following formula uses the IPIPE_GIC_THR and IPIPE_GIC_SLP registers to set the start values for the calculated threshold.

Figure 145. ISS ISP IPIPE GIC Threshold Calculation 1

$$thr_0 = clip\left(\left(GIC\_THR \times g_{LSC}\right) >> 5\right)$$

$$thr_1 = clip\left(\left(\left(GIC\_THR + GIC\_SLP\right) \times g_{LSC}\right) >> 5\right)$$

$$thr_2 = clip\left(\left(\left(GIC\_THR + 2 \times GIC\_SLP\right) \times g_{LSC}\right) >> 5\right)$$

$$thr_3 = clip\left(\left(\left(GIC\_THR + 3 \times GIC\_SLP\right) \times g_{LSC}\right) >> 5\right)$$

camss-648

The threshold values (thr1 to thr4) can be replaced with the threshold value of NF-2 by setting the IPIPE_GIC_TYP[1] SEL bit to 1, as shown in the formula in Figure 146. (The parameters nf_thr and nf_dif are identical to thr and dif in Section 3.3.3.5, *ISS ISP IPIPE 2D Noise Filter*.) GIC g is a U8Q5 gain specified by a register value of GIC_NFGAN.

Figure 146. ISS ISP IPIPE GIC Threshold Calculation 2

$$thr_0 = clip\left((nf\_thr \times g_{GIC}) >> 5\right)$$
$$thr_1 = clip\left(((nf\_thr + nf\_dif) \times g_{GIC}) >> 5\right)$$
$$thr_2 = clip\left(((nf\_thr + 2 \times nf\_dif) \times g_{GIC}) >> 5\right)$$
$$thr_3 = clip\left(((nf\_thr + 3 \times nf\_dif) \times g_{GIC}) >> 5\right)$$

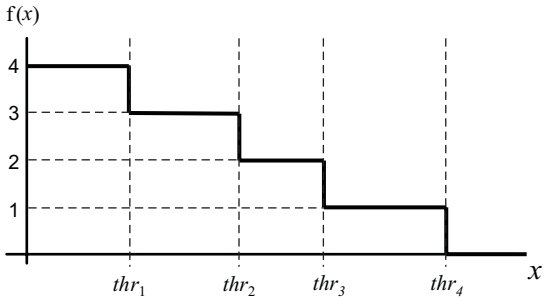
camss-649

Figure 147. ISS ISP IPIPE GIC Adaptive Prefilter Weight Function

$$w_i = \mathrm{f}\left(\left|G_0 - G_i\right|\right)$$

$$\mathrm{f}(x) = \begin{cases} 4 & x < thr_1 \\ 3 & x < thr_2 \\ 2 & x < thr_3 \\ 1 & x < thr_4 \\ 0 & thr_4 \le x \end{cases}$$

(The first case has the highest priority.)



camss-154

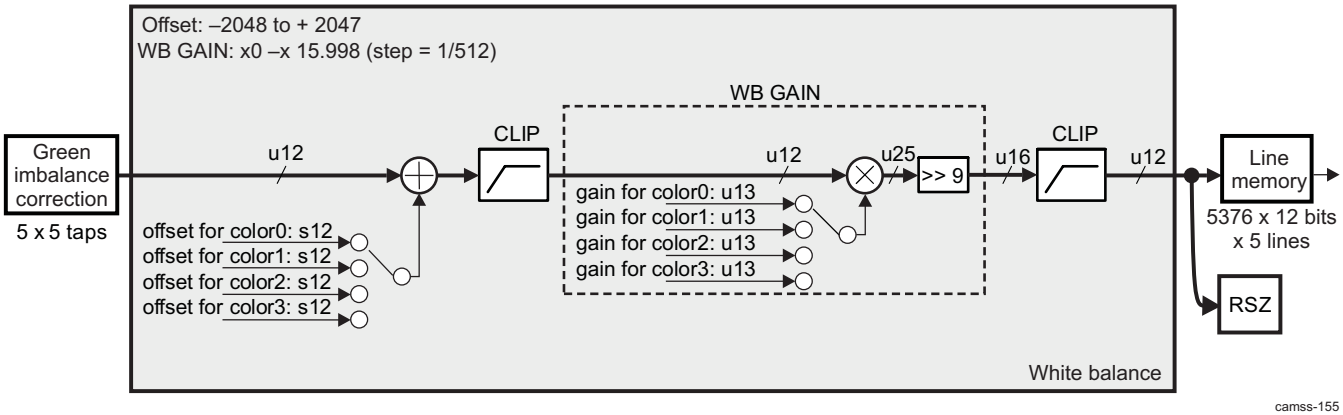
The threshold value is multiplied by radial-LSC gain if the IPIPE_GIC_TYP[2] LSC bit is set to 1 to compensate for the noise enhancement caused by the LSC module in the ISIF. For more information about radial-LSC gain, see Section 3.3.3.5, *ISS ISP IPIPE 2D Noise Filter*.

3.3.3.7 ISS ISP IPIPE White Balance

The white balance module executes white balance to each color component. White balance gain adjusts the ratio of each color existing in a CFA pattern. An offset can be applied before white balance correction (IPIPE_WB2_OFT_R, GR, Gb, or B registers).

Figure 148 is a block diagram of the white balance module. In the white balance gain adjuster, the RAW data is multiplied by a selected gain (IPIPE_WB2_WGN_R, Gr, Gb, or B registers) corresponding to the color. The white balance gain can be selected from four 13-bit values. Firmware can assign any combination of 4 pixels in horizontal and vertical directions. The precision of each gain is shown in the figure.

Figure 148. ISS ISP IPIPE White Balance

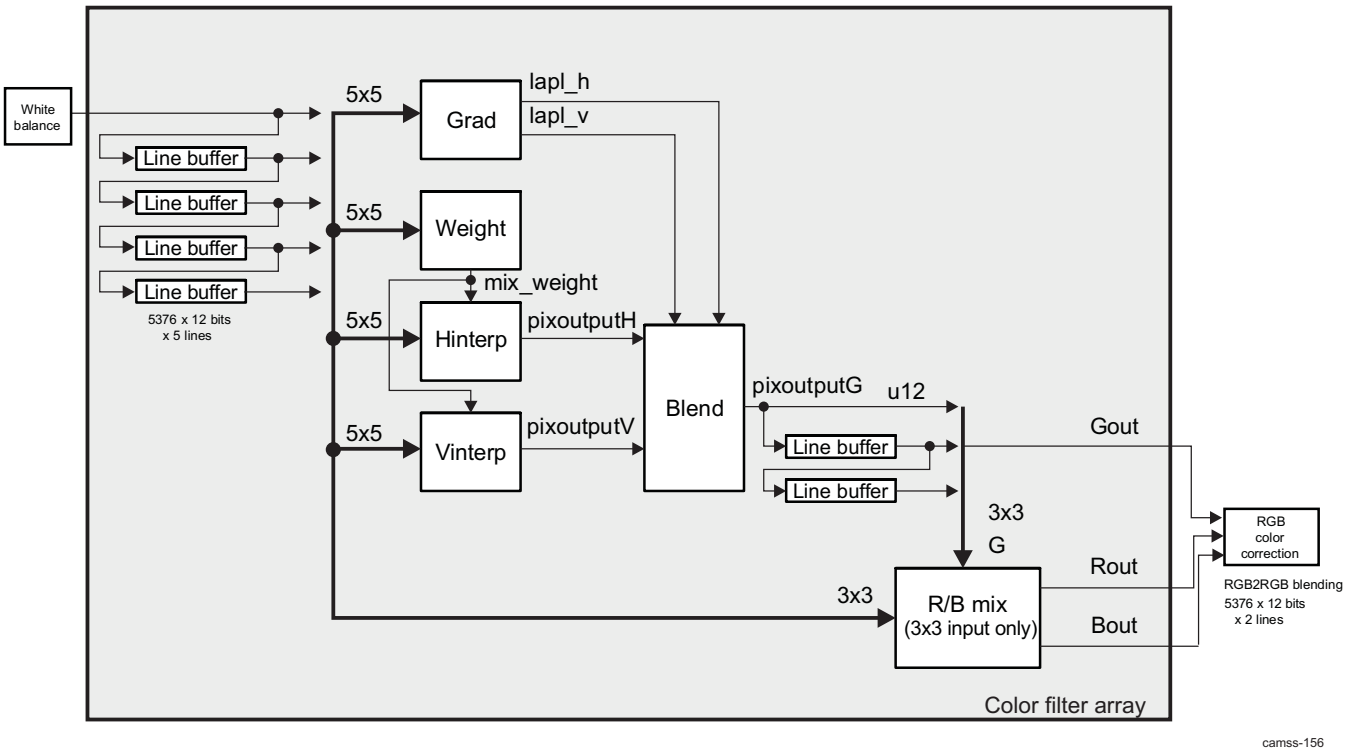


camss-155

3.3.3.8 ISS ISP IPIPE Color Filter Array Interpolation

The IPIPE color filter array (CFA) interpolation module generates RGB 4:4:4 data from the Bayer RGB pattern. Figure 149 is the block diagram of the CFA interpolation module. The CFA consists of the line delay block, edge-direction detector, and pixel interpolator. The edge-direction detector detects the most probable edge direction of an object to which the target pixel belongs, and the pixel interpolator calculates the missing color according to its edge direction within a 5 × 5 pixel window.

Figure 149. ISS ISP IPIPE CFA Interpolation



The CFA interpolation module also supports DAA, which reduces aliasing caused by undersampling.

CFA and DAA can be activated separately or together. The choice is made in the IPIPE_CFA_MODE[1:0] MODE bit field. The low value of the HP threshold can be controlled from the IPIPE_CFA_2DIR_HPF_THR register, and the HP slope can be set from the IPIPE_CFA_2DIR_HPF_SLP register. A mixed threshold can be set from the IPIPE_CFA_2DIR_MIX_THR register, and the slope of the HP can be set from the IPIPE_CFA_2DIR_MIX_SLP register. The directional threshold can be set from the IPIPE_CFA_2DIR_DIR_TRH register, and the slope from the IPIPE_CFA_2DIR_DIR_SLP register.

CAUTION

The use of DAA only is not recommended.

If using only DAA, a series of settings can be applied. Hue fraction is set from the IPIPE_CFA_MONO_HUE_FRA register, the edge of the threshold from the IPIPE_CFA_MONO_EDG_THR register, and the threshold minimum and slope from the IPIPE_CFA_MONO_THR_MIN and IPIPE_CFA_MONO_THR_SLP registers, respectively.

3.3.3.9 ISS ISP IPIPE RGB2RGB Blending Module

The RGB2RGB blending module transforms the RGB data generated by the CFA interpolation module using a 3 × 3 square matrix transformation in combination with an added offset. The RGB-to-RGB blending is calculated using the formula shown in Figure 150. Each gain range is from –8 to +7.996 with step 1/256 = 0.004, and is set in the IPIPE_RGB1_MUL_RR to IPIPE_RGB1_MUL_BB registers. The offset range for each component is from –4096 to 4095, and is set in the IPIPE_RGB1_OFT_OR to IPIPE_RGB1_OFT_OB registers.

Figure 150. ISS ISP IPIPE RGB2RGB Conversion Formula

$$\begin{pmatrix} R\_out \\ G\_out \\ B\_out \end{pmatrix} = \begin{pmatrix} gain\_RR & gain\_GR & gain\_BR \\ gain\_RG & gain\_GG & gain\_BG \\ gain\_RB & gain\_GB & gain\_BB \end{pmatrix} \begin{pmatrix} R\_in \\ G\_in \\ B\_in \end{pmatrix} + \begin{pmatrix} offset\_r \\ offset\_G \\ offset\_B \end{pmatrix}$$

camss-650

3.3.3.10 ISS ISP IPIPE Gamma Correction Module

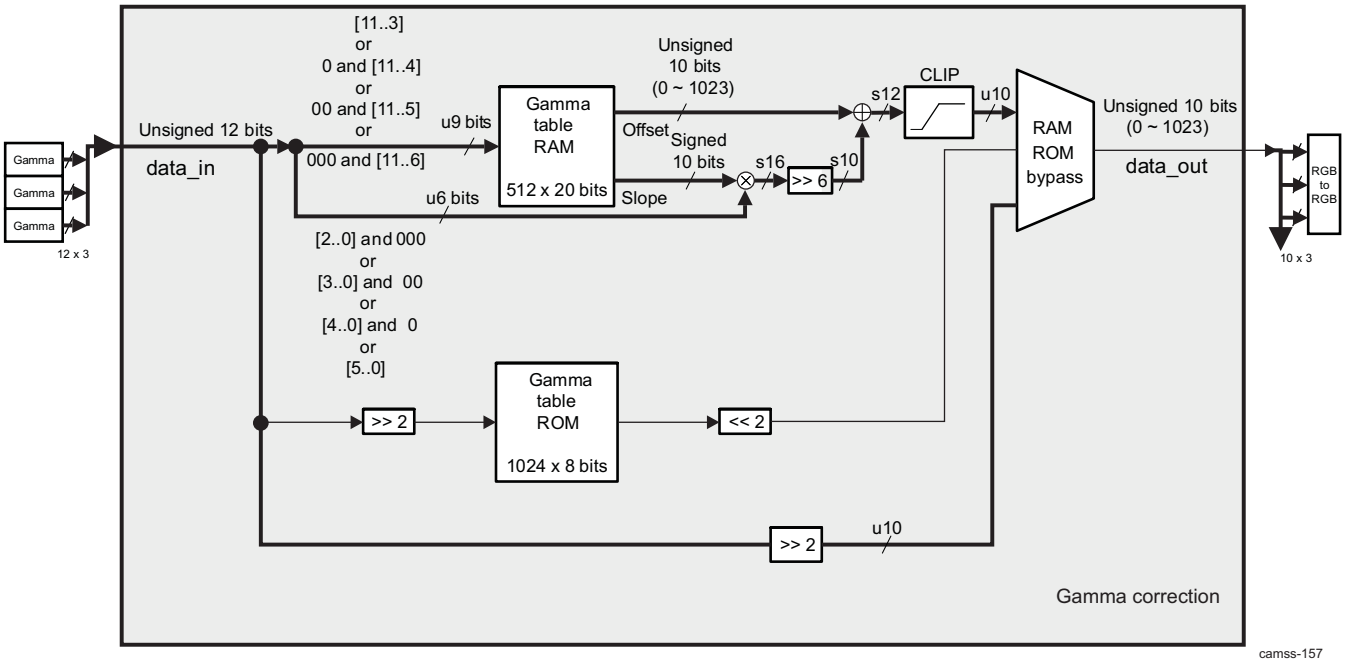
NOTE: For the memory access locations of the gamma correction module, see Section 3.3.8.

The gamma correction module performs gamma correction independently for each color in the RGB color space by using a piece-wise linear interpolation. ROM tables and RAM tables are selectable through the IPIPE_GMM_CFG[4] TBL bit. Each ROM table and RAM table has 512-entries, and each entry accommodates a 10-bit offset and 10-bit slope. The range of slope value is from –512 to +511. The ROM table has 1024 entries and an output 8-bit value.

Figure 151 is a block diagram of the gamma correction module. It is composed of two tables and one selector. When the BYPASS bit is asserted, the input data is divided by 16 (the IPIPE_GMM_CFG[0] BYPR ,IPIPE_GMM_CFG[1] BYPG, and IPIPE_GMM_CFG[2] BYPB bits).

Figure 152 shows an example of the gamma curve. Figure 153 shows offset and slope packing.

Figure 151. ISS ISP IPIPE Gamma Correction Module Block Diagram



camss-157

Figure 152. ISS ISP IPIPE Gamma Curve Example

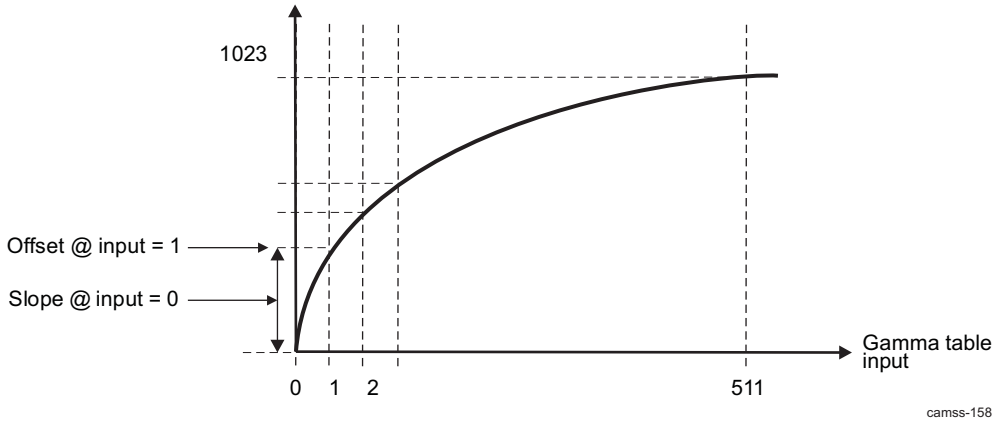
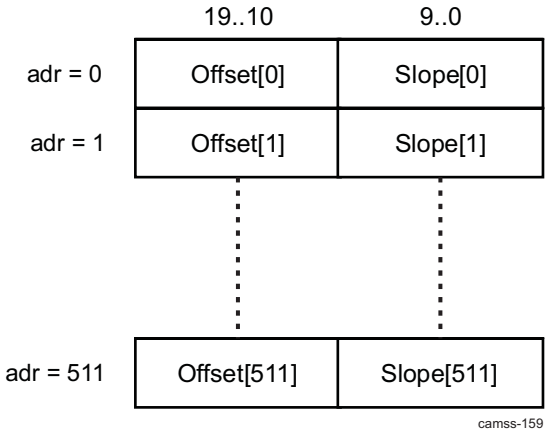


Figure 153. ISS ISP IPIPE Gamma Table Offset/Slope Packing



3.3.3.11 ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix

The second RGB2RGB blending module transforms the RGB data after gamma correction using the 3 × 3 square matrix transformation in combination with an added offset. The RGB-to-RGB blending is calculated using the formula shown in Figure 154. Each gain range is from –4 to +3.996 with step 1/256 = 0.004 (s3.8), and is set in the IPIPE_RGB2_MUL_RR to IPIPE_RGB2_MUL_BB registers. The offset is –1024 to 1023 (s11), and is set in the IPIPE_RGB2_OFT_OR to IPIPE_RGB2_OFT_OB registers.

Figure 154. ISS ISP IPIPE RGB2RGB 2nd Conversion Formula

$$\begin{pmatrix} R\_out \\ G\_out \\ B\_out \end{pmatrix} = \begin{pmatrix} gain\_RR & gain\_GR & gain\_BR \\ gain\_RG & gain\_GG & gain\_BG \\ gain\_RB & gain\_GB & gain\_BB \end{pmatrix} \begin{pmatrix} R\_in \\ G\_in \\ B\_in \end{pmatrix} + \begin{pmatrix} offset\_r \\ offset\_G \\ offset\_B \end{pmatrix}$$

camss-650

3.3.3.12 ISP IPIPE 3D-LUT Color Conversion

NOTE: For 3D-LUT memory access locations, see Section 3.3.8.

3D LUT converts RGB pixel data into another RGB data using a 9 × 9 × 9 entry 3D LUT, which is enabled when the IPIPE_3DLUT_EN[0] EN bit is set to 1. Input and output are both in 10-bit format. If the input is at the lattice point, the output is the value stored in the table shown in Figure 155.

Figure 155. ISS ISP IPIPE 3D-LUT Color Conversion Formula

$$R_{out} = LUT_R\left[P\left(R_{in}/128, G_{in}/128, B_{in}/128\right)\right] \quad R_{in} \bmod 128 = 0$$
$$G_{out} = LUT_G\left[P\left(R_{in}/128, G_{in}/128, B_{in}/128\right)\right], \text{ when } G_{in} \bmod 128 = 0$$
$$B_{out} = LUT_B\left[P\left(R_{in}/128, G_{in}/128, B_{in}/128\right)\right] \quad B_{in} \bmod 128 = 0$$

camss-651

The entry number is determined by the formula shown in Figure 156.

Figure 156. ISS ISP IPIPE 3D-LUT Entry Number Formula

$$P\left(R,G,B\right) = R + 9G + 81B$$

camss-652

If the input is not on the lattice point, the output is determined by the tetrahedral interpolation formula shown in Figure 157.

Figure 157. ISS ISP IPIPE 3D-LUT Tetrahedral Interpolation Formula

$$R\_dif = R \bmod 128, \quad R\_i = \left(R - R\_dif\right)/128$$
$$G\_dif = G \bmod 128, \quad G\_i = \left(G - G\_dif\right)/128$$
$$B\_dif = B \bmod 128, \quad B\_i = \left(B - B\_dif\right)/128$$
$$R_0 = LUT_R[P0] \quad G_0 = LUT_G[P0] \quad B_0 = LUT_B[P0]$$
$$R_1 = LUT_R[P1], \quad G_1 = LUT_G[P1], \quad B_1 = LUT_B[P1],$$
$$R_2 = LUT_R[P2] \quad G_2 = LUT_G[P2] \quad B_2 = LUT_B[P2]$$
$$R_3 = LUT_R[P3] \quad G_3 = LUT_G[P3] \quad B_3 = LUT_B[P3]$$
$$R_{out} = R_0 + \left(R_1 - R_0\right)\cdot x + \left(R_2 - R_1\right)\cdot y + \left(R_3 - R_2\right)\cdot z$$
$$G_{out} = G_0 + \left(G_1 - G_0\right)\cdot x + \left(G_2 - G_1\right)\cdot y + \left(G_3 - G_2\right)\cdot z$$
$$B_{out} = B_0 + \left(B_1 - B_0\right)\cdot x + \left(B_2 - B_1\right)\cdot y + \left(B_3 - B_2\right)\cdot z$$

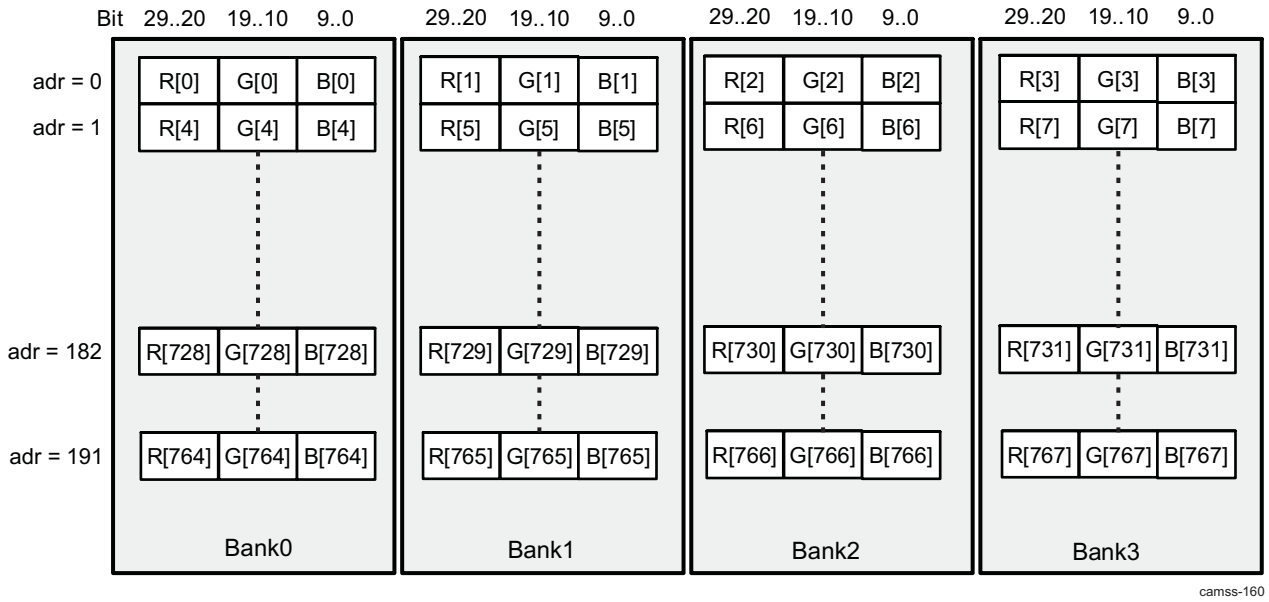
camss-653

P0-P3 and x, y, and z are determined by the following table.

Input/Output Condition					
Input Condition	**P0**	**P1**	**P2**	**P3**	**(X, Y, Z)**
$R_{dif} \geq G_{dif} \geq B_{dif}$	(R_i, G_i, B_i)	(R_i +1, G_i, B_i)	(R_i +1, G_i +1, B_i)	(R_i +1, G_i +1, B_i +1)	(R_{dif}, G_{dif}, B_{dif})
$R_{dif} \geq B_{dif} \geq G_{dif}$	(R_i, G_i, B_i)	(R_i +1, G_i, B_i)	(R_i +1, G_i, B_i +1)	(R_i +1, G_i +1, B_i +1)	(R_{dif}, B_{dif}, G_{dif})
$G_{dif} \geq R_{dif} \geq B_{dif}$	(R_i, G_i, B_i)	(R_i, G_i +1, B_i)	(R_i +1, G_i +1, B_i)	(R_i +1, G_i +1, B_i +1)	(G_{dif}, R_{dif}, B_{dif})
$G_{dif} \geq B_{dif} \geq R_{dif}$	(R_i, G_i, B_i)	(R_i, G_i +1, B_i)	(R_i, G_i +1, B_i +1)	(R_i +1, G_i +1, B_i +1)	G_{dif}, B_{dif}, R_{dif})
$B_{dif} \geq R_{dif} \geq G_{dif}$	(R_i, G_i, B_i)	(R_i, G_i, B_i +1)	(R_i +1, G_i, B_i +1)	R_i +1, G_i +1, B_i +1)	(B_{dif}, R_{dif}, G_{dif})
$B_{dif} \geq G_{dif} \geq R_{dif}$	(R_i, G_i, B_i)	(R_i, G_i, B_i +1)	(R_i, G_i +1, B_i +1)	(R_i +1, G_i +1, B_i +1)	(B_{dif}, G_{dif}, R_{dif})

P0, P1, P2, and P3 are always in different banks. The data in LUT are packed in a manner shown in Figure 158.

Figure 158. ISS ISP IPIPE 3D-LUT Data Packing



The LUT with its 9 × 9 × 9 entries are stored in four banks (total 192 × 30 bits × 3) in a local memory bank table linked with the 3D-LUT converter. The entries at and above 729 are not used by the 3D-LUT function.

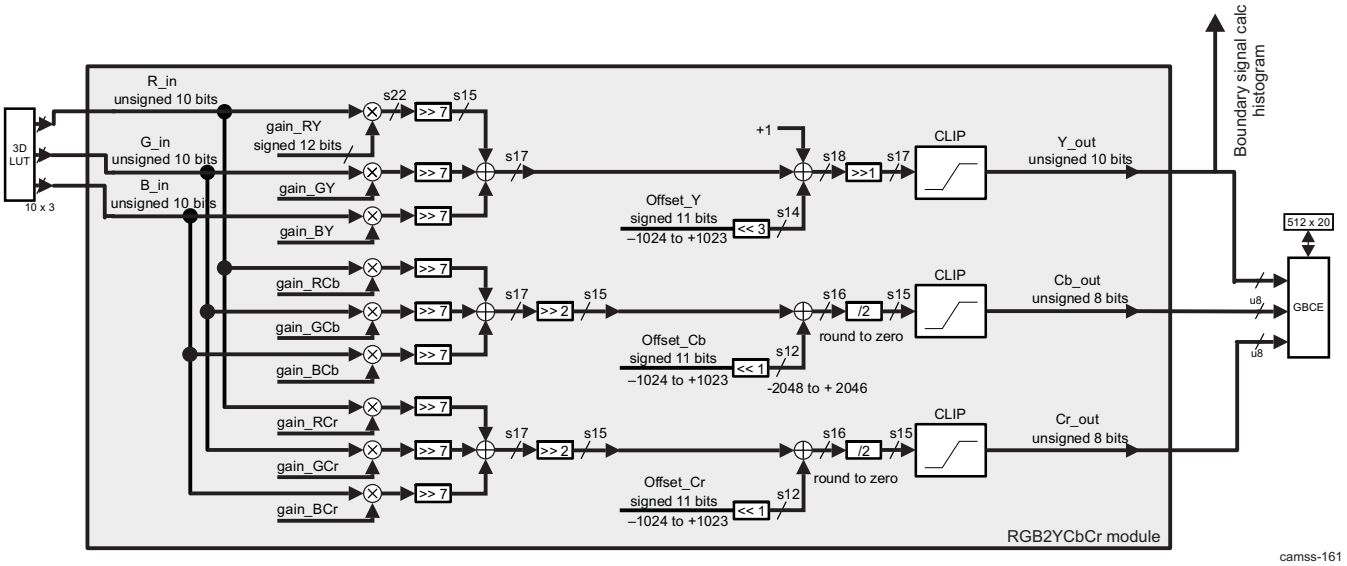
3.3.3.13 ISS ISP IPIPE RGB2YCbCr Conversion Matrix

This module transforms the RGB data to YCbCr data format using a 3 × 3 matrix transformation in combination with an added offset. While transferring, the brightness control and contrast control can be adjusted using the IPIPE_YUV_ADJ[8:15] BRT and IPIPE_YUV_ADJ[8:15] CRT bit fields, respectively. Then, the transform is calculated using the formula shown in Figure 159. Each gain range is from –8 to +7.996 with step 1/256 = 0.004, configured in the IPIPE_YUV_MUL_RY to IPIPE_YUV_MUL_BCr registers. The offset is –1024 to 1023 for Y, Cb, and Cr, configured in the IPIPE_YUV_OFT_Y to IPIPE_YUV_OFT_Cr registers. Figure 160 is the block diagram of the RGB to RGB blending module. The output is calculated by the equation.

Figure 159. ISS ISP IPIPE RGB2RGB 2nd Conversion Formula

$$\begin{pmatrix} Y\_out \\ Cb\_out \\ Cr\_out \end{pmatrix} = \begin{pmatrix} gain\_RY & gain\_GY & gain\_BY \\ gain\_RCb & gain\_GCb & gain\_BCb \\ gain\_RCr & gain\_GCr & gain\_BCr \end{pmatrix} = \begin{pmatrix} R\_in \\ G\_in \\ B\_in \end{pmatrix} + \begin{pmatrix} offset\_Y \\ offset\_Cb \\ offset\_Cr \end{pmatrix}$$

Figure 160. ISS ISP IPIPE RGB2YCbCr Module Block Diagram



3.3.3.14 ISS ISP IPIPE Global Brightness and Contrast Enhancement

NOTE: For GBCE memory access locations, see Section 3.3.8.

GBCE applies adaptive tone mapping on the Y channel for best image quality and can be enabled through the IPIPE_GBCE_EN[0] bit. When enabled, the Y channel is converted by a 1024 entry 8-bit output LUT. The LUT is calculated by software and written to the IPIPE memory table before capturing the target frame by setting the IPIPE_GBCE_TYP[0] bit to 0 for Y. See Figure 161.

Figure 161. ISS ISP IPIPE GBCE Mode 1 Formula

$$Y_{output} = LUT\left[Y_{input}\right]$$
$$Cb_{output} = Cb_{input} \quad \text{(GBCE mode 1)}$$
$$Cr_{output} = Cr_{input}$$

camss-656

GBCE also has another mode when the IPIPE_GBCE_TYP[0] bit is set to 1, where Y and C signals are multiplied by a gain value. The gain value is a function of Y input and is in unsigned 10-bit format. Figure 162 shows the second mode in greater detail.

Figure 162. ISS ISP IPIPE GBCE Mode 2 Formula

$$Y_{\text{output}} = clip_{U8}\left[\left(gain \times Y_{\text{input}}\right) >> 10\right]$$
$$Cb_{output} = clip_{U8}\left[\frac{gain \times \left(Cb_{input} - 128\right)}{256} + 128\right] \quad \text{(GBCE mode 2)}$$
$$Cr_{output} = clip_{U8}\left[\frac{gain \times \left(Cr_{input} - 128\right)}{256} + 128\right]$$
$$gain = LUT[Y_{\text{input}}]$$

camss-657

If GBCE is off, Y value is converted to u8, and Cb and Cr are not changed, as shown in Figure 163.

Figure 163. ISS ISP IPIPE GBCE Mode Off Formula

$$Y_{output} = \left(Y_{input} + 2\right) >> 2$$
$$Cb_{output} = Cb_{input} \quad \text{(GBCE off)}$$
$$Cr_{output} = Cr_{input}$$

camss-658

Then, after setting the mode, data is packed in 20 bits in the manner shown in Figure 164 and Figure 165 for mode 1 and mode 2, respectively.

Figure 164. ISS ISP IPIPE GBCE LUT Packing in Mode 1

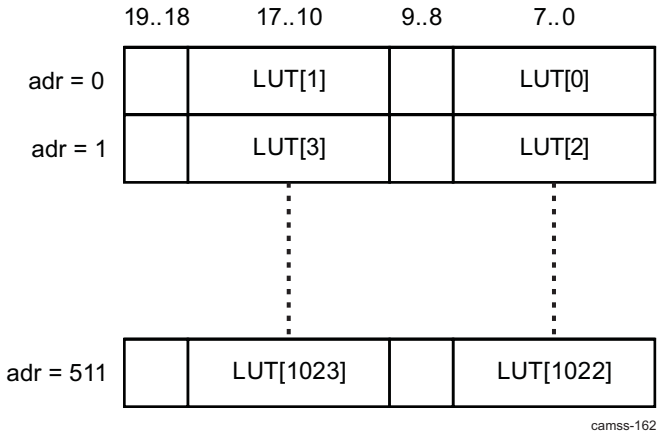
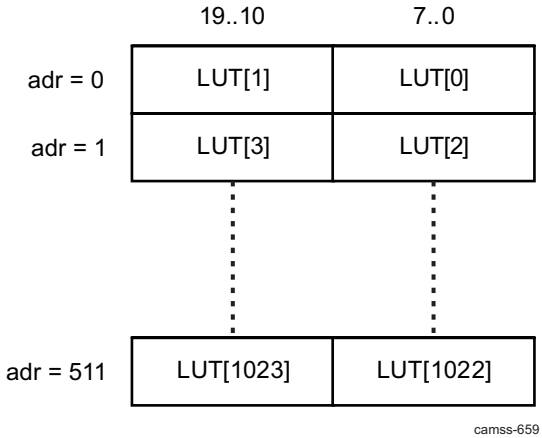


Figure 165. ISS ISP IPIPE GBCE LUT Packing in Mode 2



3.3.3.15 ISS ISP IPIPE 4:2:2 Conversion Module

The 4:2:2 conversion module converts the image data to YCbCr-4:2:2 format by taking the average of every two Cb and Cr components. Y and Cb/Cr sampling point of spatial cosited or spatial centering are selectable using the IPIPE_YUV_PHS[0] POS register. Horizontal 3 taps and 4- or 2-tap filters are used for spatial cosited and spatial centering, respectively. The module is enabled from the IPIPE_YUV_PHS[1] PLF bit.

Figure 166. ISS ISP IPIPE Chroma Subsampling Position

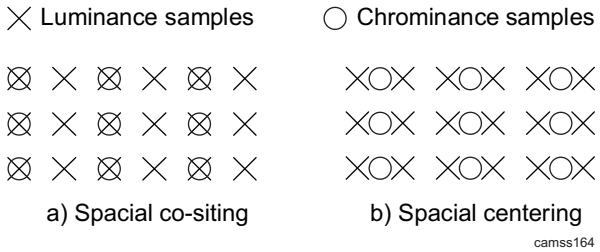
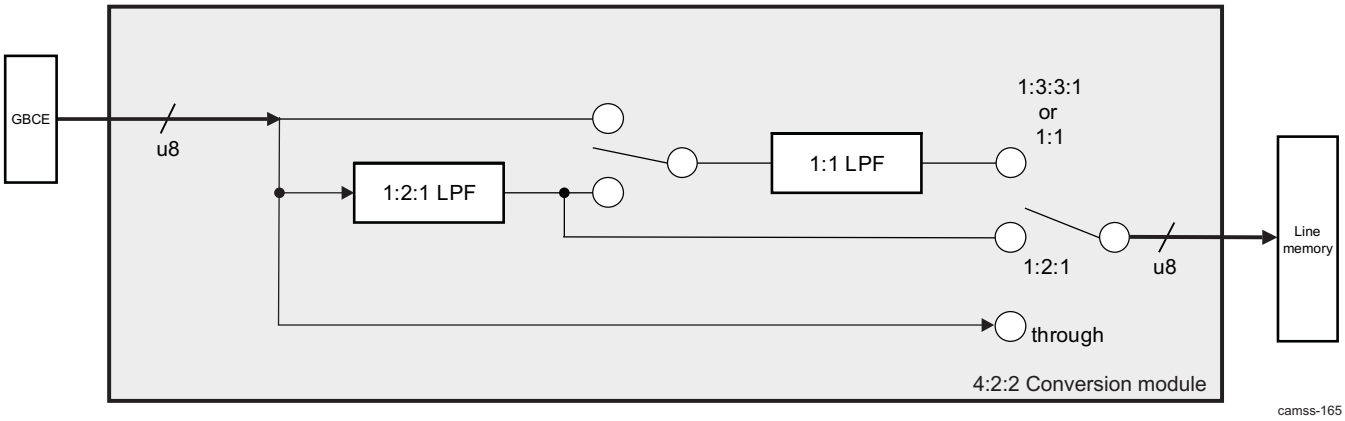


Figure 167 is a block diagram of the 4:2:2 conversion module.

Figure 167. ISS ISP IPIPE 4:2:2 Conversion Module Block Diagram



3.3.3.16 ISS ISP IPIPE 2D Edge Enhancer

NOTE: For the location of 2D edge-enhancer memory accesses, see Section 3.3.8.

The edge-enhancer module operates on the luminance (Y data) component of images to improve the image quality and can be enabled from the IPIPE_YEE_EN[0] EN bit. Edges in input images are detected by a 2D high-pass filter, and its sharpness is increased by the value from a non-linear table. Figure 168 shows a block diagram of the luminance non-linear edge-enhancer. Entry for the non-linear table is 10-bit and the output is in signed 9-bit.

In edge-enhancer mode when IPIPE_YEE_TYP[0] SEL is set to 0, linear filter with programmable coefficient is applied to the Y input. Here, M is a 5 × 5 matrix with programmable coefficients (IPIPE_YEE_MUL_xx, where x = 00, 01, 02, 10, 11, 12, 20, 21, 22). A down shift of high pass filter is applied to the edge enhancer from the IPIPE_YEE_SHF[3:0] SHF bit field (shgHPF in the formula).

Then, the HPF value is shrunk by a threshold value (u6) specified by the IPIPE_YEE_THR register (thresholdHPF in the formula), and clipped to signed 10 bits to get the index for the LUT.

Figure 168. ISS ISP IPIPE 2D Edge-Enhancer Indexing

$$index = clip\left(shrink\left(HPF, threshold_{HPF}\right), -512, 511\right)$$

$$shrink\left(x, threshold\right) = \begin{cases} x + threshold & x < -threshold \\ 0 & -threshold \le x \le threshold \\ x - threshold & threshold < x \end{cases}$$

$$clip\left(x, limit_{LOW}, limit_{HIGH}\right) = \begin{cases} -limit_{LOW} & x < -limit_{LOW} \\ x & -limit_{LOW} \le x \le limit_{HIGH} \\ limit_{HIGH} & limit_{HIGH} < x \end{cases}$$

camss-661

Moreover, the edge-enhancement intensity is looked up from the LUT through the formula shown in Figure 169 and in Table 176.

Figure 169. ISS ISP IPIPE 2D Edge Intensity LUT Formula

$$E_{int} = LUT\left[index\right]$$

camss-662

Table 176. ISS ISP IPIPE Edge-Enhancer LUT Mapping

Address (32-bit Word Address)	Bit Position	LUT Index
0x00000h	8 .. 0	1
	17 ..9	2
0x00001h	8 .. 0	2
	17 .. 9	3
0x00002h	8 .. 0	4
	17 .. 9	5
0x00003h	8 .. 0	6
	17 .. 9	7
. . .	. . .	. . .
0x000FFh	8 .. 0	510
	17 .. 9	511
0x00100h	8 .. 0	–512
	17 .. 0	–511
0x00101h	8 .. 0	–510
	17 .. 9	–509
. . .	. . .	. . .
0x001FD	8 .. 0	–6
	17 .. 9	–5
0x001FE	8 .. 0	-4
	17 .. 9	-3
0x001FFh	8 .. 0	-2
	17 .. 9	-1

Figure 170 shows the LUT packing, and Figure 171 shows the 2D edge-enhancer block diagram.

Figure 170. ISS ISP IPIPE 2D Edge-Enhancer LUT Packing

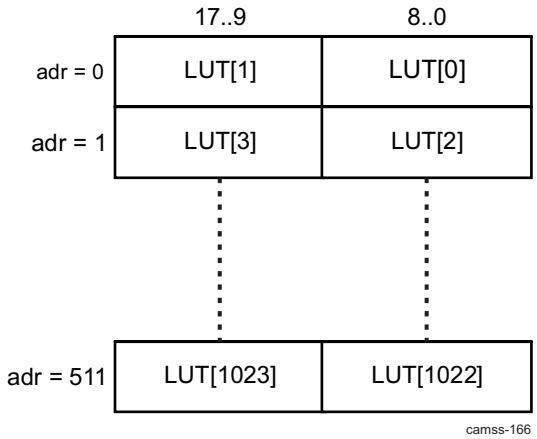
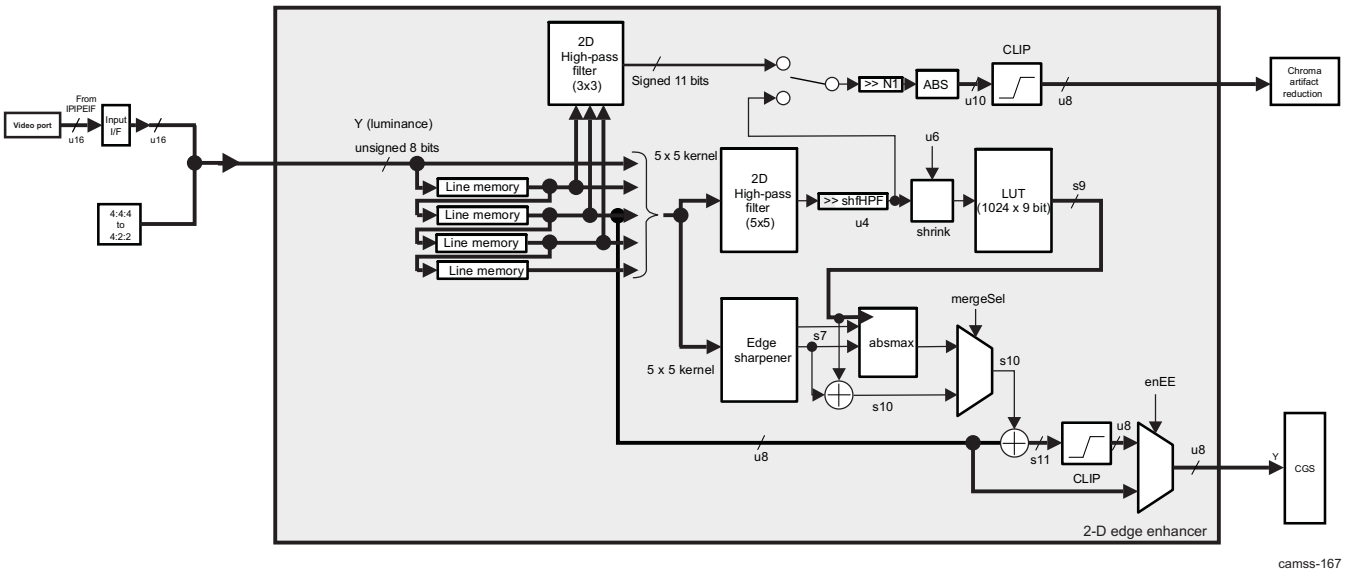


Figure 171. ISS ISP IPIPE 2D Edge-Enhancer Block Diagram



In edge sharpener mode, enabled when IPIPE_YEE_TYP[0] SEL = 1, edge clarity is enhanced without producing a Halo artifact. In this module, edge intensity is derived by the 2D linear filter with fixed coefficients shown in Figure 172.

Figure 172. ISS ISP IPIPE Edge Sharpener Details

$$S_{i,j} = \begin{pmatrix} 0 & -1 & -2 & -1 & 0 \\ -1 & 0 & 2 & 0 & -1 \\ -2 & 2 & 8 & 2 & -2 \\ -1 & 0 & 2 & 0 & -1 \\ 0 & -1 & -2 & -1 & 0 \end{pmatrix}$$

$$sharpness(h,v) = clip\left(shrink\left(g \sum_{j=-2}^{2} \sum_{i=-2}^{2} S_{i,j} Y(h+i, v+j), -threshold_{LOW}, threshold_{LOW} \right) >> 6, threshold_{HIGH} \right)$$

camss-663

The gain (g) and threshold values for the shrink/clip function ($threshold_{LOW}$, $threshold_{HIGH}$) are determined by the IPIPE_YEE_E_GAN, IPIPE_YEE_THR_1 and IPIPE_YEE_THR_2 registers. The gain g is in U12Q6, $threshold_{HIGH}$ is in U6, and $threshold_{LOW}$ is in U12Q6.

This edge intensity is then clipped by a threshold value in the formula shown in Figure 173.

Figure 173. ISS ISP IPIPE 2D Edge-Intensity Clipping Formula

$$S_{int} = \begin{cases} clip(sharpness, grad) & \text{Halo reduction on} \\ sharpness & \text{Halo reduction off} \end{cases}$$

camss-664

The threshold value (grad) is a function of the activity around the target pixel, which is derived from gradient values. Gain and offset are specified by IPIPE_YEE_G_GAN and IPIPE_YEE_G_OFT

Capping with gradient value prevents overly enhancing edges, and suppresses halo artifacts around edges.

The output from edge enhancer and edge sharpener are merged with the function shown in Figure 174.

Figure 174. ISS ISP IPIPE 2D Edge Enhancer and Sharpener Merger Formula

$$E_{merge} = \begin{cases} E_{int} + S_{int} & mergedsel = 0 \\ abs\ max(E_{int}, S_{int}) & mergedsel = 1 \end{cases}$$

$$abs\ max(x, y) = \begin{cases} x & abs(y) \le abs(x) \\ y & otherwise \end{cases}$$

The E_{merge} value is added to the Y input value to make the final output.

For chroma suppression, another 2D high pass filter (HPF) is implemented. One of the four coefficient sets shown in Figure 175 is selectable.

Figure 175. ISS ISP IPIPE 2D Edge Chroma-Suppression Coefficient Sets

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad or \quad \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

camss-667

At the end of the edge enhancer process, brightness and contrast adjustment are applied to the Y signal. The formula shown in Figure 176 describes the process.

Figure 176. ISS ISP IPIPE 2D Edge-Brightness and Contrast Adjustments Formula

$$Y_{ctr\_brt} = clip8(clip8((Y_{EE} \times CTR) >> 4) + BRT)$$

camss-668

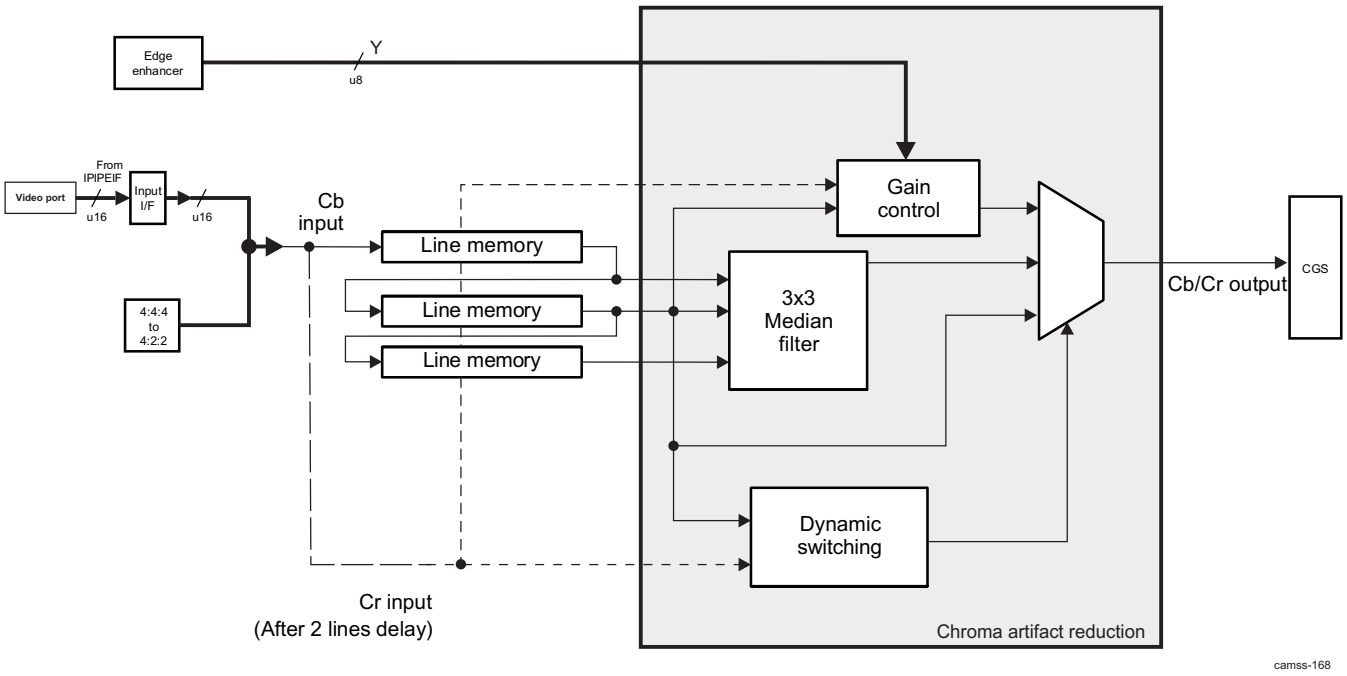
3.3.3.17 ISS ISP IPIPE Chroma Artifact Reduction

Chroma artifact reduction (CAR) reduces color artifacts at the edges. Chroma reduction is enabled through the IPIPE_CAR_EN[0] EN bit, but a resizer setting must be considered if Bayer input/output is set from the resizer source register (for more information, see IPIPE_CAR_EN and section, RESIZER).

CAR has two methods: 3 × 3 median filter and chroma gain control. Also, dynamic switching between these two methods is available to obtain the best performance of each method. The mode is configured with the IPIPE_CAR_TYP[0] TYP bit.

Figure 172 shows the Cb part of the CAR block diagram.

Figure 177. ISS ISP IPIPE CAR Block Diagram (Cb Part)



In gain control mode, the gain is calculated by the formula shown in Figure 178 (the gain, gain1, and gain2 values are in U9Q8 format.).

Figure 178. ISS ISP IPIPE CAR Gain Control Mode Formula

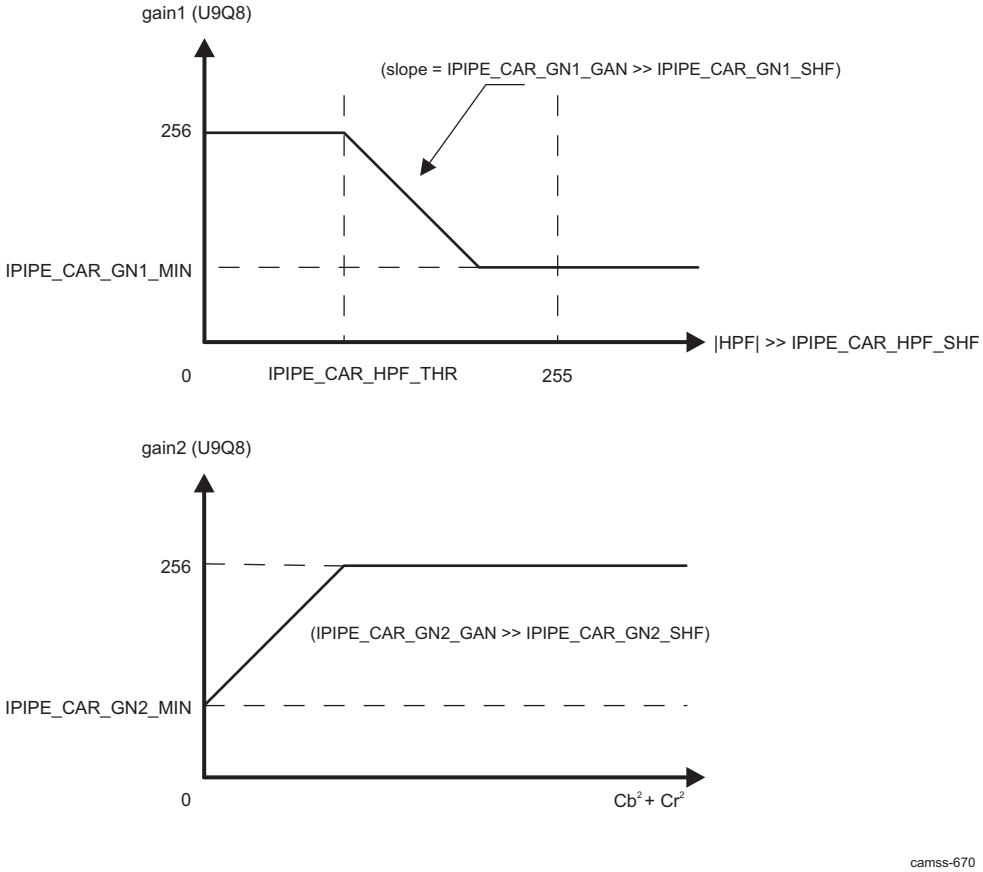
$$gain = \begin{cases} 1 & |HPF| < IPIPE\_CAR\_HPF\_THR \\ gain1 \times gain2 >> 8 & otherwise \end{cases}$$

camss-669

The type of HPF filter is determined by the IPIPE_CAR_HPF_TYP[2:0] TYP bit field. The gain values (gain1 and gain2) are calculated by the functions shown in Figure 179. The parameters are the settings of the following registers:

- IPIPE_CAR_HPF_THR
- IPIPE_CAR_HPF_SHF
- IPIPE_CAR_GN1_GAN
- IPIPE_CAR_GN1_SHF
- IPIPE_CAR_GN1_MIN
- IPIPE_CAR_GN2_GAN
- IPIPE_CAR_GN2_SHF
- IPIPE_CAR_FN2_MIN

Figure 179. ISS ISP IPIPE CAR Gain 1 and Gain 2 Functions



The output from the gain control functions is shown by the formula in Figure 180 (the fractional parts are rounded down).

Figure 180. ISS ISP IPIPE CAR Gain Control Functions Output Formula

$$Cb_{out} = \frac{(Cb_{in} - 128) \times gain}{256} + 128$$

$$Cr_{out} = \frac{(Cr_{in} - 128) \times gain}{256} + 128$$

camss-671

The dynamic switching between the 3 × 3 median filter and gain control is done in the following way:

1. If $(cb^2 + cr^2)/256$ switch_limit, use the median filter.
2. If (switch_coef)x$((cb^2 + cr^2)/256) = cb^2$, use gain control.
3. If both conditions are not met, do not apply any filter.

The values of switch_limit and switch_coef are determined by the values of the IPIPE_CAR_SW[7:0] SW0 and IPIPE_CAR_SW[15:8] SW1 bit fields, respectively. To prevent desaturation of colors, the first condition limits the area where the median filter is applied to only an unsaturated area.

With the appropriate value, this condition selectively removes the purple artifact.

3.3.3.18 ISS ISP IPIPE Chroma Gain Suppression

The chroma gain suppression (CGS) module controls the gain of Cb and Cr in dark or overly bright conditions. The function is enabled from the IPIPE_CGS_EN[0] EN bit.

Cb_{output} = (total_gain x Cb_{input}) / 256

Cr_{output} = (total_gain x Cr_{input}) / 256

The gain value is determined by the formula shown in Figure 181.

Figure 181. ISS ISP IPIPE CGS Gain Value Formula

$$gain_1 = \begin{cases} max\left(256 - g_{Y1} \times (thrY_1 - Y) >> shf_{Y1}, min\,Y_1\right) & Y \le thrY_1 \\ 256 & thrY_1 < Y < thrY_2, \\ max\left(256 - g_{Y2} \times (Y - thrY_2) >> shf_{Y2}, min\,Y_2\right) & thrY_2 \le Y \end{cases}$$

$$gain_2 = \begin{cases} max\left(256 - g_C \times (thrC - C) >> shf_C, min\,C\right) & C \le thrC \\ 256 & thrC < C' \end{cases}$$

$$0 \le thrY_1, thrY_2, thrC \le 255$$

$$0 \le g_{Y1}, g_{Y2}, g_C \le 255$$

$$0 \le shf_{Y1}, shf_{Y2}, shf_C \le 7$$

camss-673

The parameters in the formula in Figure 181 are specified by registers. Table 177 lists these parameters and registers.

Table 177. ISS ISP IPIPE CGS Gain Value Formula Parameters

Formula Parameter	Register
thrY1	IPIPE_CGS_GN1_L_THR
gY1	IPIPE_CGS_GN1_L_GAIN
shfY1	IPIPE_CGS_GN1_L_SHF
thrY2	IPIPE_CGS_GN1_H_THR
gY2	IPIPE_CGS_GN1_H_GAIN
minY1	IPIPE_CGS_GN1_L_MIN
minY2	IPIPE_CGS_GN1_H_MIN
shfY2	IPIPE_CGS_GN2_L_SHF
thrC	IPIPE_CGS_GN2_L_THR
gC	IPIPE_CGS_GN2_L_GAIN
shfC	IPIPE_CGS_GN2_L_SHF
minC	IPIPE_CGS_GN2_L_MIN

Figure 182 and Figure 183 also show these functions.

Figure 182. ISS ISP IPIPE CGS Gain 1

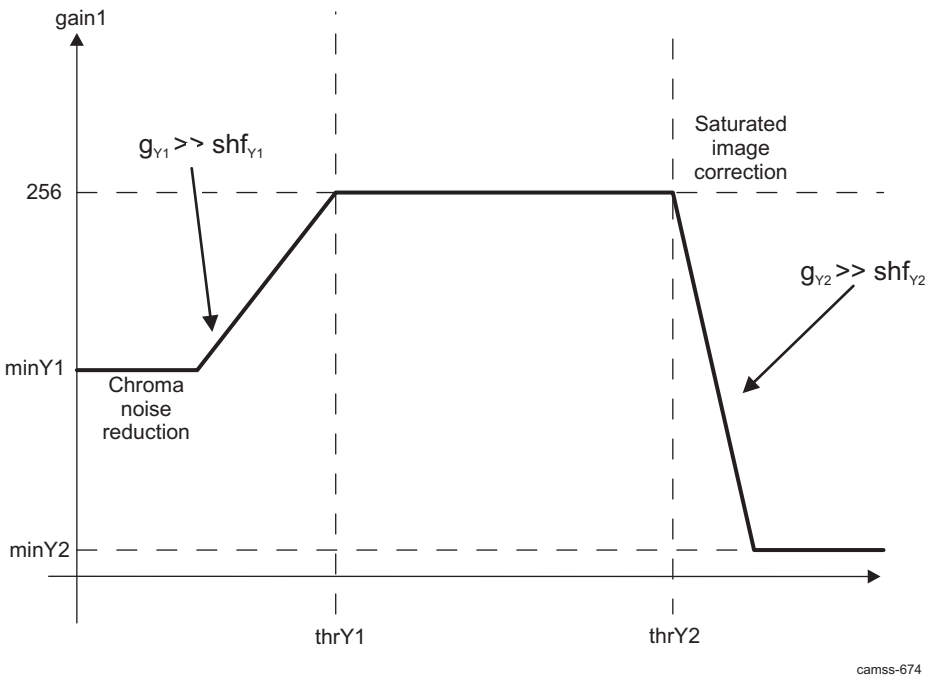
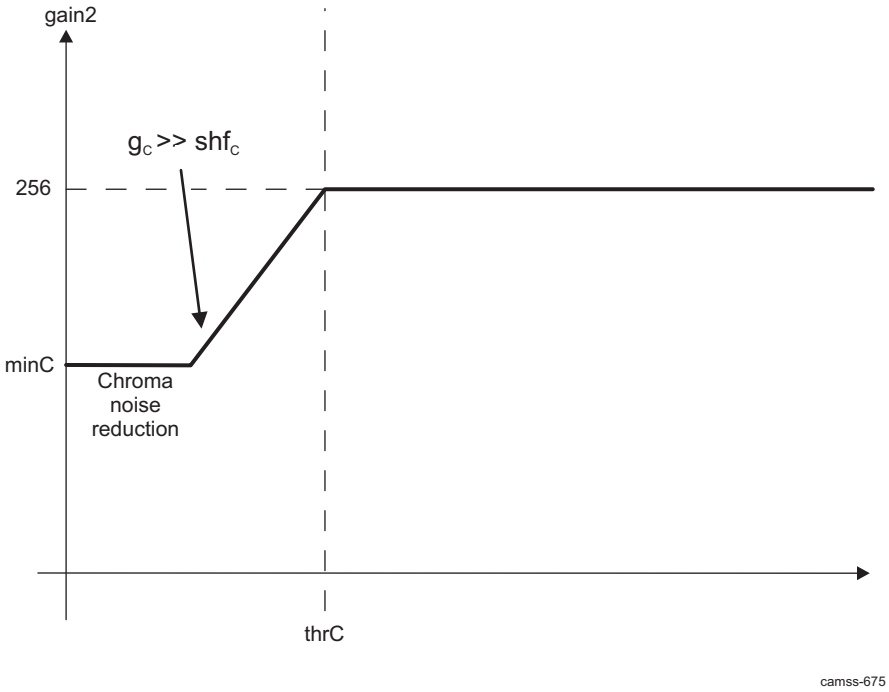


Figure 183. ISS ISP IPIPE CGS Gain 2



3.3.3.19 ISS ISP IPIPE Histogram

NOTE: The boxcar function can be used simultaneously with the histogram function if needed.

NOTE: For the locations of histogram memory access, see Section 3.3.8.

The histogram module counts the number of pixels that have a value in a region and can be enabled from the IPIPE_HST_EN[0] EN bit. Moreover, if enabled, the IPIPE_HST_MODE register can be set to work constantly or one time. If the IPIPE_HST_MODE[0] OST bit is set to 1, the histogram is disabled by clearing the IPIPE_HST_EN[0] EN bit to 0 after one run (one-shot mode).

After enabling the module, the following features are available:

- The data to be summed is taken from the DPC memory or RGB2YCbCr module. The choice is made in the IPIPE_HST_SEL[2] SEL bit.
- When data are collected from the DPC memory, the sampled colors are R/G/B/Y. Y is derived in the following method:

  Y = (HST_MUL_R*R + HST_MUL_GR*Gr + HST_MUL_GB*Gb + HST_MUL_B*B) (2)

  For the G histogram, Gb, Gr, or the average is used, through the IPIPE_HST_SEL[1:0] TYP bit field.
- Two sets of 512 × 20-bit memory are used.
- The number of bins can be set from 32 to 256 in the IPIPE_HST_PARA[13:12] BIN bit field.
- The number of regions (areas) from 1 to 4; each region can be enabled through IPIPE_HST_PARA[x] RGNx (where x = 0 to 3). The positions of the regions are defined in IPIPE_HST_x_VPS and IPIPE_HST_x_HPS, and the vertical and horizontal size are defined by IPIPE_HST_x_VSZ and IPIPE_HST_x_HSZ, respectively (where x = 0 to 3).
- The number of regions × the number of bins = 256.
- Each region can be turned on/off counting.
- The regions have priority orders.
- Each region has its own start coordinate X/Y (12 bits) and horizontal/vertical sizes (12 bits)
- When regions are overlapped, the value in the overlapped region is accumulated only in the region with the highest priority.
- The number of colors to be counted is from 1 to 4. Each color in all regions can be turned off counting (the IPIPE_HST_PARA[7:4] bit field).
- The value of each pixel is down-shifted (0 ~ 11) before counting using the IPIPE_HST_PARA[11:8] SHF bit field.
- When the value of a bin reaches (2^{20} – 1), the value is saturated until the memory is cleared.
- Number of bins: 32, 64, 128, or 256

The histogram memory can be cleared at the VD signal. When the memory is cleared, the first line of each frame cannot be sampled by the histogram if the width of the frame is larger than 512. If the width of the frame is smaller than 512, the first ceil (512/width) lines cannot be collected, where ceil(x) is the smallest integer value above x. If the clearing function is not enabled, the histogram bins are accumulated over the previous values.

The histogram has two banks of memories, which can be switched alternatively. The two memory banks are slipped into four histogram memory tables. Only two tables cam be used at a time: output memory tables 0 and 1, or tables 2 and 3. To initialize tables, the IPIPE_HST_TBL[1] CLR bit is set to 1, and to select which set of tables is to be used the IPIPE_HST_TBL[0] SEL bit can be switched between 0 and 1.

A gain for each color can be applied using the IPIPE_HST_MUL_x registers, where x = R, GR, GB, or B.

3.3.3.20 ISS ISP IPIPE Boxcar

The boxcar module generates a boxcar by taking mosaic image data and averaging the red, green, and blue pixels in an (8 × 8) or (16 × 16) block to produce one red, green, and blue output, as shown in Figure 184 and in Figure 185. Here, similar to the histogram module, the boxcar is enabled from the IPIPE_BOX_EN[0] EN bit, and if the mode is set to run once (one shot) (IPIPE_BOX_MODE[0] OST = 1), the enable bit is cleared after the run. The size of the blocks is determined from the IPIPE_BOX_SHF[0] SEL bit, where if set to 0 = 8 × 8, and 1 = 16 × 16.

The result of this operation is a full-color image with (1/64) or (1/256) area of the original image. The maximum input horizontal width is 8190 pixels when a 16 ×16 block is used; the width is 4096 when an 8 × 8 block is used. Also, the image size (width and height) must be multiple of 16 for a 16 × 16 block, and multiple of 8 for an 8 × 8 block. Boxcar operation works on 12-bit Bayer data and outputs 16-bit data. The output data is 48-bit RGB data for each 8 × 8 or 16 × 16 block. The 48-bit data is aligned in 64-bit format in SDRAM as shown in Section 3.3.3.20. The first address of SDRAM access is specified by the IPIPE_BOX_SDR_SAD_H and IPIPE_BOX_SDR_SAD_L registers. The output data are written to SDRAM continuously line by line; there is no address offset between lines. After the image transfer of each frame completes, the ipipe_eof signal is sent to buffer logic. This signal is issued at the same timing as ipipe_int_dma.

Figure 184. ISS ISP IPIPE Boxcar Operation (8 × 8 Block)

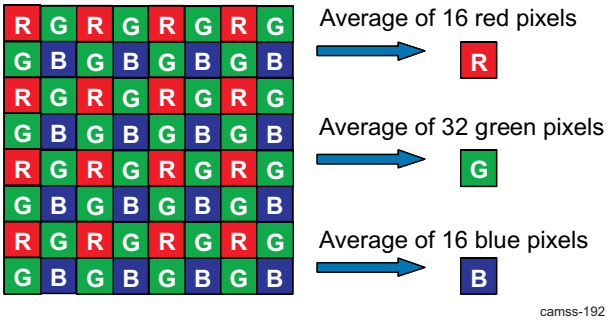


Figure 185. ISS ISP IPIPE Boxcar Operation (16 × 16 Block)

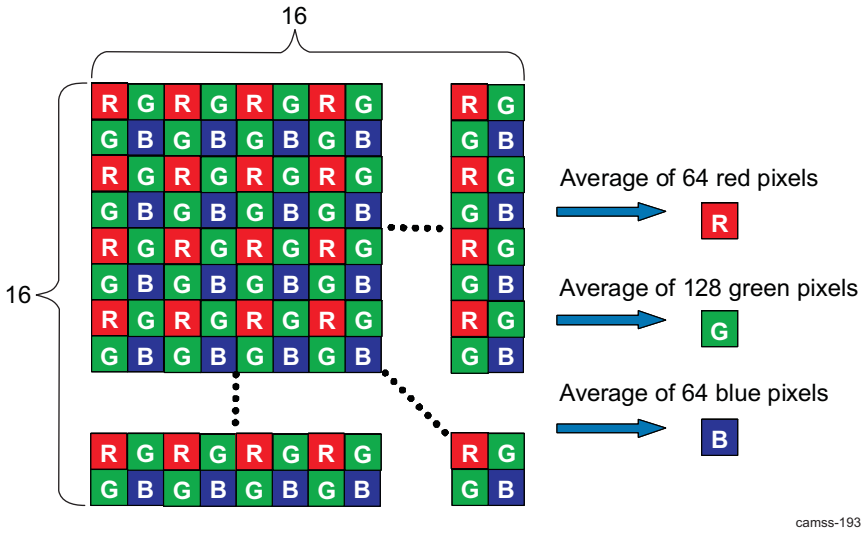
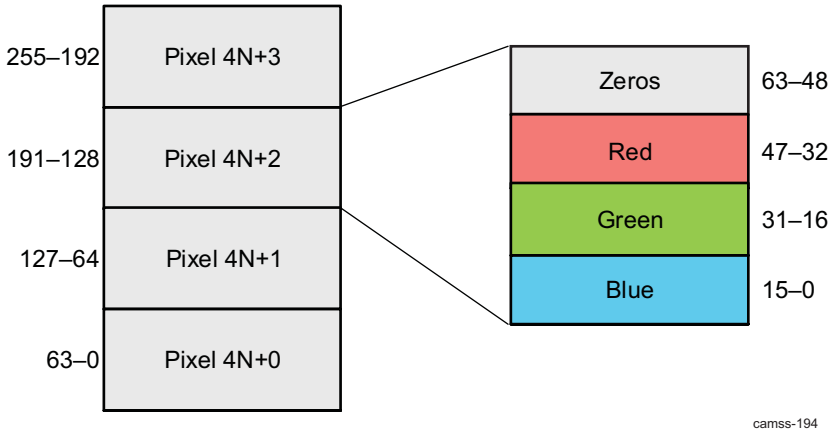


Figure 186. ISS ISP IPIPE Boxcar Data Packing in SDRAM



The right shift value is specified by the IPIPE_BOX_SHF register, which has a range of 0 to 4. (The shift down is performed to fit the 20-bit accumulated value into 16-bit output.) For green signal processing, a divide-by-two operation rounds off the least-significant bit (LSB).

3.3.3.21 ISS ISP IPIPE Boundary Signal Calculator

NOTE: For the locations of BSC memory access, see Section 3.3.8.

The BSC generates the row summations and column summations from YCbCr 4:4:4 data. Figure 187 is the block diagram of the BSC. The calculator can be initialized from the IPIPE_BSC_EN[0] EN bit, and the mode can be set from the IPIPE_BSC_MODE[0] OST bit to free run or one shot (run once and then de-initialize itself by clearing the IPIPE_BSC_EN register).

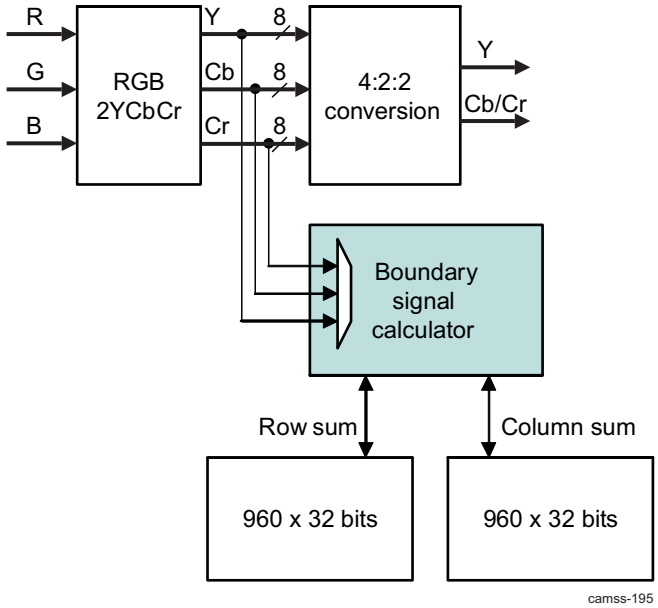
BSC generates two kinds of vectors from one of Y, Cb, or Cr data:

- Vector of sums of row pixels (can be set from the IPIPE_BSC_TYP[2] REN bit)
- Vector of sums of column pixels (can be set from the IPIPE_BSC_TYP[3] CEN bit)

For both row and column sums, up to four vectors can be generated. Both row and column sums are 16-bit values, which are stored to 960 × 32-bit memories, respectively. The maximum number of sums is 1920 for both row and column. The elements to be summed can also be selected. The IPIPE_BSC_TYP[0:1] COL bit can be set to 0xX, where X = 0 to 2 for Y, Cb, and Cr.

In case the successive frames are to be processed, there is a limitation that some of the lines (more than 16 lines) between the frames cannot be used for calculation.

Figure 187. ISS ISP IPIPE Block Diagram of BSC Module



For row sum vectors, the following parameters are required:

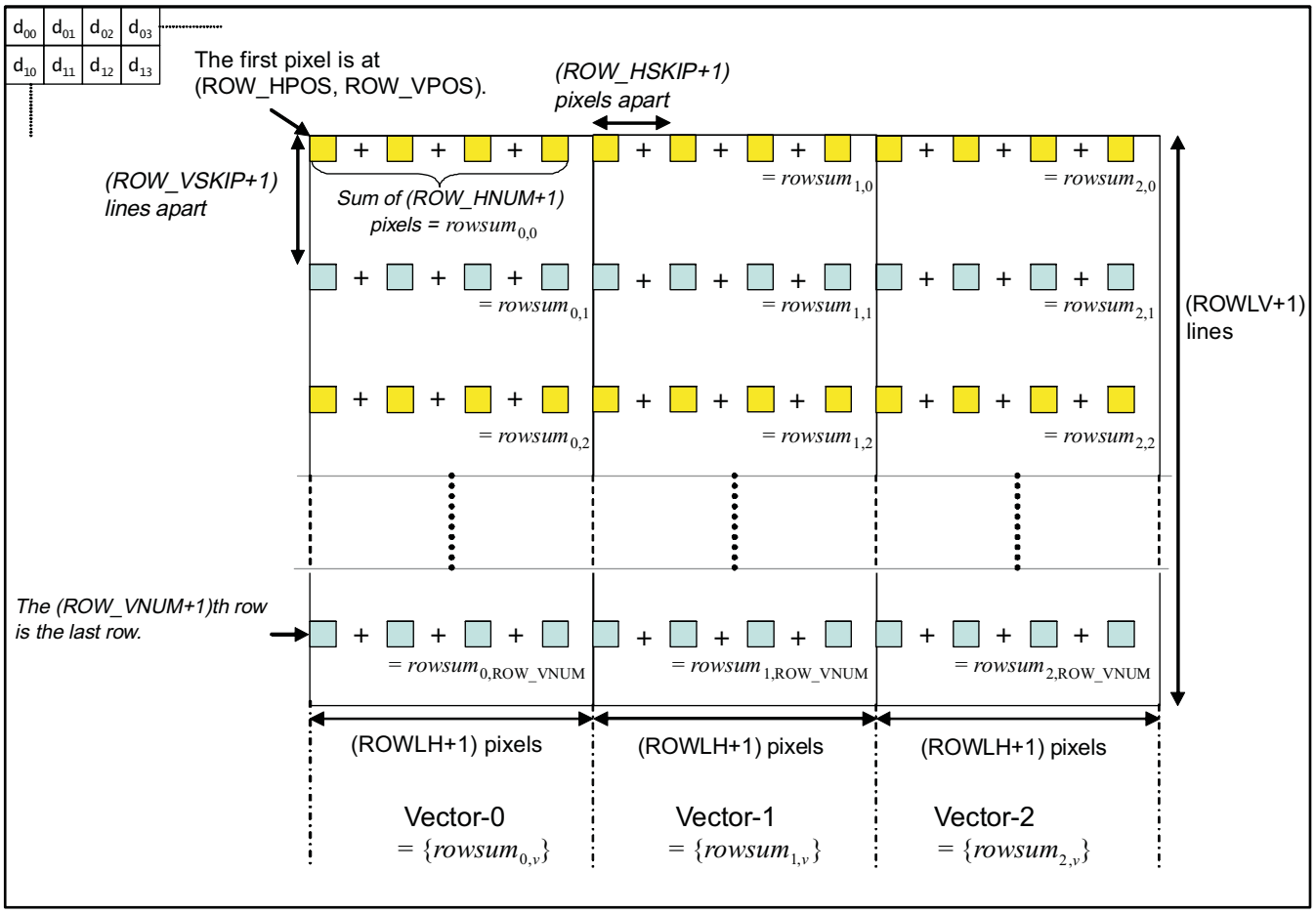
- The position of the first pixel to be summed (the IPIPE_BSC_ROW_HPOS and IPIPE_BSC_ROW_VPOS registers for horizontal and vertical, respectively)
- The spacing between adjacent sampling pixels (the IPIPE_BSC_ROW_HSKIP and IPIPE_BSC_ROW_VSKIP registers for horizontal and vertical, respectively)
- The number of sampled pixels (the IPIPE_BSC_ROW_HNUM and IPIPE_BSC_ROW_VNUM registers for horizontal and vertical, respectively)
- The number of row sum vectors (IPIPE_BSC_ROW_VCT)
- The down shift value of input data (IPIPE_BSC_ROW_SHF)

Figure 188 describes these parameters.

The number of row sums cannot exceed 1920. The IPIPE_BSC_ROW_VCT and IPIPE_BSC_ROW_VNUM registers must be set according to the following equation:

(IPIPE_BSC_ROW _VCT +1) * (IPIPE_BSC_ROW _VNUM +1) = 1920

Figure 188. ISS ISP IPIPE BSC Row Sum Vector Calculation



For column sum vectors, the following parameters are required:

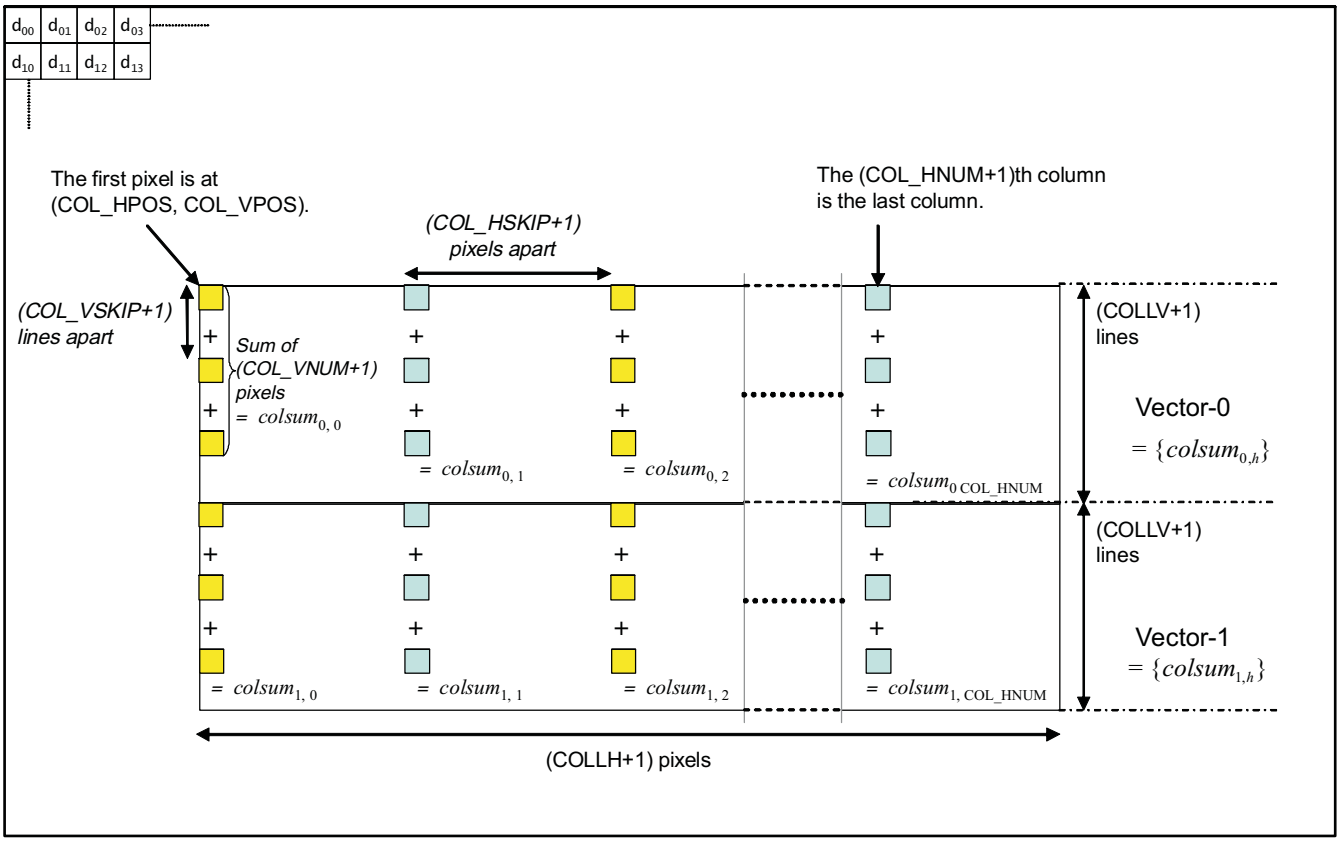
1. The position of the first pixel to be summed (the IPIPE_BSC_COL_HPOS and IPIPE_BSC_COL_VPOS registers for horizontal and vertical, respectively)
2. The spacing between adjacent sampling pixels (the IPIPE_BSC_COL_HSKIP and IPIPE_BSC_COL_VSKIP registers for horizontal and vertical, respectively)
3. The number of sampled pixels (the IPIPE_BSC_COL_HNUM and IPIPE_BSC_COL_VNUM registers for horizontal and vertical, respectively)
4. The number of column sum vectors (IPIPE_BSC_COL_VCT)
5. The down shift value of input data (IPIPE_BSC_COL_SHF)

Figure 189 describes these parameters.

The number of column sums cannot exceed 1920. The IPIPE_COL_VCT and IPIPE_COL_HNUM registers can be set according to the following equation:

(BSC_COL _VCT +1) * (BSC_COL _ HNUM +1)=1920 (3)

Figure 189. ISS ISP IPIPE BSC Column Sum Vector Calculation



camss-197

3.3.4 ISS ISP RSZ Functional Description

3.3.4.1 ISS ISP RSZ Overview

The RSZ module rescales images into various sizes ranging from x1/4096 scale-down to x16 scale-up. It also works in conjunction with the rotational engine (ROT) in SIMCOP for rotating images. The RSZ data slave interfaces support a parallel video port (VP). The RSZ module can produce two output images simultaneously, because there are two independent resizer engines. The input data used by the two resizer engines is the same.

The RSZ module has the following capabilities:

- Input/output data formats:
  - Common input data for the two resizers
  - Independent output data formats for the two resizers
  - Supports YUV4:2:2 input and YUV4:2:2/YUV4:2:0 or RGB5:6:5/ARGB32 output.
  - Supports YUV4:2:0 input data and YUV4:2:0
    - YUV4:2:0 input format not supported natively
      - Two passes required: luma followed by chroma or vice versa
      - Only supported from memory to memory
  - Supports RAW Bayer input and RAW Bayer output
    - RAW format invariant. Takes whichever RAW format at the input and writes it out unmodified: 16 bits are read, 16 bits are written out.
    - No resizing can take place on RAW data.
- Resizer capabilities:

- Input image cropping
  - Common for the two resizer engines: Same input data before and after cropping
  - Supported on YUV4:2:2 and RAW data formats
  - Supported on pass-through mode data path
  - Supported on bypass mode data path
- Dual resizer engines: RSZ-A and RSZ-B
  - Up to x20 upsampling and x1/4096 downsampling on both engines
  - Up to 200-MHz pixel throughput on both resizer engines
  - Programmable data rate control to smooth the peak memory bandwidth
- RSZ-A
  - Horizontal resolution of up to 8K pixels on RSZ-A.
- RSZ-B
  - Horizontal resolution of up to 8K pixels on resizer RSZ-B
- Independent Y and Cb/Cr phases on horizontal and vertical axis
  - Enables to take care of different YUV4:2:0 phases used in different video formats
  - Enables frame division mode: Images can be stitched together with the right phase.
- Rescaling: two modes supported
  - Normal mode for upscale and downscale: Higher flexibility but lower downscale quality
  - Downscale mode for downscale only: Lower flexibility but higher downscale quality
- Filtering: two modes supported
  - Independent settings for the horizontal and vertical directions
  - 3-tap low pass filter with 2-tap linear interpolation
  - 4-tap cubic interpolation
- Flip support of the output image
  - Horizontal flip
  - Vertical flip
- Pixel duplication on the top/bottom, left/right sides
  - Avoids losing pixels at the image boundaries because of the filtering
- Support pass-through and bypass modes: Resizer engines bypassed
  - Pass-through mode
    - RAW and YUV4:2:2 data support
    - Lower power consumption mode to transfer data to memory
    - Can transfer images larger than 8K pixels to memory
  - Bypass mode
  - RAW and YUV4:2:2 data support
  - Input buffer used. Can benefit from additional buffering in case the BL module memory is not big enough and back pressure occurs.

- Slave data interface: VP interface
  - Two VP interfaces: The programming model selects which VP is used to input data to the RSZ module. Both VPs cannot be active simultaneously.
    - VP 1: Typically connected to the IPIPE module.
    - VP 2: Typically connected to the IPIPEIF module.
  - Up to 200-MHz pixel clock
- Master data interface:
  - Two interfaces to the BL module
    - Up to 200 MHz

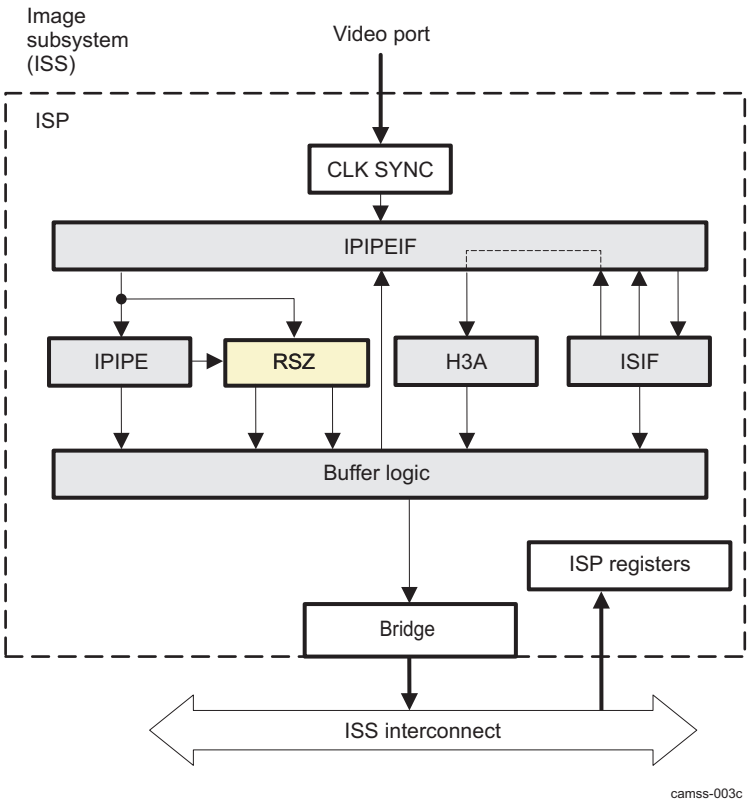
- 32-bit wide, 32-byte long requests
- Accesses are aligned on 32-byte boundaries
- Used to transfer data to memory
- Each interface is dedicated to a single output image.

– Addressing modes
- Linear
- Circular

- Configuration interface:
  – Up to 100 MHz
  – 32-bit wide
  – Used to configure the resizer registers
- Power management:
  – Independent clock domains for the two resizers
    - Each resizer engine can be gated off separately
- Error management:
  – FIFO overflow detection on the input buffers

Figure 190 show the RSZ module connections to other submodules of the ISP.

Figure 190. ISS ISP RSZ High-Level Diagram



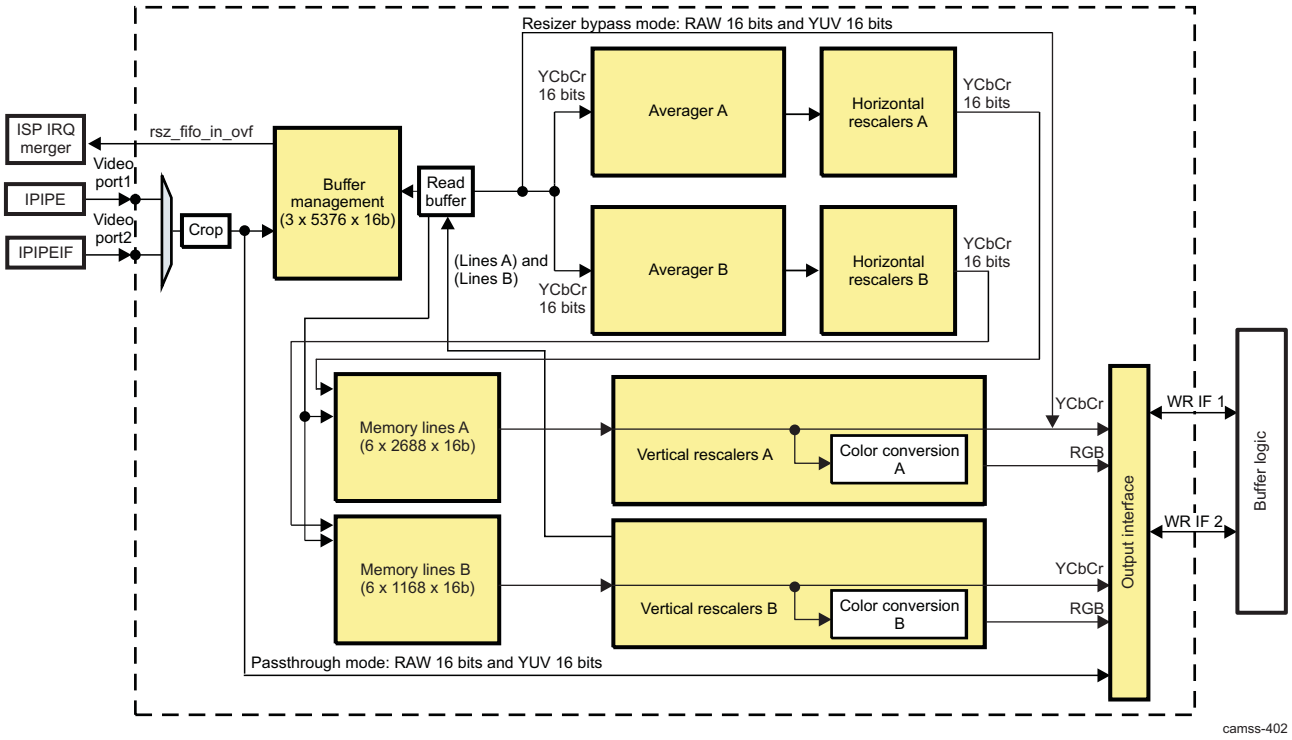
3.3.4.2 ISS ISP RSZ Top-Level Block Diagram

Figure 191 is the top-level block diagram of the RSZ module. The RSZ module comprises the following submodules: cropping, input data buffering, data requestor, averager, data saturation, and resizer Interpolation (comprised of horizontal rescaler, vertical rescaler, color conversion, and output interface) (see the following sections for more information).

The RSZ module comprises two independent resizer engines with the same capabilities (except for the memory line size). The input data can come from VP 1 or VP 2. Software must determine and control which interface is selected.

The RSZ module includes one VBUSP slave port, which is used to control the RSZ registers. It also includes two MTC master ports, which are used to pass the pixels to the BL. The BL in turn creates the burst requests to the memory subsystem (see Section 3.3.4.3, *ISS ISP RSZ Interfaces*).

Figure 191. ISS ISP RSZ Top-Level Block Diagram



3.3.4.3 ISS ISP RSZ Interfaces

The RSZ module has the following data interfaces:

- One 32-bit read/write point-to-point pending VBUSP interface
- Two slave VP interfaces for transport YUV and RAW data
- Two MTC interfaces to BL for RZA A and RZA B, with only write capabilities

3.3.4.3.1 ISS ISP RSZ VBUSP interface

The VBUSP interface is a 32-bit read/write capable interface. The VBUSP interface must be programmed in a way that back-to-back requests are possible for read and write. The RSZ_GCK_MMR[0] MMR bit enables the memory register access from the VBUSP interface to enable transfer and signal such as MMR request, direction, enable write/read data can be enabled.

3.3.4.3.2 ISS ISP RSZ Video Port Interfaces

The VP interfaces are slave interfaces; one is connected to IPIPE, and the other to IPIPEIF. These interfaces are for data transfer. Table 178 lists the format supported across IPIPE/IPIPEIF and RSZ. Signals coming from IPIPE and IPIPEIF can be write-enable signals. The RSZ_SRC_MODE[1] WRT bit is set whether or not the write enable signals are considered. This is a line-valid qualifier. This signal is sampled on the rising edge of HD, and the value is used for the full line.

Table 178. ISS ISP RSZ VP Supported Formats

	VP Signals: From IPIPE and IPIPEIF Modules (dat[15:0] Register)															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW16	R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
YUV4:2:2 16 bits	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Cb7 Cr7	Cb6 Cr6	Cb5 Cr5	Cb4 Cr4	Cb3 Cr3	Cb2 Cr2	Cb1 Cr1	Cb0 Cr0
YUV4:2:0 Y data	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Low	Low	Low	Low	Low	Low	Low	Low
YUV4:2:0 Cb/Cr data	Low	Low	Low	Low	Low	Low	Low	Low	Cb7 Cr7	Cb6 Cr6	Cb5 Cr5	Cb4 Cr4	Cb3 Cr3	Cb2 Cr2	Cb1 Cr1	Cb0 Cr0

NOTE: The formats are set from the IPIPE and IPIPEIF registers. For more information, see Section 3.3.3 and Section 3.3.2.

3.3.4.3.3 ISS ISP RSZ MTC interfaces

The RSZ module includes two write-only MTC interfaces. Their implementation enables passing a maximum of eight 32-byte requests in ten clock cycles. The RSZ must be programmed to obtain smooth and average bandwidth to buffer logic module by setting a minimum interval between two successive requests (set the RSZ_DMA_RZA[15:0] RZA and RSZ_DMA_RZB[15:0] RZB bit fields for the A and B resizers as appropriate). This setting is not expected to be dynamic. It can be fixed setting from request to request and frame to frame. When the bandwidth is set appropriately, between the first valid translated pixel and the EOF signal sent to buffer logic, the RSZ_DMA_STA[0] STATUS bit can be seen, and it is high if the transfer over the MTC interfaces is active. Figure 192 shows how RSZ_DMA_RZx for resizers A and B affects the MTC data request generator.

Figure 192. ISS ISP RSZ MTC DMA Bandwidth Control

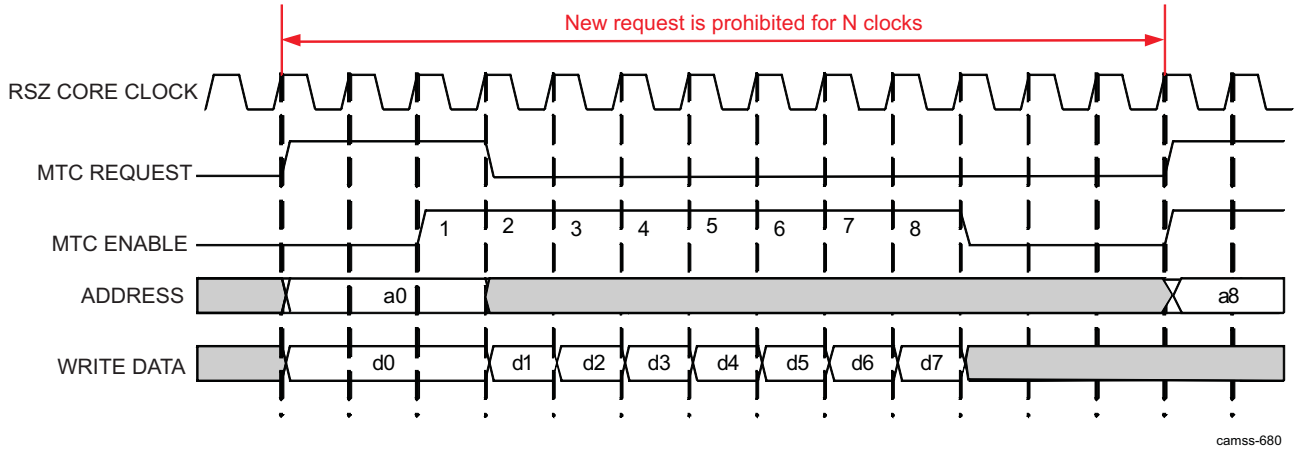
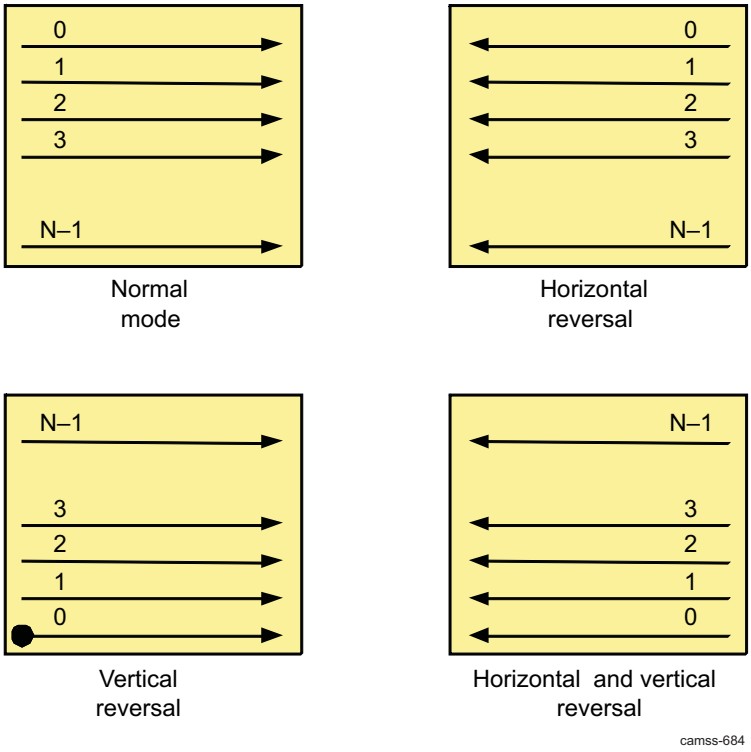


Figure 193 shows the pixel order in memory written by the MTC. The arrows do not represent the order in which data is written. Data are always written from left to right, whether horizontal reversal is enabled or not.

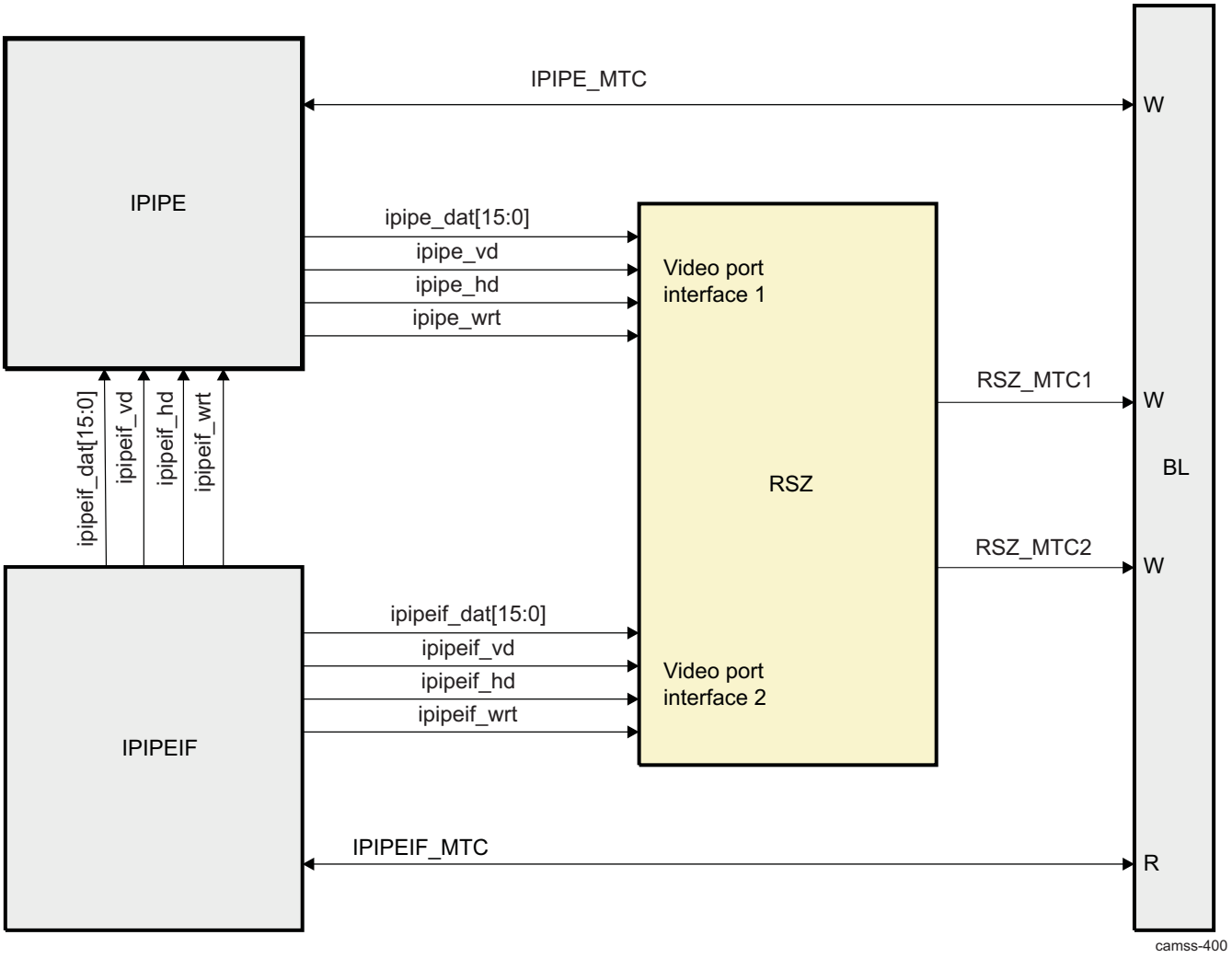
Figure 193. ISS ISP RSZ MTC Image Data Storage Pixel Order



3.3.4.4 ISS ISP RSZ Integration

Figure 194 shows how the VP and interfaces of the RSZ module are connected to surrounding modules at the ISP level. The RSZ module gets data from the IPIPEIF module or IPIPE module.

Figure 194. ISS ISP RSZ Typical Module Integration: High-Level Summary



The following constraints apply to the RSZ module:

- The data coming from the IPIPEIF module can be RAW or YUV4:2:2 data. Because the RSZ module can rescale only YUV4:2:2 data, the RSZ module must be configured in pass-through mode when RAW data is received on VP 2. It is possible to bypass the RSZ engine, if YUV4:2:2 data is sent but rescaling is not needed (bypass mode).
- The data coming from the IPIPE module can be RAW or YUV4:2:2 data. Eventually, YUV4:2:0 data can be sent through this path, but the data must be sent in two passes. Because the RSZ module can rescale only YUV4:2:2 data, the RSZ module must be configured in pass-through mode when RAW data is received on VP 1. It is possible to bypass the RSZ engine if YUV4:2:2 data is sent but rescaling is not needed (bypass mode).

VP 2 provides a way to bypass the IPIPE module when YUV4:2:2 data is received from the images sensor or when YUV4:2:2 is read back from memory with the SC or ISIF module.

Table 179 summarizes the different RSZ configuration possibilities as a function of the input data format.

Table 179. ISS ISP RSZ Data Flow vs. Input Data Format Constraints

VP 1 Data Format	VP 2 Data Format	RSZ-A Configuration	RSZ-B Configuration	Comments
RAW	N/A	Disabled	Disabled	RSZ module in pass-though mode

Table 179. ISS ISP RSZ Data Flow vs. Input Data Format Constraints (continued)

VP 1 Data Format	VP 2 Data Format	RSZ-A Configuration	RSZ-B Configuration	Comments
N/A	RAW	Disabled	Disabled	RSZ module in pass-though mode
YUV4:2:2	N/A	Disabled	Disabled	RSZ module in pass-though mode or bypass mode
YUV4:2:2	N/A	Enabled	Disabled	One output image
YUV4:2:2	N/A	Disabled	Enabled	One output image
YUV4:2:2	N/A	Enabled	Enabled	Two output images
N/A	YUV4:2:2	Disabled	Disabled	RSZ module in pass-though mode or bypass mode
N/A	YUV4:2:2	Enabled	Disabled	One output image
N/A	YUV4:2:2	Disabled	Enabled	One output image
N/A	YUV4:2:2	Enabled	Enabled	Two output images

3.3.4.5 ISS ISP RSZ Functional Description

To start up, the RSZ configuration can be set from the RSZ_SYSCONFIG register, which provides enabling the RSZ-A and RSZ-B clocks. The RSZ module does not have standalone reset and status check. Software reset must be done at the ISP level. Moreover, when enabled, the RSZ module can control the input data buffer, and when the rsz_stall_input signals are set from the RSZ_IN_FIFO_CTRL register (see Section 3.3.4.5.3.2.5), the RSZ module generates a stall signal that can be used by the master module sending data to the RSZ module when the data threshold is too high.

The RSZ_SRC_EN[0] EN bit starts the resizer processing. If the processing mode is set to one shot (one run and then turn off) from the RSZ_SRC_MODE[0] OST bit, the EN bit is cleared to 0.

The resizer can be configured to be bypassed in certain cases (see Figure 194 for the module constraints) from the RSZ_SRC_FMT0[1] BYPASS bit. The data can be sent directly from here to the output interface (bypass mode) or imported to the module buffer, but not manipulated and sent to the output interface (pass-through mode). The master device sending data to the resizer can be switched between IPIPEIF and IPIPE using the RSZ_SRC_FMT0[0] SEL bit. The resizer understanding of the data input is set from the RSZ_SRC_FMT1 register (for more information, see Table 181). The RSZ_SEQ.VRVX and RSZ_SEQ.HRVX registers can be set to flip the image horizontally or vertically, respectively (see Figure 193).

Depending on the mode to which the resizer is set, the core clock can be enabled from the RSZ_GCK_SDR register. Table 180 summarizes the behavior of the module for the different settings.

Table 180. ISS ISP RSZ Module Modes: Register Settings

Configuration Number	RSZ_SRC_ EN	RZA_EN	RZA_CLK_ EN	RZB_EN	RZB_CLK_ EN	RSZ_GCK_SDR.CORE	RSZ_SRC_FMT0.BYPASS	Comments
0	0	X	X	X	X	X	X	Data cannot go through the RSZ module.
								Interrupts are not issued.
1	1	0	X	0	X	1	0	Resizer A is disabled.
								Resizer B is disabled.
								It is best to have RZA_EN = RZB_EN = 0 to save power, but RZA_EN = RZB_EN = 1 is also supported.
								This configuration is supported but does not make sense because data cannot go through the module.
2	1	1	1	0	X	1	0	Resizer A is enabled.
								Resizer B is disabled.

Table 180. ISS ISP RSZ Module Modes: Register Settings (continued)

								It is best to have RZB_EN = 0 to save power, but RZB_EN=1 is supported as well.
3	1	0	X	1	1	1	0	Resizer A is disabled.
								Resizer B is enabled.
								It is best to have RZA_EN = 0 to save power, but RZA_EN = 1 is also supported.
4	1	1	1	1	1	1	0	Resizer A is enabled.
								Resizer B is enabled.
5	1	X	X	X	X	0	0	Bypass mode is enabled.
								Resizer core functional clock is disabled.
6	1	X	X	X	X	0	1	Pass-through mode enabled.
								Resizer core functional clock is disabled.
7	1	X	X	X	X	1	1	Pass-through mode is enabled.
								Resizer core functional clock is enabled.
								Not a preferred configuration. Configuration (6) saves power.

Table 181. ISS ISP RSZ Module Input Control: Register Settings

RSZ_SRC_FMT1.IN420	RSZ_SRC_FMT1.COL	Comments
0	X	YUV4:2:2 input. Chrominance is cosited.
1	0	YUV4:2:0 input. Valid data is Y, C is dummy. On the VP, YUV4:2:2 data is always assumed.
1	1	YUV4:2:0 input. Valid data is C, Y is dummy. On the VP, YUV4:2:2 data is always assumed.

The RSZ_YUV_PHS[0] POS bit sets the chrominance output. The RSZ module does not change the relative position of the chroma samples versus the luma samples between the input and output, and the chroma position at the output of the IPIPE module and at the output of the RSZ module must be identical. In other words, RSZ_YUV_PHS.POS = IPIPE_YUV_PHS.POS.

Settings are common for both resizer engines inside the RSZ module. Each engine (RZA or RZB) can be enabled from the RZx_EN register: select the mode from RZx_MODE, and select the input and output in the YUV color scheme from the RZx_420 register (valid only if YUV 4:2:2 is the input set from RSZ_SRC_FMT1.IN420). Table 182 summarizes the combination of settings available in the RZx_420 register.

Table 182. ISS ISP RSZ-A/RSZ-B Output Format Selection

RZx_420.YEN	RZx_420.CEN	Comments
0	0	Input is YUV4:2:2. Output is YUV4:2:2 if RZX_RGB_EN = 0 and RGB if RZB_RGB_EN = 1.
0	1	Input is YUV4:2:2. Output is the chroma of YUV4:2:0. RZX_RGB_EN is ignored. Must be used to rescale YUV4:2:0 data: 1st/2nd pass
1	0	Input is YUV4:2:2 .Output is the luma of YUV4:2:0. RZX_RGB_EN is ignored. Must be used to rescale YUV4:2:0 data: 2nd/1st pass
1	1	Input is YUV4:2:2 .Output is YUV4:2:0. RZX_RGB_EN is ignored.

3.3.4.5.1 ISS ISP RSZ Operating Modes

The RSZ module offers two basic rescaling modes. These modes are not built-in but are particular configurations, which means that other hybrid modes can be programmed. The normal mode provides more flexibility (the rescale ratio granularity is smaller) than downscale mode, but downscale mode produces better image quality (averager performs anti-aliasing):

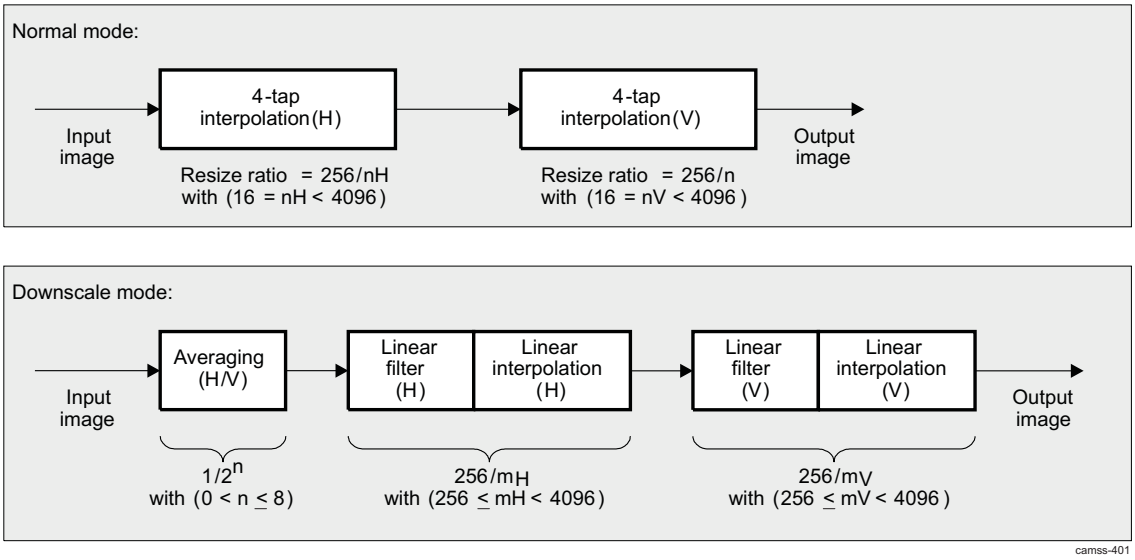
- Normal mode: The scaling process is carried out using interpolation with a 4-tap filter. The interpolation method is a 4-tap cubic convolution or a 3-tap linear filter + 2-tap linear interpolation. The user controls the type of interpolation that is used. The possible rescale ratios range from 1/16x to 20x.
- Downscale mode: The scaling process is the same as for normal mode, but an averaging function is placed before. It enables reaching much higher reduction factors while avoiding anti-aliasing artifacts. The interpolation method is a 4-tap cubic convolution or a 3-tap linear filter + 2-tap linear interpolation. The possible rescale ratios range from 1/4096x to 1x.

NOTE: The selection of the mode is independent for each resizer engine. One resizer engine can be configured in normal mode, while another is configured in downscale mode.

The RSZ module can produce two output images simultaneously, because there are two independent resizer engines. The input data used by the two resizer engines is the same. The RSZ output image sizes are limited to RSZ_GNC[0:12] RSZA_MEM_LINE_SIZE pixels/line for RSZ-A (5376 pixels/line) and RSZB_MEM_LINE_SIZE[16:28] for RSZ-B (2336 pixels/line).

Figure 195 shows the RSZ operating modes. Nothing prevents the use of linear interpolation in normal mode or bicubic interpolation in downscale mode; similarly, it is possible to mix the interpolation modes for horizontal and vertical filtering. This is fully programmable.

Figure 195. ISS ISP RSZ Operating Modes



3.3.4.5.1.1 ISS ISP RSZ Operating Modes and Maximum Input Clock

The maximum output pixel clock on both resizers is 200 MHz (100 percent optimal power performance [OPP]); that is, a pixel throughput of 200 MPix/s. Moreover, hardware takes care of the following constraints:

- When both resizer engines are configured to perform downscaling, there is no particular constraint on the VP pixel clock. The VP pixel clock can be as high as 200 MHz.
- When one resizer engine is configured to perform upscaling and the second resizer engine is configured to perform downscaling, the VP pixel clock must be limited. The VP must be lower than:

  clk_pix = (200 MHz/(Vertical Upscale Ratio * Horizontal Upscale Ratio)) (4)

  For example, if a 4x upscale ratio happens horizontally and vertically, then the input pixel clock must be lower than 200/(4*4) = 12.5 MHz.

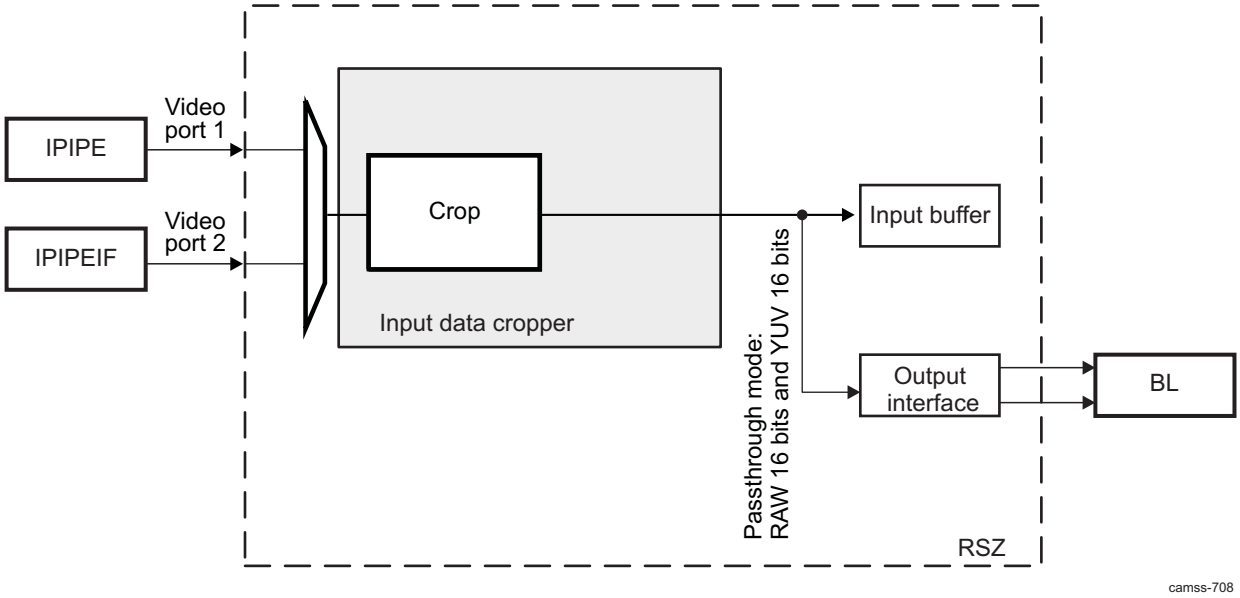
  It is the reason why it is not possible to perform digital zoom upscaling on the fly. It is necessary to acquire the pixels to memory first and to read them back at a pace that does not exceed the previously discussed constraints. At the ISS level, data can be read back from memory from the SC or ISIF module.

- When the two resizer engines are configured to perform upscaling, the VP pixel clock must be limited. In that case, the VP frequency is limited by the resizer engine having the larger rescale ratios.

3.3.4.5.2 ISS ISP RSZ Input Data Cropper

The data coming from the VPs into the RSZ module can be cropped: this applies to RAW and YUV4:2:2 data. It is mandatory to crop the data as early as possible in the RSZ processing pipeline to reduce power consumption. It is mandatory to crop the data before storing it in the input data buffer. Figure 196 is the block diagram of the RSZ input data cropper.

Figure 196. ISS ISP RSZ Input Data Cropper Block Diagram



The input data are in YUV4:2:2 interleaved format or RAW format.

For YUV4:2:2 format, the data come as Cb0, Y0, Cr0, Y1, Cb2, Y2, Cr2, Y3, etc. Y denotes the luma component value and Cb/Cr denotes the chroma component values. There are as many Y components as Cb/Cr components per line.

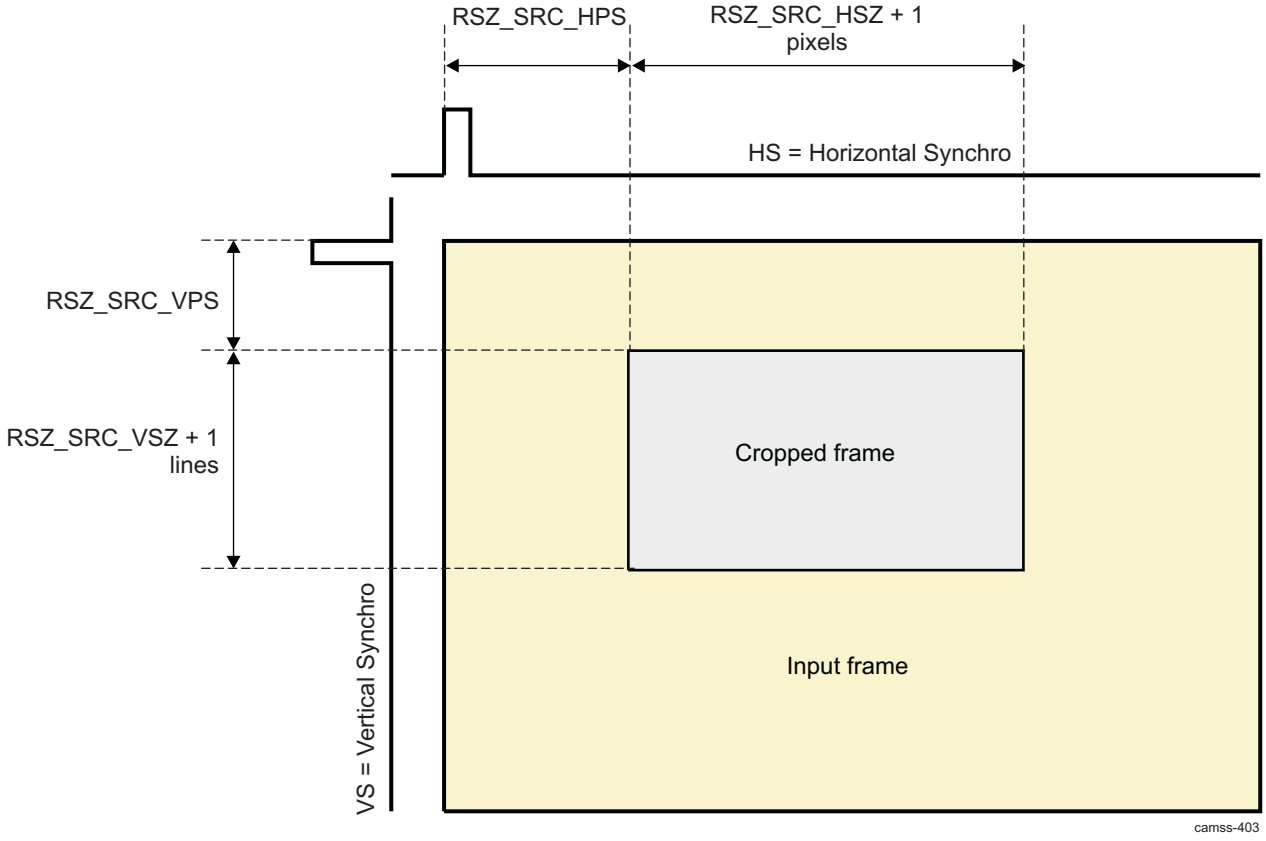
Figure 197 shows input data cropping. Only the cropped data is stored in the input data buffer. The names used in the figure correspond to the register names. If no cropping is desired, the RSZ_SRC_HPS and RSZ_SRC_VPS registers, which set the horizontal and vertical positions, must be set to 0; the start size from the RSZ_SRC_HSZ register must be set to the input image width minus 1, and the RSZ_SRC_VSZ register must be set to the input image height minus 1. These are typical settings for both resize engines inside the RESIZER. After setting, more flexibility is present through RZx_i_VPS and RZx_i_HPS for vertical and horizontal positioning, respectively, of the input/output (where x = A or B, i = I or O).

Depending on the input data format, different constraints apply to the registers that set the cropping parameters:

- For YUV4:2:2 format, the vertical start positions of the cropped frame can be even or odd. However, the horizontal start position must be even: the reason is to always start with the same pattern: Cb_{2n}, Y_{2n}, Cr_{2n}, Y_{2n+1},.. For the same reason, the horizontal size of the cropped frame (RSZ_SRC_HSZ + 1) must be an even number. Finally, the vertical size of the cropped frame can be odd or even.
- For RAW format, the vertical start position of the cropped frame can be even or odd. The vertical size can be even or odd. The horizontal resolution must be even.

These features and constraints are common for both resizer engines inside the resizer module. Figure 197 shows the input data cropping.

Figure 197. ISS ISP RSZ Input Data Cropping



3.3.4.5.3 ISS ISP RSZ Input Data Buffer

3.3.4.5.3.1 ISS ISP RSZ Input Data Buffer Overview

The input data buffering is used to store pixel data received from the cropping submodule as well as changing clock domains.

- The data is written into the buffer at the pixel clock frequency: maximum frequency is 200 MHz (remainder 100 percent OPP).
- The data is read from the buffer at the functional clock frequency: maximum frequency of 200 MHz (remainder 100 percent OPP).

The data write period into the buffer is not constant. The period follows a typical image sensor frame structure with intervals of data transmission and horizontal and vertical blanking periods (no data transmission). Blanking periods must be used by the resizer engines to keep emptying the buffer if there is data to process.

The data read period from the buffer is not constant because the resizer engines have to generate more lines or less lines during the vertical processing of the image. This is to compensate for the fact that the vertical resize ratio is set up by a fractional number. For every input line, the vertical resize process outputs N or N + 1 lines, as described in the following sections.

3.3.4.5.3.2 ISS ISP RSZ Input Data Buffer Vertical Resize Examples

The following two examples show the number of output lines that the vertical resize process must generate: in the first example N = 1, and in the second example N = 2.

3.3.4.5.3.2.1 ISS ISP RSZ Input Data Buffer Vertical Resize Example 1

The horizontal rescale ratio is set to 1.0x, and the vertical upscale ratio is set to 1.1x (fractional number). The resizer engine cannot output 1.1 lines; instead, at every 10 input lines, 11 output lines are generated. The VP pixel clock must be limited to 200/1.1 = 181 MHz.

Table 183 summarizes the number of lines that the resizer must generate. The number of lines that must be generated is not constant (1 or 2 lanes are output).

Table 183. ISS ISP RZA Input Data Buffering: Vertical Upscale Ratio = x1.1

Old Vertical Ratio	New Vertical Ratio	Vertical Resizer	Comments
	(Cumulated)	Output lines	
0	1.1	1	floor(1.1) – floor(0.0) = 1
1.1	2.2	1	
2.2	3.3	1	
3.3	4.4	1	
4.4	5.5	1	
5.5	6.6	1	
6.6	7.7	1	
7.7	8.8	1	
8.8	9.9	1	
9.9	11	2	floor(11.0) – floor(9.9) = 2
11	12.1	1	
12.1	13.2	1	
13.2	14.3	1	
[...]	[...]	[...]	

3.3.4.5.3.2.2 ISS ISP RSZ Input Data Buffer Vertical Resize Example 2

The horizontal rescale ratio is set to 1.0x, and the vertical upscale ratio is set to 2.7x (fractional number). The resizer engine cannot output 2.7 lines; instead, at every 10 input lines, 27 output lines are generated. The VP pixel clock must be limited to 200/2.7 = 74 MHz.

Table 184 summarizes the number of lines that the resizer will have to generate. The number of lines that must be generated is not constant (2 or 3 lanes are output).

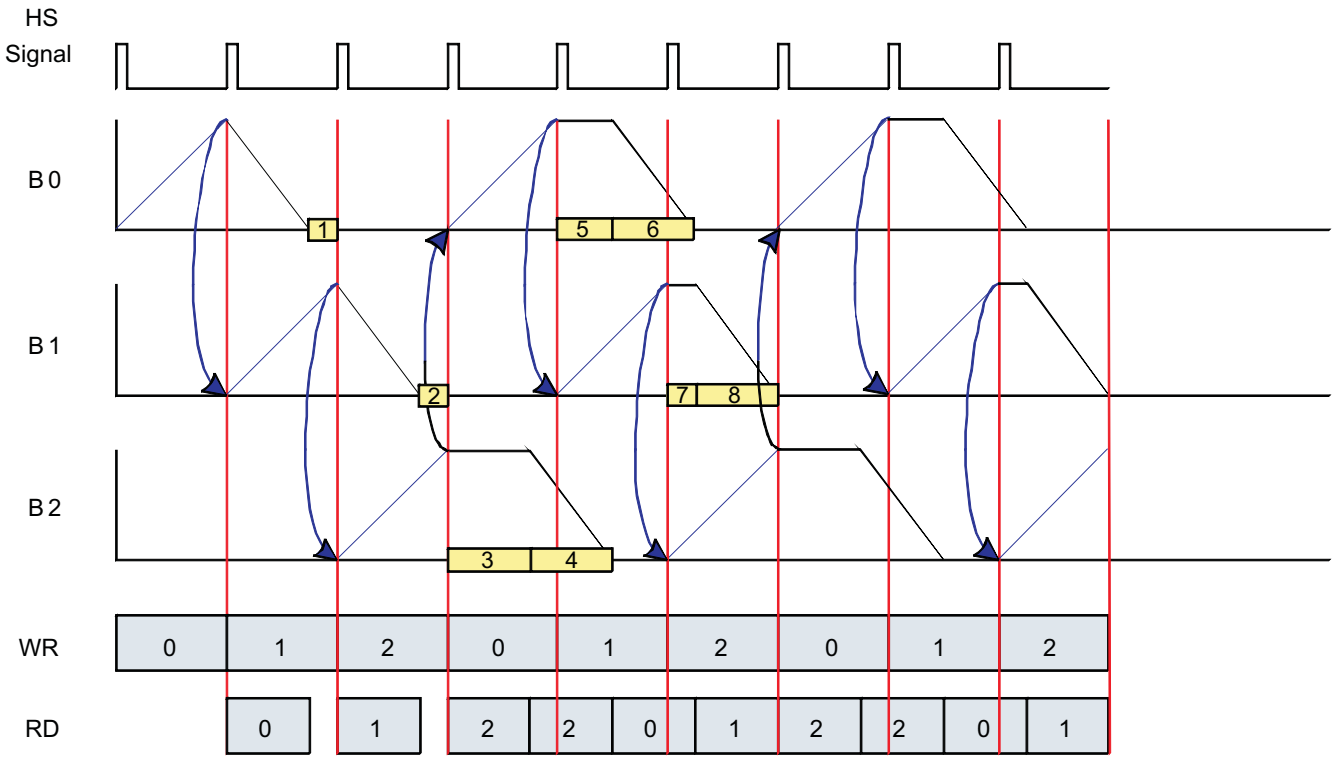
Table 184. ISS ISP RZA Input Data Buffering: Vertical Upscale Ratio = x2.7

Old Vertical Ratio	New Vertical Ratio	Vertical Resizer	Comments
	(Cumulated)	Output lines	
0	2.7	2	floor(2.7) – floor(0.0) = 2
2.7	5.4	3	floor(5.4) – floor(2.7) = 3
5.4	8.1	3	floor(8.1) – floor(5.4) = 3
8.1	10.8	2	floor(10.8) – floor(8.1) = 2
10.8	13.5	3	
13.5	16.2	3	
16.2	18.9	2	
18.9	21.6	3	
21.6	24.3	3	
24.3	27	3	
27	29.7	2	
29.7	32.4	3	
32.4	35.1	3	
[...]	[...]	[...]	

3.3.4.5.3.2.3 ISS ISP RSZ Input Data Circular Buffer

Table 183 and Table 184 show that the output data rate varies within a frame. However, the input pixel clock is always constant (except during blanking periods). Thus, a simple double-buffering scheme is not sufficient because overflow would occur. Instead, the input data buffering block uses a triple buffer mounted as a circular buffer.

Figure 198. ISS ISP RSZ-A/RSZ-B Input Data Buffering



The size of the buffer line is the maximum of the RSZ_GNC[12:0] RSZA_MEM_LINE_SIZE and RSZ_GNC[28:16] RSZB_MEM_LINE_SIZE bit fields.

3.3.4.5.3.2.4 ISS ISP RSZ Overflow Detection

Overflow can happen when a new line is to be written into the circular buffer but the destination is not empty (all data has not been read out). This event can be also triggered in the same scenario but at the RSZ master output interface FIFO. In this case the RSZ output data stall signal is asserted.

The RSZ_FIFO_OVF event is triggered in case of overflow.

3.3.4.5.3.2.5 ISS ISP RSZ Input Data Stalling

The input data buffering module generates a stall signal to prevent input FIFO overflow.

- The rsz_stall_input signal is asserted when the sum of the data in the input buffer FIFOs is greater than or equal to a programmable FIFO threshold.
  - The threshold is set up by the RSZ_IN_FIFO_CTRL[12:0] THRLD_HIGH bit field.
- The rsz_stall_input signal stays high as long as the sum of the data in the input buffer FIFOs is greater than or equal to another programmable FIFO threshold.
  - The threshold is set up by the RSZ_IN_FIFO_CTRL[28:16] THRLD_LOW bit field.

The assertion of the rsz_stall_input signal does not ensure that no more data will be sent.

3.3.4.5.4 ISS ISP RSZ Data Requestor Module

The data requestor module reads the data from the circular buffer, synchronizes the resizer engines, controls the vertical buffer switch, and updates the circular buffer counters.

Reads are asynchronous from writes. The data requestor module must be able to request pixels at a data rate of 200 MPix/s with pixels coded on 16 bits.

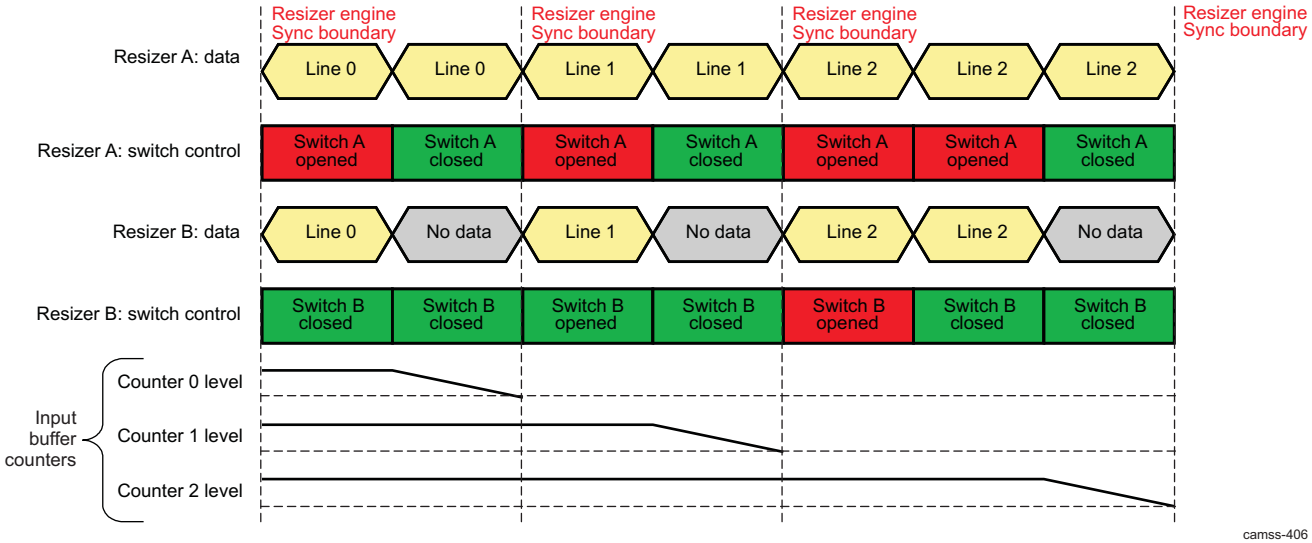
The two resizer engines are synchronized on input lines boundaries, regardless of their setting (downscale/downscale, upscale/upscale, or upscale/downscale). It means that the slowest resizer engine determines the performance of the second resizer. The synchronization boundaries are highlighted in Figure 199 (vertical lines).

- The data can be read and forwarded by the data requestor module to both resizers. That is typically the case when the resizers are configured in downscale/downscale mode.
- The data can be read and forwarded by the data request module to one resizer only. In this case the second resizer is stalled. That is typically the case when the resizers are configured in upscale/upscale or upscale/downscale modes.

The data requestor must decrease the input buffer counters only when a given line is no longer needed. For example, if a line needs to be used M times, the counter decrements the line counter only the last time the line is needed. For every input line the vertical resizers send the data requestor the number of output lines that will be output for the current line.

Figure 199 shows the data requestor behavior. Resizer A requests 2, 2, and 3 lines, and resizer B requests 1, 1, and 2 lines. Obviously, resizer A stalls resizer B.

Figure 199. ISS ISP RSZ Data Requestor Module Behavior Example



3.3.4.5.5 ISS ISP RSZ Averager

3.3.4.5.5.1 ISS ISP RSZ Use Cases

The two RSZ engines can have independent averager settings:

- Both resizers can use it.
- One resizer can bypass it and the other can use it.
- Both resizers can bypass it.

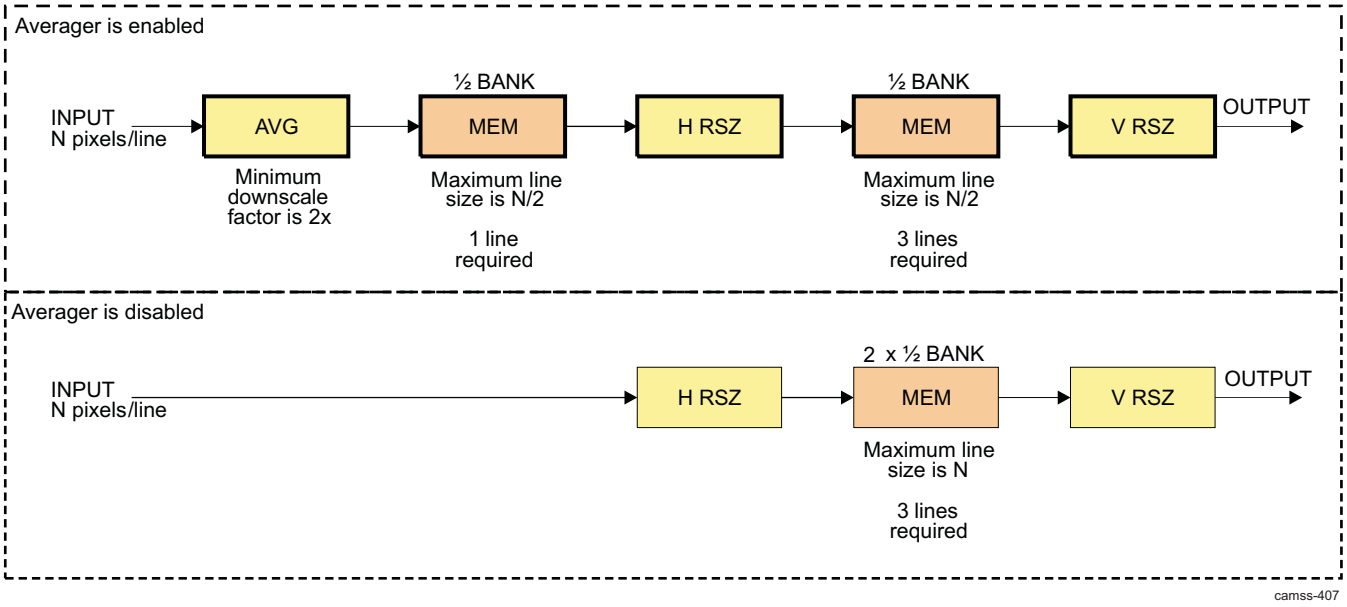
3.3.4.5.5.2 ISS ISP RSZ Memory Utilization

Vertical averaging requires memory to perform pixel data accumulation. It shares the vertical memory lines that are used for vertical filtering: this is the reason why the vertical memory lines are organized as two banks of half lines.

- The averagers output lines that are at most half the size of the input image in one memory bank.
- The horizontal resizers write their output data in the second memory banks.

Figure 200 shows the use of memory when the averager is enabled or disabled.

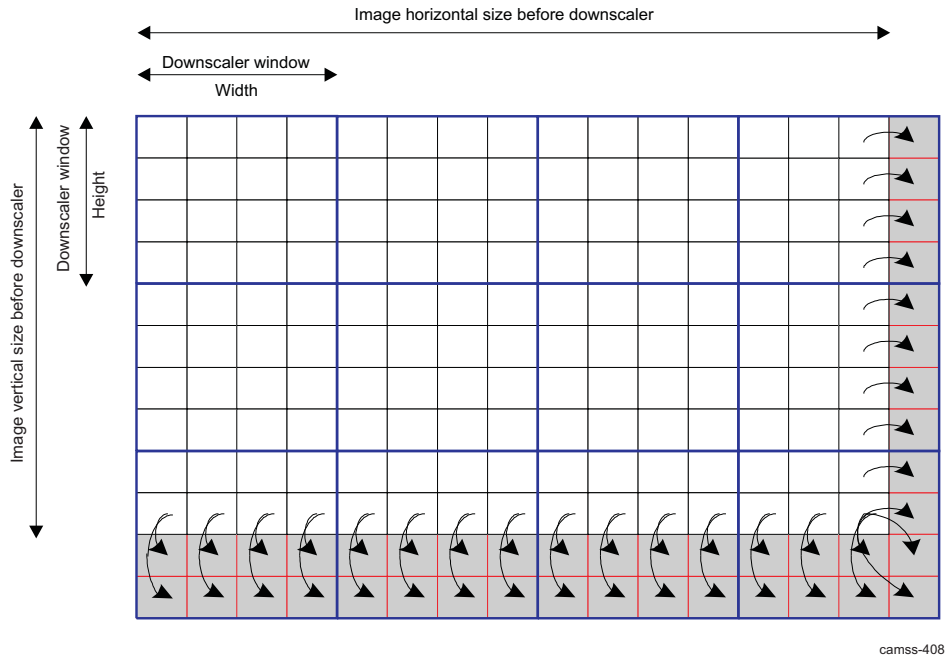
Figure 200. ISS ISP RSZ Averager Memory Utilization



3.3.4.5.5.3 ISS ISP RSZ Border Conditions

Figure 201 shows the averager behavior for border conditions. If the input image is not big enough, border duplication must occur.

Figure 201. ISS ISP RSZ Averager Border Conditions



Downscaling is enabled from the RZi_DWN_EN register. Moreover, the vertical averaging size is set by the RZA_DWN_AV[3:5] V bit field. The actual downscale ratio is given by $1/2^{(RZA\_DWN\_AV[3:5]\ V\ +1)}$. The range is from 1/2 to 1/256 in power of 2. The horizontal averaging size is set by the RZA_DWN_AV[0:2] H bit field. The actual downscale ratio is given by $1/2^{(RZA\_DWN\_AV[3:5]\ H\ +1)}$. The equations are the same for RSZ-B. The range goes from 1/2 to 1/256 in power of 2.

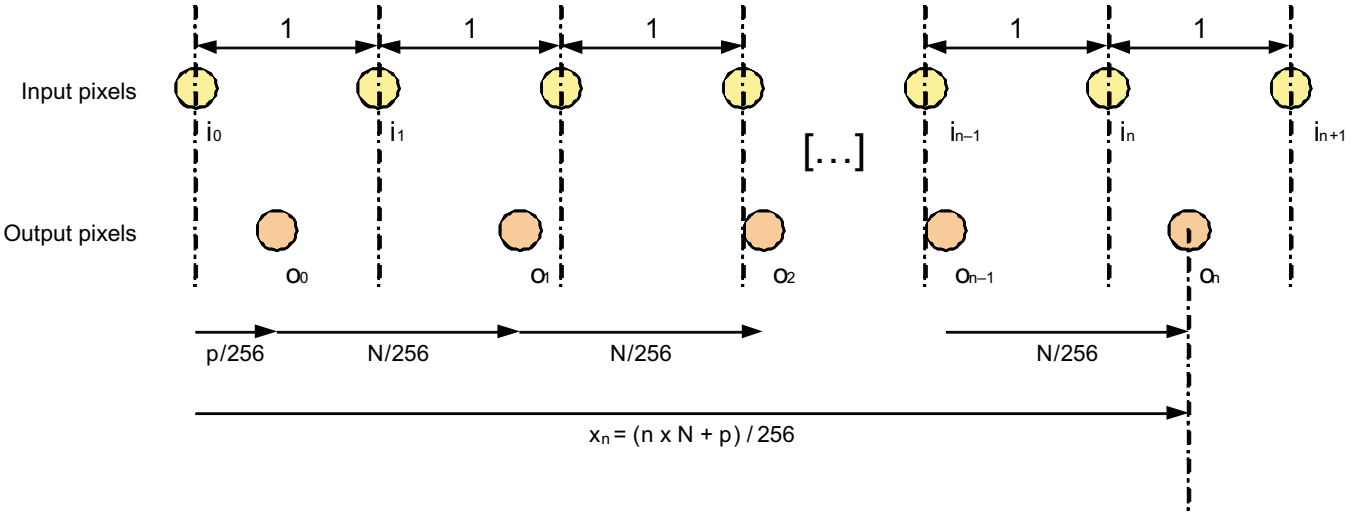
3.3.4.5.6 ISS ISP RSZ Interpolation

Figure 202 shows the basic interpolation method used in the RESIZER module. The following assumptions are made:

- The distance between each input pixel is 1.
- The magnification ration is given by 256/N and p/256 is the initial phase of the output data.

The output pixels are also evenly spaced. The distance between each output pixel is given by N/256. In the example in Figure 202, N is greater than 256. The position of the n^{th} output pixel is given by (n x N + p) / 256.

Figure 202. ISS ISP RSZ Basic Interpolation Method



Assuming the intensity of input pixels is i_0, i_1, i_2, etc. and the resized (output) pixels are o_0, o_1, o_2, etc., the n^{th} output pixel (o_n) is determined using the nearest 4 input pixels as follows:

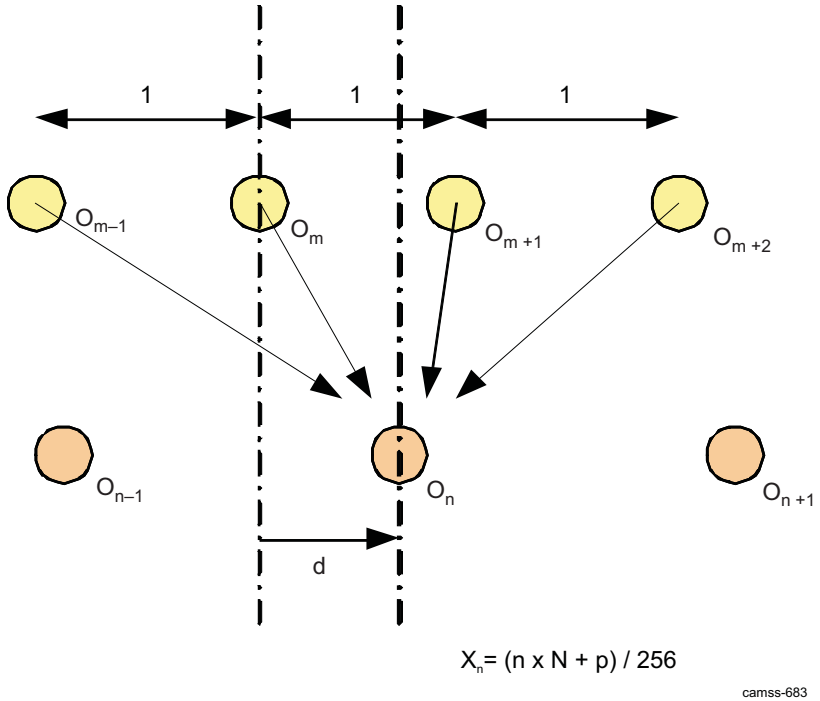
$o_n = h(1+d) \times i_1 + h(d) \times i_m + h(d-1) \times i_{m+1} + h(d-2) \times i_{m+2}$

In the previous equation h(x) is the interpolation main function. The RESIZER module supports linear and bicubic convolution interpolation functions.

Figure 203 shows the interpolation principle at the n^{th} output pixel (on) at position xn. Furthermore, the m and d parameters are as follows:

m = floor((n x N + p) / 256) and d = ((n x N + p) / 256) – m

Figure 203. ISS ISP RSZ Interpolation Filtering



For each resizer (RSZ-A or RSZ-B), and for chrominance and luminance, the interpolation method for vertical interpolation can be a 2-tap linear interpolation or a 4-tap cubic convolution (default) method. The choice is made in RZi_V_TYP[0] Y for luminance and RZi_V_TYP[1] C for chrominance, where i = A or B and is the resizer number. It is similar for horizontal interpolation from RZi_H_TYP[0] Y for luminance and RZi_H_TYP[1] C for chrominance, where i = A or B and is the resizer number.

3.3.4.5.6.1 ISS ISP RSZ Liner Interpolation Input Data

Before data interpolation, a low pass filtering (LPF) operation is required on the input data. The following equation gives the LPF function. The equation is evaluated at pixel position d_i but neighbor pixels d_{i-1} and d_{i+1} are required. The gain value g is set up by the register RZi_V_LPF and RZi_H_LPF. Different gains are possible horizontally and vertically as well as for Luma and Chroma.

$LPF_g (d_{i-1}, d_i, d_{i+1}) = d_i + g \times (d_{i-1} - 2d_{i+1} + d_{i+2})/128$

3.3.4.5.6.1.1 ISS ISP RSZ Cubic Convolution Mode

The input data is not modified in bicubic mode. Basically, the input is equal to the output.

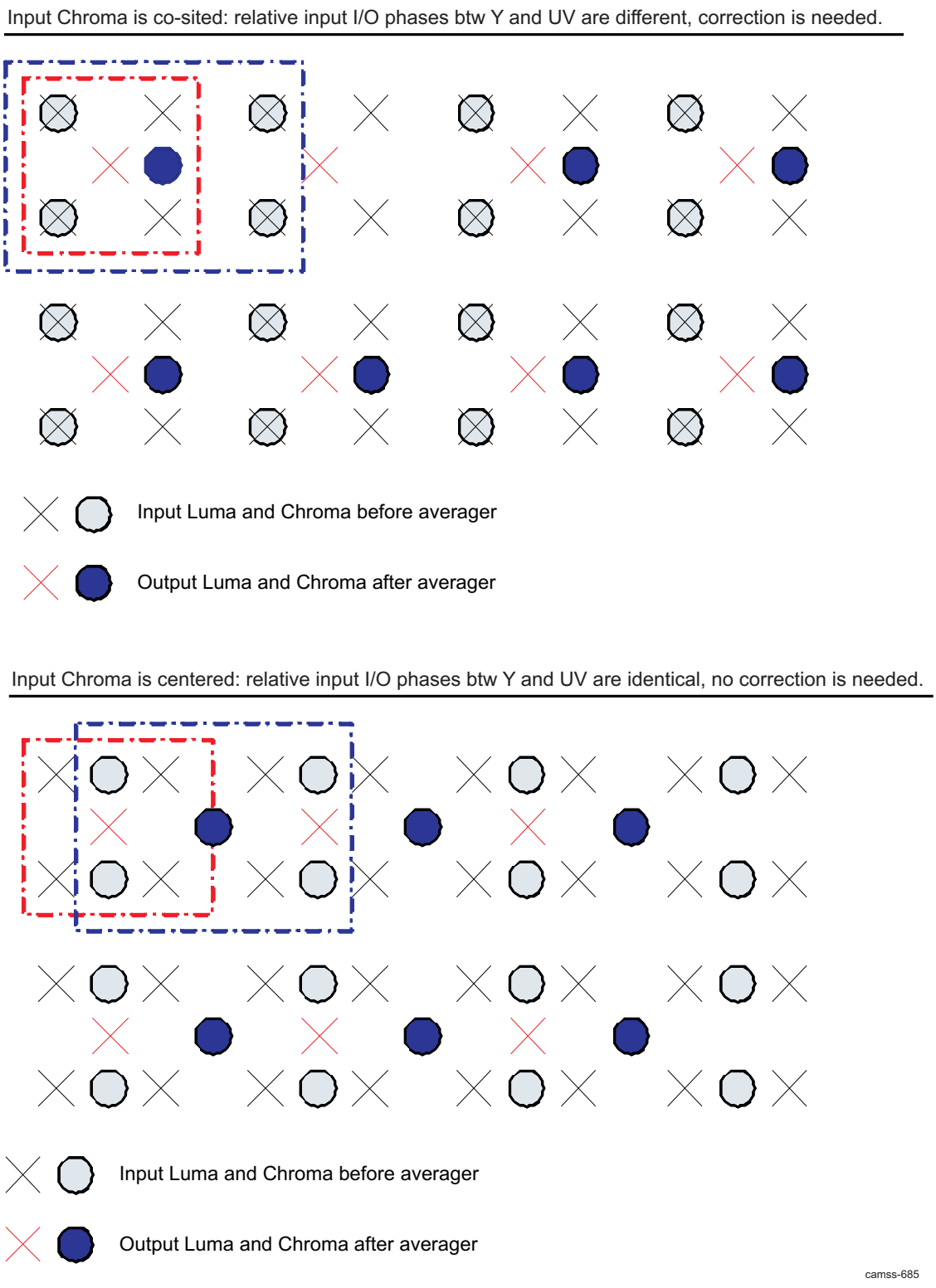
3.3.4.5.6.1.2 ISS ISP RSZ Phase Settings

The initial value for the phase value for vertical resizing is set by the RZi_V_PHS_Y for luminance and RZi_V_PHS_C for chrominance. These values are in the U14Q8 fractional format (values in the range [0 – 63.996]). When YUV4:2:2 data are output, the phase value for Luma and Chroma must be aligned; that is, RZi_V_PHS_Y = RZi_V_PHS_C.

The following constraint equation applies: | RZi_V_PHS_Y - RZx_V_PHS_C | = RZi_V_DIF. This constraint means that at most the distance between the initial phases for luminance and chrominance is not expected to exceed the distance between two Luma pixels. The absolute value is used; therefore, the initial Luma phase can be greater than the initial Chroma phase or vice versa. As a reminder, the distance between two output pixels for Luma is given by RZi_V_DIF.

The initial value for the phase value for horizontal resizing is set by the RZi_H_PHS bit field. The RZi_H_PHS_ADJ register enables adjusting the horizontal phase for the Luma component when averaging is enabled (the averager disrupts the relative sampling point between luminance and chrominance when YUV4:2:2 cosited data is input). The relative phase between Luma and Chroma is different before and after the horizontal averager. The vertical phase is not affected by the averager. Figure 204 shows the effect of the averager on the phases. RZi_H_PHS_ADJ is expected to be equal to zero if the averager is disabled.

Figure 204. ISS ISP RSZ-A/RSZ-B Phase Averager Effect



3.3.4.5.7 ISS ISP RSZ Data Saturator

After vertical rescaling and before color conversion, the output data is saturated (clipped) to programmable values that are given by the following registers:

- RSZ_YUV_Y_MIN
- RSZ_YUV_Y_MAX
- RSZ_YUV_C_MIN
- RSZ_YUV_C_MAX

The maximum Y value is set up with the RSZ_YUV_Y_MAX register. If the Input Y value is greater than the MAX value, it is clipped to MAX.

The minimum Y value is set up with the RSZ_YUV_Y_MIN register. If the Input Y value is smaller than the MIN value, it is clipped to MIN.

The maximum Cb/Cr value is set up with the RSZ_YUV_C_MAX register. If the Input Cb/Cr value is greater than the MAX value, it is clipped to MAX.

The minimum Cb/Cr value is set up with the RSZ_YUV_C_MIN register . If the Input Cb/Cr value is smaller than the MIN value, it is clipped to MIN.

3.3.4.5.8 ISS ISP RSZ Color Convertor

As mentioned previously, the resizer can support RAW, YUV4:2:0, and YUV4:2:2 formats. The resizer engines can also support RGB output: RGB5:6:5 and ARGB32.

The RGB5:6:5 data is 16 bits wide and consists of 5 bits for red, 6 bits for green, and 5 bits for blue.

The following table shows the way RGB5:6:5 is stored to memory. This data format is compatible with the display controller. Only the little-endian memory representation is supported.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R1					G1						B1					R0					G0							B0				

The ARGB32 data is 32 bits wide and consists of 8 bits for alpha, 8 bits for red, 8 bits for green, and 8 bits for blue. The alpha value is global and is set for the entire frame; registers control the alpha value: the RZx_RGB_BLD register controls the alpha values of resizer A and resizer B.

The following table shows the way ARGB32 is stored to memory. This data format is compatible with the display controller. This representation is endiannism invariant: it is the same for little endian and big endian.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A								R								G								B							

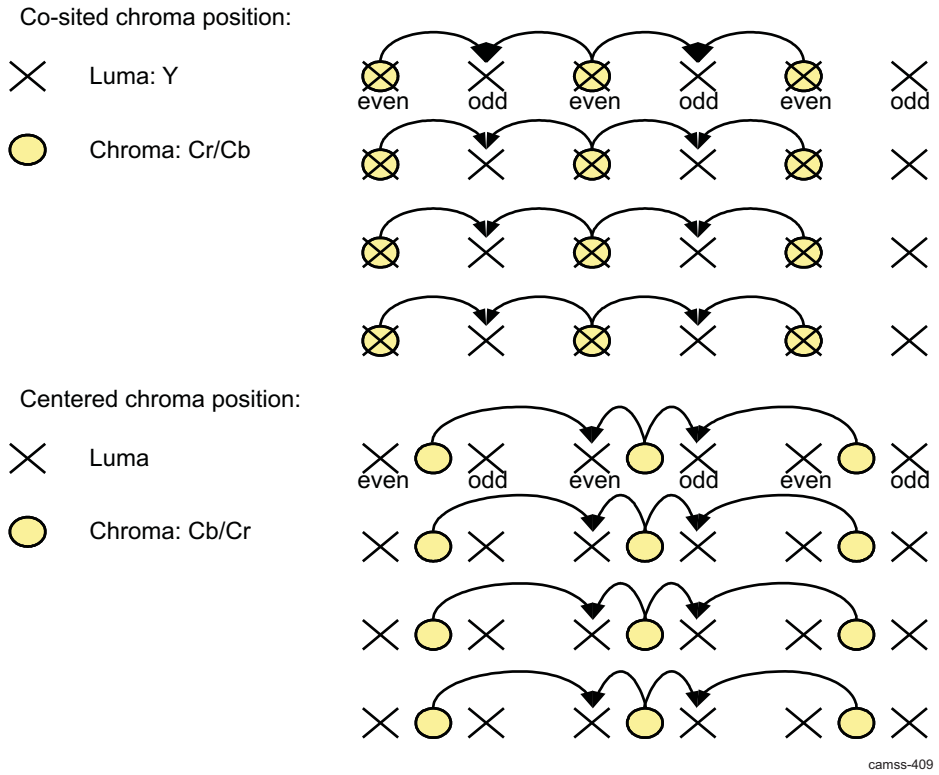
The RGB output is enabled by setting the RZx_RGB_EN[0] RGB_EN bit to 1.

The RGB format is set by the RZx_RGB_TYP[0] TYP bit (0 for ARGB32 format and 1 for RGB5:6:5).

To handle the horizontal border conditions, the leftmost Chroma sample or the right-most Chroma sample is duplicated on the left or the right.

Software must make it possible to remove 2 pixels on the left and/or right to take care of the issues that Chroma duplication introduces on the borders. The RZx_RGB_TYP[1] MSK0 and RZx_RGB_TYP[2] MSK1 bits control this feature.

Figure 205. ISS ISP RSZ Chroma Position and Upsampling



3.3.4.5.9 ISS ISP RSZ Output Interface

The output interface receives the data generated by the two resizer engines and generates the addresses and the port requests to the BL module.

- The port 1 interface is dedicated to the RSZ-A module. If the RSZ module is set up in pass-through mode, then the data is output on the port 1 interface. This interface can transfer RAW, YUV4:2:2, YUV4:2:0 and RGB data
- The port 2 interface is dedicated to the RSZ-B module. This interface can transfer RAW, YUV4:2:2, YUV4:2:0 and RGB data.

The YUV4:2:0 data format is handled differently from the other formats because the output data are written at two different memory locations: luminance in one buffer and chrominance in a second buffer. For all other formats the data are written in the same buffer.

Each data format must be stored in memory in a dedicated manner, which is summarized in Table 185.

Table 185. ISS ISP RSZ Output Interface: Data Formats

Output Format	Bytes per Pixel	Output Buffers per Image	Interface Supporting the Data Format
RAW	2	1	MTC port 1
YUV4:2:2	2 average	1	MTC port 1 port 2
YUV4:2:0	1.5 average	2	MTC port 1 port 2
RGB16	2	1	MTC port 1 port 2
ARGB32	4	1	MTC port 1 port 2

3.3.4.5.9.1 ISS ISP RSZ Circular Buffer

Figure 206 shows the parameters that are required to set up the circular buffers. As mentioned previously, there can be up to four circular buffers in case the two resizer engines are outputting YUV4:2:0 data.

The circular buffer management requires the following parameters (REZ-A or RSZ-B A or B ,chrominance or luminance Y or C, low or high part of the address, L or H. Sets the base address of the circular buffer):

- Baseline address (BAD, in registers, where x is the resizer A or B, and i is Y or C)
- Start address (SAD, in RZx_SDR_i_SAD_j registers)
- Start pointer (PTR_S, in RZx_SDR_i_PTR_S registers)
- End pointer (PTR_E, in RZx_SDR_i_PTR_E registers)
- Line offset (OFT, in RZx_SDR_i_OFT registers)

Circular Buffer Parameter	Register (for RSZ-A and RSZ-B)	Description
Baseline address	RZA_SDR_Y_BAD_H RZA_SDR_Y_BAD_L RZA_SDR_C_BAD_H RZA_SDR_C_BAD_L RZB_SDR_Y_BAD_H RZB_SDR_Y_BAD_L RZB_SDR_C_BAD_H RZB_SDR_C_BAD_L	Sets the base address of the circular buffer
Start address	RZA_SDR_Y_SAD_H RZA_SDR_Y_SAD_L RZA_SDR_C_SAD_H RZA_SDR_C_SAD_L RZB_SDR_Y_SAD_H RZB_SDR_Y_SAD_L RZB_SDR_C_SAD_H RZB_SDR_C_SAD_L	Sets the start address of the circular buffer. The first data output is written to this address. If the first line of a frame must be written at the beginning of the circular buffer memory, then SAD = BAD and PTR_S = 0.
Start pointer	RZA_SDR_Y_PTR_S RZA_SDR_C_PTR_S RZB_SDR_Y_PTR_S RZB_SDR_C_PTR_S	Sets the initial value of the circular buffer internal counter. It must be set up as PTR_S = (SAD - BAD)/OFT. PTR_S is expressed in the number of lines.
End pointer	RZA_SDR_Y_PTR_E RZA_SDR_C_PTR_E RZB_SDR_Y_PTR_E RZB_SDR_C_PTR_E	Sets the size of the circular buffer. PTR_E is expressed in the number of lines. The circular buffer can contain up to PTR_E lines.
Line offset	RZA_SDR_Y_OFT RZA_SDR_C_OFT RZB_SDR_Y_OFT RZB_SDR_C_OFT	This is the offset expressed in bytes between two lines in the circular buffer. Here: Line 0 = SAD, Line 1 = SAD + 1 x OFT, Line 2 = SAD + 2 x OFT, etc. OFT does not necessarily correspond to the size of a line in a frame; it can be bigger.

More generally, the following equations hold:

- SAD = BAD + (PTR_S x OFT) and PTR_S PTR_E

Interrupts can be triggered every time a certain number of lines are written to the circular buffer. There are independent settings for each resizer and for each possible output of each resizer.

Figure 206. ISS ISP RSZ and Circular Buffer Settings

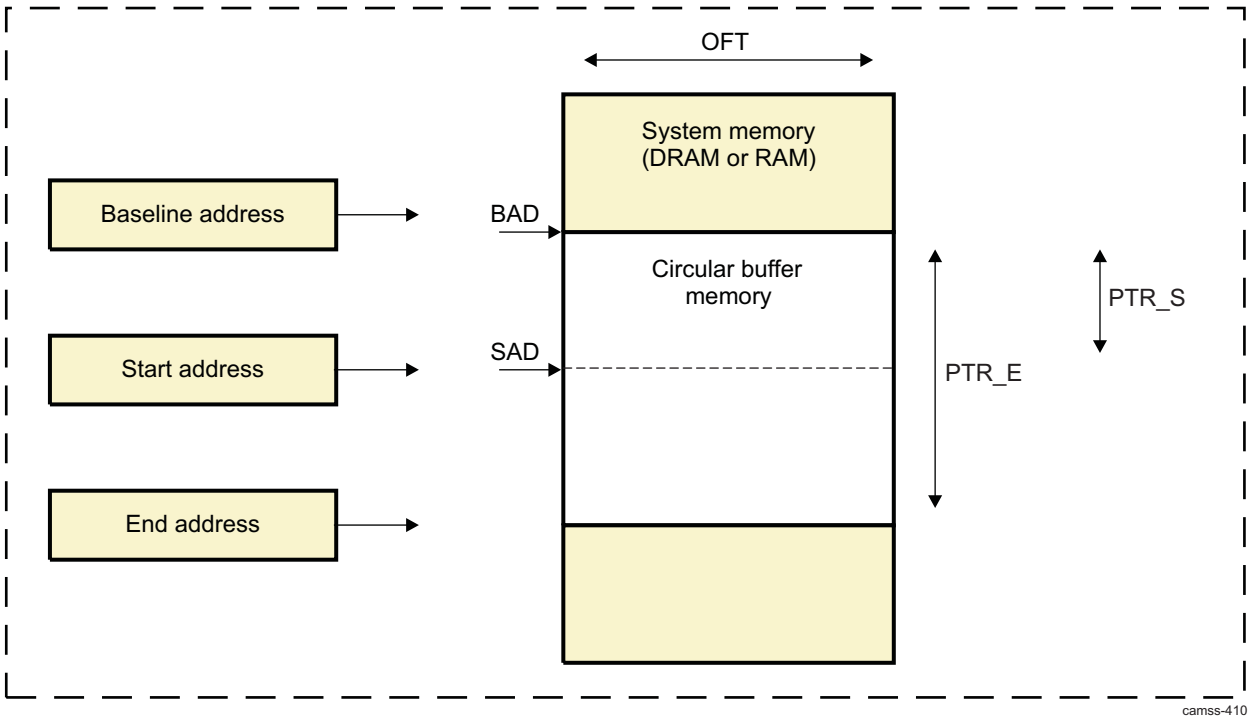


Figure 207 shows how the data are stored in the circular buffer over time when vertical flip is disabled. In this example, PTR_S = 5 and PTR_E = 7. There can be up to PTR_E = 7 lines in the circular buffer.

Figure 207. ISS ISP RSZ and Circular Buffer Settings – Example 1

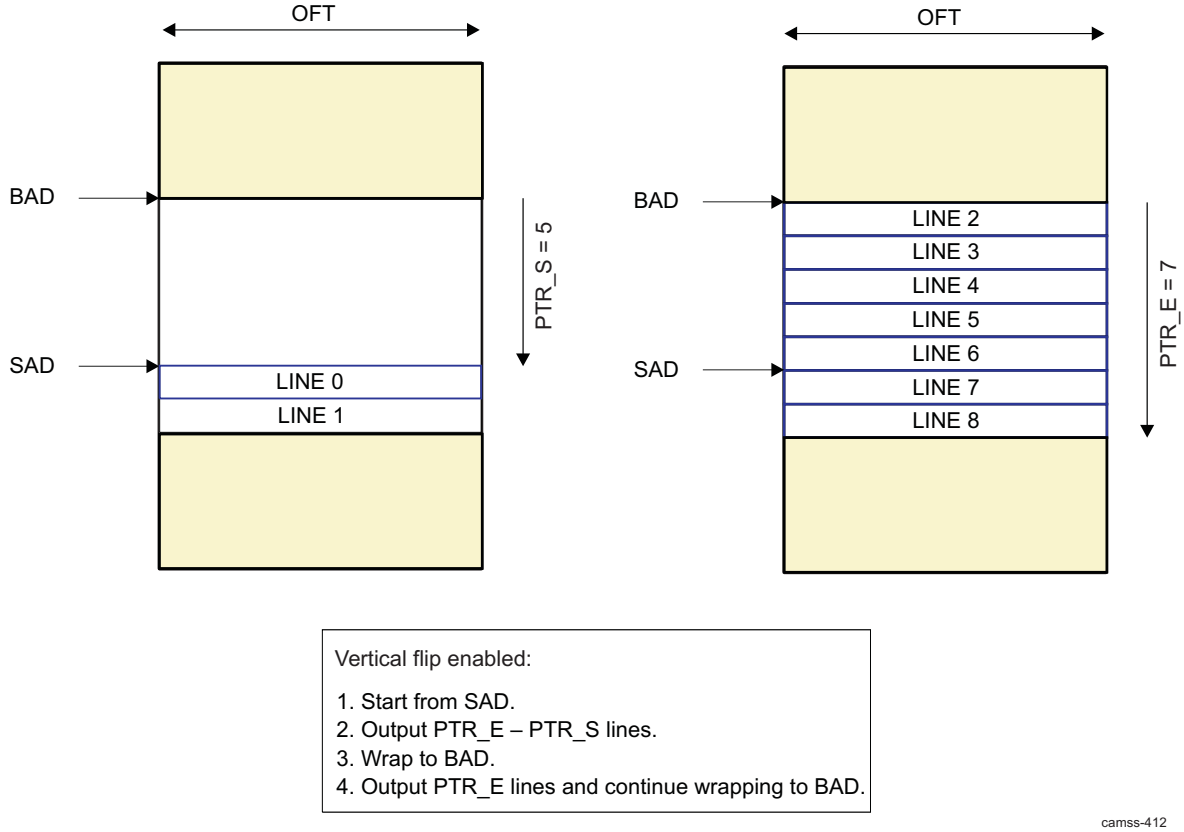
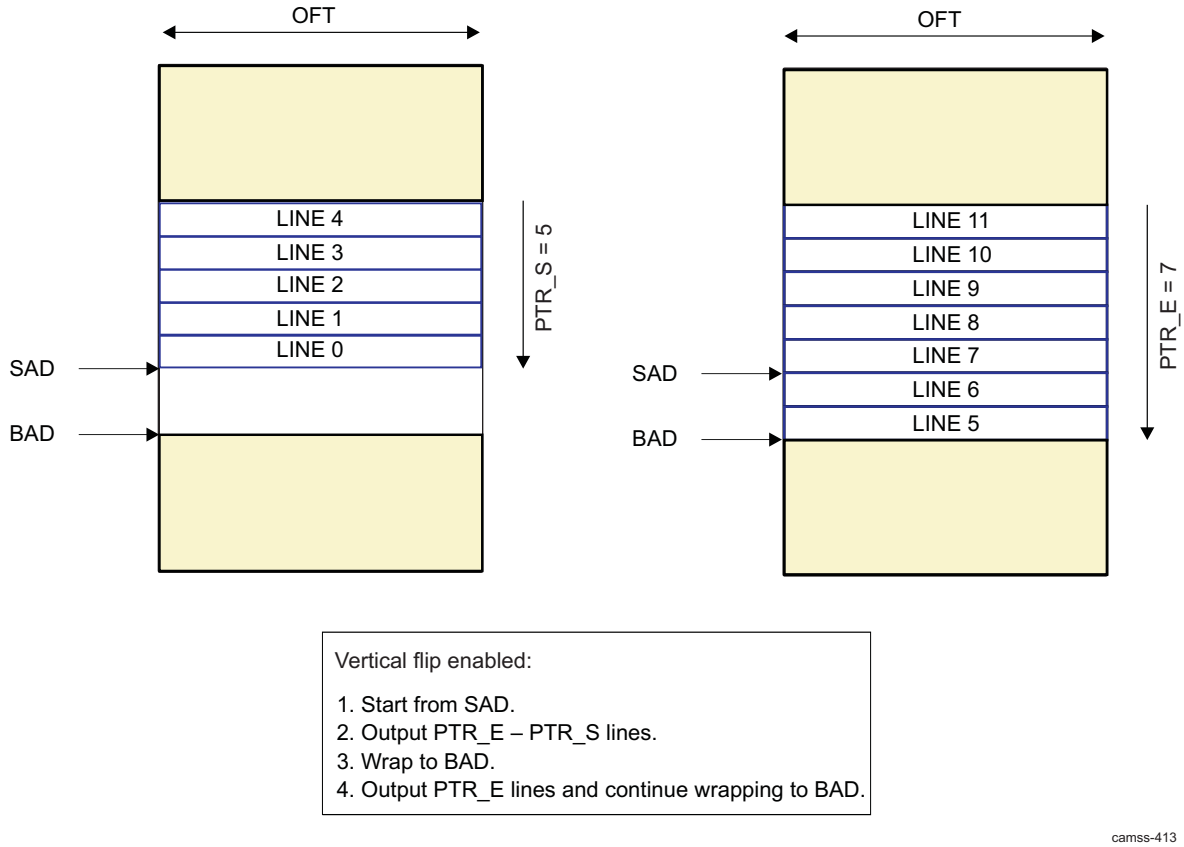


Figure 208 shows how the data are stored in the circular buffer over time when vertical flip is enabled. In this example, PTR_S = 5 and PTR_E = 7. There can be up to PTR_E = 7 lines in the circular buffer.

Figure 208. ISS ISP RSZ and Circular Buffer Settings – Example 2



3.3.5 ISS ISP H3A Functional Description

3.3.5.1 ISS ISP H3A Overview

The H3A module supports the control loops for autofocus, auto white balance, and auto exposure by collecting metrics about the imaging/video data. The metrics are used to adjust parameters for processing the imaging/video data. There are two main blocks in the H3A module:

- Autofocus (AF) engine:

  The AF submodule extracts and filters the red, green, and blue data from input image data and provides the accumulation or peaks of the data in a specified region. The specified region is a 2D block of data referred to as a paxel. The AF engine supports the following features:

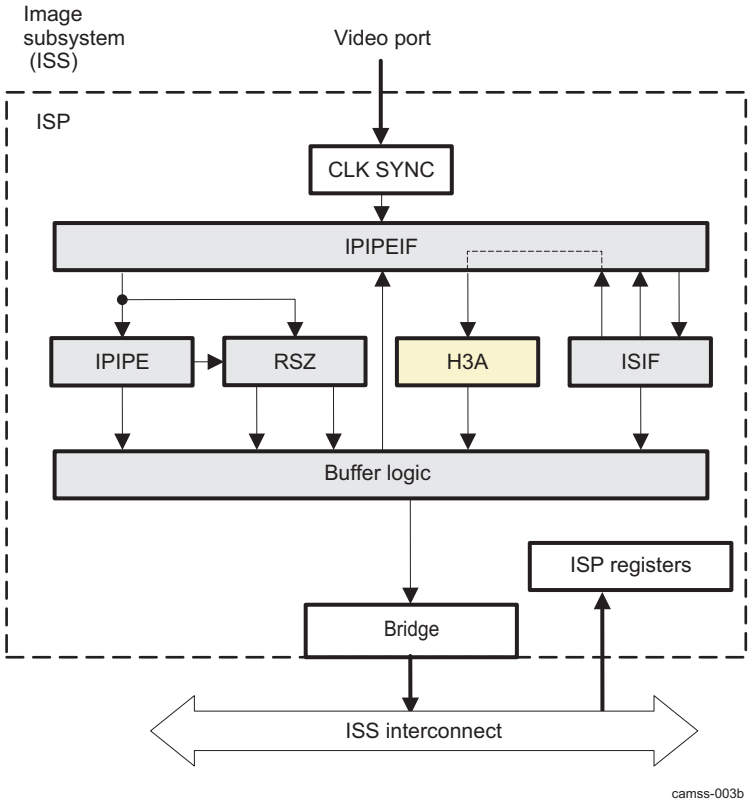
  - Peak mode in a paxel
    - Accumulation of the maximum focus value (FV) of each line in a paxel
  - Accumulation mode in a paxel
  - Accumulation of horizontal and vertical focus value in a paxel
  - Up to 12 paxels in the horizontal direction and up to 12 paxels in the vertical direction with vertical focus
  - Up to 36 paxels in the horizontal direction and up to 128 paxels in the vertical direction with horizontal focus only
  - Programmable width and height for the paxel/window
  - Programmable red, green, and blue position within a 2 × 2 matrix
  - Separate horizontal start for paxel and filtering
  - Programmable vertical and horizontal line increments within a paxel
  - Horizontal FV uses parallel infinite impulse response (IIR) filters configured in a dual-biquad configuration with individual coefficients (two filters with 11 coefficients each). The filters are intended to compute the sharpness/peaks in the frame to focus on.
  - Vertical FV uses a 5-tap FIR filter with 8-bit coefficients. With horizontal steps each paxel has up to 32 columns to be maintained for vertical FV calculation.
- Auto exposure and auto white balance (AE/AWB) engine:

  The AE/AWB engine accumulates values and checks for saturated values in a subsampling of the video data. In the case of the AE/AWB, the 2D block of data is referred to as a window. Thus, other than having different names, paxels and windows are essentially the same. However, the numbers, dimensions, and starting positions of AF paxels and AE/AWB windows are programmable separately. AE/AWB supports the following features:

  - Accumulate clipped pixels along with all nonsaturated pixels in each window per color
  - Accumulate the sum of squared pixels in each window per color
  - Minimum and maximum pixel values in each window per color
  - Supports for up to 36 horizontal windows with sum + { sum_sq or min+max} output
  - Support for up to 56 horizontal windows with sum output
  - Support for up to 128 vertical windows
  - Programmable width and height for the windows. All windows in the frame are the same size.
  - Separate vertical start coordinate and height for a black row of paxels that is different than the remaining color paxels
  - Programmable horizontal sampling points in a window
  - Programmable vertical sampling points in a window
- Maximum pixel throughput of 200 MPix/s
- Double-buffer for paxel/window accumulation
- H3A data path is 10 bits.
- Maximum input size is 3008 pixels.

Figure 209 shows the H3A module connections to other submodules of the ISP.

Figure 209. ISS ISP H3A High-Level Diagram



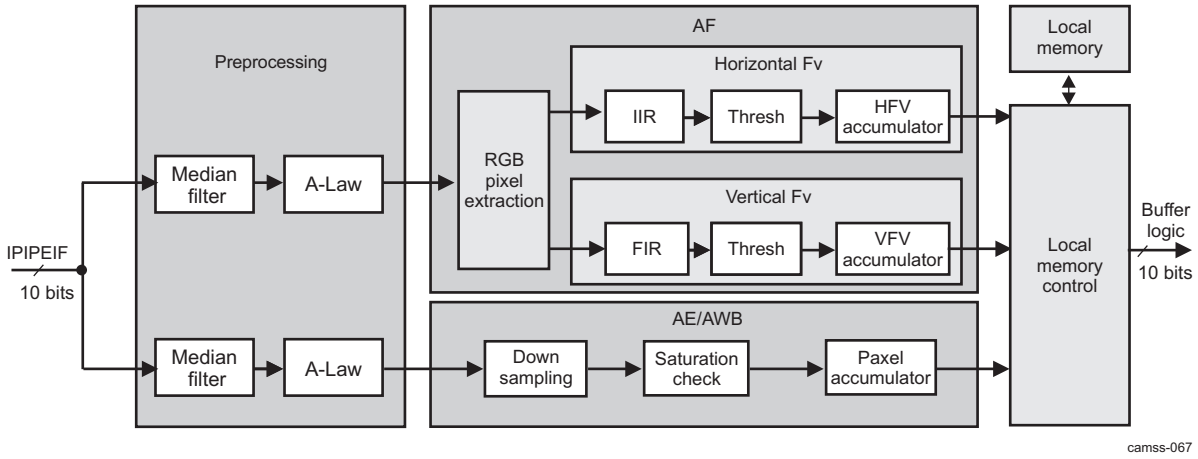
3.3.5.2 ISS ISP H3A Top-Level Block Diagram

The block diagram in Figure 210 shows the process of the AF and AE/AWB data paths through the H3A module.

The data flow before H3A is:

1. Data comes from the VP (VP) or BL.
2. The data is processed by the ISIF.
3. The data is processed by the IPIPEIF.
4. The data is 10 bits from the IPIPEIF at H3A input.

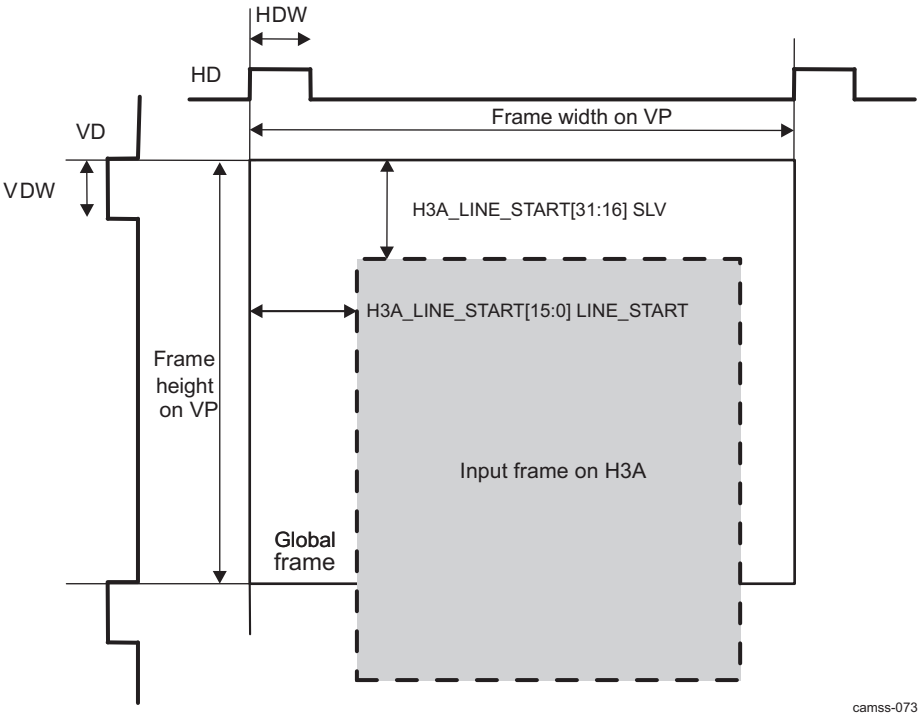
Figure 210. ISS ISP H3A Top-Level Block Diagram



3.3.5.3 ISS ISP H3A Line Framing Logic

In certain cases the number of clock cycles between HD pulses is greater than the line buffer included in the H3A. To solve this problem a framing module was added before the line buffer. The framing module uses the H3A_LINE_START register to find the position of the first pixel to place into the line buffer. All other registers reference this point as the 0 pixel for their start positions. The line size is 3008 pixels. After 3008 clock cycles the framing logic disables the line buffer and waits until the next HD. If the next HD comes before 3008 clock cycles, then the active region ends immediately and the counter waits for the H3A_LINE_START register count to be reached again. For the vertical position the H3A_LINE_START[31:16] SLV bit field can be used to determine where the start point of the frame is relative to the rising edge of VD. This logic allows for an active frame to cross VD boundaries and remain in the same frame.

Figure 211. ISS ISP H3A Frame Format Settings



NOTE: (Frame width on VP) - (H3A_LINE_START[15:0] LINE_START) must be less than or equal to 3008, because the H3A memory lines are limited to 3008 pixels.

3.3.5.4 ISS ISP H3A Optional Preprocessing

The input to the H3A module is 10-bit RAW data from the IPIPEIF. A 10-bit to 8-bit A-Law compression step can be enabled and disabled separately for the AF engine (the H3A_PCR[1] AF_ALAW_EN bit) and the AE/AWB engine (the H3A_PCR[17] AEW_ALAW_EN bit). A-Law compression offers added protection against overflowing the accumulators.

If the A-Law table is enabled, the output is 10 bits, with the upper two bits filled with 0.

For the AF process, a horizontal median filter can be enabled and disabled (the H3A_PCR[2] AF_MED_EN bit) before A-Law compression. This filter is useful for reducing temperature-induced noise. The horizontal median filter calculates the absolute difference between the current pixel (i) and pixel (i – 2), and between the current pixel (i) and pixel (i + 2). If the absolute difference exceeds a threshold, and the sign of the differences is the same, the average of pixel (i – 2) and pixel (i + 2) replaces pixel (i). The threshold of the horizontal median filter can be set in the H3A_PCR[10:3] MED_TH bit field.

3.3.5.5 ISS ISP H3A Autofocus Engine

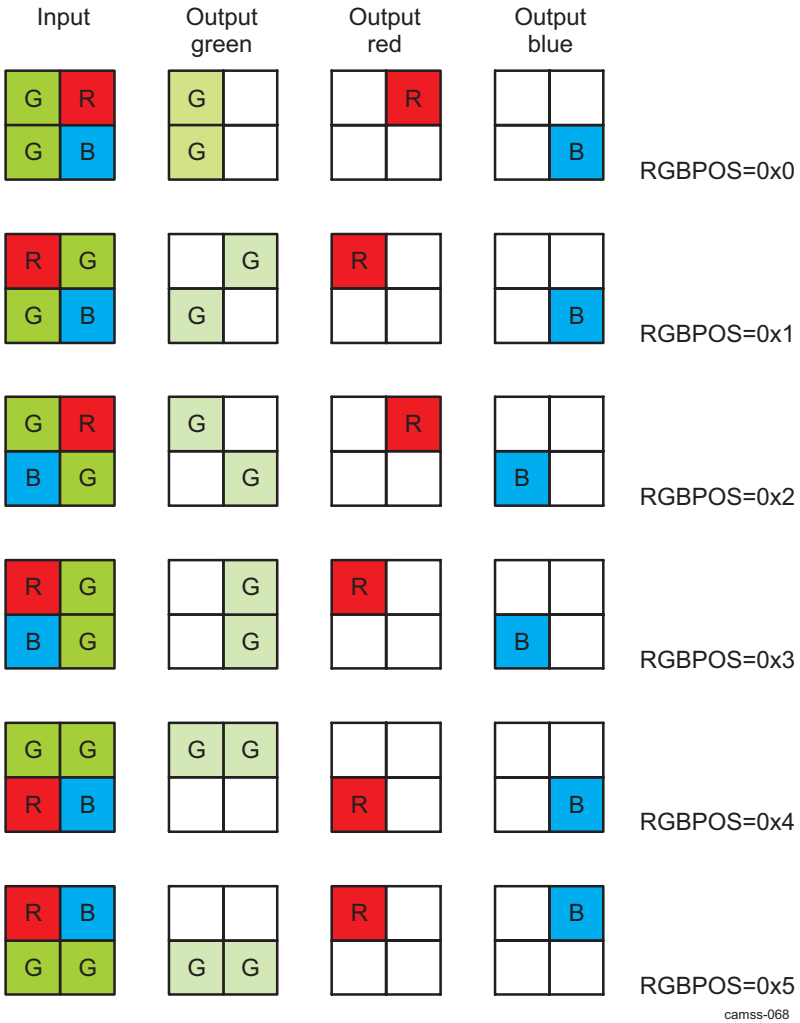
The AF engine works by extracting each green (Gr or Gb) pixel from the video stream and subtracts a fixed offset of 128 or 512 (depending of whether A-Law is enabled or disabled) from the pixel value. The offset value is then passed through an IIR filter and the absolute value of the filter output is the focus value (FV). Both FV and FV^2 are produced. The FV and FV^2 values can be accumulated or the maximum for each line/column can be accumulated. The following sections describe this process in more detail.

3.3.5.5.1 ISS ISP H3A Paxel Extraction

From the paxel starting coordinate (the H3A_AFPAXSTART[27:16] PAXSH and H3A_AFPAXSTART[11:0] PAXSV bit fields) specifies the starting point of the paxel grid, with respect to first pixel of th input image frame.

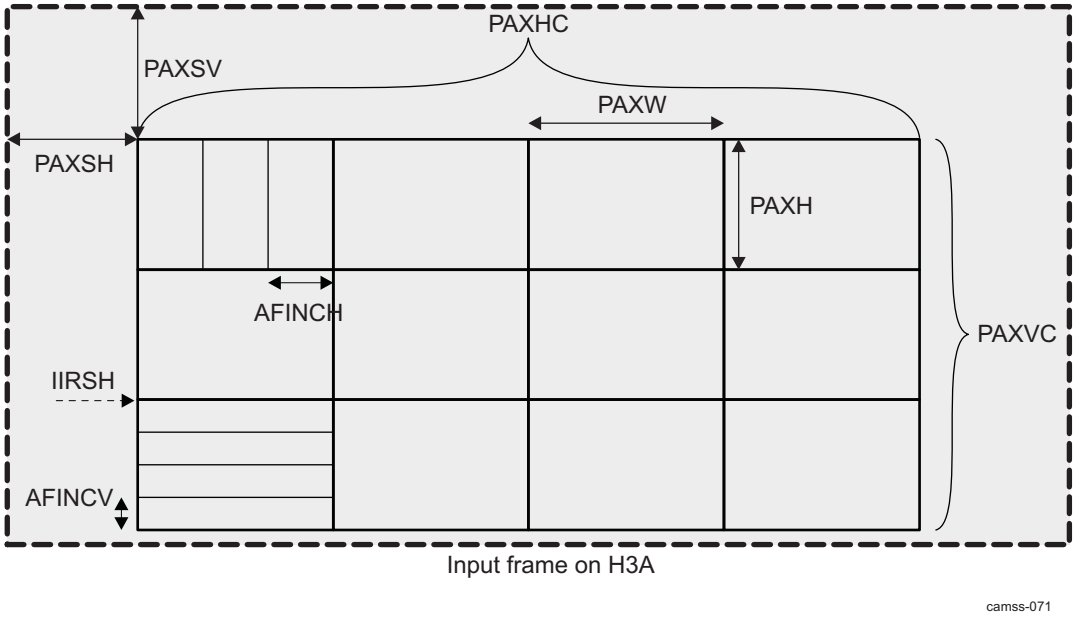
The paxel starting coordinate also indicates which color pixels are extracted if VF is enabled (that is, if H3A_PCR[20] AF_VF_EN = 1). Normally, either Gr or Gb is used for AF, but it is not important to the hardware whether it is red, green, or blue. If VF is not enabled, then the red, green, and blue pixel extraction is controlled by the H3A_PCR [13:11] RGBPOS bit field to extract the correct colors from the input stream. Figure 213 shows the available options for this bit field. The red and blue pixel positions are interchangeable. For each 2 × 2 grid, the green pixels are summed to create a single value. Because of this, the amplitude of the green output contains 2 pixels, while the red and blue outputs each contain 1 pixel.

Figure 212. ISS ISP H3A Red, Green, and Blue Pixel Extraction Examples



Each paxel is H3A_AFPAX1[23:16] PAXW × H3A_AFPAX1[7:0] PAXH (width × height) pixels. Inside each paxel, horizontal FV can skip lines, operating on one every H3A_AFPAX2[16:13] AFINCV lines. Vertical FV can skip columns, operating on one every H3A_AFPAX2[20:17] AFINCH columns. Up to 32 columns are supported for each paxel. If floor (PAXW/AFINCH) = 32, only the first 32 designated columns are operated on. Because PAXW, PAXH, AFINCV, and AFINCH are all even numbers, AF always operates on the same green color, Gr or Gb. IIR filters for the horizontal FVs start operation at column H3A_AFIIRSH[11:0] IIRSH.

Figure 213. ISS ISP H3A Horizontal/Vertical FV Paxel Configuration



NOTE: (H3A_AFPAXSTART [27:16] PAXSH) + (H3A_AFPAX2 [5:0] PAXHC) × (H3A_AFPAX1[23:16] PAXW) = [(Frame width on VP) – (H3A_LINE_START[15:0] LINE_START)] = 3008

Table 186 lists the register fields that configure the size and number of paxels.

Table 186. ISS ISP H3A Paxel Register Field Descriptions

Register Field	Bit Width	Description
H3A_AFPAX1[23:16] PAXW	8	Paxel width (in pixels)
H3A_AFPAX1[7:0] PAXH	8	Paxel height (in lines)
H3A_AFPAX2[5:0] PAXHC	6	Paxel count for horizontal direction
H3A_AFPAX2[12:6] PAXVC	7	Paxel count for vertical direction
H3A_AFPAX2[16:13] AFINCV	4	Line increments in a paxel
H3A_AFPAX2[20:17] AFINCH	4	Column increments in a paxel
H3A_AFPAXSTART[27:16] PAXSH	12	Paxel start position H
H3A_AFPAXSTART[11:0] PAXSV	12	Paxel start position V
H3A_AFIIRSH[11:0] IIRSH	12	IIR filter start position

The H3A AF engine also has an option for an advanced or normal stats collection mode. When 0xCA00 is written to the H3A_ADVANCED[31:15] ID bit field, then H3A_ADVANCED[0] AF_MODE can be used to toggle between normal and advanced AF stats collection mode. When the advanced AF stats collection mode is enabled, the ZEROS section of the AF paxel packet is filled with the sum of the maximum FVs, regardless of the color, from HFV_1 and HFV_2.

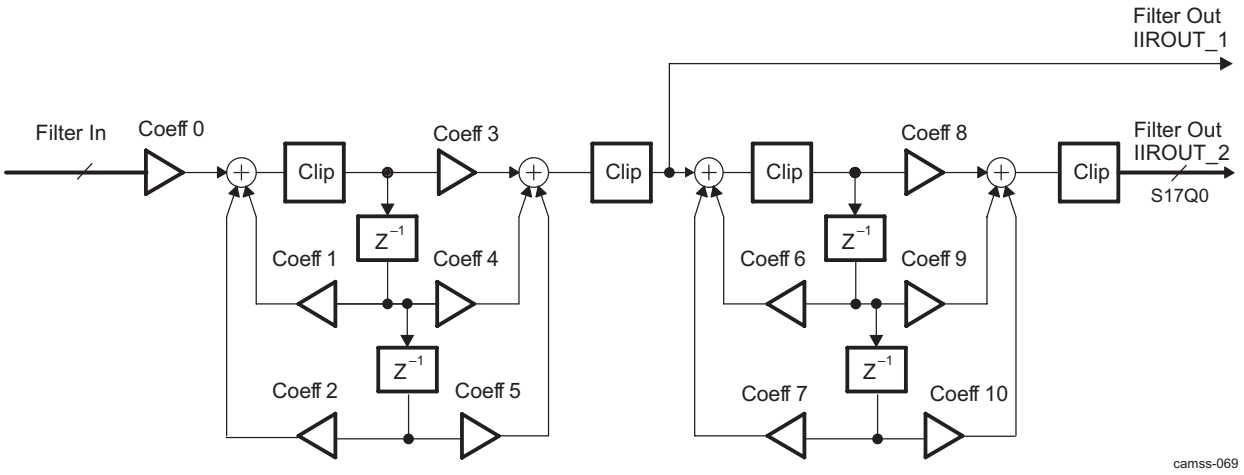
3.3.5.5.2 ISS ISP H3A Horizontal FV Calculator

The FV calculator takes the unsigned red/green/blue extracted data and subtracts 128 or 512 (depending on whether A-Law is enabled) to place the data in the range [–128:127] or [–512:511].

After removing the offset, the data is sent through two parallel IIR filters configured in a dual-biquad configuration. Each filter uses a unique set of 11 programmable coefficients. Each coefficient is 12-bits-wide with 6 bits of decimal, S12Q6 (H3A_AFCOEF010 to H3A_AFCOEF0010 for SET0, and H3A_AFCOEF110 to H3A_AFCOEF1010 for SET1). The filter-shift registers are cleared on each horizontal line at the position set by the register IIR horizontal start register (the H3A_AFIIRSH [11:0] IIRSH bit field). The absolute values of the output (16-bits-wide with 4 bits of decimal, U16Q4) of both filters are then sent to the AF accumulator module. Signed clipping is performed during the FV calculation. If the input value is m bits (signed) and the required output value is n bits, clipping transforms the input to between -2^1 and 2^1. Values lower than -2^1 are set to -2^1, and values higher than 2^1 are set to 2^1.

Figure 214 shows the IIR filter model.

Figure 214. ISS ISP H3A IIR Filter Model



3.3.5.5.3 ISS ISP H3A HFV Accumulator

The horizontal focus value (HFV) accumulator takes the output of the horizontal IIR filter and accumulates values for each paxel. The size and number of paxels is configurable by registers.

Table 186 lists the register fields that configure the size and number of paxels:

- In peak mode (H3A_PCR[14] FVMODE = 0x1), the maximum value is accumulated.
- In sum mode (H3A_PCR[14] FVMODE = 0x0), all HFV_n are accumulated in a paxel.

The following equations detail the calculation for:

- Sum of pixel values used in HFV: The pixel values that are used for filtering and accumulation of HFV are also accumulated in this sum of pixel values.
- HFV_n (HFV_n_peak for peak mode or HFV_n_sum for sum mode)
- HFV_count_n
- HFV_sq_n (HFV_sq_n_peak for peak mode or HFV_sq_n_sum for sum mode)

n = 1 or 2 for IIR1 and IIR2, respectively.

For each paxel, these six values are available for each R, G, and B component.

```
for
(k=0; k<PAXH; k++) // Loop on paxel rows
{
 rowpeak_n = 0;

 for (l=0; l<PAXW; l++) // Loop on values within a row
 {
```

```
aIIRout_n = ABS(IIRout_n);
if (aIIRout_n >= threshold_n)
{
hfval = aIIRout_n - threshold_n;
HFV_count_n++;
}
else hfval = 0;
if (hfval > rowpeak_n)
{
rowpeak_n = hfval;
}
HFV_n_sum += hfval;
HFV_sq_n_sum += (hfval* hfval + RNDADD)>> RNDSHIFT;
} // Finished looping on values in a row
HFV_n_peak += rowpeak_n;
HFV_sq_n_peak += (rowpeak_n * rowpeak_n + RNDADD)>> RNDSHIFT;

}
```

- threshold_n is H3A_HVF_THR[15:0] HTHR1and H3A_HVF_THR[31:16] HTHR2, respectively.
- IIRout_n is the IIRout_1 and IIRout_2 outputs, respectively.
- HFV_count_n and HFV_sq_n are not sent to the DMA interface if VF is disabled.
- RNDADD and RNDSHIFT depend on whether input pixels are 8-bit or 10-bit, and achieves rounding. This is automatically performed by the module.
- If VF is enabled, only the green color channel values are output to the DMA interface.
- In sum mode:
  - HFV_n = HFV_n_sum
  - HFV_sq_n = HFV_sq_n_sum
- In peak mode:
  - HFV_n = HFV_n_peak
  - HFV_sq_n = HFV_sq_n_peak

3.3.5.5.4 ISS ISP H3A VFV Calculator

The VFV calculator takes the unsigned extracted data through two FIR filters, each with a set of five coefficients (VCOEF1_x, where x = 0..4, in the H3A_VFV_CFG1 and H3A_VFV_CFG2 registers for FIR 1, and VCOEF2_x, where x = 0..4, in H3A_VFV_CFG3 and H3A_VFV_CFG4 registers). Each coefficient is 8 bits wide with 4 bits of decimal (S8Q4). The filter outcome is downshifted by 4 bits and taken absolute value to produce a 16-bit unsigned value. This is then sent to threshold H3A_VFV_CFG2[31:16] VTHR1 for FIR 1, and H3A_VFV_CFG4[31:16] VTHR2 for FIR 2, and square logic to produce VFV_n and VFV_sq_n.

3.3.5.5.5 ISS ISP H3A VFV Accumulator

The VFV accumulator takes the output of the vertical FIR filters and accumulates values for each paxel. The size and number of paxels is configurable by registers.

Table 186 lists the register fields that configure the size and number of paxels.

The following equations detail the calculation for:

- VFV_n
- VFV_count_n
- VFV_sq_n

n = 1 or 2 for FIR1 and FIR2, respectively.

For each paxel, these six values are available for each R, G, and B component.

```
FIR_coef_n = [VCOEFn_0, VCOEFn_1, VCOEFn_2, VCOEFn_3, VCOEFn_4]; /* coefficient values in S8.4
format */
aFIRout_n = (ABS(inner_product(extracted_G, FIR_coef_n)) + 8) >> 4;
```

```
if (aFIRout_n ≥ threshold_n)
{
 VFV_n = aFIRout_n - threshold_n
 VFV_count_n++;
}
else VFV_n = 0;

VFV_sq_n = (VFV_n * VFV_n + RNDADD) >> RNDSHIFT;
```

- threshold_n is H3A_VFV_CFG2[31:16] VTHR1 and H3A_VFV_CFG4[31:16] VTHR2, respectively.
- FIRout_n is the FIRout_1 and FIRout_2 outputs, respectively.
- RNDADD and RNDSHIFT depend on whether the input pixels are 8-bit or 10-bit, and achieves rounding. This is automatically performed by the module.

3.3.5.6 ISS ISP H3A AE/AWB Engine

The AE/AWB engine starts by dividing the frames into windows, and then subsamples each window into 2 × 2 blocks. For each subsampled 2 × 2 block, each pixel is accumulated. Also, each pixel is compared to a limit set in a register. If any pixels in a 2 × 2 block are greater than or equal to the limit, the block is not counted in the unsaturated block counter. Pixels greater than the limit are replaced by the limit, and the value of the pixel is accumulated.

The AE/AWB module has three output format modes, which are set through the H3A_AEWCFG[9:8] AEFMT bit field:

- Sum of square mode: H3A_AEWCFG[9:8] AEFMT = 0x0
- Min/max mode: H3A_AEWCFG[9:8] AEFMT = 0x1
- Sum-only mode: H3A_AEWCFG[9:8] AEFMT = 0x2

3.3.5.6.1 ISS ISP H3A Subsampler

The subsampler partitions the frame into windows using the size, count, and starting location parameters shown on the left in Figure 211. Each window is further sampled down to a set of 2 × 2 blocks. The horizontal and vertical distances between the start of blocks within a window is programmable using the parameters shown on the right in Figure 211.

Figure 215. ISS ISP H3A AE/AWB Window Configurations

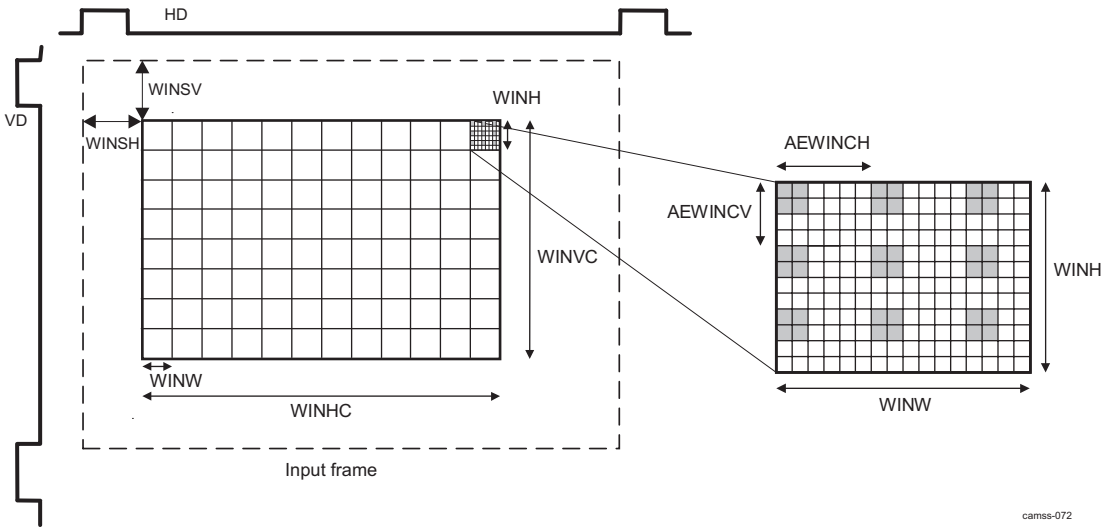


Table 187 lists the register fields that configure the window and block sizes, counts, and starting positions.

Table 187. ISS ISP H3A AE/AWB Window Register Field Descriptions

Register Field	Bit Width	Description
H3A_AEWWIN1[20:13] WINW	7	Window width (in pixels)
H3A_AEWWIN1[31:24] WINH	7	Window height (in lines)
H3A_AEWWIN1[5:0] WINHC	6	Window count for horizontal direction
H3A_AEWWIN1[12:6] WINVC	7	Window count for vertical direction
H3A_AEWINSTART[11:0] WINSH	12	Window start position H
H3A_AEWINSTART[27:16] WINSV	12	Window start position V
H3A_AEWSUBWIN[3:0] AEWINCH	4	Horizontal distance between subsamples
H3A_AEWSUBWIN[11:8] AEWINCV	4	Vertical distance between subsamples

3.3.5.6.2 ISS ISP H3A Additional Black Row of AE/AWB Windows

In addition to the 128 rows of windows, the AE/AWB module provides support for an additional row of windows for black data. This data may be useful in determining the DC offset noise of the rest of the data. The black row of windows can be before or after the regular rows of windows. The vertical start line for the black row of windows is specified in the H3A_AEWINBLK[27:16] WINSV bit field, and the height is specified in the H3A_AEWINBLK[6:0] WINH bit field. The horizontal starting pixel and horizontal width of the black row of windows are the same as for the regular rows of windows.

Figure 216 shows a black row of windows before rows of windows.

Figure 216. ISS ISP H3A Black Row of Windows Before Regular Rows of Windows

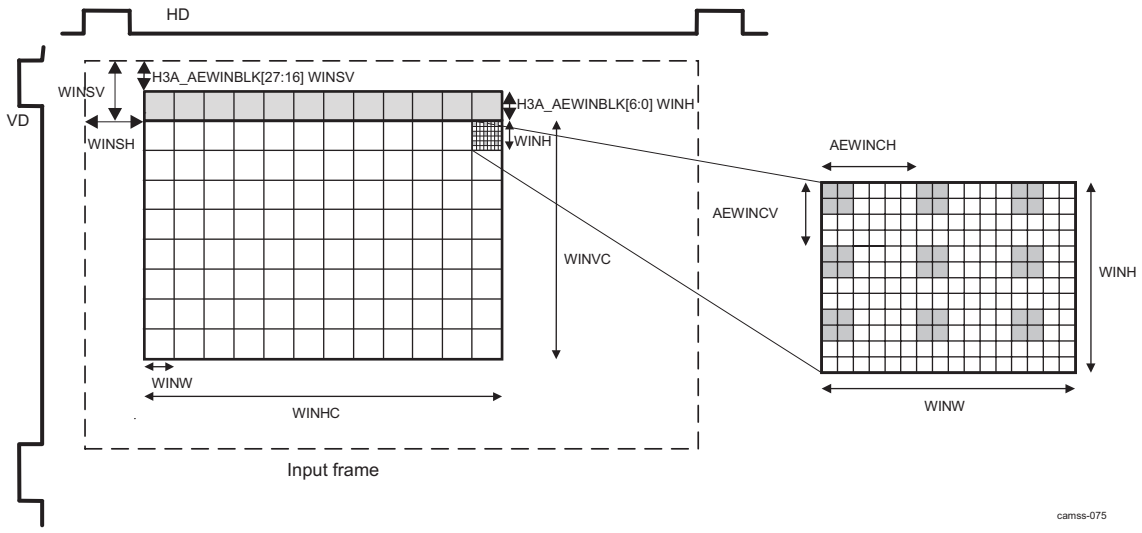


Table 188. ISS ISP H3A AE/AWB Window with Additional Black Row Register Field Descriptions

Register Field	Bit Width	Description
H3A_AEWWIN1[20:13] WINW	7	Window width (in pixels)
H3A_AEWWIN1[31:24] WINH	7	Window height (in lines)
H3A_AEWWIN1[5:0] WINHC	6	Window count for horizontal direction
H3A_AEWWIN1[12:6] WINVC	7	Window count for vertical direction
H3A_AEWINSTART[11:0] WINSH	12	Window start position H
H3A_AEWINSTART[27:16] WINSV	12	Window start position V
H3A_AEWSUBWIN[3:0] AEWINCH	4	Horizontal distance between subsamples
H3A_AEWSUBWIN[11:8] AEWINCV	4	Vertical distance between subsamples
H3A_AEWINBLK[27:16] WINSV	12	Window start position H for single black line
H3A_AEWINBLK[6:0] WINH	7	Window height (in lines) for single black line

3.3.5.6.3 ISS ISP H3A Saturation Check

The saturation check module compares the data from the subsampler to the value programmed in the H3A_PCR [31:22] AVE2LMT bit field. This value is the maximum clipping value. If all 4 pixels in the 2 × 2 block are less than the AVE2LMT value, the value of the unsaturated block counter is incremented. There is one unsaturated block counter per window. The unsaturated block counters are later written to memory.

3.3.5.6.4 ISS ISP H3A AE/AWB Accumulators

The output from the saturation check module and the subsampler module are separately accumulated for each pixel in every 2 × 2 pixel block for each window. Therefore, there are eight accumulators per window (one accumulator for each pixel in a 2 × 2 pixel block, times two sets of accumulators: clipped/saturated data and presaturated data). Each of the 4 pixels in the 2 × 2 pixel grid is associated with a color (R, Gr, B, Gb); however, the output of these accumulators is referenced by position in the grid, not color.

The accumulators are 16 bits wide, and the accumulated data is 10 bits wide. Therefore, when a window contains more than 64 pixels of the same color, an overflow risk exists. This risk can be reduced by enabling the A-Law conversion in the preprocessing stage. See Section 3.3.5.4 for details.

The AE/AWB module has a shift value for the accumulation of pixel values that is set in the H3A_AEWCFG[3:0] SUMSHFT bit field.

3.3.5.7 ISS ISP H3A DMA Interface

The DMA interface module takes the data from the AF engine and AE/AWB engine and builds packets to be sent out to the memory through the BL module.

The data interface has separate start pointers for the AF and AE/AWB engines.

- The starting address for the AF engine is the H3A_AFBUFST[31:5] AFBUFST bit field.
- The starting address for the AE/AWB engine is the H3A_AEWBUFST[31:5] AEWBUFST bit field.

The DMA interface module continuously loops through this data as it builds the packets. To optimize the transfer sizes, the DMA interface sends out an AF or AE transfer for each row of paxels or windows. This requires that each horizontal row of paxels or windows starts and ends on a 32-byte boundary. If a horizontal row of paxels or windows ends on a non-32 byte boundary, the hardware packs zeroes. The counts for the AEW that occur every eight windows is sent in the row with the 8th consecutive window.

Table 189 lists the packet formats for AF with vertical AF disabled.

Table 189. ISS ISP H3A AF Packet Format With Vertical AF Disabled

Buffer Start Address (Byte Address) H3A_AFBUFST	31 16	15 0
	Sum of pixel values used in HFV	(Paxel 0:G)
	HFV_1 (peak or sum)	(Paxel 0:G)
	HFV_2 (peak or sum)	(Paxel 0:G)
	ZEROES	(Paxel 0:G)
	Sum of pixel values used in HFV	(Paxel 0:R)
	HFV_1 (peak or sum)	(Paxel 0:R)
	HFV_2 (peak or sum)	(Paxel 0:R)
	ZEROES	(Paxel 0:R)
	Sum of pixel values used in HFV	(Paxel 0:B)
	HFV_1 (peak or sum)	(Paxel 0:B)
	HFV_2 (peak or sum)	(Paxel 0:B)
	ZEROES	(Paxel 0:B)
	Sum of pixel values used in HFV	(Paxel 1)
	HFV_1 (peak or sum)	(Paxel 1)
	HFV_2 (peak or sum)	(Paxel 1)
	ZEROES	(Paxel 1)

Table 189. ISS ISP H3A AF Packet Format With Vertical AF Disabled (continued)

Buffer Start Address (Byte Address) H3A_AFBUFST	31 16	15 0
	...	

Table 190 shows the packet formats for AF with vertical AF enabled.

Table 190. ISS ISP H3A AF Packet Format With Vertical AF Enabled

Buffer Start Address (Byte Address) H3A_AFBUFST	31 16	15 0
	Sum of pixel values used in HFV	(Paxel 0)
	HFV_1 (peak or sum)	(Paxel 0)
	HFV_sq_1 (peak or sum)	(Paxel 0)
	HFV_count_1	(Paxel 0)
	HFV_2 (peak or sum)	(Paxel 0)
	HFV_sq_2 (peak or sum)	(Paxel 0)
	HFV_count_2	(Paxel 0)
	ZEROES	(Paxel 0)
	VFV_1	(Paxel 0)
	VFV_sq_1	(Paxel 0)
	VFV_count_1	(Paxel 0)
	ZEROES	(Paxel 0)
	VFV_2	(Paxel 0)
	VFV_sq_2	(Paxel 0)
	VFV_count_2	(Paxel 0)
	ZEROES	(Paxel 0)
	Sum of pixel values used in HFV	(Paxel 1)
	HFV_1 (peak or sum)	(Paxel 1)
	HFV_sq_1 (peak or sum)	(Paxel 1)
	HFV_count_1	(Paxel 1)
	...	

Figure 217 shows the windows and subsample definition used in tables.

Figure 217. ISS ISP H3A AE/AWB Window and Subsample Definition

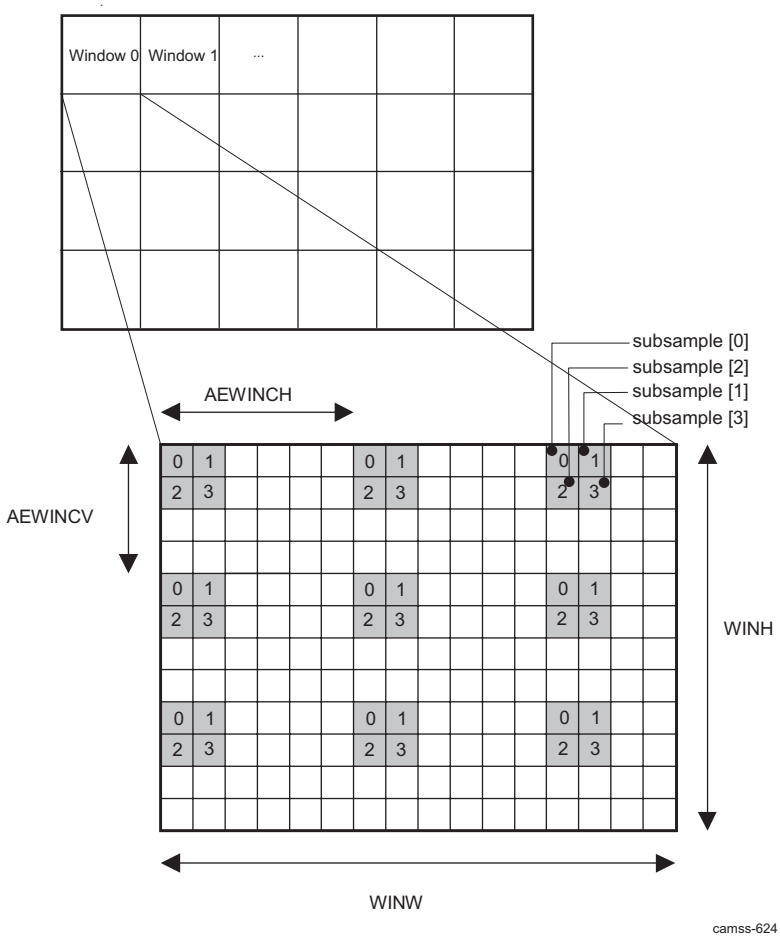


Table 191 lists the packet formats for AE/AWB for sum of square mode (H3A_AEWCFG[9:8] AEFMT = 0x0) .

Table 191. ISS ISP H3A AE/AWB Packet Format for Sum of Square Mode

	31 16	15 0	
Buffer address (byte address) H3A_AEWBUFST	Subsample Accum[1]	Subsample Accum[0]	Window 0 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
	Sum of squares[0]		
	Sum of squares[1]		
	Sum of squares[2]		
	Sum of squares[3]		
H3A_AEWBUFST + 32 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 1 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
	Sum of squares[0]		

Table 191. ISS ISP H3A AE/AWB Packet Format for Sum of Square Mode (continued)

	31 16	15 0	
	Sum of squares[1]		
	Sum of squares[2]		
	Sum of squares[3]		
H3A_AEWBUFST + 64 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 2 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
	Sum of squares[0]		
	Sum of squares[1]		
	Sum of squares[2]		
	Sum of squares[3]		
H3A_AEWBUFST + 96 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 3 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
	Sum of squares[0]		
	Sum of squares[1]		
	Sum of squares[2]		
	Sum of squares[3]		
H3A_AEWBUFST + 128 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 4 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
	Sum of squares[0]		
	Sum of squares[1]		
	Sum of squares[2]		
	Sum of squares[3]		
H3A_AEWBUFST + 160 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 5 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
	Sum of squares[0]		
	Sum of squares[1]		
	Sum of squares[2]		
	Sum of squares[3]		
H3A_AEWBUFST + 192 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 6 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
	Sum of squares[0]		
	Sum of squares[1]		
	Sum of squares[2]		
	Sum of squares[3]		

Table 191. ISS ISP H3A AE/AWB Packet Format for Sum of Square Mode (continued)

	31 16	15 0	
H3A_AEWBUFST + 224 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 7 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
	Sum of squares[0]		
	Sum of squares[1]		
	Sum of squares[2]		
	Sum of squares[3]		
H3A_AEWBUFST + 256 bytes	Unsaturated count, win 1	Unsaturated count, win 0	Unsaturated block count for the above 8 windows
	Unsaturated count, win 3	Unsaturated count, win 2	
	Unsaturated count, win 5	Unsaturated count, win 4	
	Unsaturated count, win 7	Unsaturated count, win 6	
	Data for next eight windows, and so on. If the total number of windows is not a multiple of 8, the unsaturated counters are written immediately following the last window data. For example, if the total number of windows (including the black row) are 43, the first 40 windows are written out as per the 272-byte boundary above. Then the remaining three windows are written at +0, +32, and +64 bytes. The counts are written out at +96 instead of +256-byte boundary.		

Table 192 lists the packet formats for AE/AWB in minimum-maximum mode (H3A_AEWCFG[9:8] AEFMT = 0x1).

Table 192. ISS ISP H3A AE/AWB Packet Format for Minimum-Maximum Mode

	31 16	15 0	
Buffer address (byte address) H3A_AEWBUFST	Subsample Accum[1]	Subsample Accum[0]	Window 0 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
	Minimum[1]	Minimum[0]	
	Minimum[3]	Minimum[2]	
	Maximum[1]	Maximum[0]	
	Maximum[3]	Maximum[2]	
H3A_AEWBUFST + 32 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 1 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
	Minimum[1]	Minimum[0]	
	Minimum[3]	Minimum[2]	
	Maximum[1]	Maximum[0]	
	Maximum[3]	Maximum[2]	
H3A_AEWBUFST + 64 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 2 data

Table 192. ISS ISP H3A AE/AWB Packet Format for Minimum-Maximum Mode (continued)

	31 16	15 0	
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
	Minimum[1]	Minimum[0]	
	Minimum[3]	Minimum[2]	
	Maximum[1]	Maximum[0]	
	Maximum[3]	Maximum[2]	
H3A_AEWBUFST + 96 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 3 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
	Minimum[1]	Minimum[0]	
	Minimum[3]	Minimum[2]	
	Maximum[1]	Maximum[0]	
	Maximum[3]	Maximum[2]	
H3A_AEWBUFST + 128 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 4 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
	Minimum[1]	Minimum[0]	
	Minimum[3]	Minimum[2]	
	Maximum[1]	Maximum[0]	
	Maximum[3]	Maximum[2]	
H3A_AEWBUFST + 160 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 5 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
	Minimum[1]	Minimum[0]	
	Minimum[3]	Minimum[2]	
	Maximum[1]	Maximum[0]	
	Maximum[3]	Maximum[2]	
H3A_AEWBUFST + 192 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 6 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
	Minimum[1]	Minimum[0]	
	Minimum[3]	Minimum[2]	
	Maximum[1]	Maximum[0]	
	Maximum[3]	Maximum[2]	
H3A_AEWBUFST + 224 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 7 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
	Minimum[1]	Minimum[0]	

Table 192. ISS ISP H3A AE/AWB Packet Format for Minimum-Maximum Mode (continued)

	31 16	15 0	
	Minimum[3]	Minimum[2]	
	Maximum[1]	Maximum[0]	
	Maximum[3]	Maximum[2]	
H3A_AEWBUFST + 256 bytes	Unsaturated count, win 1	Unsaturated count, win 0	Unsaturated block count for the above 8 windows
	Unsaturated count, win 3	Unsaturated count, win 2	
	Unsaturated count, win 5	Unsaturated count, win 4	
	Unsaturated count, win 7	Unsaturated count, win 6	
	Data for next eight windows, and so on. If the total number of windows is not a multiple of 8, the unsaturated counters are written immediately following the last window data. For example, if the total number of windows (including the black row) are 43, the first 40 windows are written out as per the 272-byte boundary above. Then the remaining three windows are written at +0, +32, and +64 bytes. The counts are written out at +96 instead of +256-byte boundary.		

Table 193 lists the packet formats for AE/AWB in sum-only mode (H3A_AEWCFG[9:8] AEFMT = 0x2).

Table 193. ISS ISP H3A AE/AWB Packet Format for Sum-Only Mode

	31 16	15 0	
Buffer address (byte address) H3A_AEWBUFST	Subsample Accum[1]	Subsample Accum[0]	Window 0 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
H3A_AEWBUFST + 32 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 1 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
H3A_AEWBUFST + 64 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 2 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
H3A_AEWBUFST + 96 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 3 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
H3A_AEWBUFST + 128 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 4 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	

Table 193. ISS ISP H3A AE/AWB Packet Format for Sum-Only Mode (continued)

	31 16	15 0	
H3A_AEWBUFST + 160 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 5 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
H3A_AEWBUFST + 192 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 6 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
H3A_AEWBUFST + 224 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 7 data
	Subsample Accum[3]	Subsample Accum[2]	
	Saturator Accum[1]	Saturator Accum[0]	
	Saturator Accum [3]	Saturator Accum[2]	
H3A_AEWBUFST + 256 bytes	Unsaturated count, win 1	Unsaturated count, win 0	Unsaturated block count for the above 8 windows
	Unsaturated count, win 3	Unsaturated count, win 2	
	Unsaturated count, win 5	Unsaturated count, win 4	
	Unsaturated count, win 7	Unsaturated count, win 6	
	Data for next eight windows, and so on. If the total number of windows is not a multiple of 8, the unsaturated counters are written immediately following the last window data. For example, if the total number of windows (including black row) are 43, the first 40 windows are written out as per the 272-byte boundary above. Then the remaining three windows are written at +0, +32, and +64 bytes. The counts are written out at +96 instead of +256-byte boundary.		

3.3.5.8 ISS ISP H3A Events and Status Checking

The AF and AEW engines generate an interrupt event at the end of processing each frame. However, these two interrupts are internally tied together so that only one H3A interrupt signal is generated. If the AF engine and AEW engine do not process the same frame concurrently, this should not be an issue. However, if they do run concurrently, one of two outcomes may occur:

- The H3A interrupt may seem to trigger only once for each frame. This can happen when the processing for the AF and AEW engines finishes at or near the same time. The interrupt service routine does not have enough time to clear the interrupt flag for the first interrupt before the second interrupt occurs.
- The H3A interrupt may trigger twice for each frame. This can happen when the AF engine or the AEW engine finishes processing the frame much earlier than the other engine. In this case, the interrupt service routine does have enough time to clear the interrupt flag for the first interrupt by the time the second interrupt occurs.

The outcome depends on the difference in location of the last paxel/window in the frame (determines when processing is finished), the frequency of the relative clocks in the system, the occurrence and triggering of other interrupts in the system, and the latencies of the context switching and interrupt service routine execution.

The H3A_PCR[15] BUSYAF and/or H3A_PCR[18] BUSYAEAWB status bits are set when the start of frame occurs (if the H3A_PCR[0] AF_EN and/or H3A_PCR[16] AEW_EN bits are 1 at that time). They are automatically reset to 0 at the end of processing a frame. The H3A_PCR[15] BUSYAF and/or H3A_PCR[18] BUSYAEAWB status bits may be polled to determine the end of frame status.

3.3.6 ISS ISP ISIF Functional Description

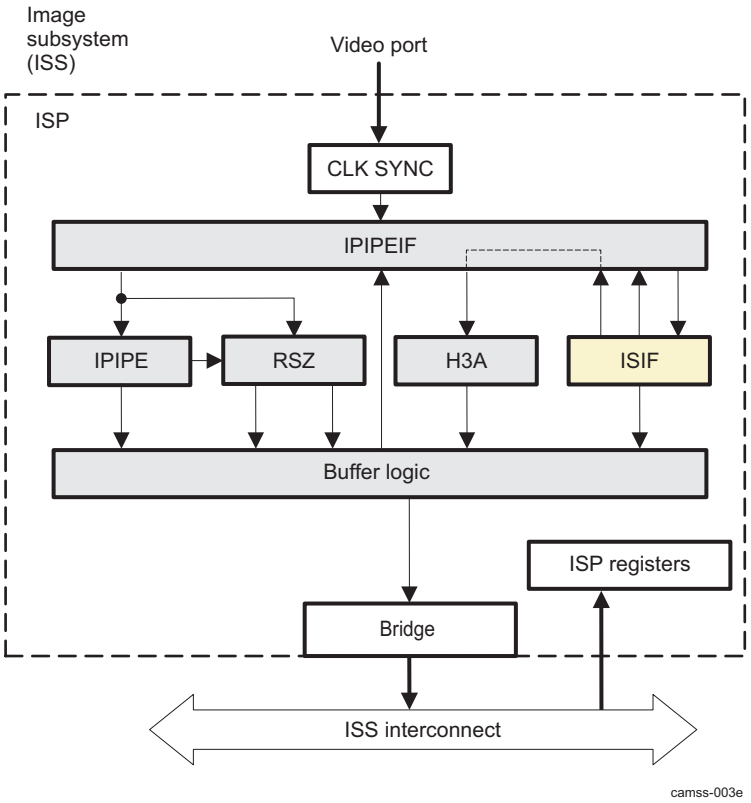
3.3.6.1 ISS ISP ISIF Overview

The image sensor interface (ISIF) module receives RAW or YUV4:2:2 data from the IPIPEIF module. The module outputs data back to the IPIPEIF module and can also output data to memory though the BL module. The ISIF module can process the incoming data and supports the following functions:

- Maximum supported image size is 32,768 × 32,768
- Supports up to 16-bit analog front end
- Sensor data linearization
- Supports Bayer and Foveon® input data format (RGB and CMYG color support)
- Supports VGA read out mode
- Supports various image data format
- Color space conversion
- Digital clamp with horizontal/vertical offset drift compensation
- Vertical line defect correction
- Programmable 2D-matrix LSC
- Gain and offset control
- Programmable horizontal/vertical culling pattern
- Maximum pixel rate clock of 200 MPix/s on the VP interface.
- 10-to-8-bit A-Law compression table inside
- 12-bit pack supported when written to memory

Figure 218 show the ISIF module connections to other submodules of the ISP.

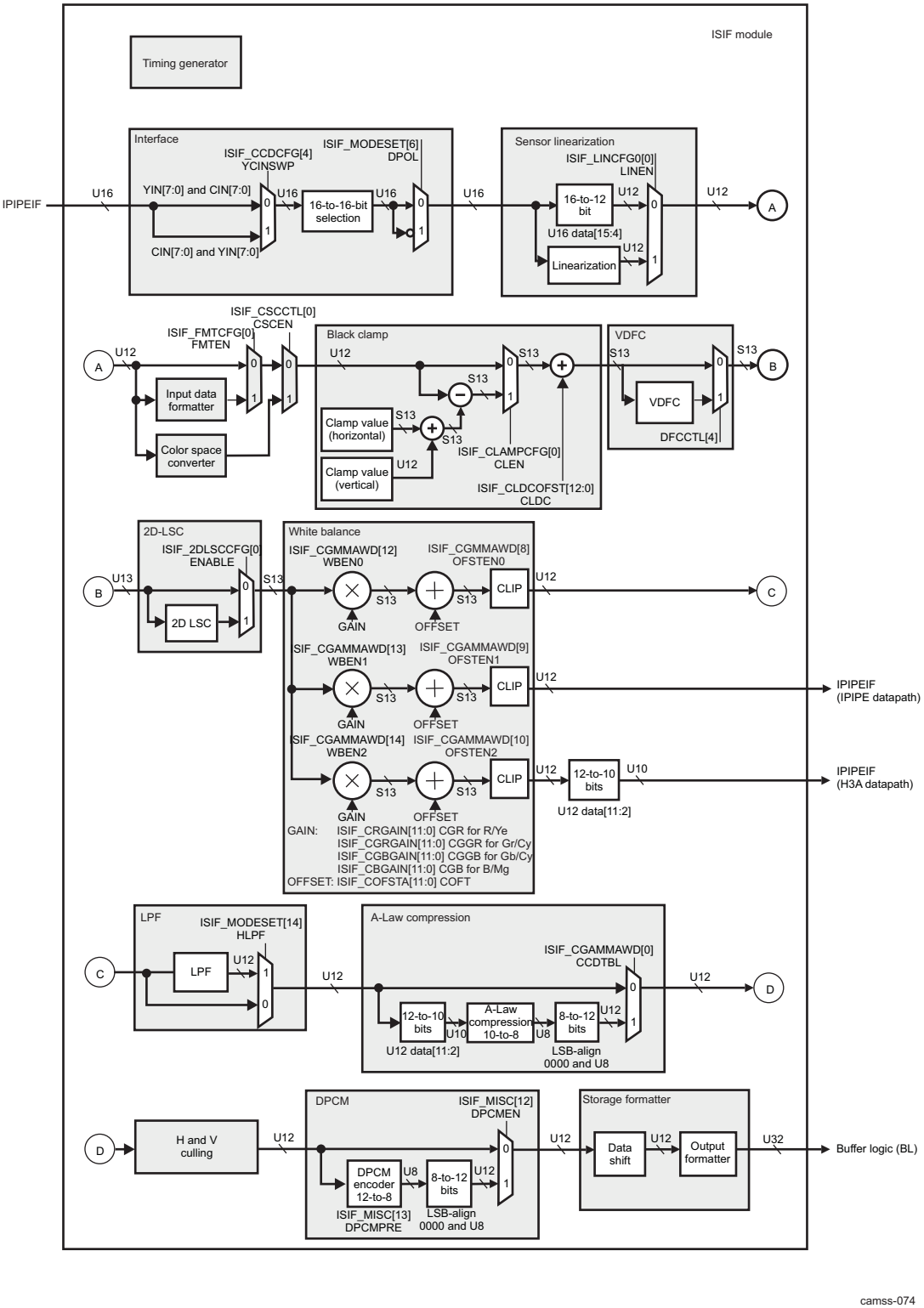
Figure 218. ISS ISP ISIF High-Level Diagram



3.3.6.2 ISS ISP ISIF Top-Level Block Diagram

Figure 219 shows the different blocks of the ISIF module.

Figure 219. ISS ISP ISIF Top-Level Block Diagram



camss-074

The following sections describe the blocks in the ISIF module.

3.3.6.3 ISS ISP ISIF Input Interface

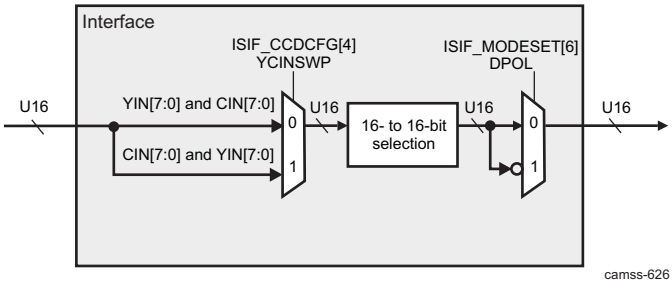
The input interface is a video interface. It comprises the horizontal (HD) and vertical (VD) synchronization signal, pixel clock (PCLK), and data (DATA). Table 194 gives more information about these different signals. The ISIF uses the HD and VD signals provided by the sensor through the VP and IPIPEIF. The pixel clock clocks data into the ISIF at a maximum rate of 200 MHz.

Table 194. ISS ISP ISIF Input Interface Signals

Name	I/O	Function
VD	I	Vertical sync signal
HD	I	Horizontal sync signal
PCLK	I	Pixel clock. This signal is the pixel clock used to load image data into the ISIF. The clock controller can configure to trigger on the rising or falling edge of the PCLK signal.
DATA	I	Data. The data interface is a 16-bit interface. When the ISIF is configured to write data to SDRAM, the write enable signal allows an external device to control which data is to be written to SDRAM. The data input can be configured from the ISIF_MODESET.INPMOD register where it can be set to RAW, YCbCr (16 bits and 8 bits). The polarity of the data can be changed from the ISIF_MODESET.DPOL as shown in Figure 220.

3.3.6.4 ISS ISP ISIF Interface

Figure 220. ISS ISP ISIF Interface Block Diagram



The ISIF supports 8- to 16-bit-wide RAW data signals and 8-/16-bit YCbCr signals, as described in Table 195. The interface can be set in the three different modes from the ISIF_MODESET[13:12] INPMOD bit field. The ISIF_CCDCFG[11] Y8POS bit selects the y signal positioning whenever YUV4:2:2 is input. Moreover, if CCIR656 input is used the width of selected bit can be set through the ISIF_CCDCFG[5] BT656 bit.

Table 195. ISS ISP ISIF Data Input Formats

ISIF Input Port Name	RAW Data	16-bit YCbCr	8-bit YCbCr
YI7	C_DATA15	Y7	
YI6	C_DATA14	Y6	
YI5	C_DATA13	Y5	
YI4	C_DATA12	Y4	
YI3	C_DATA11	Y3	
YI2	C_DATA10	Y2	
YI1	C_DATA9	Y1	

Table 195. ISS ISP ISIF Data Input Formats (continued)

ISIF Input Port Name	RAW Data	16-bit YCbCr	8-bit YCbCr
YI0	C_DATA8	Y0	
CI7	C_DATA7	Cb7,Cr7	Y7,Cb7,Cr7
CI6	C_DATA6	Cb6,Cr6	Y6,Cb6,Cr6
CI5	C_DATA5	Cb5,Cr5	Y5,Cb5,Cr5
CI4	C_DATA4	Cb4,Cr4	Y4,Cb4,Cr4
CI3	C_DATA3	Cb3,Cr3	Y3,Cb3,Cr3
CI2	C_DATA2	Cb2,Cr2	Y2,Cb2,Cr2
CI1	C_DATA1	Cb1,Cr1	Y1,Cb1,Cr1
CI0	C_DATA0	Cb0,Cr0	Y0,Cb0,Cr0

Y and C input signals can be swapped through the ISIF_CCDCFG[4] YCINSWP bit.

In case of RAW data at ISIF input, a 16- to-16-bit selection can be done: when the number of RAW data lines is less than 16, data can be connected to the upper or lower lines of C_DATA[15:0]. Lines not connected must be tied low. As shown in Table 196, the ISIF_CGAMMAWD[4:1] GWDI bit field must be configured correctly so that the MSB of the input is connected to the MSB of the 16-bit data bus in ISIF.

Table 196. ISS ISP ISIF Raw Data Connection: Selects MSB Position of Input Data

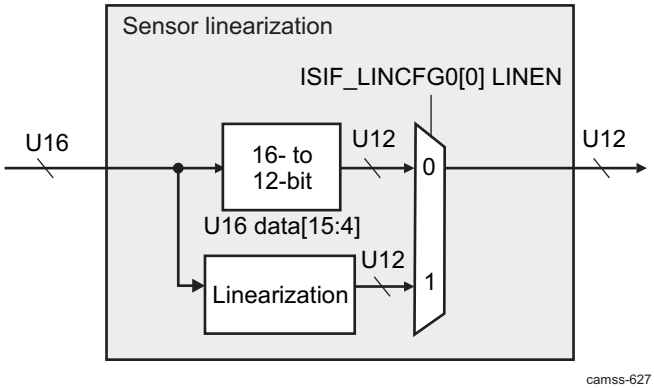
ISIF_CGAMMAWD[4:1] GWDI	16-to-16-bit Selection
0	C_DATA [15:0] = C_DATA[15:0]
1	C_DATA [15:0] = C_DATA[14:0] 0
2	C_DATA [15:0] = C_DATA[13:0] 00
3	C_DATA [15:0] = C_DATA[12:0] 000
4	C_DATA [15:0] = C_DATA[11:0] 0000
5	C_DATA [15:0] = C_DATA[10:0] 00000
6	C_DATA [15:0] = C_DATA[9:0] 000000
7	C_DATA [15:0] = C_DATA[8:0] 0000000
8	C_DATA [15:0] = C_DATA[7:0] 00000000

The polarity of the input image data can be switched through the ISIF_MODESET[6] DPOL bit.

3.3.6.5 ISS ISP ISIF Sensor Linearization

NOTE: For the memory access locations of the sensor linearization table, see Section 3.3.8.

Figure 221. ISS ISP ISIF Sensor Linearization Block Diagram



The sensor linearization module can correct for the non-linear response of image sensors. A LUT is programmed with an offset value to add to the original pixel value based on the original pixel value.

The LUT is a sampling of the linearization correction curve based on calibration of the image sensor. Intermediate values between sampling points are computed using linear interpolation. The entire correction curve is divided into seven regions, as shown in Table 197. The regions for the darkest part and the brightest part of the response curve have dense sampling. The linearization mode can be a uniform or a non-uniform sampling and can be set through the ISIF_LINCFG0[1] LINMD bit.

Table 197. ISS ISP ISIF Linearization LUT

Region	Number of Sample Points	LUT Address
table_in[15:11] == 00000	32	table_in[10:6]
table_in[15:11] == 00001	4	table_in[10:9] + 32
table_in[15:12] == 0001	4	table_in[11:10] + 36
table_in[15:13] == 001	4	table_in[12:11] + 40
table_in[15:14] == 01	4	table_in[13:12] + 44
table_in[15:14] == 10	16	table_in[13:10] + 48
table_in[15:14] == 01	128	table_in[13:7] + 64

The LUT has 192 entries and is split into two 96 × 10-bit memories, as shown in Figure 222. The table is mapped in the memory map. The LUT entries are interleaved between memory 0 and memory 1.

Figure 222. ISS ISP ISIF Linearization LUT Memories

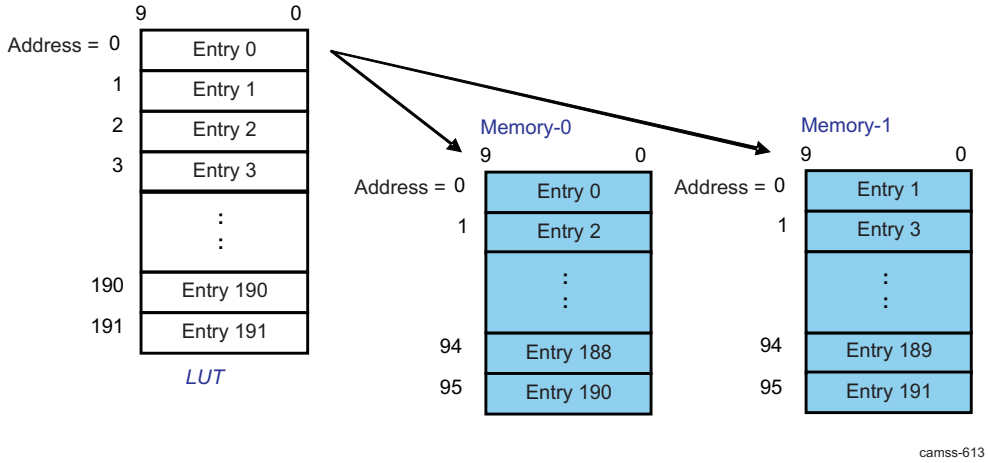
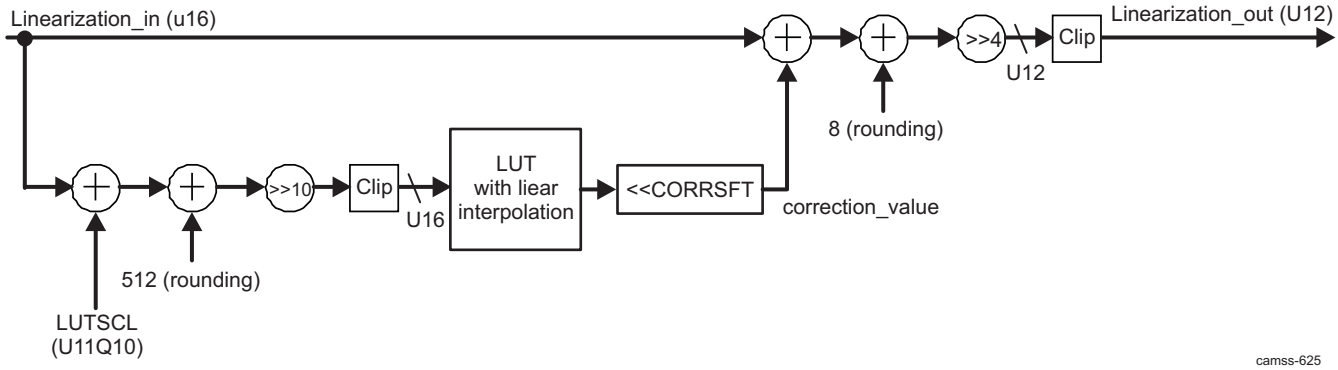


Table 198. ISS ISP ISIF LUT Memory Region

Memory Region	Address Range	Description
Memory 0	0xC000 – 0xC17F	ISIF linearity compensation LUT 0
Memory 1	0xC400 – C57F	ISIF linearity compensation LUT 1

A scale factor is applied to the input before lookup through the ISIF_LINCFG1[10:0] LUTSCL bit field. The LUT entries are signed 10-bit data (u16). After linear interpolation, the correction value is left-shifted by a programmable amount (the ISIF_LINCFG0[6:4] CORRSFT bit field), and then added to the input. This is then converted to unsigned 12-bit by right shift, followed by clipping.

Figure 223. ISS ISP ISIF Linearization Block Diagram

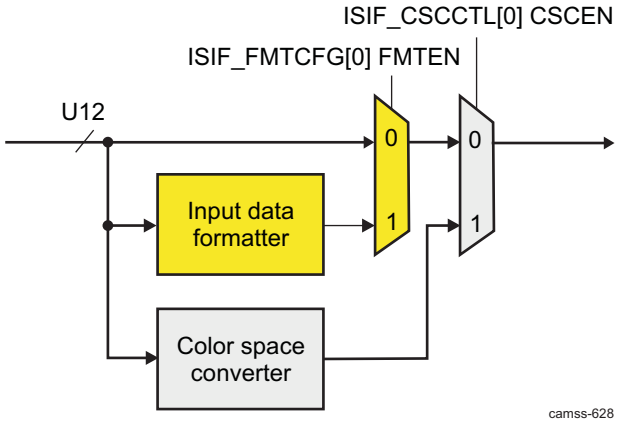


To enable the linearization module, set the ISIF_LINCFG0[0] LINEN bit to 1.

If the linearization module is disabled (ISIF_LINCFG0[0] LINEN = 0x0), a 16- to-12-bit transformation is done, and the upper 12-bits of U16 input are sent to the next block.

3.3.6.6 ISS ISP ISIF Input Data Formatter

Figure 224. ISS ISP ISIF Input Data Formatter Block Diagram



There are two functional blocks: input data formatter and color space converter, which use two 5376 × 12-bit memories (corresponds to one line of maximum 5376 pixels with each pixel equal to 12 bits). Only one of the function blocks can be enabled.

The input data formatter block allows the ISIF to handle a wide variety of current and future readout schemes other than Bayer format. Two line memories and a programmable address generator are used to translate those patterns into a standard Bayer pattern (or any other pattern). This allows the back-end processing (noise filters, interpolation, histogram, 3A statistics) to remain unchanged.

The input data formatter block also supports divided input lines. In case an input line is divided into multiple lines and fed to the ISIF, the formatter gathers the divided lines and organizes a single line. Up to four divided lines can be supported.

The input data formatter is enabled through the ISIF_FMTCFG[0] FMTEN bit.

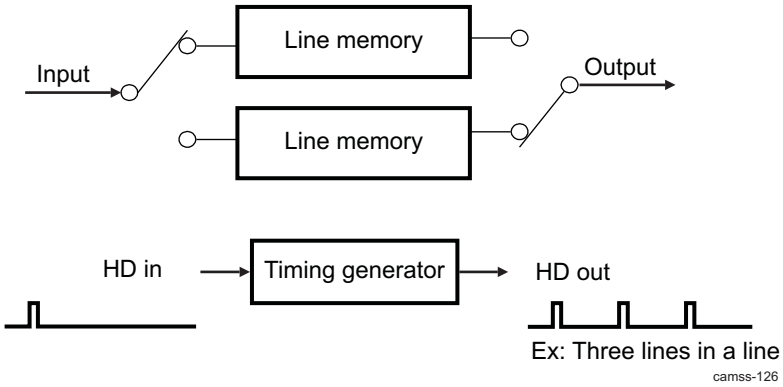
The input data formatter can split an input line into 1, 2, 3, or 4 output lines, or can combine the divided 1, 2, 3, or 4 input lines into a single line.

- Set the ISIF_FMTCFG[1] FMTCBL bit to 0 for split mode.
- Set the ISIF_FMTCFG[1] FMTCBL bit to 1 for combine mode.
- Select the number of lines in the ISIF_FMTCFG[5:4] LNUM bit field.

The Input data formatter can work in normal or line alternative mode. The choice is done through the ISIF_FMTCFG[2] LNALT bit.

Figure 225 shows an example of generating three output lines from an input line with a new, internally generated HD signal.

Figure 225. ISS ISP ISIF Splits an Input Line Into Three Output Lines



This HD signal then gates the downstream processing rather than the original sensor HD signal. Descriptions of how to configure the formatter are provided in the following sections.

Because the size of the line memories is 5376 × 12 bits, the following restrictions apply for the data formatter:

- Split mode:
  - The maximum number of pixels that can be supported in an output line if the input line is transformed into 1 output line is 5376.
  - The maximum number of pixels that can be supported in an output line if the input line is transformed into 2 output lines is 2688.
  - The maximum number of pixels that can be supported in an output line if the input line is transformed into 3 output lines is 1792.
  - The maximum number of pixels that can be supported in an output line if the input line is transformed into 4 output lines is 1344.
- Combine mode:
  - The maximum number of pixels that can be supported in an input line if 1 input line is transformed into an output line is 5376.
  - The maximum number of pixels that can be supported in an input line if 2 input lines are transformed into an output line is 2368.
  - The maximum number of pixels that can be supported in an input line if 3 input lines are transformed into an output line is 1792.
  - The maximum number of pixels that can be supported in an input line if 4 input lines are transformed into an output line is 1344.

3.3.6.6.1 ISS ISP ISIF Formatter Area Settings

As shown in Figure 226, the following registers are used to set the formatter area:

- ISIF_FMTSPH
- ISIF_FMTLNH
- ISIF_FMTLSV
- ISIF_FMTLNV

Table 199 describes these registers. The input line is input to the formatter, and the output line is output from the formatter.

Figure 226. ISS ISP ISIF Input Data Formatter Area Settings

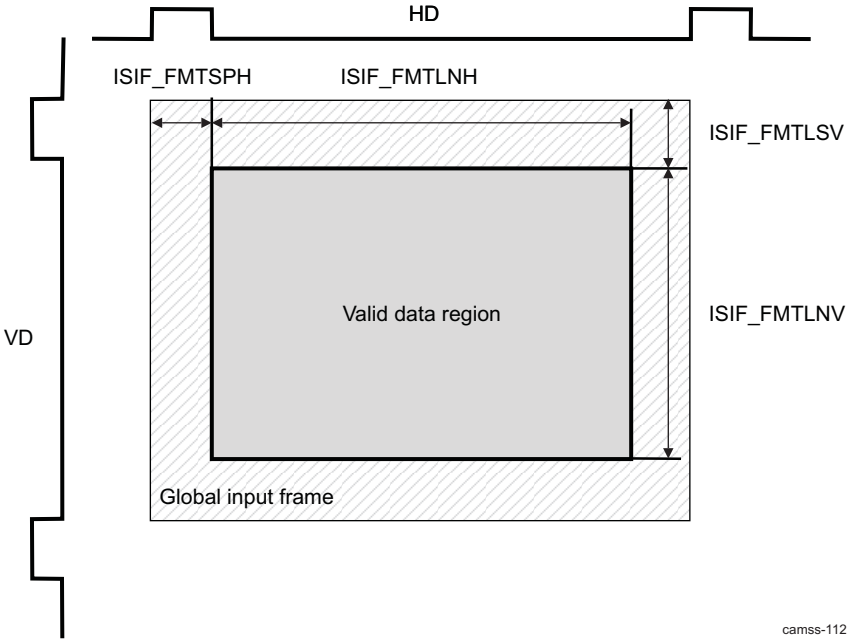


Table 199. ISS ISP ISIF Input Data Formatter Area Setting Registers

Register	Description
ISIF_FMTSPH	The first valid pixel of an input line
ISIF_FMTLNH	Valid length of a input line = FMTLNH + 1
ISIF_FMTLSV	The first valid input line
ISIF_FMTLNV	The number of the valid input lines = FMTLNV + 1

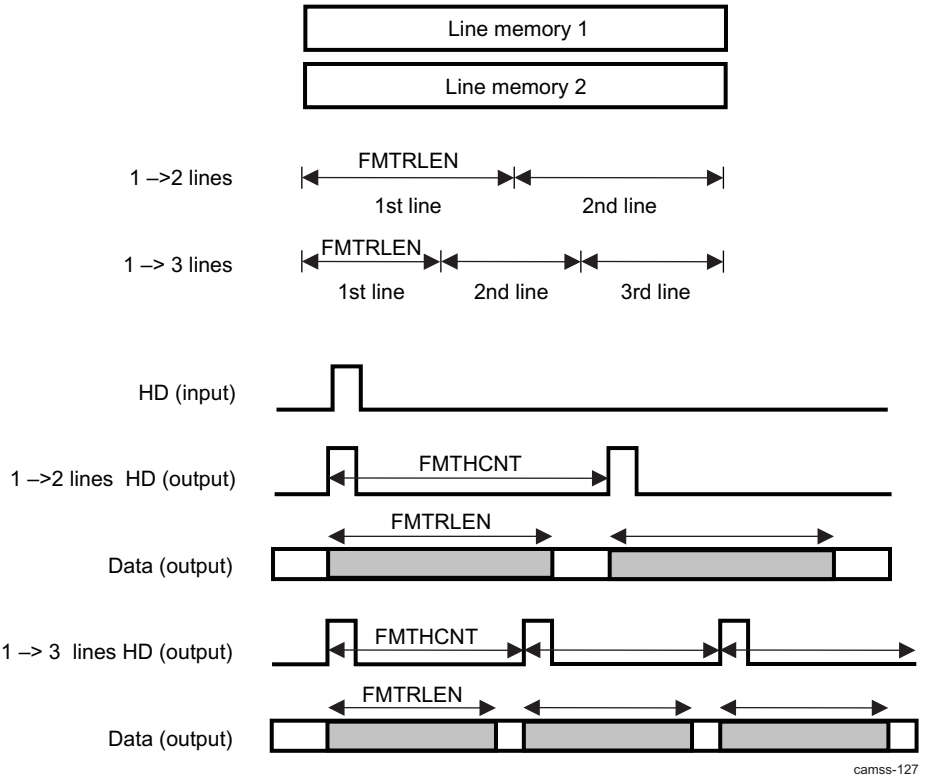
Table 200. ISS ISP ISIF Output Data Formatter Area Setting Registers

Register	Description
ISIF_FMTRLEN	The length of an output line
ISIF_FMTHCNT	HD interval for output lines
ISIF_SPH	The first pixel in an output line to be stored to SDRAM
ISIF_LNH	Number of pixels in an output line to be stored to SDRAM = LNH + 1
ISIF_LNV	The number of the output lines to be stored to SDRAM = LNV + 1

The number of pixels in an output line must be set to the ISIF_FMTRLEN register, and the HD output interval must be set to the ISIF_FMTHCNT register. It is not necessary to set the ISIF_FMTHCNT register if multiple input lines are combined into a single line.

Figure 227 shows an example of splitting an input line into two or three output lines.

Figure 227. ISS ISP ISIF Data Formatter Output Control Example



3.3.6.6.2 ISS ISP ISIF Formatter Programming

The data formatter derives its flexibility by supporting up to 16 different addresses and a program that can contain up to 32 entries.

Address pointer

There are 16 address pointer registers (ISIF_FMTAPTR0 to ISIF_FMTAPTR15), which contain:

- The ISIF_FMTAPTRx[14:13] LINE bit field: 2-bit line number to specify the output line to which it belongs: 0, 1, 2, or 3. It is valid only for the line splitting.
- The ISIF_FMTAPTRx[12:0] INIT bit field: 13-bit initial address for pointer x (where x = 0 to 15)

Each of the address values is auto-incremented or auto-decremented by a programmable value (the ISIF_FMTCFG[11:8] FMTAINC bit field).

Program

There are 32 program entry registers, which contain:

- In the ISIF_FMTPGMVF0 and ISIF_FMTPGMVF1 registers: The PGMxxEN fields (where xx = 00 to 31) set the program entry valid flag.
- In the ISIF_FMTPGMAPS0 to ISIF_FMTPGMAPS7 registers: The PGMxxAPTR fields specify the program xx address pointer (where xx = 00 to 31).
- In the ISIF_FMTPGMAPU0 and ISIF_FMTPGMAPU1 registers: The PGMxxUPDT fields (where xx = 00 to 31) set the program xx address update (increment or decrement).

Because each of the program entries has a valid flag, the formatter can also support images larger than the memory limit (length: 4480) by not choosing some of the entries.

The total program memory of 32 entries is divided into two or four SETs:

- Program memory of 16 entries each for odd and even lines to split the line:
  - SET0 for even input line: Program entry 015
  - SET1 for odd input line: Program entry 1631
- Program memory of eight entries for up to four input lines to combine the lines:
  - SET0 for first input line: Program entry 07
  - SET1 for second input line: Program entry 815
  - SET2 for third input line: Program entry 1623
  - SET3 for fourth input line: Program entry 2431

The number of program entries per SET must be specified by the ISIF_FMTPLEN register as follows:

- Number of program entries for SET0: ISIF_FMTPLEN[3:0] FMTPLEN0
- Number of program entries for SET1: ISIF_FMTPLEN[7:4] FMTPLEN1
- Number of program entries for SET2: ISIF_FMTPLEN[10:8] FMTPLEN2
- Number of program entries for SET3: ISIF_FMTPLEN[14:12] FMTPLEN3

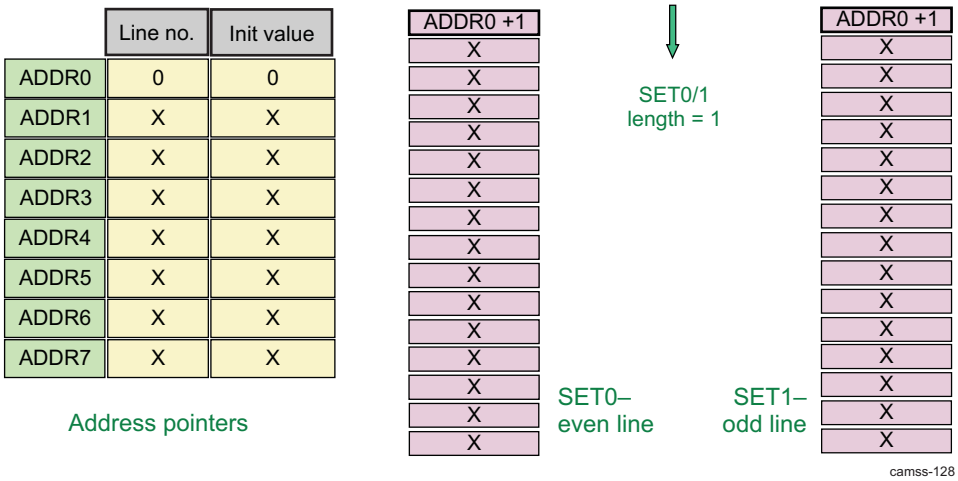
The program entry must be set from the lower registers within a SET. For instance, start from program entry 16 for odd input lines.

The following examples show the programmability of the data formatter:

- Register settings:
  - ISIF_FMTCFG[0] FMTEN = 0x1
  - ISIF_FMTCFG[1] FMTCBL = 0x0
  - ISIF_FMTCFG[11:8] FMTAINC = 0x0 (add or subtract 1)
- 1 input line - 1 output line

NOTE: ADDR0 is set to an example value. Addresses with no value in the following example are marked with X.

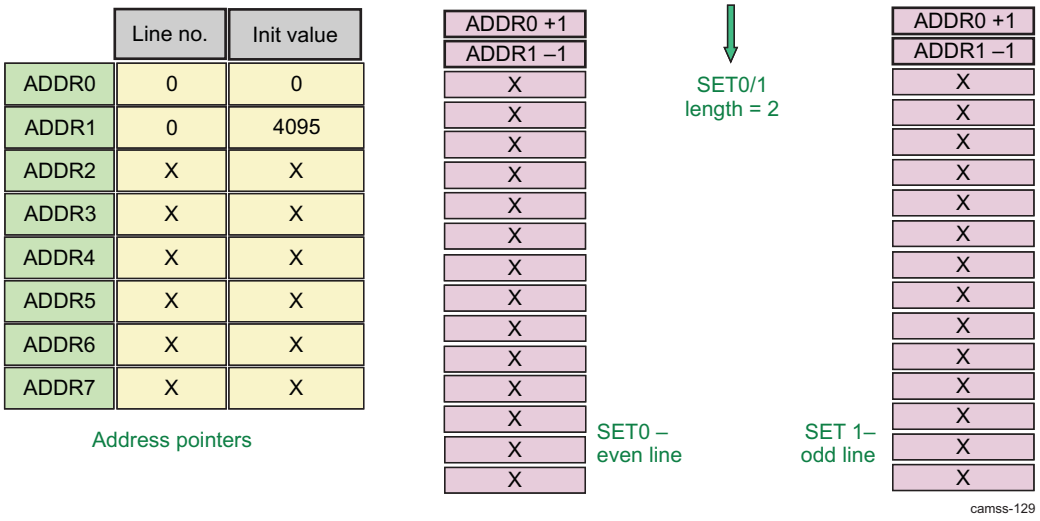
Figure 228. ISS ISP ISIF Conventional Read-Out Pattern



The following examples show the programmability of the data formatter with a 2-tap analog front end (AFE):

- Register settings:
  - ISIF_FMTCFG[0] FMTEN = 0x1
  - ISIF_FMTCFG[1] FMTCBL = 0x0
  - ISIF_FMTCFG[11:8] FMTAINC = 0x0 (add or subtract 1)
  - ISIF_FMTCFG[5:4] LNUM = 0x0
- One input line (4096) - one output line with left and right read-out
- Input - First pixel, last pixel, first pixel + 1, last pixel – 1, and so on
- Input - 0, 4095, 1, 4094, 2, 4093, 3, 4092, ..., 2047 and 2048
- Output - 0, 1, 2, 3, ..., 4094 and 4095

Figure 229. ISS ISP ISIF Conventional Read-Out Pattern With 2-tap AFE



3.3.6.6.3 ISS ISP ISIF Combine the Divided Input Lines

The formatter can gather the divided input lines and organize a single line. Figure 230 shows an example generating a single output line from three input lines and masking two out of three HD input pulses.

Figure 230. ISS ISP ISIF Combine Three Input Lines Into Single Line

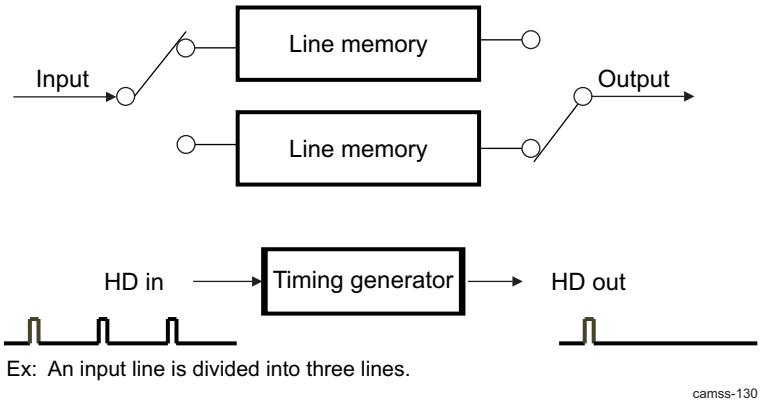


Figure 231 is an example that shows the register setting.

- Multiple input lines combined into one output line (use three input lines to one output line in this example)
- Input - 4, 5, 10, 11, ..., 2, 3, 8, 9, ..., 0, 1, 6, 7, ... (three lines read from sensor)
- Output - 0, 1, 2, 3, ... (1 line output)
- Register settings:
  - ISIF_FMTCFG[0] FMTEN = 0x1
  - ISIF_FMTCFG[1] FMTCBL = 0x1
  - ISIF_FMTCFG[11:8] FMTAINC = 0x5 (add or subtract 6)
  - ISIF_FMTCFG[5:4] LNUM = 0x2
  - SETs recycled based on LNUM

Figure 231. ISS ISP ISIF Example of Combining Three Input Lines Into a Single Line

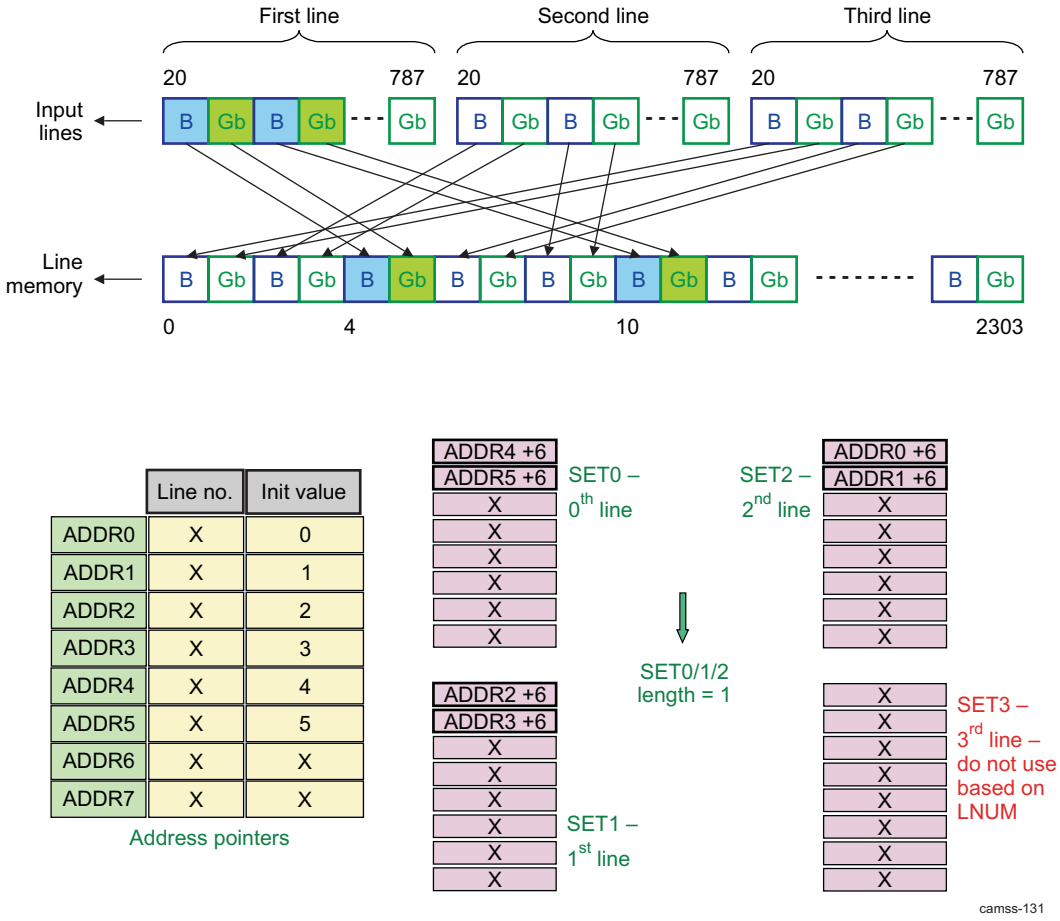


Table 201. ISS ISP ISIF Example of Combining Three Input Lines Into a Single Line: Register Setting Example

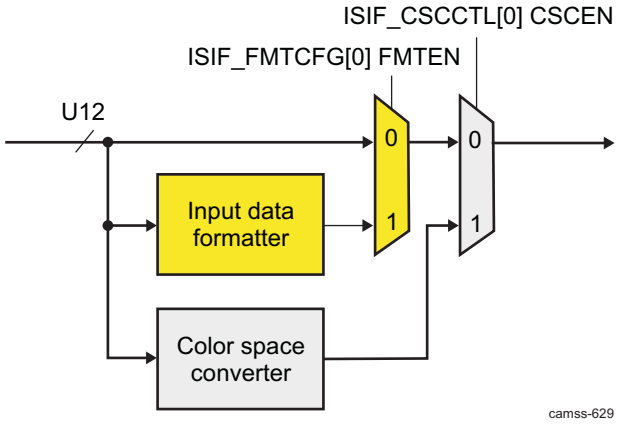
Step	Configuration Required	Size
Formatter enable	ISIF_FMTCFG[0] FMTEN	1
Combine input lines.	ISIF_FMTCFG[1] FMTCBL	1
Address increment = FMTAINC + 1 = 6	ISIF_FMTCFG[11:8] FMTAINC	5
The first valid pixel of a divided line	ISIF_FMTSPH[12:0] FMTSPH	20
Valid length of a divided line = FMTLNH + 1 = 768	ISIF_FMTLNH[12:0] FMTLNH	767
The first valid divided line	ISIF_FMTLSV[12:0] FMTSLV	16

Table 201. ISS ISP ISIF Example of Combining Three Input Lines Into a Single Line: Register Setting Example (continued)

Step	Configuration Required	Size
The number of the valid divided lines = FMTLNV + 1 = 4590	ISIF_FMTLNV[14:0] FMTLNV	4589
The length of an organized line = (FMTLNH + 1) x (LNUM + 1) = 2304	ISIF_FMTRLEN[12:0] FMTRLEN	2304
Split/combine line number = LNUM + 1 = 3.	ISIF_FMTCFG[5:4] LNUM	2
Number of PGM entries for SET0 = FMTPLEN0 + 1 = 2	ISIF_FMTPLEN[3:0] FMTPLEN0	1
Number of PGM entries for SET1 = FMTPLEN1 + 1 = 2	ISIF_FMTPLEN[7:4] FMTPLEN1	1
Number of PGM entries for SET2 = FMTPLEN2 + 1 = 2	ISIF_FMTPLEN[10:8] FMTPLEN2	1
Address Pointer 0, INIT = 0	ISIF_FMTAPTR0[12:0] INIT	0
Address Pointer 1, INIT = 1	ISIF_FMTAPTR1[12:0] INIT	1
Address Pointer 2, INIT = 2	ISIF_FMTAPTR2[12:0] INIT	2
Address Pointer 3, INIT = 3	ISIF_FMTAPTR3[12:0] INIT	3
Address Pointer 4, INIT = 4	ISIF_FMTAPTR4[12:0] INIT	4
Address Pointer 5, INIT = 5	ISIF_FMTAPTR5[12:0] INIT	5
Program 0 Valid flag	ISIF_FMTPGMVF0[0] PGM00EN	1
Program 1 Valid flag	ISIF_FMTPGMVF0[1] PGM01EN	1
Program 8 Valid flag	ISIF_FMTPGMVF0[8] PGM08EN	1
Program 9 Valid flag	ISIF_FMTPGMVF0[9] PGM09EN	1
Program 16 Valid flag	ISIF_FMTPGMVF1[0] PGM16EN	1
Program 17 Valid flag	ISIF_FMTPGMVF1[1] PGM17EN	1
Increment Address pointer = 0x0 Program 0 Address pointer = ADDR4 + 6	ISIF_FMTPGMAPU0[0] PGM0UPDT ISIF_FMTPGMAPS0[3:0] PGM0APTR	4
Increment Address pointer = 0x0 Program 1 Address pointer = ADDR5 + 6	ISIF_FMTPGMAPU0[1] PGM1UPDT ISIF_FMTPGMAPS0[7:4] PGM1APTR	5
Increment Address pointer = 0x0 Program 8 Address pointer = ADDR2 + 6	ISIF_FMTPGMAPU0[8] PGM8UPDT ISIF_FMTPGMAPS2[3:0] PGM8APTR	2
Increment Address pointer = 0x0 Program 9 Address pointer = ADDR3 + 6	ISIF_FMTPGMAPU0[9] PGM9UPDT ISIF_FMTPGMAPS2[7:4] PGM9APTR	3
Increment Address pointer = 0x0 Program 16 Address pointer = ADDR0 + 6	ISIF_FMTPGMAPU1[1] PGM17UPDT ISIF_FMTPGMAPS4[3:0] PGM16APTR	0
Increment Address pointer = 0x0 Program 17 Address pointer = ADDR1 + 6	ISIF_FMTPGMAPU1[0] PGM16UPDT ISIF_FMTPGMAPS4[7:4] PGM17APTR	1

3.3.6.7 ISS ISP ISIF Color Space Converter

Figure 232. ISS ISP ISIF Color Space Converter Block Diagram

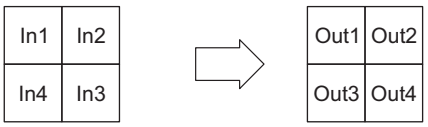


The color space converter (CSC) includes four 8-bit × 12-bit multipliers and one adder for the color space conversion. These multiplier/adder units are used for the operation described in Figure 233. Data are taken from two input lines during the operation.

Coefficients are signed 8-bit (decimal is 5 bits). Coefficients are set through the following registers:

- Coefficient M00: ISIF_CSCM0[7:0] CSCM00
- Coefficient M01: ISIF_CSCM0[15:8] CSCM01
- Coefficient M02: ISIF_CSCM1[7:0] CSCM02
- Coefficient M03: ISIF_CSCM1[15:8] CSCM03
- Coefficient M10: ISIF_CSCM2[7:0] CSCM10
- Coefficient M11: ISIF_CSCM2[15:8] CSCM11
- Coefficient M12: ISIF_CSCM3[7:0] CSCM12
- Coefficient M13: ISIF_CSCM3[15:8] CSCM13
- Coefficient M20: ISIF_CSCM4[7:0] CSCM20
- Coefficient M21: ISIF_CSCM4[15:8] CSCM21
- Coefficient M22: ISIF_CSCM5[7:0] CSCM22
- Coefficient M23: ISIF_CSCM5[15:8] CSCM23
- Coefficient M30: ISIF_CSCM6[7:0] CSCM30
- Coefficient M31: ISIF_CSCM6[15:8] CSCM31
- Coefficient M32: ISIF_CSCM7[7:0] CSCM32
- Coefficient M33: ISIF_CSCM7[15:8] CSCM33

Figure 233. ISS ISP ISIF Color Space Converter Operation



$$\begin{pmatrix} Out1 \\ Out2 \\ Out3 \\ Out4 \end{pmatrix} = \begin{pmatrix} M00 & M01 & M02 & M03 \\ M10 & M11 & M12 & M13 \\ M20 & M21 & M22 & M23 \\ M30 & M31 & M32 & M33 \end{pmatrix} \begin{pmatrix} In1 \\ In2 \\ In3 \\ In4 \end{pmatrix}$$

M00–M33: Signed 8-bit data with 5-bit decimal the value range –4=< Mxx<4

camss-618

The CSC can convert CMYG filtered CCD data to Bayer matrix (RGBG) data, as shown in Figure 234.

Figure 234. ISS ISP ISIF Color Space Converter Operation: CMYG to RGBG

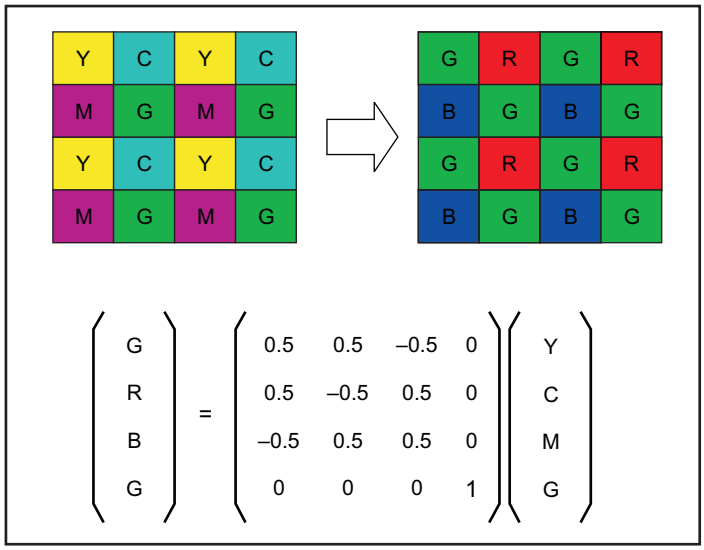


Figure 235 through Figure 237 show which input pixels are used for the operation. There is one-line latency between the input and the output.

Figure 235. ISS ISP ISIF Color Space Conversion Example

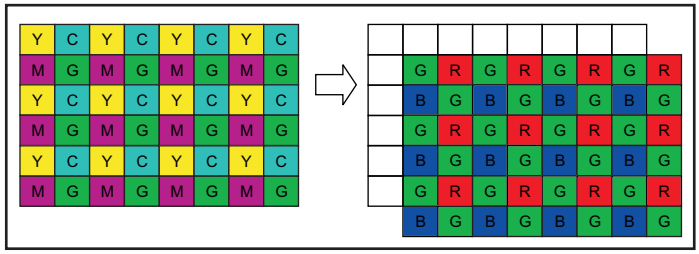


Figure 236. ISS ISP ISIF 1st Pixel/1st Line Generation

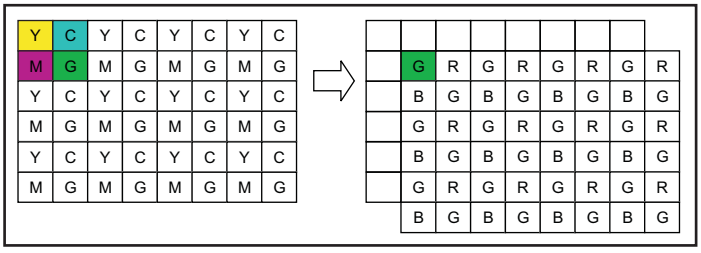
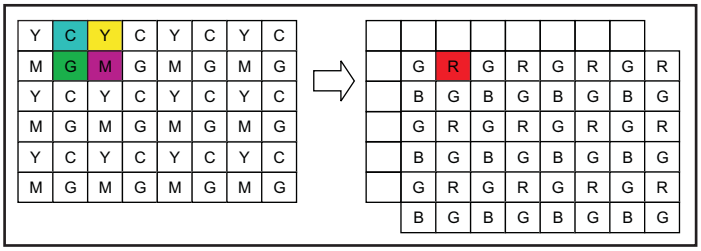


Figure 237. ISS ISP ISIF 2nd Pixel/1st Line Generation



As shown in Figure 238 through Figure 241, the operation for the last pixel and the second last pixel uses the same input data.

Figure 238. ISS ISP ISIF 2nd Last Pixel/1st Line Generation

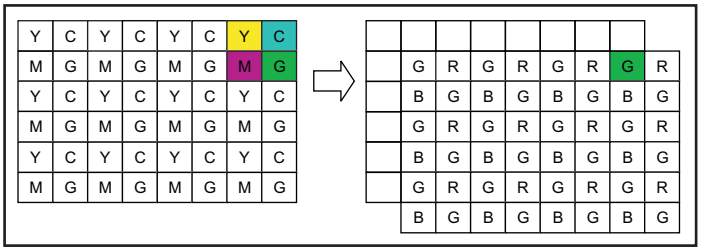


Figure 239. ISS ISP ISIF Last Pixel/1st Line Generation

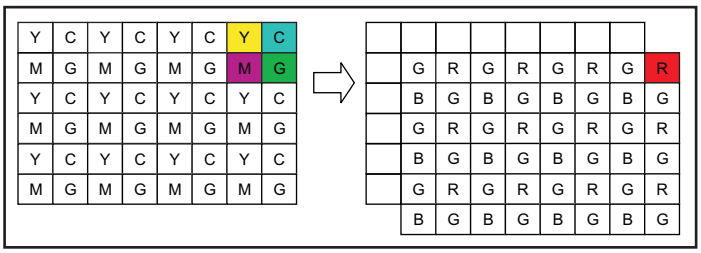


Figure 240. ISS ISP ISIF 1st Pixel/Last Line Generation

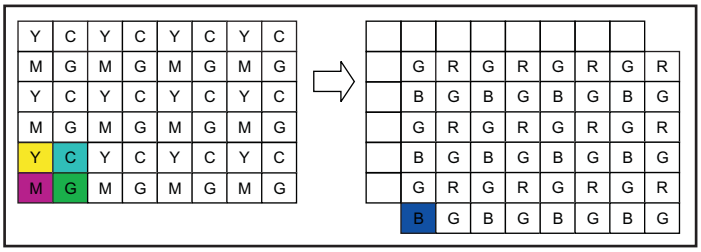
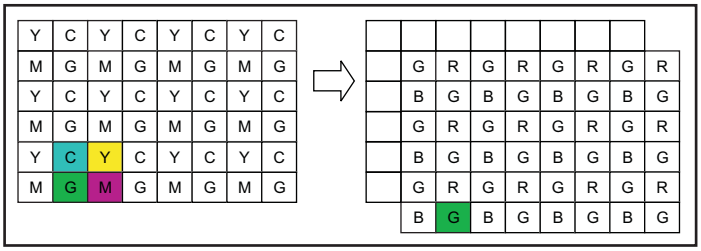


Figure 241. ISS ISP ISIF 2nd Pixel/Last Line Generation



Also, the operation for the last line and the second last line uses the same input data (see Figure 242 and Figure 243).

Figure 242. ISS ISP ISIF 2nd Last Pixel/Last Line Generation

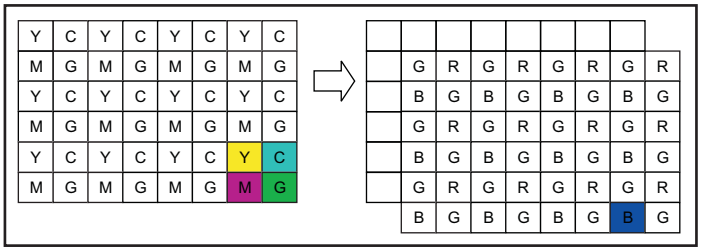
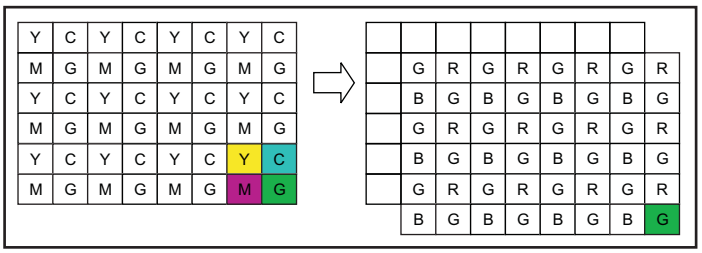


Figure 243. ISS ISP ISIF Last Pixel/Last Line Generation



In addition to the registers specific to the color space converter, some of the registers are shared with the input data formatter to configure the valid area:

- ISIF_FMTSPH
- ISIF_FMTLNH
- ISIF_FMTLSV
- ISIF_FMTLNV

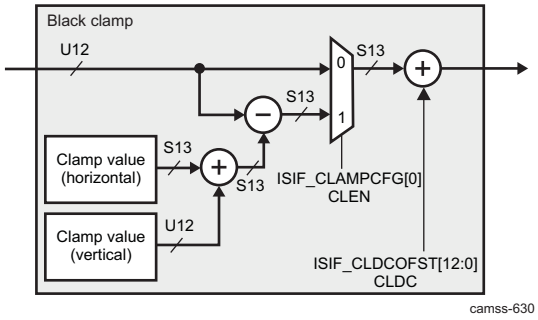
There must be at least 1 invalid pixel at the end of the line and one invalid line at the end of the frame.

To enable the color space conversion, set the ISIF_CSCCTL[0] CSCEN bit to 1.

3.3.6.8 ISS ISP ISIF Black Clamp

NOTE: For the memory access locations of the ISIF clamp, see Section 3.3.8.

Figure 244. ISS ISP ISIF Black Clamp Block Diagram



The clamp value is calculated based on the pixel value of the OB region of the sensor. The clamp value is calculated separately for horizontal and vertical directions to compensate the offset drift in both horizontal and vertical directions. The sum of the horizontal and vertical clamp values is subtracted from the image data, and then the additional DC offset is added (the ISIF_CLDCOFST[12:0] CLDC bit field, an S13Q0 value). This value is added whether the black clamp module is enabled or not.

The horizontal clamp is disabled through the ISIF_CLAMPCFG[2:1] CLHMD bit field.

To enable the black clamp module, set the ISIF_CLAMPCFG[0] CLEN bit to 1. The ISIF_PPLN[15:0] PPLN bit field sets the pixel per line, and the number of pixel clock periods in one line HD period equals PPLN + 1 pixel clock. The ISIF_PPLN[15:0] bit field is not used when the input is are already HD/VD.

3.3.6.8.1 ISS ISP ISIF Clamp Value for Horizontal Direction

The clamp value for horizontal direction is calculated using the pixel values at the upper OB region.

The maximum pixel value to be used for the clamp value calculation can be limited to 1023 if the pixel value limitation is enabled (ISIF_CLHWIN0[6] CLHLMT = 1).

Clamp value calculation for horizontal direction can be disabled in case there is no upper OB region. The operating modes are:

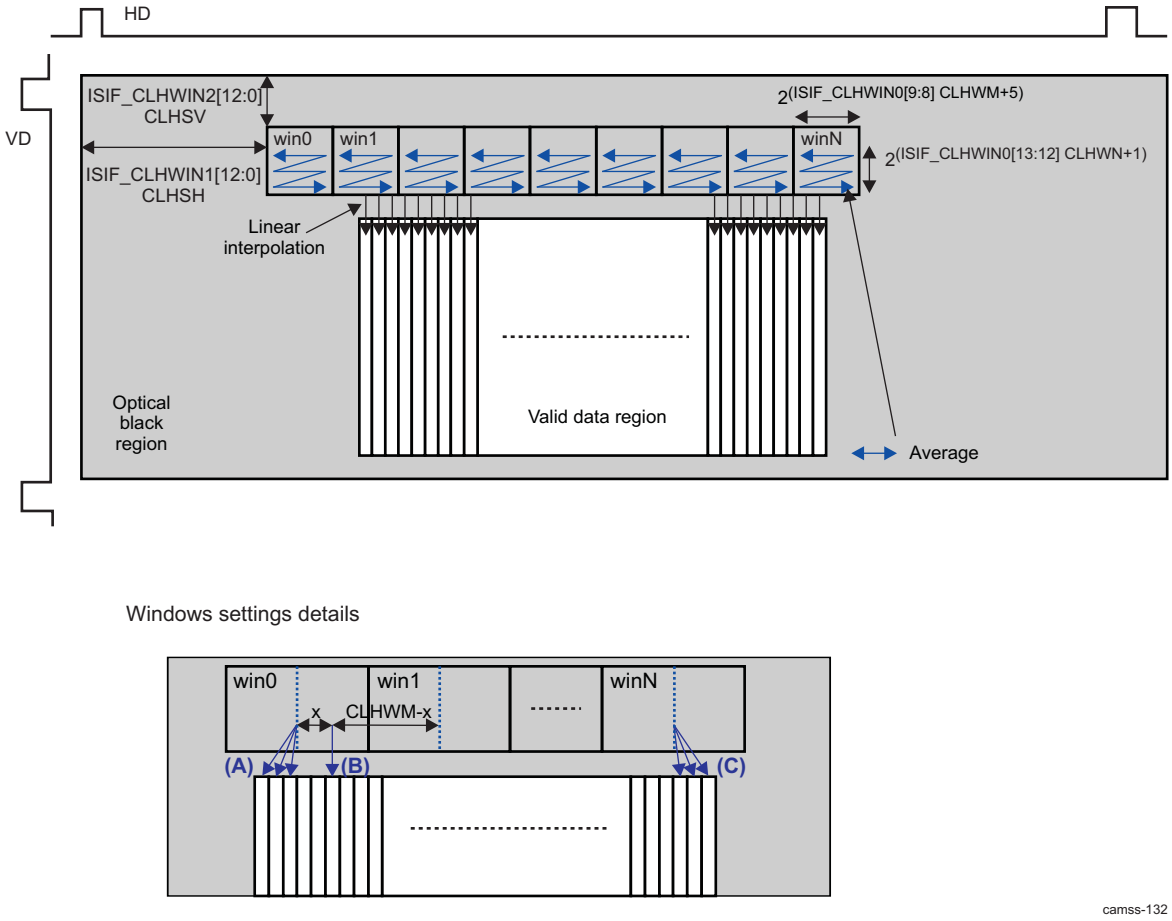
- Horizontal clamp value calculation is enabled. The calculated horizontal clamp value is subtracted from the image data along with the vertical clamp value (ISIF_CLAMPCFG[2:1] CLHMD = 0x1).
- Horizontal clamp value is not updated. The horizontal clamp value used for the previous image is subtracted from the image data along with the vertical clamp value (ISIF_CLAMPCFG[2:1] CLHMD = 0x2).
- Horizontal clamp value is not updated. Only the vertical clamp value is subtracted from the image data (ISIF_CLAMPCFG[2:1] CLHMD = 0x0).

The number of windows in a row is set with the ISIF_CLHWIN0[4:0] CLHWC bit field.

Up to 32 windows in a row can be set for clamp value calculation. All the windows have the same size in a format [$2^{(ISIF\_CLHWIN0[9:8]\ CLHWM+5)}$] pixels by [$2^{(ISIF\_CLHWIN0[13:12]\ CLHWN+1)}$] lines.

The ISIF_CLHWIN2[12:0] CLHSV and ISIF_CLHWIN1[12:0] CLHSH bit fields enable setting the position of the first optical black clamp window in the frame. The pixel and line offset are in a range [0:8191]. The ISIF_HDW register sets the width of the HD.

Figure 245. ISS ISP ISIF Clamp Value for Horizontal Direction



The clamp value for horizontal direction calculation steps is:

1. Calculate the average of the pixel value in each window (ave_win$_0$ to ave_win$_N$).

Calculation Steps	win0	win1	win2	...	wini	wini + 1	...	winN
Average of the pixel value	ave_win$_0$	ave_win$_1$	ave_win$_2$		ave_win$_i$	ave_win$_{i+1}$		ave_win$_N$

2. Set the average of the left-most window or the right-most window as the base value B_V:
   - B_V = ave_win$_0$ (if ISIF_CLHWIN0[5] CLHWBS = 0x0, case 1)
   - B_V = ave_win$_N$ (if ISIF_CLHWIN0[5] CLHWBS = 0x1, case 2)
3. Subtract the base value from the average of each window. Use this value as a clamp value for each window.

Calculation Steps	win0	...	wini	wini+1	...	winN
Clamp value for each window	clamp_win$_0$ = ave_win$_0$- B_V		clamp_win$_i$ = ave_win$_i$-B_V	clamp_win$_{i+1}$ = ave_win$_{i+1}$- B_V		clamp_win$_N$ = ave_win$_N$-B_V

4. Acquire the horizontal distance (X and CLHWM – X) from the valid pixel to be processed to the center of the closest two windows.

5. Calculate the clamp value of the valid pixel by linear interpolation, using the clamp value of the closest two windows (i and i + 1).
   - Case 1: interpolated_clamp_win$_X$ = (clamp_win$_{i+1}$ – clamp_win$_i$) * X / CLHWM
   - Case 2: interpolated_clamp_win$_X$ = (clamp_win$_i$ – clamp_win$_N$) * (CLHWM - X) / CLHWM + (clamp_win$_{i+1}$ – clamp_win$_N$) * X / CLHWM
6. If the valid pixel is on the left of the center of the left-most window, the clamp value of the left-most window is applied. If the valid pixel is on the right of the center of the right-most window, the clamp value of the right-most window is applied.

The clamp values calculated (A), (B) and (C) are:

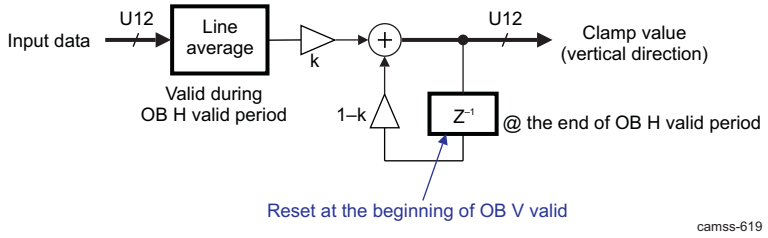
- Case1: Base is win0 (Left-most window: ISIF_CLHWIN0[5] CLHWBS = 0x0)
  - (A): zero
  - (B): (clamp_win$_{i+1}$ – clamp_win$_i$) * X / CLHWM
  - (C): (clamp_win$_N$ – clamp_win$_0$)
- Case2: Base is winN (Right-most window: ISIF_CLHWIN0[5] CLHWBS = 0x1)
  - (A): (clamp_win$_0$ – clamp_win$_N$)
  - (B): (clamp_win$_i$ – clamp_win$_N$) * (CLHWM – X) / CLHWM + (clamp_win$_{i+1}$ – clamp_win$_N$) * X / CLHWM
  - (C): zero

Each interpolated value interpolated_clamp_win$_X$ is then subtracted to the associated column.

3.3.6.8.2 ISS ISP ISIF Clamp Value for Vertical Direction

The clamp value for vertical direction is calculated using the pixel values at the left or right OB region. Line average is calculated for the OB H valid period ($2^{(ISIF\_CLVWIN0[2:0]\ CLVOBH\ +\ 1)}$). The averages for the previous lines are also added back to reduce the difference between the lines, as shown in Figure 246.

Figure 246. ISS ISP ISIF Clamp Value for Vertical Direction Calculation



- Clamp Value (V_n) = Line Average (V_n) * k + Clamp Value (V_1) * (1-k)
- k = ISIF_CLVWIN0[15:8] CLVCOEF

The position of the first vertical black clamp window is set with the ISIF_CLVWIN2[12:0] CLVSV and ISIF_CLVWIN1[12:0] CLVSH bit fields. The number of vertical windows is set with the ISIF_CLVWIN3[12:0] CLVOBV bit field. ISIF_VDW sets the width of the VD.

The accumulator, which holds the vertical clamp value for the previous line, is reset at the beginning of the OB V valid. The reset value can be selected through the ISIF_CLVWIN0[5:4] CLVRVSL bit field:

- ISIF_CLVWIN0[5:4] CLVRVSL = 0x0: The base value is calculated for horizontal direction (left-most window win0 or right-most winn set with ISIF_CLHWIN0[5] CLHWBS).
- ISIF_CLVWIN0[5:4] CLVRVSL = 0x1: The base value is set through the configuration register (ISIF_CLVRV[11:0] CLVRV).
- ISIF_CLVWIN0[5:4] CLVRVSL = 0x2: No update (same as the previous image)

The following figures show the OB valid settings and associated vertical clamp value calculation when OB region is at the left (see Figure 247) and when OB region is at the right (see Figure 248). Each line average value is subtracted from the associated line valid region data.

Figure 247. ISS ISP ISIF Clamp Value for Vertical Direction With OB Region at the Left

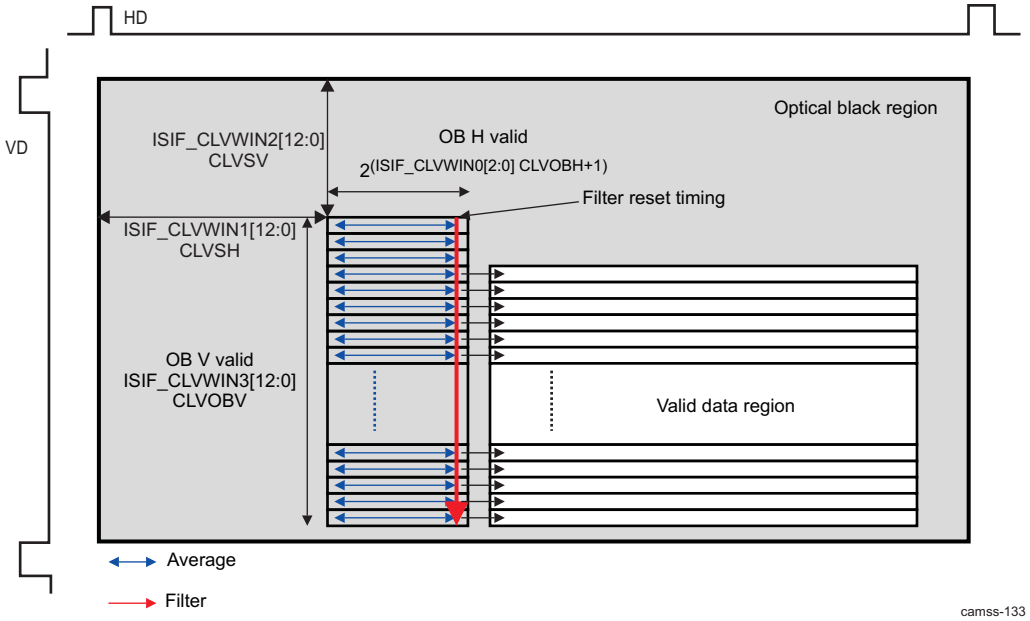
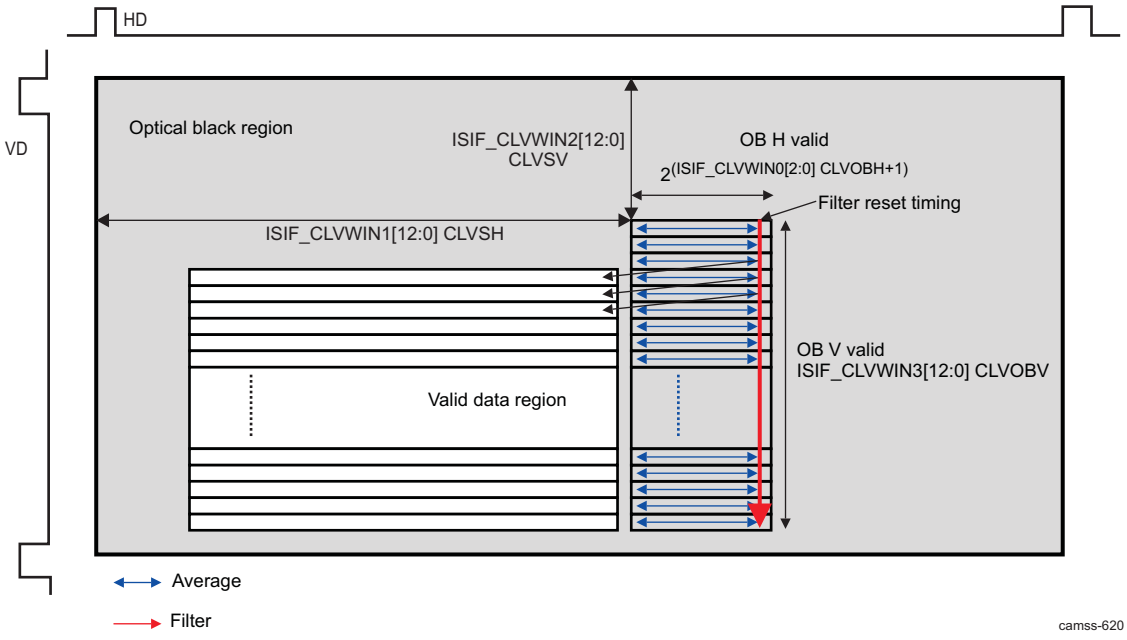


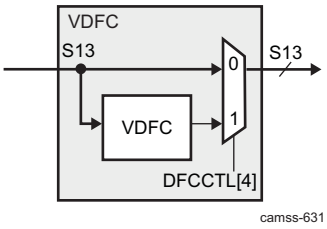
Figure 248. ISS ISP ISIF Clamp Value for Vertical Direction With OB Region at the Right



3.3.6.9 ISS ISP ISIF Vertical Line Defect Correction (VDFC)

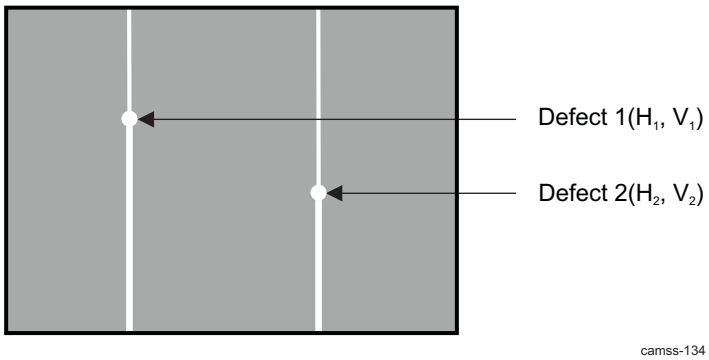
Figure 249 shows the block diagram of the vertical line defect (VDFC) correction.

Figure 249. ISS ISP ISIF Vertical Line Defect Correction Block Diagram



The VDFC block can correct up to eight vertical line defects (see Figure 250).

Figure 250. ISS ISP ISIF Vertical Line Defects



The correction method is common to all the defects and can be selected through the ISIF_DFCCTL[6:5] VDFCSL bit field.

There are two different methods to correct vertical line defects (ISIF_DFCCTL[6:5] VDFCSL):

- Method 1: Data is replaced by an average (ISIF_DFCCTL[6:5] VDFCSL = 0x2):
  - The defect is replaced by the average of pixel (i – 2) and pixel (i + 2)
- Method 2: Data is subtracted by a defect level (ISIF_DFCCTL[6:5] VDFCSL = 0x0 or 0x1):
  - A saturation level is defined in the ISIF_VDFSATLV[11:0] VDFSLV bit field.
  - The coordinates of the defect:
    - Are defined in the ISIF_DFCMEM0[12:0] DFCMEM0 and ISIF_DFCMEM1[12:0] DFCMEM1 bit fields
    - Are 13 bits wide for horizontal and vertical direction, so an image size up to 8192 × 8192 is supported
  - If the data is not saturated (data VDFSLV):
    - The defect is corrected by subtracting the defect level. A different defect level is defined for:
      - The point of the defect (V = Vdefect): SUB1 defect level is defined in the ISIF_DFCMEM2[7:0] DFCMEM2 bit field.
      - The pixels lower than the defect (V Vdefect): SUB2 defect level is defined in the ISIF_DFCMEM3[7:0] DFCMEM3 bit field.
      - The pixels upper than the defect (V Vdefect): Defect level is defined in the ISIF_DFCMEM4[7:0] DFCMEM4 bit field.
    - Each defect level (value to be subtracted from the data) described previously can be up-shifted through the ISIF_DFCCTL[10:8] VDFLSFT bit field.
    - Vertical line defect correction for upper pixels can be disabled through the ISIF_DFCCTL[7] VDFCUDA bit.
  - If the data is saturated (VDFSLV), there are two possibilities:

- ISIF_DFCCTL[6:5] VDFCSL = 0x0: Data is simply fed through (not subtracted).
- ISIF_DFCCTL[6:5] VDFCSL = 0x1: Horizontal interpolation ((i 2) + (i + 2))/2 (data is replaced by interpolation or data is subtracted with interpolation)

The ISIF_LPFR register sets the number of half lines per frame or field: VD period = (L PFR+ 1)/2 lines. LPFR is not used when HD and VD are inputs.

The following paragraphs concern only method 2 correction.

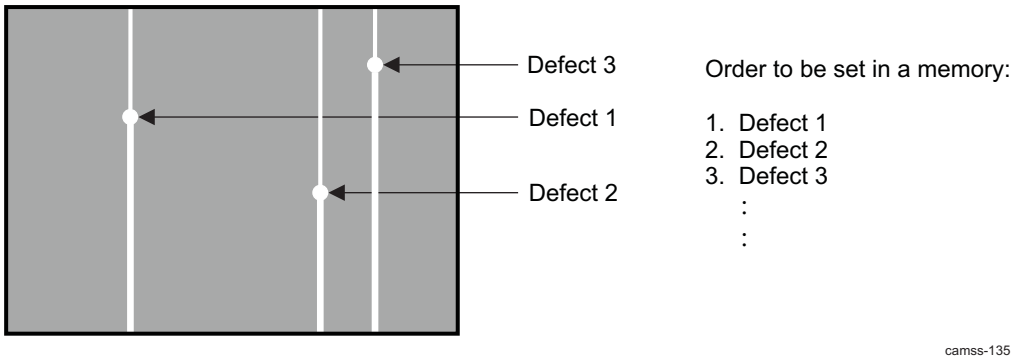
The coordinates of the defects and the defect levels to be subtracted from the data must be set to the processing shown in Table 202.

Table 202. ISS ISP ISIF Vertical Line Defect Table in Memory

Bit	Defect Information
12:0	Vertical position of the defects
25:13	Horizontal position of the defects
33:26	Defect level of the vertical line defect position (V = Vdefect)
41:34	Defect level of the pixels upper than the vertical line defect (V Vdefect)
49:42	Defect level of the pixels lower than the vertical line defect (V Vdefect)

The defect must be set from left to right, as shown in Figure 251.

Figure 251. ISS ISP ISIF Vertical Line Defects



Vertical line defect correction is enable by setting the ISIF_DFCCTL[4] VDFCEN bit to 1, but the procedure explained in Section 3.3.6.9.1, *ISS ISP ISIF Vertical Line Defect Table Update Procedure*, must be respected.

3.3.6.9.1 ISS ISP ISIF Vertical Line Defect Table Update Procedure

The following procedure must be respected to write the vertical line defect table in memory.

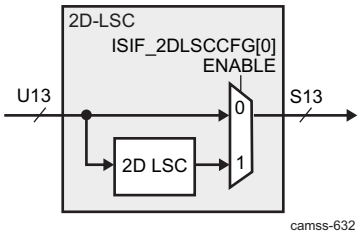
1. ISIF_DFCMEMCTL[4] DFCMCLR = 0x1
2. Ensure that ISIF_DFCCTL[4] VDFCEN is disabled (0x0).
3. Write the V coordinate of the first defect to the ISIF_DFCMEM0[12:0] DFCMEM0 bit field.
4. Write the H coordinate of the first defect to the ISIF_DFCMEM1[12:0] DFCMEM1 bit field.
5. Set the defect level to:
   - ISIF_DFCMEM2[7:0] DFCMEM2
   - ISIF_DFCMEM3[7:0] DFCMEM3
   - ISIF_DFCMEM4[7:0] DFCMEM4

6. Set the ISIF_DFCMEMCTL[0] DFCMWR bit to 1 with the ISIF_DFCMEMCTL[2] DFCMARST bit set to 1.
7. Wait until the ISIF_DFCMEMCTL[0] DFCMWR bit is cleared.
8. Write the next data to:
   - ISIF_DFCMEM0[12:0] DFCMEM0
   - ISIF_DFCMEM1[12:0] DFCMEM1
   - ISIF_DFCMEM2[7:0] DFCMEM2
   - ISIF_DFCMEM3[7:0] DFCMEM3
   - ISIF_DFCMEM4[7:0] DFCMEM4
9. Set the ISIF_DFCMEMCTL[0] DFCMWR bit to 1 with the ISIF_DFCMEMCTL[2] DFCMARST bit cleared.
10. Repeat 4~5 times until all entries (up to 8) are written to the vertical line defect table.
11. In case the defect entry is less than 8, an extra write cycle is required to fill the next table location with a certain value.
12. Clear the ISIF_DFCMEM0[12:0] DFCMEM0 bit field to all 0, set the ISIF_DFCMEM1[12:0] DFCMEM1 to all 1, and set the ISIF_DFCMEMCTL[0] DFCMWR bit to 1 with the ISIF_DFCMEMCTL[2] DFCMARST bit cleared.
13. Enable VDFC by setting the ISIF_DFCCTL[1] VDFCEN bit.

3.3.6.10 ISS ISP ISIF Lens Shading Correction (2D-LSC)

NOTE: For the memory access locations of the 2D-LSC table, see Section 3.3.8.

Figure 252. ISS ISP ISIF 2D-LSC Block Diagram



LSC is useful for correcting optical artifacts that cause image brightness to decrease starting from the center of the image and going out to the edges.

The LSC module implements a per pixel offset and gain adjustment in the RAW Bayer domain (2 × 2 color pattern). The offset is applied before gain multiplication.

The offset and gains are stored in a LUT, which is stored in SDRAM and is loaded in real time. The submodule prefetches the data from SDRAM such that no underflow occurs. Underflow occurs when the offset and gain data required for the current pixel are not available.

The data stored in the LUT is downsampled; that is, there is no gain or offset per pixel. The downsampling factor is programmable. High downsampling ratios lead to a smaller LUT, lower accuracy, and lower memory bandwidth. A low downsampling ratio leads to a bigger LUT, higher accuracy, and higher memory bandwidth.

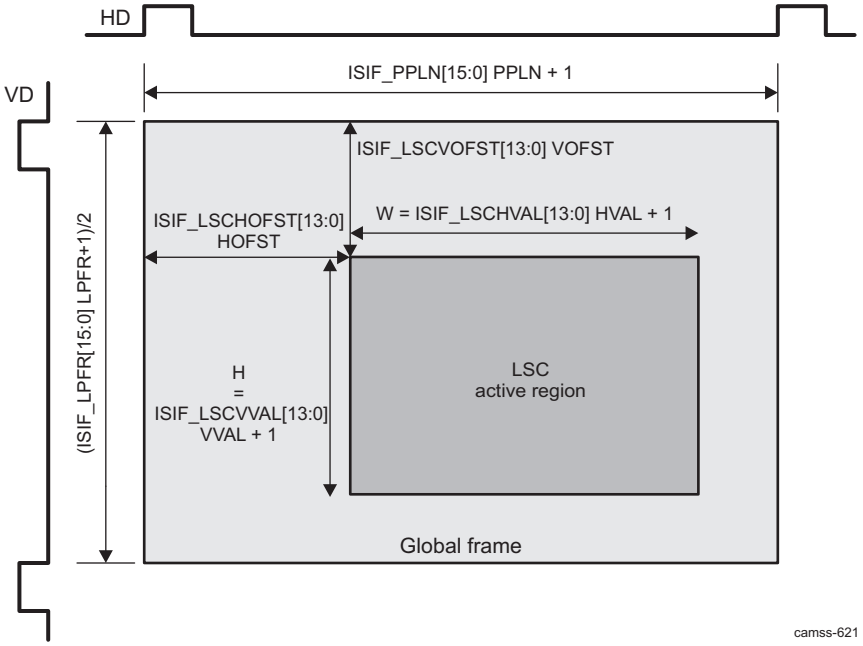
When the offset and gain values are loaded, they are upscaled to the incoming image resolution. The missing table values are computed through bilinear interpolation.

To enable the 2D-LSC module, set the ISIF_2DLSCCFG[0] ENABLE bit to 1.

3.3.6.10.1 ISS ISP ISIF 2D-LSC Active Region Settings

The gain and offset maps are internally up-sampled to full resolution before being applied to the image. In order to account for all the possible cropping schemes and zoom ratios, the 2D-LSC can be configured such that a single gain map can be stored in memory that maps to sensor lens. The 2D-LSC active region is defined by Figure 253.

Figure 253. ISS ISP ISIF 2D-LSC Active Region for ISIF Input Frame



3.3.6.10.1.1 ISS ISP ISIF 2D-LSC Gain and Offset Tables

The gain and offset map are MxN downsampled:

- M is the horizontal sampling factor.
- N is the vertical sampling factor.
- M and N are {8, 16, 32, 64, 128} independently.
- N = M. M is set in the ISIF_2DLSCCFG[14:12] GAIN_MODE_M bit field.
- N is set in the ISIF_2DLSCCFG[10:8] GAIN_MODE_N bit field.

The starting point of the preconfigured lens shading map can be modified in software to align with the ISIF input image frame. The location of the gain and offset mask data in memory is specified by :

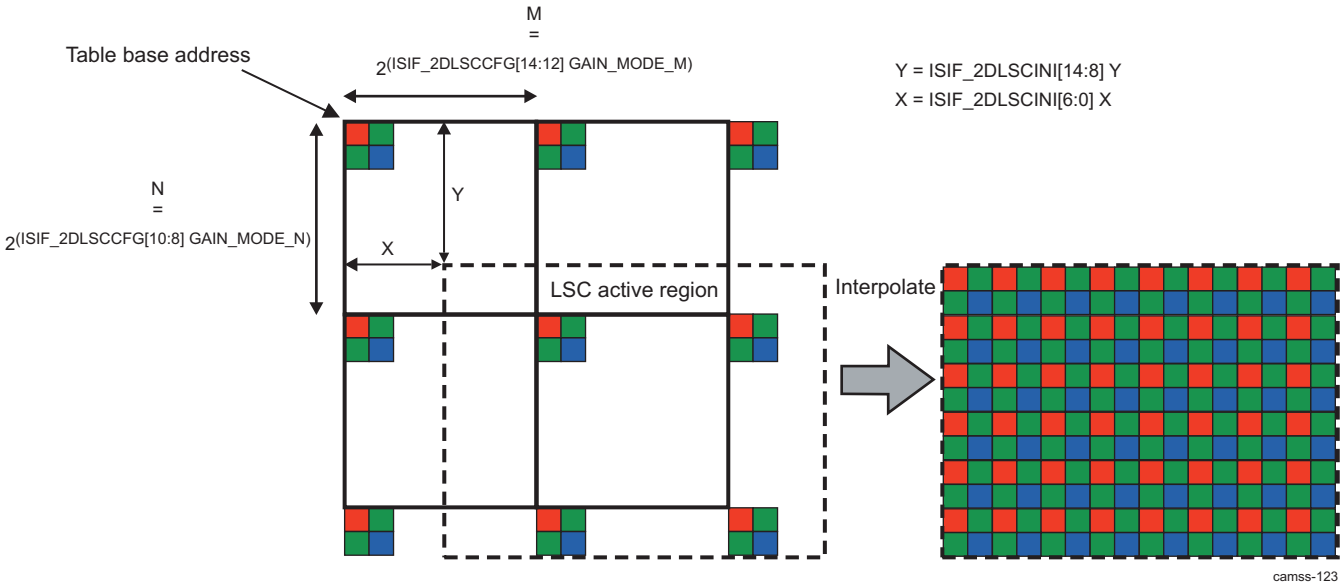
- For gain map:
  - Table base address: ISIF_2DLSCGRBU[15:0] BASE31_16 and ISIF_2DLSCGRBL[15:0] BASE15_0
- For offset map:
  - Table base address: ISIF_2DLSCORBU[15:0] BASE and ISIF_2DLSCORBL[15:0] BASE

The input address must be aligned to a 4-byte boundary.

In the full resolution case, the address is set to the beginning of the map. When the LSC active region is defined over a cropped region of the full image, the SDRAM input address can be set to the upper-left corner of the grid closest to the beginning of the active region, and the ISIF_2DLSCINI[6:0] X and ISIF_2DLSCINI[14:8] Y bit fields mark the offset into the upsampled gain and offset map where the active region begins. Figure 254 shows the LSC active region with respect to the gain and offset map grid. Because (X, Y) deals with the pixel phase inside a gain and offset map grid, X and Y must each be less than M and N, respectively.

Before applying the gain or offset table, the table is internally upsampled and interpolated back to full resolution via bilinear interpolation.

Figure 254. ISS ISP ISIF Gain and Offset Mask Upsampling via Bilinear Interpolation



The gain table format is set in the ISIF_2DLSCCFG[3:1] GAIN_FORMAT bit field. 8-bit entries are supported in the gain map (in U8Q8, U8Q7, U8Q6, and U8Q5 format with optional base of 1.0 to shift the range up).

8-bit entries are supported in the offset map (in S8Q0 format). An optional shifting up for offsets is possible. The shift up value is selected through the ISIF_2DLSCOFST[6:4] OFSTSFT bit field. A scaling factor for offsets is defined in the ISIF_2DLSCOFST[15:8] OFSTSF bit field.

The offset control in the 2D-LSC module can be enabled or disabled through the ISIF_2DLSCOFST[0] OFSTEN bit.

For an LSC active region size of W × H (the output and input are the same size), and the gain and offset maps are MxN downsampled, a table with the following values is needed:

- (ceil[(X + W) / M] + 1) x (ceil[(Y + H) / N] + 1) × 4 bytes data in external memory organized as:
  - (ceil[(InitX + W) / M] +1) lines of data
  - Each line having at least (ceil[(InitY + H) / N] + 1) × 4 data points

  Extra data at end of each line can be skipped by the line offset register parameter:
  - For gain map:
    - Line offset: ISIF_2DLSCGROF[15:0] OFFSET
  - For offset map:
    - Line offset: ISIF_2DLSCOROF[15:0] OFFSET
  - Each line offset must start at a 32-bit aligned boundary.

3.3.6.10.1.2 ISS ISP ISIF 2D-LSC Gain and Offset Table Upsampling

Upsampling of the pixel-by-pixel gains is performed by locating the four same-color anchors for each destination gain value and applying bilinear interpolation.

The gain and offset mask function is neutral to color pattern. The starting color of the gain and offset mask must be consistent with the starting color of the image, and can be any color. To align starting colors, the X and Y values must be even. The 2D-LSC engine upsamples each phase of the mask data as a separate plane and applies the upsampled mask to the image with the same color phasing. In other words, the red gains are interpolated with red gains, and applied to the red input pixels. The same is done for each of the other three colors in the color pattern. The 2D-LSC module is designed to work with Bayer CFA data, having the R/Gr/Gb/B color pattern. For the purpose of functional description, assume red is the starting color, but any other starting color or other 2 × 2 pattern can be used by placing color gains in the appropriate order.

3.3.6.10.1.3 ISS ISP ISIF Application of Gain and Offset to Image Pixels

The gain value interpolated for each pixel is multiplied with a corresponding input pixel. An offset is applied before the gain. The product is rounded to the nearest integer and then clipped or saturated to the valid range of 13 bits.

The following equation describes the operation of the LSC in terms of offset and gain:

out[x,y] = (in[x,y] + a*(ofst[x,y] T)) * gain[x,y] (5)

- in[x, y] are the input pixels, 13 bits signed.
- ofst[x, y] are the upsampled offset points.
- T is the upshift value applied to the result of the offset interpolation points (0 – 5) set through the ISIF_2DLSCOFST[6:4] OFSTSFT bit field.
- a is the offset gain value in U8Q7 format set through the ISIF_2DLSCOFST[15:8] OFSTSF bit field.
- gain[x, y] are the upsampled gain points.
- out[x, y] are the resulting output pixels, 13 bits signed.

3.3.6.10.1.4 ISS ISP ISIF Enabling/ Disabling the 2D-LSC Module

LSC operates on a single frame or continuously, depending on the firmware programming.

Upon power-on reset (POR), the 2D-LSC module is disabled and input pixels are copied to the output, bypassing any shading operation.

When enabling or disabling the 2D-LSC, caution must be taken on the timing of register modifications. To avoid causing a prefetch error or other unexpected behavior, the following safeguards must be implemented:

1. While configuring the 2D-LSC registers, the input clock into the ISIF should be toggling.
2. All of the 2D-LSC registers must be configured appropriately before enabling the ISIF_2DLSCCFG[0] ENABLE bit.
3. After setting the ENABLE bit to 1, the hardware immediately begins fetching the first two rows of gain and offset data entries from external memory. When this is complete, the ISIF_2DLSCIRQST[2] PREFETCH_COMPETED status flag is set.

NOTE: If the ENABLE bit is disabled before the ISIF_2DLSCIRQST[3] SOF status flag is set, the ISIF_2DLSCIRQST[1] PREFETCH_ERROR flag is set and the state of the 2D-LSC submodule may lead to unexpected errors. Therefore, the ENABLE bit must not be disabled until after the ISIF_2DLSCIRQST[3] SOF status flag is set.

4. Appropriate gains and offsets are applied to the image pixels. Pixels outside the LSC active region are passed through unaltered. When the 2D-LSC operation on the active region completes, the ISIF_2DLSCIRQST[0] DONE status flag is set.
5. At this point:
   - If the ENABLE bit is still set to 1, the hardware immediately begins to prefetch the gain and offset data entries for the next frame and waits for the active region of the next frame to arrive.
   - If the ENABLE bit is set to 0, it stops LSC operation once the active region is passed, and goes into idle until the ENABLE bit is written to 1 again.

NOTE: To provide a mechanism for firmware to recover from the LSC module waiting indefinitely for the input image, if LSC_ENABLE is written to 0 after it has started gain/offset map prefetching, but before the LSC gets to the next active region, the LSC operation is aborted and turned idle, and any prefetched gain/offset entries are discarded. This can happen before or after the next start-of-frame.

NOTE: Therefore, because of the constraints set in point 3, the ENABLE register bit must be disabled only after the ISIF_2DLSCIRQST[3] SOF status flag is set and before the ISIF_2DLSCIRQST[0] DONE status flag is set for that same frame.

It is suggested that when the 2D-LSC or the whole ISIF needs to be disabled for switching modes, the ISIF_2DLSCIRQST[3] SOF interrupt be enabled so that software knows when it is safe to disable the 2D-LSC. Then the ISIF can be disabled after the ISIF_2DLSCIRQST[0] DONE status signal is set for that frame.

NOTE: The LSC_ENABLE bit, once written to 1, must not be cleared until at least one vpi_clk clock cycle after start-of-frame, to ensure correct processing.

3.3.6.10.1.5 ISS ISP ISIF 2D-LSC Events and Status Checking

The 2D-LSC submodule can generate events on a single interrupt line. These events are further remapped at the ISP level in ISP5_IRQENABLE_SET__0.ISIF_INT_3 - ISP5_IRQENABLE_SET__3.ISIF_INT3.

Four 2D-LSC events can be generated:

- DONE: LSC done. This event triggers when the LSC submodules transition from ACTIVE state to IDLE state.
- PREFETCH_ERROR: Gain table prefetch error. This event triggers when the tables stored in SDRAM are read too slowly. After this event is asserted, the LSC disables the LSC computation until the beginning of the next frame.
- PREFETCH_COMPLETE: Gain table prefetch complete. This event triggers when data prefetching from SDRAM completes. Data prefetching must complete by the time the first pixel of a frame comes. The event triggers when the buffer contains three full rows of data.
- SOF: This event signals the start of the LSC valid region. The LSC configuration registers for the next frame can be updated after the LSC SOF triggers.

The ISIF_2DLSCIRQEN register can be configured to select which events are masked and which are propagated to the LSC interrupt signal. The ISIF_2DLSCIRQST register can be read and cleared to identify which events have occurred.

In addition, the 2D-LSC module provides the following status bit:

- BUSY: This indicates that LSC has entered the active region vertically. This bit remains on during horizontal blanking, and turns off only after the entire active region of the current frame is processed.

3.3.6.10.1.6 ISS ISP ISIF Supported On-the-Fly 2D-LSC Configurations

The 2D-LSC prefetch memory is equal to 2 × 1536 × 32 bits. This memory is sized to fetch three lines of 8-bit gain and 8-bit offset × four color components per paxel. Given an image sensor of horizontal resolution H, there are floor [(H / ISIF_2DLSCCFG[14:12] GAIN_MODE_M) + 1] paxels per line, where M is the horizontal LSC paxel size.

Table 203 shows the LSC horizontal paxel size, which can be supported for different image sensor resolutions. When M = 8, some resolutions cannot be supported on the fly (orange-shaded cells in the table); the way to process such large images is to use vertical frame division.

Table 203. ISS ISP ISIF Supported On-the-Fly LSC Configurations

MPix	Aspect Ratio		Line Size	Horizontal LSC Paxel Size: M = $2^{[ISIF\_2DLSCCFG[14:12]\ GAIN\_MODE\_M]}$				
				8	16	32	64	128
Maximum	–	–	5376	2019	1011	507	255	129
16	16	9	5333	2003	1003	503	253	128
16	4	3	4619	1735	869	436	220	111
16	3	2	4899	1840	922	462	233	118
12	16	9	4619	1735	869	436	220	111
12	4	3	4000	1503	753	378	191	97
12	3	2	4243	1594	798	401	202	102
10	16	9	4216	1584	794	398	201	102
10	4	3	3651	1372	688	345	174	89
10	3	2	3873	1455	729	366	185	94
8	16	9	3771	1417	710	357	180	91
8	4	3	3266	1228	615	309	156	80
8	3	2	3464	1302	653	328	165	84

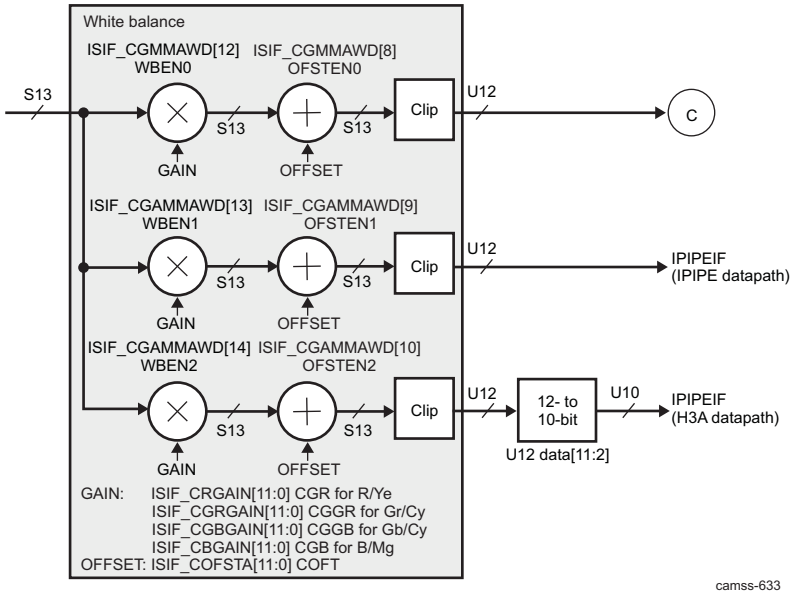
3.3.6.10.1.7 ISS ISP ISIF Bandwidth Requirements on BL Read Port

See Section 3.3.6.18.2 for details.

3.3.6.11 ISS ISP ISIF White Balance

Figure 255 shows the white balance block diagram.

Figure 255. ISS ISP ISIF White Balance Block Diagram



Color pattern settings are set through the ISIF_CCOLP register. Moreover, through this register the pixel position from 0 to 3 can be set to the needed Bayer universal camera filter color pattern (RGB/CYGM).

The CFA pattern can be in two modes, stripe or mosaic, and is set through the ISIF_CGAMMAWD[5] CFAP bit.

There are color-dependent gain controls for the three outputs: BL output, IPIPEIF (IPIPE path) output, and IPIPEIF (H3A path) output. Gain applied to each data is selected according to the pixel position and the color pattern settings. Gain factors are common for the three data paths. Gain is in U11Q9 format, which ranges from 0 to 3 + 511/512.

The gain factor is set through the following registers:

- R/Ye gain: ISIF_CRGAIN[11:0] CGR
- Gr/Cy gain: ISIF_CGRGAIN[11:0] CGGR
- Gb/Cy gain: ISIF_CGBGAIN[11:0] CGGB
- B/Mg gain: ISIF_CBGAIN[11:0] CGB

Gain control can be enabled or disabled individually for each path.

- Enable or disable gain for the BL path: ISIF_CGAMMAWD[12] WBEN0.
- Enable or disable gain for the IPIPEIF (IPIPE) path: ISIF_CGAMMAWD[13] WBEN1.
- Enable or disable gain for the IPIPEIF (H3A) path: ISIF_CGAMMAWD[14] WBEN2.

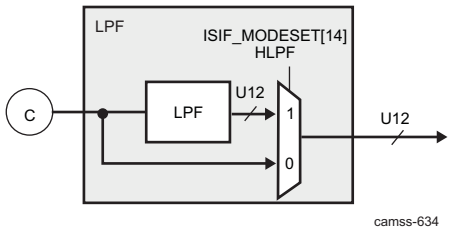
After the gain control, a single offset value can be added to each path individually. This offset is common for the three paths and is set through the ISIF_COFSTA[11:0] COFT bit field. The offset value is U12, which ranges from 0 to 4095. Data (S13) are then truncated to U12.

- Enable or disable offset for the BL path: ISIF_CGAMMAWD[8] OFSTEN0.
- Enable or disable offset for the IPIPEIF (IPIPE) path: ISIF_CGAMMAWD[9] OFSTEN1.
- Enable or disable offset for the IPIPEIF (H3A) path: ISIF_CGAMMAWD[10] OFSTEN2.

3.3.6.12 ISS ISP ISIF Low-Pass Filter (LPF)

Figure 256 shows the low-pass filter block diagram.

Figure 256. ISS ISP ISIF Low-Pass Filter Block Diagram



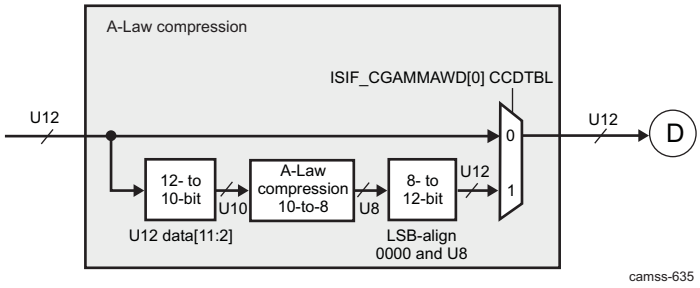
An optional horizontal low-pass anti-aliasing filter (LPF) can be applied (the ISIF_MODESET[14] HLPF bit) after reframing. The low-pass filter consists of a simple 3-tap (1/4, 1/2, and 1/4) filter. Two pixels on the left and two pixels on the right of each line are cropped if the filter is enabled. Use of the LPF is intended for bandwidth reduction if culling is enabled.

NOTE: For YUV data, the LPF must be disabled (ISIF_MODESET[14] HLPF = 0x0).

3.3.6.13 ISS ISP ISIF A-Law Compression

Figure 257shows the A-Law compression block diagram.

Figure 257. ISS ISP ISIF A-Law Compression Block Diagram



An optional 10-to-8-bit A-Law compression using a fixed A-Law table can be applied (ISIF_CGAMMAWD[0] CCDTBL) as the final processing stage. Using this compression causes the data width to be reduced to 8 bits and allows packing to 8 bits/pixel when saving to memory. Because data resolution can be greater than 10 bits at this stage, the 10 bits for input to the A-Law operation must be selected (ISIF_CGAMMAWD[4:1] GWDI).

The IPIPEIF module has an inverse A-Law table (A-Law decompression) option so that this nonlinear operation can be reversed if this saved data is to be read back in for further processing.

NOTE: Do not use A-Law compression (ISIF_CGAMMAWD[0] CCDTBL = 0) with YUV data.

Figure 258 shows the A-Law table diagram, and Figure 259 shows the A-Law table values.

Figure 258. ISS ISP ISIF A-Law Table Diagram

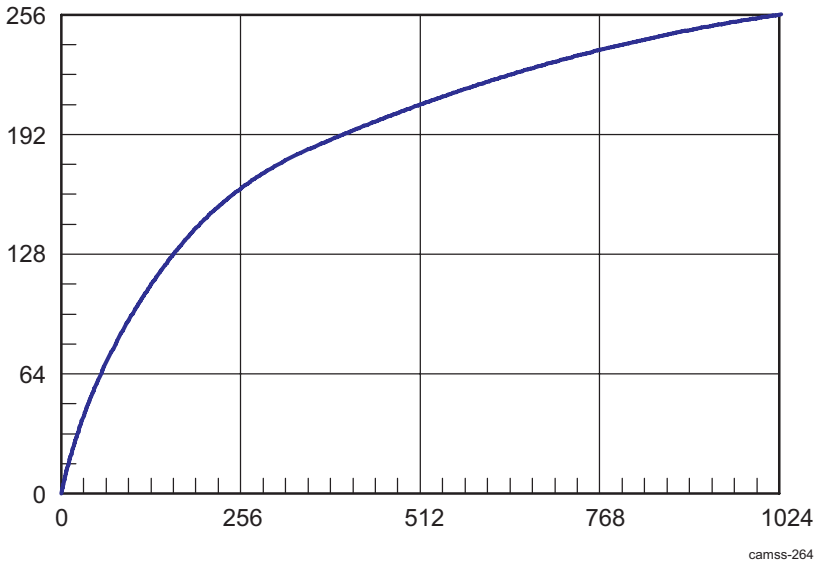


Figure 259. ISS ISP ISIF A-Law Table Values

Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law
0	0	64	64	128	112	192	140	256	161	320	176	384	189	448	200
1	1	65	65	129	113	193	141	257	161	321	176	385	189	449	200
2	2	66	66	130	113	194	141	258	161	322	177	386	189	450	200
3	3	67	67	131	114	195	142	259	161	323	177	387	189	451	200
4	4	68	68	132	114	196	142	260	162	324	177	388	190	452	200
5	5	69	69	133	115	197	142	261	162	325	177	389	190	453	200
6	6	70	70	134	115	198	143	262	162	326	177	390	190	454	201
7	7	71	71	135	116	199	143	263	162	327	178	391	190	455	201
8	8	72	72	136	116	200	143	264	163	328	178	392	190	456	201
9	9	73	73	137	117	201	144	265	163	329	178	393	190	457	201
10	10	74	74	138	117	202	144	266	163	330	178	394	191	458	201
11	11	75	75	139	118	203	144	267	163	331	178	395	191	459	201
12	12	76	76	140	118	204	145	268	164	332	179	396	191	460	201
13	13	77	77	141	119	205	145	269	164	333	179	397	191	461	202
14	14	78	78	142	119	206	145	270	164	334	179	398	191	462	202
15	15	79	78	143	120	207	146	271	164	335	179	399	191	463	202
16	16	80	79	144	120	208	146	272	165	336	179	400	192	464	202
17	17	81	80	145	121	209	146	273	165	337	180	401	192	465	202
18	18	82	81	146	121	210	147	274	165	338	180	402	192	466	202
19	19	83	82	147	122	211	147	275	166	339	180	403	192	467	202
20	20	84	83	148	122	212	147	276	166	340	180	404	192	468	203
21	21	85	84	149	123	213	148	277	166	341	181	405	193	469	203
22	22	86	84	150	123	214	148	278	166	342	181	406	193	470	203
23	23	87	85	151	124	215	148	279	167	343	181	407	193	471	203
24	24	88	86	152	124	216	149	280	167	344	181	408	193	472	203
25	25	89	87	153	125	217	149	281	167	345	181	409	193	473	203
26	26	90	88	154	125	218	149	282	167	346	182	410	193	474	204
27	27	91	88	155	125	219	150	283	168	347	182	411	194	475	204
28	28	92	89	156	126	220	150	284	168	348	182	412	194	476	204
29	29	93	90	157	126	221	150	285	168	349	182	413	194	477	204
30	30	94	91	158	127	222	151	286	168	350	182	414	194	478	204
31	31	95	91	159	127	223	151	287	168	351	183	415	194	479	204
32	32	96	92	160	128	224	151	288	169	352	183	416	194	480	204
33	33	97	93	161	128	225	152	289	169	353	183	417	195	481	205
34	34	98	93	162	129	226	152	290	169	354	183	418	195	482	205
35	35	99	94	163	129	227	152	291	169	355	183	419	195	483	205
36	36	100	95	164	129	228	152	292	170	356	184	420	195	484	205
37	37	101	96	165	130	229	153	293	170	357	184	421	195	485	205
38	38	102	96	166	130	230	153	294	170	358	184	422	195	486	205
39	39	103	97	167	131	231	153	295	170	359	184	423	196	487	205
40	40	104	98	168	131	232	154	296	171	360	184	424	196	488	206
41	41	105	98	169	132	233	154	297	171	361	185	425	196	489	206
42	42	106	99	170	132	234	154	298	171	362	185	426	196	490	206
43	43	107	100	171	132	235	155	299	171	363	185	427	196	491	206
44	44	108	100	172	133	236	155	300	172	364	185	428	196	492	206
45	45	109	101	173	133	237	155	301	172	365	185	429	197	493	206
46	46	110	102	174	134	238	155	302	172	366	185	430	197	494	206
47	47	111	102	175	134	239	156	303	172	367	186	431	197	495	207
48	48	112	103	176	134	240	156	304	173	368	186	432	197	496	207
49	49	113	103	177	135	241	156	305	173	369	186	433	197	497	207
50	50	114	104	178	135	242	157	306	173	370	186	434	197	498	207
51	51	115	105	179	136	243	157	307	173	371	186	435	198	499	207
52	52	116	105	180	136	244	157	308	173	372	187	436	198	500	207
53	53	117	106	181	136	245	157	309	174	373	187	437	198	501	207
54	54	118	106	182	137	246	158	310	174	374	187	438	198	502	208
55	55	119	107	183	137	247	158	311	174	375	187	439	198	503	208
56	56	120	108	184	137	248	158	312	174	376	187	440	198	504	208
57	57	121	108	185	138	249	159	313	175	377	188	441	198	505	208
58	58	122	109	186	138	250	159	314	175	378	188	442	199	506	208
59	59	123	109	187	139	251	159	315	175	379	188	443	199	507	208
60	60	124	110	188	139	252	159	316	175	380	188	444	199	508	208
61	61	125	110	189	139	253	160	317	175	381	188	445	199	509	208
62	62	126	111	190	140	254	160	318	176	382	188	446	199	510	209
63	63	127	112	191	140	255	160	319	176	383	189	447	199	511	209

camss-262

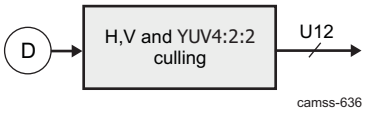
Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law
512	209	576	217	640	224	704	231	768	237	832	243	896	248	960	253
513	209	577	217	641	225	705	231	769	237	833	243	897	248	961	253
514	209	578	217	642	225	706	231	770	237	834	243	898	248	962	253
515	209	579	217	643	225	707	231	771	237	835	243	899	248	963	253
516	209	580	218	644	225	708	232	772	238	836	243	900	248	964	253
517	210	581	218	645	225	709	232	773	238	837	243	901	248	965	253
518	210	582	218	646	225	710	232	774	238	838	243	902	248	966	253
519	210	583	218	647	225	711	232	775	238	839	243	903	249	967	253
520	210	584	218	648	225	712	232	776	238	840	243	904	249	968	253
521	210	585	218	649	225	713	232	777	238	841	244	905	249	969	253
522	210	586	218	650	226	714	232	778	238	842	244	906	249	970	254
523	210	587	218	651	226	715	232	779	238	843	244	907	249	971	254
524	211	588	219	652	226	716	232	780	238	844	244	908	249	972	254
525	211	589	219	653	226	717	232	781	238	845	244	909	249	973	254
526	211	590	219	654	226	718	233	782	238	846	244	910	249	974	254
527	211	591	219	655	226	719	233	783	239	847	244	911	249	975	254
528	211	592	219	656	226	720	233	784	239	848	244	912	249	976	254
529	211	593	219	657	226	721	233	785	239	849	244	913	249	977	254
530	211	594	219	658	226	722	233	786	239	850	244	914	249	978	254
531	211	595	219	659	227	723	233	787	239	851	244	915	249	979	254
532	212	596	220	660	227	724	233	788	239	852	244	916	250	980	254
533	212	597	220	661	227	725	233	789	239	853	245	917	250	981	254
534	212	598	220	662	227	726	233	790	239	854	245	918	250	982	254
535	212	599	220	663	227	727	233	791	239	855	245	919	250	983	254
536	212	600	220	664	227	728	233	792	239	856	245	920	250	984	255
537	212	601	220	665	227	729	234	793	239	857	245	921	250	985	255
538	212	602	220	666	227	730	234	794	240	858	245	922	250	986	255
539	212	603	220	667	227	731	234	795	240	859	245	923	250	987	255
540	213	604	220	668	227	732	234	796	240	860	245	924	250	988	255
541	213	605	221	669	228	733	234	797	240	861	245	925	250	989	255
542	213	606	221	670	228	734	234	798	240	862	245	926	250	990	255
543	213	607	221	671	228	735	234	799	240	863	245	927	250	991	255
544	213	608	221	672	228	736	234	800	240	864	245	928	250	992	255
545	213	609	221	673	228	737	234	801	240	865	246	929	250	993	255
546	213	610	221	674	228	738	234	802	240	866	246	930	251	994	255
547	214	611	221	675	228	739	235	803	240	867	246	931	251	995	255
548	214	612	221	676	228	740	235	804	240	868	246	932	251	996	255
549	214	613	221	677	228	741	235	805	240	869	246	933	251	997	255
550	214	614	222	678	229	742	235	806	241	870	246	934	251	998	255
551	214	615	222	679	229	743	235	807	241	871	246	935	251	999	255
552	214	616	222	680	229	744	235	808	241	872	246	936	251	1000	255
553	214	617	222	681	229	745	235	809	241	873	246	937	251	1001	255
554	214	618	222	682	229	746	235	810	241	874	246	938	251	1002	255
555	215	619	222	683	229	747	235	811	241	875	246	939	251	1003	255
556	215	620	222	684	229	748	235	812	241	876	246	940	251	1004	255
557	215	621	222	685	229	749	235	813	241	877	246	941	251	1005	255
558	215	622	222	686	229	750	236	814	241	878	247	942	251	1006	255
559	215	623	223	687	229	751	236	815	241	879	247	943	252	1007	255
560	215	624	223	688	230	752	236	816	241	880	247	944	252	1008	255
561	215	625	223	689	230	753	236	817	242	881	247	945	252	1009	255
562	215	626	223	690	230	754	236	818	242	882	247	946	252	1010	255
563	216	627	223	691	230	755	236	819	242	883	247	947	252	1011	255
564	216	628	223	692	230	756	236	820	242	884	247	948	252	1012	255
565	216	629	223	693	230	757	236	821	242	885	247	949	252	1013	255
566	216	630	223	694	230	758	236	822	242	886	247	950	252	1014	255
567	216	631	223	695	230	759	236	823	242	887	247	951	252	1015	255
568	216	632	224	696	230	760	236	824	242	888	247	952	252	1016	255
569	216	633	224	697	230	761	237	825	242	889	247	953	252	1017	255
570	216	634	224	698	231	762	237	826	242	890	247	954	252	1018	255
571	217	635	224	699	231	763	237	827	242	891	248	955	252	1019	255
572	217	636	224	700	231	764	237	828	242	892	248	956	252	1020	255
573	217	637	224	701	231	765	237	829	243	893	248	957	253	1021	255
574	217	638	224	702	231	766	237	830	243	894	248	958	253	1022	255
575	217	639	224	703	231	767	237	831	243	895	248	959	253	1023	255

camss-263

3.3.6.14 ISS ISP ISIF Culling

Figure 260 shows the culling block diagram.

Figure 260. ISS ISP ISIF Culling Block Diagram



The culling block performs a programmable decimation function for horizontal, vertical, and YUV4:2:2 data directions. The horizontal and vertical decimation of image data can be controlled by two registers.

The horizontal culling operation allows selected pixel data to be culled (deleted) from a line. The ISIF_CULH register specifies the horizontal culling pattern for even and odd lines:

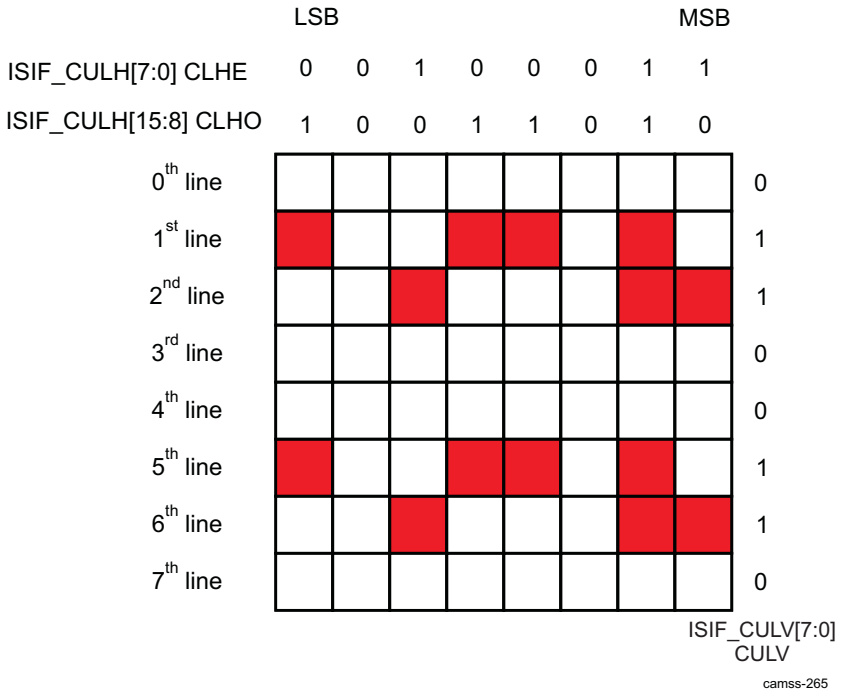
- Even lines: ISIF_CULH[15:8] CLHE
- Odd lines: ISIF_CULH[16:0] CLHO

The vertical culling operation allows selected lines to be culled from a frame. The ISIF_CULV register specifies the pattern for the vertical direction. The LSBs of CULV represent the top line of the CCD; the MSB is the seventh line.

Figure 261 is an example of how register values apply the decimation pattern to the data. The red pixels are saved to memory and the white pixels are discarded. In this example, CULH = 0x59C4 and CULV = 0x0066.

NOTE: Culling can be used with YUV data, but care must be taken to preserve the YUV4:2:2 output format.

Figure 261. ISS ISP ISIF Example for Decimation Pattern



3.3.6.15 ISS ISP ISIF 12-to-8 bit DCPM Compression Block

In ISIF, there is a DPCM compression block, which is between the culling module and the storage formatter module. This block can compress 12-bit image data to 8 bits for bandwidth reduction in transmission between the ISIF and SDRAM. An 8- to-12-bit DPCM decoder at the IPIPEIF decompresses data for IPIPE processing.

Two different predictors are used for the compression system. The first predictor is simple (simple predictor), and the second predictor is slightly more complex (advanced predictor). Because the advanced predictor gives a slightly better prediction for the pixel value, the image quality can be improved using it. Because the simple predictor is very simple, the processing power and memory requirements are reduced using it, when the image quality is already sufficiently high.

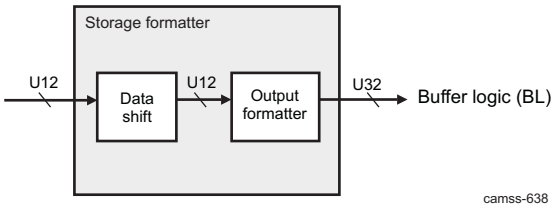
- Advanced predictor: This predictor uses only the previous same color component value as a prediction value. Therefore, only 2-pixel memory is required.
- This predictor uses four previous pixel values when the prediction value is evaluated. This means that the other color component values are also used when the prediction value has been defined.

The function is controlled from the ISIF_MISC[12] DPCMEN and ISIF_MISC[13] DPCMPRE bits.

3.3.6.16 ISP ISIF Storage Formatter

Figure 262 shows the storage formatter block diagram.

Figure 262. ISS ISP ISIF Storage Formatter Block Diagram



Data are stored to the lower bits of a 16-bit SDRAM word, or can be 8- or 12-bit packed. The ISIF_HSIZE[11:0] HSIZE bit field can specify the memory address offsets between lines of memory (offset in 32-byte units). If set, the ISIF_HSIZE[12] ADCR bit can decrement the memory address line and the line can be horizontally flipped in memory.

In case of RAW data, a data shift module is used: data to be stored can be right-shifted according to the value set at the ISIF_MODESET[10:8] CCDW bit field, as described in Table 204.

Table 204. ISS ISP ISIF RAW Data Shifting

ISIF_MODESET[10:8] CCDW	Output Format	
	MSB	LSB
000	0000 U12 data[11:0]	
001	00000 U12 data[11:1]	
010	000000 U12 data[11:2]	
011	0000000 U12 data[11:3]	
100	00000000 U12 data[11:4]	

Table 205 shows the format where data are stored to the lower bits of a 16-bit word and the format where data are packed to 8 bits. The unused bits are filled with zeros.

Table 205. ISS ISP ISIF SDRAM Data Format

	Upper Word		Lower Word	
	MSB(31)	LSB(16)	MSB(15)	LSB(0)
12 bit	0	Pixel 1	0	Pixel 0
11 bit	0	Pixel 1	0	Pixel 0
10 bit	0	Pixel 1	0	Pixel 0
9 bit	0	Pixel 1	0	Pixel 0
8 bit	0	Pixel 1	0	Pixel 0
8-bit packed	Pixel 3	Pixel 2	Pixel 1	Pixel 0

Table 206 shows the format where data are packed to 12 bits.

Table 206. ISS ISP ISIF SDRAM Data Format for 12-bit Packed

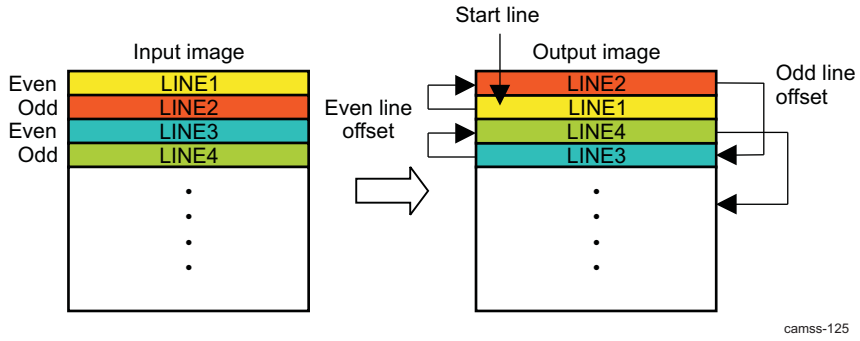
	Upper Word		Lower Word	
	MSB(31)	LSB(16)	MSB(15)	LSB(0)
12 bit	Pixel2[7:0]	Pixel1	Pixel 0	
	Pixel5[3:0]	Pixel4	Pixel3	Pixel2[11:8]
	Pixel7	Pixel6	Pixel5[11:4]	

In case of YUV, YUV data is stored in memory in packed YUV4:2:2 mode, using 2 pixels per 32 bits, as shown in Table 207.

The output formatter can configure to any image format by using the SDRAM line offset register and offset control registers. Figure 263 shows how to construct a frame format in SDRAM. The ISIF_CADU[10:0] CADU bit field specifies the memory destination (upper 11 bits) to SDRAM (the address is the value of the set bit multiplied by 32 bytes). On the other hand, the ISIF_CADL[15:0] CADL bit field sets the memory destination to SDRAM (lower 16 bits) (the address is the value of the set bit multiplied by 32 bytes).

- ISIF_SPH
- ISIF_LNH
- ISIF_SLV0
- ISIF_SLV1
- ISIF_LNV

Figure 263. ISS ISP ISIF Frame Image Format Conversion



3.3.6.17 ISS ISP ISIF YCbCr signal Processing

The ISIF accepts 4:2:2 sampled YCbCr input data . The luminance and color difference signals are 8 bits each, scaled 0 to 255. The color difference signals are multiplexed into one 8-bit bus beginning with a Cb sample. The Y and CbCr buses can be input parallel (16-bit mode) or be time-multiplexed and input as a single bus (8-bit mode).

The 16- or 8-bit YCbCr data is stored in SDRAM as 4:2:2 format. Table 207 lists the data format in SDRAM. Y data typically has a range of 16 to 235; however, it is possible to subtract a DC value from the Y signal.

Table 207. ISS ISP ISIF Memory Output Format for YUV Data

Memory Address	Upper Word		Lower Word	
	MSB(31)	LSB(16)	MSB(15)	LSB(0)
N	Y1	Cr0	Y0	Cb0
N + 1	Y3	Cr1	Y2	Cb1
N + 2	Y5	Cr2	Y4	Cb2

3.3.6.18 ISS ISP ISIF Expected Bandwidth on BL Ports

The ISIF module has a write port and a read port connected to the BL. This section summarizes the expected bandwidth on these ports.

3.3.6.18.1 ISS ISP ISIF Write Port

The write port is used to write pixels to memory after the data have passed through the storage formatter. Data storage to SDRAM is controlled by the ISIF_SYNCEN[1] DWEN bit. The module allows writing the data as 16, 12, and 8 bits per pixel. The bit width is controlled by the ISIF_CCDCFG[1:0] SDRPACK bit field.

The write port generates a burst of 32 bytes on the MTC interface. The delay between consecutive bursts is proportional to the input pixel clock. Hence, the write port does not request peak bandwidth traffic.

Table 208 lists the estimated delay between 32-byte MTC requests for different pixel clock frequencies and assumes the L3 clock is 200 MHz.

Table 208. ISP ISIF ISIF Module: Write Port Bandwidth

Pixel Clock	Maximum Bandwidth 2 bytes/s pixel MB/s	Expected Delay Between MTC Requests
200	400	16 cycles = 80 ns
100	200	32 cycles = 160 ns
10	20	320 cycles = 1600 ns

3.3.6.18.2 ISS ISP ISIF Read Port

The read port is used to read gain and offset data from SDRAM required for the LSC computation. When LSC is enabled, 8-bit gain values are read and 8-bit offset values can optionally also be read. The LSC gain computation can be enabled or disabled by setting the ISIF_2DLSCCFG[0] ENABLE bit. The LSC offset computation can be enabled by setting the ISIF_2DLSCOFST[0] OFSTEN bit.

The LSC submodule fetches four 8-bit gain values per paxel and optionally four 8-bit offset values per paxel. This is a maximum 8 bytes per paxel.

The bandwidth that is generated by the LSC module is also proportional to the paxel size. The paxel size is set up by the ISIF_2DLSCCFG[14:12] GAIN_MODE_M and ISIF_2DLSCCFG[10:8] GAIN_MODE_N bit fields. The possible values are 8, 16, 32, 64 and 128. Smaller values lead to higher memory bandwidth requirements. Hence, the worst case is achieved by setting an 8 × 8 paxel size.

When the LSC submodule is enabled it automatically prefetches two lines of gain values and two lines of offset values (if this is enabled). When the first VD comes, it again requests one line of gain values and one line of offset values (if this is enabled). Then, it again fetches one line of gain values and one line of offset values (if this is enabled) after ISIF_2DLSCCFG[10:8] GAIN_MODE_N lines. It continues to do so until the last row of paxels. For the last row of paxels, it fetch two lines of gain values and two lines of offset values (if this is enabled), which are used for the following frame.

By default, the LSC submodule creates peak bandwidth requirements. To avoid this, the MTC bandwidth limiter must be used to space the request over time.

The MTC bandwidth limiter must be used to smooth the bandwidth requirements of the LSC module. The MTC bandwidth limiter can be set with the ISP5_BL_MTC_1.ISIF_R register.

The principle is that instead of reading the gain and offset data as fast as possible, use the time that it takes for ISIF_2DLSCCFG[10:8] GAIN_MODE_N lines to pass through the ISP to read the data.

Table 209 gives the estimated delay between 32-byte MTC requests for different pixel clock frequencies and assumes the L3 clock at 200 MHz.

Table 209. ISS ISP ISIF Read Port Bandwidth

Pixel Clock	Max Bandwidth MB/s	Expected Delay Between MTC Requests
200	25.07	255 cycles = 1275 ns
100	12.5	510 cycles = 2550 ns
10	1.25	5103 cycles = 2515 ns

When the bandwidth limiter is used, ensure that there is enough time for the data prefetching.

- The LSC submodule must be enabled at least 2 × ISIF_2DLSCCFG[10:8] GAIN_MODE_N lines before the first VD.
- There must be at least ISIF_2DLSCCFG[10:8] GAIN_MODE_N lines of blanking. If there is not enough blanking, multiply the bandwidth requirement by 2 (that is, ensure two lines of gain and offset data can be fetched within the time of ISIF_2DLSCCFG[10:8] GAIN_MODE_N lines.

3.3.6.19 ISS ISP ISIF Events and Status Checking

The ISIF module can generate four different interrupts: VDINT0, VDINT1, VDINT2, and 2DLSCINT. The ISIF_SYNCEN[0] SYEN bit must be enabled to receive any of the ISIF interrupts.

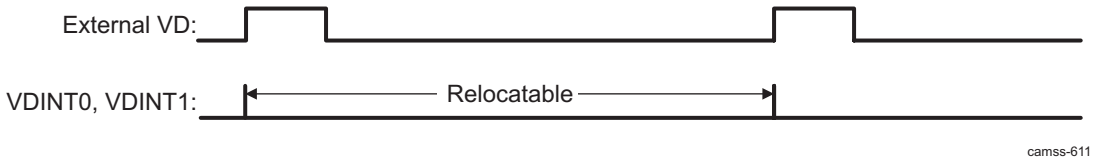
3.3.6.19.1 ISS ISP ISIF VDINT0, VDINT1, and VDINT2 Interrupts

As shown in Figure 264, the VDINT0, VDINT1, and VDINT2 interrupts occur relative to the VD pulse. The trigger timing is selected by using the ISIF_MODESET[2] VDPOL bit setting. VDINT0, VDINT1, and VDINT2 occur after receiving the number of horizontal lines (HD pulse signals) set in the ISIF_VDINT0[14:0] CDV0, ISIF_VDINT1[14:0] CDV1, and ISIF_VDINT2[14:0] CDV2 register fields, respectively.

NOTE: In the case of BT.656 input mode, there is a VD at the beginning of each field. Therefore, there are two interrupts for each frame (that is, one for each field).

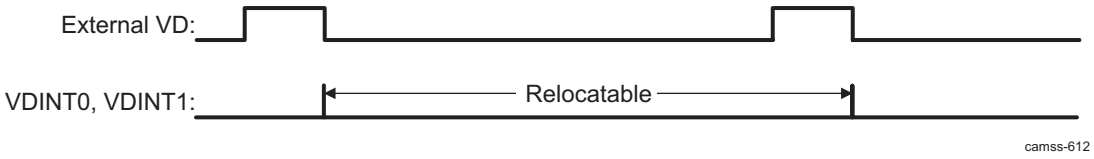
If the ISIF_MODESET[2] VDPOL bit is set to 0, the VDINT0, VDINT1, and VDINT2 HD counters begin counting HD pulses from the rising edge of the external VD.

Figure 264. ISS ISP ISIF VDINT0/VDINT1/VDINT2 Interrupt Behavior When VDPOL = 0



If the ISIF_MODESET[2] VDPOL bit is set to 1, the VDINT0, VDINT1, and VDINT2 HD counters begin counting HD pulses from the falling edge of the external VD.

Figure 265. ISS ISP ISIF VDINT0/VDINT1/VDINT2 Interrupt Behavior When VDPOL = 1



3.3.6.19.2 ISS ISP ISIF 2DLSCINT Interrupt

See Section 3.3.6.10.1.5 for more information.

3.3.6.19.3 ISS ISP ISIF Status Checking

The ISIF_MODESET[15] MDFS bit is set when the field status is on an even field, and it is cleared when the field status is on an odd field.

The 2D-LSC has a register that monitors the status of the LSC. See Section 3.3.6.10.1.5 for more information.

3.3.7 ISS ISP BL Functional Description

3.3.7.1 ISS ISP BL Overview

The BL module arbitrates and merges the memory requests of the ISP master module. The BL module also generates interrupts upon frame completion for the following modules. The interrupt generation is delayed until the transfer completes (ack is returned). The following interrupts are delayed by the BL module:

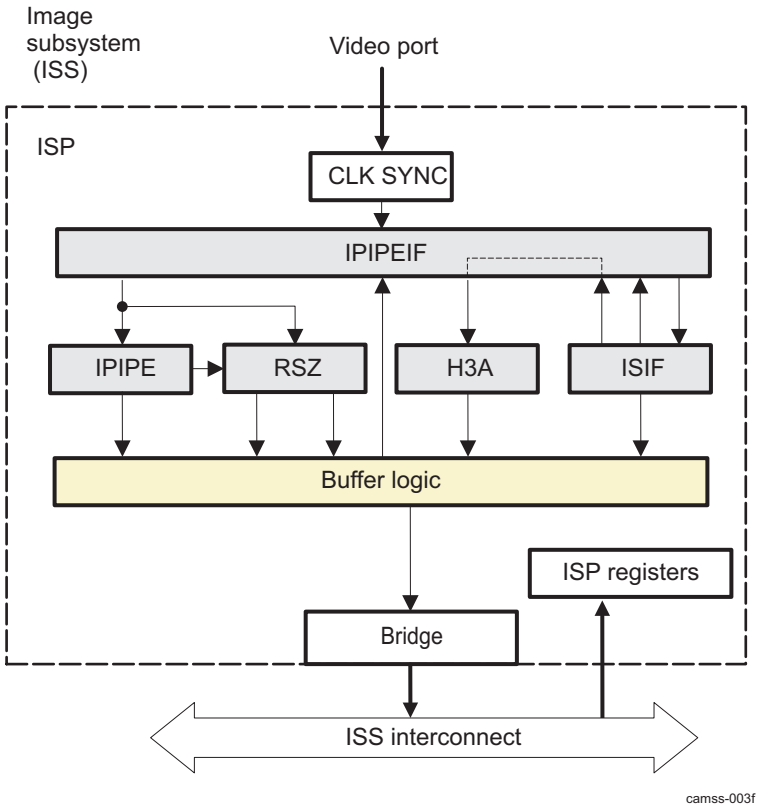
- RSZ module RSZ-A EOF
- RSZ module resizer RSZ-B EOF
- H3A module EOF
- IPIPE module EOF

The BL uses two different types of interfaces:

- MTC protocol is used between the ISP modules and the BL.
- VBUSM is use on the BL master port interface.

Figure 266 show the BL module connections to other submodules of the ISP.

Figure 266. ISS ISP BL High-Level Diagram



3.3.7.2 ISS ISP BL Functional Description

The BL merges the memory requests of the ISP master module to memory (read/write). The BL interfaces with all the ISP modules through a 32-bit-wide bus following.

The ISP modules make memory requests of 32 bytes. Additional signals, SOF for read and EOF for write, are included to deal with boundary conditions in frame transitions.

The BL arbitration is divided into two parts: a bus hog and a fixed priority arbitration. Bus hog refers to the property of the buffer logic that gives higher priority to the module that last sent or received data. RESIZER module MTC write port 0 and RESIZER module MTC write port 1 are excluded from the bus hog.

The buffer logic is to be programmed to maximize the memory bandwidth: it makes maximum burst requests of 128 bytes (8 × 128 bits) for reads and writes. The BL can generate burst sizes of 2 × 128, 4 × 128, 6 × 128, and 8 × 128 bits.

NOTE: The ISP interface supports burst sizes of only 1, 2, 4, and 8 × 128 bits. If the BL generates a 6 × 128-bit request, it is broken into a 4 × 128-bit request, followed by another 2 × 128-bit request.

To use the memory bandwidth efficiently, the BL interfaces with the memory through a high-bandwidth bus (128 bits wide).

The BL handles memory requests for the following modules:

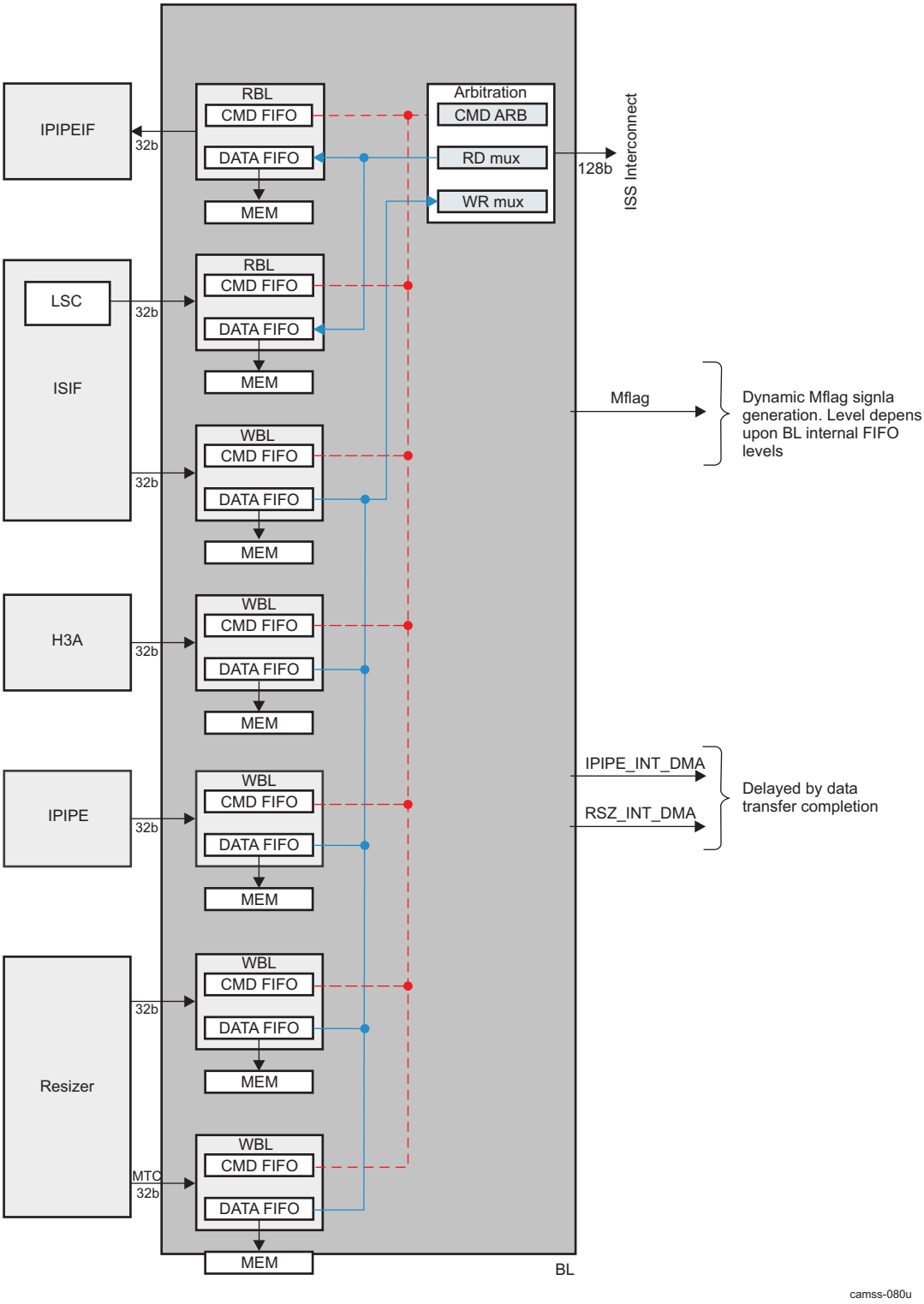
- IPIPEIF module read port
- ISIF-LSC module read port
- ISIF module write port
- IPIPE-BOXCAR module write port
- H3A module write port
- RESIZER module write port 0
- RESIZER module write port 1

From a use case point of view, the following sharing and priority arrangement is used. All reads have higher priority than writes; for reads: IPIPEIF ISIF-LSC, and for writes: ISIF IPIPE-BOXCAR RESIZER 0 RESIZER 1 H3A.

NOTE: BL can generate a static or a dynamic MFlag signal. The MFlag signal is used by the ISS arbitration to consider the urgency of the requests coming from the ISP. The dynamic MFlag feature is enabled from the ISP5_CTRL[21] MFLAG bit.

Figure 267 shows the BL top-level block diagram. The figure highlights the two clock domains that are used.

Figure 267. ISS ISP BL Block Diagram



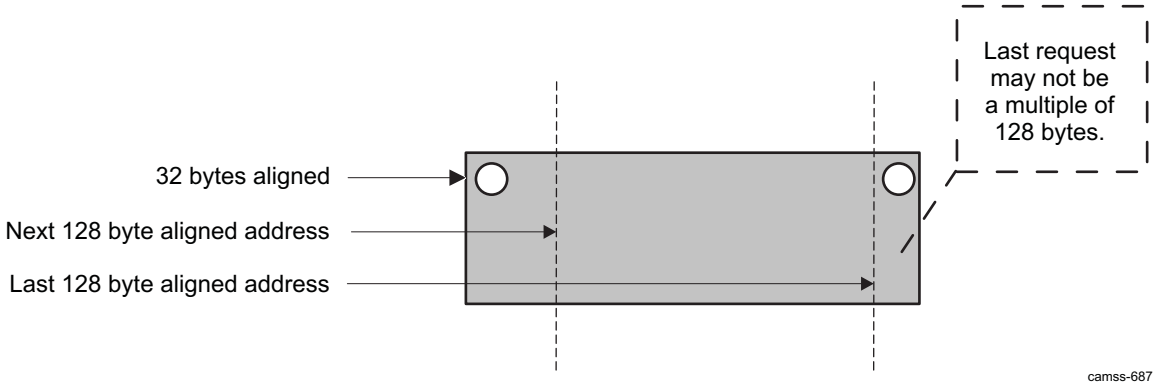
NOTE: The BL module has no registers. The configurations come from the top level of the ISP. See Table 260 for register details.

3.3.7.3 ISS ISP BL Address Alignment

The BL module ensures maximum memory efficiency by realigning data to a 128-byte address boundary. In all cases, the BL accesses are 32-byte-aligned: address [4:0] is always 0.

This is required when the input address is a multiple of 32 bytes, not 128 bytes. The BL issues a non-aligned burst until it reaches a 128-byte boundary, and then keeps making a 128-byte request until the end of the line. Eventually, although the last burst in a line may not be a multiple of 128 bytes, it will always be a multiple of 32 bytes, as shown in Figure 268.

Figure 268. ISS ISP BL Address Alignment



3.3.7.4 ISS ISP BL Out-of-Order Responses

BL supports out-of-order responses. The out-of-order response is handled by having up to 16 outstanding CIDs (reads + writes). The maximum number of outstanding CIDs is set up by the ISP5_CTRL[7:4] VBUSM_CIDS bit field. There is one outstanding request per CID.

If any of the CIDs are not outstanding, a command is accepted from the highest prioritized buffer with a request. The CID availability is cleared when status complete is received. The outstanding commands can be all reads or all writes or any combination. The CID that is allocated is the lowest number of the 16 that are available when the command is accepted. As soon as a CID is released it is used if a command is available.

The VBUSM2OCP module transforms the VBUMS CIDs into OCP MFlag signals. Note that there can be only one outstanding per tag. It is not authorized to make a new request on a tag which is already waiting for a response.

3.3.7.5 ISS ISP BL Stalling

The BL can stall the requests from the initiator modules for reads and writes

3.3.7.5.1 ISS ISP BL Stalling Write Requests

One reason for the BL to stall write requests is that there is usually back pressure on the system memory and requests cannot be issued as fast as required. In that case the BL must stall requests from the initiator modules. The BL module cannot know whether stalling the request will lead to an initiator overflow or not. The RESIZER module can support back pressure from the BL module when it is configured in bypass mode. The input buffer memory can be used to store data from the IPIPE and IPIPEIF modules . If the RESIZER input buffer overflows, an interrupt is issued.

3.3.7.5.2 ISS ISP BL Stalling Read Requests

The BL may also stall read requests from the ISIF-LSC and IPIPEIF modules. When data comes from an image sensor there is no way to stop the sensor. If the read request is stalled too long, the LSC module may eventually underflow. When the IPIPEIF module reads all its data from memory, the IPIPEIF stalls transfers to successive modules by masking the clock.

3.3.7.6 ISS ISP BL Dynamic and Static MFlag Generation

NOTE: The following applies when ISP5_CTRL[21] MFLAG = 1. The MFlag value is static.

The BL outputs cpriority[2:0] and cepriority[2:0] signals on its VBUS master port interface. On the VBUSM side, these signals are aligned with the request (creq). Both values are actually repeating register input signals. The lower value (0) corresponds to the higher priority.

The MFlag signal does not need to be aligned with the request. The signal can change value anytime.

In the ISP VBUSM2 interface bridge, the cepriority signal is ignored and only the cpriority signal is used. Because the priority value is set up by a register setting, the value is not going to be dynamically modified obviously. Table 210 shows how the cpriority values are mapped to the interface MFlag signal, which is present on the ISP master interface and set in the ISP5_CTRL[3:1] VBUSM_CPRIORITY bit field.

Table 210. ISS ISP BL cpriority to MFlag With ISP5_CTRL[21] MFLAG = 1

MFlag[1:0]	Description
00	Normal priority cpriority[2:0] = 4, 5, 6, 7
01	Medium priority cpriority[2:0] = 2, 3
10	Reserved
11	High priority cpriority[2:0] = 0, 1

NOTE: The following applies when the ISP5_CTRL[21] MFLAG = 0. The MFlag value is dynamic.

This feature is enabled at reset. The general idea for dynamic MFlag generation is to monitor the FIFO levels. Thresholds are used to increase or decrease the MFlag values. There are different implementations for read and write requestors.

An individual MFlag value is generated for each FIFO and then ORed altogether and exported at the BL boundary. The MFlag signal generation does not affect the BL arbitration scheme.

- Writes: To prevent overflows in the BL, dynamic MFlag signal generation gives higher priority to FIFOs that are almost full:
  - Low FIFO level = 50 percent
  - High FIFO level = 75 percent
  - FIFOs with more than 75 percent fill level have high priority: MFlag = 11
  - FIFOs between 50 and 75 percent fill level have medium priority: MFlag = 01
  - FIFOs below 50 percent fill level have normal priority: MFlag = 00

Table 211 gives the low- and high-level priority thresholds for write initiators.

Table 211. ISS ISP BL MFlag Write Low- and High-Level Priority Thresholds

	ISIF	H3A	IPIPE	RSZ
Access type	Write	Write	Write	Write
Buffer size	64 × 128	64 × 128	48 × 128	64 × 136
50% low level	32 × 128	32 × 128	24 × 128	32 × 136
75% high level	48 × 128	48 × 128	36 × 128	48 × 128

- Reads: The dynamic MFlag signal generation depends on the reserved data units in the initiator data FIFO. The reserved data units correspond to read commands waiting to be sent on the interface bus, plus the read commands that have been sent on the interface bus and for which data responses have not yet arrived.

- The total FIFO size in bytes can be expressed as reserved data units (bytes) + data bytes stored (bytes) + empty space (bytes). By definition, empty space is lower than a burst size (128 bytes).
  - Low FIFO level = 25 percent
  - High FIFO level = 50 percent
  - Data bytes stored + empty space (bytes) = 50 percent of FIFO size: MFlag = 00
  - Data bytes stored + empty space (bytes) 50 percent of FIFO size: MFlag = 01
  - Data bytes stored + empty space (bytes) 25 percent of FIFO size: MFlag = 11

Table 212 gives the low- and high-level priority thresholds for read initiators.

Table 212. ISS ISP BL MFlag Read Low- and High-Level Priority Thresholds

	IPIPEIF	ISIF-LSC
Access type	Read	Read
Buffer size	64 × 128	32 × 128
25% buffer size	16 × 128	8 × 128
50% buffer size	32 × 128	16 × 128

3.3.7.7 ISS ISP BL VBUSM2OCP Last Beat Command Delay

The VBUSM2OCP module bridge implements the following function to work around a limitation of the BL module, which does not send back-to-back requests to the ISS, thereby leading to possible situations where the ISP loses arbitration at the ISS level.

To fully benefit from dynamic MFlag generation (see Section 3.3.7.6), the following function is present in the VBUSM2OCP module bridge:

- The delay occurs only if BL MFflag = ISP5_BL_VBUSM[5] MFLAG_THRES.
- The MFlag value used is whatever is available when the last beat comes on the interface bus.
- The last beat of the interface request (read or write) is held until one cycle before a new command (read or write). This is achieved by masking the last beat of the interface command at the ISP interface.

  The last beat is unmasked on the first event of one cycle before a new interface command, or the delay counter that uses the ISP5_BL_VBUSM[4:0] LASTCMD_DLY value counter expires (has decremented to zero). The ISP5_BL_VBUSM[4:0] LASTCMD_DLY bit field must be set before the request on the BL starts. If the value of the ISP5_BL_VBUSM[4:0] LASTCMD_DLY bit field is changed during the pending requests, the delay counter is not updated.

3.3.7.8 ISS ISP BL Peak Memory Bandwidth Reduction

To limit the peak memory bandwidth generated by the IPIPEIF (read port), ISIF (read port), and H3A (write port) modules, a bandwidth limiter is placed between the modules and the BL. The resizer module has this function built in and therefore does not need a bandwidth limiter.

The bandwidth limiter enables control of the minimum interval between two consecutive memory requests.

This function is controlled by the ISP5_BL_MTC_1 and ISP5_BL_MTC_2 registers. When the registers are set to 0, the function is not modified (that is, the bandwidth limiter is disabled). For the resizer module, it is controlled by the following registers: RSZ_DMA_RZA and RSZ_DMA_RZB.

3.3.8 ISS ISP Memory Mapping

A total of 64KB is reserved for the ISP registers and memories. Table 213 and Table 214 describe the memory map.

Table 213. ISS ISP Memory Mapping for Cortex-M3 Private Access

Memory Mapping	Start	End	Size	Comments
ISS ISP5 SYS1	0x5505 0000	0x5505 009F	160	ISP5 configuration registers (set 1)
ISS ISP5 SYS2	0x5505 00A0	0x5505 03FF	864	ISP5 configuration registers (set 2)
ISS RESIZER registers	0x5505 0400	0x5505 07FF	1024	RESIZER configuration registers
ISS IPIPE registers	0x5505 0800	0x5505 0FFF	2048	IPIPE configuration registers
ISS ISIF registers	0x5505 1000	0x5505 11FF	512	ISIF configuration registers
ISS IPIPEIF registers	0x5505 1200	0x5505 13FF	512	IPIPEIF configuration registers
ISS H3A registers	0x5505 1400	0x5505 15FF	512	H3A configuration registers
Reserved	0x5505 1600	0x5505 17FF	512	Reserved
Reserved	0x5505 1800	0x5505 1BFF	1024	Reserved
Reserved	0x5505 1C00	0x5505 1DFF	512	Reserved
Reserved	0x5505 1E00	0x5505 1FFF	512	Reserved
HST memory 0	0x5505 2000	0x5505 27FF	2048	IPIPE histogram
HST memory 1	0x5505 2800	0x5505 2FFF	2048	IPIPE histogram
HST memory 2	0x5505 3000	0x5505 37FF	2048	IPIPE histogram
HST memory 3	0x5505 3800	0x5505 3FFF	2048	IPIPE histogram
BSC memory 1	0x5505 6000	0x5505 6EFF	8192	IPIPE Boundary Signal Calc column sum
DPC table 0	0x5505 8000	0x5505 81FF	1024	IPIPE Defect(Fault) Pixel Correction address table
DPC table 1	0x5505 8400	0x5505 85FF	1024	IPIPE Defect(Fault) Pixel Correction address table
YEE table	0x5505 8800	0x5505 8FFF	2048	IPIPE Y-data Edge Enhance table
GBC table	0x5505 9000	0x5505 97FF	2048	IPIPE GBCE LUT
3DLUT table0	0x5505 9800	0x5505 9AFF	1024	IPIPE 3D LUT
3DLUT table1	0x5505 9C00	0x5505 9EFF	1024	IPIPE 3D LUT
3DLUT table2	0x5505 A000	0x5505 A2FF	1024	IPIPE 3D LUT
3DLUT table3	0x5505 A400	0x5505 A6FF	1024	IPIPE 3D LUT
GAMR table	0x5505 A800	0x5505 AFFF	2048	IPIPE Gamma correction table (R)
GAMG table	0x5505 B000	0x5505 B7FF	2048	IPIPE Gamma correction table (G)
GAMB table	0x5505 B800	0x5505 BFFF	2048	IPIPE Gamma correction table (B)
LIN table0	0x5505 C000	0x5505 C17F	1024	ISIF Linearization table
LIN table1	0x5505 C400	0x5505 C57F	1024	ISIF Linearization table
DCCLAMP	0x5505 C800	0x5505 C9FF	2048	ISIF Digital Clamp
LSC table0	0x5505 D000	0x5505 E7FF	6144	ISIF Lens Shading gain table
LSC table1	0x5505 E800	0x5505 FFFF	6144	ISIF Lens Shading gain table

Table 214. ISS ISP Memory Mapping for L3 Interconnect Access

Memory Mapping	Start	End	Size	Comments
ISS ISP5 SYS1	0x5C01 0000	0x5C01 009F	160	ISP5 configuration registers (set 1)
ISS ISP5 SYS2	0x5C01 00A0	0x5C01 03FF	864	ISP5 configuration registers (set 2)
ISS RESIZER registers	0x5C01 0400	0x5C01 07FF	1024	RESIZER configuration registers
ISS IPIPE registers	0x5C01 0800	0x5C01 0FFF	2048	IPIPE configuration registers
ISS ISIF registers	0x5C01 1000	0x5C01 11FF	512	ISIF configuration registers
ISS IPIPEIF registers	0x5C01 1200	0x5C01 13FF	512	IPIPEIF configuration registers
ISS H3A registers	0x5C01 1400	0x5C01 15FF	512	H3A configuration registers
Reserved	0x5C01 1600	0x5C01 17FF	512	Reserved
Reserved	0x5C01 1800	0x5C01 1BFF	1024	Reserved
Reserved	0x5C01 1C00	0x5C01 1DFF	512	Reserved
Reserved	0x5C01 1E00	0x5C01 1FFF	512	Reserved
HST memory 0	0x5C01 2000	0x5C01 27FF	2048	IPIPE histogram
HST memory 1	0x5C01 2800	0x5C01 2FFF	2048	IPIPE histogram
HST memory 2	0x5C01 3000	0x5C01 37FF	2048	IPIPE histogram
HST memory 3	0x5C01 3800	0x5C01 3FFF	2048	IPIPE histogram
BSC memory 1	0x5C01 6000	0x5C01 6EFF	8192	IPIPE Boundary Signal Calc column sum
DPC table 0	0x5C01 8000	0x5C01 81FF	1024	IPIPE Defect(Fault) Pixel Correction address table
DPC table 1	0x5C01 8400	0x5C01 85FF	1024	IPIPE Defect(Fault) Pixel Correction address table
YEE table	0x5C01 8800	0x5C01 8FFF	2048	IPIPE Y-data Edge Enhance table
GBC table	0x5C01 9000	0x5C01 97FF	2048	IPIPE GBCE LUT
3DLUT table0	0x5C01 9800	0x5C01 9AFF	1024	IPIPE 3D LUT
3DLUT table1	0x5C01 9C00	0x5C01 9EFF	1024	IPIPE 3D LUT
3DLUT table2	0x5C01 A000	0x5C01 A2FF	1024	IPIPE 3D LUT
3DLUT table3	0x5C01 A400	0x5C01 A6FF	1024	IPIPE 3D LUT
GAMR table	0x5C01 A800	0x5C01 AFFF	2048	IPIPE Gamma correction table (R)
GAMG table	0x5C01 B000	0x5C01 B7FF	2048	IPIPE Gamma correction table (G)
GAMB table	0x5C01 B800	0x5C01 BFFF	2048	IPIPE Gamma correction table (B)
LIN table0	0x5C01 C000	0x5C01 C17F	1024	ISIF Linearization table
LIN table1	0x5C01 C400	0x5C01 C57F	1024	ISIF Linearization table
DCCLAMP	0x5C01 C800	0x5C01 C9FF	2048	ISIF Digital Clamp
LSC table0	0x5C01 D000	0x5C01 E7FF	6144	ISIF Lens Shading gain table
LSC table1	0x5C01 E800	0x5C01 FFFF	6144	ISIF Lens Shading gain table

3.4 ISS ISP Programming Model

NOTE: The preferred way to perform memory-to-memory processing with ISP is to use the Stall Controller (SC) module at the ISS level (see , *ISS Interfaces*.). It is possible to use the IPIPEIF read port for memory-to-memory processing, but it is not the preferred way because it does not provide enough granularity on the fractional clock divider for up to 20x digital zoom.

3.4.1 ISS ISP ISIF Programming Model

This section discusses issues related to the software control of the ISIF. It lists the registers that are required to be programmed in different modes, describes how to enable and disable the ISIF and how to check the status of the ISIF, discusses the different register access types, and enumerates several programming constraints.

3.4.1.1 ISS ISP ISIF Hardware Setup/Initialization

This section discusses the configuration of the ISIF required before image processing can begin.

3.4.1.1.1 ISS ISP ISIF Reset Behavior

Upon hardware reset of the ISP, all of the registers in the ISIF, except the defect table registers, are reset to their reset values. Because the defect table registers are stored in internal RAM, they do not have reset values. If the reset is a chip-level POR (reset after power is applied), the values of the defect table register are unknown. If the reset is an ISP module reset (when power remains active), the contents of this memory remain the same as before the reset.

3.4.1.1.2 ISS ISP ISIF Register Setup

Before enabling the ISIF, the hardware must be properly configured through register writes. Figure 269 and Figure 270 show the sequence to be used for RAW and YUV data before enabling the ISIF. The register settings for each process in the following sequence are described in Table 215.

Figure 269. ISS ISP ISIF Initialization Flow Chart – Part One

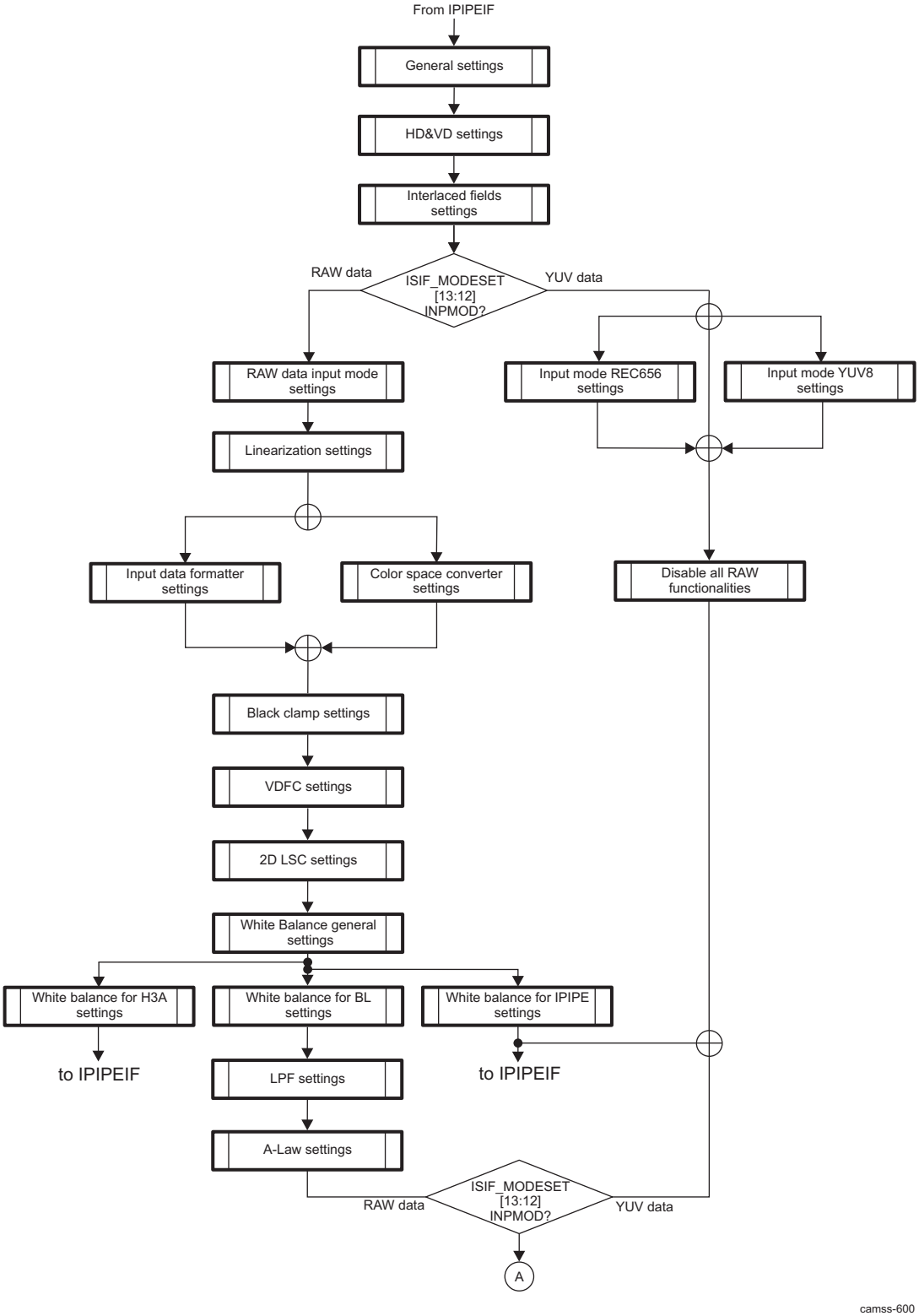


Figure 270. ISS ISP ISIF Initialization Flow Chart – Part Two

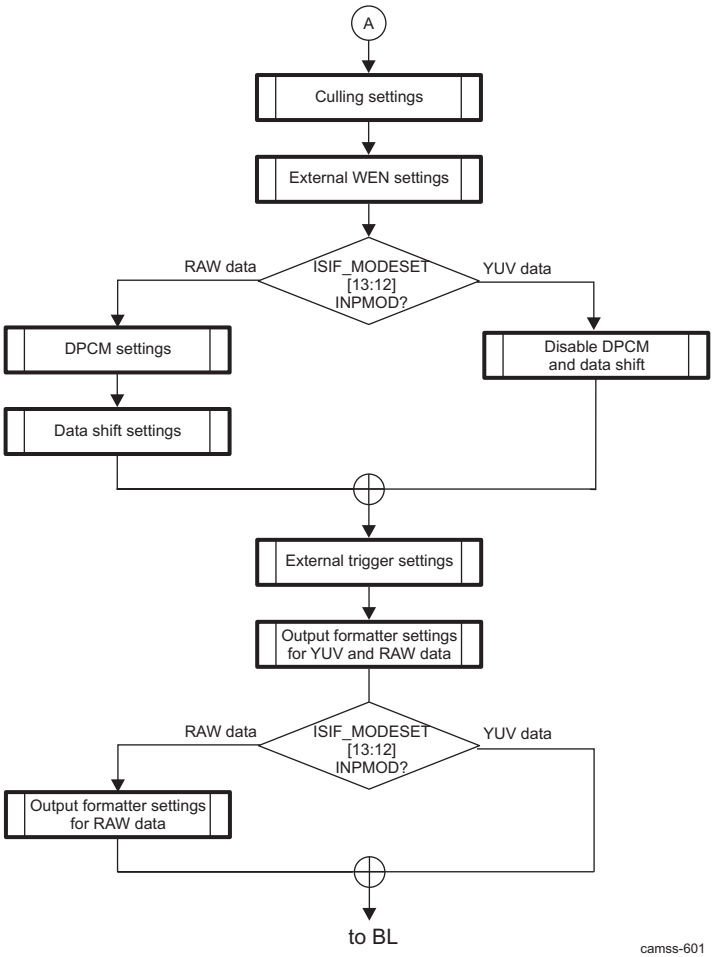


Table 215. ISS ISP ISIF Required Configuration Parameters

Step	Configuration Required	Value
General settings		
Set the Field Indicator signal direction.	ISIF_MODESET[1] FIDD	
Set the VD signal polarity.	ISIF_MODESET[2] VDPOL	
Set the HD signal polarity.	ISIF_MODESET[3] HDPOL	
Set the Field Indicator signal polarity.	ISIF_MODESET[4] FIPOL	
HD and VD settings		
Set the HD and VD signal directions.	ISIF_MODESET[0] HDVDD	
If: HD and VD are set as output (HDVDD = 0x1):		
Set the HD width.	ISIF_HDW[11:0] HDW	
Set the VD width.	ISIF_VDW[11:0] VDW	
Set the HD period.	ISIF_PPLN[15:0] PPLN	
Set the VD period.	ISIF_LPFR[15:0] LPFR	
End		
Interlaced fields settings		
Select the type of image sensor (progressive or interlaced)	ISIF_MODESET[7] CCDMD	
Input mode settings		
Set the data input mode.	ISIF_MODESET[13:12] INPMOD	

Table 215. ISS ISP ISIF Required Configuration Parameters (continued)

Step	Configuration Required	Value
Input mode settings for REC656 data		
Select CCIR Rec.656 interface or not.	ISIF_REC656IF[1] R656ON	
If: input is REC656 (R656ON = 0x1)		
Set error correction of FVH code.	ISIF_REC656IF[0] ECCFVH	
Selects bit width of CCIR656.	ISIF_CCDCFG[5] BT656	
End		
Input mode settings for YCC 8 bit data (if INPMOD = 0x2)		
Selects Y position signal	ISIF_CCDCFG[11] Y8POS	
RAW data processing: input settings		
Enable or disable MSB inverse of CIN port (YUV format).	ISIF_CCDCFG[13] MSBINVI	
Select Y and C swapping.	ISIF_CCDCFG[4] YCINSWP	
Selects MSB position of input data (16 bits-to-16 bits)	ISIF_CGAMMAWD[4:1] GWDI	
Set the image sensor data polarity.	ISIF_MODESET[6] DPOL	
Select the CFA pattern mode.	ISIF_CGAMMAWD[5] CFAP	
Specifies the color pattern	ISIF_CCOLP[7:6] CP0_F0	
	ISIF_CCOLP[5:4] CP1_F0	
	ISIF_CCOLP[3:2] CP2_F0	
	ISIF_CCOLP[1:0] CP3_F0	
Linearization settings		
Enable linearization or not.	ISIF_LINCFG0[0] LINEN	
If: Linearization enabled (LINEN = 0x1)		
Select linearization mode.	ISIF_LINCFG0[1] LINMD	
Select the shift value.	ISIF_LINCFG0[6:4] CORRSFT	
Set the scale factor for LUT.	ISIF_LINCFG1[10:0] LUTSCL	
Set up linearization LUT.		
End		
Input data formatter settings		
Enable data formatter or not.	ISIF_FMTCFG[0] FMTEN	
If: Data formatter is enabled (FMTEN = 0x1)		
Select the combine input lines.	ISIF_FMTCFG[1] FMTCBL	
Select the mode normal or alternative.	ISIF_FMTCFG[2] LNALT	
Select the split/combine number of lines.	ISIF_FMTCFG[5:4] LNUM	
Set the address increment.	ISIF_FMTCFG[11:8] FMTAINC	
Set the number of program entries per SET.	ISIF_FMTPLEN[3:0] FMTPLEN0	
	ISIF_FMTPLEN[7:4] FMTPLEN1	
	ISIF_FMTPLEN[10:8] FMTPLEN2	
	ISIF_FMTPLEN[14:12] FMTPLEN3	
Set the first pixel in a line.	ISIF_FMTSPH[12:0] FMTSPH	
Set the number of pixels in a line.	ISIF_FMTLNH[12:0] FMTLNH	
Set the start line vertical.	ISIF_FMTLSV[12:0] FMTSLV	
Set the number of lines in a vertical.	ISIF_FMTLNV[14:0] FMTLNV	
Set the number of pixels in an output line.	ISIF_FMTRLEN[12:0] FMTRLEN	
Set the HD interval for output lines.	ISIF_FMTHCNT[12:0] FMTHCNT	
Set up to 16 address pointers.	ISIF_FMTAPTRx[14:13] LINE (x = 0 to 15)	
	ISIF_FMTAPTRx[12:0] INIT (x = 0 to 15)	

Table 215. ISS ISP ISIF Required Configuration Parameters (continued)

Step	Configuration Required	Value
Set the 32 possible program entry valid flag.	ISIF_FMTPGMVF0	
	ISIF_FMTPGMVF1	
Set the 32 possible address pointers.	ISIF_FMTPGMAPS0	
	ISIF_FMTPGMAPS1	
	ISIF_FMTPGMAPS2	
	ISIF_FMTPGMAPS3	
	ISIF_FMTPGMAPS4	
	ISIF_FMTPGMAPS5	
	ISIF_FMTPGMAPS6	
	ISIF_FMTPGMAPS7	
Set the 32 possible address update (increment or decrement).	ISIF_FMTPGMAPU0	
	ISIF_FMTPGMAPU1	
End		
Color space converter settings		
Enable or disable color space converter.	ISIF_CSCCTL[0] CSCEN	
If: Color space converter is enabled (CSCEN = 0x1)		
Set the color space converter coefficients.	ISIF_CSCM0	
	ISIF_CSCM1	
	ISIF_CSCM2	
	ISIF_CSCM3	
	ISIF_CSCM4	
	ISIF_CSCM5	
	ISIF_CSCM6	
	ISIF_CSCM7	
End		
Black Clamp settings		
Enable or disable black clamp.	ISIF_CLAMPCFG[0] CLEN	
If: Black clamp is enabled (CLEN = 0x1)		
Set the DC offset for black clamp.	ISIF_CLDCOFST[12:0] CLDC	
[Horizontal Black Clamp]		
Set the horizontal clamp mode.	ISIF_CLAMPCFG[2:1] CLHMD	
Set the vertical dimension of a window.	ISIF_CLHWIN0[13:12] CLHWN	
Set the horizontal dimension of a window.	ISIF_CLHWIN0[9:8] CLHWM	
Enable or disable limitation for horizontal.	ISIF_CLHWIN0[6] CLHLMT	
Select base window.	ISIF_CLHWIN0[5] CLHWBS	
Set the window count per color.	ISIF_CLHWIN0[4:0] CLHWC	
Set window start position (H).	ISIF_CLHWIN1[12:0] CLHSH	
Set the window start position (V).	ISIF_CLHWIN2[12:0] CLHSV	
[Vertical Black Clamp]		
Set the black clamp start position.	ISIF_CLSV[12:0] CLSV	
Set the vertical black clamp reset value.	ISIF_CLVRV[11:0] CLVRV	
Set the line average coefficient.	ISIF_CLVWIN0[15:8] CLVCOEF	
Select the reset value for the clamp value of the previous line.	ISIF_CLVWIN0[5:4] CLVRVSL	
Select the optical black H valid.	ISIF_CLVWIN0[2:0] CLVOBH	
Set the window start position (H).	ISIF_CLVWIN1[12:0] CLVSH	
Set the window start position (V).	ISIF_CLVWIN2[12:0] CLVSV	
Select the optical black V valid.	ISIF_CLVWIN3[12:0] CLVOBV	

Table 215. ISS ISP ISIF Required Configuration Parameters (continued)

Step	Configuration Required	Value
End		
Set the DC offset for black clamp (RAW data only).	ISIF_CLDCOFST[12:0] CLDC	DC offset available for YUV
Vertical line defect correction (VDFC) settings		
Disable vertical line defect correction.	ISIF_DFCCTL[4] VDFCEN	0x0
If: Vertical line defect correction will be enabled (VDFCEN = 0x1)		
Select the mode.	ISIF_DFCCTL[6:5] VDFCSL	
Select upper pixels correction enable or disable.	ISIF_DFCCTL[7] VDFCUDA	
Set the shift value.	ISIF_DFCCTL[10:8] VDFLSFT	
Set the saturation level.	ISIF_VDFSATLV[11:0] VDFSLV	
Clear memories.	ISIF_DFCMEMCTL[4] DFCMCLR	0x1
Vertical line defect table update procedure.	See Section 3.3.6.9.1 for details.	
	Use the following registers:	
	ISIF_DFCMEMCTL[2] DFCMARST	
	ISIF_DFCMEMCTL[0] DFCMWR	
	ISIF_DFCMEM0	
	ISIF_DFCMEM1	
	ISIF_DFCMEM2	
	ISIF_DFCMEM3	
	ISIF_DFCMEM4	
End		
Enable vertical line defect correction.	ISIF_DFCCTL[4] VDFCEN	0x1
2D Lens Shading Compensation (LSC) settings		
Disable lens shading compensation.	ISIF_2DLSCCFG[0] ENABLE	0x0
If: 2D-LSC will be enabled (ENABLE = 0x1)		
Set the H direction data offset.	ISIF_LSCHOFST[13:0] HOFST	
Set the V direction data offset.	ISIF_LSCVOFST[13:0] VOFST	
Set the number of valid pixels in H direction.	ISIF_LSCHVAL[13:0] HVAL	
Set the number of valid lines in V direction.	ISIF_LSCVVAL[13:0] VVAL	
Define the horizontal dimension of a paxel.	ISIF_2DLSCCFG[14:12] GAIN_MODE_M	
Define the vertical dimension of a paxel.	ISIF_2DLSCCFG[10:8] GAIN_MODE_N	
Set gain format table.	ISIF_2DLSCCFG[3:1] GAIN_FORMAT	
Enable or disable offset control.	ISIF_2DLSCOFST[0] OFSTEN	
Select shift up value for offsets.	ISIF_2DLSCOFST[6:4] OFSTSFT	
Set scaling factor for offset.	ISIF_2DLSCOFST[15:8] OFSTSF	
Set the initial Y position.	ISIF_2DLSCINI[14:8] Y	
Set the initial X position.	ISIF_2DLSCINI[6:0] X	
Set the gain table base address.	ISIF_2DLSCGRBU[15:0] BASE31_16	
	ISIF_2DLSCGRBL[15:0] BASE15_0	
Set the gain table offset (length of one row).	ISIF_2DLSCGROF[15:0] OFFSET	
Set the offset table base address.	ISIF_2DLSCORBU[15:0] BASE	
	ISIF_2DLSCORBL[15:0] BASE	
Set the offset table offset (length of one row).	ISIF_2DLSCOROF	

Table 215. ISS ISP ISIF Required Configuration Parameters (continued)

Step	Configuration Required	Value
Enable useful interrupts	ISIF_2DLSCIRQEN[3] SOF	
	ISIF_2DLSCIRQEN[2] PREFETCH_COMPLETED	
	ISIF_2DLSCIRQEN[1] PREFETCH_ERROR	
	ISIF_2DLSCIRQEN[0] DONE	
Set up LSC gain table and offset table in SDRAM. See Figure 256 for details.		
Wait seven clock periods before enabling LSC.		
Enable lens shading compensation.	ISIF_2DLSCCFG[0] ENABLE	0x1
End		
White balance color settings		
Set R/Ye gain.	ISIF_CRGAIN[11:0] CGR	
Set Gr/Cy gain.	ISIF_CGRGAIN[11:0] CGGR	
Set Gr/Cy gain.	ISIF_CGBGAIN[11:0] CGGB	
Set B/Mg gain.	ISIF_CBGAIN[11:0] CGB	
Set offset.	ISIF_COFSTA[11:0] COFT	
[For BL output]		
Enable or disable white balance for BL path.	ISIF_CGAMMAWD[12] WBNEN0	
Enable or disable offset control for BL path.	ISIF_CGAMMAWD[8] OFSTEN0	
[For IPIPE (through IPIPEIF) output]		
Enable or disable white balance for IPIPE path.	ISIF_CGAMMAWD[13] WBEN1	
Enable or disable offset control for IPIPE path.	ISIF_CGAMMAWD[9] OFSTEN1	
[For H3A (through IPIPEIF) output]		
Enable or disable white balance for H3A path.	ISIF_CGAMMAWD[14] WBEN2	
Enable or disable offset control for H3A path.	ISIF_CGAMMAWD[10] OFSTEN2	
LPF settings (for BL output only) (RAW data only)		
Enable or disable low pass filter.	ISIF_MODESET[14] HLPF	
A-Law compression settings (for BL output only) (RAW data only)		
Enable or disable A-Law compression.	ISIF_CGAMMAWD[0] CCDTBL	
Culling settings (for BL output only) (RAW or YUV data)		
Set the culling pattern in odd lines.	ISIF_CULH[15:8] CLHO	
Set the culling pattern in even lines.	ISIF_CULH[7:0] CLHE	
Set the culling pattern in vertical lines.	ISIF_CULV[7:0] CULV	
External WEN settings (for BL output only)		
Select external WEN use or not.	ISIF_MODESET[5] SWEN	
If: External WEN is used (SWEN = 0x1):		
Specifies the CCD valid area	ISIF_CCDCFG[8] WENLOG	
End		
DPCM settings (for BL output only) RAW data only		
Select the predictor for DPCM encoder.	ISIF_MISC[13] DPCMPRE	
Enable or disable DPCM encoding.	ISIF_MISC[12] DPCMEN	
Data shift settings (for BL output only) (RAW data only)		
Select the data shift value when image is written to memory	ISIF_MODESET[10:8] CCDW	

Table 215. ISS ISP ISIF Required Configuration Parameters (continued)

Step	Configuration Required	Value
External trigger settings (for BL output only)		
If: External trigger is selected (EXTRG = 0x1):		
Select the trigger source signal.	ISIF_CCDCFG[9] TRGSEL	
End		
Output formatter (for BL output only) RAW and YUV		
Set the memory address decrement.	ISIF_HSIZE[12] ADCR	
Set the memory address offset between lines.	ISIF_HSIZE[11:0] HSIZE	
Set the first pixel in a line to be stored in memory.	ISIF_SPH[14:0] SPH	
Set the number of pixels in a line to be stored in memory.	ISIF_LNH[14:0] LNH	
Set the start line vertical for field 0.	ISIF_SLV0[14:0] SLV0	
Set the start line vertical for field 1.	ISIF_SLV1[14:0] SLV1	
Set the number of lines to be stored in memory.	ISIF_LNV[14:0] LNV	
Enable or disable the storage of image in memory.	ISIF_SYNCEN[1] DWEN	
Set the memory destination address.	ISIF_CADU[10:0] CADU	
	ISIF_CADL[10:0] CADL	
Enable or disable MSB inverse of COUT port.	ISIF_CCDCFG[14] MSBINVO	
Enable or disable byte swap when SDRAM capturing.	ISIF_CCDCFG[12] BSWD	
Select Y and C swapping.	ISIF_CCDCFG[2] YCOUNTSWP	
Select SDRAM pack mode.	ISIF_CCDCFG[1:0] SDRPACK	
Enable or disable VD/HD output.	ISIF_SYNCEN[0] SYEN	

3.4.1.2 ISS ISP ISIF Enable/Disable Hardware

The ISIF is enabled by setting the ISIF_SYNCEN[0] SYEN bit. This is done after all the required registers discussed in the previous section are programmed.

With respect to the write enable bit and output address, the following procedure must be followed:

1. Set the data output address (ISIF_CADU and ISIF_CADL).
2. Enable HD/VD and WEN at the same time (ISIF_MODESET[1] DWEN and ISIF_SYNCEN[0] SYEN).

If the ISIF_SYNCEN[0] SYEN bit is written before the output address and the SDRAM write enable bit (not recommended but may be required for a particular mode), data begins to be written to the old address value and not the one recently programmed. The desired response can be achieved if the following procedure is followed:

1. Enable HD/VD (ISIF_SYNCEN[0] SYEN).
2. Set the output address (ISIF_CADU and ISIF_CADL).
3. Wait for the next VD.
4. Enable WEN (ISIF_MODESET[1] DWEN).

The ISIF always operates in continuous mode. In other words, after enabling the ISIF, it continues to process sequential frames until the ISIF_SYNCEN[0] SYEN bit is cleared by software. When this happens, the frame being processed is disabled immediately and does not continue to process the current frame.

When the HD/VD signals are set to outputs (ISIF_MODESET[0] HDVDD = 0x1), fetching and processing of the frame begin immediately upon setting the ISIF_SYNCEN[0] SYEN bit.

When the HD/VD signals are set to inputs (ISIF_MODESET[0] HDVDD = 0x0), processing of the frame is dependent on the input timing of the external sensor/decoder. To ensure that data from the external device is not missed, the ISIF must be enabled before data transmission from the external device. In this way, the ISIF waits for data from the external device.

3.4.1.3 ISS ISP ISIF Register Accessibility During Frame Processing

There are two types of register access in the ISIF:

- Shadowed registers (event latched registers)

  Shadowed registers are those that can be read and written at any time, but the written values take effect (are latched) only at certain times based on some event. Reads still return the most recent write even though the settings are not used until the specific event occurs.
- Busy-writable registers

  These registers/fields can be read or written even if the module is busy. Changes to the underlying settings takes place instantly.

The following registers/fields can be set as shadow registers, or optionally set as busy-writable registers. When the ISIF_CCDCFG[15] VDLC bit is set to 0, these registers are shadowed; when the ISIF_CCDCFG[15] VDLC bit is set to 1, these registers are busy-writable. All other ISIF registers not included in this list are always busy-writable.

ISIF_SYNCEN[1] DWEN	ISIF_SLV1	ISIF_CGRGAIN	ISIF_FMTLSV
ISIF_MODESET[14] HLPF	ISIF_CULH	ISIF_CGBGAIN	ISIF_FMTLNV
ISIF_HDW	ISIF_CULV	ISIF_CBGAIN	ISIF_LSCHOFST
ISIF_VDW	ISIF_HSIZE	ISIF_COFSTA	ISIF_LSCVOFST
ISIF_PPLN	ISIF_SDOFST	ISIF_CLAMPCFG[0] CLEN	ISIF_DFCCTL
ISIF_LPFR	ISIF_CADU	ISIF_MISC	ISIF_VDFSATLV
ISIF_SPH	ISIF_CADL	ISIF_CGAMMAWD	
ISIF_LNH	ISIF_CCOLP	ISIF_FMTSPH	
ISIF_SLV0	ISIF_CRGAIN	ISIF_FMTLNH	

3.4.1.4 ISS ISP ISIF Interframe Operations

Between frames, it may be necessary to enable or disable functions or to modify the memory pointers. Because the ISIF_SYNCEN[1] DWEN bit and the memory pointer registers are shadowed, these modifications can occur any time before the end of the frame and the data will be latched in for the next frame. Likewise, the 2D-LSC registers can be changed after receiving the LSC SOF interrupt but before it starts to prefetch the gain values for the next frame (the end of the LSC active region is reached). The host controller can perform these changes upon receiving an interrupt.

3.4.1.5 ISS ISP ISIF Summary of Constraints

The following is a list of register configuration constraints to adhere to when programming the ISIF. It can be used as a quick checklist. More detailed register setting constraints can be found in the individual register descriptions.

- PCLK cannot be higher than 200 MHz.
- If SDRAM output port is enabled:
  - The memory output line offset and address must be on 32-byte boundaries.
  - ISIF_LNH[14:0] LNH –1 must be a multiple of 32 bytes.
  - ISIF_SPH, ISIF_LNH, ISIF_SLV0, ISIF_SLV1, and ISIF_LNV must be cleared to 0 within the same VD period that the ISIF_SYNCEN[1] DWEN bit is cleared to 0.

  - ISIF_SPH, ISIF_LNH, ISIF_SLV0, ISIF_SLV1, and ISIF_LNV must be set from 0 to the appropriate values within the same VD period that the ISIF_SYNCEN[1] DWEN bit is set to 1.
- In RAW input mode:
  - ISIF_CCDCFG[4] YCINSWP must be set to 0.
- If DPCM compression is enabled:
  - Horizontal culling must not be used. Use the input formatter instead.
- For 2D-LSC:
  - N=M (where M = horizontal downsampling factor, N = vertical downsampling factor)
  - The ISIF_2DLSCINI register values must be even numbers.
  - Maximum widths with respect to selected M value (see Table 216)

Table 216. ISS ISP ISIF Maximum Line Width Versus M Value

M	Maximum Line Width
8	2040
16	4080
32	8160
64	16,320
128	16,384

3.4.2 ISS ISP IPIPEIF Programming Model

This section discusses issues related to the software control of the IPIPEIF module. It lists the registers that are required to be programmed in different modes, describes how to enable and disable the IPIPEIF module and how to check the status of the IPIPEIF module, discusses the different register access types, and enumerates several programming constraints.

3.4.2.1 ISS ISP IPIPEIF Hardware Setup/Initialization

This section discusses the configuration of the IPIPEIF module required before image processing can begin.

3.4.2.1.1 ISS ISP IPIPEIF Reset Behavior

Upon hardware reset of the ISP (ISP5_SYSCONFIG[1] SOFTRESET = 0x1), all the registers in the IPIPEIF are reset to their reset values.

3.4.2.1.2 ISS ISP IPIPEIF Register Setup

Before enabling the IPIPEIF, the hardware must be properly configured through register writes. Table 217 identifies the register parameters that must be programmed before enabling the IPIPEIF module (depending on the functions needed).

Table 217. ISS ISP IPIPEIF Required Configuration Parameters

Step	Configuration Required	Value
General settings		
Select the input sources for IPIPEIF.	IPIPEIF_CFG1[15:14] INPSRC1	
	IPIPEIF_CFG1[3:2] INPSRC2	
Select VD sync polarity.	IPIPEIF_CFG2[2] VDPOL	
Select HD sync polarity.	IPIPEIF_CFG2[1] HDPOL	
Select the interrupt source.	IPIPEIF_CFG2[0] INTSW	
Select the input clock source.	IPIPEIF_CFG1[10] CLKSEL	
Set the clock divider value.	IPIPEIF_CLKDIV[15:0] CLKDIV	
Set the data type: YUV or RAW.	IPIPEIF_CFG2[3] YUV16	

Table 217. ISS ISP IPIPEIF Required Configuration Parameters (continued)

Step	Configuration Required	Value
Settings to perform if data is from BL		
Set HD setting.	IPIPEIF_PPLN[12:0] PPLN	
Set VD setting.	IPIPEIF_LPFR[12:0] LPFR	
Set the number of valid pixels in a line.	IPIPEIF_HNUM[12:0] HNUM	
Set the number of valid lines.	IPIPEIF_VNUM[12:0] VNUM	
Set memory address information.	IPIPEIF_ADDRU[10:0] ADDRU	
	IPIPEIF_ADDRL[15:0] ADDRL	
	IPIPEIF_ADOFS[11:0] ADOFS	
Set one-shot mode, if needed.	IPIPEIF_CFG1[0] ONESHOT	
Use SYNCOFF function only for double-buffering. See Section 3.3.2.3.2.1 for details.	IPIPEIF_ENABLE[1] SYNCOFF	
Settings to perform for RAW data from ISIF or BL		
Select the unpack function.	IPIPEIF_CFG1[9:8] UNPACK	
Settings to perform for YUV for data from ISIF		
Set the data type to YUV.	IPIPEIF_CFG2[3] YUV16	
If YUV16 = 0x1, enables or not the conversion from 8 bits to 16 bits	IPIPEIF_CFG2[6] YUV8	
If YUV16 = 0x1, set the way the data is unpacked.	IPIPEIF_CFG2[7] YUV8P	
DCPM function	RAW data from Buffer Logic (BL)	
Enable or disable DPCM decompression.	IPIPEIF_DPCM[0] ENA	
Select DPCM prediction mode.	IPIPEIF_DPCM[1] PRED	
Select DPCM bit mode.	IPIPEIF_DPCM[2] BITS	
Select inverse A-Law function.	IPIPEIF_CFG1[9:8] UNPACK	
Select SDRAM read data shift. - For RAW data - DPCM enabled and A- Law disabled - or DPCM disabled	IPIPEIF_CFG1[13:11] DATASFT	
Dark frame subtraction (DFS) function	RAW data from VP (video port), BL(buffer logic, SDRAM) or ISIF	
Set defect pixel correction (DPC1) for VP or ISIF inputs.	IPIPEIF_DPC1[12] ENA	
Set the associated threshold for DPC1.	IPIPEIF_DPC1[11:0] TH	
Set defect pixel correction (DPC2) for BL inputs.	IPIPEIF_DPC2[12] ENA	
Set the associated threshold for DPC1.	IPIPEIF_DPC2[11:0] TH	
Set the direction of subtraction.	IPIPEIF_CFG2[5] DFSDIR	
Set the averaging filter, horizontal pixel decimator, and gain function for IPIPE data path.		
Enable averaging filter function.	IPIPEIF_CFG1[7] AVGFILT	
Enable horizontal pixel decimation function.	IPIPEIF_CFG1[1] DECIM	
Set horizontal resizing value.	IPIPEIF_RSZ[6:0] RSZ	
Set the resizer initial position.	IPIPEIF_INIRSZ[13] ALNSYNC	
	IPIPEIF_INIRSZ[12:0] INIRSZ	
Set the data gain (only for RAW data).	IPIPEIF_GAIN[9:0] GAIN	
Set the output clipping value.	IPIPEIF_OCLIP[11:0] OCLIP	

Table 217. ISS ISP IPIPEIF Required Configuration Parameters (continued)

Step	Configuration Required	Value
Set the averaging filter and horizontal pixel decimator function for H3A data path.		
Enable averaging filter function.	IPIPEIF_RSZ3A[8] AVGFILT	
Enable horizontal pixel decimation function.	IPIPEIF_RSZ3A[9] DECIM	
Set horizontal resizing value.	IPIPEIF_RSZ3A[6:0] RSZ	
Set the resizer initial position.	IPIPEIF_RSZ3A[13] ALNSYNC	
	IPIPEIF_RSZ3A[12:0] INIRSZ	

For information about YUV data coming and unpacking from the ISIF module, see Section 3.3.2.13.

3.4.2.2 ISS ISP IPIPEIF Enable/Disable Hardware

When IPIPEIF_CFG1[15:14] INPSRC1 or IPIPEIF_CFG1[3:2] INPSRC2 = 0, the IPIPEIF does not need to be enabled: it receives data from the VP and pushes it to the ISIF, H3A, IPIPE, and RESIZER modules.

If IPIPEIF_CFG1[15:14] INPSRC1 or IPIPEIF_CFG1[3:2] INPSRC2 ≠ 0, the IPIPEIF module begins to fetch data from the BL by setting the IPIPEIF_ENABLE[0] ENABLE bit. Writing the enable bit must be the last step of the configuration.

When the input source is the BL, the IPIPEIF can optionally operate in one-shot mode or continuous mode by setting the IPIPEIF_CFG1[0] ONESHOT parameter. If one-shot mode is enabled, then after enabling the IPIPEIF, the IPIPEIF_ENABLE[0] ENABLE bit is automatically turned off (set to 0) and only a single frame is processed from memory. In this mode, fetching and processing of the frame begins immediately upon setting the IPIPEIF_ENABLE[0] ENABLE bit.

When the input source is the ISIF, processing of the frame is dependent on the timing of the ISIF. To ensure that data from the ISIF is not missed, the IPIPEIF must be enabled before the ISIF. In this way, the IPIPEIF waits for data from the ISIF.

When the IPIPEIF is in continuous mode, it can be disabled by clearing the IPIPEIF_ENABLE[0] ENABLE bit after processing of the last frame. The disable takes place immediately because it is a busy-writable register.

An EOF interrupt can indicate to other modules that the frame treatment is finished. See Section 3.3.2.15 for more information.

3.4.2.3 ISS ISP IPIPEIF Register Accessibility During Frame Processing

There are two types of register access in the IPIPEIF:

- Shadow registers

  These registers/fields can be read and written (if the field is writable) at any time. However, the written values take effect only at the start of a frame (VD rising edge). Reads still return the most recent write even though the settings are not used until the next start of frame. The following are the shadowed registers in the IPIPEIF:

IPIPEIF_PPLN	IPIPEIF_ADDRU	IPIPEIF_CFG1[1] DECIM
IPIPEIF_HNUM	IPIPEIF_ADOFS	IPIPEIF_CFG1[7] AVGFILT
IPIPEIF_VNUM	IPIPEIF_ENABLE[1] SYNCOFF	IPIPEIF_RSZ
IPIPEIF_GAIN	IPIPEIF_RSZ3A[9] DECIM	IPIPEIF_RSZ3A[8] AVGFILT
IPIPEIF_RSZ3A[6:0] RSZ		

- Busy-writable registers

  These registers/fields can be read or written even if the module is busy. Changes to the underlying settings takes place instantaneously. Registers that are not shadowed are busy-writable.

Only for busy-writable registers, the ideal procedure for changing the IPIPEIF registers is IF (PCR.BUSY == 0) OR IF (EOF interrupt occurs):

- Disable IPIPEIF (IPIPEIF_ENABLE[0] ENABLE = 0x0).
- Change registers.
- Enable IPIPEIF (IPIPEIF_ENABLE[0] ENABLE = 0x1).

3.4.2.4 ISS ISP IPIPEIF Interframe Operations

Between frames, it may be necessary to enable or disable functions or to modify the memory pointers. Because several of the registers are shadowed, these modifications can take place any time before the end of the frame, and the data is latched in for the next frame. The host controller can perform these changes upon receiving an interrupt.

When reading input data from the BL, the host must check IPIPEIF_DTUF or the ISP5_IRQSTATUS_RAW2_i[1] IPIPEIF_UDF register (if this event is enabled and mapped to the ISP IRQ lines) during vertical blanking to see if an underflow occurred.

If an underflow occurs it means the system is congested because too much bandwidth is being generated. It is most likely that the scenario being passed is too memory bandwidth-intensive. If the bit is set, software must clear the bit. If the process is taking place from memory-to-memory, the software may try to re-initiate the data flow. Software must decide the sequence to execute after an underflow.

3.4.2.5 ISS ISP IPIPEIF Summary of Constraints

The following is a list of register configuration constraints to adhere to when programming the IPIPEIF module. It can be used as a quick checklist. More detailed register setting constraints can be found in the individual register descriptions.

- If BL is the source for the IPIPEIF:
  - The memory output line offset and address must be on 32-byte boundaries.
  - In DFS block, IPIPEIF_LPFR must be 0, because the first line cannot be fetched.
  - IPIPEIF_PPLN IPIPEIF_HNUM
  - IPIPEIF_LPFR IPIPEIF_VNUM + 1
- There are restrictions on IPIPEIF_CFG1[15:14] INPSRC1 and IPIPEIF_CFG1[3:2] INPSRC2 combinations. See Table 169 for details.

3.4.3 ISS ISP IPIPE Programming Model

This section describes the low-level hardware programming sequences for the configuration and use of the ISS PIPE module.

3.4.3.1 Global Initialization

3.4.3.1.1 Surrounding Modules Global Initialization

This initialization of surrounding modules is based on the integration of the ISS ISP.

3.4.3.1.2 ISS IPIPE Global Initialization

3.4.3.1.2.1 Main Sequence – ISS ISP IPIPE Global Initialization

This procedure initializes the ISS ISP PIPE modules after a POR or software reset.

Table 218. ISS Global Initialization

Step	Register/Bit Field/Programming Model	Value
Set the IPIPE processing modes.	IPIPE_SRC_MODE[0] OST	0x-
Select the IPIPEIF module as a source.	IPIPE_SRC_MODE[1] WRT	0x1
Set the vertical start position of the window to process.	IPIPE_SRC_VPS[15:0] VAL	0x-
Set the horizontal start position of the window to process.	IPIPE_SRC_HPS[15:0] VAL	0x-
Set the vertical size of the processing area.	IPIPE_SRC_VSZ[12:0] VAL	0X-
Set the horizontal size of the processing area.	IPIPE_SRC_HSZ[12:1] VAL	0x-
Enable clk_arm_g0.	IPIPE_GCK_MMR[0] REG	0x-
Enable the clk_pix_g3.	IPIPE_GCK_PIX[3] G3	0x-
Enable the clk_pix_g2.	IPIPE_GCK_PIX[2] G2	0x-
Enable the clk_pix_g1.	IPIPE_GCK_PIX[1] G1	0x-
Enable the clk_pix_g0.	IPIPE_GCK_PIX[0] G0	0x-
Enable IPIPE module.	IPIPE_SRC_EN[0] EN	0x1

3.4.3.1.2.2 Subsequence – ISS ISP IPIPE Defect Pixel Correction Initialization

3.4.3.1.2.2.1 Subsequence – ISS ISP IPIPE LUT Defect Pixel Correction (LUT DPC)

Table 219. ISS ISP IPIPE LUT Defect Pixel Correction (LUT DPC)

Step	Register/Bit Field/Programming Model	Value
Set the address of the first valid data in the LUT.	IPIPE_DPC_LUT_ADR[9:0] ADR	0x-
Set the LUT type.	IPIPE_DPC_LUT_SEL[1] TBL	0x-
Set the size of the LUT.	IPIPE_DPC_LUT_SIZ[9:0] SIZ	0x-
Set the dot replacement correction method .	IPIPE_DPC_LUT_SEL[0] DOT	0x-
Enable LUT DPC.	IPIPE_DPC_LUT_EN[0] EN	0x1

3.4.3.1.2.2.2 Subsequence – ISS ISP IPIPE OTF Defect Pixel Correction (OTF DPC)

Table 220. ISS ISP IPIPE OTF Defect Pixel Correction (OTF DPC)

Step	Register/Bit Field/Programming Model	Value
Set the adaptive OTF DPC method.	IPIPE_DPC_OTF_TYP[1] TYP IPIPE_DPC_OTF_TYP[0]ALG	0x-
IF: Is adaptive OTF DPC method 2 selected?	IPIPE_DPC_OTF_TYP[1] TYP IPIPE_DPC_OTF_TYP[0]ALG	= 0x1 = 0x0
Set the detection thresholds for each color in method 2.	IPIPE_DPC_OTF_2_D_THR_R[11:0] VAL IPIPE_DPC_OTF_2_D_THR_GR[11:0] VAL IPIPE_DPC_OTF_2_D_THR_GB[11:0] VAL IPIPE_DPC_OTF_2_D_THR_B[11:0] VAL	0x-
Set the correction thresholds for each color in method 2.	IPIPE_DPC_OTF_2_C_THR_R[11:0] VAL IPIPE_DPC_OTF_2_C_THR_GR[11:0] VAL IPIPE_DPC_OTF_2_C_THR_GB[11:0] VAL IPIPE_DPC_OTF_2_C_THR_B[11:0] VAL	0x-
EISE IF: Is adaptive OTF DPC method 3 selected?	IPIPE_DPC_OTF_TYP[1] TYP IPIPE_DPC_OTF_TYP[0]ALG	= 0x1 = 0x1

Table 220. ISS ISP IPIPE OTF Defect Pixel Correction (OTF DPC) (continued)

Step	Register/Bit Field/Programming Model	Value
Set the detection thresholds for each color in method 3.	IPIPE_DPC_OTF_3_D_THR[11:6] VAL IPIPE_DPC_OTF_3_D_SPL[5:0] VAL IPIPE_DPC_OTF_3_D_MIN[11:0] VAL IPIPE_DPC_OTF_3_D_MAX[11:0] VAL	0x-
Set the correction thresholds for each color in method 3.	IPIPE_DPC_OTF_3_C_THR[11:6] VAL IPIPE_DPC_OTF_3_C_SLP[5:0] VAL IPIPE_DPC_OTF_3_C_MIN[11:0] VAL IPIPE_DPC_OTF_3_C_MAX[11:0] VAL	0x-
ENDIF		
Enable OTF DPC.	IPIPE_DPC_OTF_EN[0] EN	0x1

3.4.3.1.2.3 Subsequence – ISS ISP IPIPE Noise Filter Initialization

Table 221. ISS ISP IPIPE Noise Filter Initialization

Step	Register/Bit Field/Programming Model	Value
Set the sampling method of green pixels.	IPIPE_D2F_1ST_TYP[7] TYP	0x-
Set the downshift value in LUT reference address.	IPIPE_D2F_1ST_TYP[6:5] SHF	0x-
Set threshold table for the noise filter algorithm.	IPIPE_D2F_1ST_THR_00[9:0] VAL IPIPE_D2F_1ST_THR_01[9:0] VAL IPIPE_D2F_1ST_THR_02[9:0] VAL IPIPE_D2F_1ST_THR_03[9:0] VAL IPIPE_D2F_1ST_THR_04[9:0] VAL IPIPE_D2F_1ST_THR_05[9:0] VAL IPIPE_D2F_1ST_THR_06[9:0] VAL IPIPE_D2F_1ST_THR_07[9:0] VAL	0x-
Set the intensity table, which stores averaging weight.	IPIPE_D2F_1ST_STR_00[4:0]VAL IPIPE_D2F_1ST_STR_01[4:0]VAL IPIPE_D2F_1ST_STR_02[4:0]VAL IPIPE_D2F_1ST_STR_03[4:0]VAL IPIPE_D2F_1ST_STR_04[4:0]VAL IPIPE_D2F_1ST_STR_05[4:0]VAL IPIPE_D2F_1ST_STR_06[4:0]VAL IPIPE_D2F_1ST_STR_07[4:0]VAL	0x-
Set the SPR in 2D-noise filter.	IPIPE_D2F_1ST_SPR_00[4:0] VAL IPIPE_D2F_1ST_SPR_01[4:0] VAL IPIPE_D2F_1ST_SPR_02[4:0] VAL IPIPE_D2F_1ST_SPR_03[4:0] VAL IPIPE_D2F_1ST_SPR_04[4:0] VAL IPIPE_D2F_1ST_SPR_05[4:0] VAL IPIPE_D2F_1ST_SPR_06[4:0] VAL IPIPE_D2F_1ST_SPR_07[4:0] VAL	0x-
Set the noise filter 1 edge detection minimum value.	IPIPE_D2F_1ST_EDG_MIN[10:0] VAL	0x-
Set the noise filter 1 edge detection maximum value.	IPIPE_D2F_1ST_EDG_MAX[10:0] VAL	0x-
Set the spread factor (SPR).	IPIPE_D2F_1ST_TYP[4:0] SPR	0x-
Specify the total gain (GAN) for each color.	IPIPE_LSC_GAN_R[7:0] VAL IPIPE_LSC_GAN_GR[7:0] VAL IPIPE_LSC_GAN_GB[7:0] VAL IPIPE_LSC_GAN_B[7:0] VAL	0x-
Specify the offset (OFT) values.	IPIPE_LSC_OFT_R[7:0] VAL IPIPE_LSC_OFT_GR[7:0] VAL IPIPE_LSC_OFT_GB[7:0] VAL IPIPE_LSC_OFT_B[7:0] VAL	0x-
Specify vertical offset.	IPIPE_LSC_VOFT[12:0] LSC_VOFT IPIPE_LSC_VA2[12:0] VAL IPIPE_LSC_VA1[12:0] VAL IPIPE_LSC_VS[7:4] VS2 IPIPE_LSC_VS[3:0]VS1	0x-

Table 221. ISS ISP IPIPE Noise Filter Initialization (continued)

Step	Register/Bit Field/Programming Model	Value
Specify horizontal offset.	IPIPE_LSC_HOFT[12:0] VAL IPIPE_LSC_HA2[12:0] VAL IPIPE_LSC_HA1[12:0] VAL IPIPE_LSC_HS[7:4] HS2 IPIPE_LSC_HS[3:0] HS1	0x-
Set threshold shift value.	IPIPE_LSC_SHF[3:0] VAL	0x-
Set threshold maximum value.	IPIPE_LSC_MAX [8:0] VAL	0x-
Apply LSC gain to threshold values.	IPIPE_D2F_1ST_TYP[8] LSC	0x1
Enable noise filter 1.	IPIPE_D2F_1ST_EN[0] EN	0x1

3.4.3.1.2.4 Subsequence – ISS ISP IPIPE Green Imbalance Correction (GIC) Initialization

Table 222. ISS ISP IPIPE Green Imbalance Correction (GIC) Initialization

Step	Register/Bit Field/Programming Model	Value
Select algorithm type.	IPIPE_GIC_TYP[0] TYP	0x-
Set the gain value.	IPIPE_GIC_GAN[7:0] VAL	0x-
Set the THR value.	IPIPE_GIC_THR[11:0] VAL	0x-
Set the SLP value.	IPIPE_GIC_SLP[11:0] VAL	0x-
IF : Is NF2 used?	IPIPE_D2F_2ND_EN[0] EN	= 0x1
Replaced the threshold with the value of NF-2.	IPIPE_GIC_TYP[1] SEL	0x1
Set GIC gain value.	IPIPE_GIC_NFGAIN[7:0] VAL	0x-
Apply LSC gain to threshold value.	IPIPE_GIC_TYP[2] LSC	0x-
ENDIF		
Enable the GIC.	IPIPE_GIC_EN[0] EN	0x1

3.4.3.1.2.5 Subsequence – ISS ISP IPIPE White Balance Initialization

Table 223. ISS ISP IPIPE White Balance Initialization

Step	Register/Bit Field/Programming Model	Value
Set the offset before white balance.	IPIPE_WB2_OFT_R[11:0] VAL IPIPE_WB2_OFT_GR[11:0] VAL IPIPE_WB2_OFT_GB[11:0] VAL IPIPE_WB2_OFT_B[11:0] VAL	0x-
Set the white balance gain.	IPIPE_WB2_WGN_R[12:0] VAL IPIPE_WB2_WGN_GR[12:0] VAL IPIPE_WB2_WGN_GB[12:0] VAL IPIPE_WB2_WGN_B[12:0] VAL	0x-

3.4.3.1.2.6 Subsequence – ISS ISP IPIPE Color Filter Array (CFA) Interpolation Initialization

Table 224. ISS ISP IPIPE Color Filter Array (CFA) Interpolation Initialization

Step	Register/Bit Field/Programming Model	Value
Activate CFA and/or DAA.	IPIPE_CFA_MODE[1:0] MODE	0x-
Set the low value of the HP threshold.	IPIPE_CFA_2DIR_HPF_THR[12:0] VAL	0x-
Set the HP slope value.	IPIPE_CFA_2DIR_HPF_SLP[9:0] VAL	0x-
Set a mix threshold.	IPIPE_CFA_2DIR_MIX_THR[12:0] VAL	0x-
Set the slope of the HP.	IPIPE_CFA_2DIR_MIX_SLP[9:0] VAL	0x-

Table 224. ISS ISP IPIPE Color Filter Array (CFA) Interpolation Initialization (continued)

Step	Register/Bit Field/Programming Model	Value
Set the directional threshold.	IPIPE_CFA_2DIR_DIR_TRH[9:0] VAL	0x-
Set the direction slope.	IPIPE_CFA_2DIR_DIR_SLP[6:0]VAL	0x-
Set the ND weight.	IPIPE_CFA_2DIR_NDWT[5:0] VAL	0x-
IF: Is DAA ONLY selected?(1)	IPIPE_CFA_MODE[1:0] MODE	= 0x2
Set the hue fraction.	IPIPE_CFA_MONO_HUE_FRA[5:0] VAL	0x-
Set the edge of the threshold.	IPIPE_CFA_MONO_EDG_THR[7:0] VAL	0x-
Set the threshold minimum value.	IPIPE_CFA_MONO_THR_MIN[9:0] VAL	0x-
Set the minimum slope value .	IPIPE_CFA_MONO_THR_SLP[9:0] VAL	0x-
Set the SLP minimum value.	IPIPE_CFA_MONO_SLP_MIN[9:0] VAL	0x-
Set the threshold slope value.	IPIPE_CFA_MONO_SLP_SLP[9:0] VAL	0x-
Set the LP weight.	IPIPE_CFA_MONO_LPWT[5:0] VAL	0x-
ENDIF		

(1) The use of DAA ONLY is not recommended.

3.4.3.1.2.7 Subsequence – ISS ISP IPIPE RGB2RGB Blending Module Initialization

Table 225. ISS ISP IPIPE RGB2RGB Blending Module Initialization

Step	Register/Bit Field/Programming Model	Value
Set gain range.	IPIPE_RGB1_MUL_RR[11:0] VAL IPIPE_RGB1_MUL_GR[11:0] VAL IPIPE_RGB1_MUL_BR[11:0] VAL IPIPE_RGB1_MUL_RG[11:0] VAL IPIPE_RGB1_MUL_GG[11:0] VAL IPIPE_RGB1_MUL_BG[11:0] VAL IPIPE_RGB1_MUL_RB[11:0] VAL IPIPE_RGB1_MUL_GB[11:0] VAL IPIPE_RGB1_MUL_BB[11:0] VAL	0x-
Se the offset range for each component.	IPIPE_RGB1_OFT_OR[12:0] VAL IPIPE_RGB1_OFT_OG[12:0] VAL IPIPE_RGB1_OFT_OB[12:0] VAL	0x-

3.4.3.1.2.8 Subsequence – ISS ISP IPIPE Gamma Correction Module Initialization

Table 226. ISS ISP IPIPE Gamma Correction Module Initialization

Step	Register/Bit Field/Programming Model	Value
Select the gamma table.	IPIPE_GMM_CFG[4] TBL	0x-
Set the size of the gamma table.	IPIPE_GMM_CFG[6:5]SIZ	0x-
(optional) Insert bypass bit for each color.	IPIPE_GMM_CFG[0] BYPR IPIPE_GMM_CFG[1] BYPG IPIPE_GMM_CFG[2] BYPB	0x1

3.4.3.1.2.9 Subsequence – ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix Initialization

Table 227. ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix Initialization

Step	Register/Bit Field/Programming Model	Value
Set the gain range.	IPIPE_RGB2_MUL_RR[10:0] VAL IPIPE_RGB2_MUL_GR[10:0] VAL IPIPE_RGB2_MUL_BR[10:0] VAL IPIPE_RGB2_MUL_RG[10:0] VAL IPIPE_RGB2_MUL_GG[10:0] VAL IPIPE_RGB2_MUL_BG[10:0] VAL IPIPE_RGB2_MUL_RB[10:0] VAL IPIPE_RGB2_MUL_GB[10:0] VAL IPIPE_RGB2_MUL_BB[10:0] VAL	0x-
Set the offset.	IPIPE_RGB2_OFT_OR[10:0] VAL IPIPE_RGB2_OFT_OG[10:0] VAL IPIPE_RGB2_OFT_OB[10:0] VAL	0x-

3.4.3.1.2.10 Subsequence – ISS ISP IPIPE 3D-LUT Color Conversion Initialization

Table 228. ISP IPIPE 3D-LUT Color Conversion Initialization

Step	Register/Bit Field/Programming Model	Value
Enable 3D-LUT color conversion.	IPIPE_3DLUT_EN[0] EN	0x1

3.4.3.1.2.11 Subsequence – ISS ISP IPIPE RGB2YCbCr Conversion Matrix Initialization

Table 229. ISS ISP IPIPE RGB2YCbCr Conversion Matrix Initialization

Step	Register/Bit Field/Programming Model	Value
Set the brightness control.	IPIPE_YUV_ADJ[15:8] BRT	0x-
Set the contrast control.	IPIPE_YUV_ADJ[7:0] CRT	0x-
Configure the gain range.	IPIPE_YUV_MUL_RY[11:0]VAL IPIPE_YUV_MUL_GY[11:0]VAL IPIPE_YUV_MUL_BY[11:0]VAL IPIPE_YUV_MUL_RCB[11:0]VAL IPIPE_YUV_MUL_GCB[11:0]VAL IPIPE_YUV_MUL_BCB[11:0]VAL IPIPE_YUV_MUL_RCR[11:0]VAL IPIPE_YUV_MUL_GCR[11:0]VAL IPIPE_YUV_MUL_BCR[11:0]VAL	0x-
Set the output offset value for Y.	IPIPE_YUV_OFT_Y[10:0] VAL	0x-
Set the output offset value for Cr.	IPIPE_YUV_OFT_CR[10:0] VAL	0x-
Set the output offset value for Cb.	IPIPE_YUV_OFT_CB[10:0] VAL	0x-

3.4.3.1.2.12 Subsequence – ISS ISP IPIPE Global Brightness and Contrast Enhancement Initialization

Table 230. ISS ISP IPIPE Global Brightness and Contrast Enhancement Initialization

Step	Register/Bit Field/Programming Model	Value
Set the global brightness and contrast enhancement method.	IPIPE_GBCE_TYP[0] TYP	0x-
Enable global brightness and contrast enhancement.	IPIPE_GBCE_EN[0] EN	0x1

3.4.3.1.2.13 Subsequence – ISS ISP IPIPE 4:2:2 Conversion Module Initialization

Table 231. ISS ISP IPIPE 4:2:2 Conversion Module Initialization

Step	Register/Bit Field/Programming Model	Value
If: RAW BAYER data set as an input?	IPIPE_SRC_FMT.[1:0] FMT	= 0x0
Select the Y and Cb/Cr sampling point.	IPIPE_YUV_PHS[0] POS	0x-
Enable 4:2:2 conversion module.	IPIPE_YUV_PHS[1] PLF	0x-
ELSE		

3.4.3.1.2.14 Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization

Table 232. ISS ISP IPIPE 2D Edge Enhancer Initialization

Step	Register/Bit Field/Programming Model	Value
Set the merging method.	IPIPE_YEE_TYP[0] SEL	0x-
Set the downshift length of HPF in the edge enhancer.	IPIPE_YEE_SHF[3:0] SHF	0x-
Set the edge enhancer lower threshold before referring to the LUT.	IPIPE_YEE_THR[5:0] VAL	0x-
Set the multiplier coefficient in the HPF.	IPIPE_YEE_MUL_00[9:0] VAL IPIPE_YEE_MUL_01[9:0] VAL IPIPE_YEE_MUL_02[9:0] VAL IPIPE_YEE_MUL_10[9:0] VAL IPIPE_YEE_MUL_11[9:0] VAL IPIPE_YEE_MUL_12[9:0] VAL IPIPE_YEE_MUL_20[9:0] VAL IPIPE_YEE_MUL_21[9:0] VAL IPIPE_YEE_MUL_22[9:0] VAL	0x-
Set the edge sharpener HPF value lower limit.	IPIPE_YEE_E_THR_1[11:0] VAL	0x-
Set the edge sharpener HPF value upper limit.	IPIPE_YEE_E_THR_2[5:0] VAL	0x-
Set the edge sharpener gain value on gradient.	IPIPE_YEE_G_GAN[7:0] VAL	0x-
Set the edge sharpener gain value.	IPIPE_YEE_E_GAN[7:0] VAL	0x-
Set the edge sharpener offset value on gradient.	IPIPE_YEE_G_OFT[5:0] VAL	0x-
Enable the 2D edge enhancer.	IPIPE_YEE_EN[0] EN	0x1

3.4.3.1.2.15 Subsequence – ISS ISP IPIPE Chroma Artifact Reduction Initialization

Table 233. ISS ISP IPIPE Chroma Artifact Reduction Initialization

Step	Register/Bit Field/Programming Model	Value
Select mode of the fault color suppression.	IPIPE_CAR_TYP[0] TYP	0x-
Set the switching function.	IPIPE_CAR_SW[7:0] SW0 IPIPE_CAR_SW[15:8] SW1	0x-
Set the type of the HPF filter.	IPIPE_CAR_HPF_TYP[2:0] TYP	0x-
Set the threshold of the gain function for HPF value.	IPIPE_CAR_HPF_THR[7:0] VAL	0x-
Set the downshift value for HPF.	IPIPE_CAR_HPF_SHF[1:0] VAL	0x-
Set the intensity of the gain function for HPF value.	IPIPE_CAR_GN1_GAN[7:0] VAL	0x-
Set the downshift value of the gain function on HPF value.	IPIPE_CAR_GN1_SHF[2:0] VAL	0x-
Set the lower limit of the gain function on HPF value.	IPIPE_CAR_GN1_MIN[8:0] VAL	0x-
Set the intensity of the gain function for Chroma value.	IPIPE_CAR_GN2_GAN[7:0] VAL	0x-

Table 233. ISS ISP IPIPE Chroma Artifact Reduction Initialization (continued)

Step	Register/Bit Field/Programming Model	Value
Set the down shift value of the gain function on Chroma value.	IPIPE_CAR_GN2_SHF[3:0] VAL	0x-
Set the lower limit of the gain function on Chroma value.	IPIPE_CAR_GN2_MIN[8:0] VAL	0x-
Enable Chroma artifact reduction.	IPIPE_CAR_EN[0] EN	0x1

3.4.3.1.2.16 Subsequence – ISS ISP IPIPE Chroma Gain Suppression Initialization

Table 234. ISS ISP IPIPE Chroma Gain Suppression Initialization

Step	Register/Bit Field/Programming Model	Value
Set the threshold 1 value of thrY1.	IPIPE_CGS_GN1_L_THR[7:0] VAL	0x-
Set the gain value of gY1	IPIPE_CGS_GN1_L_GAIN[7:0] VAL	0x-
Set the shift value of the shfY1.	IPIPE_CGS_GN1_L_SHF[7:0] VAL	0x-
Set threshold 2 of the gain function for Y value.	IPIPE_CGS_GN1_H_THR[7:0] VAL	0x-
Set slope 2 of the gain function for Y value.	IPIPE_CGS_GN1_H_GAIN[7:0] VAL	0x-
Set the lower limit 1 of the gain function 1.	IPIPE_CGS_GN1_L_MIN[7:0] VAL	0x-
Set limit 2 of the gain function.	IPIPE_CGS_GN1_H_MIN[7:0] VAL	0x-
Set the shift value for Y2.	IPIPE_CGS_GN2_L_SHF[7:0] VAL	0x-
Set slope 2 of the gain function for Y value.	IPIPE_CGS_GN2_L_THR[7:0] VAL	0x-
Set the downshift value 3 of the gain function on Y.	IPIPE_CGS_GN2_L_GAIN[7:0] VAL	0x-
Set the downshift value of function Y.	IPIPE_CGS_GN2_L_SHF[7:0] VAL	0x-
Set the lower limit 3 of the gain function.	IPIPE_CGS_GN2_L_MIN[7:0] VAL	0x-
Enable Chroma gain suppression.	IPIPE_CGS_EN[0] EN	0x1

3.4.3.1.2.17 Subsequence – ISS ISP IPIPE Histogram Initialization

Table 235. ISS ISP IPIPE Histogram Initialization

Step	Register/Bit Field/Programming Model	Value
Select input source.	IPIPE_HST_SEL[2] SEL	0x-
Select the Bayer mode.	IPIPE_HST_SEL[1:0] TYP	0x-
Set the number of bins.	IPIPE_HST_PARA[13:12] BIN	0x-
Enable region(area) 0.	IPIPE_HST_PARA[0] RGN0	0x-
Enable region(area) 1.	IPIPE_HST_PARA[1] RGN1	0x-
Enable region(area) 2.	IPIPE_HST_PARA[2] RGN2	0x-
Enable region(area) 3.	IPIPE_HST_PARA[3] RGN3	0x-
Set the vertical position of region 0.	IPIPE_HST_0_VPS[12:1] VAL	0x-
Set the vertical position of region 1.	IPIPE_HST_1_VPS[12:1] VAL	0x-
Set the vertical position of region 2.	IPIPE_HST_2_VPS[12:1] VAL	0x-
Set the vertical position of region 3.	IPIPE_HST_3_VPS[12:1] VAL	0x-
Set the horizontal position of region 0.	IPIPE_HST_0_HPS[12:1] VAL	0x-
Set the horizontal position of region 1.	IPIPE_HST_0_HPS[12:1] VAL	0x-
Set the horizontal position of region 2.	IPIPE_HST_0_HPS[12:1] VAL	0x-
Set the horizontal position of region 3.	IPIPE_HST_0_HPS[12:1] VAL	0x-
Set the vertical size of region 0.	IPIPE_HST_0_VSZ[12:1] VAL	0x-
Set the vertical size of region 1.	IPIPE_HST_1_VSZ[12:1] VAL	0x-
Set the vertical size of region 2.	IPIPE_HST_2_VSZ[12:1] VAL	0x-

Table 235. ISS ISP IPIPE Histogram Initialization (continued)

Step	Register/Bit Field/Programming Model	Value
Set the vertical size of region 3.	IPIPE_HST_3_VSZ[12:1] VAL	0x-
Set the horizontal size of region 0.	IPIPE_HST_0_HSZ[12:1] VAL	0x-
Set the horizontal size of region 1.	IPIPE_HST_1_HSZ[12:1] VAL	0x-
Set the horizontal size of region 2.	IPIPE_HST_2_HSZ[12:1] VAL	0x-
Set the horizontal size of region 3.	IPIPE_HST_3_HSZ[12:1] VAL	0x-
Enable selection of the color pattern 0 (R).	IPIPE_HST_PARA[4] COL0	0x-
Enable selection of the color pattern 1 (G).	IPIPE_HST_PARA[5] COL1	0x-
Enable selection of the color pattern 2 (B).	IPIPE_HST_PARA[6] COL2	0x-
Enable selection of the color pattern 3 (Y).	IPIPE_HST_PARA[7] COL3	0x-
Set the shift length of the input data.	IPIPE_HST_PARA[11:8] SHF	0x-
Enable/disable histogram memory clear.	IPIPE_HST_TBL[1] CLR	0x-
Select table for store the histogram data.	IPIPE_HST_TBL[1] CLR	0x-
IF: Is input from noise filter?	IPIPE_HST_SEL[2] SEl	0x0
Gain for color for R.	IPIPE_HST_MUL_R[7:0]GAIN	0x-
Gain for color for GR.	IPIPE_HST_MUL_GR[7:0]GAIN	0x-
Gain for color for GB.	IPIPE_HST_MUL_GB[7:0]GAIN	0x-
Gain for color for B.	IPIPE_HST_MUL_B[7:0]GAIN	0x-
ENDIF		
Select processing mode.	IPIPE_HST_MODE[0]OST	0x-
Enable histogram.	IPIPE_HST_EN[0] EN	0x1

3.4.3.1.2.18 Subsequence – ISS ISP IPIPE Boxcar Initialization

Table 236. ISS ISP IPIPE Boxcar Initialization

Step	Register/Bit Field/Programming Model	Value
Select the processing mode.	IPIPE_BOX_MODE[0] OST	0x-
Set the size of blocks.	IPIPE_BOX_TYP[0] SEL	0x-
Set the higher 16 bits of the first address of output in memory.	IPIPE_BOX_SDR_SAD_H[15:0] VAL	0x-
Set the lower 16 bits of the first address of output in memory.	IPIPE_BOX_SDR_SAD_L[15:5] VAL	0x-
Set the downshift value.	IPIPE_BOX_SHF[2:0] VAL	0x-
Enable boxcar.	IPIPE_BOX_EN[0] EN	0x1

3.4.3.1.2.19 Subsequence – ISS ISP IPIPE Boundary Signal Calculator Initialization

Table 237. ISS ISP IPIPE Boundary Signal Calculator Initialization

Step	Register/Bit Field/Programming Model	Value
Select processing mode.	IPIPE_BSC_MODE[0] OST	0x-
Enable row sampling.	IPIPE_BSC_TYP[2] REN	0x1
Enable column sampling.	IPIPE_BSC_TYP[3] CEN	0x1
Selects the element to be summed. (Y, Cb or Cr)	IPIPE_BSC_TYP[0:1] COL	0x-
Set the horizontal position of the first pixel to be summed.	IPIPE_BSC_ROW_HPOS[12:0] VAL	0x-
Set the vertical position of the first pixel to be summed.	IPIPE_BSC_ROW_VPOS[12:0] VAL	0x-

Table 237. ISS ISP IPIPE Boundary Signal Calculator Initialization (continued)

Step	Register/Bit Field/Programming Model	Value
Set the horizontal interval of the pixels in a row to be summed.	IPIPE_BSC_ROW_HSKIP[4:0] VAL	0x-
Set the vertical interval of the pixels in a row to be summed.	IPIPE_BSC_ROW_VSKIP[4:0] VAL	0x-
Set the horizontal number of samples in the area covered by a row sum vector.	IPIPE_BSC_ROW_HNUM[12:0] VAL	0x-
Set the height of the area covered by a row sum vector.	IPIPE_BSC_ROW_VNUM[12:1] VAL	0x-
Set the number of row sum vectors.	IPIPE_BSC_ROW_VCT[1:0] VAL	0x-
Set the downshift value for row sum vectors.	IPIPE_BSC_ROW_SHF[2:0] VAL	0x-
Set the horizontal position of the first sampling pixel.	IPIPE_BSC_COL_HPOS[12:0] VAL	0x-
Set the vertical position of the first sampling pixel.	IPIPE_BSC_COL_VPOS[12:0] VAL	0x-
Set the horizontal interval of the columns.	IPIPE_BSC_COL_HSKIP[4:0] VAL	0x-
Set the vertical interval of the pixels in a column to be summed.	IPIPE_BSC_COL_VSKIP[4:0] VAL	0x-
Set the width of the area covered by a column sum vector.	IPIPE_BSC_COL_HNUM[12:1] VAL	0x-
Set the vertical number of samples in the area covered by a column sum vector.	IPIPE_BSC_COL_VNUM[12:0] VAL	0x-
Set the number of column sum vectors.	IPIPE_BSC_COL_VCT[1:0] VAL	0x-
Set the downshift value for column sum vectors.	IPIPE_BSC_COL_SHF[2:0] VAL	0x-
Enable boundary signal calculator.	IPIPE_BSC_EN[0] EN	0x-

3.4.3.2 ISS ISP IPIPE Operational Modes Configuration

3.4.3.2.1 ISS ISP IPIPE Processing Path: Case 1 Configuration

Table 238. ISS ISP IPIPE Processing Path: Case 1 Configuration

Step	Register/Bit Field/Programming Model	Value
Select processing path: Case 1	IPIPE_SRC_FMT[1:0] FMT	0x0
Configure IPIPE DPC.	See Section 3.4.3.1.2.2, *Subsequence – ISS ISP IPIPE Defect Pixel Correction Initialization*.	
Configure IPIPE 2D noise filter.	See Section 3.4.3.1.2.3, *Subsequence – ISS ISP IPIPE Noise Filter Initialization*.	
Configure IPIPE GIC.	See Section 3.4.3.1.2.4, *Subsequence – ISS ISP IPIPE Green Imbalance Correction (GIC) Initialization*.	
Configure IPIPE white balance.	See Section 3.4.3.1.2.5, *Subsequence – ISS ISP IPIPE White Balance Initialization*.	
Configure IPIPE color filter array (CFA) interpolation.	See Section 3.4.3.1.2.6, *Subsequence – ISS ISP IPIPE Color Filter Array (CFA) Interpolation Initialization*.	
Configure IPIPE RGB2RGB blending module.	See Section 3.4.3.1.2.7, *Subsequence – ISS ISP IPIPE RGB2RGB Blending Module Initialization*.	
Configure IPIPE gamma correction module.	See Section 3.4.3.1.2.8, *Subsequence – ISS ISP IPIPE Gamma Correction Module Initialization*.	
Configure IPIPE 2nd RGB2RGB conversion matrix.	See Section 3.4.3.1.2.9, *Subsequence – ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix Initialization*.	
Enable 3D-LUT color conversion.	IPIPE_3DLUT_EN[0] EN	0x1
Configure IPIPE RGB2YCbCr conversion matrix.	See Section 3.4.3.1.2.11, *Subsequence – ISS ISP IPIPE RGB2YCbCr Conversion Matrix Initialization*.	

Table 238. ISS ISP IPIPE Processing Path: Case 1 Configuration (continued)

Step	Register/Bit Field/Programming Model	Value
Configure IPIPE global brightness and contrast enhancement.	See Section 3.4.3.1.2.12, *Subsequence – ISS ISP IPIPE Global Brightness and Contrast Enhancement Initialization*.	
Configure IPIPE 4:2:2 conversion module.	See Section 3.4.3.1.2.13, *Subsequence – ISS ISP IPIPE 4:2:2 Conversion Module Initialization*.	
Configure IPIPE 2D edge enhancer.	See Section 3.4.3.1.2.14, *Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization*.	
Configure IPIPE Chroma artifact reduction.	See Section 3.4.3.1.2.15, *Subsequence – ISS ISP IPIPE Chroma Artifact Reduction Initialization*.	
Configure IPIPE Chroma gain suppression.	See Section 3.4.3.1.2.16, *Subsequence – ISS ISP IPIPE Chroma Gain Suppression Initialization*.	
Enable IPIPE module.	IPIPE_SRC_EN[0] EN	0x1

3.4.3.2.2 ISS ISP IPIPE Processing Path: Case 2 Configuration

Table 239. ISS ISP IPIPE Processing Path: Case 2 Configuration

Step	Register/Bit Field/Programming Model	Value
Select processing path: Case 2.	IPIPE_SRC_FMT[1:0] FMT	0x1
Configure IPIPE DPC.	See Section 3.4.3.1.2.2, *Subsequence – ISS ISP IPIPE Defect Pixel Correction Initialization*.	
Configure IPIPE 2D noise filter.	See Section 3.4.3.1.2.3, *Subsequence – ISS ISP IPIPE Noise Filter Initialization*.	
Configure IPIPE GIC.	See Section 3.4.3.1.2.4, *Subsequence – ISS ISP IPIPE Green Imbalance Correction (GIC) Initialization*.	
Configure IPIPE white balance.	See Section 3.4.3.1.2.5, *Subsequence – ISS ISP IPIPE White Balance Initialization*.	
Enable IPIPE module.	IPIPE_SRC_EN[0] EN	0x1

3.4.3.2.3 ISS ISP IPIPE Processing Path: Case 3 Configuration

Table 240. ISS ISP IPIPE Processing Path: Case 3 Configuration

Step	Register/Bit Field/Programming Model	Value
Select processing path: Case 3.	IPIPE_SRC_FMT[1:0] FMT	0x3
Configure IPIPE 2D edge enhancer.	See Section 3.4.3.1.2.14, *Subsequence - ISS ISP IPIPE 2D Edge Enhancer Initialization*.	
Configure IPIPE Chroma artifact reduction.	See Section 3.4.3.1.2.15, *Subsequence - ISS ISP IPIPE Chroma Artifact Reduction Initialization.*	
Configure IPIPE Chroma gain suppression.	See Section 3.4.3.1.2.16, *Subsequence - ISS ISP IPIPE Chroma Gain Suppression Initialization*.	
Enable IPIPE module.	IPIPE_SRC_EN[0] EN	0x1

3.4.3.2.4 ISS ISP IPIPE Processing Path: Case 4 Configuration

Table 241. ISS ISP IPIPE Processing Path: Case 4 Configuration

Step	Register/Bit Field/Programming Model	Value
Select processing path: Case 4.	IPIPE_SRC_FMT[1:0] FMT	0x2
Configure IPIPE boxcar.	See Section 3.4.3.1.2.18, *Subsequence – ISS ISP IPIPE Boxcar Initialization*.	

Table 241. ISS ISP IPIPE Processing Path: Case 4 Configuration (continued)

Step	Register/Bit Field/Programming Model	Value
Configure IPIEE histogram.	See Section 3.4.3.1.2.17, *Subsequence – ISS ISP IPIPE Histogram Initialization*.	
Enable IPIPE module.	IPIPE_SRC_EN[0] EN	0x1

3.4.4 ISS ISP RSZ Programming Model

This section discusses programming configuration steps related to software control of the RSZ. It lists the registers that must be programmed in different modes, describes how to enable and disable the RSZ and how to check the status of the image resizing procedure, and discusses the different register access types and several programming constraints.

3.4.4.1 ISS ISP RSZ Hardware Setup/Initialization

This section discusses the configuration of the RSZ required before image processing can begin.

3.4.4.1.1 Initialization

This section identifies the requirements for initializing the surrounding modules when the RSZ module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the RSZ. For more information, see Section 3.3.4.4, *ISS ISP RSZ Integration.*

Table 242. ISS ISP RSZ Surrounding Modules Global Initialization

Surrounding Modules	Comments
PRCM	Module interface and functional clocks must be enabled. RSZ is part of ISP, which is part of ISS. See Section 3.2.1.1, *ISS ISP Clocks*.
(optional) MPU INTC	The MPU INTC must be configured to enable interrupts from the RSZ. For information about how to configure the local ISS interrupt channels, see Section 3.2.2, *ISS ISP Interrupt Tree*. Then, to configure the outside ISS boundary channels, see the *Interrupt Controllers* chapter in the TRM.
(optional) IPIPE	Configure the IPIPE module to process and pass data to the RSZ module. See Section 3.4.3, *ISS ISP IPIPE Programming Model* .
(optional) IPIPEIF	Configure the IPIPEIF module to process and pass data to the RSZ module. See Section 3.4.2, *ISS ISP IPIPEIF Programming Model* .
(optional) BL	Configure the BL for reading back data from memory. See Section 3.4.6, *ISS ISP BL Programming Model* .
(optional) CSI2	If the data comes from the sensor, configure the CSI2-A interface. See Section 2.6.4, *ISS CSI2 Programming Model*.
(optional) SC	If the data comes from the sensor, configure the SC interface.

NOTE: The MPU INTC configurations are necessary, if the interrupt-based communication modes are used. The IPIPE or IPIPEIF configuration is also necessary for putting data into the RSZ. Moreover, if data comes from memory, the BL must be configured. If data comes from the image sensor CSI2 or SC, the interface must be configured.

3.4.4.1.2 ISS ISP RSZ Initial Register Setup

Before enabling the RSZ, the hardware must be properly configured through register writes. Table 243 identifies the sequence to be used before enabling the hardware.

Table 243. ISS ISP RSZ Initial Register Setup

Step	Register/Bit Field/Programming Model	Value
Set the MMR clock to enable RSZ register read/write access.	RSZ_GCK_MMR[0] MMR	0x1
Determine whether or not a bypass mode will be used and set the RSZ core functional clock accordingly.	RSZ_GCK_SDR[0] CORE	0x-
IF: Is the core clock enabled?	RSZ_GCK_SDR[0] CORE	= 0x1
Enable RSZ engine clocks.	RSZ_SYSCONFIG[8] RSZA_CLK_EN RSZ_SYSCONFIG[9] RSZB_CLK_EN	0x1
ELSE: The input data buffer in not used?	RSZ_GCK_SDR[0] CORE	= 0x0
Set the bypass mode accordingly.	RSZ_SRC_FMT0[1] BYPASS	0x-
ENDIF		
Enable the RSZ core clock.	RSZ_GCK_SDR[0] CORE	0x1
Determine the upscale ratio and functional clock, and adjust the fractional clock divider as appropriate.	RSZ_FRACDIV[15:0] RSZ_FRACDIV	0x-
Set the low threshold of the RSZ input data buffer as appropriate.	RSZ_IN_FIFO_CTRL[28:16] THRLD_LOW	0x-
Set the high threshold of the RSZ input data buffer as appropriate.	RSZ_IN_FIFO_CTRL[12:0] THRLD_HIGH	0x-
Determine whether IPIPE or IPEPIF will be the source of input and set this bit accordingly.	RSZ_SRC_FMT0[0] SEL	0x-
Determine whether the RSZ will process as long as data is present in the input buffer, or whether it will wait for a WEN signal from the input source (IPIPE or IPIPEIF) to process lines arrived only during WEN high state.	RSZ_SRC_MODE[1] WRT	0x-
Select the processing mode (one shot or free running). This bit controls the RSZ module. There are also additional mode settings for the two RSZ engines within. After RSZ reset, the mode is automatically set to free running.	RSZ_SRC_MODE[0] OST	0x-
Configure the number of interrupt intervals for writing lines into CBUFF for the two RSZ engines.	RSZ_IRQ_RZA[12:0] RZA RSZ_IRQ_RZB[12:0] RZB	0x-
Enable the RSZ module.	RSZ_SRC_EN[0] EN	0x1
Enable the RSZ engine A.	RZA_EN[0] EN	0x1
Enable the RSZ engine B.	RZB_EN[0] EN	0x1

NOTE: RSZ engine A or RSZ engine B must be enabled after setting all needed configuration parameters.

3.4.4.1.3 ISS ISP RSZ Reset Behavior

Because the RSZ module has no software reset, software can issue one at the ISP level. Moreover, upon hardware reset, all the registers in the RSZ are reset to their reset values. Table 244 identifies the proper software sequence before and after RSZ reset issues at the ISP level.

Table 244. ISS ISP RSZ Reset Behavior

Step	Register/Bit Field/Programming Model	Value
ELSE IF: Is FIFO overflow or blanking error present?	ISP5_IRQSTATUS_i[18] RSZ_FIFO_IN_OVF or ISP5_IRQSTATUS_[19] RSZ_FIFO_IN_BLK_ERR	= 0x1
Clear the FIFO blanking event.	ISP5_IRQENABLE_CLR_i[19] RSZ_FIFO_IN_BLK_ERR	0x1
Clear the overflow event.	SP5_IRQENABLE_CLR_i[18] RSZ_FIFO_IN_OVF	0x1
ELSE: No FIFO overflow or blanking error is present?		
Wait until there is no DMA process and the module ready for reset.	ISP5_SYSCONFIG[1] SOFTRESET and ISP5_IRQSTATUS_i[15] RSZ_INT_DMA	= 0x0
ENDIF		
Disable the source data.	RSZ_SRC_EN[0] EN	0x0
Reset the RSZ at the ISP level.	ISP5_SYSCONFIG[1] SOFTRESET	0x1
Check whether the RSZ has been reset properly.	ISP5_SYSCONFIG[1] SOFTRESET	0x-

3.4.4.2 ISS ISP RSZ Global Image Processing Settings

This section discusses the configuration of the RSZ global settings before/during image frame processing. Table 245 identifies the global RSZ functional sequence, which includes global control, frame settings, bandwidth control, and reversal output image frames.

Table 245. ISS ISP RSZ Global Image Processing Settings

Step	Register/Bit Field/Programming Model	Value
Initialize the RSZ module.	See Section 3.4.4.1.	
Determine input data type and configure the source parameters.	See Section 3.4.4.2.1.	
Determine if vertical flip of the output image will be performed by the two RSZ engines and set accordingly.	RSZ_SEQ[1] VRVA	0x-
	RSZ_SEQ[3] VRVB	
Determine if horizontal flip of the output image will be performed by the two RSZ engines and set accordingly.	RSZ_SEQ[0] HRVA	0x-
	RSZ_SEQ[2] HRVB	
Set the baseline address of the RSZ output to CBUFF. Best performance can be achieved by assigning address on a 128-byte boundary.	RZA_SDR_Y_BAD_H[15:0] Y_BAD_H	0x-
	RZA_SDR_Y_BAD_L[15:0] Y_BAD_L	
	RZB_SDR_Y_BAD_H[15:0] Y_BAD_H	
	RZB_SDR_Y_BAD_L[15:0] Y_BAD_L	
	RZA_SDR_C_BAD_H[15:0] C_BAD_H	
	RZA_SDR_C_BAD_L[15:0] C_BAD_L	
	RZB_SDR_C_BAD_H[15:0] C_BAD_H	
	RZB_SDR_C_BAD_L[15:0] C_BAD_L	

Table 245. ISS ISP RSZ Global Image Processing Settings (continued)

Step	Register/Bit Field/Programming Model	Value
Set the start address of the RSZ output to CBUFF. The first data output will be written to this address. If the first line of a frame is written at the beginning of the CBUFF memory, then SAD = BAD and PTR_S = 0. It is strongly advised to set up this address on a 128-byte boundary, which will lead to the best performance.	RZA_SDR_Y_SAD_H[15:0] Y_SAD_H	0x-
	RZA_SDR_Y_SAD_L[15:0] Y_SAD_L	
	RZB_SDR_Y_SAD_H[15:0] Y_SAD_H	
	RZB_SDR_Y_SAD_L[15:0] Y_SAD_L	
	RZA_SDR_C_SAD_H[15:0] C_SAD_H	
	RZA_SDR_C_SAD_L [15:0] C_SAD_L	
	RZB_SDR_C_SAD_H[15:0] C_SAD_H	
	RZB_SDR_C_SAD_L[15:0] C_SAD_L	
Set the start pointer of the CBUFF internal counter. It must be set up as PTR_S = (SAD – BAD) / OFT. PTR_S is expressed in number of lines.	RZA_SDR_Y_PTR_S[12:0] Y_PTR_S	0x-
	RZB_SDR_Y_PTR_S[12:0] Y_PTR_S	
	RZA_SDR_C_PTR_S[12:0] C_PTR_S	
	RZB_SDR_C_PTR_S[12:0] C_PTR_S	
Set the end pointer of the CBUFF internal counter. PTR_E is expressed in number of lines. The CBUFF can contain up to PTR_E lines.	RZA_SDR_Y_PTR_E[12:0] Y_PTR_E	0x-
	RZB_SDR_Y_PTR_E[12:0] Y_PTR_E	
	RZA_SDR_C_PTR_E[12:0] C_PTR_E	
	RZB_SDR_C_PTR_E[12:0] C_PTR_E	
Set the line of offset expressed in bytes between two lines in the CBUFF (Line 0 = SAD, Line 1 = SAD + 1 x OFT, Line 2 = SAD + 2 x OFT). PTR_E is expressed in number of lines. The CBUFF can contain up to PTR_E lines. Note: OFT does not necessarily correspond to the size of a line in a frame; it can be much bigger. The line offset must be a multiple of 128 bytes.	RZA_SDR_Y_OFT[16:0] Y_OFT	0x-
	RZB_SDR_Y_OFT[16:0] Y_OFT	
	RZA_SDR_C_OFT[16:0] C_OFT	
	RZB_SDR_C_OFT[16:0] C_OFT	
Determine output data type, whether it will be flipped, and configure the output memory addresses.	See Section 3.4.4.2.2.	

NOTE: When data output is 4:2:2 or 4:2:0-Y only, the following values are not needed:

- RZx_SDR_C_BAD_x
- RZx_SDR_C_SAD_x
- RZx_SDR_C_PTR_S
- RZx_SDR_C_PTR_E
- RZx_SDR_C_OFT

When output is 4:2:0-C only, the following values are not needed:

- RZx_SDR_Y_BAD_x
- RZx_SDR_Y_SAD_x
- RZx_SDR_Y_PTR_S
- RZx_SDR_Y_PTR_E
- RZx_SDR_Y_OFT

3.4.4.2.1 ISS ISP RSZ Global Image Processing Settings – Subsequence 1

The procedure shown in Figure 271 determines which of the RSZ engines is enabled and the type of input data, and configures the source data parameters.

Figure 271. ISS ISP RSZ Global Image Processing Settings – Subsequence 1

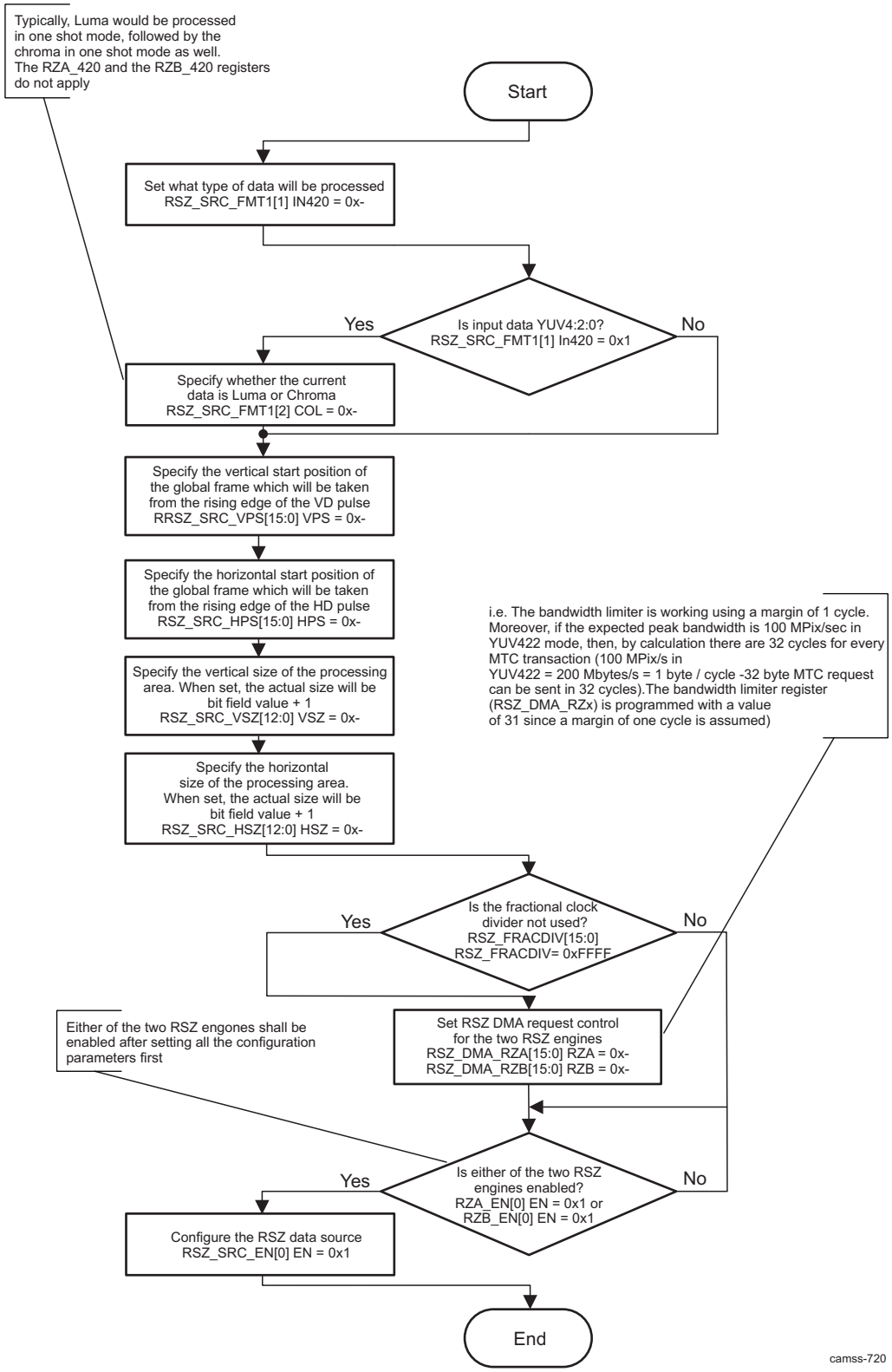


Table 246. Register Call Summary for ISS ISP RSZ Global Image Processing Settings – Subsequence 1

Register Name	Register Name	Register Name
RZA_420[1] CEN	RZA_420[0] YEN	RSZ_SRC_FMT1[1] IN420
RZB_420[0] CEN	RZB_420[1] YEN	RSZ_SRC_FMT1[2] COL
RSZ_SRC_EN[0] EN	RSZ_SRC_VPS[15:0] VPS	RSZ_SRC_HPS[15:0] HPS
RSZ_SRC_VSZ[12:0] VSZ	RSZ_SRC_HSZ[12:0] HSZ	RSZ_FRACDIV[15:0] RSZ_FRACDIV
RSZ_DMA_RZA[15:0] RZA	RSZ_DMA_RZB[15:0] RZB	

3.4.4.2.2 ISS ISP RSZ Global Image Processing Settings – Subsequence 2

The procedure shown in Figure 272 determines the type of output data and whether it will be flipped. Then it configures the RSZ engines accordingly.

Figure 272. ISS ISP RSZ Global Image Processing Settings – Subsequence 2

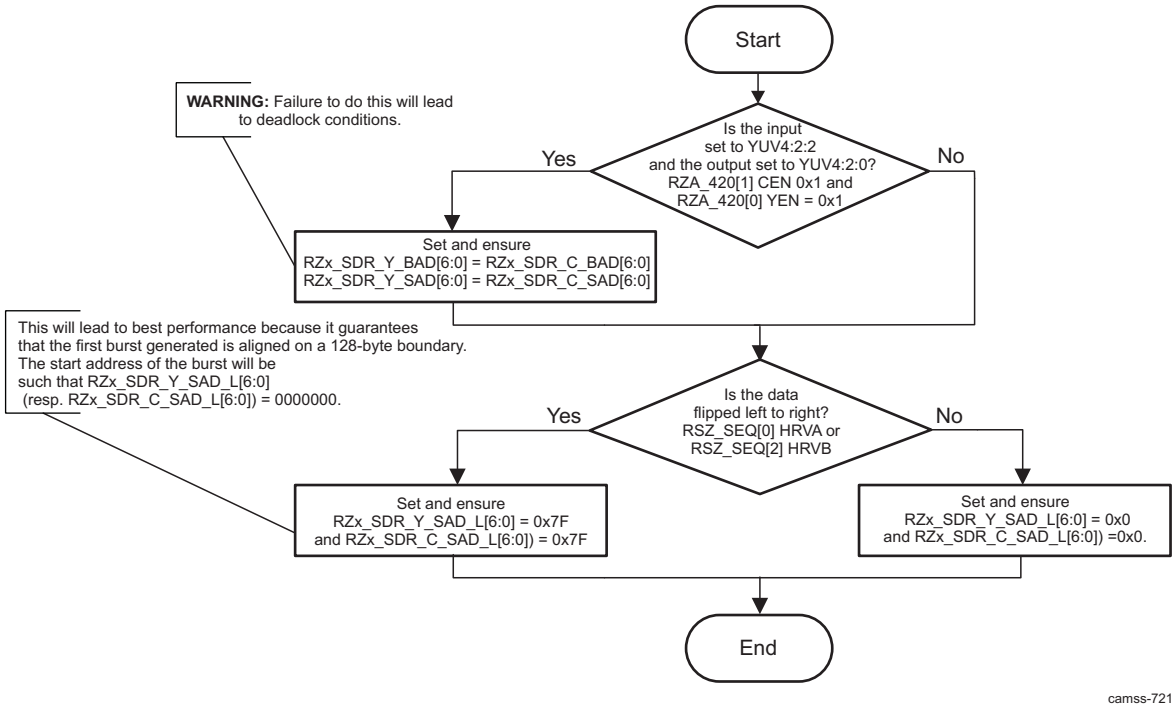


Table 247. Register Call Summary for ISS ISP RSZ Global Image Processing Settings – Subsequence 2

Register Name	Register Name	Register Name
RZA_420[1] CEN	RZA_420[0] YEN	RZB_420[1] CEN
RZB_420[0] YEN	RZA_SDR_Y_SAD_L	RZA_SDR_C_SAD_L
RZB_SDR_Y_SAD_L	RZB_SDR_C_SAD_L	RSZ_SEQ[0] HRVA
RSZ_SEQ[2] HRVB		

3.4.4.3 ISS ISP RSZ Engines Interframe Image Processing Settings

This section discusses the configuration of the RSZ interframe image processing. Table 248 identifies the setup sequence for the two different engines within the RSZ module.

Table 248. ISS ISP RSZ Engines Interframe Image Processing Settings

Step	Register/Bit Field/Programming Model	Value
IF: Is the resizer module set to free-running mode?	RSZ_SRC_MODE[0] OST	= 0x0
Set either of the engines to free-running mode to keep popping data out of the internal buffer. Otherwise, it will lead to an overflow event after the first frame.	RZA_MODE[0] MODE RZB_MODE[0] MODE	0x0
ELSE: Is the resizer module set to one-shot mode?		
Set both RSZ engines to one-shot mode.	RZA_MODE[0] MODE RZB_MODE[0] MODE	0x1
ENDIF		
Set the output format. See Table 182.	RZA_420[0] YEN RZA_420[1] CEN RZB_420[0] YEN RZB_420[1] CEN	0x-
On the side of the RSZ_SRC_VPS, RSZ_SRC_HPS, RSZ_SRC_VSZ, and RSZ_SRC_HSZ registers, set the engine complementary function to crop within the global frame and assign proper vertical start position of the input frame.	RZA_I_VPS[12:0] VPS RZB_I_VPS[12:0] VPS	0x-
Assign proper horizontal start position of the input frame.	RZA_I_HPS[12:0] HPS RZB_I_HPS[12:0] HPS	0x-
Assign proper vertical size of the output frame.	RZA_O_VSZ[12:0] VPS RZB_O_VSZ[12:0] VPS	0x-
Assign proper horizontal size of the output frame.	RZA_O_HSZ[12:1] HPS RZB_O_HSZ[12:1] HPS	0x-
Set the phase position for the Chroma and Luma element and configure the averager.	See Section 3.4.4.3.1	
Set the vertical resize value for the two engines. (vertical resize ration = 256 / RZx_V_DIF).	RZA_V_DIF[13:0] V RZB_V_DIF[13:0] V	0x-
Set the horizontal resize value for the two engines. (horizontal resize ration = 256 / RZx_H_DIF).	RZA_H_DIF[13:0] H RZB_H_DIF[13:0] H	0x-
Select vertical method of resizing filtering (linear or cubic interpolation) for the Luma and Chroma elements.	RZA_V_TYP[0] Y RZB_V_TYP[0] Y RZA_V_TYP[1] C RZB_V_TYP[1] C	0x-
Select horizontal method of resizing filtering (linear or cubic interpolation) for the Luma and Chroma elements.	RZA_H_TYP[0] Y RZB_H_TYP[0] Y RZA_H_TYP[1] C RZB_H_TYP[1] C	0x-
IF: Is the interpolation method linear?	RZx_V_TYP[0] Y **AND** RZx_V_TYP[1] C RZx_H_TYP[0] Y **AND** RZx_H_TYP[1] C	= 0x1
Set the needed vertical LPF intensity for the Luma and Chroma elements.	RZA_V_LPF[5:0] Y RZA_V_LPF[11:6] C RZB_V_LPF[5:0] Y RZB_V_LPF[11:6] C	0x-
Set the needed horizontal LPF intensity for the Luma and Chroma elements.	RZA_H_LPF[5:0] Y RZA_H_LPF[11:6] C RZB_H_LPF[5:0] Y RZB_H_LPF[11:6] C	0x-
ENDIF		
Set the Chroma saturation MAX and MIN values.	RSZ_YUV_C_MAX[7:0] MAX RSZ_YUV_C_MIN[7:0] MIN	0x-
Set the Luma saturation MAX and MIN values.	RSZ_YUV_Y_MAX[7:0] MAX RSZ_YUV_Y_MIN[7:0] MIN	0x-

Table 248. ISS ISP RSZ Engines Interframe Image Processing Settings (continued)

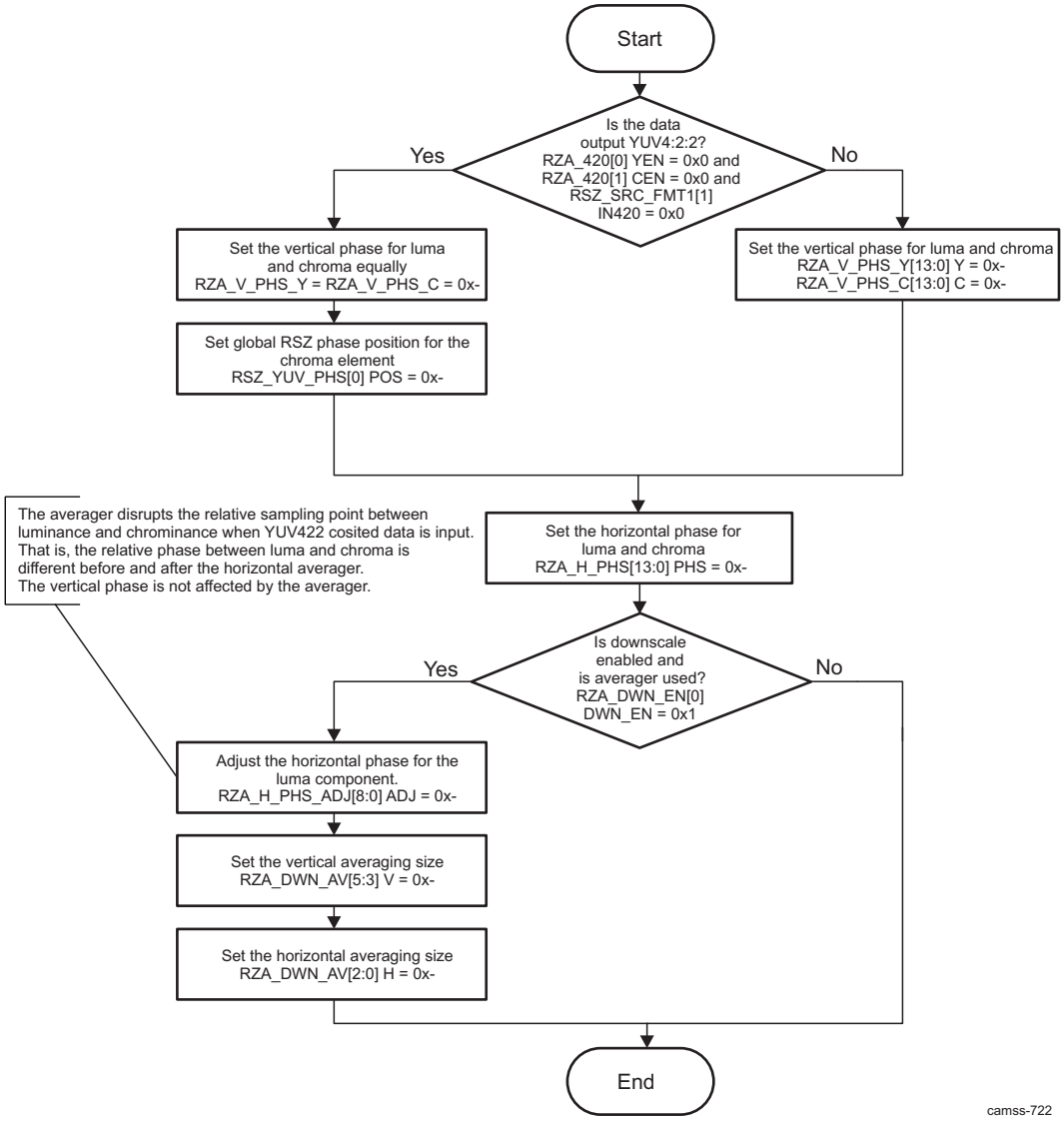
Step	Register/Bit Field/Programming Model	Value
(optional)Set the mode to RGB conversion, configure the alpha value and set additional pixel masking.	See Section 3.4.4.3.2.	

3.4.4.3.1 ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 1

The procedure shown in Figure 273 sets the phase position for the Chroma and Luma elements and configures the averager.

NOTE: This procedure configures RSZ engine A. The procedure for RSZ engine B is identical.

Figure 273. ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 1



NOTE: When masking is used, boundaries affect the leftmost/rightmost 2 pixels in up-conversion.

Table 249. Register Call Summary for ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 1

Register Name	Register Name	Register Name
RZA_420[0] YEN	RZA_420[1] CEN	RSZ_SRC_FMT1[1] IN420
RZA_V_PHS_Y	RZA_V_PHS_C	RSZ_YUV_PHS[0] POS
RZA_V_PHS_Y[13:0] Y	RZA_V_PHS_C[13:0] C	RZA_H_PHS [13:0] PHS
RZA_DWN_EN[0] DWN_EN	RZA_H_PHS_ADJ[8:0] ADJ	RZA_DWN_AV[5:3] V
RZA_DWN_AV[2:0] H		

3.4.4.3.2 ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 2

The procedure shown in Figure 274 sets the mode to RGB conversion, configures the alpha value, and sets additional pixel masking.

NOTE: This procedure configures RSZ engine A. The procedure for RSZ engine B is identical.

Figure 274. ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 2

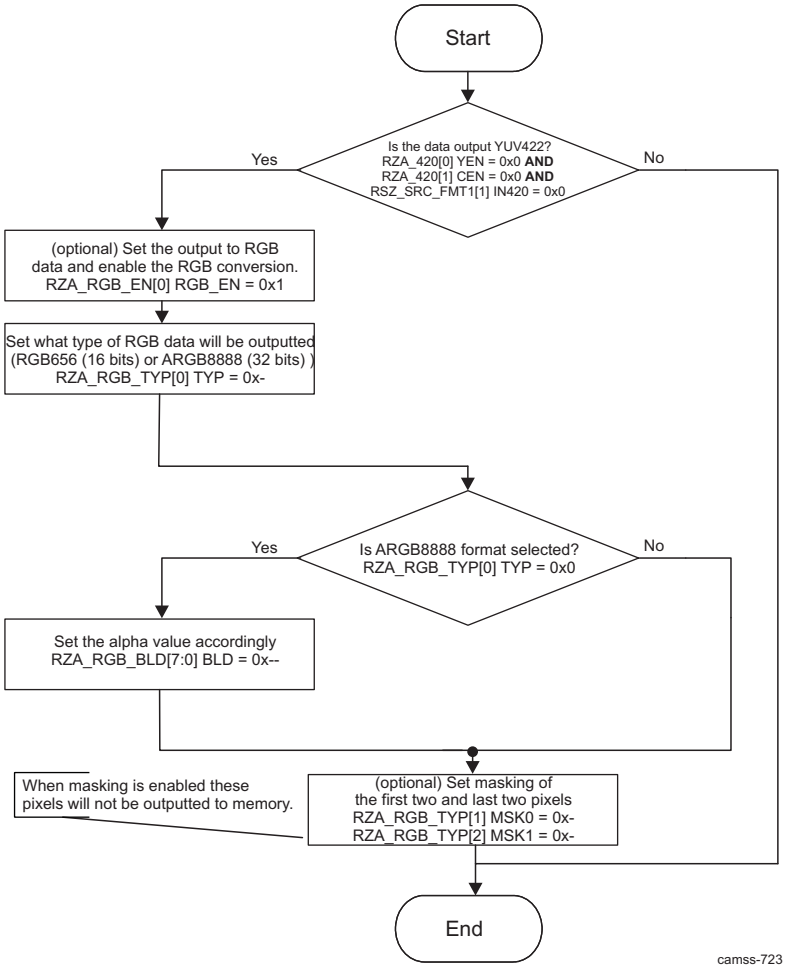


Table 250. Register Call Summary for ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 2

Register Name	Register Name	Register Name
RZA_420[0] YEN	RZA_420[1] CEN	RSZ_SRC_FMT1[1] IN420

Table 250. Register Call Summary for ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 2 (continued)

Register Name	Register Name	Register Name
RZA_RGB_EN[0] RGB_EN	RZA_RGB_TYP[0] TYP	RZA_RGB_BLD[7:0] BLD
RZA_RGB_TYP[1] MSK0	RZA_RGB_TYP[2] MSK1	

3.4.4.4 ISS ISP RSZ Programming Constraints

The RSZ module contains shadowed and nonshadowed registers. Shadowed registers can be updated anytime during the resizing operation, but the new setting does not take effect until the next rsz_int_reg event. Shadowed registers can be updated for the next frame after the rsz_int_reg event triggers.

Nonshadowed registers must be programmed before enabling the RSZ module or between frames (that is, after the rsz_int_dma and the entire frame have come on the VPORT IF, and before the VD of the next frame). Table 251 list the nonshadowed registers.

Table 251. ISS ISP RSZ Nonshadowed Registers

Nonshadowed Registers
RSZ_SYSCONFIG
RSZ_IN_FIFO_CTRL
RSZ_GCK_MMR
RSZ_GCK_SDR
RSZ_SRC_MODE
RSZ_SRC_FMT0
RSZ_SRC_VPS
RSZ_SRC_HPS
RSZ_SRC_EN

3.4.5 ISS ISP H3A Programming Model

This section discusses issues related to the software control of the H3A module. It lists which registers are required to be programmed in different modes, how to enable and disable the H3A, how to check the status of the H3A, discusses the different register access types, and enumerates several programming constraints.

3.4.5.1 ISS ISP H3A Hardware Setup/Initialization

This section discusses the configuration of the H3A required before image processing can begin.

3.4.5.1.1 ISS ISP H3A Reset Behavior

Upon hardware reset of the ISP, all the registers in the H3A are reset to their reset values.

3.4.5.1.2 ISS ISP H3A Register Setup

For register configuration purposes, the AF and AEW engines of the H3A can be configured independently. Because there are separate enable bits for each engine, this section is divided into the AF engine and the AEW engine.

3.4.5.1.2.1 ISS ISP H3A AF Engine

Before enabling the AF engine, the hardware must be properly configured through register writes. Table 252 lists the register parameters that must be programmed before enabling the AF engine of the H3A.

Table 252. ISS ISP H3A AF Engine Required Configuration Parameters

Step	Configuration Required	Value
AF optional preprocessing settings		
Note: A suggestion is to use the averaging filter in the IPIPEIF to reduce noise before generating AF statistics.		
Enable or disable the median filter.	H3A_PCR[2] AF_MED_EN	
Set the median filter threshold, if the filter is enabled.	H3A_PCR[10:3] MED_TH	
Enable or disable A-Law compression.	H3A_PCR[1] AF_ALAW_EN	
Set AF general settings.		
Set the focus value accumulation mode.	H3A_PCR[14] FVMODE	
Set the input start information.	H3A_LINE_START[31:16] SLV	
	H3A_LINE_START[15:0] LINE_START	
Set the output SDRAM destination start address.	H3A_AFBUFST[31:5] AFBUFST	
RGB pixel extraction and paxel settings		
Set the RGB positions in the pixel.	H3A_PCR[13:11] RGBPOS	
Set the paxel width.	H3A_AFPAX1[23:16] PAXW	
Set the paxel height.	H3A_AFPAX1[7:0] PAXH	
Set the paxel horizontal start position.	H3A_AFPAXSTART[27:16] PAXSH	
Set the paxel vertical start position.	H3A_AFPAXSTART[11:0] PAXSV	
Set the column increment.	H3A_AFPAX2[20:17] AFINCH	
Set the line increment.	H3A_AFPAX2[16:13] AFINCV	
Set the vertical paxel count.	H3A_AFPAX2[12:6] PAXVC	
Set the horizontal paxel count.	H3A_AFPAX2[5:0] PAXHC	
Horizontal focus value calculator settings		
Set the horizontal threshold for the two IIR.	H3A_HVF_THR[31:16] HTHR2	
	H3A_HVF_THR[15:0] HTHR1	
Set the IIR horizontal start position.	H3A_AFIIRSH[11:0] IIRSH	
Set the coefficients for the SET 0 IIR.	H3A_AFCOEF010	
	H3A_AFCOEF032	
	H3A_AFCOEF054	
	H3A_AFCOEF076	
	H3A_AFCOEF098	
	H3A_AFCOEF0010	
Set the coefficients for the SET 1 IIR.	H3A_AFCOEF110	
	H3A_AFCOEF132	
	H3A_AFCOEF154	
	H3A_AFCOEF176	
	H3A_AFCOEF198	
	H3A_AFCOEF1010	
Vertical focus value calculator settings		
Enable or disable vertical AF focus value calculation.	H3A_PCR[20] AF_VF_EN	
Set the vertical FIR 1 coefficients and threshold.	H3A_VFV_CFG1	
	H3A_VFV_CFG2	
Set the vertical FIR 2 coefficients and threshold.	H3A_VFV_CFG3	
	H3A_VFV_CFG4	

The following references offer guidelines on how to program the filter coefficients and make use of the H3A output.

- V. Peddigari, M. Gamadia, and N. Kehtarnavaz, *Real-time implementation issues in passive automatic focusing for digital still cameras*, Journal of Imaging Science and Technology, vol. 49, no. 2, pp. 114-123, Mar/Apr 2005.
- M. Gamadia and N. Kehtarnavaz, *A real-time continuous automatic focus algorithm for digital cameras*, in Proceedings of IEEE Southwest Symposium on Image Analysis and Interpretation 2006, pp. 163 - 167, Mar. 2006.
- M. Gamadia, N. Kehtarnavaz, and K. Roberts-Hoffman, *Low-light auto-focus enhancement for digital and cell-phone camera image pipelines*, IEEE Transactions on Consumer Electronics, vol. 53, no. 2, pp. 249-257, May 2007.

3.4.5.1.2.2 ISS ISP H3A AEW Engine

Before enabling the AEW engine, the hardware must be properly configured through register writes. Table 253 lists the register parameters that must be programmed before enabling the AEW engine of the H3A.

Table 253. ISS ISP H3A AEW Engine Required Configuration Parameters

Step	Configuration Required	Value
AEW optional preprocessing settings		
Enable or disable the median filter.	H3A_PCR[19] AEW_MED_EN	
Set the median filter threshold.	H3A_PCR[10:3] MED_TH	
Enable or disable A-Law compression.	H3A_PCR[17] AEW_ALAW_EN	
Set AEW general settings.		
Set the saturation limit.	H3A_PCR[31:22] AVE2LMT	
Set the AE/AWB output format.	H3A_AEWCFG[9:8] AEFMT	
Set the AE/AWB shift value for sum of pixels.	H3A_AEWCFG[3:0] SUMFST	
Set the input start information.	H3A_LINE_START[31:16] SLV H3A_LINE_START[15:0] LINE_START	
Set the output SDRAM destination start address.	H3A_AEWBUFST[31:5] AEWBUFST	
Set the AE/AWB window configuration settings.		
Set the window width (in pixels).	H3A_AEWWIN1[20:13] WINW	
Set the window height (in lines).	H3A_AEWWIN1[31:24] WINH	
Set the window count for horizontal direction.	H3A_AEWWIN1[5:0] WINHC	
Set the window count for vertical direction.	H3A_AEWWIN1[12:6] WINVC	
Set the window start position H.	H3A_AEWINSTART[11:0] WINSH	
Set the window start position V.	H3A_AEWINSTART[27:16] WINSV	
Set the horizontal distance between subsamples.	H3A_AEWSUBWIN[3:0] AEWINCH	
Set the vertical distance between subsamples.	H3A_AEWSUBWIN[11:8] AEWINCV	
Set the vertical start position for single black line of windows.	H3A_AEWINBLK[27:16] WINSV	
Set the height for the single black line of windows.	H3A_AEWINBLK[6:0] WINH	

3.4.5.2 ISS ISP H3A Enable/Disable Hardware

Setting the H3A_PCR[0] AF_EN bit enables the AF engine, and setting the H3A_PCR[16] AEW_EN bit enables the AEW engine. This is done after all of the required registers discussed in the previous section are programmed.

The H3A operates in continuous mode. Processing of the frame is dependent on the timing of the IPIPEIF. To ensure that data from the IPIPEIF is not missed, the H3A must be enabled before the IPIPEIF. In this way, the H3A waits for data from the IPIPEIF. The AF engine or the AEW engine can be disabled by clearing the H3A_PCR[0] AF_EN or H3A_PCR[16] AEW_EN bit, respectively, during the processing of the last frame. The disable is latched in at the end of the frame in which it was written.

3.4.5.3 ISS ISP H3A Register Accessibility During Frame Processing

There are two types of register access in the H3A module:

- Shadow registers

  These registers/fields can be read and written (if the field is writable) at any time. However, the written values take effect only at the start of a frame. Reads still return the most recent write even though the settings are not used until the next start of frame.

  The only shadowed registers in the H3A module are:

  – H3A_AFPAX1
  – H3A_AFPAX2
  – H3A_AFPAXSTART
  – H3A_AFIIRSH
  – H3A_AEWWIN1
  – H3A_AEWINSTART
  – H3A_AEWINBLK
  – H3A_AEWSUBWIN
  – H3A_AEWCFG
  – H3A_AEWBUFST

- Busy-lock registers

  All other registers, except those described previously, belong to this category.

  Busy-lock registers cannot be written when the module is busy (H3A_PCR[15] BUSYAF == 1 OR H3A_PCR[18] BUSYAEAWB == 1). Writes are allowed to occur, but no change occurs in the registers (blocked writes from the hardware perspective, but allowed write from the software perspective). Once the busy bit in the H3A_PCR register (H3A_PCR[15] BUSYAF or H3A_PCR[18] BUSYAEAWB bit) is reset to 0, the busy-lock registers can be written.

The ideal procedure for changing the H3A registers if (H3A_PCR[15] BUSYAF == 0 or H3A_PCR[18] BUSYAEAWB == 0) or if (EOF interrupt occurs) is:

1. Disable AF or AE/AWB.
2. Change registers.
3. Enable AF or AE/AWB.

3.4.5.4 ISS ISP H3A Interframe Operations

Between frames, it may be necessary to modify the memory pointers before processing the next frame. Because the H3A_PCR register and memory pointer registers are shadowed, these modifications can take place any time before the end of the frame, and the data will be latched in for the next frame. The host controller can perform these changes upon receiving an interrupt.

3.4.5.5 ISS ISP H3A Summary of Constraints

The following is a list of register configuration constraints to adhere to when programming the H3A. It can be used as a quick checklist. More detailed register setting constraints can be found in the individual register descriptions.

- The output addresses (H3A_AFBUFST[31:5] AFBUFST and) must be on 64-byte boundaries.
- Each horizontal row of paxels () or windows () starts on a 32-byte boundary.
- If the pixel clock frequency ISP_FCLK/2 and vertical focus is enabled, the constraints listed in Table 254 apply.

Table 254. ISS ISP H3A Constraints When PCLK ISP_FCLK/2 and Vertical Focus Enabled

Field	Constraint
H3A_AEWWIN1[20:13] WINW	7
H3A_AFPAX1[23:16] PAXW	7
H3A_AFIIRSH[11:0] IIRSH	None
H3A_AFPAXSTART[27:16] PAXSH	= H3A_AFIIRSH[11:0] IIRSH + 2
H3A_AFPAX2[20:17] AFINCH	None
H3A_AFPAX2[5:0] PAXHC	=11

- If the pixel clock frequency ISP_FCLK/2 and vertical focus is disabled, the constraints listed in Table 255 apply.

Table 255. ISS ISP H3A Constraints When PCLK ISP_FCLK/2 and Vertical Focus Disabled

Field	Constraint
H3A_AEWWIN1[20:13] WINW	7
H3A_AFPAX1[23:16] PAXW	7
H3A_AFIIRSH[11:0] IIRSH	Must be even
H3A_AFPAXSTART[27:16] PAXSH	= H3A_AFIIRSH[11:0] IIRSH + 2 and must be even
H3A_AFPAX2[20:17] AFINCH	(1+ H3A_AFPAX1[23:16] PAXW / H3A_AFPAX2[20:17] AFINCH) * H3A_AFPAX2[5:0] PAXHC must be between 4 and 384
H3A_AFPAX2[5:0] PAXHC	=35

- If the pixel clock frequency = ISP_FCLK/2 and vertical focus is enabled, the constraints listed in Table 256 apply.

Table 256. ISS ISP H3A Constraints When PCLK = ISP_FCLK/2 and Vertical Focus Enabled

Field	Constraint
H3A_AEWWIN1[20:13] WINW	15
H3A_AFPAX1[23:16] PAXW	15
H3A_AFIIRSH[11:0] IIRSH	None
H3A_AFPAXSTART[27:16] PAXSH	= H3A_AFIIRSH[11:0] IIRSH + 2
H3A_AFPAX2[20:17] AFINCH	None
H3A_AFPAX2[5:0] PAXHC	=11

- If the pixel clock frequency = ISP_FCLK/2 and vertical focus is disabled, the constraints in Table 257 apply.

Table 257. ISS ISP H3A Constraints When PCLK = ISP_FCLK/2 and Vertical Focus Disabled

Field	Constraint
H3A_AEWWIN1[20:13] WINW	15
H3A_AFPAX1[23:16] PAXW	15
H3A_AFIIRSH[11:0] IIRSH	Must be even
H3A_AFPAXSTART[27:16] PAXSH	= H3A_AFIIRSH[11:0] IIRSH + 2 and must be even
H3A_AFPAX2[20:17] AFINCH	(1+ H3A_AFPAX1[23:16] PAXW / H3A_AFPAX2[20:17] AFINCH) * H3A_AFPAX2[5:0] PAXHC must be between 4 and 384 and H3A_AFPAX2[20:17] AFINCH modulo H3A_AFPAX1[23:16] PAXW !=1

Table 257. ISS ISP H3A Constraints When PCLK = ISP_FCLK/2 and Vertical Focus Disabled (continued)

Field	Constraint
H3A_AFPAX2[5:0] PAXHC	=35

AF engine

- The paxel horizontal start value must be greater than or equal to the IIR horizontal start position.
- The paxel start/end and IIR filter start positions must not be set within the first 2 and the last 2 pixels (to check).
- The width (H3A_AFPAX1[23:16] PAXW) and height (H3A_AFPAX1[7:0] PAXH) of the paxels must be an even number.
- The minimum width of the paxel (H3A_AFPAX1[23:16] PAXW) must be 8 pixels.
- The number of columns to increment in a paxel (H3A_AFPAX2[20:17] AFINCH) must be even and is restricted to 2 to 32.
- The number of lines to increment in a paxel (H3A_AFPAX2[16:13] AFINCV) must be even and is restricted to 0 to 30.
- The maximum number of vertical paxels in a frame (H3A_AFPAX2[12:6] PAXVC) must not exceed 128.
- The number of paxels in the horizontal direction (H3A_AFPAX2[5:0] PAXHC) has a valid range from 1 to 35.
- If vertical mode is enabled:
  - The paxel horizontal start position (H3A_AFPAXSTART[27:16] PAXSH) must be even.
  - The lower bit of the H3A_AFPAXSTART[27:16] PAXSH bit field and the lower bit of the H3A_AFIIRSH[11:0 IIRSH] bit field must be equal.
  - The H3A_AFPAXSTART[11:0] PAXSV bit field must be = 8.
- If vertical mode is not enabled, the H3A_AFIIRSH[11:0] IIRSH bit field must be even.
- Paxels cannot overlap the last pixel in a line.
- Paxels must be adjacent to one another.

AEW engine

- The width (H3A_AEWWIN1[20:13] WINW) and height (H3A_AEWWIN1[31:24] WINH) of the windows must be an even number.
- The minimum width of the window (H3A_AEWWIN1[20:13] WINW) must be 8 pixels.
- The window height (H3A_AEWWIN1[31:24] WINH) has a valid range from 2 to 512.
- The maximum number of vertical windows in a frame (H3A_AEWWIN1[12:6] WINVC) must not exceed 128.
- The number of horizontal windows (H3A_AEWWIN1[5:0] WINHC) has a valid range from 1 to 35.
- The vertical and horizontal window start position (H3A_AEWWINSTART) has a valid range from 0 to 4095.
- The vertical window start position for single black lines (H3A_AEWINBLK[27:16] WINSV) has a valid range from 0 to 4095.
- The horizontal window start position for single black lines (H3A_AEWINBLK[6:0] WINH) must be even and has a valid range from 2 to 256
- The subsampling windows can start only on even numbers.
- The vertical and horizontal sampling point increment (H3A_AEWSUBWIN) has a valid range from 2 to 32.

3.4.6 ISS ISP BL Programming Model

The procedure listed in Table 258 initializes the buffer logic.

Table 258. ISS ISP BL Settings

Step	Register/Bit Field/Programming Model	Value
Sets the memory access priority registers	ISP5_MPSR	
Sets the minimum interval between two memory requests for ISIF read port	ISP5_BL_MTC_1[31:16] ISIF_R	
Sets the minimum interval between two memory requests for IPIPEIF read port	ISP5_BL_MTC_1[15:0] IPIPEIF_R	
Sets the minimum interval between two memory requests for H3A write port	ISP5_BL_MTC_1[31:16] H3A_W	
Set the maximum number of CIDs/tags that the BL can use.	ISP5_CTRL[7:4] VBUSM_CIDS	
Set the BL VBUSM priority setting.	ISP5_CTRL[3:1] VBUSM_CPRIORITY	
Set write posted/nonposted.	ISP5_CTRL[0] OCP_WRNP	
Enable the BL clock.	ISP5_CTRL[15] BL_CLK_ENABLE	0x1

3.5 ISS ISP Registers

Table 259 lists the ISS ISP instances.

Table 259. ISS ISP Instance Summary

Module Name	Base Address Cortex-M3 Private Access	Base Address L3 Interconnect	Size
ISS_ISP5_SYS1	0x5505 0000	0x5C01 0000	160 bytes
ISS_ISP5_SYS2	0x5505 00A0	0x5C01 00A0	864 bytes
ISS_RESIZER	0x5505 0400	0x5C01 0400	1KB
ISS_IPIPE	0x5505 0800	0x5C01 0800	2KB
ISS_ISIF	0x5505 1000	0x5C01 1000	512 bytes
ISS_IPIPEIF	0x5505 1200	0x5C01 1200	128 bytes
ISS_H3A	0x5505 1400	0x5C01 1400	512 bytes

3.5.1 ISS ISP5 SYS1 Registers

Table 260 summarizes the ISS ISP5 SYS1 registers.

Table 260. ISS ISP5 SYS1 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ISS_ISP5_SYS1 Base Address Cortex-M3 Private Access	ISS_ISP5_SYS1 Base Address L3 Interconnect
ISP5_REVISION	R	32	0x0000 0000	0x5505 0000	0x5C01 0000
ISP5_HWINFO1	R	32	0x0000 0004	0x5505 0004	0x5C01 0004
ISP5_HWINFO2	R	32	0x0000 0008	0x5505 0008	0x5C01 0008
ISP5_SYSCONFIG	RW	32	0x0000 0010	0x5505 0010	0x5C01 0010
RESERVED	RW	32	0x0000 0020	0x5505 0020	0x5C01 0020
ISP5_IRQSTATUS_RAW_i (1)	RW	32	0x0000 0024 + (0x10 * i)	0x5505 0024 + (0x10 * i)	0x5C01 0024 + (0x10 * i)
ISP5_IRQSTATUS_i (1)	RW	32	0x0000 0028 + (0x10 * i)	0x5505 0028 + (0x10 * i)	0x5C01 0028 + (0x10 * i)
ISP5_IRQENABLE_SET_i (1)	RW	32	0x0000 002C + (0x10 * i)	0x5505 002C + (0x10 * i)	0x5C01 002C + (0x10 * i)
ISP5_IRQENABLE_CLR_i (1)	RW	32	0x0000 0030 + (0x10 * i)	0x5505 0030 + (0x10 * i)	0x5C01 0030 + (0x10 * i)
ISP5_DMAENABLE_SET	RW	32	0x0000 0064	0x5505 0064	0x5C01 0064
ISP5_DMAENABLE_CLR	RW	32	0x0000 0068	0x5505 0068	0x5C01 0068
ISP5_CTRL	RW	32	0x0000 006C	0x5505 006C	0x5C01 006C
RESERVED	R	32	0x0000 0070	0x5505 0070	0x5C01 0070
RESERVED	R	32	0x0000 0074	0x5505 0074	0x5C01 0074
RESERVED	R	32	0x0000 0078	0x5505 0078	0x5C01 0078
ISP5_MPSR	RW	32	0x0000 007C	0x5505 007C	0x5C01 007C
ISP5_BL_MTC_1	RW	32	0x0000 0080	0x5505 0080	0x5C01 0080
ISP5_BL_MTC_2	RW	32	0x0000 0084	0x5505 0084	0x5C01 0084
ISP5_BL_VBUSM	RW	32	0x0000 0088	0x5505 0088	0x5C01 0088

(1) i = 0 to 3

3.5.1.1 ISP5_REVISION

Table 261. ISP5_REVISION

Address Offset	0x0000 0000		
Physical Address	0x5505 0000 0x5C01 0000	**Instance**	ISS_ISP5_SYS1_CORTEX-M3 ISS_ISP5_SYS1_L3
Description	IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	See (1)

(1) TI internal data

3.5.1.2 ISP5_HWINFO1

Table 262. ISP5_HWINFO1

Address Offset	0x0000 0004		
Physical Address	0x5505 0004 0x5C01 0004	**Instance**	ISS_ISP5_SYS1_CORTEX-M3 ISS_ISP5_SYS1_L3
Description	GENERIC PARAMETER REGISTER Information about the hardware configuration of the IP module.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ISIF_RFM_LINE_SIZE								RESERVED			IPIPE_LINE_SIZE												

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	ISIF_RFM_LINE_SIZE	Memory line size for the data reformatter in the ISIF module.	R	0x1500
15:13	RESERVED		R	0x0
12:0	IPIPE_LINE_SIZE	Memory line size for the IPIPE module	R	0x1500

3.5.1.3 ISP5_HWINFO2

Table 263. ISP5_HWINFO2

Address Offset	0x0000 0008		
Physical Address	0x5505 0008 0x5C01 0008	**Instance**	ISS_ISP5_SYS1_CORTEX-M3 ISS_ISP5_SYS1_L3
Description	GENERIC PARAMETER REGISTER Information about the hardware configuration of the IP module.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			H3A_LINE_SIZE												

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	H3A_LINE_SIZE	Memory line size for the H3A module	R	0x0BC0

3.5.1.4 ISP5_SYSCONFIG

Table 264. ISP5_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x5505 0010 0x5C01 0010	**Instance**	ISS_ISP5_SYS1_CORTEX-M3 ISS_ISP5_SYS1_L3
Description	Clock management configuration		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										STANDBYMODE		RESERVED		SOFTRESET	AUTO_IDLE

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0000000
5:4	STANDBYMODE	Configuration of the local initiator state management mode. By definition, initiator may generate read/write transaction as long as it is out of STANDBY state. 0x0: Force-standby mode: local initiator is unconditionally placed in standby state. Backup mode, for debug only. 0x1: No-standby mode: local initiator is unconditionally placed out of standby state. Backup mode, for debug only. 0x2: Smart-standby mode: local initiator standby status depends on local conditions, that is, the module's functional requirement from the initiator. IP module must not generate (initiator-related) wake-up events. Generation of the MStandby signal must be initiated by the firmware by writing ISP5_CTRL.MSTANDBY = 1. 0x3: Reserved	RW	0x2
3:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1	SOFTRESET	Software reset. The soft reset will cause the MStandby to be asserted as the reset value of the ISP5_CTRL.MSTANDBY bit is 1. After a soft reset, the software must ensure not to perform any access for 16 clock cycles (OCP-slave port frequency) after writing this bit. The OCP slave port is running at half the frequency of the functional clock. Before issuing a soft reset, the software must ensure that no more traffic is being generated by the ISP. Basically, it means that the camera module must be stopped from sending data and/or that the ISP modules are disabled. The last interrupt triggered by the ISP design upon completion of the frame processing is rsz_int_dma. This rsz_int_dma event must be used to enable clean termination of the processing. The software must wait a few hundred cycles to trigger the soft reset after upon assertion of the rsz_int_dma, this is to ensure that the BL is completely drained. Software must set the ISP in standby mode before issuing the soft reset: Set ISP5_SYSCONFIG.STANDBYMODE = 2 (smart standby). Set ISP5_CTRL.MSTANDBY to 1. Poll for ISP5_CTRL.MSTANDBY_WAIT = 1. Then, the soft reset can be applied (ISP5_SYSCONFIG.SOFTRESET = 1). Write 0x0: No action Write 0x1: Initiate software reset Read 0x1: Reset (software or other) ongoing Read 0x0: Reset done, no pending action	RW	0
0	AUTO_IDLE	Auto clock gating. Always enabled.	R	1

3.5.1.5 ISP5_IRQSTATUS_RAW_i

Table 265. ISP5_IRQSTATUS_RAW_i

Address Offset	0x0000 0024 + (0x10 * i)	**Index**	i = 0 to 3
Physical Address	0x5505 0024 + (0x10 * i) 0x5C01 0024 + (0x10 * i)	**Instance**	ISS_ISP5_SYS1_CORTEX-M3 ISS_ISP5_SYS1_L3
Description	Per-event raw interrupt status vector. Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug. The ISP outputs four interrupt lines, ISP_IRQ0 to ISP_IRQ3. Any internal ISP event can be merged on the four lines. A same event must be enabled on only one interrupt line.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCP_ERR_IRQ	RESERVED	IPIPE_INT_DPC_RNEW1	IPIPE_INT_DPC_RNEW0	IPIPE_INT_DPC_INI	RESERVED	IPIPE_INT_EOF	H3A_INT_EOF	RSZ_INT_EOF1	RSZ_INT_EOF0	RESERVED		RSZ_FIFO_IN_BLK_ERR	RSZ_FIFO_OVF	RSZ_INT_CYC_RZB	RSZ_INT_CYC_RZA	RSZ_INT_DMA	RSZ_INT_LAST_PIX	RSZ_INT_REG	H3A_INT	AF_INT	AEW_INT	IPIPEIF_IRQ	IPIPE_INT_HST	IPIPE_INT_BSC	IPIPE_INT_DMA	IPIPE_INT_LAST_PIX	IPIPE_INT_REG	ISIF_INT_3	ISIF_INT_2	ISIF_INT_1	ISIF_INT_0

Bits	Field Name	Description	Type	Reset
31	OCP_ERR_IRQ	An OCP error has been received on the ISP master port. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
30	RESERVED		R	0
29	IPIPE_INT_DPC_RNEW1	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
28	IPIPE_INT_DPC_RNEW0	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
27	IPIPE_INT_DPC_INI	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
26	RESERVED		R	0
25	IPIPE_INT_EOF	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
24	H3A_INT_EOF	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
23	RSZ_INT_EOF1	RESIZER module event: This event signals that the BL has received the EOF signal from the resizer B engine which happens one the last transfer in the frame has happened. Note that because the BL has FIFOs it may take some additional time before true data transfer completion to memory. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
22	RSZ_INT_EOF0	RESIZER module event: This event signals that the BL has received the EOF signal from the resizer A engine which happens one the last transfer in the frame has happened. Note that because the BL has FIFOs it may take some additional time before true data transfer completion to memory. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
21:20	RESERVED		R	0x0
19	RSZ_FIFO_IN_BLK_ERR	This event signals that the minimum vertical blanking period has not been respected causing errors in the input data buffering submodule. This event will be triggered when the rsz_int_reg event of frame N is triggered before the rsz_int_dma of frame N + 1. This event would typically happen at the transition between two frames because there is not enough vertical blanking between frames: the firmware must take care to ensure enough vertical blanking. The hardware cannot recover from this error. It will be required to perform a reset of the IP. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
18	RSZ_FIFO_OVF	This event signals that overflow happened in the input data buffering submodule. This event would typically happen while processing a frame because the video port pixel clock is too high: the firmware must take care to use a lower pixel clock at the input of the resizer module. Depending on the mode being used, the overflow can happen at different places: 1. Bypass mode: overflow happened in the input circular buffer. 2. Pass through mode: overflow happened on the module output interface (MTC) 3. Normal resize mode: overflow happened in the input circular buffer. The hardware cannot recover from this error. It will be required to perform a reset of the IP. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
17	RSZ_INT_CYC_RZB	RESIZER module event: This event is the circular interrupt for RESIZER #B. An event can be triggered every time that RSZ_IRQ_RZB output lines have been written out to the RZB_SDR_Y buffer. The range can go from 1 to 8192 lines. Usually, this value should be such that the circular buffer vertical size (set by the RZB_SDR_Y_PTR_E register) is a multiple of RSZ_IRQ_RZB. Note that at the time the interrupt is triggered, the actual data write has not taken place. It may take a few hundred of cycles to complete the data write into system memory. This is not an issue since the start of the buffer is read first. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
16	RSZ_INT_CYC_RZA	RESIZER module event: This event is the circular interrupt for RESIZER #A. An event can be triggered every time that RSZ_IRQ_RZA output lines have been written out to the RZA_SDR_Y buffer. The range can go from 1 to 8192 lines. Usually, the circular buffer vertical size (set by the RZA_SDR_Y_PTR_E register) should be a multiple of RSZ_IRQ_RZA. Note that at the time the interrupt is triggered, the actual data write has not taken place. It may take a few hundred of cycles to complete the data write into system memory. This is not an issue since the start of the buffer is read first. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
15	RSZ_INT_DMA	This event is triggered when the last EOF (of the two MTC interfaces) is sent out to the BL and that the resizer core has returned to idle. rsz_int_dma is a true indication that all processing is finished for the particular frame on both resizer engines. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
14	RSZ_INT_LAST_PIX	This event is triggered when the last pixel of the valid area is received. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
13	RSZ_INT_REG	This event is triggered when the first pixel of the valid area is received. Shadowed registers can be updated at any time but the new value will take effect on the next rsz_int_reg event. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
12	H3A_INT	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
11	AF_INT	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
10	AEW_INT	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
9	IPIPEIF_IRQ	IPIPEIF module interrupt Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
8	IPIPE_INT_HST	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
7	IPIPE_INT_BSC	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
6	IPIPE_INT_DMA	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
5	IPIPE_INT_LAST_PIX	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
4	IPIPE_INT_REG	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
3	ISIF_INT_3	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
2	ISIF_INT_2	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
1	ISIF_INT_1	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
0	ISIF_INT_0	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

3.5.1.6 ISP5_IRQSTATUS_i

Table 266. ISP5_IRQSTATUS_i

Address Offset	0x0000 0028 + (0x10 * i)	**Index**	i = 0 to 3
Physical Address	0x5505 0028 + (0x10 * i) 0x5C01 0028 + (0x10 * i)	**Instance**	ISS_ISP5_SYS1_CORTEX-M3 ISS_ISP5_SYS1_L3
Description	Per-event "enabled" interrupt status vector. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled). The ISP outputs four interrupt lines, ISP_IRQ0 to ISP_IRQ3. Any internal ISP event can be merged on the four lines. A same event must be enabled on only one interrupt line.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCP_ERR_IRQ	RESERVED	IPIPE_INT_DPC_RNEW1	IPIPE_INT_DPC_RNEW0	IPIPE_INT_DPC_INI	RESERVED	IPIPE_INT_EOF	H3A_INT_EOF	RSZ_INT_EOF1	RSZ_INT_EOF0	RESERVED		RSZ_FIFO_IN_BLK_ERR	RSZ_FIFO_OVF	RSZ_INT_CYC_RZB	RSZ_INT_CYC_RZA	RSZ_INT_DMA	RSZ_INT_LAST_PIX	RSZ_INT_REG	H3A_INT	AF_INT	AEW_INT	IPIPEIF_IRQ	IPIPE_INT_HST	IPIPE_INT_BSC	IPIPE_INT_DMA	IPIPE_INT_LAST_PIX	IPIPE_INT_REG	ISIF_INT_3	ISIF_INT_2	ISIF_INT_1	ISIF_INT_0

Bits	Field Name	Description	Type	Reset
31	OCP_ERR_IRQ	An OCP error has been received on the ISP master port. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
30	RESERVED		R	0
29	IPIPE_INT_DPC_RNEW1	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
28	IPIPE_INT_DPC_RNEW0	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
27	IPIPE_INT_DPC_INI	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
26	RESERVED		R	0
25	IPIPE_INT_EOF	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
24	H3A_INT_EOF	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
23	RSZ_INT_EOF1	RESIZER module event: This event signals that the BL has received the EOF signal from the resizer B engine which happens one the last transfer in the frame has happened. Note that because the BL has FIFOs it may take some additional time before true data transfer completion to memory. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
22	RSZ_INT_EOF0	RESIZER module event: This event signals that the BL has received the EOF signal from the resizer A engine which happens one the last transfer in the frame has happened. Note that because the BL has FIFOs it may take some additional time before true data transfer completion to memory. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
21:20	RESERVED		R	0x0
19	RSZ_FIFO_IN_BLK_ERR	This event signals that the minimum vertical blanking period has not been respected causing errors in the input data buffering submodule. This event will be triggered when the rsz_int_reg event of frame N is triggered before the rsz_int_dma of frame N + 1. This event would typically happen at the transition between two frames because there is not enough vertical blanking between frames: the firmware must take care to ensure enough vertical blanking. The hardware cannot recover from this error. It will be required to perform a reset of the IP. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
18	RSZ_FIFO_OVF	This event signals that overflow happened in the input data buffering submodule or in the RSZ output interface. This event would typically happen while processing a frame because the video port pixel clock is too high: the firmware must take care to use a lower pixel clock at the input of the resizer module. The hardware cannot recover from this error. It will be required to perform a reset of the IP. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
17	RSZ_INT_CYC_RZB	RESIZER module event: This event is the circular interrupt for RESIZER #B. An event can be triggered every time that RSZ_IRQ_RZB output lines have been written out to the RZB_SDR_Y buffer. The range can go from 1 to 8192 lines. Usually, this value should be such that the circular buffer vertical size (set by the RZB_SDR_Y_PTR_E register) is a multiple of RSZ_IRQ_RZB. Note that at the time the interrupt is triggered, the actual data write has not taken place. It may take a few hundred of cycles to complete the data write into system memory. This is not an issue since the start of the buffer is read first. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
16	RSZ_INT_CYC_RZA	RESIZER module event: This event is the circular interrupt for RESIZER #A. An event can be triggered every time that RSZ_IRQ_RZA output lines have been written out to the RZA_SDR_Y buffer. The range can go from 1 to 8192 lines. Usually, the circular buffer vertical size (set by the RZA_SDR_Y_PTR_E register) should be a multiple of RSZ_IRQ_RZA. Note that at the time the interrupt is triggered, the actual data write has not taken place. It may take a few hundred of cycles to complete the data write into system memory. This is not an issue since the start of the buffer is read first. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
15	RSZ_INT_DMA	This event is triggered when the last EOF (of the two MTC interfaces) is sent out to the BL and that the resizer core has returned to idle. rsz_int_dma is a true indication that all processing is finished for the particular frame on both resizer engines. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
14	RSZ_INT_LAST_PIX	This event is triggered when the last pixel of the valid area is received. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
13	RSZ_INT_REG	This event is triggered when the first pixel of the valid area is received. Shadowed registers can be updated at any time but the new value will take effect on the next rsz_int_reg event. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
12	H3A_INT	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
11	AF_INT	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
10	AEW_INT	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
9	IPIPEIF_IRQ	IPIPEIF module interrupt Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
8	IPIPE_INT_HST	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
7	IPIPE_INT_BSC	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
6	IPIPE_INT_DMA	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
5	IPIPE_INT_LAST_PIX	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
4	IPIPE_INT_REG	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
3	ISIF_INT_3	LSC interrupt issued by 2D-LSC block. Four types of 2D-LSC can be generated and mapped to the INT_3 line. For more information, see Section 3.3.6.10.1.5, *ISS ISP ISIF 2D-LSC Events and Status Checking.* Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
2	ISIF_INT_2	VD interrupt 2 event. Read this bit to check the interrupt mapped to the INT_2 line. This interrupt is also set based on the VD pulse position after receiving a configured number of horizontal pulse signals. For more information, see Section 3.3.6.19.1, *ISS ISP ISIF VDINT0, VDINT1 and VDINT2 Interrupts.* Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
1	ISIF_INT_1	VD interrupt 1 event. Read this bit to check the interrupt mapped to the INT_1 line. This interrupt is also set based on the VD pulse position after receiving a configured number of horizontal pulse signals. For more information, see Section 3.3.6.19.1, *ISS ISP ISIF VDINT0, VDINT1 and VDINT2 Interrupts.* Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
0	ISIF_INT_0	VD interrupt 0 event. Read this bit to check the interrupt mapped to the INT_0 line. This interrupt is also set based on the VD pulse position after receiving a configured number of horizontal pulse signals. For more information, see Section 3.3.6.19.1, *ISS ISP ISIF VDINT0, VDINT1 and VDINT2 Interrupts.* Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

NOTE: ISP submodule interrupts are mapped to ISP top-level lines. Moreover, ISP top-level lines are mapped to the ISS top interrupt merger level. For information about how IRQ lines are handled at the ISP level, see Table 166. For information about how IRQs are handled at the ISS top-level before the signals leave ISS boundaries, see , *ISS Interrupt Merger.*

3.5.1.7 ISP5_IRQENABLE_SET_i

Table 267. ISP5_IRQENABLE_SET_i

Address Offset	0x0000 002C + (0x10 * i)	**Index**	i = 0 to 3
Physical Address	0x5505 002C + (0x10 * i) 0x5C01 002C + (0x10 * i)	**Instance**	ISS_ISP5_SYS1_CORTEX-M3 ISS_ISP5_SYS1_L3
Description	Per-event interrupt enable bit vector. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register. The ISP outputs four interrupt lines, ISP_IRQ0 to ISP_IRQ3. Any internal ISP event can be merged on the four lines. A same event must be enabled on only one interrupt line.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCP_ERR_IRQ	RESERVED	IPIPE_INT_DPC_RNEW1	IPIPE_INT_DPC_RNEW0	IPIPE_INT_DPC_INI	RESERVED	IPIPE_INT_EOF	H3A_INT_EOF	RSZ_INT_EOF1	RSZ_INT_EOF0	RESERVED		RSZ_FIFO_IN_BLK_ERR	RSZ_FIFO_OVF	RSZ_INT_CYC_RZB	RSZ_INT_CYC_RZA	RSZ_INT_DMA	RSZ_INT_LAST_PIX	RSZ_INT_REG	H3A_INT	AF_INT	AEW_INT	IPIPEIF_IRQ	IPIPE_INT_HST	IPIPE_INT_BSC	IPIPE_INT_DMA	IPIPE_INT_LAST_PIX	IPIPE_INT_REG	ISIF_INT_3	ISIF_INT_2	ISIF_INT_1	ISIF_INT_0

Bits	Field Name	Description	Type	Reset
31	OCP_ERR_IRQ	An OCP error has been received on the ISP master port. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
30	RESERVED		R	0
29	IPIPE_INT_DPC_RNEW1	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
28	IPIPE_INT_DPC_RNEW0	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
27	IPIPE_INT_DPC_INI	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
26	RESERVED		R	0
25	IPIPE_INT_EOF	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
24	H3A_INT_EOF	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
23	RSZ_INT_EOF1	RESIZER module event: This event signals that the BL has received the EOF signal from the resizer B engine which happens one the last transfer in the frame has happened. Note that because the BL has FIFOs it may take some additional time before true data transfer completion to memory. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
22	RSZ_INT_EOF0	RESIZER module event: This event signals that the BL has received the EOF signal from the resizer A engine which happens one the last transfer in the frame has happened. Note that because the BL has FIFOs it may take some additional time before true data transfer completion to memory. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
21:20	RESERVED		R	0x0
19	RSZ_FIFO_IN_BLK_ERR	This event signals that the minimum vertical blanking period has not been respected causing errors in the input data buffering submodule. This event will be triggered when the rsz_int_reg event of frame N is triggered before the rsz_int_dma of frame N + 1. This event would typically happen at the transition between two frames because there is not enough vertical blanking between frames: the firmware must take care to ensure enough vertical blanking. The hardware cannot recover from this error. It will be required to perform a reset of the IP. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
18	RSZ_FIFO_OVF	This event signals that overflow happened in the input data buffering submodule or in the RSZ output interface. This event would typically happen while processing a frame because the video port pixel clock is too high: the firmware must take care to use a lower pixel clock at the input of the resizer module. The hardware cannot recover from this error. It will be required to perform a reset of the IP. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
17	RSZ_INT_CYC_RZB	RESIZER module event: This event is the circular interrupt for RESIZER #B. An event can be triggered every time that RSZ_IRQ_RZB output lines have been written out to the RZB_SDR_Y buffer. The range can go from 1 to 8192 lines. Usually, this value should be such that the circular buffer vertical size (set by the RZB_SDR_Y_PTR_E register) is a multiple of RSZ_IRQ_RZB. Note that at the time the interrupt is triggered, the actual data write has not taken place. It may take a few hundred of cycles to complete the data write into system memory. This is not an issue since the start of the buffer is read first. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
16	RSZ_INT_CYC_RZA	RESIZER module event: This event is the circular interrupt for RESIZER #A. An event can be triggered every time that RSZ_IRQ_RZA output lines have been written out to the RZA_SDR_Y buffer. The range can go from 1 to 8192 lines. Usually, the circular buffer vertical size (set by the RZA_SDR_Y_PTR_E register) should be a multiple of RSZ_IRQ_RZA. Note that at the time the interrupt is triggered, the actual data write has not taken place. It may take a few hundred of cycles to complete the data write into system memory. This is not an issue since the start of the buffer is read first. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
15	RSZ_INT_DMA	This event is triggered when the last EOF (of the two MTC interfaces) is sent out to the BL and that the resizer core has returned to idle. rsz_int_dma is a true indication that all processing is finished for the particular frame on both resizer engines. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
14	RSZ_INT_LAST_PIX	This event is triggered when the last pixel of the valid area is received. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
13	RSZ_INT_REG	This event is triggered when the first pixel of the valid area is received. Shadowed registers can be updated at any time but the new value will take effect on the next rsz_int_reg event. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
12	H3A_INT	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
11	AF_INT	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
10	AEW_INT	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
9	IPIPEIF_IRQ	IPIPEIF module interrupt Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
8	IPIPE_INT_HST	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
7	IPIPE_INT_BSC	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
6	IPIPE_INT_DMA	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
5	IPIPE_INT_LAST_PIX	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
4	IPIPE_INT_REG	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
3	ISIF_INT_3	LSC interrupt issued by the 2D-LSC block. Four types of 2D-LSC can be generated and mapped to the INT_3 line. For more information, see Section 3.3.6.10.1.5, *ISS ISP ISIF 2D-LSC Events and Status Checking.* Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
2	ISIF_INT_2	VD interrupt 2 event. Set this bit to enable the interrupt and map it to the INT_2 line. This interrupt is set based on the VD pulse position after receiving a configured number of horizontal pulse signals. For more information, see Section 3.3.6.19.1, *ISS ISP ISIF VDINT0, VDINT1 and VDINT2 Interrupts.* Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
1	ISIF_INT_1	VD interrupt 1 event. Set this bit to enable the interrupt and map it to the INT_1 line. This interrupt is set based on the VD pulse position after receiving a configured number of horizontal pulse signals. For more information, see Section 3.3.6.19.1, *ISS ISP ISIF VDINT0, VDINT1 and VDINT2 Interrupts.* Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
0	ISIF_INT_0	VD interrupt 0 event. Set this bit to enable the interrupt mapped to INT_0 line. This interrupt is set based on the VD pulse position after receiving a configured number of horizontal pulse signals. For more information, see Section 3.3.6.19.1, *ISS ISP ISIF VDINT0, VDINT1 and VDINT2 Interrupts.* Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

NOTE: Setting the ISP submodule interrupts and mapping them to the ISP lines requires a configuration to receive the IRQ events at a higher ISS interrupt merger level. For information about how IRQ lines are handled at the ISP level, see Table 166. For information about how IRQs are handled at the ISS top-level before the signals leave ISS boundaries, see , *ISS Interrupt Merger.*

3.5.1.8 ISP5_IRQENABLE_CLR_i

Table 268. ISP5_IRQENABLE_CLR_i

Address Offset	0x0000 0030 + (0x10 * i)	**Index**	i = 0 to 3
Physical Address	0x5505 0030 + (0x10 * i) 0x5C01 0030 + (0x10 * i)	**Instance**	ISS_ISP5_SYS1_CORTEX-M3 ISS_ISP5_SYS1_L3
Description	Per-event interrupt enable bit vector. Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register. The ISP outputs four interrupt lines, ISP_IRQ0 to ISP_IRQ3. Any internal ISP event can be merged on the four lines. A same event must be enabled on only one interrupt line.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCP_ERR_IRQ	RESERVED	IPIPE_INT_DPC_RNEW1	IPIPE_INT_DPC_RNEW0	IPIPE_INT_DPC_INI	RESERVED	IPIPE_INT_EOF	H3A_INT_EOF	RSZ_INT_EOF1	RSZ_INT_EOF0	RESERVED		RSZ_FIFO_IN_BLK_ERR	RSZ_FIFO_OVF	RSZ_INT_CYC_RZB	RSZ_INT_CYC_RZA	RSZ_INT_DMA	RSZ_INT_LAST_PIX	RSZ_INT_REG	H3A_INT	AF_INT	AEW_INT	IPIPEIF_IRQ	IPIPE_INT_HST	IPIPE_INT_BSC	IPIPE_INT_DMA	IPIPE_INT_LAST_PIX	IPIPE_INT_REG	ISIF_INT_3	ISIF_INT_2	ISIF_INT_1	ISIF_INT_0

Bits	Field Name	Description	Type	Reset
31	OCP_ERR_IRQ	An OCP error has been received on the ISP master port. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
30	RESERVED		R	0
29	IPIPE_INT_DPC_RNEW1	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
28	IPIPE_INT_DPC_RNEW0	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
27	IPIPE_INT_DPC_INI	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
26	RESERVED		R	0
25	IPIPE_INT_EOF	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW	0

Bits	Field Name	Description	Type	Reset
24	H3A_INT_EOF	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW	0
23	RSZ_INT_EOF1	RESIZER module event: This event signals that the BL has received the EOF signal from the resizer B engine which happens one the last transfer in the frame has happened. Note that because the BL has FIFOs it may take some additional time before true data transfer completion to memory. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW	0
22	RSZ_INT_EOF0	RESIZER module event: This event signals that the BL has received the EOF signal from the resizer A engine which happens one the last transfer in the frame has happened. Note that because the BL has FIFOs it may take some additional time before true data transfer completion to memory. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW	0
21:20	RESERVED		R	0x0
19	RSZ_FIFO_IN_BLK_ERR	This event signals that the minimum vertical blanking period has not been respected causing errors in the input data buffering submodule. This event will be triggered when the rsz_int_reg event of frame N is triggered before the rsz_int_dma of frame N + 1. This event would typically happen at the transition between two frames because there is not enough vertical blanking between frames: the firmware must take care to ensure enough vertical blanking. The hardware cannot recover from this error. It will be required to perform a reset of the IP. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
18	RSZ_FIFO_OVF	RESIZER module event: This event signals that overflow happened in the input data buffering submodule or in the RSZ output interface.. This event would typically happen because the video port pixel clock is too high. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
17	RSZ_INT_CYC_RZB	RESIZER module event: This event is the circular interrupt for RESIZER #B. An event can be triggered every time that RSZ_IRQ_RZB output lines have been written out to the RZB_SDR_Y buffer. The range can go from 1 to 8192 lines. Usually, this value should be such that the circular buffer vertical size (set by the RZB_SDR_Y_PTR_E register) is a multiple of RSZ_IRQ_RZB. Note that at the time the interrupt is triggered, the actual data write has not taken place. It may take a few hundred of cycles to complete the data write into system memory. This is not an issue since the start of the buffer is read first. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
16	RSZ_INT_CYC_RZA	RESIZER module event: This event is the circular interrupt for RESIZER #A. An event can be triggered every time that RSZ_IRQ_RZA output lines have been written out to the RZA_SDR_Y buffer. The range can go from 1 to 8192 lines. Usually, the circular buffer vertical size (set by the RZA_SDR_Y_PTR_E register) should be a multiple of RSZ_IRQ_RZA. Note that at the time the interrupt is triggered, the actual data write has not taken place. It may take a few hundred of cycles to complete the data write into system memory. This is not an issue since the start of the buffer is read first. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
15	RSZ_INT_DMA	RESIZER module event: This event is triggered when the last EOF (of the two MTC interfaces) is sent out to the BL and that the resizer core has returned to idle. rsz_int_dma is a true indication that all processing is finished for the particular frame on both resizer engines. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
14	RSZ_INT_LAST_PIX	RESIZER module event: This event is triggered when the last pixel of the valid area is received. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
13	RSZ_INT_REG	RESIZER module event: This event is triggered when the first pixel of the valid area is received. Shadowed registers can be updated at any time but the new value will take effect on the next rsz_int_reg event. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
12	H3A_INT	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
11	AF_INT	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
10	AEW_INT	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
9	IPIPEIF_IRQ	IPIPEIF module interrupt Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
8	IPIPE_INT_HST	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
7	IPIPE_INT_BSC	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
6	IPIPE_INT_DMA	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
5	IPIPE_INT_LAST_PIX	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
4	IPIPE_INT_REG	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
3	ISIF_INT_3	Set this bit to disable the LSC interrupt issued by the 2D-LSC block. Four types of 2D-LSC can be generated and mapped to the INT_3 line. For more information, see Section 3.3.6.10.1.5, *ISS ISP ISIF 2D-LSC Events and Status Checking.* Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
2	ISIF_INT_2	VD interrupt 2 event. Set this bit to disable the interrupt mapped to the INT_2 line. This interrupt is set based on the VD pulse position after receiving a configured number of horizontal pulse signals. For more information, see Section 3.3.6.19.1, *ISS ISP ISIF VDINT0, VDINT1 and VDINT2 Interrupts.* Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
1	ISIF_INT_1	VD interrupt 1 event. Set this bit to disable the interrupt mapped to the INT_1 line. This interrupt is set based on the VD pulse position after receiving a configured number of horizontal pulse signals. For more information, see Section 3.3.6.19.1, *ISS ISP ISIF VDINT0, VDINT1 and VDINT2 Interrupts.* Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
0	ISIF_INT_0	VD interrupt 0 event. Set this bit to disable the interrupt mapped to the INT_0 line. This interrupt is set based on the VD pulse position after receiving a configured number of horizontal pulse signals. For more information, see Section 3.3.6.19.1, *ISS ISP ISIF VDINT0, VDINT1 and VDINT2 Interrupts.* Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

NOTE: Setting or disabling the ISP submodule interrupts mapped to the ISP lines requires a configuration to receive or disable the IRQ events at a higher ISS interrupt merger level. For information about how IRQ lines are handled at the ISP level, see Table 166. For information about how IRQs are handled at the ISS top-level before the signals leave ISS boundaries, see , *ISS Interrupt Merger.*

3.5.1.9 ISP5_DMAENABLE_SET

Table 269. ISP5_DMAENABLE_SET

Address Offset	0x0000 0064		
Physical Address	0x5505 0064 0x5C01 0064	**Instance**	ISS_ISP5_SYS1_CORTEX-M3 ISS_ISP5_SYS1_L3
Description	Per-line DMA enable bit vector Write 1 to set (enable DMA request generation). Readout equal to corresponding _CLR register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											IPIPE_INT_DPC_RNEW1	IPIPE_INT_LAST_PIX	IPIPE_INT_DPC_RNEW0	IPIPE_INT_HST	IPIPE_INT_BSC	

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0000000
4	IPIPE_INT_DPC_RNEW1	Enable for ISP DMA request generation on line #2 This DMA request must be set to transfer the DPC data from memory to the IPIPE internal RAM. Write 0x0: No action Write 0x1: Enable DMA Read 0x1: DMA enabled Read 0x0: DMA disabled	RW W1toSet	0
3	IPIPE_INT_LAST_PIX	Enable for ISP DMA request generation on line #3 This DMA request must be set to transfer the GAMMA data from memory to the IPIPE internal RAM or to initialize the DPC table. One must set the ISP5_CTRL.DMA3_CFG register before enabling this DMA request. Write 0x0: No action Write 0x1: Enable DMA Read 0x1: DMA enabled Read 0x0: DMA disabled	RW W1toSet	0
2	IPIPE_INT_DPC_RNEW0	Enable for ISP DMA request generation on line #2 This DMA request must be set to transfer the DPC data from memory to the IPIPE internal RAM. Write 0x0: No action Write 0x1: Enable DMA Read 0x1: DMA enabled Read 0x0: DMA disabled	RW W1toSet	0
1	IPIPE_INT_HST	Enable for ISP DMA request generation on line #1 This DMA request must be set to transfer the HIST data from the IPIPE internal RAM to memory. Write 0x0: No action Write 0x1: Enable DMA Read 0x1: DMA enabled Read 0x0: DMA disabled	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
0	IPIPE_INT_BSC	Enable for ISP DMA request generation on line #0 This DMA request must be set to transfer the BSC data from the IPIPE internal RAM to memory. Write 0x0: No action Write 0x1: Enable DMA Read 0x1: DMA enabled Read 0x0: DMA disabled	RW W1toSet	0

3.5.1.10 ISP5_DMAENABLE_CLR

Table 270. ISP5_DMAENABLE_CLR

Address Offset	0x0000 0068		
Physical Address	0x5505 0068 0x5C01 0068	**Instance**	ISS_ISP5_SYS1_CORTEX-M3 ISS_ISP5_SYS1_L3
Description	Per-line DMA clear bit vector Write 1 to clear (disable DMA request generation). Readout equal to corresponding _SET register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											IPIPE_INT_DPC_RNEW1	IPIPE_INT_LAST_PIX	IPIPE_INT_DPC_RNEW0	IPIPE_INT_HST	IPIPE_INT_BSC	

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0000000
4	IPIPE_INT_DPC_RNEW1	Clear for ISP DMA request generation on line ISS_DMA2. This DMA request must be set to transfer the DPC data from memory to the IPIPE internal RAM. Write 0x0: No action Write 0x1: Disable DMA Read 0x1: DMA enabled Read 0x0: DMA disabled	RW W1toClr	0
3	IPIPE_INT_LAST_PIX	Clear for ISP DMA request generation on ISS_DMA3. This DMA request must be set to transfer the GAMMA data from memory to the IPIPE internal RAM. Write 0x0: No action Write 0x1: Disable DMA Read 0x1: DMA enabled Read 0x0: DMA disabled	RW W1toClr	0
2	IPIPE_INT_DPC_RNEW0	Clear for ISP DMA request generation on ISS_DMA2. This DMA request must be set to transfer the DPC data from memory to the IPIPE internal RAM. Write 0x0: No action Write 0x1: Disable DMA Read 0x1: DMA enabled Read 0x0: DMA disabled	RW W1toClr	0
1	IPIPE_INT_HST	Clear for ISP DMA request generation on ISS_DMA1. This DMA request must be set to transfer the HIST data from the IPIPE internal RAM to memory. Write 0x0: No action Write 0x1: Disable DMA Read 0x1: DMA enabled Read 0x0: DMA disabled	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
0	IPIPE_INT_BSC	Clear for ISP DMA request generation on ISS_DMA0. This DMA request must be set to transfer the BSC data from the IPIPE internal RAM to memory. Write 0x0: No action Write 0x1: Disable DMA Read 0x1: DMA enabled Read 0x0: DMA disabled	RW W1toClr	0

3.5.1.11 ISP5_CTRL

Table 271. ISP5_CTRL

Address Offset	0x0000 006C		
Physical Address	0x5505 006C 0x5C01 006C	**Instance**	ISS_ISP5_SYS1_CORTEX-M3 ISS_ISP5_SYS1_L3
Description	ISP5 CONTROL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA3_CFG		RESERVED		BSC_RD_CHK	HST_RD_CHK	DPC_EVT_INI	MSTANDBY	VD_PULSE_EXT	PCLK_INV	MFLAG	MSTANDDBY_WAIT	RESERVED				BL_CLK_ENABLE	ISIF_CLK_ENABLE	H3A_CLK_ENABLE	RSZ_CLK_ENABLE	IPIPE_CLK_ENABLE	IPIPEIF_CLK_ENABLE	SYNC_ENABLE	PSYNC_CLK_SEL	VBUSM_CIDS				VBUSM_CPRIORITY			OCP_WRNP

Bits	Field Name	Description	Type	Reset
31:30	DMA3_CFG	This bit field selects the DMA transfer configuration which is used with the ISS_DMA3 DMA request signal. This DMA request is generated from IPIPE_INT_LAST_PIXEL event. One can choose to use this DMA request to transfer the DPC initialization data, the gamma table, or both. 0x0: No DMA request associated with ISS_DMA3. 0x1: DPC DMA request associated with ISS_DMA3. Expected DMA transfer size is 2KB in the range 0x8000-0x87FF. DPC_EVT_INI must be set to 0. 0x2: GAMMA DMA request associated with ISS_DMA3. Expected DMA transfer size is 6KB in the range 0xA800-0xBFFF. 0x3: DPC + GAMMA DMA request associated with ISS_DMA3. Expected DMA transfer size is 8KB in the range 0x8000-0x87FF and 0xA800-0xBFFF. DPC_EVT_INI must be set to 0.	RW	0x0
29:28	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
27	BSC_RD_CHK	When the BSC computation is enabled and the BSC DMA request is not used to read out the data, this register ensures that the data is read fast enough, else an interrupt ISP5_IRQSTATUS2_i[5] IPIPE_BSC_ERR is triggered. The hardware sets automatically this bit to 1 when software can start reading the memory. **CAUTION** It is the software responsibility to set this bit to 0 after reading the data. Once the MPU has read the BSC data, it must clear this register, else the ISP5_IRQSTATUS2_i[5] IPIPE_BSC_ERR will occur. Write 0x0: Clears the signal to avoid error generation. The software must write this bit to 0 after the last data read. Write 0x1: Reserved Read 0x1: The MPU can read the data from the memory. Needs to complete fast enough to avoid the interrupt generation. Read 0x0: No interrupt generation can happen	RW	0
26	HST_RD_CHK	When the HISTOGRAM computation is enabled and the HST DMA request is not used to read out the data, this register ensures that the data is read fast enough, else an interrupt ISP5_IRQSTATUS2_i[5] IPIPE_HST_ERR is triggered. The hardware sets automatically this bit to 1 when software can start reading the memory. **CAUTION** It is the software responsibility to set this bit to 0 after reading the data. Once the MPU has read the histogram data, it must clear this register, else the ISP5_IRQSTATUS2_i[5] IPIPE_HST_ERR will occur. Write 0x0: Clears the signal to avoid error generation. The software must write this bit to 0 after the last data read. Write 0x1: Reserved Read 0x1: The MPU can read the data from the memory. Needs to complete fast enough to avoid the interrupt generation. Read 0x0: No interrupt generation can happen	RW	0
25	DPC_EVT_INI	Select the IPIPE module event to be used to generate the DMA requests for the DPC submodule. 0x0: IPIPE_INT_LAST_PIX event is selected. 0x1: IPIPE_INT_DPC_INI event is selected.	RW	0

Bits	Field Name	Description	Type	Reset
24	MSTANDBY	MStandby signal assertion and de-assertion control for power management transitions. After software reset, this bit is asserted. Write "1" to transition from normal mode to idle mode. The firmware needs to ensure that no more ISP processing is ongoing before setting up this bit. Write "0" to transition from idle mode to normal mode. The software should poll ISP5_CTRL.MSTANDBY_WAIT = 0 after writing ISP5_CTRL.MSTANDBY = 0 in a transition from idle to normal mode. 0x0: De-assert MStandby signal. May not be immediate due to power management handshaking btw the MStandby and Wait signals. 0x1: Assert MStandby signal	RW	1
23	VD_PULSE_EXT	VD pulse extension enable This bit enables or disables the VD extension bridge. By default, the bridge is enabled. At ISS level, it is expected that ISP5_CTRL.VD_PULSE_EXT = 1 when the VPORT gets data from the CSI2 RX module and ISP5_CTRL.VD_PULSE_EXT = 0 when the VPORT gets data from the parallel interface or the SC module. There must be at least three clock cycles between the time this bit is modified and the HD/VD pulse for start of frame comes. 0x0: Disabled 0x1: Enabled	RW	1
22	PCLK_INV	Pixel clock inversion This bit enables or disables pixel clock inversion. The ISP always samples the data on the rising edge of the pixel clock. Enabling the inversion shifts the resampling period by 1/2 a pixel clock period. 0x0: Normal 0x1: Inversed	RW	0
21	MFLAG	MFlag signal generation control This bit controls how the OCP MFlag signal is generated on the ISS NOC. 0x0: The MFlag value is dynamic. 0x1: The MFlag value is static. The value is set with the ISP5_CTRL[3:1] VBUSM_CPRIORITY.	RW	0
20	MSTANDDBY_WAIT	MStandby / Wait power management status bit. The power management framework of the ISP is based on the handshaking of the MStandby and Wait signals. The software is not supposed to write inside the ISP slave port and initiate traffic when ISP5_CTRL.MSTANDBY bit is written. The software can poll this bit to know when Wait signal is deasserted. Read 0x1: MStandby signal is asserted. Read 0x0: MStandby signal is deasserted.	R	-
19:16	RESERVED		R	0x0
15	BL_CLK_ENABLE	BL clock enable 0x0: Disable 0x1: Enable	RW	0
14	ISIF_CLK_ENABLE	ISIF clock enable The ISP will return OCP_ERROR if one tries to program the module MMR or memory when the clock is disabled. There must be at least three clock cycles between the time this bit is modified and the HD/VD pulse for start of frame comes. 0x0: Disable 0x1: Enable	RW	0

Bits	Field Name	Description	Type	Reset
13	H3A_CLK_ENABLE	H3A clock enable The ISP will return OCP_ERROR if one tries to program the module MMR or memory when the clock is disabled. 0x0: Disable 0x1: Enable	RW	0
12	RSZ_CLK_ENABLE	RESIZER clock enable The ISP will return OCP_ERROR if one tries to program the module MMR or memory when the clock is disabled. 0x0: Disable 0x1: Enable	RW	0
11	IPIPE_CLK_ENABLE	IPIPE clock enable The ISP will return OCP_ERROR if one tries to program the module MMR or memory when the clock is disabled. 0x0: Disable 0x1: Enable	RW	0
10	IPIPEIF_CLK_ENABLE	IPIPEIF clock enable The ISP will return OCP_ERROR if one tries to program the module MMR or memory when the clock is disabled. 0x0: Disable 0x1: Enable	RW	0
9	SYNC_ENABLE	PCLK Sync module enable 0x0: Disable. must be used only when the video port is not receiving data; for example, data is read from the IPIPEIF module memory read port. 0x1: Enable. must only be used when the video port is not receiving data.	RW	0
8	PSYNC_CLK_SEL	PCLK Sync clock select. This bit selects the clock which is used to resynchronize the input pixel clock. 0x0: MMR_CLK. Can be used if the input pixel clock is always lower than 100 MHz. 0x1: DMA_CLK. must be used if the pixel clock is higher than 100 MHz.	RW	0
7:4	VBUSM_CIDS	BL MAX VBUSM CIDs The BL module supports up to 16 CIDs/tags. This bit field sets up the maximum number of CISs/tags that the BL can use. The actual number of CIDs/tags is setup by VBUSM_CIDS + 1. Tag number 0 to VBUSM_CIDS are used.	RW	0xF
3:1	VBUSM_CPRIORITY	BL VBUSM priority setting 0x0: High Priority VBUSM cpriority[2:0] = 0 0x1: High Priority VBUSM cpriority[2:0] = 1 0x2: Medium Priority VBUSM cpriority[2:0] = 2 0x3: Medium Priority VBUSM cpriority[2:0] = 3 0x4: Normal Priority VBUSM cpriority[2:0] = 4 0x5: Normal Priority VBUSM cpriority[2:0] = 5 0x6: Normal Priority VBUSM cpriority[2:0] = 6 0x7: Normal Priority VBUSM cpriority[2:0] = 7	RW	0x4

Bits	Field Name	Description	Type	Reset
0	OCP_WRNP	ISP OCP master port non-posted write control. 0x0: All writes are non posted. 0x1: All writes are posted.	RW	0

3.5.1.12 ISP5_MPSR

Table 272. ISP5_MPSR

Address Offset	0x0000 007C		
Physical Address	0x5505 007C 0x5C01 007C	**Instance**	ISS_ISP5_SYS1_CORTEX-M3 ISS_ISP5_SYS1_L3
Description	ISP memory access register. One need to pay attention when setting the bit fields in this register such that there is no conflict between the CPU and module accesses. Usually, the ISP modules must have access to the memories and it is only when the ISP is idle (vertical blanking period or module disabled that the CPU can access the memories.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							IPIPE_GAMMA_RGB_COPY	RESERVED			IPIPE_BSC_TB1	IPIPE_BSC_TB0	IPIPE_HST_TB3	IPIPE_HST_TB2	IPIPE_HST_TB1	IPIPE_HST_TB0	IPIPE_D3L_TB3	IPIPE_D3L_TB2	IPIPE_D3L_TB1	IPIPE_D3L_TB0	IPIPE_GBC_TB	IPIPE_YEE_TB	IPIPE_GMM_TBR	IPIPE_GMM_TBG	IPIPE_GMM_TBB	IPIPE_DPC_TB	ISIF_DCLAMP	ISIF_LSC_TB1	ISIF_LSC_TB0	ISIF_LIN_TB	RESERVED

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	IPIPE_GAMMA_RGB_COPY	GAMMA table RGB Copy This bit must be enable when one wants to use the same Gamma table for the R, G and B color components. When the CPU writes the R table, it is automatically copied to the G and B tables if this bit is set. 0x0: Copy disable Independent RGB gamma table 0x1: Copy enable Common RGB Gamma table	RW	0
23:21	RESERVED		R	0x0
20	IPIPE_BSC_TB1	IPIPE BSC TB1 memory access priority This memory is expected to be read by the CPU or the DMA to get BSC information during vertical blanking period. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
19	IPIPE_BSC_TB0	IPIPE BSC TB0 memory access priority This memory is expected to be read by the CPU or the DMA to get BSC information during vertical blanking period. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
18	IPIPE_HST_TB3	IPIPE histogram memory #3 access priority This memory is expected to be read by the CPU or the DMA to get HST information during vertical blanking period. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0

Bits	Field Name	Description	Type	Reset
17	IPIPE_HST_TB2	IPIPE histogram memory #2 access priority This memory is expected to be read by the CPU or the DMA to get HST information during vertical blanking period. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
16	IPIPE_HST_TB1	IPIPE histogram memory #1 access priority This memory is expected to be read by the CPU or the DMA to get HST information during vertical blanking period. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
15	IPIPE_HST_TB0	IPIPE histogram memory #0 access priority This memory is expected to be read by the CPU or the DMA to get HST information during vertical blanking period. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
14	IPIPE_D3L_TB3	D3L TB3 memory access priority This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
13	IPIPE_D3L_TB2	D3L TB2 memory access priority This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
12	IPIPE_D3L_TB1	D3L TB1 memory access priority This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
11	IPIPE_D3L_TB0	D3L TB0 memory access priority This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
10	IPIPE_GBC_TB	IPIPE GBC TB memory access priority This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
9	IPIPE_YEE_TB	YEE TB memory access priority This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0

Bits	Field Name	Description	Type	Reset
8	IPIPE_GMM_TBR	IPIPE Gamma LUT R memory arbitration This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
7	IPIPE_GMM_TBG	IPIPE Gamma LUT G memory arbitration This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
6	IPIPE_GMM_TBB	IPIPE Gamma LUT B memory arbitration This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
5	IPIPE_DPC_TB	IPIPE defect pixel memory arbitration This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
4	ISIF_DCLAMP	ISIF DC accumulation memory arbitration This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
3	ISIF_LSC_TB1	ISIF LSC memory 1 access This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods. 0x0: MODULE has memory access When the module has memory access, the potential concurrent CPU accesses (on the ISP memory map) to read the memory are stalled. The CPU will eventually get back the data during the horizontal or vertical blanking periods when the module is not making access anymore. 0x1: CPU has memory access When the CPU has memory access (read or write), it will cause data corruption if the module tries to perform concurrent memory accesses. The module cannot know that the read or write access has not taken place because of CPU accesses.	RW	0
2	ISIF_LSC_TB0	ISIF LSC memory 0 access This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods. 0x0: MODULE has memory access. When the module has memory access, the potential concurrent CPU accesses (on the ISP memory map) to read the memory are stalled. The CPU will eventually get back the data during the horizontal or vertical blanking periods when the module is not making access anymore. 0x1: CPU has memory access When the CPU has memory access (read or write), it will cause data corruption if the module tries to perform concurrent memory accesses. The module cannot know that the read or write access has not taken place because of CPU accesses.	RW	0

Bits	Field Name	Description	Type	Reset
1	ISIF_LIN_TB	ISIF linearity compensation memory arbitration This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
0	RESERVED		R	0

3.5.1.13 ISP5_BL_MTC_1

Table 273. ISP5_BL_MTC_1

Address Offset	0x0000 0080		
Physical Address	0x5505 0080 0x5C01 0080	**Instance**	ISS_ISP5_SYS1_CORTEX-M3 ISS_ISP5_SYS1_L3
Description	MEMORY REQUEST MINIMUM INTERVAL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISIF_R																IPIPEIF_R															

Bits	Field Name	Description	Type	Reset
31:16	ISIF_R	Sets the minimum interval btw two consecutive memory requests for the ISIF-Read port. Specified in number of interface clock cycles.	RW	0x0000
15:0	IPIPEIF_R	Sets the minimum interval btw two consecutive memory requests for the IPIPEIF-Read port. Specified in number of interface clock cycles.	RW	0x0000

3.5.1.14 ISP5_BL_MTC_2

Table 274. ISP5_BL_MTC_2

Address Offset	0x0000 0084		
Physical Address	0x5505 0084 0x5C01 0084	**Instance**	ISS_ISP5_SYS1_CORTEX-M3 ISS_ISP5_SYS1_L3
Description	MEMORY REQUEST MINIMUM INTERVAL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H3A_W																RESERVED															

Bits	Field Name	Description	Type	Reset
31:16	H3A_W	Sets the minimum interval btw two consecutive memory requests for the H3A-Write port. Specified in number of interface clock cycles.	RW	0x0000
15:0	RESERVED		R	0x0000

3.5.1.15 ISP5_BL_VBUSM

Table 275. ISP5_BL_VBUSM

Address Offset	0x0000 0088		
Physical Address	0x5505 0088 0x5C01 0088	**Instance**	ISS_ISP5_SYS1_CORTEX-M3 ISS_ISP5_SYS1_L3
Description	BL VBUSM TUNING REGISTER The settings in the register are static and not expected to be modified dynamically.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										MFLAG_THRES	LASTCMD_DLY				

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0000000
5	MFLAG_THRES	MFLAG Threshold value The value of this bit field is a threshold that is compared to the MFlag output of the ISP5. If the BL MFlag signal is greater or equal to this threshold, the last beat of the VBUSM command is delayed by ISP5_BL_VBUSM[4:0] LASTCMD_DLY cycles. Only values 0 and 1 are valid, the least-significant bit is tied off to 1 to make a 2-bit field. 0x0: Thres = 1 0x1: Thres = 3	RW	1
4:0	LASTCMD_DLY	The value of this bit field represents a delay expressed in cycles (L3 clock). This value is used to delay the last beat of the VBUSM command such that the ISP does not loose arbitration at the ISS level because the BL does not generate back to back requests by default. The last beat is delayed until the counter expires or the new request is accepted. This delay is used when the MFlag output of the ISP is greater or equal to ISP5_BL_VBUSM[5] MFLAG_THRES. One can set this value to 0 to disable the last command beat delay.	RW	0x04

3.5.2 ISS ISP5 SYS2 Registers

CAUTION

The ISS ISP5 SYS2 registers are limited to 32 bit and 16 bit data accesses; 8bit data access is not allowed and can corrupt register content.

Table 276 summarizes the ISS ISP5 SYS2 registers.

Table 276. ISS ISP5 SYS2 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ISS_ISP5_SYS2 Base Address Cortex-M3 Private Access	ISS_ISP5_SYS2 Base Address L3 Interconnect
ISP5_KEY_EN1	R	32	0x0000 0000	0x5505 00A0	0x5C01 00A0
ISP5_KEY_EN2	R	32	0x0000 0004	0x5505 00A4	0x5C01 00A4
ISP5_KEY_EN3	R	32	0x0000 0008	0x5505 00A8	0x5C01 00A8
ISP5_KEY_EN4	R	32	0x0000 000C	0x5505 00AC	0x5C01 00AC
ISP5_KEY_EN5	R	32	0x0000 0010	0x5505 00B0	0x5C01 00B0
ISP5_KEY_EN6	R	32	0x0000 0014	0x5505 00B4	0x5C01 00B4
ISP5_IRQSTATUS_RAW2_i (1)	RW	32	0x0000 0018 + (0x10 * i)	0x5505 00B8 + (0x10 * i)	0x5C01 00B8 + (0x10 * i)
ISP5_IRQSTATUS2_i (1)	RW	32	0x0000 001C + (0x10 * i)	0x5505 00BC + (0x10 * i)	0x5C01 00BC + (0x10 * i)
ISP5_IRQENABLE_SET2_i (1)	RW	32	0x0000 0020 + (0x10 * i)	0x5505 00C0 + (0x10 * i)	0x5C01 00C0 + (0x10 * i)
ISP5_IRQENABLE_CLR2_i (1)	RW	32	0x0000 0024 + (0x10 * i)	0x5505 00C4 + (0x10 * i)	0x5C01 00C4 + (0x10 * i)

(1) i = 0 to 3

3.5.2.1 ISP5_KEY_EN1

Table 277. ISP5_KEY_EN1

Address Offset	0x0000 0000		
Physical Address	0x5505 00A0 0x5C01 00A0	**Instance**	ISS_ISP5_SYS2_CORTEX-M3 ISS_ISP5_SYS2_L3
Description	IPIPE eFuse enable.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															KEY1_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	KEY1_EN	eFuse enable Equals 1 when ISP5_EFUSE3_EN = 1 or 0 otherwise. Read 0x1: Enable Read 0x0: Disable	R	0

3.5.2.2 ISP5_KEY_EN2

Table 278. ISP5_KEY_EN2

Address Offset	0x0000 0004		
Physical Address	0x5505 00A4 0x5C01 00A4	**Instance**	ISS_ISP5_SYS2_CORTEX-M3 ISS_ISP5_SYS2_L3
Description	ISIF eFuse enable.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															KEY1_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	KEY1_EN	eFuse enable Equals 1 when ISP5_EFUSE1_EN = 1 or 0 otherwise. Read 0x1: Enable Read 0x0: Disable	R	1

3.5.2.3 ISP5_KEY_EN3

Table 279. ISP5_KEY_EN3

Address Offset	0x0000 0008		
Physical Address	0x5505 00A8 0x5C01 00A8	**Instance**	ISS_ISP5_SYS2_CORTEX-M3 ISS_ISP5_SYS2_L3
Description	ISIF eFuse enable.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															KEY_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	KEY_EN	eFuse enable Equals 1 when ISP5_EFUSE3_EN = 1 or 0 otherwise. Read 0x1: Enable Read 0x0: Disable	R	0

3.5.2.4 ISP5_KEY_EN4

Table 280. ISP5_KEY_EN4

Address Offset	0x0000 000C		
Physical Address	0x5505 00AC 0x5C01 00AC	**Instance**	ISS_ISP5_SYS2_CORTEX-M3 ISS_ISP5_SYS2_L3
Description	IPIPEIF eFuse enable.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														KEY2_EN	KEY1_EN

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	KEY2_EN	eFuse enable Equals 1 when ISP5_EFUSE4_EN = 1 or 0 otherwise. Read 0x1: Enable Read 0x0: Disable	R	0
0	KEY1_EN	eFuse enable Equals 1 when ISP5_EFUSE1_EN = 1 or 0 otherwise. Read 0x1: Enable Read 0x0: Disable	R	1

3.5.2.5 ISP5_KEY_EN5

Table 281. ISP5_KEY_EN5

Address Offset	0x0000 0010		
Physical Address	0x5505 00B0 0x5C01 00B0	**Instance**	ISS_ISP5_SYS2_CORTEX-M3 ISS_ISP5_SYS2_L3
Description	H3A eFuse enable.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															KEY_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	KEY_EN	eFuse enable Equals 1 when ISP5_EFUSE2_EN = 1 or 0 otherwise. Read 0x1: Enable Read 0x0: Disable	R	0

3.5.2.6 ISP5_KEY_EN6

Table 282. ISP5_KEY_EN6

Address Offset	0x0000 0014		
Physical Address	0x5505 00B4 0x5C01 00B4	**Instance**	ISS_ISP5_SYS2_CORTEX-M3 ISS_ISP5_SYS2_L3
Description	H3A eFuse enable.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															KEY_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	KEY_EN	eFuse enable Equals 1 when ISP5_EFUSE3_EN = 1 or 0 otherwise. Read 0x1: Enable Read 0x0: Disable	R	0

3.5.2.7 ISP5_IRQSTATUS_RAW2_i

Table 283. ISP5_IRQSTATUS_RAW2_i

Address Offset	0x0000 0018 + (0x10 * i)	**Index**	i = 0 to 3
Physical Address	0x5505 00B8 + (0x10 * i) 0x5C01 00B8 + (0x10 * i)	**Instance**	ISS_ISP5_SYS2_CORTEX-M3 ISS_ISP5_SYS2_L3
Description	Per-event raw interrupt status vector. Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug. Overflow / underflow errors are not recoverable at ISP level, a software reset is required at ISS level. The ISP outputs 4 interrupt lines ISP5_IRQ0 to ISP5_IRQ3. Any internal ISP event can be merged on the 4 lines. A same event must be enabled on only one interrupt line.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										IPIPE_BSC_ERR	IPIPE_HST_ERR	ISIF_OVF	IPIPE_BOXCAR_OVF	IPIPEIF_UDF	H3A_OVF

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0000000
5	IPIPE_BSC_ERR	IPIPE BSC memory read error This error will happen when the BSC data is not read fast enough by the MPU or the DMA. When the data is read with the MPU, one need to pay attention to clear the ISP5_CTRL[27] BSC_RD_CHK bit immediately after reading the last data, else this event will be set. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
4	IPIPE_HST_ERR	IPIPE HISTOGRAM memory read error This error will happen when the histogram data is not read fast enough by the MPU or the DMA. When the data is read with the MPU, one need to pay attention to clear the ISP5_CTRL[26] HST_RD_CHK bit immediately after reading the last data, else this event will be set. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
3	ISIF_OVF	ISIF module overflow Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
2	IPIPE_BOXCAR_OVF	IPIPE BOXCAR module overflow Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
1	IPIPEIF_UDF	IPIPEIF module underflow interrupt Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
0	H3A_OVF	H3A module overflow interrupt. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

3.5.2.8 ISP5_IRQSTATUS2_i

Table 284. ISP5_IRQSTATUS2_i

Address Offset	0x0000 001C + (0x10 * i)	**Index**	i = 0 to 3
Physical Address	0x5505 00BC + (0x10 * i) 0x5C01 00BC + (0x10 * i)	**Instance**	ISS_ISP5_SYS2_CORTEX-M3 ISS_ISP5_SYS2_L3
Description	Per-event "enabled" interrupt status vector. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled). Overflow / underflow errors are not recoverable at ISP level, a software reset is required at ISS level. The ISP outputs 4 interrupt lines ISP5_IRQ0 to ISP5_IRQ3. Any internal ISP event can be merged on the 4 lines. A same event must be enabled on only one interrupt line.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										IPIPE_BSC_ERR	IPIPE_HST_ERR	ISIF_OVF	IPIPE_BOXCAR_OVF	IPIPEIF_UDF	H3A_OVF

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0000000
5	IPIPE_BSC_ERR	IPIPE BSC memory read error This error will happen when the BSC data is not read fast enough by either the MPU or the DMA. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
4	IPIPE_HST_ERR	IPIPE HISTOGRAM memory read error This error will happen when the histogram data is not read fast enough by either the MPU or the DMA. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
3	ISIF_OVF	ISIF module overflow Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
2	IPIPE_BOXCAR_OVF	IPIPE BOXCAR module overflow Overflow errors are not recoverable at ISP level, a software reset is required at ISS level. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
1	IPIPEIF_UDF	IPIPEIF module underflow interrupt Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
0	H3A_OVF	H3A module overflow interrupt. Overflow errors are not recoverable at ISP level, a software reset is required at ISS level. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

3.5.2.9 ISP5_IRQENABLE_SET2_i

Table 285. ISP5_IRQENABLE_SET2_i

Address Offset	0x0000 0020 + (0x10 * i)	**Index**	i = 0 to 3
Physical Address	0x5505 00C0 + (0x10 * i) 0x5C01 00C0 + (0x10 * i)	**Instance**	ISS_ISP5_SYS2_CORTEX-M3 ISS_ISP5_SYS2_L3
Description	Per-event interrupt enable bit vector. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register. Overflow / underflow errors are not recoverable at ISP level, a software reset is required at ISS level. The ISP outputs 4 interrupt lines ISP5_IRQ0 to ISP5_IRQ3. Any internal ISP event can be merged on the 4 lines. A same event must be enabled on only one interrupt line.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										IPIPE_BSC_ERR	IPIPE_HST_ERR	ISIF_OVF	IPIPE_BOXCAR_OVF	IPIPEIF_UDF	H3A_OVF

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0000000
5	IPIPE_BSC_ERR	IPIPE BSC memory read error This error will happen when the BSC data is not read fast enough by either the MPU or the DMA.	RW W1toSet	0
4	IPIPE_HST_ERR	IPIPE HISTOGRAM memory read error This error will happen when the histogram data is not read fast enough by either the MPU or the DMA.	RW W1toSet	0
3	ISIF_OVF	ISIF module overflow	RW W1toSet	0
2	IPIPE_BOXCAR_OVF	IPIPE BOXCAR module overflow	RW W1toSet	0
1	IPIPEIF_UDF	IPIPEIF module underflow interrupt	RW W1toSet	0
0	H3A_OVF	H3A module overflow interrupt. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled	RW W1toSet	0

3.5.2.10 ISP5_IRQENABLE_CLR2_i

Table 286. ISP5_IRQENABLE_CLR2_i

Address Offset	0x0000 0024 + (0x10 * i)	**Index**	i = 0 to 3
Physical Address	0x5505 00C4 + (0x10 * i) 0x5C01 00C4 + (0x10 * i)	**Instance**	ISS_ISP5_SYS2_CORTEX-M3 ISS_ISP5_SYS2_L3
Description	Per-event interrupt enable bit vector. Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register. Overflow / underflow errors are not recoverable at ISP level, a software reset is required at ISS level. The ISP outputs 4 interrupt lines ISP5_IRQ0 to ISP5_IRQ3. Any internal ISP event can be merged on the 4 lines. A same event must be enabled on only one interrupt line.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										IPIPE_BSC_ERR	IPIPE_HST_ERR	ISIF_OVF	IPIPE_BOXCAR_OVF	IPIPEIF_UDF	H3A_OVF

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0000000
5	IPIPE_BSC_ERR	IPIPE BSC memory read error This error will happen when the BSC data is not read fast enough by either the MPU or the DMA. Write 0x0: No action Write 0x1: Disable Interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled	RW W1toClr	0
4	IPIPE_HST_ERR	IPIPE HISTOGRAM memory read error This error will happen when the histogram data is not read fast enough by either the MPU or the DMA. Write 0x0: No action Write 0x1: Disable Interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled	RW W1toClr	0
3	ISIF_OVF	ISIF module overflow Write 0x0: No action Write 0x1: Disable Interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled	RW W1toClr	0
2	IPIPE_BOXCAR_OVF	IPIPE BOXCAR module overflow Write 0x0: No action Write 0x1: Disable Interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
1	IPIPEIF_UDF	IPIPEIF module underflow interrupt Write 0x0: No action Write 0x1: Disable Interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled	RW W1toClr	0
0	H3A_OVF	H3A module overflow interrupt. Write 0x0: No action Write 0x1: Disable Interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled	RW W1toClr	0

3.5.3 ISS RESIZER Registers

Table 287 summarizes the ISS RESIZER registers.

Table 287. ISS RESIZER Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ISS_RESIZER Base Address Cortex-M3 Private Access	ISS_RESIZER Base Address L3 Interconnect
RSZ_REVISION	R	32	0x0000 0000	0x5505 0400	0x5C01 0400
RSZ_SYSCONFIG	RW	32	0x0000 0004	0x5505 0404	0x5C01 0404
RESERVED	R	32	0x0000 0008	0x5505 0408	0x5C01 0408
RSZ_IN_FIFO_CTRL	RW	32	0x0000 000C	0x5505 040C	0x5C01 040C
RSZ_GNC	R	32	0x0000 0010	0x5505 0410	0x5C01 0410
RSZ_FRACDIV	RW	32	0x0000 0014	0x5505 0414	0x5C01 0414
RSZ_SRC_EN	RW	32	0x0000 0020	0x5505 0420	0x5C01 0420
RSZ_SRC_MODE	RW	32	0x0000 0024	0x5505 0424	0x5C01 0424
RSZ_SRC_FMT0	RW	32	0x0000 0028	0x5505 0428	0x5C01 0428
RSZ_SRC_FMT1	RW	32	0x0000 002C	0x5505 042C	0x5C01 042C
RSZ_SRC_VPS	RW	32	0x0000 0030	0x5505 0430	0x5C01 0430
RSZ_SRC_VSZ	RW	32	0x0000 0034	0x5505 0434	0x5C01 0434
RSZ_SRC_HPS	RW	32	0x0000 0038	0x5505 0438	0x5C01 0438
RSZ_SRC_HSZ	RW	32	0x0000 003C	0x5505 043C	0x5C01 043C
RSZ_DMA_RZA	RW	32	0x0000 0040	0x5505 0440	0x5C01 0440
RSZ_DMA_RZB	RW	32	0x0000 0044	0x5505 0444	0x5C01 0444
RSZ_DMA_STA	R	32	0x0000 0048	0x5505 0448	0x5C01 0448
RSZ_GCK_MMR	RW	32	0x0000 004C	0x5505 044C	0x5C01 044C
RESERVED	R	32	0x0000 0050	0x5505 0450	0x5C01 0450
RSZ_GCK_SDR	RW	32	0x0000 0054	0x5505 0454	0x5C01 0454
RSZ_IRQ_RZA	RW	32	0x0000 0058	0x5505 0458	0x5C01 0458
RSZ_IRQ_RZB	RW	32	0x0000 005C	0x5505 045C	0x5C01 045C
RSZ_YUV_Y_MIN	RW	32	0x0000 0060	0x5505 0460	0x5C01 0460
RSZ_YUV_Y_MAX	RW	32	0x0000 0064	0x5505 0464	0x5C01 0464
RSZ_YUV_C_MIN	RW	32	0x0000 0068	0x5505 0468	0x5C01 0468
RSZ_YUV_C_MAX	RW	32	0x0000 006C	0x5505 046C	0x5C01 046C
RSZ_YUV_PHS	RW	32	0x0000 0070	0x5505 0470	0x5C01 0470
RSZ_SEQ	RW	32	0x0000 0074	0x5505 0474	0x5C01 0474
RZA_EN	RW	32	0x0000 0078	0x5505 0478	0x5C01 0478
RZA_MODE	RW	32	0x0000 007C	0x5505 047C	0x5C01 047C
RZA_420	RW	32	0x0000 0080	0x5505 0480	0x5C01 0480
RZA_I_VPS	RW	32	0x0000 0084	0x5505 0484	0x5C01 0484
RZA_I_HPS	RW	32	0x0000 0088	0x5505 0488	0x5C01 0488
RZA_O_VSZ	RW	32	0x0000 008C	0x5505 048C	0x5C01 048C
RZA_O_HSZ	RW	32	0x0000 0090	0x5505 0490	0x5C01 0490
RZA_V_PHS_Y	RW	32	0x0000 0094	0x5505 0494	0x5C01 0494
RZA_V_PHS_C	RW	32	0x0000 0098	0x5505 0498	0x5C01 0498
RZA_V_DIF	RW	32	0x0000 009C	0x5505 049C	0x5C01 049C
RZA_V_TYP	RW	32	0x0000 00A0	0x5505 04A0	0x5C01 04A0
RZA_V_LPF	RW	32	0x0000 00A4	0x5505 04A4	0x5C01 04A4
RZA_H_PHS	RW	32	0x0000 00A8	0x5505 04A8	0x5C01 04A8
RZA_H_PHS_ADJ	RW	32	0x0000 00AC	0x5505 04AC	0x5C01 04AC
RZA_H_DIF	RW	32	0x0000 00B0	0x5505 04B0	0x5C01 04B0

Table 287. ISS RESIZER Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	ISS_RESIZER Base Address Cortex-M3 Private Access	ISS_RESIZER Base Address L3 Interconnect
RZA_H_TYP	RW	32	0x0000 00B4	0x5505 04B4	0x5C01 04B4
RZA_H_LPF	RW	32	0x0000 00B8	0x5505 04B8	0x5C01 04B8
RZA_DWN_EN	RW	32	0x0000 00BC	0x5505 04BC	0x5C01 04BC
RZA_DWN_AV	RW	32	0x0000 00C0	0x5505 04C0	0x5C01 04C0
RZA_RGB_EN	RW	32	0x0000 00C4	0x5505 04C4	0x5C01 04C4
RZA_RGB_TYP	RW	32	0x0000 00C8	0x5505 04C8	0x5C01 04C8
RZA_RGB_BLD	RW	32	0x0000 00CC	0x5505 04CC	0x5C01 04CC
RZA_SDR_Y_BAD_H	RW	32	0x0000 00D0	0x5505 04D0	0x5C01 04D0
RZA_SDR_Y_BAD_L	RW	32	0x0000 00D4	0x5505 04D4	0x5C01 04D4
RZA_SDR_Y_SAD_H	RW	32	0x0000 00D8	0x5505 04D8	0x5C01 04D8
RZA_SDR_Y_SAD_L	RW	32	0x0000 00DC	0x5505 04DC	0x5C01 04DC
RZA_SDR_Y_OFT	RW	32	0x0000 00E0	0x5505 04E0	0x5C01 04E0
RZA_SDR_Y_PTR_S	RW	32	0x0000 00E4	0x5505 04E4	0x5C01 04E4
RZA_SDR_Y_PTR_E	RW	32	0x0000 00E8	0x5505 04E8	0x5C01 04E8
RZA_SDR_C_BAD_H	RW	32	0x0000 00EC	0x5505 04EC	0x5C01 04EC
RZA_SDR_C_BAD_L	RW	32	0x0000 00F0	0x5505 04F0	0x5C01 04F0
RZA_SDR_C_SAD_H	RW	32	0x0000 00F4	0x5505 04F4	0x5C01 04F4
RZA_SDR_C_SAD_L	RW	32	0x0000 00F8	0x5505 04F8	0x5C01 04F8
RZA_SDR_C_OFT	RW	32	0x0000 00FC	0x5505 04FC	0x5C01 04FC
RZA_SDR_C_PTR_S	RW	32	0x0000 0100	0x5505 0500	0x5C01 0500
RZA_SDR_C_PTR_E	RW	32	0x0000 0104	0x5505 0504	0x5C01 0504
RZB_EN	RW	32	0x0000 0108	0x5505 0508	0x5C01 0508
RZB_MODE	RW	32	0x0000 010C	0x5505 050C	0x5C01 050C
RZB_420	RW	32	0x0000 0110	0x5505 0510	0x5C01 0510
RZB_I_VPS	RW	32	0x0000 0114	0x5505 0514	0x5C01 0514
RZB_I_HPS	RW	32	0x0000 0118	0x5505 0518	0x5C01 0518
RZB_O_VSZ	RW	32	0x0000 011C	0x5505 051C	0x5C01 051C
RZB_O_HSZ	RW	32	0x0000 0120	0x5505 0520	0x5C01 0520
RZB_V_PHS_Y	RW	32	0x0000 0124	0x5505 0524	0x5C01 0524
RZB_V_PHS_C	RW	32	0x0000 0128	0x5505 0528	0x5C01 0528
RZB_V_DIF	RW	32	0x0000 012C	0x5505 052C	0x5C01 052C
RZB_V_TYP	RW	32	0x0000 0130	0x5505 0530	0x5C01 0530
RZB_V_LPF	RW	32	0x0000 0134	0x5505 0534	0x5C01 0534
RZB_H_PHS	RW	32	0x0000 0138	0x5505 0538	0x5C01 0538
RZB_H_PHS_ADJ	RW	32	0x0000 013C	0x5505 053C	0x5C01 053C
RZB_H_DIF	RW	32	0x0000 0140	0x5505 0540	0x5C01 0540
RZB_H_TYP	RW	32	0x0000 0144	0x5505 0544	0x5C01 0544
RZB_H_LPF	RW	32	0x0000 0148	0x5505 0548	0x5C01 0548
RZB_DWN_EN	RW	32	0x0000 014C	0x5505 054C	0x5C01 054C
RZB_DWN_AV	RW	32	0x0000 0150	0x5505 0550	0x5C01 0550
RZB_RGB_EN	RW	32	0x0000 0154	0x5505 0554	0x5C01 0554
RZB_RGB_TYP	RW	32	0x0000 0158	0x5505 0558	0x5C01 0558
RZB_RGB_BLD	RW	32	0x0000 015C	0x5505 055C	0x5C01 055C
RZB_SDR_Y_BAD_H	RW	32	0x0000 0160	0x5505 0560	0x5C01 0560
RZB_SDR_Y_BAD_L	RW	32	0x0000 0164	0x5505 0564	0x5C01 0564

Table 287. ISS RESIZER Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	ISS_RESIZER Base Address Cortex-M3 Private Access	ISS_RESIZER Base Address L3 Interconnect
RZB_SDR_Y_SAD_H	RW	32	0x0000 0168	0x5505 0568	0x5C01 0568
RZB_SDR_Y_SAD_L	RW	32	0x0000 016C	0x5505 056C	0x5C01 056C
RZB_SDR_Y_OFT	RW	32	0x0000 0170	0x5505 0570	0x5C01 0570
RZB_SDR_Y_PTR_S	RW	32	0x0000 0174	0x5505 0574	0x5C01 0574
RZB_SDR_Y_PTR_E	RW	32	0x0000 0178	0x5505 0578	0x5C01 0578
RZB_SDR_C_BAD_H	RW	32	0x0000 017C	0x5505 057C	0x5C01 057C
RZB_SDR_C_BAD_L	RW	32	0x0000 0180	0x5505 0580	0x5C01 0580
RZB_SDR_C_SAD_H	RW	32	0x0000 0184	0x5505 0584	0x5C01 0584
RZB_SDR_C_SAD_L	RW	32	0x0000 0188	0x5505 0588	0x5C01 0588
RZB_SDR_C_OFT	RW	32	0x0000 018C	0x5505 058C	0x5C01 058C
RZB_SDR_C_PTR_S	RW	32	0x0000 0190	0x5505 0590	0x5C01 0590
RZB_SDR_C_PTR_E	RW	32	0x0000 0194	0x5505 0594	0x5C01 0594

3.5.3.1 RSZ_REVISION

Table 288. RSZ_REVISION

Address Offset	0x0000 0000		
Physical Address	0x5505 0400 0x5C01 0400	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	See(1)

(1) TI internal data

3.5.3.2 RSZ_SYSCONFIG

Table 289. RSZ_SYSCONFIG

Address Offset	0x0000 0004		
Physical Address	0x5505 0404 0x5C01 0404	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	SYSTEM CONFIGURATION REGISTER This register is not shadowed. There is no standalone software reset for the resizer module.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						RSZB_CLK_EN	RSZA_CLK_EN	RESERVED							AUTOGATING

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9	RSZB_CLK_EN	Resizer B clock enable This bit enable to enable / disable the RESIZER B clock. Note that it is a second level clock enable. This bit has effect only if RSZ_GCK_SDR is set to 1. 0x0: off 0x1: on	RW	0
8	RSZA_CLK_EN	Resizer A clock enable This bit enable to enable / disable the RESIZER A clock. Note that it is a second level clock enable. This bit has effect only if RSZ_GCK_SDR is set to 1. 0x0: off 0x1: on	RW	0
7:1	RESERVED		R	0x00
0	AUTOGATING	Internal Clock Gating Strategy Enables or disables auto clock gating. **NOTE:** This is only for debug purposes. When this bit is set to "0" autogating is not performed on any of the clocks, thus the clocks stay free running. Though, they are still controlled by the clock enable bit fields RSZA_CLK_EN and RSZB_CLK_EN. 0x0: Clocks are free running 0x1: Automatic clock gating strategy.	RW	1

3.5.3.3 RSZ_IN_FIFO_CTRL

Table 290. RSZ_IN_FIFO_CTRL

Address Offset	0x0000 000C		
Physical Address	0x5505 040C 0x5C01 040C	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	INPUT DATA BUFFER CONTROL REGISTER This register is not shadowed		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								THRLD_LOW								RESERVED								THRLD_HIGH							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	THRLD_LOW	When RSZ_IN_FIFO_CTRL.THRLD_HIGH = RSZ_IN_FIFO_CTRL.THRLD_LOW, the rsz_stall_input is not asserted. The only purpose of the RSZ_IN_FIFO_CTRL.THRLD_LOW register is to prevent rsz_stall_input signal assertion.	RW	0x0000
15:13	RESERVED		R	0x0
12:0	THRLD_HIGH	High threshold value. The rsz_stall_input signal is asserted if 2 lines of circular buffer are full and the third line has more pixels than RSZ_IN_FIFO_CTRL.THRLD_HIGH. The rsz_stall_input signal stays high as long as one full line is not free for receiving further data. THRLD_HIGH is in terms of line size and can at max be programmed equal to the input line size (RSZ_SRC_HSZ).	RW	0x0000

3.5.3.4 RSZ_GNC

Table 291. RSZ_GNC

Address Offset	0x0000 0010		
Physical Address	0x5505 0410 0x5C01 0410	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	GENERIC PARAMETER REGISTER		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RSZB_MEM_LINE_SIZE								RESERVED								RSZA_MEM_LINE_SIZE							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	RSZB_MEM_LINE_SIZE	Resizer #B memory line size (pixels). The output image cannot exceed this size.	R	0x0920
15:13	RESERVED		R	0x0
12:0	RSZA_MEM_LINE_SIZE	Resizer #A memory line size (pixels). The output image cannot exceed this size.	R	0x1500

3.5.3.5 RSZ_FRACDIV

Table 292. RSZ_FRACDIV

Address Offset	0x0000 0014		
Physical Address	0x5505 0414 0x5C01 0414	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	Fractional clock divider settings		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RSZ_FRACDIV															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	RSZ_FRACDIV	Fractional clock divider value. The fractional clock divider gates the read requests made to the input data buffer such that the input data buffer is read at an average frequency equal to FFCLK instead of FCLK. The value of FFCLK depends upon the upscaling ratios as well as the input pixel clock: see the functional spec. We have FFCLK = FCLK / FRACDIV MHz and RSZ_FRACDIV = 65536 / FRACDIV. When RSZ_ FRACDIV = 65536, we have: FFCLK = FCLK.	RW	0xFFFF

3.5.3.6 RSZ_SRC_EN

Table 293. RSZ_SRC_EN

Address Offset	0x0000 0020		
Physical Address	0x5505 0420 0x5C01 0420	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER ENABLE REGISTER This register is not shadowed		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	EN	Resizer module enable The start flag of the RESIZER module. When EN is set to 1, the RESIZER module starts the processing from the next rising edge of the VD pulse. If the processing mode of the RESIZER module is set to "one shot", the EN bit is cleared to 0 after the end of the processing. One has to pay attention that when this bit is enabled and 0x0: Disable 0x1: Enable	RW	0

3.5.3.7 RSZ_SRC_MODE

Table 294. RSZ_SRC_MODE

Address Offset	0x0000 0024		
Physical Address	0x5505 0424 0x5C01 0424	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	This register is not shadowed		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														WRT	OST

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	WRT	Video port WEN signal selection This bit selects whether the WEN signal which is present on the IPIPE and IPIPEIF video port is used or not to select the input data. If WRT is 0, the RESIZER module ignores the WEN signal and processes all image frame while RESIZER is enabled. If WRT is 1, the RESIZER module only processes the lines that arrived while the WEN is high. HD is used to sample the WEN signal. 0x0: Disable 0x1: Enable	RW	0
0	OST	The processing mode selection of the RESIZER module. Value 0 indicates the mode of free run, value 1 indicates the mode of one shot. 0x0: Free running 0x1: One shot	RW	0

3.5.3.8 RSZ_SRC_FMT0

Table 295. RSZ_SRC_FMT0

Address Offset	0x0000 0028		
Physical Address	0x5505 0428 0x5C01 0428	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	This register is not shadowed		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														BYPASS	SEL

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	BYPASS	Pass Through This bit enables or disables the RESIZER module pass through mode. The pass through mode can transfer images which are 8K pixel wide. When it is enabled, the input data buffer and the resizer engines are bypassed. 0x0: Pass through off = normal output mode, the input data buffer is used. 0x1: Pass through on = normal output mode, the input data buffer is bypassed.	RW	0
0	SEL	Input selection This bit selects which of the two video port is selected to push data through the RESIZER module. 0x0: IPIPE 0x1: IPIPEIF	RW	0

3.5.3.9 RSZ_SRC_FMT1

Table 296. RSZ_SRC_FMT1

Address Offset	0x0000 002C		
Physical Address	0x5505 042C 0x5C01 042C	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													COL	IN420	RAW

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	COL	Y/C selection This bit is valid only if the input data is YUV4:2:0 (IN420 = 1). It enables to specify where the data which is input to the RESIZER module is luma or chroma data. 0x0: Y data is input 0x1: Chroma data is input	RW	0
1	IN420	Chroma Format Selection This bit sets the chroma undersampling when YUV data is input to the RESIZER module. 0x0: YUV4:2:2 is input 0x1: YUV4:2:0 is input	RW	0
0	RAW	Pass-through mode input data format selection This bit affects the horizontal reversal (flipping) process. 0x0: Flipping preserves YCbCr format 0x1: Flipping preserves RAW format	RW	0

3.5.3.10 RSZ_SRC_VPS

Table 297. RSZ_SRC_VPS

Address Offset	0x0000 0030		
Physical Address	0x5505 0430 0x5C01 0430	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	VERTICAL POSITION REGISTER This register is not shadowed		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VPS															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	VPS	Vertical Start Position Sets the vertical position of the global frame from the rising edge of the VD. The RSZ module will start the image processing from the VPS'th line. This value can be odd or even whatever the input data format.	RW	0x0000

3.5.3.11 RSZ_SRC_VSZ

Table 298. RSZ_SRC_VSZ

Address Offset	0x0000 0034		
Physical Address	0x5505 0434 0x5C01 0434	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	VERTICAL SIZER REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VSZ															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	VSZ	Vertical Processing Size Sets the vertical size of the processing area. The RSZ module will process (VSZ+1) lines. This value can be odd or even whatever the input data format.	RW	0x0000

3.5.3.12 RSZ_SRC_HPS

Table 299. RSZ_SRC_HPS

Address Offset	0x0000 0038		
Physical Address	0x5505 0438 0x5C01 0438	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	HORIZONTAL POSITION REGISTER This register is not shadowed.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HPS															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	HPS	Horizontal Start Position The RSZ_SRC_HPS register has two functions: The first function is to compensate for possible delay between the HD pulse and the first valid data. It is possible for this delay to be different than 0 when the RESIZER module gets its input data from the VP connected to the IPIPEIF module (the offset value can be odd or even). When data are coming from the IPIPE module, it is not required to resynchronize HD and the first valid data. The second function is to crop the data in the horizontal direction. When used for cropping, only RSZ_SRC_HPS must be even or null.	RW	0x0000

3.5.3.13 RSZ_SRC_HSZ

Table 300. RSZ_SRC_HSZ

Address Offset	0x0000 003C		
Physical Address	0x5505 043C 0x5C01 043C	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	HORIZONTAL SIZE REGISTER The HSZ value is given by HSZ concatenated with HSZ_LSB.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			HSZ												

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	HSZ	Horizontal size Sets the horizontal size of the processing area. The RSZ module processes (HSZ+1) pixels. (HSZ+1) must be even for YUV4:2:2 and RAW data. The valid available values for HSZ are 1~xxxx.	RW	0x0000

3.5.3.14 RSZ_DMA_RZA

Table 301. RSZ_DMA_RZA

Address Offset	0x0000 0040		
Physical Address	0x5505 0440 0x5C01 0440	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - MEMORY REQUEST MINIMUM INTERVAL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RZA															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	RZA	Sets the minimum interval btw two consecutive memory request for resizer #A. Specified in number of interface clock cycles. Values of 0, 1, and 2 are used as a condition to keep the bandwidth limiter off. When this function is enabled, this value must be greater than 10 cycles.	RW	0x0000

3.5.3.15 RSZ_DMA_RZB

Table 302. RSZ_DMA_RZB

Address Offset	0x0000 0044		
Physical Address	0x5505 0444 0x5C01 0444	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - MEMORY REQUEST MINIMUM INTERVAL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RZB															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	RZB	Sets the minimum interval btw two consecutive memory request for resizer #B. Specified in number of interface clock cycles. Values of 0, 1, and 2 are used as a condition to keep the bandwidth limiter off. When this function is enabled, this value must be greater than 10 cycles.	RW	0x0000

3.5.3.16 RSZ_DMA_STA

Table 303. RSZ_DMA_STA

Address Offset	0x0000 0048		
Physical Address	0x5505 0448 0x5C01 0448	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER STATUS REGISTER		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															STATUS

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	STATUS	Resizer process status This bit is set in the time window from rsz_int_reg to rsz_int_dma. Read 0x1: Active Read 0x0: Not active	R	0

3.5.3.17 RSZ_GCK_MMR

Table 304. RSZ_GCK_MMR

Address Offset	0x0000 004C		
Physical Address	0x5505 044C 0x5C01 044C	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	MMR CLOCK CONTROL REGISTER This register is not shadowed.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MMR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	MMR	The on/off selection of the MMR interface clock which is used for MMR register access. 0x0: Off 0x1: On	RW	0

3.5.3.18 RSZ_GCK_SDR

Table 305. RSZ_GCK_SDR

Address Offset	0x0000 0054		
Physical Address	0x5505 0454 0x5C01 0454	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	CORE CLOCK CONTROL REGISTER This register is not shadowed.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CORE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	CORE	RSZ Core Clock Enable. This bit enables or disables the resizer core functional clock. When this bit is off, the resizer core (interpolator) is automatically bypassed (resizer-bypass mode of pass-through mode is selected depending on RSZ_SRC_FMT0 value). In resizer-bypass mode or pass-through mode, no up-scaling or downscaling process is operated. 0x0: Resizer core clock disabled. Resizer in bypass mode, if RSZ_SRC_FMT0.BYPASS = 0 Resizer in pass-through, if RSZ_SRC_FMT0.BYPASS = 1 0x1: Resizer core clock enabled. Resizer in rescaling mode, if RSZ_SRC_FMT0.BYPASS = 0 Resizer in pass-through, if RSZ_SRC_FMT0.BYPASS = 1	RW	0

3.5.3.19 RSZ_IRQ_RZA

Table 306. RSZ_IRQ_RZA

Address Offset	0x0000 0058		
Physical Address	0x5505 0458 0x5C01 0458	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - CIRCULAR BUFFER INTERRUPT INTERVAL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RZA							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	RZA	Resizer A circular buffer interval Sets the circular buffer interval for Resizer A. The interrupt is triggered every time (RZA+1) lines are written to the circular buffer (Y buffer). The range goes from 1 to 8192 lines. Usually, the circular buffer vertical size should be a multiple of RZA.	RW	0x0000

3.5.3.20 RSZ_IRQ_RZB

Table 307. RSZ_IRQ_RZB

Address Offset	0x0000 005C		
Physical Address	0x5505 045C 0x5C01 045C	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - CIRCULAR BUFFER INTERRUPT INTERVAL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RZB							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	RZB	Resizer B circular buffer interval Sets the circular buffer interval for Resizer B. The interrupt is triggered every time (RZB+1) lines are written to the circular buffer (Y buffer). The range goes from 1 to 8192 lines. Usually, the circular buffer vertical size should be a multiple of RZB.	RW	0x0000

3.5.3.21 RSZ_YUV_Y_MIN

Table 308. RSZ_YUV_Y_MIN

Address Offset	0x0000 0060		
Physical Address	0x5505 0460 0x5C01 0460	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	LUMINANCE SATURATION REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MIN							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	MIN	The minimum value of Luminance (8bits unsigned). If the value of the Luminance is smaller than VAL, it will be clipped to VAL. This bit field must be set to its default values when the resizer is set in pass-though mode.	RW	0x00

3.5.3.22 RSZ_YUV_Y_MAX

Table 309. RSZ_YUV_Y_MAX

Address Offset	0x0000 0064		
Physical Address	0x5505 0464 0x5C01 0464	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	LUMINANCE SATURATION REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MAX							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	MAX	The maximum value of Luminance (8bits unsigned). If the value of the Luminance is larger than VAL, it will be clipped to VAL. This bit field must be set to its default values when the resizer is set in pass-through mode.	RW	0xFF

3.5.3.23 RSZ_YUV_C_MIN

Table 310. RSZ_YUV_C_MIN

Address Offset	0x0000 0068		
Physical Address	0x5505 0468 0x5C01 0468	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	CHROMINANCE SATURATION REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MIN							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	MIN	The minimum value of Chrominance (8bits unsigned). If the value of the Chrominance is smaller than VAL, it will be clipped to VAL. This bit field must be set to its default values when the resizer is set in pass-though mode.	RW	0x00

3.5.3.24 RSZ_YUV_C_MAX

Table 311. RSZ_YUV_C_MAX

Address Offset	0x0000 006C		
Physical Address	0x5505 046C 0x5C01 046C	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	CHROMINANCE SATURATION REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MAX							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	MAX	The maximum value of Chrominance (8bits unsigned). If the value of the Chrominance is larger than VAL, it will be clipped to VAL. This bit field must be set to its default values when the resizer is set in pass-through mode.	RW	0xFF

3.5.3.25 RSZ_YUV_PHS

Table 312. RSZ_YUV_PHS

Address Offset	0x0000 0070		
Physical Address	0x5505 0470 0x5C01 0470	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	The phase position of the output of the Chrominance.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															POS

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	POS	The phase position of the output of the chrominance. The RESIZER module does not change the relative position of the chroma samples vs. the luma samples between the input and output and the chroma position at the output of the IPIPE module and at the output of the RESIZER module must be identical. In other words, we must have RSZ_YUV_PHS.POS = IPIPE_YUV_PHS.POS. 0x0: Same position with Luminance: cosited 0x1: The middle of the luminance: centered	RW	0

3.5.3.26 RSZ_SEQ

Table 313. RSZ_SEQ

Address Offset	0x0000 0074		
Physical Address	0x5505 0474 0x5C01 0474	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											CRV	VRVB	HRVB	VRVA	HRVA

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0000000
4	CRV	Chroma sampling point change 0x0: Chroma sampling point is not changed 0x1: Chroma sampling point is changed from odd-numbered pixels to even-number pixels. The pixel at the left end is removed and the pixel at the right end is duplicated.	RW	0
3	VRVB	Resizer B - Vertical reversal of output image 0x0: Processed pixels are output in the order of input (normal operation) in vertical direction. 0x1: The order of output data is flipped top to bottom.	RW	0
2	HRVB	Resizer B -Horizontal reversal of output image 0x0: Processed pixels are output in the order of input (normal operation) in horizontal direction. 0x1: The order of output data is flipped left to right.	RW	0
1	VRVA	Resizer A - Vertical reversal of output image 0x0: Processed pixels are output in the order of input (normal operation) in vertical direction. 0x1: The order of output data is flipped top to bottom.	RW	0
0	HRVA	Resizer A - Horizontal reversal of output image 0x0: Processed pixels are output in the order of input (normal operation) in horizontal direction. 0x1: The order of output data is flipped left to right.	RW	0

3.5.3.27 RZA_EN

Table 314. RZA_EN

Address Offset	0x0000 0078		
Physical Address	0x5505 0478 0x5C01 0478	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - ENABLE REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	EN	Enable resizer #A This bit is latched on video port VD input. The reason is that the resizer must only starts the processing on a clean frame boundary. In one-shot mode, this bit is negated on VD. 0x0: Disable 0x1: Enable	RW	0

3.5.3.28 RZA_MODE

Table 315. RZA_MODE

Address Offset	0x0000 007C		
Physical Address	0x5505 047C 0x5C01 047C	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER #A MODE REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	MODE	Select "Free Run mode" or "One Shot Mode" 0x0: Free run 0x1: One shot	RW	0

3.5.3.29 RZA_420

Table 316. RZA_420

Address Offset	0x0000 0080		
Physical Address	0x5505 0480 0x5C01 0480	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	YEN/CEN: 0/0: in = YUV4:2:2 input, out = YUV4:2:2 output 0/1: in = YUV4:2:2 input, out = Chrominance of YUV4:2:0 output 1/0: in = YUV4:2:2 input, out = Luminance of YUV4:2:0 output 1/1: in = YUV4:2:2 input, out = YUV4:2:0 output		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CEN	YEN

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	CEN	Output Enable for Chrominance This bit is valid in 422 input mode. When CEN=0 and YEN=0, output is 422 0x0: C output disable 0x1: C output enable and 422to420 conversion enabled	RW	0
0	YEN	Output Enable for Luminance. This bit is valid in 422 input mode. When CEN=0 and YEN=0, output is 422 0x0: Y output disable 0x1: Y output enable and 422to420 conversion enabled	RW	0

3.5.3.30 RZA_I_VPS

Table 317. RZA_I_VPS

Address Offset	0x0000 0084		
Physical Address	0x5505 0484 0x5C01 0484	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - INPUT VERTICAL START REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VPS															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	VPS	Input Vertical Position Sets the vertical start position of the input image within the global frame. It enables to crop data into the global frame. After SRC_VPS, the Vps'th line is processed as the first line in each image.	RW	0x0000

3.5.3.31 RZA_I_HPS

Table 318. RZA_I_HPS

Address Offset	0x0000 0088		
Physical Address	0x5505 0488 0x5C01 0488	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - INPUT HORIZONTAL START REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HPS															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	HPS	Input Horizontal Position Sets the horizontal position of the first pixel for each line within the global frame. After SRC_HPS, the pixel at the VAL'th position is processed as the first pixel. This value must be even.	RW	0x0000

3.5.3.32 RZA_O_VSZ

Table 319. RZA_O_VSZ

Address Offset	0x0000 008C		
Physical Address	0x5505 048C 0x5C01 048C	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - OUTPUT VERTICAL SIZER REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								VSZ							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	VSZ	The target output size of the resized image. The number of output lines is (VSZ+1). Set 479, when 480 lines of output is required.	RW	0x0000

3.5.3.33 RZA_O_HSZ

Table 320. RZA_O_HSZ

Address Offset	0x0000 0090		
Physical Address	0x5505 0490 0x5C01 0490	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - OUTPUT HORIZONTAL SIZE REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								HSZ							HSZ_LSB

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:1	HSZ	The horizontal size of output image. The number of pixel in each line is (HSZ+1). Set 479, when 480 pixels are required. This value must be lower than the max memory line size supported by the resizer engine, except in RAW pass through mode. Note that the LSB of the 13-bit HSZ value is fixed to 1 such that the horizontal size is always even.	RW	0x000
0	HSZ_LSB	The least significant bit of HSZ is forced to 1.	R	1

3.5.3.34 RZA_V_PHS_Y

Table 321. RZA_V_PHS_Y

Address Offset	0x0000 0094		
Physical Address	0x5505 0494 0x5C01 0494	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - INITIAL LUMINANCE PHASE OF VERTICAL RESIZING PROCESS When YUV4:2:2 data are output, the phase values for luma and chroma should typically be equal, that is, RZX_V_PHS_Y= RZX_V_PHS_C. The following constraints apply when setting the initial vertical phases ABS(RZX_V_PHS_Y - RZX_V_PHS_C) RZX_V_DIF. This constraint means that at most the distance between the initial phases for luminance and chrominance is not expected to exceed the distance between two luma pixels. Note that the absolute value is used, hence, the initial luma phase can be greater than the initial chroma phase or the other way around. As a reminder, the distance between two output pixels for luma is given by RZX_V_DIF.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								Y							

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	Y	The initial value for the luma phase in vertical resizing process. This value is in U14Q8 fractional format.	RW	0x0000

3.5.3.35 RZA_V_PHS_C

Table 322. RZA_V_PHS_C

Address Offset	0x0000 0098		
Physical Address	0x5505 0498 0x5C01 0498	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - INITIAL CHROMINANCE PHASE OF VERTICAL RESIZING PROCESS When YUV4:2:2 data are output, the phase values for luma and chroma should typically be equal, that is, RZX_V_PHS_Y= RZX_V_PHS_C. The following constraints apply when setting the initial vertical phases ABS(RZX_V_PHS_Y - RZX_V_PHS_C) RZX_V_DIF. This constraint means that at most the distance between the initial phases for luminance and chrominance is not expected to exceed the distance between two luma pixels. Note that the absolute value is used, hence, the initial luma phase can be greater than the initial chroma phase or the other way around. As a reminder, the distance between two output pixels for luma is given by RZX_V_DIF.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								C							

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	C	The initial value for the chroma phase in vertical resizing process. This value is in U14Q8 fractional format.	RW	0x0000

3.5.3.36 RZA_V_DIF

Table 323. RZA_V_DIF

Address Offset	0x0000 009C		
Physical Address	0x5505 049C 0x5C01 049C	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - VERTICAL RESIZER REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																V															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	V	The parameter for vertical resize. The actual resizing ratio is 256/RZA_V_DIF. In normal mode: 16 = RZA_V_DIF = 4096. In down-scale mode: 256 = RZA_V_DIF = 4096.	RW	0x0000

3.5.3.37 RZA_V_TYP

Table 324. RZA_V_TYP

Address Offset	0x0000 00A0		
Physical Address	0x5505 04A0 0x5C01 04A0	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - INTERPOLATION METHOD FOR VERTICAL RESIZING		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														C	Y

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	C	Selection of resizing method for chrominance: vertical 0x0: 4-tap cubic convolution (default) 0x1: 2-tap linear interpolation	RW	0
0	Y	Selection of resizing method for luminance: vertical 0x0: 4-tap cubic convolution (default) 0x1: 2-tap linear interpolation	RW	0

3.5.3.38 RZA_V_LPF

Table 325. RZA_V_LPF

Address Offset	0x0000 00A4		
Physical Address	0x5505 04A4 0x5C01 04A4	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - VERTICAL LPF INTENSITY REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				C								Y			

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:6	C	The intensity parameter for chroma vertical low pass filtering.	RW	0x00
5:0	Y	The intensity parameter for luma vertical low pass filtering.	RW	0x00

3.5.3.39 RZA_H_PHS

Table 326. RZA_H_PHS

Address Offset	0x0000 00A8		
Physical Address	0x5505 04A8 0x5C01 04A8	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - INITIAL PHASE OF HORIZONTAL RESIZING PROCESS		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PHS							

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	PHS	Initial value for the phase in horizontal resizing process, that is, the sampling position is shifted. This value is in U14Q8 fractional format. Example: If RZX_H_PHS = 128, the first output pixel is sampled at the center of the first two valid input pixels. If RZX_I_HPS=100 and RZX_H_PHS=128, the first output pixel is resampled at the center of the 100-th and the 101-st input pixels.	RW	0x0000

3.5.3.40 RZA_H_PHS_ADJ

Table 327. RZA_H_PHS_ADJ

Address Offset	0x0000 00AC		
Physical Address	0x5505 04AC 0x5C01 04AC	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - LUMINANCE HORIZONTAL PHASE ADJUSTMENT The RZA_H_PHS_ADJ register enables to adjust the horizontal phase for the luma component when averaging is enabled (the horizontal averaging disrupts the relative sampling point between luminance and chrominance when YUV4:2:2 cosited data is input), that is, the relative phase between luma and chroma is different before and after the horizontal averager.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ADJ							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8:0	ADJ	Horizontal phase adjustment value. This value is in U9Q8 fractional format. This value is expected to be equal to zero if the averager is disabled or if input chroma is centered.	RW	0x000

3.5.3.41 RZA_H_DIF

Table 328. RZA_H_DIF

Address Offset	0x0000 00B0		
Physical Address	0x5505 04B0 0x5C01 04B0	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - HORIZONTAL RESIZER REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								H							

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	H	The parameter for horizontal resizing process. The actual resizing ratio is 256/VAL. In normal mode 16= RSZ_RZA_H_DIF=4096 In down-scale mode 256=RSZ_RZA_H_DIF=4096	RW	0x0000

3.5.3.42 RZA_H_TYP

Table 329. RZA_H_TYP

Address Offset	0x0000 00B4		
Physical Address	0x5505 04B4 0x5C01 04B4	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	Resize-A		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														C	Y

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	C	Selection of resizing method for chrominance: horizontal 0x0: 4-tap cubic convolution (default) 0x1: 2-tap linear interpolation	RW	0
0	Y	Selection of resizing method for luminance: horizontal 0x0: 4-tap cubic convolution (default) 0x1: 2-tap linear interpolation	RW	0

3.5.3.43 RZA_H_LPF

Table 330. RZA_H_LPF

Address Offset	0x0000 00B8		
Physical Address	0x5505 04B8 0x5C01 04B8	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - HORIZONTAL LPF INTENSITY REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C								Y							

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:6	C	Horizontal LPF Intensity for Chrominance	RW	0x00
5:0	Y	Selection of resizing method for Luminance in horizontal direction	RW	0x00

3.5.3.44 RZA_DWN_EN

Table 331. RZA_DWN_EN

Address Offset	0x0000 00BC		
Physical Address	0x5505 04BC 0x5C01 04BC	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER #A - DOWNSCALE ENABLE REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															DWN_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	DWN_EN	Resizer downscale enable 0x0: Off. Normal operation: upscale and downscale are allowed. 0x1: On. Downscale mode.	RW	0

3.5.3.45 RZA_DWN_AV

Table 332. RZA_DWN_AV

Address Offset	0x0000 00C0		
Physical Address	0x5505 04C0 0x5C01 04C0	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	Resize-A		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								V				H			

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0000000
5:3	V	Vertical averaging size : $1/2^{(VWT+1)}$ The range goes from 1/2 to 1/256 in power of 2. 0:_DIV2 1/2 down scale 1:_DIV4 1/4 down scale 2:_DIV8 1/8 down scale 3:_DIV16 1/16 down scale 4:_DIV32 1/32 down scale 5:_DIV64 1/64 down scale 6:_DIV128 1/128 down scale 7:_DIV256 1/256 down scale	RW	0x0
2:0	H	Horizontal averaging size : $1/2^{(HWT+1)}$ The range goes from 1/2 to 1/256 in power of 2. 0:_DIV2 1/2 down scale 1:_DIV4 1/4 down scale 2:_DIV8 1/8 down scale 3:_DIV16 1/16 down scale 4:_DIV32 1/32 down scale 5:_DIV64 1/64 down scale 6:_DIV128 1/128 down scale 7:_DIV256 1/256 down scale	RW	0x0

3.5.3.46 RZA_RGB_EN

Table 333. RZA_RGB_EN

Address Offset	0x0000 00C4		
Physical Address	0x5505 04C4 0x5C01 04C4	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER #A - RGB OUTPUT ENABLE		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															RGB_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	RGB_EN	Enable of RGB output In pass through mode, this register must be 0. This bit can only be set to 1 when YUV4:2:2 data are output. YUV4:2:2 data output is selected when SRC_FMT1.IN420 = 0 and RZA_420.YEN = RZA_420.CEN = 0 0x0: Off (YCbCr output) 0x1: On (RGB output)	RW	0

3.5.3.47 RZA_RGB_TYP

Table 334. RZA_RGB_TYP

Address Offset	0x0000 00C8		
Physical Address	0x5505 04C8 0x5C01 04C8	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - RGB OUTPUT CONTROL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													MSK1	MSK0	TYP

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	MSK1	Enables masking of the last 2 pixels This bit is used to mask the 2 last pixels at the image boundary which are affected by the YUV4:2:2 to YUV4:4:4 conversion. 0x0: output the last 2 pixels 0x1: mask the last 2 pixels (Resizer do not output them.)	RW	0
1	MSK0	Enables masking of the first 2 pixels This bit is used to mask the 2 first pixels at the image boundary which are affected by the YUV4:2:2 to YUV4:4:4 conversion. 0x0: output the first 2 pixels 0x1: mask the first 2 pixels (Resizer do not output them.)	RW	0
0	TYP	16bit/32bit output selection 0x0: 32-bit output: alpha + R + G + B (8 bit each) This mode comes with performance degradation. The maximum input frequency in this mode is 160 MHz. This due to the fact that the output is 4 bytes / pixel. 0x1: 16-bit output: R(5 bit) + G (6 bit) + B (5 bit)	RW	0

3.5.3.48 RZA_RGB_BLD

Table 335. RZA_RGB_BLD

Address Offset	0x0000 00CC		
Physical Address	0x5505 04CC 0x5C01 04CC	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - RGB BLEND REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								BLD							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	BLD	The alpha value used in 32-bit RGBA output mode	RW	0x00

3.5.3.49 RZA_SDR_Y_BAD_H

Table 336. RZA_SDR_Y_BAD_H

Address Offset	0x0000 00D0		
Physical Address	0x5505 04D0 0x5C01 04D0	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - OUTPUT MEMORY BASE ADDRESS REGISTER (HIGH) This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0, RGB5:6:5, RGBA. RAW: RAW data is written to this address YUV4:2:2: YUV data is written to this address YUV4:2:0: Y data is written to this address RGB5:6:5: 16-bit RGB data is written to this address RGBA: 32-bit RGBA data is written to this address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_BAD_H															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	Y_BAD_H	Memory Base Address Sets the 16 upper bits of the 32-bit base address of the circular buffer in memory.	RW	0x0000

3.5.3.50 RZA_SDR_Y_BAD_L

Table 337. RZA_SDR_Y_BAD_L

Address Offset	0x0000 00D4		
Physical Address	0x5505 04D4 0x5C01 04D4	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - OUTPUT MEMORY BASE ADDRESS REGISTER (LOW) This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0, RGB5:6:5, RGBA. RAW: RAW data is written to this address YUV4:2:2: YUV data is written to this address YUV4:2:0: Y data is written to this address RGB5:6:5: 16-bit RGB data is written to this address RGBA: 32-bit RGBA data is written to this address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_BAD_L															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	Y_BAD_L	Memory Base Address Sets the 16 lower bits of the 32-bit base address of the circular buffer in memory. It is a byte address. YUV4:2:0 format (output data on 8 bits): The two least significant bits must be set to 00 when horizontal reversal mode is off. The two least significant bits must be set to 11 when horizontal reversal mode is on. RAW, RGB5:6:5 and YUV4:2:2 formats (output data on 16 bits): The two least significant bits must be set to 00 when horizontal reversal mode is off. The two least significant bits must be set to 11 when horizontal reversal mode is on. RGBA format (output data on 32 bits): The three least significant bits must be set to "000" when horizontal reversal mode is off. The three least significant bits must be set to "111" when horizontal reversal mode is on.	RW	0x0000

3.5.3.51 RZA_SDR_Y_SAD_H

Table 338. RZA_SDR_Y_SAD_H

Address Offset	0x0000 00D8		
Physical Address	0x5505 04D8 0x5C01 04D8	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - OUTPUT MEMORY START ADDRESS REGISTER (HIGH) This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0, RGB5:6:5, RGBA. RAW: RAW data is written to this address YUV4:2:2: YUV data is written to this address YUV4:2:0: Y data is written to this address RGB5:6:5: 16-bit RGB data is written to this address RGBA: 32-bit RGBA data is written to this address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_SAD_H															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	Y_SAD_H	Memory Start Address Sets the 16 upper bits of the 32-bit start address in memory.	RW	0x0000

3.5.3.52 RZA_SDR_Y_SAD_L

Table 339. RZA_SDR_Y_SAD_L

Address Offset	0x0000 00DC		
Physical Address	0x5505 04DC 0x5C01 04DC	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - OUTPUT MEMORY START ADDRESS REGISTER (LOW) This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0, RGB5:6:5, RGBA. RAW: RAW data is written to this address YUV4:2:2: YUV data is written to this address YUV4:2:0: Y data is written to this address RGB5:6:5: 16-bit RGB data is written to this address RGBA: 32-bit RGBA data is written to this address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_SAD_L															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	Y_SAD_L	Memory Start Address Sets 16 lower bits of the 32-bit start address in memory. It is a byte address. For every frame, the first line of data will be written to this address. We have: SAD = BAD + (PTR_S x OFT) and PTR_S PTR_E If the first line must be written at the beginning of the circular buffer memory then SAD = BAD and PTR_S = 0. YUV4:2:0 format (output data on 8 bits): The two least significant bits must be set to 00 when horizontal reversal mode is off. The two least significant bits must be set to 11 when horizontal reversal mode is on. RAW, RGB5:6:5 and YUV4:2:2 formats (output data on 16 bits): The two least significant bits must be set to 00 when horizontal reversal mode is off. The two least significant bits must be set to 11 when horizontal reversal mode is on. RGBA format (output data on 32 bits): The three least significant bits must be set to "000" when horizontal reversal mode is off. The three least significant bits must be set to "111" when horizontal reversal mode is on.	RW	0x0000

3.5.3.53 RZA_SDR_Y_OFT

Table 340. RZA_SDR_Y_OFT

Address Offset	0x0000 00E0		
Physical Address	0x5505 04E0 0x5C01 04E0	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - OUTPUT MEMORY OFFSET REGISTER This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0 or RGBA.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_OFT															

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0000
16:0	Y_OFT	Memory Line Offset Sets the size of each line in the circular buffer. It is expressed in bytes and unsigned. Note that OFT does not necessary corresponds to the size of a line in a frame, it can be much bigger. The line offset must be a multiple of 128 bytes (bits [6:0] of RZX_SDR_Y_OFT and RZX_SDR_C_OFT must be set to 0). Example: line 0 address = SAD line 1 address = SAD + 1 x OFT line 2 address = SAD + 2 x OFT	RW	0x00000

3.5.3.54 RZA_SDR_Y_PTR_S

Table 341. RZA_SDR_Y_PTR_S

Address Offset	0x0000 00E4		
Physical Address	0x5505 04E4 0x5C01 04E4	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - OUTPUT MEMORY START ADDRESS REGISTER This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0 or RGBA.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			Y_PTR_S												

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	Y_PTR_S	Start Line of Memory Pointer Sets the vertical position of the first output line in the output memory space. This value is expressed in number of lines. The hardware uses it to set up the initial value of the circular buffer. It must be set up such as PTR_S = (SAD - BAD) / OFT. This value must be set to 0 when RSZ_RZA_SDR_Y_BAD = RSZ_RZA_SDR_Y_SAD.	RW	0x0000

3.5.3.55 RZA_SDR_Y_PTR_E

Table 342. RZA_SDR_Y_PTR_E

Address Offset	0x0000 00E8		
Physical Address	0x5505 04E8 0x5C01 04E8	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - OUTPUT MEMORY END ADDRESS REGISTER This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0 or RGBA.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								Y_PTR_E							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	Y_PTR_E	End Line of Memory Pointer Sets the maximum number of lines to be stored in the output memory space. This value is expressed in number of lines. When the number of output lines exceeds this value, the address restarts from the first address in the output memory space (BAD).	RW	0x0000

3.5.3.56 RZA_SDR_C_BAD_H

Table 343. RZA_SDR_C_BAD_H

Address Offset	0x0000 00EC		
Physical Address	0x5505 04EC 0x5C01 04EC	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - OUTPUT MEMORY BASE ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0. U and V data are written into this buffer.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_BAD_H															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	C_BAD_H	Memory Base Address Sets the 16 higher bits of the 32-bit base address of the circular buffer in memory.	RW	0x0000

3.5.3.57 RZA_SDR_C_BAD_L

Table 344. RZA_SDR_C_BAD_L

Address Offset	0x0000 00F0		
Physical Address	0x5505 04F0 0x5C01 04F0	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - OUTPUT MEMORY BASE ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0. U and V data are written into this buffer.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_BAD_L															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	C_BAD_L	Memory Base Address Sets the 16 lower bits of the 32-bit base address of the circular buffer in memory. It is a byte address. YUV4:2:0 format (output data on 8 bits): The two least significant bits must be set to 00 when horizontal reversal mode is off. The two least significant bits must be set to 11 when horizontal reversal mode is on.	RW	0x0000

3.5.3.58 RZA_SDR_C_SAD_H

Table 345. RZA_SDR_C_SAD_H

Address Offset	0x0000 00F4		
Physical Address	0x5505 04F4 0x5C01 04F4	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - OUTPUT MEMORY START ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0. U and V data are written into this buffer.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_SAD_H															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	C_SAD_H	Memory Base Address Sets the 16 higher bits of the 32-bit start address in memory.	RW	0x0000

3.5.3.59 RZA_SDR_C_SAD_L

Table 346. RZA_SDR_C_SAD_L

Address Offset	0x0000 00F8		
Physical Address	0x5505 04F8 0x5C01 04F8	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - OUTPUT MEMORY START ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0. U and V data are written into this buffer.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_SAD_L															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	C_SAD_L	Memory Base Address Sets the 16 lower bits of the 32-bit start address in memory. It is a byte address. For every frame, the first line of data will be written to this address. We have: SAD = BAD + (PTR_S x OFT) and PTR_S PTR_E If the first line must be written at the beginning of the circular buffer memory then SAD = BAD and PTR_S = 0. YUV4:2:0 format (output data on 8 bits): The two least significant bits must be set to 00 when horizontal reversal mode is off. The two least significant bits must be set to 11 when horizontal reversal mode is on.	RW	0x0000

3.5.3.60 RZA_SDR_C_OFT

Table 347. RZA_SDR_C_OFT

Address Offset	0x0000 00FC		
Physical Address	0x5505 04FC 0x5C01 04FC	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - OUTPUT MEMORY OFFSET REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_OFT															

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0000
16:0	C_OFT	Memory Line Offset Sets the size of each line in the circular buffer. It is expressed in bytes and unsigned. Note that OFT does not necessary corresponds to the size of a line in a frame, it can be much bigger. The line offset must be a multiple of 128 bytes (bits [6:0] of RZX_SDR_Y_OFT and RZX_SDR_C_OFT must be set to 0). Example: line 0 address = SAD line 1 address = SAD + 1 x OFT line 2 address = SAD + 2 x OFT	RW	0x00000

3.5.3.61 RZA_SDR_C_PTR_S

Table 348. RZA_SDR_C_PTR_S

Address Offset	0x0000 0100		
Physical Address	0x5505 0500 0x5C01 0500	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - OUTPUT MEMORY START ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								C_PTR_S							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	C_PTR_S	Start Line of Memory Pointer Sets the vertical position of the first output line in the output memory space. This value is expressed in number of lines. The hardware uses it to set up the initial value of the circular buffer. It must be set up such as PTR_S = (SAD - BAD) / OFT. This value must be set to 0 when RSZ_RZA_SDR_C_BAD = RSZ_RZA_SDR_C_SAD.	RW	0x0000

3.5.3.62 RZA_SDR_C_PTR_E

Table 349. RZA_SDR_C_PTR_E

Address Offset	0x0000 0104		
Physical Address	0x5505 0504 0x5C01 0504	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER A - OUTPUT MEMORY END ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								C_PTR_E							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	C_PTR_E	End Line of Memory Pointer Sets the maximum number of lines to be stored in the output memory space. This value is expressed in number of lines. When the number of output lines exceeds this value, the address restarts from the first address in the output memory space (BAD).	RW	0x0000

3.5.3.63 RZB_EN

Table 350. RZB_EN

Address Offset	0x0000 0108		
Physical Address	0x5505 0508 0x5C01 0508	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - ENABLE REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	EN	Enable resizer #A This bit is latched on the video port VD input signal. The reason is that the resizer must only starts the processing on a clean frame boundary. In one-shot mode, this bit is negated on VD. 0x0: Disable 0x1: Enable	RW	0

3.5.3.64 RZB_MODE

Table 351. RZB_MODE

Address Offset	0x0000 010C		
Physical Address	0x5505 050C 0x5C01 050C	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B MODE REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	MODE	Select "Free Run mode" or "One Shot Mode" 0x0: Free run 0x1: One shot	RW	0

3.5.3.65 RZB_420

Table 352. RZB_420

Address Offset	0x0000 0110		
Physical Address	0x5505 0510 0x5C01 0510	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	YEN/CEN: 0/0: in = YUV4:2:2 input, out = YUV4:2:2 output 0/1: in = YUV4:2:2 input, out = Chrominance of YUV4:2:0 output 1/0: in = YUV4:2:2 input, out = Luminance of YUV4:2:0 output 1/1: in = YUV4:2:2 input, out = YUV4:2:0 output		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CEN	YEN

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	CEN	Output Enable for Chrominance This bit is valid in 422 input mode. When CEN=0 and YEN=0, output is 422 0x0: C output disable 0x1: C output enable and 422to420 conversion enabled	RW	0
0	YEN	Output Enable for Luminance. This bit is valid in 422 input mode. When CEN=0 and YEN=0, output is 422 0x0: Y output disable 0x1: Y output enable and 422to420 conversion enabled	RW	0

3.5.3.66 RZB_I_VPS

Table 353. RZB_I_VPS

Address Offset	0x0000 0114		
Physical Address	0x5505 0514 0x5C01 0514	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - INPUT VERTICAL START REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								VPS							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	VPS	Input Vertical Position Sets the vertical start position of the input image within the global frame. It enables to crop data into the global frame. After SRC_VPS, the Vps'th line is processed as the first line in each image.	RW	0x0000

3.5.3.67 RZB_I_HPS

Table 354. RZB_I_HPS

Address Offset	0x0000 0118		
Physical Address	0x5505 0518 0x5C01 0518	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - INPUT HORIZONTAL START REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								HPS							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	HPS	Input Horizontal Position Sets the horizontal position of the first pixel for each line within the global frame. After SRC_HPS, the pixel at the VAL'th position is processed as the first pixel. This value must be even.	RW	0x0000

3.5.3.68 RZB_O_VSZ

Table 355. RZB_O_VSZ

Address Offset	0x0000 011C		
Physical Address	0x5505 051C 0x5C01 051C	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - OUTPUT VERTICAL SIZER REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								VSZ							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	VSZ	The target output size of the resized image. The number of output lines is (VSZ+1). Set 479, when 480 lines of output is required.	RW	0x0000

3.5.3.69 RZB_O_HSZ

Table 356. RZB_O_HSZ

Address Offset	0x0000 0120		
Physical Address	0x5505 0520 0x5C01 0520	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - OUTPUT HORIZONTAL SIZE REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								HSZ							HSZ_LSB

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:1	HSZ	The horizontal size of output image. The number of pixel in each line is (HSZ+1). Set 479, when 480 pixels are required. This value must be lower than the max memory line size supported by the resizer engine, except in RAW pass through mode. Note that the LSB of the 13-bit HSZ value is fixed to 1 such that the horizontal size is always even.	RW	0x000
0	HSZ_LSB	The least significant bit of HSZ is forced to 1.	R	1

3.5.3.70 RZB_V_PHS_Y

Table 357. RZB_V_PHS_Y

Address Offset	0x0000 0124		
Physical Address	0x5505 0524 0x5C01 0524	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - INITIAL LUMINANCE PHASE OF VERTICAL RESIZING PROCESS When YUV4:2:2 data are output, the phase values for luma and chroma should typically be equal, that is, RZX_V_PHS_Y= RZX_V_PHS_C. The following constraints apply when setting the initial vertical phases ABS(RZX_V_PHS_Y - RZX_V_PHS_C) RZX_V_DIF. This constraint means that at most the distance between the initial phases for luminance and chrominance is not expected to exceed the distance between two luma pixels. Note that the absolute value is used, hence, the initial luma phase can be greater than the initial chroma phase or the other way around. As a reminder, the distance between two output pixels for luma is given by RZX_V_DIF.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								Y							

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	Y	The initial value for the luma phase in vertical resizing process. This value is in U14Q8 fractional format.	RW	0x0000

3.5.3.71 RZB_V_PHS_C

Table 358. RZB_V_PHS_C

Address Offset	0x0000 0128		
Physical Address	0x5505 0528 0x5C01 0528	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - INITIAL CHROMINANCE PHASE OF VERTICAL RESIZING PROCESS When YUV4:2:2 data are output, the phase values for luma and chroma should typically be equal, that is, RZX_V_PHS_Y= RZX_V_PHS_C. The following constraints apply when setting the initial vertical phases ABS(RZX_V_PHS_Y - RZX_V_PHS_C) RZX_V_DIF. This constraint means that at most the distance between the initial phases for luminance and chrominance is not expected to exceed the distance between two luma pixels. Note that the absolute value is used, hence, the initial luma phase can be greater than the initial chroma phase or the other way around. As a reminder, the distance between two output pixels for luma is given by RZX_V_DIF.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								C							

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	C	The initial value for the chroma phase in vertical resizing process. This value is in U14Q8 fractional format.	RW	0x0000

3.5.3.72 RZB_V_DIF

Table 359. RZB_V_DIF

Address Offset	0x0000 012C		
Physical Address	0x5505 052C 0x5C01 052C	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - VERTICAL RESIZER REGISTERR		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																V															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	V	The parameter for vertical resize. The actual resizing ratio is 256/RZB_V_DIF. In normal mode: 16 = RZB_V_DIF = 4096. In down-scale mode: 256 = RZB_V_DIF = 4096.	RW	0x0000

3.5.3.73 RZB_V_TYP

Table 360. RZB_V_TYP

Address Offset	0x0000 0130		
Physical Address	0x5505 0530 0x5C01 0530	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - INTERPOLATION METHOD FOR VERTICAL RESIZING		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														C	Y

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	C	Selection of resizing method for chrominance: vertical 0x0: 4-tap cubic convolution (default) 0x1: 2-tap linear interpolation	RW	0
0	Y	Selection of resizing method for luminance: vertical 0x0: 4-tap cubic convolution (default) 0x1: 2-tap linear interpolation	RW	0

3.5.3.74 RZB_V_LPF

Table 361. RZB_V_LPF

Address Offset	0x0000 0134		
Physical Address	0x5505 0534 0x5C01 0534	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - VERTICAL LPF INTENSITY REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				C								Y			

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:6	C	The intensity parameter for chroma vertical low pass filtering.	RW	0x00
5:0	Y	The intensity parameter for luma vertical low pass filtering.	RW	0x00

3.5.3.75 RZB_H_PHS

Table 362. RZB_H_PHS

Address Offset	0x0000 0138		
Physical Address	0x5505 0538 0x5C01 0538	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - INITIAL PHASE OF HORIZONTAL RESIZING PROCESS		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PHS							

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	PHS	Initial value for the phase in horizontal resizing process, that is, the sampling position is shifted. This value is in U14Q8 fractional format. Example: If RZX_H_PHS = 128, the first output pixel is sampled at the center of the first two valid input pixels. If RZX_I_HPS=100 and RZX_H_PHS=128, the first output pixel is resampled at the center of the 100-th and the 101-st input pixels.	RW	0x0000

3.5.3.76 RZB_H_PHS_ADJ

Table 363. RZB_H_PHS_ADJ

Address Offset	0x0000 013C		
Physical Address	0x5505 053C 0x5C01 053C	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - LUMINANCE HORIZONTAL PHASE ADJUSTMENT The RZA_H_PHS_ADJ register enables to adjust the horizontal phase for the luma component when averaging is enabled (the horizontal averaging disrupts the relative sampling point between luminance and chrominance when YUV4:2:2 cosited data is input), that is, the relative phase between luma and chroma is different before and after the horizontal averager.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ADJ							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8:0	ADJ	Horizontal phase adjustment value. This value is in U9Q8 fractional format. This value is expected to be equal to zero if the averager is disabled or if input chroma is centered.	RW	0x000

3.5.3.77 RZB_H_DIF

Table 364. RZB_H_DIF

Address Offset	0x0000 0140		
Physical Address	0x5505 0540 0x5C01 0540	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - HORIZONTAL RESIZER REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								H							

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	H	The parameter for horizontal resizing process. The actual resizing ratio is 256/VAL. In normal mode 16= RSZ_RZA_H_DIF=4096 In down-scale mode 256=RSZ_RZA_H_DIF=4096	RW	0x0000

3.5.3.78 RZB_H_TYP

Table 365. RZB_H_TYP

Address Offset	0x0000 0144		
Physical Address	0x5505 0544 0x5C01 0544	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														C	Y

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	C	Selection of resizing method for chrominance: horizontal 0x0: 4-tap cubic convolution (default) 0x1: 2-tap linear interpolation	RW	0
0	Y	Selection of resizing method for luminance: horizontal 0x0: 4-tap cubic convolution (default) 0x1: 2-tap linear interpolation	RW	0

3.5.3.79 RZB_H_LPF

Table 366. RZB_H_LPF

Address Offset	0x0000 0148		
Physical Address	0x5505 0548 0x5C01 0548	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - HORIZONTAL LPF INTENSITY REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				C								Y			

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:6	C	Horizontal LPF Intensity for Chrominance	RW	0x00
5:0	Y	Selection of resizing method for Luminance in horizontal direction	RW	0x00

3.5.3.80 RZB_DWN_EN

Table 367. RZB_DWN_EN

Address Offset	0x0000 014C		
Physical Address	0x5505 054C 0x5C01 054C	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - DOWNSCALE ENABLE REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															DWN_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	DWN_EN	Resizer downscale enable 0x0: Off. Normal operation: upscale and downscale are allowed. 0x1: On. Downscale mode.	RW	0

3.5.3.81 RZB_DWN_AV

Table 368. RZB_DWN_AV

Address Offset	0x0000 0150		
Physical Address	0x5505 0550 0x5C01 0550	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								V				H			

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0000000
5:3	V	Vertical averaging size : $1/2^{(VWT+1)}$ The range goes from 1/2 to 1/256 in power of 2. 0:_DIV2 1/2 down scale 1:_DIV4 1/4 down scale 2:_DIV8 1/8 down scale 3:_DIV16 1/16 down scale 4:_DIV32 1/32 down scale 5:_DIV64 1/64 down scale 6:_DIV128 1/128 down scale 7:_DIV256 1/256 down scale	RW	0x0
2:0	H	Horizontal averaging size : $1/2^{(HWT+1)}$ The range goes from 1/2 to 1/256 in power of 2. 0:_DIV2 1/2 down scale 1:_DIV4 1/4 down scale 2:_DIV8 1/8 down scale 3:_DIV16 1/16 down scale 4:_DIV32 1/32 down scale 5:_DIV64 1/64 down scale 6:_DIV128 1/128 down scale 7:_DIV256 1/256 down scale	RW	0x0

3.5.3.82 RZB_RGB_EN

Table 369. RZB_RGB_EN

Address Offset	0x0000 0154		
Physical Address	0x5505 0554 0x5C01 0554	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - RGB OUTPUT ENABLE		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															RGB_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	RGB_EN	Enable of RGB output In pass through mode, this register must be 0. This bit can only be set to 1 when YUV4:2:2 data are output. YUV4:2:2 data output is selected when SRC_FMT1.IN420 = 0 and RZB_420.YEN = RZB_420.CEN = 0 0x0: Off (YCbCr output) 0x1: On (RGB output)	RW	0

3.5.3.83 RZB_RGB_TYP

Table 370. RZB_RGB_TYP

Address Offset	0x0000 0158		
Physical Address	0x5505 0558 0x5C01 0558	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - RGB OUTPUT CONTROL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													MSK1	MSK0	TYP

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	MSK1	Enables masking of the last 2 pixels This bit is used to mask the 2 last pixels at the image boundary which are affected by the YUV4:2:2 to YUV4:4:4 conversion. 0x0: output the last 2 pixels 0x1: mask the last 2 pixels (Resizer do not output them.)	RW	0
1	MSK0	Enables masking of the first 2 pixels This bit is used to mask the 2 first pixels at the image boundary which are affected by the YUV4:2:2 to YUV4:4:4 conversion. 0x0: output the first 2 pixels 0x1: mask the first 2 pixels (Resizer do not output them.)	RW	0
0	TYP	16bit/32bit output selection 0x0: 32-bit output: alpha + R + G + B (8 bit each) This mode comes with performance degradation. The maximum input frequency in this mode is 160 MHz. This due to the fact that the output is 4 bytes / pixel. 0x1: 16-bit output: R(5 bit) + G (6 bit) + B (5 bit)	RW	0

3.5.3.84 RZB_RGB_BLD

Table 371. RZB_RGB_BLD

Address Offset	0x0000 015C		
Physical Address	0x5505 055C 0x5C01 055C	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - RGB BLEND REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								BLD							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	BLD	The alpha value used in 32-bit RGBA output mode	RW	0x00

3.5.3.85 RZB_SDR_Y_BAD_H

Table 372. RZB_SDR_Y_BAD_H

Address Offset	0x0000 0160		
Physical Address	0x5505 0560 0x5C01 0560	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - OUTPUT MEMORY BASE ADDRESS REGISTER This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0, RGB5:6:5, RGBA. RAW: RAW data is written to this address YUV4:2:2: YUV data is written to this address YUV4:2:0: Y data is written to this address RGB5:6:5: 16-bit RGB data is written to this address RGBA: 32-bit RGBA data is written to this address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_BAD_H															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	Y_BAD_H	Memory Base Address Sets 16 upper bits of the 32-bit base address of the circular buffer in memory.	RW	0x0000

3.5.3.86 RZB_SDR_Y_BAD_L

Table 373. RZB_SDR_Y_BAD_L

Address Offset	0x0000 0164		
Physical Address	0x5505 0564 0x5C01 0564	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - OUTPUT MEMORY BASE ADDRESS REGISTER This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0, RGB5:6:5, RGBA. RAW: RAW data is written to this address YUV4:2:2: YUV data is written to this address YUV4:2:0: Y data is written to this address RGB5:6:5: 16-bit RGB data is written to this address RGBA: 32-bit RGBA data is written to this address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_BAD_L															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	Y_BAD_L	Memory Base Address Sets the 16 lower bits of the 32-bit base address of the circular buffer in memory. It is a byte address. YUV4:2:0 format (output data on 8 bits): The two least significant bits must be set to 00 when horizontal reversal mode is off. The two least significant bits must be set to 11 when horizontal reversal mode is on. RAW, RGB5:6:5 and YUV4:2:2 formats (output data on 16 bits): The two least significant bits must be set to 00 when horizontal reversal mode is off. The two least significant bits must be set to 11 when horizontal reversal mode is on. RGBA format (output data on 32 bits): The three least significant bits must be set to 000 when horizontal reversal mode is off. The three least significant bits must be set to 111 when horizontal reversal mode is on.	RW	0x0000

3.5.3.87 RZB_SDR_Y_SAD_H

Table 374. RZB_SDR_Y_SAD_H

Address Offset	0x0000 0168		
Physical Address	0x5505 0568 0x5C01 0568	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - OUTPUT MEMORY START ADDRESS REGISTER This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0, RGB5:6:5, RGBA. RAW: RAW data is written to this address YUV4:2:2: YUV data is written to this address YUV4:2:0: Y data is written to this address RGB5:6:5: 16-bit RGB data is written to this address RGBA: 32-bit RGBA data is written to this address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_SAD_H															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	Y_SAD_H	Memory Start Address Sets 16 upper bits of the 32-bit start address in memory.	RW	0x0000

3.5.3.88 RZB_SDR_Y_SAD_L

Table 375. RZB_SDR_Y_SAD_L

Address Offset	0x0000 016C		
Physical Address	0x5505 056C 0x5C01 056C	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - OUTPUT MEMORY START ADDRESS REGISTER This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0, RGB5:6:5, RGBA. RAW: RAW data is written to this address YUV4:2:2: YUV data is written to this address YUV4:2:0: Y data is written to this address RGB5:6:5: 16-bit RGB data is written to this address RGBA: 32-bit RGBA data is written to this address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_SAD_L															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	Y_SAD_L	Memory Start Address Sets the 16 lower bits of the 32-bit start address in memory. It is a byte address. For every frame, the first line of data will be written to this address. We have: SAD = BAD + (PTR_S x OFT) and PTR_S PTR_E If the first line must be written at the beginning of the circular buffer memory then SAD = BAD and PTR_S = 0. YUV4:2:0 format (output data on 8 bits): The two least significant bits must be set to 00 when horizontal reversal mode is off. The two least significant bits must be set to 11 when horizontal reversal mode is on. RAW, RGB5:6:5 and YUV4:2:2 formats (output data on 16 bits): The two least significant bits must be set to 00 when horizontal reversal mode is off. The two least significant bits must be set to 11 when horizontal reversal mode is on. RGBA format (output data on 32 bits): The three least significant bits must be set to 000 when horizontal reversal mode is off. The three least significant bits must be set to 111 when horizontal reversal mode is on.	RW	0x0000

3.5.3.89 RZB_SDR_Y_OFT

Table 376. RZB_SDR_Y_OFT

Address Offset	0x0000 0170		
Physical Address	0x5505 0570 0x5C01 0570	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - OUTPUT MEMORY OFFSET REGISTER This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0 or RGBA.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_OFT															

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0000
16:0	Y_OFT	Memory Line Offset Sets the size of each line in the circular buffer. It is expressed in bytes and unsigned. Note that OFT does not necessary corresponds to the size of a line in a frame, it can be much bigger. The line offset must be a multiple of 128 bytes (bits [6:0] of RZX_SDR_Y_OFT and RZX_SDR_C_OFT must be set to 0). Example: line 0 address = SAD line 1 address = SAD + 1 x OFT line 2 address = SAD + 2 x OFT	RW	0x00000

3.5.3.90 RZB_SDR_Y_PTR_S

Table 377. RZB_SDR_Y_PTR_S

Address Offset	0x0000 0174		
Physical Address	0x5505 0574 0x5C01 0574	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - OUTPUT MEMORY START ADDRESS REGISTER This register is used if the output data format is one of the following: RAW, YUV4;2:2, YUV4:2:0 or RGBA.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			Y_PTR_S												

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	Y_PTR_S	Start Line of Memory Pointer Sets the vertical position of the first output line in the output memory space. This value is expressed in number of lines. The hardware uses it to set up the initial value of the circular buffer. It must be set up such as PTR_S = (SAD - BAD) / OFT. This value must be set to 0 when RSZ_RZA_SDR_Y_BAD = RSZ_RZA_SDR_Y_SAD.	RW	0x0000

3.5.3.91 RZB_SDR_Y_PTR_E

Table 378. RZB_SDR_Y_PTR_E

Address Offset	0x0000 0178		
Physical Address	0x5505 0578 0x5C01 0578	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - OUTPUT MEMORY END ADDRESS REGISTER This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0 or RGBA.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			Y_PTR_E												

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	Y_PTR_E	End Line of Memory Pointer Sets the maximum number of lines to be stored in the output memory space. This value is expressed in number of lines. When the number of output lines exceeds this value, the address restarts from the first address in the output memory space (BAD).	RW	0x0000

3.5.3.92 RZB_SDR_C_BAD_H

Table 379. RZB_SDR_C_BAD_H

Address Offset	0x0000 017C		
Physical Address	0x5505 057C 0x5C01 057C	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - OUTPUT MEMORY BASE ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0. U and V data are written into this buffer.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_BAD_H															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	C_BAD_H	Memory Base Address Sets the 16 upper bits of the 32-bit base address of the circular buffer in memory.	RW	0x0000

3.5.3.93 RZB_SDR_C_BAD_L

Table 380. RZB_SDR_C_BAD_L

Address Offset	0x0000 0180		
Physical Address	0x5505 0580 0x5C01 0580	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - OUTPUT MEMORY BASE ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0. U and V data are written into this buffer.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_BAD_L															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	C_BAD_L	Memory Base Address Sets the 16 lower bits of the 32-bit base address of the circular buffer in memory. It is a byte address. YUV4:2:0 format (output data on 8 bits): The two least significant bits must be set to 00 when horizontal reversal mode is off. The two least significant bits must be set to 11 when horizontal reversal mode is on.	RW	0x0000

3.5.3.94 RZB_SDR_C_SAD_H

Table 381. RZB_SDR_C_SAD_H

Address Offset	0x0000 0184		
Physical Address	0x5505 0584 0x5C01 0584	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - OUTPUT MEMORY START ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0. U and V data are written into this buffer.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_SAD_H															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	C_SAD_H	Memory Base Address Sets the 16 upper bits of the 32-bit start address in memory.	RW	0x0000

3.5.3.95 RZB_SDR_C_SAD_L

Table 382. RZB_SDR_C_SAD_L

Address Offset	0x0000 0188		
Physical Address	0x5505 0588 0x5C01 0588	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - OUTPUT MEMORY START ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0. U and V data are written into this buffer.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_SAD_L															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	C_SAD_L	Memory Base Address Sets the 16 lower bits of the 32-bit start address in memory. It is a byte address. For every frame, the first line of data will be written to this address (C_SAD_H/C_SAD_L). We have: SAD = BAD + (PTR_S x OFT) and PTR_S PTR_E If the first line must be written at the beginning of the circular buffer memory then SAD = BAD and PTR_S = 0. YUV4:2:0 format (output data on 8 bits): The two least significant bits must be set to 00 when horizontal reversal mode is off. The two least significant bits must be set to 11 when horizontal reversal mode is on.	RW	0x0000

3.5.3.96 RZB_SDR_C_OFT

Table 383. RZB_SDR_C_OFT

Address Offset	0x0000 018C		
Physical Address	0x5505 058C 0x5C01 058C	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - OUTPUT MEMORY OFFSET REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_OFT															

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0000
16:0	C_OFT	Memory Line Offset Sets the size of each line in the circular buffer. It is expressed in bytes and unsigned. Note that OFT does not necessary corresponds to the size of a line in a frame, it can be much bigger. The line offset must be a multiple of 128 bytes (bits [6:0] of RZX_SDR_Y_OFT and RZX_SDR_C_OFT must be set to 0). Example: line 0 address = SAD line 1 address = SAD + 1 x OFT line 2 address = SAD + 2 x OFT	RW	0x00000

3.5.3.97 RZB_SDR_C_PTR_S

Table 384. RZB_SDR_C_PTR_S

Address Offset	0x0000 0190		
Physical Address	0x5505 0590 0x5C01 0590	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - OUTPUT MEMORY START ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_PTR_S															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	C_PTR_S	Start Line of Memory Pointer Sets the vertical position of the first output line in the output memory space. This value is expressed in number of lines. The hardware uses it to set up the initial value of the circular buffer. It must be set up such as PTR_S = (SAD - BAD) / OFT. This value must be set to 0 when RSZ_RZA_SDR_C_BAD = RSZ_RZA_SDR_C_SAD.	RW	0x0000

3.5.3.98 RZB_SDR_C_PTR_E

Table 385. RZB_SDR_C_PTR_E

Address Offset	0x0000 0194		
Physical Address	0x5505 0594 0x5C01 0594	**Instance**	ISS_RESIZER_CORTEX-M3 ISS_RESIZER_L3
Description	RESIZER B - OUTPUT MEMORY END ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_PTR_E															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	C_PTR_E	End Line of Memory Pointer Sets the maximum number of lines to be stored in the output memory space. This value is expressed in number of lines. When the number of output lines exceeds this value, the address restarts from the first address in the output memory space (BAD).	RW	0x0000

3.5.4 ISS IPIPE Registers

Table 386 summarizes the ISS IPIPE registers.

Table 386. ISS IPIPE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ISS_IPIPE Base Address Cortex-M3 Private Access	ISS_IPIPE Base Address L3 Interconnect
IPIPE_SRC_EN	RW	32	0x0000 0000	0x5505 0800	0x5C01 0800
IPIPE_SRC_MODE	RW	32	0x0000 0004	0x5505 0804	0x5C01 0804
IPIPE_SRC_FMT	RW	32	0x0000 0008	0x5505 0808	0x5C01 0808
IPIPE_SRC_COL	RW	32	0x0000 000C	0x5505 080C	0x5C01 080C
IPIPE_SRC_VPS	RW	32	0x0000 0010	0x5505 0810	0x5C01 0810
IPIPE_SRC_VSZ	RW	32	0x0000 0014	0x5505 0814	0x5C01 0814
IPIPE_SRC_HPS	RW	32	0x0000 0018	0x5505 0818	0x5C01 0818
IPIPE_SRC_HSZ	RW	32	0x0000 001C	0x5505 081C	0x5C01 081C
IPIPE_SEL_SBU	RW	32	0x0000 0020	0x5505 0820	0x5C01 0820
IPIPE_SRC_STA	R	32	0x0000 0024	0x5505 0824	0x5C01 0824
IPIPE_GCK_MMR	RW	32	0x0000 0028	0x5505 0828	0x5C01 0828
IPIPE_GCK_PIX	RW	32	0x0000 002C	0x5505 082C	0x5C01 082C
RESERVED	R	32	0x0000 0030	0x5505 0830	0x5C01 0830
IPIPE_DPC_LUT_EN	RW	32	0x0000 0034	0x5505 0834	0x5C01 0834
IPIPE_DPC_LUT_SEL	RW	32	0x0000 0038	0x5505 0838	0x5C01 0838
IPIPE_DPC_LUT_ADR	RW	32	0x0000 003C	0x5505 083C	0x5C01 083C
IPIPE_DPC_LUT_SIZ	RW	32	0x0000 0040	0x5505 0840	0x5C01 0840
IPIPE_DPC_OTF_EN	RW	32	0x0000 0044	0x5505 0844	0x5C01 0844
IPIPE_DPC_OTF_TYP	RW	32	0x0000 0048	0x5505 0848	0x5C01 0848
IPIPE_DPC_OTF_2_D_THR_R	RW	32	0x0000 004C	0x5505 084C	0x5C01 084C
IPIPE_DPC_OTF_2_D_THR_GR	RW	32	0x0000 0050	0x5505 0850	0x5C01 0850
IPIPE_DPC_OTF_2_D_THR_GB	RW	32	0x0000 0054	0x5505 0854	0x5C01 0854
IPIPE_DPC_OTF_2_D_THR_B	RW	32	0x0000 0058	0x5505 0858	0x5C01 0858
IPIPE_DPC_OTF_2_C_THR_R	RW	32	0x0000 005C	0x5505 085C	0x5C01 085C
IPIPE_DPC_OTF_2_C_THR_GR	RW	32	0x0000 0060	0x5505 0860	0x5C01 0860
IPIPE_DPC_OTF_2_C_THR_GB	RW	32	0x0000 0064	0x5505 0864	0x5C01 0864
IPIPE_DPC_OTF_2_C_THR_B	RW	32	0x0000 0068	0x5505 0868	0x5C01 0868
IPIPE_DPC_OTF_3_SHF	RW	32	0x0000 006C	0x5505 086C	0x5C01 086C
IPIPE_DPC_OTF_3_D_THR	RW	32	0x0000 0070	0x5505 0870	0x5C01 0870
IPIPE_DPC_OTF_3_D_SPL	RW	32	0x0000 0074	0x5505 0874	0x5C01 0874
IPIPE_DPC_OTF_3_D_MIN	RW	32	0x0000 0078	0x5505 0878	0x5C01 0878
IPIPE_DPC_OTF_3_D_MAX	RW	32	0x0000 007C	0x5505 087C	0x5C01 087C
IPIPE_DPC_OTF_3_C_THR	RW	32	0x0000 0080	0x5505 0880	0x5C01 0880
IPIPE_DPC_OTF_3_C_SLP	RW	32	0x0000 0084	0x5505 0884	0x5C01 0884
IPIPE_DPC_OTF_3_C_MIN	RW	32	0x0000 0088	0x5505 0888	0x5C01 0888
IPIPE_DPC_OTF_3_C_MAX	RW	32	0x0000 008C	0x5505 088C	0x5C01 088C
IPIPE_LSC_VOFT	RW	32	0x0000 0090	0x5505 0890	0x5C01 0890
IPIPE_LSC_VA2	RW	32	0x0000 0094	0x5505 0894	0x5C01 0894
IPIPE_LSC_VA1	RW	32	0x0000 0098	0x5505 0898	0x5C01 0898
IPIPE_LSC_VS	RW	32	0x0000 009C	0x5505 089C	0x5C01 089C
IPIPE_LSC_HOFT	RW	32	0x0000 00A0	0x5505 08A0	0x5C01 08A0
IPIPE_LSC_HA2	RW	32	0x0000 00A4	0x5505 08A4	0x5C01 08A4
IPIPE_LSC_HA1	RW	32	0x0000 00A8	0x5505 08A8	0x5C01 08A8

Table 386. ISS IPIPE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	ISS_IPIPE Base Address Cortex-M3 Private Access	ISS_IPIPE Base Address L3 Interconnect
IPIPE_LSC_HS	RW	32	0x0000 00AC	0x5505 08AC	0x5C01 08AC
IPIPE_LSC_GAN_R	RW	32	0x0000 00B0	0x5505 08B0	0x5C01 08B0
IPIPE_LSC_GAN_GR	RW	32	0x0000 00B4	0x5505 08B4	0x5C01 08B4
IPIPE_LSC_GAN_GB	RW	32	0x0000 00B8	0x5505 08B8	0x5C01 08B8
IPIPE_LSC_GAN_B	RW	32	0x0000 00BC	0x5505 08BC	0x5C01 08BC
IPIPE_LSC_OFT_R	RW	32	0x0000 00C0	0x5505 08C0	0x5C01 08C0
IPIPE_LSC_OFT_GR	RW	32	0x0000 00C4	0x5505 08C4	0x5C01 08C4
IPIPE_LSC_OFT_GB	RW	32	0x0000 00C8	0x5505 08C8	0x5C01 08C8
IPIPE_LSC_OFT_B	RW	32	0x0000 00CC	0x5505 08CC	0x5C01 08CC
IPIPE_LSC_SHF	RW	32	0x0000 00D0	0x5505 08D0	0x5C01 08D0
IPIPE_LSC_MAX	RW	32	0x0000 00D4	0x5505 08D4	0x5C01 08D4
IPIPE_D2F_1ST_EN	RW	32	0x0000 00D8	0x5505 08D8	0x5C01 08D8
IPIPE_D2F_1ST_TYP	RW	32	0x0000 00DC	0x5505 08DC	0x5C01 08DC
IPIPE_D2F_1ST_THR_00	RW	32	0x0000 00E0	0x5505 08E0	0x5C01 08E0
IPIPE_D2F_1ST_THR_01	RW	32	0x0000 00E4	0x5505 08E4	0x5C01 08E4
IPIPE_D2F_1ST_THR_02	RW	32	0x0000 00E8	0x5505 08E8	0x5C01 08E8
IPIPE_D2F_1ST_THR_03	RW	32	0x0000 00EC	0x5505 08EC	0x5C01 08EC
IPIPE_D2F_1ST_THR_04	RW	32	0x0000 00F0	0x5505 08F0	0x5C01 08F0
IPIPE_D2F_1ST_THR_05	RW	32	0x0000 00F4	0x5505 08F4	0x5C01 08F4
IPIPE_D2F_1ST_THR_06	RW	32	0x0000 00F8	0x5505 08F8	0x5C01 08F8
IPIPE_D2F_1ST_THR_07	RW	32	0x0000 00FC	0x5505 08FC	0x5C01 08FC
IPIPE_D2F_1ST_STR_00	RW	32	0x0000 0100	0x5505 0900	0x5C01 0900
IPIPE_D2F_1ST_STR_01	RW	32	0x0000 0104	0x5505 0904	0x5C01 0904
IPIPE_D2F_1ST_STR_02	RW	32	0x0000 0108	0x5505 0908	0x5C01 0908
IPIPE_D2F_1ST_STR_03	RW	32	0x0000 010C	0x5505 090C	0x5C01 090C
IPIPE_D2F_1ST_STR_04	RW	32	0x0000 0110	0x5505 0910	0x5C01 0910
IPIPE_D2F_1ST_STR_05	RW	32	0x0000 0114	0x5505 0914	0x5C01 0914
IPIPE_D2F_1ST_STR_06	RW	32	0x0000 0118	0x5505 0918	0x5C01 0918
IPIPE_D2F_1ST_STR_07	RW	32	0x0000 011C	0x5505 091C	0x5C01 091C
Reserved					
IPIPE_D2F_1ST_EDG_MIN	RW	32	0x0000 0140	0x5505 0940	0x5C01 0940
IPIPE_D2F_1ST_EDG_MAX	RW	32	0x0000 0144	0x5505 0944	0x5C01 0944
IPIPE_D2F_2ND_EN	RW	32	0x0000 0148	0x5505 0948	0x5C01 0948
IPIPE_D2F_2ND_TYP	RW	32	0x0000 014C	0x5505 094C	0x5C01 094C
IPIPE_D2F_2ND_THR00	RW	32	0x0000 0150	0x5505 0950	0x5C01 0950
IPIPE_D2F_2ND_THR01	RW	32	0x0000 0154	0x5505 0954	0x5C01 0954
IPIPE_D2F_2ND_THR02	RW	32	0x0000 0158	0x5505 0958	0x5C01 0958
IPIPE_D2F_2ND_THR03	RW	32	0x0000 015C	0x5505 095C	0x5C01 095C
IPIPE_D2F_2ND_THR04	RW	32	0x0000 0160	0x5505 0960	0x5C01 0960
IPIPE_D2F_2ND_THR05	RW	32	0x0000 0164	0x5505 0964	0x5C01 0964
IPIPE_D2F_2ND_THR06	RW	32	0x0000 0168	0x5505 0968	0x5C01 0968
IPIPE_D2F_2ND_THR07	RW	32	0x0000 016C	0x5505 096C	0x5C01 096C
IPIPE_D2F_2ND_STR_00	RW	32	0x0000 0170	0x5505 0970	0x5C01 0970
IPIPE_D2F_2ND_STR_01	RW	32	0x0000 0174	0x5505 0974	0x5C01 0974
IPIPE_D2F_2ND_STR_02	RW	32	0x0000 0178	0x5505 0978	0x5C01 0978

Table 386. ISS IPIPE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	ISS_IPIPE Base Address Cortex-M3 Private Access	ISS_IPIPE Base Address L3 Interconnect
IPIPE_D2F_2ND_STR_03	RW	32	0x0000 017C	0x5505 097C	0x5C01 097C
IPIPE_D2F_2ND_STR_04	RW	32	0x0000 0180	0x5505 0980	0x5C01 0980
IPIPE_D2F_2ND_STR_05	RW	32	0x0000 0184	0x5505 0984	0x5C01 0984
IPIPE_D2F_2ND_STR_06	RW	32	0x0000 0188	0x5505 0988	0x5C01 0988
IPIPE_D2F_2ND_STR_07	RW	32	0x0000 018C	0x5505 098C	0x5C01 098C
Reserved					
IPIPE_D2F_2ND_EDG_MIN	RW	32	0x0000 01B0	0x5505 09B0	0x5C01 09B0
IPIPE_D2F_2ND_EDG_MAX	RW	32	0x0000 01B4	0x5505 09B4	0x5C01 09B4
IPIPE_GIC_EN	RW	32	0x0000 01B8	0x5505 09B8	0x5C01 09B8
IPIPE_GIC_TYP	RW	32	0x0000 01BC	0x5505 09BC	0x5C01 09BC
IPIPE_GIC_GAN	RW	32	0x0000 01C0	0x5505 09C0	0x5C01 09C0
IPIPE_GIC_NFGAIN	RW	32	0x0000 01C4	0x5505 09C4	0x5C01 09C4
IPIPE_GIC_THR	RW	32	0x0000 01C8	0x5505 09C8	0x5C01 09C8
IPIPE_GIC_SLP	RW	32	0x0000 01CC	0x5505 09CC	0x5C01 09CC
IPIPE_WB2_OFT_R	RW	32	0x0000 01D0	0x5505 09D0	0x5C01 09D0
IPIPE_WB2_OFT_GR	RW	32	0x0000 01D4	0x5505 09D4	0x5C01 09D4
IPIPE_WB2_OFT_GB	RW	32	0x0000 01D8	0x5505 09D8	0x5C01 09D8
IPIPE_WB2_OFT_B	RW	32	0x0000 01DC	0x5505 09DC	0x5C01 09DC
IPIPE_WB2_WGN_R	RW	32	0x0000 01E0	0x5505 09E0	0x5C01 09E0
IPIPE_WB2_WGN_GR	RW	32	0x0000 01E4	0x5505 09E4	0x5C01 09E4
IPIPE_WB2_WGN_GB	RW	32	0x0000 01E8	0x5505 09E8	0x5C01 09E8
IPIPE_WB2_WGN_B	RW	32	0x0000 01EC	0x5505 09EC	0x5C01 09EC
IPIPE_CFA_MODE	RW	32	0x0000 01F0	0x5505 09F0	0x5C01 09F0
IPIPE_CFA_2DIR_HPF_THR	RW	32	0x0000 01F4	0x5505 09F4	0x5C01 09F4
IPIPE_CFA_2DIR_HPF_SLP	RW	32	0x0000 01F8	0x5505 09F8	0x5C01 09F8
IPIPE_CFA_2DIR_MIX_THR	RW	32	0x0000 01FC	0x5505 09FC	0x5C01 09FC
IPIPE_CFA_2DIR_MIX_SLP	RW	32	0x0000 0200	0x5505 0A00	0x5C01 0A00
IPIPE_CFA_2DIR_DIR_TRH	RW	32	0x0000 0204	0x5505 0A04	0x5C01 0A04
IPIPE_CFA_2DIR_DIR_SLP	RW	32	0x0000 0208	0x5505 0A08	0x5C01 0A08
IPIPE_CFA_2DIR_NDWT	RW	32	0x0000 020C	0x5505 0A0C	0x5C01 0A0C
IPIPE_CFA_MONO_HUE_FRA	RW	32	0x0000 0210	0x5505 0A10	0x5C01 0A10
IPIPE_CFA_MONO_EDG_THR	RW	32	0x0000 0214	0x5505 0A14	0x5C01 0A14
IPIPE_CFA_MONO_THR_MIN	RW	32	0x0000 0218	0x5505 0A18	0x5C01 0A18
IPIPE_CFA_MONO_THR_SLP	RW	32	0x0000 021C	0x5505 0A1C	0x5C01 0A1C
IPIPE_CFA_MONO_SLP_MIN	RW	32	0x0000 0220	0x5505 0A20	0x5C01 0A20
IPIPE_CFA_MONO_SLP_SLP	RW	32	0x0000 0224	0x5505 0A24	0x5C01 0A24
IPIPE_CFA_MONO_LPWT	RW	32	0x0000 0228	0x5505 0A28	0x5C01 0A28
IPIPE_RGB1_MUL_RR	RW	32	0x0000 022C	0x5505 0A2C	0x5C01 0A2C
IPIPE_RGB1_MUL_GR	RW	32	0x0000 0230	0x5505 0A30	0x5C01 0A30
IPIPE_RGB1_MUL_BR	RW	32	0x0000 0234	0x5505 0A34	0x5C01 0A34
IPIPE_RGB1_MUL_RG	RW	32	0x0000 0238	0x5505 0A38	0x5C01 0A38
IPIPE_RGB1_MUL_GG	RW	32	0x0000 023C	0x5505 0A3C	0x5C01 0A3C
IPIPE_RGB1_MUL_BG	RW	32	0x0000 0240	0x5505 0A40	0x5C01 0A40
IPIPE_RGB1_MUL_RB	RW	32	0x0000 0244	0x5505 0A44	0x5C01 0A44
IPIPE_RGB1_MUL_GB	RW	32	0x0000 0248	0x5505 0A48	0x5C01 0A48

Table 386. ISS IPIPE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	ISS_IPIPE Base Address Cortex-M3 Private Access	ISS_IPIPE Base Address L3 Interconnect
IPIPE_RGB1_MUL_BB	RW	32	0x0000 024C	0x5505 0A4C	0x5C01 0A4C
IPIPE_RGB1_OFT_OR	RW	32	0x0000 0250	0x5505 0A50	0x5C01 0A50
IPIPE_RGB1_OFT_OG	RW	32	0x0000 0254	0x5505 0A54	0x5C01 0A54
IPIPE_RGB1_OFT_OB	RW	32	0x0000 0258	0x5505 0A58	0x5C01 0A58
IPIPE_GMM_CFG	RW	32	0x0000 025C	0x5505 0A5C	0x5C01 0A5C
IPIPE_RGB2_MUL_RR	RW	32	0x0000 0260	0x5505 0A60	0x5C01 0A60
IPIPE_RGB2_MUL_GR	RW	32	0x0000 0264	0x5505 0A64	0x5C01 0A64
IPIPE_RGB2_MUL_BR	RW	32	0x0000 0268	0x5505 0A68	0x5C01 0A68
IPIPE_RGB2_MUL_RG	RW	32	0x0000 026C	0x5505 0A6C	0x5C01 0A6C
IPIPE_RGB2_MUL_GG	RW	32	0x0000 0270	0x5505 0A70	0x5C01 0A70
IPIPE_RGB2_MUL_BG	RW	32	0x0000 0274	0x5505 0A74	0x5C01 0A74
IPIPE_RGB2_MUL_RB	RW	32	0x0000 0278	0x5505 0A78	0x5C01 0A78
IPIPE_RGB2_MUL_GB	RW	32	0x0000 027C	0x5505 0A7C	0x5C01 0A7C
IPIPE_RGB2_MUL_BB	RW	32	0x0000 0280	0x5505 0A80	0x5C01 0A80
IPIPE_RGB2_OFT_OR	RW	32	0x0000 0284	0x5505 0A84	0x5C01 0A84
IPIPE_RGB2_OFT_OG	RW	32	0x0000 0288	0x5505 0A88	0x5C01 0A88
IPIPE_RGB2_OFT_OB	RW	32	0x0000 028C	0x5505 0A8C	0x5C01 0A8C
IPIPE_3DLUT_EN	RW	32	0x0000 0290	0x5505 0A90	0x5C01 0A90
IPIPE_YUV_ADJ	RW	32	0x0000 0294	0x5505 0A94	0x5C01 0A94
IPIPE_YUV_MUL_RY	RW	32	0x0000 0298	0x5505 0A98	0x5C01 0A98
IPIPE_YUV_MUL_GY	RW	32	0x0000 029C	0x5505 0A9C	0x5C01 0A9C
IPIPE_YUV_MUL_BY	RW	32	0x0000 02A0	0x5505 0AA0	0x5C01 0AA0
IPIPE_YUV_MUL_RCB	RW	32	0x0000 02A4	0x5505 0AA4	0x5C01 0AA4
IPIPE_YUV_MUL_GCB	RW	32	0x0000 02A8	0x5505 0AA8	0x5C01 0AA8
IPIPE_YUV_MUL_BCB	RW	32	0x0000 02AC	0x5505 0AAC	0x5C01 0AAC
IPIPE_YUV_MUL_RCR	RW	32	0x0000 02B0	0x5505 0AB0	0x5C01 0AB0
IPIPE_YUV_MUL_GCR	RW	32	0x0000 02B4	0x5505 0AB4	0x5C01 0AB4
IPIPE_YUV_MUL_BCR	RW	32	0x0000 02B8	0x5505 0AB8	0x5C01 0AB8
IPIPE_YUV_OFT_Y	RW	32	0x0000 02BC	0x5505 0ABC	0x5C01 0ABC
IPIPE_YUV_OFT_CB	RW	32	0x0000 02C0	0x5505 0AC0	0x5C01 0AC0
IPIPE_YUV_OFT_CR	RW	32	0x0000 02C4	0x5505 0AC4	0x5C01 0AC4
IPIPE_YUV_PHS	RW	32	0x0000 02C8	0x5505 0AC8	0x5C01 0AC8
IPIPE_GBCE_EN	RW	32	0x0000 02CC	0x5505 0ACC	0x5C01 0ACC
IPIPE_GBCE_TYP	RW	32	0x0000 02D0	0x5505 0AD0	0x5C01 0AD0
IPIPE_YEE_EN	RW	32	0x0000 02D4	0x5505 0AD4	0x5C01 0AD4
IPIPE_YEE_TYP	RW	32	0x0000 02D8	0x5505 0AD8	0x5C01 0AD8
IPIPE_YEE_SHF	RW	32	0x0000 02DC	0x5505 0ADC	0x5C01 0ADC
IPIPE_YEE_MUL_00	RW	32	0x0000 02E0	0x5505 0AE0	0x5C01 0AE0
IPIPE_YEE_MUL_01	RW	32	0x0000 02E4	0x5505 0AE4	0x5C01 0AE4
IPIPE_YEE_MUL_02	RW	32	0x0000 02E8	0x5505 0AE8	0x5C01 0AE8
IPIPE_YEE_MUL_10	RW	32	0x0000 02EC	0x5505 0AEC	0x5C01 0AEC
IPIPE_YEE_MUL_11	RW	32	0x0000 02F0	0x5505 0AF0	0x5C01 0AF0
IPIPE_YEE_MUL_12	RW	32	0x0000 02F4	0x5505 0AF4	0x5C01 0AF4
IPIPE_YEE_MUL_20	RW	32	0x0000 02F8	0x5505 0AF8	0x5C01 0AF8
IPIPE_YEE_MUL_21	RW	32	0x0000 02FC	0x5505 0AFC	0x5C01 0AFC

Table 386. ISS IPIPE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	ISS_IPIPE Base Address Cortex-M3 Private Access	ISS_IPIPE Base Address L3 Interconnect
IPIPE_YEE_MUL_22	RW	32	0x0000 0300	0x5505 0B00	0x5C01 0B00
IPIPE_YEE_THR	RW	32	0x0000 0304	0x5505 0B04	0x5C01 0B04
IPIPE_YEE_E_GAN	RW	32	0x0000 0308	0x5505 0B08	0x5C01 0B08
IPIPE_YEE_E_THR_1	RW	32	0x0000 030C	0x5505 0B0C	0x5C01 0B0C
IPIPE_YEE_E_THR_2	RW	32	0x0000 0310	0x5505 0B10	0x5C01 0B10
IPIPE_YEE_G_GAN	RW	32	0x0000 0314	0x5505 0B14	0x5C01 0B14
IPIPE_YEE_G_OFT	RW	32	0x0000 0318	0x5505 0B18	0x5C01 0B18
IPIPE_CAR_EN	RW	32	0x0000 031C	0x5505 0B1C	0x5C01 0B1C
IPIPE_CAR_TYP	RW	32	0x0000 0320	0x5505 0B20	0x5C01 0B20
IPIPE_CAR_SW	RW	32	0x0000 0324	0x5505 0B24	0x5C01 0B24
IPIPE_CAR_HPF_TYP	RW	32	0x0000 0328	0x5505 0B28	0x5C01 0B28
IPIPE_CAR_HPF_SHF	RW	32	0x0000 032C	0x5505 0B2C	0x5C01 0B2C
IPIPE_CAR_HPF_THR	RW	32	0x0000 0330	0x5505 0B30	0x5C01 0B30
IPIPE_CAR_GN1_GAN	RW	32	0x0000 0334	0x5505 0B34	0x5C01 0B34
IPIPE_CAR_GN1_SHF	RW	32	0x0000 0338	0x5505 0B38	0x5C01 0B38
IPIPE_CAR_GN1_MIN	RW	32	0x0000 033C	0x5505 0B3C	0x5C01 0B3C
IPIPE_CAR_GN2_GAN	RW	32	0x0000 0340	0x5505 0B40	0x5C01 0B40
IPIPE_CAR_GN2_SHF	RW	32	0x0000 0344	0x5505 0B44	0x5C01 0B44
IPIPE_CAR_GN2_MIN	RW	32	0x0000 0348	0x5505 0B48	0x5C01 0B48
IPIPE_CGS_EN	RW	32	0x0000 034C	0x5505 0B4C	0x5C01 0B4C
IPIPE_CGS_GN1_L_THR	RW	32	0x0000 0350	0x5505 0B50	0x5C01 0B50
IPIPE_CGS_GN1_L_GAIN	RW	32	0x0000 0354	0x5505 0B54	0x5C01 0B54
IPIPE_CGS_GN1_L_SHF	RW	32	0x0000 0358	0x5505 0B58	0x5C01 0B58
IPIPE_CGS_GN1_L_MIN	RW	32	0x0000 035C	0x5505 0B5C	0x5C01 0B5C
IPIPE_CGS_GN1_H_THR	RW	32	0x0000 0360	0x5505 0B60	0x5C01 0B60
IPIPE_CGS_GN1_H_GAIN	RW	32	0x0000 0364	0x5505 0B64	0x5C01 0B64
IPIPE_CGS_GN1_H_SHF	RW	32	0x0000 0368	0x5505 0B68	0x5C01 0B68
IPIPE_CGS_GN1_H_MIN	RW	32	0x0000 036C	0x5505 0B6C	0x5C01 0B6C
IPIPE_CGS_GN2_L_THR	RW	32	0x0000 0370	0x5505 0B70	0x5C01 0B70
IPIPE_CGS_GN2_L_GAIN	RW	32	0x0000 0374	0x5505 0B74	0x5C01 0B74
IPIPE_CGS_GN2_L_SHF	RW	32	0x0000 0378	0x5505 0B78	0x5C01 0B78
IPIPE_CGS_GN2_L_MIN	RW	32	0x0000 037C	0x5505 0B7C	0x5C01 0B7C
IPIPE_BOX_EN	RW	32	0x0000 0380	0x5505 0B80	0x5C01 0B80
IPIPE_BOX_MODE	RW	32	0x0000 0384	0x5505 0B84	0x5C01 0B84
IPIPE_BOX_TYP	RW	32	0x0000 0388	0x5505 0B88	0x5C01 0B88
IPIPE_BOX_SHF	RW	32	0x0000 038C	0x5505 0B8C	0x5C01 0B8C
IPIPE_BOX_SDR_SAD_H	RW	32	0x0000 0390	0x5505 0B90	0x5C01 0B90
IPIPE_BOX_SDR_SAD_L	RW	32	0x0000 0394	0x5505 0B94	0x5C01 0B94
RESERVED	R	32	0x0000 0398	0x5505 0B98	0x5C01 0B98
IPIPE_HST_EN	RW	32	0x0000 039C	0x5505 0B9C	0x5C01 0B9C
IPIPE_HST_MODE	RW	32	0x0000 03A0	0x5505 0BA0	0x5C01 0BA0
IPIPE_HST_SEL	RW	32	0x0000 03A4	0x5505 0BA4	0x5C01 0BA4
IPIPE_HST_PARA	RW	32	0x0000 03A8	0x5505 0BA8	0x5C01 0BA8
IPIPE_HST_0_VPS	RW	32	0x0000 03AC	0x5505 0BAC	0x5C01 0BAC
IPIPE_HST_0_VSZ	RW	32	0x0000 03B0	0x5505 0BB0	0x5C01 0BB0

Table 386. ISS IPIPE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	ISS_IPIPE Base Address Cortex-M3 Private Access	ISS_IPIPE Base Address L3 Interconnect
IPIPE_HST_0_HPS	RW	32	0x0000 03B4	0x5505 0BB4	0x5C01 0BB4
IPIPE_HST_0_HSZ	RW	32	0x0000 03B8	0x5505 0BB8	0x5C01 0BB8
IPIPE_HST_1_VPS	RW	32	0x0000 03BC	0x5505 0BBC	0x5C01 0BBC
IPIPE_HST_1_VSZ	RW	32	0x0000 03C0	0x5505 0BC0	0x5C01 0BC0
IPIPE_HST_1_HPS	RW	32	0x0000 03C4	0x5505 0BC4	0x5C01 0BC4
IPIPE_HST_1_HSZ	RW	32	0x0000 03C8	0x5505 0BC8	0x5C01 0BC8
IPIPE_HST_2_VPS	RW	32	0x0000 03CC	0x5505 0BCC	0x5C01 0BCC
IPIPE_HST_2_VSZ	RW	32	0x0000 03D0	0x5505 0BD0	0x5C01 0BD0
IPIPE_HST_2_HPS	RW	32	0x0000 03D4	0x5505 0BD4	0x5C01 0BD4
IPIPE_HST_2_HSZ	RW	32	0x0000 03D8	0x5505 0BD8	0x5C01 0BD8
IPIPE_HST_3_VPS	RW	32	0x0000 03DC	0x5505 0BDC	0x5C01 0BDC
IPIPE_HST_3_VSZ	RW	32	0x0000 03E0	0x5505 0BE0	0x5C01 0BE0
IPIPE_HST_3_HPS	RW	32	0x0000 03E4	0x5505 0BE4	0x5C01 0BE4
IPIPE_HST_3_HSZ	RW	32	0x0000 03E8	0x5505 0BE8	0x5C01 0BE8
IPIPE_HST_TBL	RW	32	0x0000 03EC	0x5505 0BEC	0x5C01 0BEC
IPIPE_HST_MUL_R	RW	32	0x0000 03F0	0x5505 0BF0	0x5C01 0BF0
IPIPE_HST_MUL_GR	RW	32	0x0000 03F4	0x5505 0BF4	0x5C01 0BF4
IPIPE_HST_MUL_GB	RW	32	0x0000 03F8	0x5505 0BF8	0x5C01 0BF8
IPIPE_HST_MUL_B	RW	32	0x0000 03FC	0x5505 0BFC	0x5C01 0BFC
IPIPE_BSC_EN	RW	32	0x0000 0400	0x5505 0C00	0x5C01 0C00
IPIPE_BSC_MODE	RW	32	0x0000 0404	0x5505 0C04	0x5C01 0C04
IPIPE_BSC_TYP	RW	32	0x0000 0408	0x5505 0C08	0x5C01 0C08
IPIPE_BSC_ROW_VCT	RW	32	0x0000 040C	0x5505 0C0C	0x5C01 0C0C
IPIPE_BSC_ROW_SHF	RW	32	0x0000 0410	0x5505 0C10	0x5C01 0C10
IPIPE_BSC_ROW_VPOS	RW	32	0x0000 0414	0x5505 0C14	0x5C01 0C14
IPIPE_BSC_ROW_VNUM	RW	32	0x0000 0418	0x5505 0C18	0x5C01 0C18
IPIPE_BSC_ROW_VSKIP	RW	32	0x0000 041C	0x5505 0C1C	0x5C01 0C1C
IPIPE_BSC_ROW_HPOS	RW	32	0x0000 0420	0x5505 0C20	0x5C01 0C20
IPIPE_BSC_ROW_HNUM	RW	32	0x0000 0424	0x5505 0C24	0x5C01 0C24
IPIPE_BSC_ROW_HSKIP	RW	32	0x0000 0428	0x5505 0C28	0x5C01 0C28
IPIPE_BSC_COL_VCT	RW	32	0x0000 042C	0x5505 0C2C	0x5C01 0C2C
IPIPE_BSC_COL_SHF	RW	32	0x0000 0430	0x5505 0C30	0x5C01 0C30
IPIPE_BSC_COL_VPOS	RW	32	0x0000 0434	0x5505 0C34	0x5C01 0C34
IPIPE_BSC_COL_VNUM	RW	32	0x0000 0438	0x5505 0C38	0x5C01 0C38
IPIPE_BSC_COL_VSKIP	RW	32	0x0000 043C	0x5505 0C3C	0x5C01 0C3C
IPIPE_BSC_COL_HPOS	RW	32	0x0000 0440	0x5505 0C40	0x5C01 0C40
IPIPE_BSC_COL_HNUM	RW	32	0x0000 0444	0x5505 0C44	0x5C01 0C44
IPIPE_BSC_COL_HSKIP	RW	32	0x0000 0448	0x5505 0C48	0x5C01 0C48

3.5.4.1 IPIPE_SRC_EN

Table 387. IPIPE_SRC_EN

Address Offset	0x0000 0000		
Physical Address	0x5505 0800 0x5C01 0800	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	This register is not shadowed		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	EN	The start flag of the IPIPE module. When EN is 1, the IPIPE module starts a processing from the next rising edge of the VD. If the processing mode of the IPIPE module is one shot, the EN is cleared to 0 immediately after the processing has started. 0x0: waiting 0x1: start/busy	RW	0

3.5.4.2 IPIPE_SRC_MODE

Table 388. IPIPE_SRC_MODE

Address Offset	0x0000 0004		
Physical Address	0x5505 0804 0x5C01 0804	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														WRT	OST

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1	WRT	The mode selection of the ipipeif_wrt which is an input port of the IPIPE module. If WRT is 0, the IPIPE module does not use the ipipeif_wrt. Else the IPIPE module uses it. 0x0: Disable 0x1: Enable	RW	0
0	OST	The processing mode selection of the IPIPE module. Value 0 indicates the mode of free run, value 1 indicates the mode of one shot. 0x0: Free run 0x1: One shot	RW	0

3.5.4.3 IPIPE_SRC_FMT

Table 389. IPIPE_SRC_FMT

Address Offset	0x0000 0008		
Physical Address	0x5505 0808 0x5C01 0808	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														FMT	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1:0	FMT	IPIPE module data path selection 0x0: IN: RAW BAYER OUT: YUV4:2:2 Note that the IPIPE YUV4:2:2 output goes to the RESIZER module where it can be further be converted in YUV4:2:0 or RGB format. 0x1: IN: RAW BAYER OUT: RAW BAYER The data are output after the White Balance module. It enables to bypass a large part of the IPIPE module. 0x2: IN: RAW BAYER OUT: DISABLED The data are only going to BOXCAR and HISTOGRAM modules. 0x3: IN: YUV4:2:2 OUT: YUV4:2:2 Note that the IPIPE YUV4:2:2 output goes to the RESIZER module where it can be further be converted in YUV4:2:0 or RGB format.	RW	0x0

3.5.4.4 IPIPE_SRC_COL

Table 390. IPIPE_SRC_COL

Address Offset	0x0000 000C		
Physical Address	0x5505 080C 0x5C01 080C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								OO		OE		EO		EE	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:6	OO	The color pattern of the odd line and odd pixel. This parameter is valid when IPIPE_SRC[FMT] is 0,1,2. 0x0: R 0x1: Gr 0x2: Gb 0x3: B	RW	0x3
5:4	OE	The color pattern of the odd line and even pixel. This parameter is valid when IPIPE_SRC[FMT] is 0,1,2. 0x0: R 0x1: Gr 0x2: Gb 0x3: B	RW	0x2
3:2	EO	The color pattern of the even line and odd pixel. This parameter is valid when IPIPE_SRC[FMT] is 0,1,2. 0x0: R 0x1: Gr 0x2: Gb 0x3: B	RW	0x1
1:0	EE	The color pattern of the even line and even pixel. This parameter is valid when IPIPE_SRC[FMT] is 0,1,2. 0x0: R 0x1: Gr 0x2: Gb 0x3: B	RW	0x0

3.5.4.5 IPIPE_SRC_VPS

Table 391. IPIPE_SRC_VPS

Address Offset	0x0000 0010		
Physical Address	0x5505 0810 0x5C01 0810	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	VAL	The vertical position of the global frame from the rising edge of the VD. The IPIPE module will start an image processing from VAL line.	RW	0x0000

3.5.4.6 IPIPE_SRC_VSZ

Table 392. IPIPE_SRC_VSZ

Address Offset	0x0000 0014		
Physical Address	0x5505 0814 0x5C01 0814	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			VAL												

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	The vertical size of the processing area. The VAL0 can not be written. The IPIPE module will process (VAL+1) lines.	RW	0x0000

3.5.4.7 IPIPE_SRC_HPS

Table 393. IPIPE_SRC_HPS

Address Offset	0x0000 0018		
Physical Address	0x5505 0818 0x5C01 0818	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	VAL	The horizontal position of the global frame from the rising edge of the HD. The IPIPE module will start an image processing from VAL clock.	RW	0x0000

3.5.4.8 IPIPE_SRC_HSZ

Table 394. IPIPE_SRC_HSZ

Address Offset	0x0000 001C		
Physical Address	0x5505 081C 0x5C01 081C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VAL															VAL_0

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:1	VAL	The horizontal size of the processing area. The VAL0 is fixed. The IPIPE module processes (VAL+1) clocks.	RW	0x000
0	VAL_0	This is the LSB of the VAL[12:0]. This bit is read only.	R	1

3.5.4.9 IPIPE_SEL_SBU

Table 395. IPIPE_SEL_SBU

Address Offset	0x0000 0020		
Physical Address	0x5505 0820 0x5C01 0820	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															EDOF

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	EDOF	EDOF port selection This bit must not be enabled since the EDOF module is not implemented. This is a provision for a future revision of the IP. 0x0: Not used 0x1: Used	RW	0

3.5.4.10 IPIPE_SRC_STA

Table 396. IPIPE_SRC_STA

Address Offset	0x0000 0024		
Physical Address	0x5505 0824 0x5C01 0824	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	IPIPE STATUS REGISTER		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED											VAL4	VAL3	VAL2	VAL1	VAL0

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4	VAL4	Status of Histogram Process (busy status).	R	0
3	VAL3	Status of Histogram bank select.	R	0
2	VAL2	Status of BSC process (busy status).	R	0
1	VAL1	Status of Boxcar process (busy status).	R	0
0	VAL0	Status of Boxcar process (error status). This bit will be triggered when an overflow happens while transferring the boxcar data to memory. Instead of polling for this register, it is preferable to use the IPIPE_BOXCAR_OVF interrupt. Overflow errors are non recoverable at ISP level and require a software reset at ISS level.	R	0

3.5.4.11 IPIPE_GCK_MMR

Table 397. IPIPE_GCK_MMR

Address Offset	0x0000 0028		
Physical Address	0x5505 0828 0x5C01 0828	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															REG

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	REG	The on/off selection of the clk_arm_g0 that is used for some ARM register access. 0x0: Off 0x1: On	RW	0

3.5.4.12 IPIPE_GCK_PIX

Table 398. IPIPE_GCK_PIX

Address Offset	0x0000 002C		
Physical Address	0x5505 082C 0x5C01 082C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	This register is not shadowed		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED												G3	G2	G1	G0

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:4	RESERVED		R	0x000
3	G3	The on/off selection of the clk_pix_g3 that is used for the IPIPE processes of EE and "CAR". 0x0: Disable 0x1: Enable	RW	0
2	G2	The on/off selection of the clk_pix_g2 that is used for the IPIPE processes of CFA to "422", "Histogram(YCbCr input)", and "Boundary Signal Calculator". 0x0: Disable 0x1: Enable	RW	0
1	G1	The on/off selection of the clk_pix_g1 that is used for the IPIPE processes of "DefectCorrection" to "WhiteBalance", and "Histogram(RAW input)". 0x0: Disable 0x1: Enable	RW	0
0	G0	The on/off selection of the clk_pix_g0 that is used for the IPIPE processing of "Boxcar". 0x0: Disable 0x1: Enable	RW	0

3.5.4.13 IPIPE_DPC_LUT_EN

Table 399. IPIPE_DPC_LUT_EN

Address Offset	0x0000 0034		
Physical Address	0x5505 0834 0x5C01 0834	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	EN	Enable of LUT defect pixel correction. 0x0: Off 0x1: On	RW	0

3.5.4.14 IPIPE_DPC_LUT_SEL

Table 400. IPIPE_DPC_LUT_SEL

Address Offset	0x0000 0038		
Physical Address	0x5505 0838 0x5C01 0838	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														TBL	DOT

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1	TBL	LUT table type selection. 0x0: Up to 1024 entries. (use IPIPE_DPC_LUT_SIZ) 0x1: infinity number of entries. (not use IPIPE_DPC_LUT_SIZ)	RW	0
0	DOT	Replace dot selection on processing method 0. 0x0: Replace with black dot 0x1: Replace with white dot	RW	0

3.5.4.15 IPIPE_DPC_LUT_ADR

Table 401. IPIPE_DPC_LUT_ADR

Address Offset	0x0000 003C		
Physical Address	0x5505 083C 0x5C01 083C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								ADR							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	ADR	The address of the first valid data in look-up-table	RW	0x000

3.5.4.16 IPIPE_DPC_LUT_SIZ

Table 402. IPIPE_DPC_LUT_SIZ

Address Offset	0x0000 0040		
Physical Address	0x5505 0840 0x5C01 0840	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								SIZ							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	SIZ	The number of valid data in look-up-table. (SIZ+1)	RW	0x000

3.5.4.17 IPIPE_DPC_OTF_EN

Table 403. IPIPE_DPC_OTF_EN

Address Offset	0x0000 0044		
Physical Address	0x5505 0844 0x5C01 0844	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Enable of adaptive defect pixel correction module.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	EN	0x0: Off 0x1: On	RW	0

3.5.4.18 IPIPE_DPC_OTF_TYP

Table 404. IPIPE_DPC_OTF_TYP

Address Offset	0x0000 0048		
Physical Address	0x5505 0848 0x5C01 0848	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	To select MinMax Algorithm following values are set ALG: 0 TYP: 0 IPIPE_DPC_OTF_2_D_THR_x: 0 IPIPE_DPC_OTF_2_C_THR_x: maximum value		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														TYP	ALG

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1	TYP	0x0: with MAX1/MIN1. 0x1: with MAX2/MIN2.	RW	0
0	ALG	0x0: MinMax2 (DPC2.0) 0x1: MinMax3 (DPC3.0)	RW	0

3.5.4.19 IPIPE_DPC_OTF_2_D_THR_R

Table 405. IPIPE_DPC_OTF_2_D_THR_R

Address Offset	0x0000 004C		
Physical Address	0x5505 084C 0x5C01 084C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Defect detection threshold value for each color (DPC2.0)	RW	0x000

3.5.4.20 IPIPE_DPC_OTF_2_D_THR_GR

Table 406. IPIPE_DPC_OTF_2_D_THR_GR

Address Offset	0x0000 0050		
Physical Address	0x5505 0850 0x5C01 0850	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Defect detection threshold value for each color (DPC2.0)	RW	0x000

3.5.4.21 IPIPE_DPC_OTF_2_D_THR_GB

Table 407. IPIPE_DPC_OTF_2_D_THR_GB

Address Offset	0x0000 0054		
Physical Address	0x5505 0854 0x5C01 0854	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Defect detection threshold value for each color (DPC2.0)	RW	0x000

3.5.4.22 IPIPE_DPC_OTF_2_D_THR_B

Table 408. IPIPE_DPC_OTF_2_D_THR_B

Address Offset	0x0000 0058		
Physical Address	0x5505 0858 0x5C01 0858	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Defect detection threshold value for each color (DPC2.0)	RW	0x000

3.5.4.23 IPIPE_DPC_OTF_2_C_THR_R

Table 409. IPIPE_DPC_OTF_2_C_THR_R

Address Offset	0x0000 005C		
Physical Address	0x5505 085C 0x5C01 085C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Defect correction threshold value for each color (DPC2.0)	RW	0x000

3.5.4.24 IPIPE_DPC_OTF_2_C_THR_GR

Table 410. IPIPE_DPC_OTF_2_C_THR_GR

Address Offset	0x0000 0060		
Physical Address	0x5505 0860 0x5C01 0860	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Defect correction threshold value for each color (DPC2.0)	RW	0x000

3.5.4.25 IPIPE_DPC_OTF_2_C_THR_GB

Table 411. IPIPE_DPC_OTF_2_C_THR_GB

Address Offset	0x0000 0064		
Physical Address	0x5505 0864 0x5C01 0864	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Defect correction threshold value for each color (DPC2.0)	RW	0x000

3.5.4.26 IPIPE_DPC_OTF_2_C_THR_B

Table 412. IPIPE_DPC_OTF_2_C_THR_B

Address Offset	0x0000 0068		
Physical Address	0x5505 0868 0x5C01 0868	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Defect correction threshold value for each color (DPC2.0)	RW	0x000

3.5.4.27 IPIPE_DPC_OTF_3_SHF

Table 413. IPIPE_DPC_OTF_3_SHF

Address Offset	0x0000 006C		
Physical Address	0x5505 086C 0x5C01 086C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														SHF	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1:0	SHF		RW	0x0

3.5.4.28 IPIPE_DPC_OTF_3_D_THR

Table 414. IPIPE_DPC_OTF_3_D_THR

Address Offset	0x0000 0070		
Physical Address	0x5505 0870 0x5C01 0870	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL										VAL_RESERVED	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:6	VAL	DPC3 Defect detection threshold 12-bit value. Bits [5:0] cannot be written.	RW	0x00
5:0	VAL_RESERVED	DPC3 Defect detection threshold 12-bit value. Bits [5:0] cannot be written.	R	0x00

3.5.4.29 IPIPE_DPC_OTF_3_D_SPL

Table 415. IPIPE_DPC_OTF_3_D_SPL

Address Offset	0x0000 0074		
Physical Address	0x5505 0874 0x5C01 0874	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:6	RESERVED		R	0x000
5:0	VAL	Defect detection threshold slope for DPC3.0	RW	0x00

3.5.4.30 IPIPE_DPC_OTF_3_D_MIN

Table 416. IPIPE_DPC_OTF_3_D_MIN

Address Offset	0x0000 0078		
Physical Address	0x5505 0878 0x5C01 0878	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL		RW	0x000

3.5.4.31 IPIPE_DPC_OTF_3_D_MAX

Table 417. IPIPE_DPC_OTF_3_D_MAX

Address Offset	0x0000 007C		
Physical Address	0x5505 087C 0x5C01 087C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Defect detection threshold MAX for DPC3.0	RW	0x000

3.5.4.32 IPIPE_DPC_OTF_3_C_THR

Table 418. IPIPE_DPC_OTF_3_C_THR

Address Offset	0x0000 0080		
Physical Address	0x5505 0880 0x5C01 0880	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL										VAL_RESERVED	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:6	VAL	Defect correction threshold for DPC3.0	RW	0x00
5:0	VAL_RESERVED	Defect correction threshold for DPC3.0 Bits VAL[5-0] cannot be written.	RW	0x00

3.5.4.33 IPIPE_DPC_OTF_3_C_SLP

Table 419. IPIPE_DPC_OTF_3_C_SLP

Address Offset	0x0000 0084		
Physical Address	0x5505 0884 0x5C01 0884	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL_RESERVED				VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:6	VAL_RESERVED	Defect correction threshold slope for DPC3.0 VAL[11-6] cannot be written.	R	0x00
5:0	VAL	Defect correction threshold slope for DPC3.0	RW	0x00

3.5.4.34 IPIPE_DPC_OTF_3_C_MIN

Table 420. IPIPE_DPC_OTF_3_C_MIN

Address Offset	0x0000 0088		
Physical Address	0x5505 0888 0x5C01 0888	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Defect correction threshold MIN for DPC3.0	RW	0x000

3.5.4.35 IPIPE_DPC_OTF_3_C_MAX

Table 421. IPIPE_DPC_OTF_3_C_MAX

Address Offset	0x0000 008C		
Physical Address	0x5505 088C 0x5C01 088C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Defect correction threshold MAX for DPC3.0	RW	0x000

3.5.4.36 IPIPE_LSC_VOFT

Table 422. IPIPE_LSC_VOFT

Address Offset	0x0000 0090		
Physical Address	0x5505 0890 0x5C01 0890	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	LSC VOFT		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				LSC_VOFT											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	LSC_VOFT		RW	0x0000

3.5.4.37 IPIPE_LSC_VA2

Table 423. IPIPE_LSC_VA2

Address Offset	0x0000 0094		
Physical Address	0x5505 0894 0x5C01 0894	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	LSC VA2	RW	0x0000

3.5.4.38 IPIPE_LSC_VA1

Table 424. IPIPE_LSC_VA1

Address Offset	0x0000 0098		
Physical Address	0x5505 0898 0x5C01 0898	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	LSC VA1	RW	0x0000

3.5.4.39 IPIPE_LSC_VS

Table 425. IPIPE_LSC_VS

Address Offset	0x0000 009C		
Physical Address	0x5505 089C 0x5C01 089C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VS2				VS1			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:4	VS2	LSC VS2	RW	0x0
3:0	VS1	LSC VS1	RW	0x0

3.5.4.40 IPIPE_LSC_HOFT

Table 426. IPIPE_LSC_HOFT

Address Offset	0x0000 00A0		
Physical Address	0x5505 08A0 0x5C01 08A0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

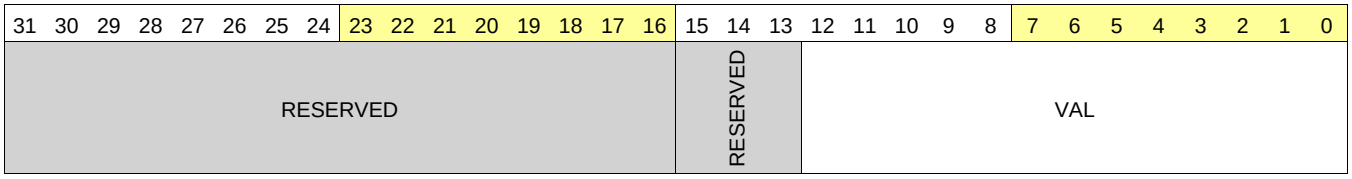
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			VAL												

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	LSC HOFT	RW	0x0000

3.5.4.41 IPIPE_LSC_HA2

Table 427. IPIPE_LSC_HA2

Address Offset	0x0000 00A4		
Physical Address	0x5505 08A4 0x5C01 08A4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

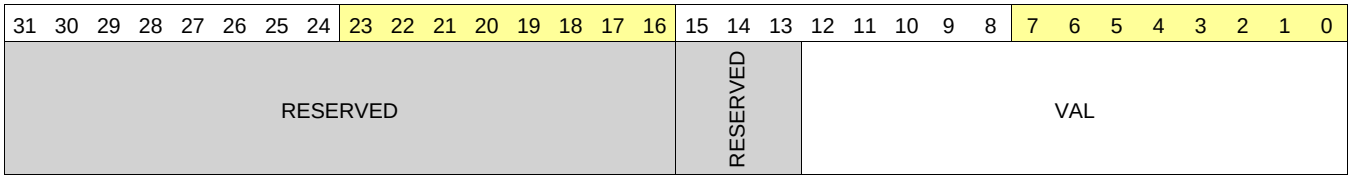


Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	LSC HA2	RW	0x0000

3.5.4.42 IPIPE_LSC_HA1

Table 428. IPIPE_LSC_HA1

Address Offset	0x0000 00A8		
Physical Address	0x5505 08A8 0x5C01 08A8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	LSC HA1	RW	0x0000

3.5.4.43 IPIPE_LSC_HS

Table 429. IPIPE_LSC_HS

Address Offset	0x0000 00AC		
Physical Address	0x5505 08AC 0x5C01 08AC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								HS2				HS1			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:4	HS2	LSC HS2	RW	0x0
3:0	HS1	LSC HS1	RW	0x0

3.5.4.44 IPIPE_LSC_GAN_R

Table 430. IPIPE_LSC_GAN_R

Address Offset	0x0000 00B0		
Physical Address	0x5505 08B0 0x5C01 08B0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	GAN R	RW	0x00

3.5.4.45 IPIPE_LSC_GAN_GR

Table 431. IPIPE_LSC_GAN_GR

Address Offset	0x0000 00B4		
Physical Address	0x5505 08B4 0x5C01 08B4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	GAN GR	RW	0x00

3.5.4.46 IPIPE_LSC_GAN_GB

Table 432. IPIPE_LSC_GAN_GB

Address Offset	0x0000 00B8		
Physical Address	0x5505 08B8 0x5C01 08B8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	GAN GB	RW	0x00

3.5.4.47 IPIPE_LSC_GAN_B

Table 433. IPIPE_LSC_GAN_B

Address Offset	0x0000 00BC		
Physical Address	0x5505 08BC 0x5C01 08BC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	GAN B	RW	0x00

3.5.4.48 IPIPE_LSC_OFT_R

Table 434. IPIPE_LSC_OFT_R

Address Offset	0x0000 00C0		
Physical Address	0x5505 08C0 0x5C01 08C0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	LSC OFT R	RW	0x00

3.5.4.49 IPIPE_LSC_OFT_GR

Table 435. IPIPE_LSC_OFT_GR

Address Offset	0x0000 00C4		
Physical Address	0x5505 08C4 0x5C01 08C4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	LSC OFT GR	RW	0x00

3.5.4.50 IPIPE_LSC_OFT_GB

Table 436. IPIPE_LSC_OFT_GB

Address Offset	0x0000 00C8		
Physical Address	0x5505 08C8 0x5C01 08C8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	LSC OFT GB	RW	0x00

3.5.4.51 IPIPE_LSC_OFT_B

Table 437. IPIPE_LSC_OFT_B

Address Offset	0x0000 00CC		
Physical Address	0x5505 08CC 0x5C01 08CC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	LSC OFT B	RW	0x00

3.5.4.52 IPIPE_LSC_SHF

Table 438. IPIPE_LSC_SHF

Address Offset	0x0000 00D0		
Physical Address	0x5505 08D0 0x5C01 08D0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED												VAL			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:4	RESERVED		R	0x000
3:0	VAL	LSC SHF	RW	0x0

3.5.4.53 IPIPE_LSC_MAX

Table 439. IPIPE_LSC_MAX

Address Offset	0x0000 00D4		
Physical Address	0x5505 08D4 0x5C01 08D4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:9	RESERVED		R	0x00
8:0	VAL	LSC MAX	RW	0x000

3.5.4.54 IPIPE_D2F_1ST_EN

Table 440. IPIPE_D2F_1ST_EN

Address Offset	0x0000 00D8		
Physical Address	0x5505 08D8 0x5C01 08D8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	EN	Enable of noise filter-1 module. 0x0: off 0x1: on	RW	0

3.5.4.55 IPIPE_D2F_1ST_TYP

Table 441. IPIPE_D2F_1ST_TYP

Address Offset	0x0000 00DC		
Physical Address	0x5505 08DC 0x5C01 08DC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED						SEL	LSC	TYP	SHF			SPR			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9	SEL	Select SPR value source 0x0: use single SPR value 0x1: use LUT SPR values	RW	0
8	LSC	Apply LSC gain to threshold values 0x0: off 0x1: on	RW	0
7	TYP	The sampling method of green pixels. 0x0: Box (same as R or B) 0x1: Diamond mode	RW	0
6:5	SHF	The d value (down shift value) in look-up-table reference address.	RW	0x0
4:0	SPR	The SP value ("spread" value) in noise filter-1 algorithm.	RW	0x00

3.5.4.56 IPIPE_D2F_1ST_THR_00

Table 442. IPIPE_D2F_1ST_THR_00

Address Offset	0x0000 00E0		
Physical Address	0x5505 08E0 0x5C01 08E0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED						VAL									

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Threshold values in noise filter-1 algorithm.	RW	0x000

3.5.4.57 IPIPE_D2F_1ST_THR_01

Table 443. IPIPE_D2F_1ST_THR_01

Address Offset	0x0000 00E4		
Physical Address	0x5505 08E4 0x5C01 08E4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Threshold values in noise filter-1 algorithm.	RW	0x000

3.5.4.58 IPIPE_D2F_1ST_THR_02

Table 444. IPIPE_D2F_1ST_THR_02

Address Offset	0x0000 00E8		
Physical Address	0x5505 08E8 0x5C01 08E8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Threshold values in noise filter-1 algorithm.	RW	0x000

3.5.4.59 IPIPE_D2F_1ST_THR_03

Table 445. IPIPE_D2F_1ST_THR_03

Address Offset	0x0000 00EC		
Physical Address	0x5505 08EC 0x5C01 08EC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Threshold values in noise filter-1 algorithm.	RW	0x000

3.5.4.60 IPIPE_D2F_1ST_THR_04

Table 446. IPIPE_D2F_1ST_THR_04

Address Offset	0x0000 00F0		
Physical Address	0x5505 08F0 0x5C01 08F0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Threshold values in noise filter-1 algorithm.	RW	0x000

3.5.4.61 IPIPE_D2F_1ST_THR_05

Table 447. IPIPE_D2F_1ST_THR_05

Address Offset	0x0000 00F4		
Physical Address	0x5505 08F4 0x5C01 08F4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Threshold values in noise filter-1 algorithm.	RW	0x000

3.5.4.62 IPIPE_D2F_1ST_THR_06

Table 448. IPIPE_D2F_1ST_THR_06

Address Offset	0x0000 00F8		
Physical Address	0x5505 08F8 0x5C01 08F8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Threshold values in noise filter-1 algorithm.	RW	0x000

3.5.4.63 IPIPE_D2F_1ST_THR_07

Table 449. IPIPE_D2F_1ST_THR_07

Address Offset	0x0000 00FC		
Physical Address	0x5505 08FC 0x5C01 08FC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Threshold values in noise filter-1 algorithm.	RW	0x000

3.5.4.64 IPIPE_D2F_1ST_STR_00

Table 450. IPIPE_D2F_1ST_STR_00

Address Offset	0x0000 0100		
Physical Address	0x5505 0900 0x5C01 0900	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED													VAL		

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4:0	VAL	Noise filter-1 intensity values (STR).	RW	0x00

3.5.4.65 IPIPE_D2F_1ST_STR_01

Table 451. IPIPE_D2F_1ST_STR_01

Address Offset	0x0000 0104		
Physical Address	0x5505 0904 0x5C01 0904	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4:0	VAL	Noise filter-1 intensity values (STR).	RW	0x00

3.5.4.66 IPIPE_D2F_1ST_STR_02

Table 452. IPIPE_D2F_1ST_STR_02

Address Offset	0x0000 0108		
Physical Address	0x5505 0908 0x5C01 0908	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4:0	VAL	Noise filter-1 intensity values (STR).	RW	0x00

3.5.4.67 IPIPE_D2F_1ST_STR_03

Table 453. IPIPE_D2F_1ST_STR_03

Address Offset	0x0000 010C		
Physical Address	0x5505 090C 0x5C01 090C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4:0	VAL	Noise filter-1 intensity values (STR).	RW	0x00

3.5.4.68 IPIPE_D2F_1ST_STR_04

Table 454. IPIPE_D2F_1ST_STR_04

Address Offset	0x0000 0110		
Physical Address	0x5505 0910 0x5C01 0910	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4:0	VAL	Noise filter-1 intensity values (STR).	RW	0x00

3.5.4.69 IPIPE_D2F_1ST_STR_05

Table 455. IPIPE_D2F_1ST_STR_05

Address Offset	0x0000 0114		
Physical Address	0x5505 0914 0x5C01 0914	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4:0	VAL	Noise filter-1 intensity values (STR).	RW	0x00

3.5.4.70 IPIPE_D2F_1ST_STR_06

Table 456. IPIPE_D2F_1ST_STR_06

Address Offset	0x0000 0118		
Physical Address	0x5505 0918 0x5C01 0918	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4:0	VAL	Noise filter-1 intensity values (STR).	RW	0x00

3.5.4.71 IPIPE_D2F_1ST_STR_07

Table 457. IPIPE_D2F_1ST_STR_07

Address Offset	0x0000 011C		
Physical Address	0x5505 091C 0x5C01 091C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4:0	VAL	Noise filter-1 intensity values (STR).	RW	0x00

3.5.4.72 IPIPE_D2F_1ST_EDG_MIN

Table 458. IPIPE_D2F_1ST_EDG_MIN

Address Offset	0x0000 0140		
Physical Address	0x5505 0940 0x5C01 0940	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED					VAL										

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	Noise filter-1 edge detection MIN	RW	0x000

3.5.4.73 IPIPE_D2F_1ST_EDG_MAX

Table 459. IPIPE_D2F_1ST_EDG_MAX

Address Offset	0x0000 0144		
Physical Address	0x5505 0944 0x5C01 0944	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	Noise filter-1 edge detection MAX	RW	0x000

3.5.4.74 IPIPE_D2F_2ND_EN

Table 460. IPIPE_D2F_2ND_EN

Address Offset	0x0000 0148		
Physical Address	0x5505 0948 0x5C01 0948	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	EN	Enable of noise filter-2 module. 0x0: off 0x1: on	RW	0

3.5.4.75 IPIPE_D2F_2ND_TYP

Table 461. IPIPE_D2F_2ND_TYP

Address Offset	0x0000 014C		
Physical Address	0x5505 094C 0x5C01 094C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED						SEL	LSC	TYP	SHF			SPR			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9	SEL	Select SPR value source 0x0: use single SPR value 0x1: use LUT SPR values	RW	0
8	LSC	Apply LSC gain to threshold values 0x0: off 0x1: on	RW	0
7	TYP	The sampling method of green pixels. 0x0: Box (Same as R or B) 0x1: Diamond mode	RW	0
6:5	SHF	The d value (down shift value) in look-up-table reference address.	RW	0x0
4:0	SPR	The SP value ("spread" value) in noise filter-2 algorithm.	RW	0x00

3.5.4.76 IPIPE_D2F_2ND_THR00

Table 462. IPIPE_D2F_2ND_THR00

Address Offset	0x0000 0150		
Physical Address	0x5505 0950 0x5C01 0950	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Threshold values in noise filter-2 algorithm.	RW	0x000

3.5.4.77 IPIPE_D2F_2ND_THR01

Table 463. IPIPE_D2F_2ND_THR01

Address Offset	0x0000 0154		
Physical Address	0x5505 0954 0x5C01 0954	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Threshold values in noise filter-2 algorithm.	RW	0x000

3.5.4.78 IPIPE_D2F_2ND_THR02

Table 464. IPIPE_D2F_2ND_THR02

Address Offset	0x0000 0158		
Physical Address	0x5505 0958 0x5C01 0958	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Threshold values in noise filter-2 algorithm.	RW	0x000

3.5.4.79 IPIPE_D2F_2ND_THR03

Table 465. IPIPE_D2F_2ND_THR03

Address Offset	0x0000 015C		
Physical Address	0x5505 095C 0x5C01 095C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Threshold values in noise filter-2 algorithm.	RW	0x000

3.5.4.80 IPIPE_D2F_2ND_THR04

Table 466. IPIPE_D2F_2ND_THR04

Address Offset	0x0000 0160		
Physical Address	0x5505 0960 0x5C01 0960	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Threshold values in noise filter-2 algorithm.	RW	0x000

3.5.4.81 IPIPE_D2F_2ND_THR05

Table 467. IPIPE_D2F_2ND_THR05

Address Offset	0x0000 0164		
Physical Address	0x5505 0964 0x5C01 0964	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Threshold values in noise filter-2 algorithm.	RW	0x000

3.5.4.82 IPIPE_D2F_2ND_THR06

Table 468. IPIPE_D2F_2ND_THR06

Address Offset	0x0000 0168		
Physical Address	0x5505 0968 0x5C01 0968	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Threshold values in noise filter-2 algorithm.	RW	0x000

3.5.4.83 IPIPE_D2F_2ND_THR07

Table 469. IPIPE_D2F_2ND_THR07

Address Offset	0x0000 016C		
Physical Address	0x5505 096C 0x5C01 096C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Threshold values in noise filter-2 algorithm.	RW	0x000

3.5.4.84 IPIPE_D2F_2ND_STR_00

Table 470. IPIPE_D2F_2ND_STR_00

Address Offset	0x0000 0170		
Physical Address	0x5505 0970 0x5C01 0970	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4:0	VAL	Noise filter-2 intensity values (STR).	RW	0x00

3.5.4.85 IPIPE_D2F_2ND_STR_01

Table 471. IPIPE_D2F_2ND_STR_01

Address Offset	0x0000 0174		
Physical Address	0x5505 0974 0x5C01 0974	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4:0	VAL	Noise filter-2 intensity values (STR).	RW	0x00

3.5.4.86 IPIPE_D2F_2ND_STR_02

Table 472. IPIPE_D2F_2ND_STR_02

Address Offset	0x0000 0178		
Physical Address	0x5505 0978 0x5C01 0978	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4:0	VAL	Noise filter-2 intensity values (STR).	RW	0x00

3.5.4.87 IPIPE_D2F_2ND_STR_03

Table 473. IPIPE_D2F_2ND_STR_03

Address Offset	0x0000 017C		
Physical Address	0x5505 097C 0x5C01 097C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4:0	VAL	Noise filter-2 intensity values (STR).	RW	0x00

3.5.4.88 IPIPE_D2F_2ND_STR_04

Table 474. IPIPE_D2F_2ND_STR_04

Address Offset	0x0000 0180		
Physical Address	0x5505 0980 0x5C01 0980	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4:0	VAL	Noise filter-2 intensity values (STR).	RW	0x00

3.5.4.89 IPIPE_D2F_2ND_STR_05

Table 475. IPIPE_D2F_2ND_STR_05

Address Offset	0x0000 0184		
Physical Address	0x5505 0984 0x5C01 0984	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4:0	VAL	Noise filter-2 intensity values (STR).	RW	0x00

3.5.4.90 IPIPE_D2F_2ND_STR_06

Table 476. IPIPE_D2F_2ND_STR_06

Address Offset	0x0000 0188		
Physical Address	0x5505 0988 0x5C01 0988	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4:0	VAL	Noise filter-2 intensity values (STR).	RW	0x00

3.5.4.91 IPIPE_D2F_2ND_STR_07

Table 477. IPIPE_D2F_2ND_STR_07

Address Offset	0x0000 018C		
Physical Address	0x5505 098C 0x5C01 098C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4:0	VAL	Noise filter-2 intensity values (STR).	RW	0x00

3.5.4.92 IPIPE_D2F_2ND_EDG_MIN

Table 478. IPIPE_D2F_2ND_EDG_MIN

Address Offset	0x0000 01B0		
Physical Address	0x5505 09B0 0x5C01 09B0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	Noise filter-2 edge detection MIN	RW	0x000

3.5.4.93 IPIPE_D2F_2ND_EDG_MAX

Table 479. IPIPE_D2F_2ND_EDG_MAX

Address Offset	0x0000 01B4		
Physical Address	0x5505 09B4 0x5C01 09B4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Noise Filter 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	Noise filter-2 edge detection MAX	RW	0x000

3.5.4.94 IPIPE_GIC_EN

Table 480. IPIPE_GIC_EN

Address Offset	0x0000 01B8		
Physical Address	0x5505 09B8 0x5C01 09B8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Green Imbalance Correction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	EN	Enable signal of PreFilter module 0x0: off 0x1: on	RW	0

3.5.4.95 IPIPE_GIC_TYP

Table 481. IPIPE_GIC_TYP

Address Offset	0x0000 01BC		
Physical Address	0x5505 09BC 0x5C01 09BC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Green Imbalance Correction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED													LSC	SEL	TYP

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:3	RESERVED		R	0x0000
2	LSC	Applies LSC gain to threshold value 0x0: Disable 0x1: Enable	RW	0
1	SEL	Threshold Selection This bit selects the threshold either from the register value (GIC_THR) or threshold table of NF-2 0x0: use GIC register value 0x1: use NF2 threshold	RW	0
0	TYP	Algorithm select 0x0: simple averaging algorithm 0x1: adaptive gain control algorithm	RW	0

3.5.4.96 IPIPE_GIC_GAN

Table 482. IPIPE_GIC_GAN

Address Offset	0x0000 01C0		
Physical Address	0x5505 09C0 0x5C01 09C0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Green Imbalance Correction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	VAL specifies the PreFilter gain.	RW	0x00

3.5.4.97 IPIPE_GIC_NFGAIN

Table 483. IPIPE_GIC_NFGAIN

Address Offset	0x0000 01C4		
Physical Address	0x5505 09C4 0x5C01 09C4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Green Imbalance Correction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL		RW	0x00

3.5.4.98 IPIPE_GIC_THR

Table 484. IPIPE_GIC_THR

Address Offset	0x0000 01C8		
Physical Address	0x5505 09C8 0x5C01 09C8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Green Imbalance Correction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Threshold-1 in the adaptive GIC algorithm. For constant gain mode, set IPIPE_GIC_THR=maximum value.	RW	0x000

3.5.4.99 IPIPE_GIC_SLP

Table 485. IPIPE_GIC_SLP

Address Offset	0x0000 01CC		
Physical Address	0x5505 09CC 0x5C01 09CC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Green Imbalance Correction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Slope (THR2-THR1) of GIC algorithm.	RW	0x000

3.5.4.100 IPIPE_WB2_OFT_R

Table 486. IPIPE_WB2_OFT_R

Address Offset	0x0000 01D0		
Physical Address	0x5505 09D0 0x5C01 09D0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	White Balance Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Offset before white balance (S12) -2048 to +2047	RW	0x000

3.5.4.101 IPIPE_WB2_OFT_GR

Table 487. IPIPE_WB2_OFT_GR

Address Offset	0x0000 01D4		
Physical Address	0x5505 09D4 0x5C01 09D4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	White Balance Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Offset before white balance (S12) -2048 to +2047	RW	0x000

3.5.4.102 IPIPE_WB2_OFT_GB

Table 488. IPIPE_WB2_OFT_GB

Address Offset	0x0000 01D8		
Physical Address	0x5505 09D8 0x5C01 09D8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	White Balance Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Offset before white balance (S12) -2048 to +2047	RW	0x000

3.5.4.103 IPIPE_WB2_OFT_B

Table 489. IPIPE_WB2_OFT_B

Address Offset	0x0000 01DC		
Physical Address	0x5505 09DC 0x5C01 09DC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	White Balance Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Offset before white balance (S12) -2048 to +2047	RW	0x000

3.5.4.104 IPIPE_WB2_WGN_R

Table 490. IPIPE_WB2_WGN_R

Address Offset	0x0000 01E0		
Physical Address	0x5505 09E0 0x5C01 09E0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	White Balance Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	White balance gain for R in U4.9 format 0 to +15.998	RW	0x0200

3.5.4.105 IPIPE_WB2_WGN_GR

Table 491. IPIPE_WB2_WGN_GR

Address Offset	0x0000 01E4		
Physical Address	0x5505 09E4 0x5C01 09E4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	White Balance Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	White balance gain for Gr in U4.9 format 0 to +15.998	RW	0x0200

3.5.4.106 IPIPE_WB2_WGN_GB

Table 492. IPIPE_WB2_WGN_GB

Address Offset	0x0000 01E8		
Physical Address	0x5505 09E8 0x5C01 09E8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	White Balance Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	White balance gain for Gb in U4.9 format 0 to +15.998	RW	0x0200

3.5.4.107 IPIPE_WB2_WGN_B

Table 493. IPIPE_WB2_WGN_B

Address Offset	0x0000 01EC		
Physical Address	0x5505 09EC 0x5C01 09EC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	White Balance Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	White balance gain for B in U4.9 format 0 to +15.998	RW	0x0200

3.5.4.108 IPIPE_CFA_MODE

Table 494. IPIPE_CFA_MODE

Address Offset	0x0000 01F0		
Physical Address	0x5505 09F0 0x5C01 09F0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	CFA Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														MODE	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1:0	MODE	Algorithm selection 0x0: 2DirAC 0x1: 2DirAC + DigitalAntialiasing (DAA) 0x2: DigitalAntialiasing (DAA) 0x3: cosited Bayer format Sets the CFA in cosited Bayer format: the sampling points of R/Gr/Gb/B in each 2x2 block is at the same position. In this mode, the CFA interpolator simply copies R, G, and B to all points in the 2x2 block. (G is an average of Gb and Gr.)	RW	0x0

3.5.4.109 IPIPE_CFA_2DIR_HPF_THR

Table 495. IPIPE_CFA_2DIR_HPF_THR

Address Offset	0x0000 01F4		
Physical Address	0x5505 09F4 0x5C01 09F4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	CFA: HP Value Low Threshold		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			VAL												

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	HPF_THR 2DirCFA HP Value Low Threshold	RW	0x0000

3.5.4.110 IPIPE_CFA_2DIR_HPF_SLP

Table 496. IPIPE_CFA_2DIR_HPF_SLP

Address Offset	0x0000 01F8		
Physical Address	0x5505 09F8 0x5C01 09F8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	CFA: HP Value Slope		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	HPF_SLP 2DirCFA HP Value Slope	RW	0x000

3.5.4.111 IPIPE_CFA_2DIR_MIX_THR

Table 497. IPIPE_CFA_2DIR_MIX_THR

Address Offset	0x0000 01FC		
Physical Address	0x5505 09FC 0x5C01 09FC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	CFA: HP Mix Threshold		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	MIX_THR 2DirCFA HP Mix Threshold	RW	0x0000

3.5.4.112 IPIPE_CFA_2DIR_MIX_SLP

Table 498. IPIPE_CFA_2DIR_MIX_SLP

Address Offset	0x0000 0200		
Physical Address	0x5505 0A00 0x5C01 0A00	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	CFA Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	MIX_SLP 2DirCFA HP Mix Slope	RW	0x000

3.5.4.113 IPIPE_CFA_2DIR_DIR_TRH

Table 499. IPIPE_CFA_2DIR_DIR_TRH

Address Offset	0x0000 0204		
Physical Address	0x5505 0A04 0x5C01 0A04	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	CFA: Direction Threshold		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	DIR_THR 2DirCFA Direction Threshold	RW	0x000

3.5.4.114 IPIPE_CFA_2DIR_DIR_SLP

Table 500. IPIPE_CFA_2DIR_DIR_SLP

Address Offset	0x0000 0208		
Physical Address	0x5505 0A08 0x5C01 0A08	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	CFA: Direction Slope		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:7	RESERVED		R	0x000
6:0	VAL	DIR_SLP 2DirCFA Direction Slope	RW	0x00

3.5.4.115 IPIPE_CFA_2DIR_NDWT

Table 501. IPIPE_CFA_2DIR_NDWT

Address Offset	0x0000 020C		
Physical Address	0x5505 0A0C 0x5C01 0A0C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:6	RESERVED		R	0x000
5:0	VAL	ND Weight 2DirCFA NonDirectional Weight	RW	0x00

3.5.4.116 IPIPE_CFA_MONO_HUE_FRA

Table 502. IPIPE_CFA_MONO_HUE_FRA

Address Offset	0x0000 0210		
Physical Address	0x5505 0A10 0x5C01 0A10	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Mono CFA Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:6	RESERVED		R	0x000
5:0	VAL	HUE_FRA DAA Hue Fraction	RW	0x00

3.5.4.117 IPIPE_CFA_MONO_EDG_THR

Table 503. IPIPE_CFA_MONO_EDG_THR

Address Offset	0x0000 0214		
Physical Address	0x5505 0A14 0x5C01 0A14	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	monoCFA THR SLP		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	EDGE_THR DAA Edge Threshold	RW	0x00

3.5.4.118 IPIPE_CFA_MONO_THR_MIN

Table 504. IPIPE_CFA_MONO_THR_MIN

Address Offset	0x0000 0218		
Physical Address	0x5505 0A18 0x5C01 0A18	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Mono CFA Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	THR_MIN DAA Threshold Minimum	RW	0x000

3.5.4.119 IPIPE_CFA_MONO_THR_SLP

Table 505. IPIPE_CFA_MONO_THR_SLP

Address Offset	0x0000 021C		
Physical Address	0x5505 0A1C 0x5C01 0A1C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	CFA: Threshold Slope		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	THR_SLP DAA Threshold Slope	RW	0x000

3.5.4.120 IPIPE_CFA_MONO_SLP_MIN

Table 506. IPIPE_CFA_MONO_SLP_MIN

Address Offset	0x0000 0220		
Physical Address	0x5505 0A20 0x5C01 0A20	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	CFA: Threshold Minimum		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	SLP_MIN DAA Slope Minimum	RW	0x000

3.5.4.121 IPIPE_CFA_MONO_SLP_SLP

Table 507. IPIPE_CFA_MONO_SLP_SLP

Address Offset	0x0000 0224		
Physical Address	0x5505 0A24 0x5C01 0A24	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	CFA: Threshold Slope		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	SLP_SLP DAA Slope Slope	RW	0x000

3.5.4.122 IPIPE_CFA_MONO_LPWT

Table 508. IPIPE_CFA_MONO_LPWT

Address Offset	0x0000 0228		
Physical Address	0x5505 0A28 0x5C01 0A28	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	CFA: LP Weight		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:6	RESERVED		R	0x000
5:0	VAL	LPWT DAA LP Weight	RW	0x00

3.5.4.123 IPIPE_RGB1_MUL_RR

Table 509. IPIPE_RGB1_MUL_RR

Address Offset	0x0000 022C		
Physical Address	0x5505 0A2C 0x5C01 0A2C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient. 011111111111 = 2047/256 = 7.99609375 011111111110 = 2046/256 [...] 000011111111 = 255/256 000100000000 = 256/256 = 1 000100000001 = 257/256 [...] 000000000001 = 1/256 000000000000 = 0/256 = 0 111111111111 = -1/256 = -0.00390625 111111111110 = -2/256 [...] 100000000001 = -2047/256 100000000000 = -2048/256 = -8.	RW	0x100

3.5.4.124 IPIPE_RGB1_MUL_GR

Table 510. IPIPE_RGB1_MUL_GR

Address Offset	0x0000 0230		
Physical Address	0x5505 0A30 0x5C01 0A30	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0x000

3.5.4.125 IPIPE_RGB1_MUL_BR

Table 511. IPIPE_RGB1_MUL_BR

Address Offset	0x0000 0234		
Physical Address	0x5505 0A34 0x5C01 0A34	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0x000

3.5.4.126 IPIPE_RGB1_MUL_RG

Table 512. IPIPE_RGB1_MUL_RG

Address Offset	0x0000 0238		
Physical Address	0x5505 0A38 0x5C01 0A38	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0x000

3.5.4.127 IPIPE_RGB1_MUL_GG

Table 513. IPIPE_RGB1_MUL_GG

Address Offset	0x0000 023C		
Physical Address	0x5505 0A3C 0x5C01 0A3C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0x100

3.5.4.128 IPIPE_RGB1_MUL_BG

Table 514. IPIPE_RGB1_MUL_BG

Address Offset	0x0000 0240		
Physical Address	0x5505 0A40 0x5C01 0A40	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0x000

3.5.4.129 IPIPE_RGB1_MUL_RB

Table 515. IPIPE_RGB1_MUL_RB

Address Offset	0x0000 0244		
Physical Address	0x5505 0A44 0x5C01 0A44	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0x000

3.5.4.130 IPIPE_RGB1_MUL_GB

Table 516. IPIPE_RGB1_MUL_GB

Address Offset	0x0000 0248		
Physical Address	0x5505 0A48 0x5C01 0A48	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0x000

3.5.4.131 IPIPE_RGB1_MUL_BB

Table 517. IPIPE_RGB1_MUL_BB

Address Offset	0x0000 024C		
Physical Address	0x5505 0A4C 0x5C01 0A4C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0x100

3.5.4.132 IPIPE_RGB1_OFT_OR

Table 518. IPIPE_RGB1_OFT_OR

Address Offset	0x0000 0250		
Physical Address	0x5505 0A50 0x5C01 0A50	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	The output offset value for R. (s13) -4096 to +4095	RW	0x0000

3.5.4.133 IPIPE_RGB1_OFT_OG

Table 519. IPIPE_RGB1_OFT_OG

Address Offset	0x0000 0254		
Physical Address	0x5505 0A54 0x5C01 0A54	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	The output offset value for G. (s13) -4096 to +4095	RW	0x0000

3.5.4.134 IPIPE_RGB1_OFT_OB

Table 520. IPIPE_RGB1_OFT_OB

Address Offset	0x0000 0258		
Physical Address	0x5505 0A58 0x5C01 0A58	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	The output offset value for B. (s13) -4096 to +4095	RW	0x0000

3.5.4.135 IPIPE_GMM_CFG

Table 521. IPIPE_GMM_CFG

Address Offset	0x0000 025C		
Physical Address	0x5505 0A5C 0x5C01 0A5C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								SIZ		TBL	RESERVED	BYPB	BYPG	BYPR	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:7	RESERVED		R	0x000
6:5	SIZ	The size of the gamma table. 0x0: 64 words 0x1: 128 words 0x2: 256 words 0x3: 512 words	RW	0x3
4	TBL	Selection of Gamma table. 0x0: RAM 0x1: ROM	RW	0
3	RESERVED		RW	0
2	BYPB	Gamma correction mode for B 0x0: Not bypassed 0x1: Bypassed	RW	1
1	BYPG	Gamma correction mode for G 0x0: Not bypassed 0x1: Bypassed	RW	1
0	BYPR	Gamma correction mode for R 0x0: Not bypassed 0x1: Bypassed	RW	1

3.5.4.136 IPIPE_RGB2_MUL_RR

Table 522. IPIPE_RGB2_MUL_RR

Address Offset	0x0000 0260		
Physical Address	0x5505 0A60 0x5C01 0A60	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB conversion after gamma		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The matrix coefficient. 011111111111 = 2047/256 = 7.99609375 011111111110 = 2046/256 000011111111 = 255/256 000100000000 = 256/256 = 1 000100000001 = 257/256 000000000001 = 1/256 000000000000 = 0/256 = 0 111111111111 = -1/256 = -0.00390625 111111111110 = -2/256 100000000001 = -2047/256 100000000000 = -2048/256 = -8.	RW	0x100

3.5.4.137 IPIPE_RGB2_MUL_GR

Table 523. IPIPE_RGB2_MUL_GR

Address Offset	0x0000 0264		
Physical Address	0x5505 0A64 0x5C01 0A64	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB conversion after gamma		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The matrix coefficient.	RW	0x000

3.5.4.138 IPIPE_RGB2_MUL_BR

Table 524. IPIPE_RGB2_MUL_BR

Address Offset	0x0000 0268		
Physical Address	0x5505 0A68 0x5C01 0A68	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB conversion after gamma		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The matrix coefficient.	RW	0x000

3.5.4.139 IPIPE_RGB2_MUL_RG

Table 525. IPIPE_RGB2_MUL_RG

Address Offset	0x0000 026C		
Physical Address	0x5505 0A6C 0x5C01 0A6C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB conversion after gamma		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The matrix coefficient.	RW	0x000

3.5.4.140 IPIPE_RGB2_MUL_GG

Table 526. IPIPE_RGB2_MUL_GG

Address Offset	0x0000 0270		
Physical Address	0x5505 0A70 0x5C01 0A70	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB conversion after gamma		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The matrix coefficient.	RW	0x100

3.5.4.141 IPIPE_RGB2_MUL_BG

Table 527. IPIPE_RGB2_MUL_BG

Address Offset	0x0000 0274		
Physical Address	0x5505 0A74 0x5C01 0A74	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB conversion after gamma		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The matrix coefficient.	RW	0x000

3.5.4.142 IPIPE_RGB2_MUL_RB

Table 528. IPIPE_RGB2_MUL_RB

Address Offset	0x0000 0278		
Physical Address	0x5505 0A78 0x5C01 0A78	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB conversion after gamma		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The matrix coefficient.	RW	0x000

3.5.4.143 IPIPE_RGB2_MUL_GB

Table 529. IPIPE_RGB2_MUL_GB

Address Offset	0x0000 027C		
Physical Address	0x5505 0A7C 0x5C01 0A7C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB conversion after gamma		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The matrix coefficient.	RW	0x000

3.5.4.144 IPIPE_RGB2_MUL_BB

Table 530. IPIPE_RGB2_MUL_BB

Address Offset	0x0000 0280		
Physical Address	0x5505 0A80 0x5C01 0A80	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB conversion after gamma		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The matrix coefficient.	RW	0x100

3.5.4.145 IPIPE_RGB2_OFT_OR

Table 531. IPIPE_RGB2_OFT_OR

Address Offset	0x0000 0284		
Physical Address	0x5505 0A84 0x5C01 0A84	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB conversion after gamma		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The output offset value for R S10 number: -1024 to +1023	RW	0x000

3.5.4.146 IPIPE_RGB2_OFT_OG

Table 532. IPIPE_RGB2_OFT_OG

Address Offset	0x0000 0288		
Physical Address	0x5505 0A88 0x5C01 0A88	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB conversion after gamma		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The output offset value for G S10 number: -1024 to +1023	RW	0x000

3.5.4.147 IPIPE_RGB2_OFT_OB

Table 533. IPIPE_RGB2_OFT_OB

Address Offset	0x0000 028C		
Physical Address	0x5505 0A8C 0x5C01 0A8C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to RGB conversion after gamma		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The output offset value for B S10 number: -1024 to +1023	RW	0x000

3.5.4.148 IPIPE_3DLUT_EN

Table 534. IPIPE_3DLUT_EN

Address Offset	0x0000 0290		
Physical Address	0x5505 0A90 0x5C01 0A90	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	3D-LUT		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	EN	Enables or disable the 3D-LUT function. The table entries are store in internal memory. 0x0: Disable 0x1: Enable	RW	0

3.5.4.149 IPIPE_YUV_ADJ

Table 535. IPIPE_YUV_ADJ

Address Offset	0x0000 0294		
Physical Address	0x5505 0A94 0x5C01 0A94	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to YUV Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BRT								CRT							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	BRT	The offset value for brightness control.	RW	0x00
7:0	CRT	The multiplier coefficient value for contrast control. 00000000 = 0/16 = 0 00000001 = 1/16 00001111 = 15/16 00010000 = 16/16 = 1 00010001 = 17/16 11111110 = 254/16 11111111 = 255/16 = 15.9375	RW	0x10

3.5.4.150 IPIPE_YUV_MUL_RY

Table 536. IPIPE_YUV_MUL_RY

Address Offset	0x0000 0298		
Physical Address	0x5505 0A98 0x5C01 0A98	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to YUV Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Matrix Coefficient for RY S4.8 = -8 to +7.996	RW	0x04D

3.5.4.151 IPIPE_YUV_MUL_GY

Table 537. IPIPE_YUV_MUL_GY

Address Offset	0x0000 029C		
Physical Address	0x5505 0A9C 0x5C01 0A9C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to YUV Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Matrix Coefficient for GY S4.8 = -8 to +7.996	RW	0x096

3.5.4.152 IPIPE_YUV_MUL_BY

Table 538. IPIPE_YUV_MUL_BY

Address Offset	0x0000 02A0		
Physical Address	0x5505 0AA0 0x5C01 0AA0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to YUV Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Matrix Coefficient for BY S4.8 = -8 to +7.996	RW	0x01D

3.5.4.153 IPIPE_YUV_MUL_RCB

Table 539. IPIPE_YUV_MUL_RCB

Address Offset	0x0000 02A4		
Physical Address	0x5505 0AA4 0x5C01 0AA4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to YUV Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0xFD5

3.5.4.154 IPIPE_YUV_MUL_GCB

Table 540. IPIPE_YUV_MUL_GCB

Address Offset	0x0000 02A8		
Physical Address	0x5505 0AA8 0x5C01 0AA8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to YUV Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0xFAB

3.5.4.155 IPIPE_YUV_MUL_BCB

Table 541. IPIPE_YUV_MUL_BCB

Address Offset	0x0000 02AC		
Physical Address	0x5505 0AAC 0x5C01 0AAC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to YUV Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0x080

3.5.4.156 IPIPE_YUV_MUL_RCR

Table 542. IPIPE_YUV_MUL_RCR

Address Offset	0x0000 02B0		
Physical Address	0x5505 0AB0 0x5C01 0AB0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to YUV Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0x080

3.5.4.157 IPIPE_YUV_MUL_GCR

Table 543. IPIPE_YUV_MUL_GCR

Address Offset	0x0000 02B4		
Physical Address	0x5505 0AB4 0x5C01 0AB4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to YUV Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0xF95

3.5.4.158 IPIPE_YUV_MUL_BCR

Table 544. IPIPE_YUV_MUL_BCR

Address Offset	0x0000 02B8		
Physical Address	0x5505 0AB8 0x5C01 0AB8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to YUV Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0xFEB

3.5.4.159 IPIPE_YUV_OFT_Y

Table 545. IPIPE_YUV_OFT_Y

Address Offset	0x0000 02BC		
Physical Address	0x5505 0ABC 0x5C01 0ABC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to YUV Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The output offset value for Y	RW	0x000

3.5.4.160 IPIPE_YUV_OFT_CB

Table 546. IPIPE_YUV_OFT_CB

Address Offset	0x0000 02C0		
Physical Address	0x5505 0AC0 0x5C01 0AC0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to YUV Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED					VAL										

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The output offset value for Cb For Cb/Cr, set (0x80 + offset value) here. (0x80 for zero offset.)	RW	0x080

3.5.4.161 IPIPE_YUV_OFT_CR

Table 547. IPIPE_YUV_OFT_CR

Address Offset	0x0000 02C4		
Physical Address	0x5505 0AC4 0x5C01 0AC4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	RGB to YUV Conversion Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED					VAL										

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The output offset value for Cr For Cb/Cr, set (0x80 + offset value) here. (0x80 for zero offset.)	RW	0x080

3.5.4.162 IPIPE_YUV_PHS

Table 548. IPIPE_YUV_PHS

Address Offset	0x0000 02C8		
Physical Address	0x5505 0AC8 0x5C01 0AC8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	YUV4:2:2 down sampling register. This register controls the YUV4:4:4 to YUV4:2:2 chroma downsampling. This register is valid if IPIPE_SRC_FMT.FMT = 0 (RAW input and YUV output). IPIPE_YUV_PHS = 0 leads to pure subsampling, no filtering, cosited chroma output. IPIPE_YUV_PHS = 1 leads to (1, 1) 1 filtering, centered chroma output. IPIPE_YUV_PHS = 2 leads to (1, 2, 1) 1 filtering, cosited chroma output. IPIPE_YUV_PHS = 3 leads to (1, 3, 3, 1) 3 filtering, centered chroma output. When the chroma output is cosited, and that downsampling is enabled in the RESIZER module, one need to take care that the averager disrupts the relative phase for luma and chroma color components. The RZA_H_PHS_ADJ and RZB_H_PHS_ADJ registers need to be used to fix the disruption.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														LPF	POS

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1	LPF	121-LPF enable for chrominance samples. This register is valid if IPIPE_SRC_FMT.FMT = 0 (RAW input and YUV output). 0x0: off 0x1: on	RW	0
0	POS	This bit sets the output position of the chrominance sample with regards to the luma sample positions. One can choose between centered and cosited. This register is valid if IPIPE_SRC_FMT.FMT = 0 (RAW input and YUV output). The RESIZER module does not change the relative position of the chroma samples vs. the luma samples between the input and output and the chroma position at the output of the IPIPE module and at the output of the RESIZER module must be identical. In other words, we must have RSZ_YUV_PHS.POS = IPIPE_YUV_PHS.POS. 0x0: Cosited = same position with luminance 0x1: Centered = middle of the luminance	RW	0

3.5.4.163 IPIPE_GBCE_EN

Table 549. IPIPE_GBCE_EN

Address Offset	0x0000 02CC		
Physical Address	0x5505 0ACC 0x5C01 0ACC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Global brightness contrast enhancement		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	EN	Enable of GBCE module. 0x0: Off 0x1: On	RW	0

3.5.4.164 IPIPE_GBCE_TYP

Table 550. IPIPE_GBCE_TYP

Address Offset	0x0000 02D0		
Physical Address	0x5505 0AD0 0x5C01 0AD0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Global brightness contrast enhancement		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															TYP

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	TYP	GBCE method selection 0x0: Y value table Y out = LUT[Y in] Cb out = Cb in Cr out = Cr in 0x1: Gain table Y out = LUT[Y in] * Y in Cb out = LUT[Y in] * Cb in Cr out = LUT[Y in] * Cr in	RW	0

3.5.4.165 IPIPE_YEE_EN

Table 551. IPIPE_YEE_EN

Address Offset	0x0000 02D4		
Physical Address	0x5505 0AD4 0x5C01 0AD4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Edge Enhancer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	EN	The on/off selection of the Edge enhancer. 0x0: Disable 0x1: Enable	RW	0

3.5.4.166 IPIPE_YEE_TYP

Table 552. IPIPE_YEE_TYP

Address Offset	0x0000 02D8		
Physical Address	0x5505 0AD8 0x5C01 0AD8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Edge Enhancer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														HAL	SEL

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1	HAL	Halo reduction in Edge Sharpener module	RW	0
0	SEL	Merging method between Edge Enhancer and Edge Sharpener 0x0: EE + ES 0x1: Max (EE, ES)	RW	0

3.5.4.167 IPIPE_YEE_SHF

Table 553. IPIPE_YEE_SHF

Address Offset	0x0000 02DC		
Physical Address	0x5505 0ADC 0x5C01 0ADC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Edge Enhancer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED												SHF			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:4	RESERVED		R	0x000
3:0	SHF	Down shift length of high pass filter (HPF) in edge enhancer.	RW	0x0

3.5.4.168 IPIPE_YEE_MUL_00

Table 554. IPIPE_YEE_MUL_00

Address Offset	0x0000 02E0		
Physical Address	0x5505 0AE0 0x5C01 0AE0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Edge Enhancer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Multiplier coefficient in HPF. 0111111111 = 511 0111111110 = 510 0000000001 = 1 0000000000 = 0 1111111111 = -1 1000000001 = -511 1000000000 = -512	RW	0x000

3.5.4.169 IPIPE_YEE_MUL_01

Table 555. IPIPE_YEE_MUL_01

Address Offset	0x0000 02E4		
Physical Address	0x5505 0AE4 0x5C01 0AE4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Edge Enhancer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Multiplier coefficient in HPF.	RW	0x000

3.5.4.170 IPIPE_YEE_MUL_02

Table 556. IPIPE_YEE_MUL_02

Address Offset	0x0000 02E8		
Physical Address	0x5505 0AE8 0x5C01 0AE8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Edge Enhancer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Multiplier coefficient in HPF.	RW	0x000

3.5.4.171 IPIPE_YEE_MUL_10

Table 557. IPIPE_YEE_MUL_10

Address Offset	0x0000 02EC		
Physical Address	0x5505 0AEC 0x5C01 0AEC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Edge Enhancer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Multiplier coefficient in HPF.	RW	0x000

3.5.4.172 IPIPE_YEE_MUL_11

Table 558. IPIPE_YEE_MUL_11

Address Offset	0x0000 02F0		
Physical Address	0x5505 0AF0 0x5C01 0AF0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Edge Enhancer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Multiplier coefficient in HPF.	RW	0x000

3.5.4.173 IPIPE_YEE_MUL_12

Table 559. IPIPE_YEE_MUL_12

Address Offset	0x0000 02F4		
Physical Address	0x5505 0AF4 0x5C01 0AF4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Edge Enhancer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Multiplier coefficient in HPF.	RW	0x000

3.5.4.174 IPIPE_YEE_MUL_20

Table 560. IPIPE_YEE_MUL_20

Address Offset	0x0000 02F8		
Physical Address	0x5505 0AF8 0x5C01 0AF8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Edge Enhancer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Multiplier coefficient in HPF.	RW	0x000

3.5.4.175 IPIPE_YEE_MUL_21

Table 561. IPIPE_YEE_MUL_21

Address Offset	0x0000 02FC		
Physical Address	0x5505 0AFC 0x5C01 0AFC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Edge Enhancer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Multiplier coefficient in HPF.	RW	0x000

3.5.4.176 IPIPE_YEE_MUL_22

Table 562. IPIPE_YEE_MUL_22

Address Offset	0x0000 0300		
Physical Address	0x5505 0B00 0x5C01 0B00	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Edge Enhancer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Multiplier coefficient in HPF.	RW	0x000

3.5.4.177 IPIPE_YEE_THR

Table 563. IPIPE_YEE_THR

Address Offset	0x0000 0304		
Physical Address	0x5505 0B04 0x5C01 0B04	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Edge Enhancer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:6	RESERVED		R	0x000
5:0	VAL	Edge Enhancer lower threshold before referring to LUT. If HPF -IPIPE_YEE_THR - output is HPF + IPIPE_YEE_THR If HPF IPIPE_YEE_THR - output is HPF - IPIPE_YEE_THR Otherwise, output is zero.	RW	0x00

3.5.4.178 IPIPE_YEE_E_GAN

Table 564. IPIPE_YEE_E_GAN

Address Offset	0x0000 0308		
Physical Address	0x5505 0B08 0x5C01 0B08	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Edge Enhancer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Edge sharpener gain	RW	0x000

3.5.4.179 IPIPE_YEE_E_THR_1

Table 565. IPIPE_YEE_E_THR_1

Address Offset	0x0000 030C		
Physical Address	0x5505 0B0C 0x5C01 0B0C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Edge Enhancer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Edge sharpener HPF value lower limit	RW	0x000

3.5.4.180 IPIPE_YEE_E_THR_2

Table 566. IPIPE_YEE_E_THR_2

Address Offset	0x0000 0310		
Physical Address	0x5505 0B10 0x5C01 0B10	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Edge Enhancer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:6	RESERVED		R	0x000
5:0	VAL	Edge sharpener HPF value upper limit (after 6-bit right shift)	RW	0x00

3.5.4.181 IPIPE_YEE_G_GAN

Table 567. IPIPE_YEE_G_GAN

Address Offset	0x0000 0314		
Physical Address	0x5505 0B14 0x5C01 0B14	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Edge Enhancer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	Edge sharpener, gain value on gradient	RW	0x00

3.5.4.182 IPIPE_YEE_G_OFT

Table 568. IPIPE_YEE_G_OFT

Address Offset	0x0000 0318		
Physical Address	0x5505 0B18 0x5C01 0B18	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Edge Enhancer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:6	RESERVED		R	0x000
5:0	VAL	Edge sharpener, offset value on gradient	RW	0x00

3.5.4.183 IPIPE_CAR_EN

Table 569. IPIPE_CAR_EN

Address Offset	0x0000 031C		
Physical Address	0x5505 0B1C 0x5C01 0B1C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Artifact Reduction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	EN	The on/off selection of the fault color suppression. When the RSZ_SRC_FMT.FMT0=1 (Bayer input, Bayer output), this bit must be 0. 0x0: Disable 0x1: Enable	RW	0

3.5.4.184 IPIPE_CAR_TYP

Table 570. IPIPE_CAR_TYP

Address Offset	0x0000 0320		
Physical Address	0x5505 0B20 0x5C01 0B20	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Artifact Reduction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															TYP

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	TYP	The mode selection of the fault color suppression. To select median filter, set TYP=1 and IPIPE_CAR_SWT.SW0=255 (default values) 0x0: Gain control 0x1: Dynamic switching	RW	1

3.5.4.185 IPIPE_CAR_SW

Table 571. IPIPE_CAR_SW

Address Offset	0x0000 0324		
Physical Address	0x5505 0B24 0x5C01 0B24	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Artifact Reduction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SW1								SW0							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	SW1	Threshold-2 for switching function (Select gain control)	RW	0x00
7:0	SW0	Threshold-1 for switching function (Select median filter)	RW	0xFF

3.5.4.186 IPIPE_CAR_HPF_TYP

Table 572. IPIPE_CAR_HPF_TYP

Address Offset	0x0000 0328		
Physical Address	0x5505 0B28 0x5C01 0B28	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Artifact Reduction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														TYP	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:3	RESERVED		R	0x0000
2:0	TYP	0x0: Y 0x1: Horizontal HPF 0x2: Vertical HPF 0x3: 2D HPF 0x4: 2D HPF from edge enhancer 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x0

3.5.4.187 IPIPE_CAR_HPF_SHF

Table 573. IPIPE_CAR_HPF_SHF

Address Offset	0x0000 032C		
Physical Address	0x5505 0B2C 0x5C01 0B2C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Artifact Reduction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														VAL	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1:0	VAL	Down shift value for HPF.	RW	0x0

3.5.4.188 IPIPE_CAR_HPF_THR

Table 574. IPIPE_CAR_HPF_THR

Address Offset	0x0000 0330		
Physical Address	0x5505 0B30 0x5C01 0B30	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Artifact Reduction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	The threshold of the gain function for HPF value	RW	0x00

3.5.4.189 IPIPE_CAR_GN1_GAN

Table 575. IPIPE_CAR_GN1_GAN

Address Offset	0x0000 0334		
Physical Address	0x5505 0B34 0x5C01 0B34	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Artifact Reduction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	The intensity of the gain function for HPF value	RW	0x00

3.5.4.190 IPIPE_CAR_GN1_SHF

Table 576. IPIPE_CAR_GN1_SHF

Address Offset	0x0000 0338		
Physical Address	0x5505 0B38 0x5C01 0B38	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Artifact Reduction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														VAL	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:3	RESERVED		R	0x0000
2:0	VAL	The down shift value of the gain function on HPF value	RW	0x0

3.5.4.191 IPIPE_CAR_GN1_MIN

Table 577. IPIPE_CAR_GN1_MIN

Address Offset	0x0000 033C		
Physical Address	0x5505 0B3C 0x5C01 0B3C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Artifact Reduction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:9	RESERVED		R	0x00
8:0	VAL	The lower limit of the gain function on HPF value	RW	0x000

3.5.4.192 IPIPE_CAR_GN2_GAN

Table 578. IPIPE_CAR_GN2_GAN

Address Offset	0x0000 0340		
Physical Address	0x5505 0B40 0x5C01 0B40	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Artifact Reduction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	The intensity of the gain function for chroma value	RW	0x00

3.5.4.193 IPIPE_CAR_GN2_SHF

Table 579. IPIPE_CAR_GN2_SHF

Address Offset	0x0000 0344		
Physical Address	0x5505 0B44 0x5C01 0B44	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Artifact Reduction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												VAL			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:0	VAL	The down shift value of the gain function on chroma value	RW	0x0

3.5.4.194 IPIPE_CAR_GN2_MIN

Table 580. IPIPE_CAR_GN2_MIN

Address Offset	0x0000 0348		
Physical Address	0x5505 0B48 0x5C01 0B48	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Artifact Reduction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:9	RESERVED		R	0x00
8:0	VAL	The lower limit of the gain function on chroma value	RW	0x000

3.5.4.195 IPIPE_CGS_EN

Table 581. IPIPE_CGS_EN

Address Offset	0x0000 034C		
Physical Address	0x5505 0B4C 0x5C01 0B4C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Gain Suppression		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	EN	Enables or disables chroma gain suppression 0x0: Disable 0x1: Enable	RW	0

3.5.4.196 IPIPE_CGS_GN1_L_THR

Table 582. IPIPE_CGS_GN1_L_THR

Address Offset	0x0000 0350		
Physical Address	0x5505 0B50 0x5C01 0B50	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Gain Suppression		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL		RW	0x00

3.5.4.197 IPIPE_CGS_GN1_L_GAIN

Table 583. IPIPE_CGS_GN1_L_GAIN

Address Offset	0x0000 0354		
Physical Address	0x5505 0B54 0x5C01 0B54	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Gain Suppression		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL		RW	0x00

3.5.4.198 IPIPE_CGS_GN1_L_SHF

Table 584. IPIPE_CGS_GN1_L_SHF

Address Offset	0x0000 0358		
Physical Address	0x5505 0B58 0x5C01 0B58	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Gain Suppression		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														VAL	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:3	RESERVED		R	0x0000
2:0	VAL		RW	0x0

3.5.4.199 IPIPE_CGS_GN1_L_MIN

Table 585. IPIPE_CGS_GN1_L_MIN

Address Offset	0x0000 035C		
Physical Address	0x5505 0B5C 0x5C01 0B5C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Gain Suppression		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	The lower limit 1 of the gain function 1	RW	0x00

3.5.4.200 IPIPE_CGS_GN1_H_THR

Table 586. IPIPE_CGS_GN1_H_THR

Address Offset	0x0000 0360		
Physical Address	0x5505 0B60 0x5C01 0B60	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Gain Suppression		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	The threshold 2 of the gain function for Y value	RW	0x00

3.5.4.201 IPIPE_CGS_GN1_H_GAIN

Table 587. IPIPE_CGS_GN1_H_GAIN

Address Offset	0x0000 0364		
Physical Address	0x5505 0B64 0x5C01 0B64	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Gain Suppression		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	The slope 2 of the gain function for Y value	RW	0x00

3.5.4.202 IPIPE_CGS_GN1_H_SHF

Table 588. IPIPE_CGS_GN1_H_SHF

Address Offset	0x0000 0368		
Physical Address	0x5505 0B68 0x5C01 0B68	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Gain Suppression		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														VAL	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:3	RESERVED		R	0x0000
2:0	VAL	The down shift value of the gain function on Y	RW	0x0

3.5.4.203 IPIPE_CGS_GN1_H_MIN

Table 589. IPIPE_CGS_GN1_H_MIN

Address Offset	0x0000 036C		
Physical Address	0x5505 0B6C 0x5C01 0B6C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Gain Suppression		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	The lower limit 2 of the gain function	RW	0x00

3.5.4.204 IPIPE_CGS_GN2_L_THR

Table 590. IPIPE_CGS_GN2_L_THR

Address Offset	0x0000 0370		
Physical Address	0x5505 0B70 0x5C01 0B70	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Gain Suppression		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	The slope 2 of the gain function for Y value	RW	0x00

3.5.4.205 IPIPE_CGS_GN2_L_GAIN

Table 591. IPIPE_CGS_GN2_L_GAIN

Address Offset	0x0000 0374		
Physical Address	0x5505 0B74 0x5C01 0B74	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Gain Suppression		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	The down shift value 3 of the gain function on Y	RW	0x00

3.5.4.206 IPIPE_CGS_GN2_L_SHF

Table 592. IPIPE_CGS_GN2_L_SHF

Address Offset	0x0000 0378		
Physical Address	0x5505 0B78 0x5C01 0B78	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														VAL	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:3	RESERVED		R	0x0000
2:0	VAL		RW	0x0

3.5.4.207 IPIPE_CGS_GN2_L_MIN

Table 593. IPIPE_CGS_GN2_L_MIN

Address Offset	0x0000 037C		
Physical Address	0x5505 0B7C 0x5C01 0B7C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Chroma Gain Suppression		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	The lower limit 3 of the gain function	RW	0x00

3.5.4.208 IPIPE_BOX_EN

Table 594. IPIPE_BOX_EN

Address Offset	0x0000 0380		
Physical Address	0x5505 0B80 0x5C01 0B80	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boxcar Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	EN	This bit enables or disables the BOXCAR functionality. The BOXCAR output is written to SDRAM. You need to set the IPIPE_BOX_SDR_SAD_H and IPIPE_BOX_SDR_SAD_L registers with the appropriate address. 0x0: Disable 0x1: Enable	RW	0

3.5.4.209 IPIPE_BOX_MODE

Table 595. IPIPE_BOX_MODE

Address Offset	0x0000 0384		
Physical Address	0x5505 0B84 0x5C01 0B84	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boxcar Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															OST

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	OST	The processing mode selection of the Boxcar function. A 0 indicates the mode of the free run, a 1 indicates the mode of the one shot. 0x0: Free run 0x1: One shot	RW	0

3.5.4.210 IPIPE_BOX_TYP

Table 596. IPIPE_BOX_TYP

Address Offset	0x0000 0388		
Physical Address	0x5505 0B88 0x5C01 0B88	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boxcar Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															SEL

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	SEL	Block size in boxcar sampling 0x0: 8 × 8 0x1: 16 × 16	RW	0

3.5.4.211 IPIPE_BOX_SHF

Table 597. IPIPE_BOX_SHF

Address Offset	0x0000 038C		
Physical Address	0x5505 0B8C 0x5C01 0B8C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boxcar Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														VAL	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:3	RESERVED		R	0x0000
2:0	VAL	The down shift value applied to the boxcar computation result. R out = SUM (Rij) SHF G out = (SUM (Gr ij)/2 + SUM (Gr ij)/2) SHF B out = SUM (Gij) SHF	RW	0x0

3.5.4.212 IPIPE_BOX_SDR_SAD_H

Table 598. IPIPE_BOX_SDR_SAD_H

Address Offset	0x0000 0390		
Physical Address	0x5505 0B90 0x5C01 0B90	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boxcar Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	VAL	The higher 11 bits of the first address of output in memory.	RW	0x0000

3.5.4.213 IPIPE_BOX_SDR_SAD_L

Table 599. IPIPE_BOX_SDR_SAD_L

Address Offset	0x0000 0394		
Physical Address	0x5505 0B94 0x5C01 0B94	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boxcar Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VAL														VAL_RESERVED	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	VAL	The lower 16 bits of the first address of output in memory.	RW	0x000
4:0	VAL_RESERVED	Ensures 32-byte alignment.	R	0x00

3.5.4.214 IPIPE_HST_EN

Table 600. IPIPE_HST_EN

Address Offset	0x0000 039C		
Physical Address	0x5505 0B9C 0x5C01 0B9C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	EN	This bit enables or disables the HISTOGRAM functionality. When enabled, the HISTOGRAM computation will start the processing from the next rising edge of the VD pulse. If the processing mode of the HISTOGRAM is one shot, the enable bit will be cleared to 0 immediately after the processing has started. 0x0: disable 0x1: start/busy	RW	0

3.5.4.215 IPIPE_HST_MODE

Table 601. IPIPE_HST_MODE

Address Offset	0x0000 03A0		
Physical Address	0x5505 0BA0 0x5C01 0BA0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															OST

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	OST	The processing mode selection of the Histogram module. A 0 indicates the mode of the free run, a 1 indicates the mode of the one shot. 0x0: Free run 0x1: One shot	RW	0

3.5.4.216 IPIPE_HST_SEL

Table 602. IPIPE_HST_SEL

Address Offset	0x0000 03A4		
Physical Address	0x5505 0BA4 0x5C01 0BA4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED													SEL	TYP	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:3	RESERVED		R	0x0000
2	SEL	Input selection. When SEL0=0, RGBY are sampled from the output of the line buffer in noise filter-2. When SEL0=1, YCbCr are sampled at the output of RGB2YCbCr module. Y is sampled twice. 0x0: From noise filter input 0x1: From RGBtoYUV	RW	0
1:0	TYP	G selection in Bayer mode (SEL0=0) 0x0: Gb 0x1: Gr 0x2: (Gb+Gr)/2 0x3: Reserved	RW	0x0

3.5.4.217 IPIPE_HST_PARA

Table 603. IPIPE_HST_PARA

Address Offset	0x0000 03A8		
Physical Address	0x5505 0BA8 0x5C01 0BA8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram COL0, COL1, COL2, and COL3 should be set to 1.		
Type	RW		

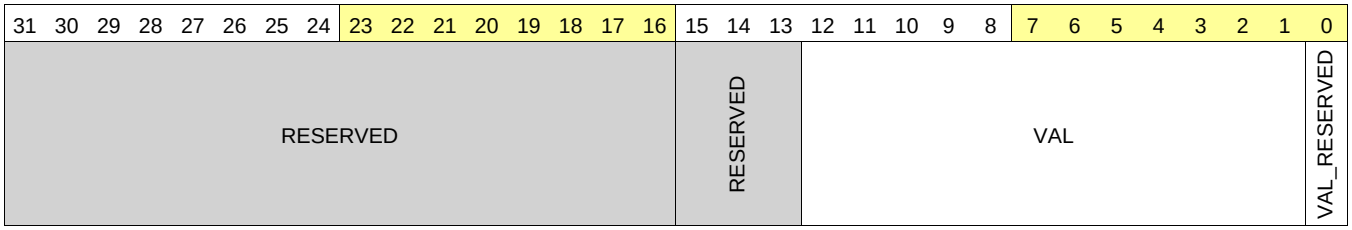
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		BIN		SHF				COL3	COL2	COL1	COL0	RGN3	RGN2	RGN1	RGN0

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:14	RESERVED		R	0x0
13:12	BIN	The number of the bins. 0x0: 32 0x1: 64 0x2: 128 0x3: 256	RW	0x0
11:8	SHF	The shift length of the input data. data = (INPUT SHF)	RW	0x0
7	COL3	The on/off selection of the color pattern 3 (Y). 0x0: Disable 0x1: Enable	RW	0
6	COL2	The on/off selection of the color pattern 2 (B). 0x0: Disable 0x1: Enable	RW	0
5	COL1	The on/off selection of the color pattern 1 (G). 0x0: Disable 0x1: Enable	RW	0
4	COL0	The on/off selection of the color pattern 0 (R). 0x0: Disable 0x1: Enable	RW	0
3	RGN3	The on/off selection of the region 3. 0x0: Disable 0x1: Enable	RW	0
2	RGN2	The on/off selection of the region 2. 0x0: Disable 0x1: Enable	RW	0
1	RGN1	The on/off selection of the region 1. 0x0: Disable 0x1: Enable	RW	0
0	RGN0	The on/off selection of the region 0. 0x0: Disable 0x1: Enable	RW	0

3.5.4.218 IPIPE_HST_0_VPS

Table 604. IPIPE_HST_0_VPS

Address Offset	0x0000 03AC		
Physical Address	0x5505 0BAC 0x5C01 0BAC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The vertical position of the region 0 from the IPIPE_SRC_VPS. The region 0 will start the Histogram processing from VAL line. VAL[0] can not be written.	RW	0x000
0	VAL_RESERVED	The vertical position of the region 0 from the IPIPE_SRC_VPS. The region 0 will start the Histogram processing from VAL line. VAL[0] can not be written.	R	0

3.5.4.219 IPIPE_HST_0_VSZ

Table 605. IPIPE_HST_0_VSZ

Address Offset	0x0000 03B0		
Physical Address	0x5505 0BB0 0x5C01 0BB0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		

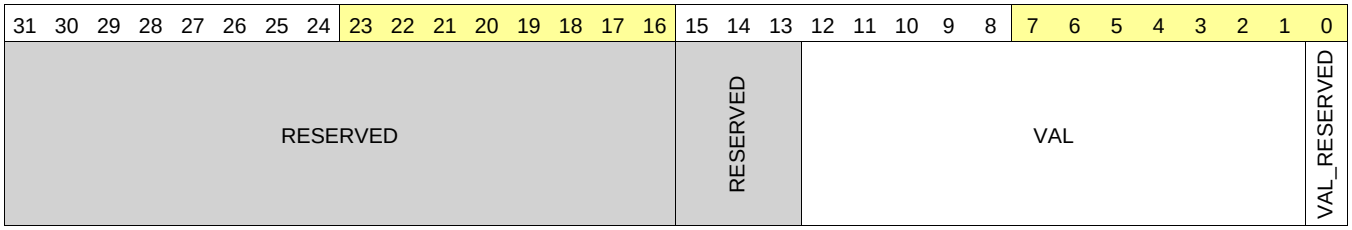
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			VAL												VAL_RESERVED

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The vertical size of the region 0. The Histogram processing of the region 0 will process (VAL+1) lines.	RW	0x000
0	VAL_RESERVED	The vertical size of the region 0. The Histogram processing of the region 0 will process (VAL+1) lines. VAL[0] cannot be written.	R returns 1s	1

3.5.4.220 IPIPE_HST_0_HPS

Table 606. IPIPE_HST_0_HPS

Address Offset	0x0000 03B4		
Physical Address	0x5505 0BB4 0x5C01 0BB4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		

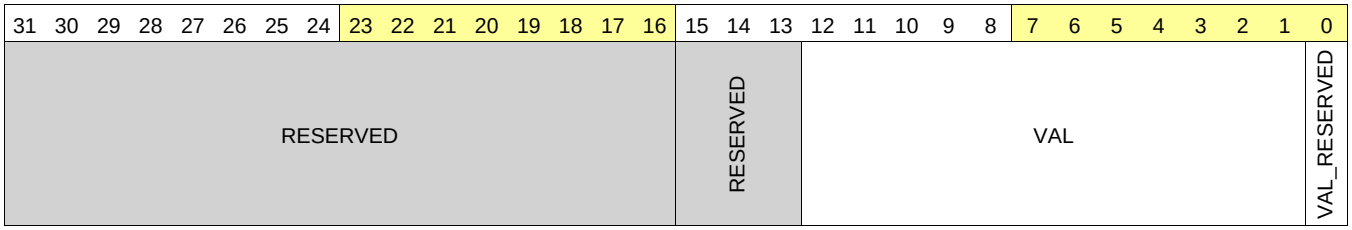


Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The horizontal position of the region 0 from the IPIPE_SRC_HPS. The region 0 will start the Histogram processing from VAL clocks. VAL[0] can not be written.	RW	0x000
0	VAL_RESERVED	The horizontal position of the region 0 from the IPIPE_SRC_HPS. The region 0 will start the Histogram processing from VAL clocks. VAL[0] can not be written.	R	0

3.5.4.221 IPIPE_HST_0_HSZ

Table 607. IPIPE_HST_0_HSZ

Address Offset	0x0000 03B8		
Physical Address	0x5505 0BB8 0x5C01 0BB8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		

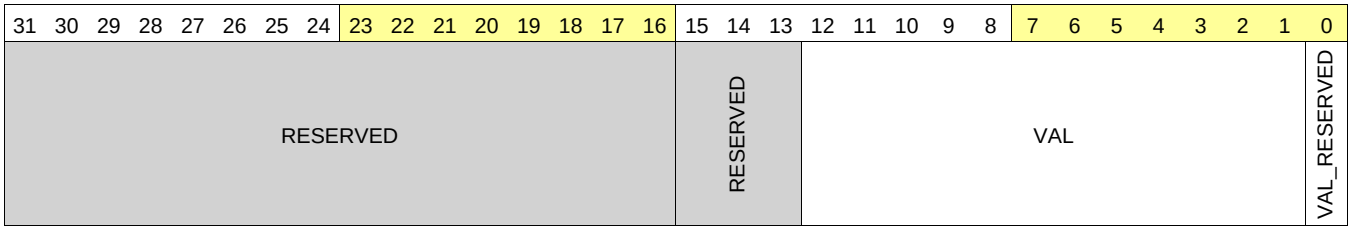


Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The horizontal size of the region 0. The Histogram processing of the region 0 will process (VAL+1) clocks. VAL[0] cannot be written.	RW	0x000
0	VAL_RESERVED	The horizontal size of the region 0. The Histogram processing of the region 0 will process (VAL+1) clocks. VAL[0] cannot be written.	R returns 1s	1

3.5.4.222 IPIPE_HST_1_VPS

Table 608. IPIPE_HST_1_VPS

Address Offset	0x0000 03BC		
Physical Address	0x5505 0BBC 0x5C01 0BBC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		

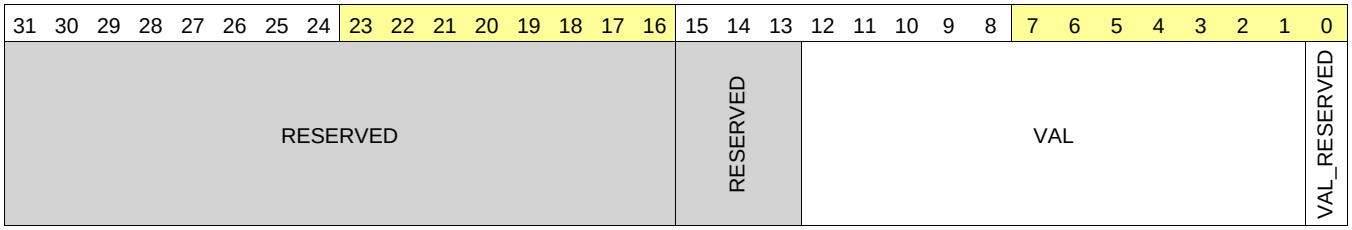


Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The vertical position of the region 0 from the IPIPE_SRC_VPS. The region 1 will start the Histogram processing from VAL line. VAL[0] can not be written.	RW	0x000
0	VAL_RESERVED	The vertical position of the region 0 from the IPIPE_SRC_VPS. The region 1 will start the Histogram processing from VAL line. VAL[0] can not be written.	R	0

3.5.4.223 IPIPE_HST_1_VSZ

Table 609. IPIPE_HST_1_VSZ

Address Offset	0x0000 03C0		
Physical Address	0x5505 0BC0 0x5C01 0BC0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The vertical size of the region 1. The Histogram processing of the region 1 will process (VAL+1) lines. VAL[0] cannot be written.	RW	0x000
0	VAL_RESERVED	The vertical size of the region 1. The Histogram processing of the region 1 will process (VAL+1) lines. VAL[0] cannot be written.	R returns 1s	1

3.5.4.224 IPIPE_HST_1_HPS

Table 610. IPIPE_HST_1_HPS

Address Offset	0x0000 03C4		
Physical Address	0x5505 0BC4 0x5C01 0BC4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			VAL												VAL_RESERVED

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The horizontal position of the region 0 from the IPIPE_SRC_HPS. The region 1 will start the Histogram processing from VAL clocks. VAL[0] can not be written.	RW	0x000
0	VAL_RESERVED	The horizontal position of the region 0 from the IPIPE_SRC_HPS. The region 1 will start the Histogram processing from VAL clocks. VAL[0] can not be written.	R	0

3.5.4.225 IPIPE_HST_1_HSZ

Table 611. IPIPE_HST_1_HSZ

Address Offset	0x0000 03C8		
Physical Address	0x5505 0BC8 0x5C01 0BC8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		

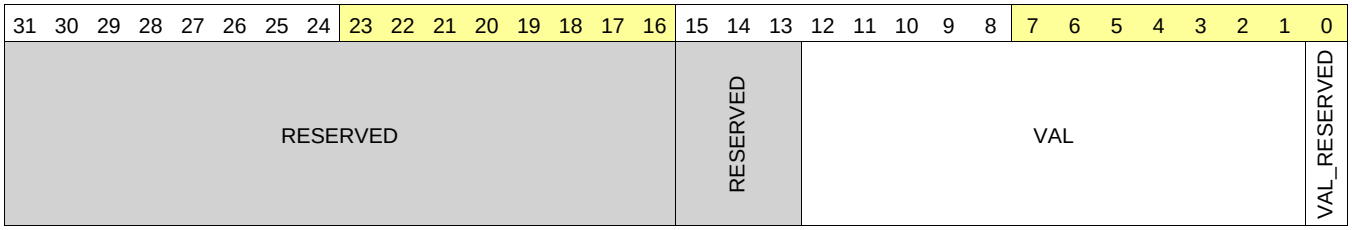
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			VAL												VAL_RESERVED

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The horizontal size of the region 1. The Histogram processing of the region 1 will process (VAL+1) clocks. VAL[0] cannot be written.	RW	0x000
0	VAL_RESERVED	The horizontal size of the region 1. The Histogram processing of the region 1 will process (VAL+1) clocks. VAL[0] cannot be written.	R returns 1s	1

3.5.4.226 IPIPE_HST_2_VPS

Table 612. IPIPE_HST_2_VPS

Address Offset	0x0000 03CC		
Physical Address	0x5505 0BCC 0x5C01 0BCC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		

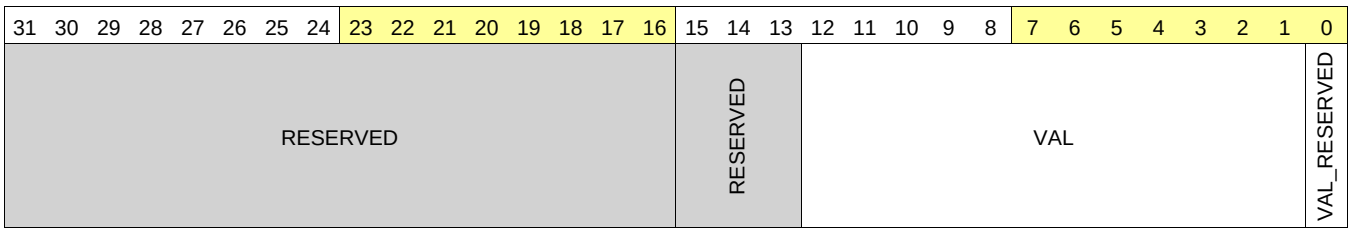


Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The vertical position of the region 0 from the IPIPE_SRC_VPS. The region 2 will start the Histogram processing from VAL line. VAL[0] can not be written.	RW	0x000
0	VAL_RESERVED	The vertical position of the region 0 from the IPIPE_SRC_VPS. The region 2 will start the Histogram processing from VAL line. VAL[0] can not be written.	R	0

3.5.4.227 IPIPE_HST_2_VSZ

Table 613. IPIPE_HST_2_VSZ

Address Offset	0x0000 03D0		
Physical Address	0x5505 0BD0 0x5C01 0BD0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		

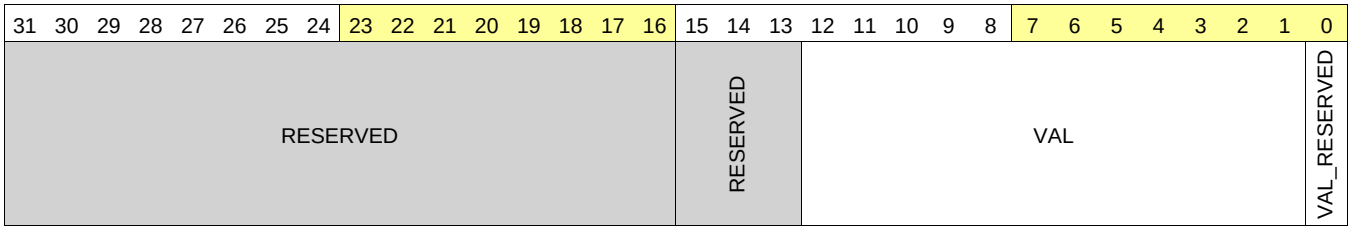


Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The vertical size of the region 2. The Histogram processing of the region 2 will process (VAL+1) lines. VAL[0] cannot be written.	RW	0x000
0	VAL_RESERVED	The vertical size of the region 2. The Histogram processing of the region 2 will process (VAL+1) lines. VAL[0] cannot be written.	R returns 1s	1

3.5.4.228 IPIPE_HST_2_HPS

Table 614. IPIPE_HST_2_HPS

Address Offset	0x0000 03D4		
Physical Address	0x5505 0BD4 0x5C01 0BD4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		

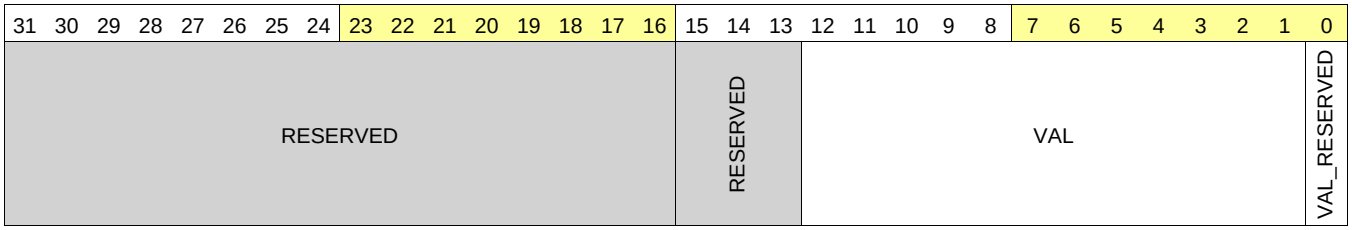


Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The horizontal position of the region 0 from the IPIPE_SRC_HPS. The region 2 will start the Histogram processing from VAL clocks. VAL[0] can not be written.	RW	0x000
0	VAL_RESERVED	The horizontal position of the region 0 from the IPIPE_SRC_HPS. The region 2 will start the Histogram processing from VAL clocks. VAL[0] can not be written.	R	0

3.5.4.229 IPIPE_HST_2_HSZ

Table 615. IPIPE_HST_2_HSZ

Address Offset	0x0000 03D8		
Physical Address	0x5505 0BD8 0x5C01 0BD8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		

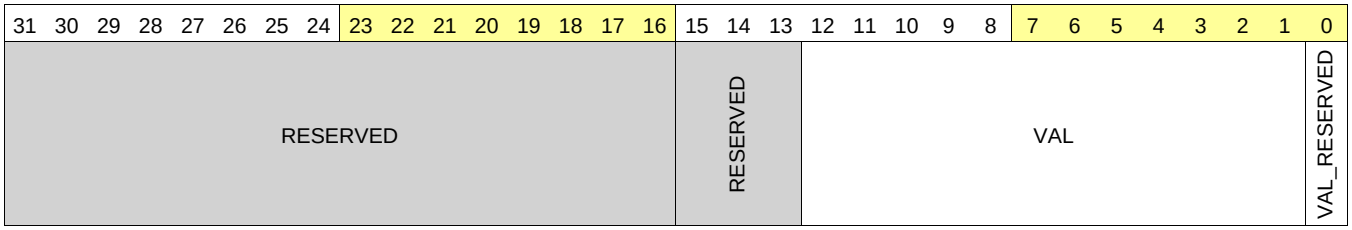


Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The horizontal size of the region 2. The Histogram processing of the region 2 will process (VAL+1) clocks. VAL[0] cannot be written.	RW	0x000
0	VAL_RESERVED	The horizontal size of the region 2. The Histogram processing of the region 2 will process (VAL+1) clocks. VAL[0] cannot be written.	R returns 1s	1

3.5.4.230 IPIPE_HST_3_VPS

Table 616. IPIPE_HST_3_VPS

Address Offset	0x0000 03DC		
Physical Address	0x5505 0BDC 0x5C01 0BDC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		

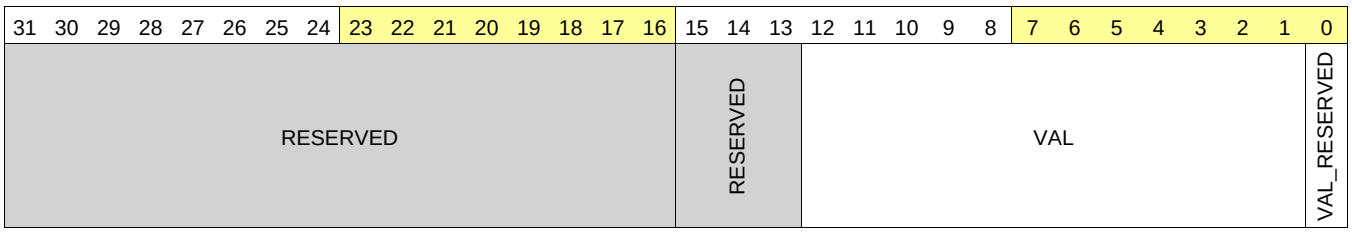


Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The vertical position of the region 0 from the IPIPE_SRC_VPS. The region 3 will start the Histogram processing from VAL line. VAL[0] can not be written.	RW	0x000
0	VAL_RESERVED	The vertical position of the region 0 from the IPIPE_SRC_VPS. The region 3 will start the Histogram processing from VAL line. VAL[0] can not be written.	R	0

3.5.4.231 IPIPE_HST_3_VSZ

Table 617. IPIPE_HST_3_VSZ

Address Offset	0x0000 03E0		
Physical Address	0x5505 0BE0 0x5C01 0BE0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		

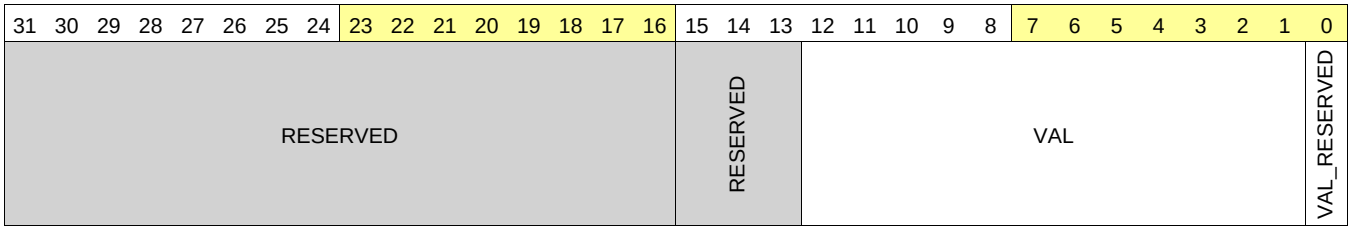


Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The vertical size of the region 3. The Histogram processing of the region 3 will process (VAL+1) lines. VAL[0] cannot be written.	RW	0x000
0	VAL_RESERVED	The vertical size of the region 3. The Histogram processing of the region 3 will process (VAL+1) lines. VAL[0] cannot be written.	R returns 1s	1

3.5.4.232 IPIPE_HST_3_HPS

Table 618. IPIPE_HST_3_HPS

Address Offset	0x0000 03E4		
Physical Address	0x5505 0BE4 0x5C01 0BE4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		

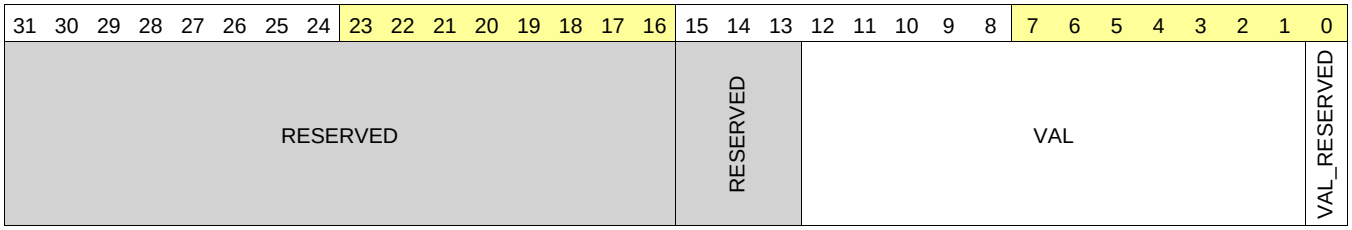


Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The horizontal position of the region 0 from the IPIPE_SRC_HPS. The region 3 will start the Histogram processing from VAL clocks. VAL[0] can not be written.	RW	0x000
0	VAL_RESERVED	The horizontal position of the region 0 from the IPIPE_SRC_HPS. The region 3 will start the Histogram processing from VAL clocks. VAL[0] can not be written.	R	0

3.5.4.233 IPIPE_HST_3_HSZ

Table 619. IPIPE_HST_3_HSZ

Address Offset	0x0000 03E8		
Physical Address	0x5505 0BE8 0x5C01 0BE8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The horizontal size of the region 3. The Histogram processing of the region 3 will process (VAL+1) clocks. VAL[0] cannot be written.	RW	0x000
0	VAL_RESERVED	The horizontal size of the region 3. The Histogram processing of the region 3 will process (VAL+1) clocks. VAL[0] cannot be written.	R returns 1s	1

3.5.4.234 IPIPE_HST_TBL

Table 620. IPIPE_HST_TBL

Address Offset	0x0000 03EC		
Physical Address	0x5505 0BEC 0x5C01 0BEC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														CLR	SEL

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1	CLR	Histogram memory clear. The histogram can be cleared before the start of operations. However, the clear takes 512 cycles and therefore: + if line size 512, the first line must not be used for histogram computation. + if line size 512, ceil (512/line size) lines must not be used for histogram computation. It's the programmer's responsibility to set the histogram computation area outside the "clear" area. 0x0: Disable 0x1: Enable	RW	0
0	SEL	This bit must be used to select which memory is used to store the histogram data. By selecting alternatively one or the other bit, one can double buffer the histogram output buffer. The 4 KB memory can either be read by the CPU or a DMA request. 0x0: Use Table 0 and 1 = 4KB in the memory ISP map. 0x1: Use Table 2 and 3 = 4KB in the memory ISP map.	RW	0

3.5.4.235 IPIPE_HST_MUL_R

Table 621. IPIPE_HST_MUL_R

Address Offset	0x0000 03F0		
Physical Address	0x5505 0BF0 0x5C01 0BF0	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								GAIN							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	GAIN	Gain	RW	0x00

3.5.4.236 IPIPE_HST_MUL_GR

Table 622. IPIPE_HST_MUL_GR

Address Offset	0x0000 03F4		
Physical Address	0x5505 0BF4 0x5C01 0BF4	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								GAIN							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	GAIN	Gain	RW	0x00

3.5.4.237 IPIPE_HST_MUL_GB

Table 623. IPIPE_HST_MUL_GB

Address Offset	0x0000 03F8		
Physical Address	0x5505 0BF8 0x5C01 0BF8	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								GAIN							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	GAIN	Gain	RW	0x00

3.5.4.238 IPIPE_HST_MUL_B

Table 624. IPIPE_HST_MUL_B

Address Offset	0x0000 03FC		
Physical Address	0x5505 0BFC 0x5C01 0BFC	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Histogram		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								GAIN							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	GAIN	Gain	RW	0x00

3.5.4.239 IPIPE_BSC_EN

Table 625. IPIPE_BSC_EN

Address Offset	0x0000 0400		
Physical Address	0x5505 0C00 0x5C01 0C00	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boundary Signal Calculator		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	EN	The start flag of the Boundary Signal Calculator. 0x0: Disable 0x1: Enable	RW	0

3.5.4.240 IPIPE_BSC_MODE

Table 626. IPIPE_BSC_MODE

Address Offset	0x0000 0404		
Physical Address	0x5505 0C04 0x5C01 0C04	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boundary Signal Calculator		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															OST

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	OST	The processing mode selection of the BSC module. 0x0: Free run 0x1: One shot	RW	0

3.5.4.241 IPIPE_BSC_TYP

Table 627. IPIPE_BSC_TYP

Address Offset	0x0000 0408		
Physical Address	0x5505 0C08 0x5C01 0C08	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boundary Signal Calculator		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED												CEN	REN	COL	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:4	RESERVED		R	0x000
3	CEN	Enable of column sampling. 0x0: Off 0x1: On	RW	0
2	REN	Enable of row sampling 0x0: Off 0x1: On	RW	0
1:0	COL	Selects the element to be summed. (Y, Cb, or Cr) 0x0: Y 0x1: Cb 0x2: Cr 0x3: Reserved	RW	0x0

3.5.4.242 IPIPE_BSC_ROW_VCT

Table 628. IPIPE_BSC_ROW_VCT

Address Offset	0x0000 040C		
Physical Address	0x5505 0C0C 0x5C01 0C0C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boundary Signal Calculator		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														VAL	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1:0	VAL	The number of row sum vectors. value = ROWNUM + 1 Range: 1-4	RW	0x0

3.5.4.243 IPIPE_BSC_ROW_SHF

Table 629. IPIPE_BSC_ROW_SHF

Address Offset	0x0000 0410		
Physical Address	0x5505 0C10 0x5C01 0C10	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boundary Signal Calculator		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														VAL	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:3	RESERVED		R	0x0000
2:0	VAL	The down shift value for row sum vectors. Range: 0-7	RW	0x0

3.5.4.244 IPIPE_BSC_ROW_VPOS

Table 630. IPIPE_BSC_ROW_VPOS

Address Offset	0x0000 0414		
Physical Address	0x5505 0C14 0x5C01 0C14	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boundary Signal Calculator		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	The vertical position of the first sampling pixel; the first row to be summed.	RW	0x0000

3.5.4.245 IPIPE_BSC_ROW_VNUM

Table 631. IPIPE_BSC_ROW_VNUM

Address Offset	0x0000 0418		
Physical Address	0x5505 0C18 0x5C01 0C18	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boundary Signal Calculator		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								VAL							VAL_0

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:1	VAL	The height of the area covered by a row sum vector. Height of the region = (ROW_VNUM + 1) × (ROW_VSKIP+1) IPIPE_BSC_ROW_VNUM must be odd, LSB = 1.	RW	0x000
0	VAL_0	The LSB must be odd.	R	1

3.5.4.246 IPIPE_BSC_ROW_VSKIP

Table 632. IPIPE_BSC_ROW_VSKIP

Address Offset	0x0000 041C		
Physical Address	0x5505 0C1C 0x5C01 0C1C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boundary Signal Calculator		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4:0	VAL	The interval of the rows. Interval = ROWSKIPV + 1 Range: 1-32	RW	0x00

3.5.4.247 IPIPE_BSC_ROW_HPOS

Table 633. IPIPE_BSC_ROW_HPOS

Address Offset	0x0000 0420		
Physical Address	0x5505 0C20 0x5C01 0C20	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boundary Signal Calculator		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			VAL												

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	The horizontal position of the first sampling pixel; the first pixel in a row to be summed.	RW	0x0000

3.5.4.248 IPIPE_BSC_ROW_HNUM

Table 634. IPIPE_BSC_ROW_HNUM

Address Offset	0x0000 0424		
Physical Address	0x5505 0C24 0x5C01 0C24	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boundary Signal Calculator		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	The horizontal number of samples in the area covered by a row sum vector. Width of the region = (ROW_HNUM + 1) × (ROW_HSKIP+1)	RW	0x0000

3.5.4.249 IPIPE_BSC_ROW_HSKIP

Table 635. IPIPE_BSC_ROW_HSKIP

Address Offset	0x0000 0428		
Physical Address	0x5505 0C28 0x5C01 0C28	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boundary Signal Calculator		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED													VAL		

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4:0	VAL	The interval of the pixels in a row to be summed. Interval = ROWSKIPH + 1 Range: 1-32	RW	0x00

3.5.4.250 IPIPE_BSC_COL_VCT

Table 636. IPIPE_BSC_COL_VCT

Address Offset	0x0000 042C		
Physical Address	0x5505 0C2C 0x5C01 0C2C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boundary Signal Calculator		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														VAL	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1:0	VAL	The number of column sum vectors. number = COLNUM + 1 Range: 1-4	RW	0x0

3.5.4.251 IPIPE_BSC_COL_SHF

Table 637. IPIPE_BSC_COL_SHF

Address Offset	0x0000 0430		
Physical Address	0x5505 0C30 0x5C01 0C30	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boundary Signal Calculator		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														VAL	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:3	RESERVED		R	0x0000
2:0	VAL	The down shift value for column sum vectors. Range: 0-7	RW	0x0

3.5.4.252 IPIPE_BSC_COL_VPOS

Table 638. IPIPE_BSC_COL_VPOS

Address Offset	0x0000 0434		
Physical Address	0x5505 0C34 0x5C01 0C34	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boundary Signal Calculator		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	The vertical position of the first sampling pixel; the first pixel in a column to be summed.	RW	0x0000

3.5.4.253 IPIPE_BSC_COL_VNUM

Table 639. IPIPE_BSC_COL_VNUM

Address Offset	0x0000 0438		
Physical Address	0x5505 0C38 0x5C01 0C38	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boundary Signal Calculator		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	The vertical number of samples in the area covered by a column sum vector. Height of the region = (COL_VNUM + 1) × (COL_VSKIP + 1)	RW	0x0000

3.5.4.254 IPIPE_BSC_COL_VSKIP

Table 640. IPIPE_BSC_COL_VSKIP

Address Offset	0x0000 043C		
Physical Address	0x5505 0C3C 0x5C01 0C3C	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boundary Signal Calculator		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								VAL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4:0	VAL	The interval of the pixels in a column to be summed. Interval = COLSKIPV + 1 Range: 1-32	RW	0x00

3.5.4.255 IPIPE_BSC_COL_HPOS

Table 641. IPIPE_BSC_COL_HPOS

Address Offset	0x0000 0440		
Physical Address	0x5505 0C40 0x5C01 0C40	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boundary Signal Calculator		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			VAL												

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	The horizontal position of the first sampling pixel; the first column to be summed.	RW	0x0000

3.5.4.256 IPIPE_BSC_COL_HNUM

Table 642. IPIPE_BSC_COL_HNUM

Address Offset	0x0000 0444		
Physical Address	0x5505 0C44 0x5C01 0C44	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boundary Signal Calculator		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								VAL							VAL_0

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:1	VAL	The width of the area covered by a column sum vector. Width of the region = (COL_HNUM + 1) × (COL_HSKIP + 1) IPIPE_BSC_COL_HNUM must be odd, LSB = 1.	RW	0x000
0	VAL_0	The LSB must be odd	R	1

3.5.4.257 IPIPE_BSC_COL_HSKIP

Table 643. IPIPE_BSC_COL_HSKIP

Address Offset	0x0000 0448		
Physical Address	0x5505 0C48 0x5C01 0C48	**Instance**	ISS_IPIPE_CORTEX-M3 ISS_IPIPE_L3
Description	Boundary Signal Calculator		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED													VAL		

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4:0	VAL	The interval of the columns. Interval = COL_HSKIP + 1 Range: 1-32	RW	0x00

3.5.5 ISS ISIF Registers

CAUTION

The ISS ISIF registers are limited to 32-bit and 16-bit data accesses; 8-bit data access is not allowed and can corrupt register content.

Table 644 summarizes the ISS ISIF registers.

Table 644. ISS ISIF Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ISS_ISIF Base Address Cortex-M3 Private Access	ISS_ISIF Base Address L3 Interconnect
ISIF_SYNCEN	RW	32	0x0000 0000	0x5505 1000	0x5C01 1000
ISIF_MODESET	RW	32	0x0000 0004	0x5505 1004	0x5C01 1004
ISIF_HDW	RW	32	0x0000 0008	0x5505 1008	0x5C01 1008
ISIF_VDW	RW	32	0x0000 000C	0x5505 100C	0x5C01 100C
ISIF_PPLN	RW	32	0x0000 0010	0x5505 1010	0x5C01 1010
ISIF_LPFR	RW	32	0x0000 0014	0x5505 1014	0x5C01 1014
ISIF_SPH	RW	32	0x0000 0018	0x5505 1018	0x5C01 1018
ISIF_LNH	RW	32	0x0000 001C	0x5505 101C	0x5C01 101C
RESERVED	RW	32	0x0000 0020	0x5505 1020	0x5C01 1020
RESERVED	RW	32	0x0000 0024	0x5505 1024	0x5C01 1024
ISIF_LNV	RW	32	0x0000 0028	0x5505 1028	0x5C01 1028
ISIF_CULH	RW	32	0x0000 002C	0x5505 102C	0x5C01 102C
ISIF_CULV	RW	32	0x0000 0030	0x5505 1030	0x5C01 1030
ISIF_HSIZE	RW	32	0x0000 0034	0x5505 1034	0x5C01 1034
RESERVED	RW	32	0x0000 0038	0x5505 1038	0x5C01 1038
ISIF_CADU	RW	32	0x0000 003C	0x5505 103C	0x5C01 103C
ISIF_CADL	RW	32	0x0000 0040	0x5505 1040	0x5C01 1040
ISIF_LINCFG0	RW	32	0x0000 0044	0x5505 1044	0x5C01 1044
ISIF_LINCFG1	RW	32	0x0000 0048	0x5505 1048	0x5C01 1048
ISIF_CCOLP	RW	32	0x0000 004C	0x5505 104C	0x5C01 104C
ISIF_CRGAIN	RW	32	0x0000 0050	0x5505 1050	0x5C01 1050
ISIF_CGRGAIN	RW	32	0x0000 0054	0x5505 1054	0x5C01 1054
ISIF_CGBGAIN	RW	32	0x0000 0058	0x5505 1058	0x5C01 1058
ISIF_CBGAIN	RW	32	0x0000 005C	0x5505 105C	0x5C01 105C
ISIF_COFSTA	RW	32	0x0000 0060	0x5505 1060	0x5C01 1060
ISIF_VDINT0	RW	32	0x0000 0070	0x5505 1070	0x5C01 1070
ISIF_VDINT1	RW	32	0x0000 0074	0x5505 1074	0x5C01 1074
ISIF_VDINT2	RW	32	0x0000 0078	0x5505 1078	0x5C01 1078
ISIF_MISC	RW	32	0x0000 007C	0x5505 107C	0x5C01 107C
ISIF_CGAMMAWD	RW	32	0x0000 0080	0x5505 1080	0x5C01 1080
ISIF_REC656IF	RW	32	0x0000 0084	0x5505 1084	0x5C01 1084
ISIF_CCDCFG	RW	32	0x0000 0088	0x5505 1088	0x5C01 1088
ISIF_DFCCTL	RW	32	0x0000 008C	0x5505 108C	0x5C01 108C
ISIF_VDFSATLV	RW	32	0x0000 0090	0x5505 1090	0x5C01 1090
ISIF_DFCMEMCTL	RW	32	0x0000 0094	0x5505 1094	0x5C01 1094
ISIF_DFCMEM0	RW	32	0x0000 0098	0x5505 1098	0x5C01 1098
ISIF_DFCMEM1	RW	32	0x0000 009C	0x5505 109C	0x5C01 109C

Table 644. ISS ISIF Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	ISS_ISIF Base Address Cortex-M3 Private Access	ISS_ISIF Base Address L3 Interconnect
ISIF_DFCMEM2	RW	32	0x0000 00A0	0x5505 10A0	0x5C01 10A0
ISIF_DFCMEM3	RW	32	0x0000 00A4	0x5505 10A4	0x5C01 10A4
ISIF_DFCMEM4	RW	32	0x0000 00A8	0x5505 10A8	0x5C01 10A8
ISIF_CLAMPCFG	RW	32	0x0000 00AC	0x5505 10AC	0x5C01 10AC
ISIF_CLDCOFST	RW	32	0x0000 00B0	0x5505 10B0	0x5C01 10B0
ISIF_CLSV	RW	32	0x0000 00B4	0x5505 10B4	0x5C01 10B4
ISIF_CLHWIN0	RW	32	0x0000 00B8	0x5505 10B8	0x5C01 10B8
ISIF_CLHWIN1	RW	32	0x0000 00BC	0x5505 10BC	0x5C01 10BC
ISIF_CLHWIN2	RW	32	0x0000 00C0	0x5505 10C0	0x5C01 10C0
ISIF_CLVRV	RW	32	0x0000 00C4	0x5505 10C4	0x5C01 10C4
ISIF_CLVWIN0	RW	32	0x0000 00C8	0x5505 10C8	0x5C01 10C8
ISIF_CLVWIN1	RW	32	0x0000 00CC	0x5505 10CC	0x5C01 10CC
ISIF_CLVWIN2	RW	32	0x0000 00D0	0x5505 10D0	0x5C01 10D0
ISIF_CLVWIN3	RW	32	0x0000 00D4	0x5505 10D4	0x5C01 10D4
ISIF_LSCHOFST	RW	32	0x0000 00D8	0x5505 10D8	0x5C01 10D8
ISIF_LSCVOFST	RW	32	0x0000 00DC	0x5505 10DC	0x5C01 10DC
ISIF_LSCHVAL	RW	32	0x0000 00E0	0x5505 10E0	0x5C01 10E0
ISIF_LSCVVAL	RW	32	0x0000 00E4	0x5505 10E4	0x5C01 10E4
ISIF_2DLSCCFG	RW	32	0x0000 00E8	0x5505 10E8	0x5C01 10E8
ISIF_2DLSCOFST	RW	32	0x0000 00EC	0x5505 10EC	0x5C01 10EC
ISIF_2DLSCINI	RW	32	0x0000 00F0	0x5505 10F0	0x5C01 10F0
ISIF_2DLSCGRBU	RW	32	0x0000 00F4	0x5505 10F4	0x5C01 10F4
ISIF_2DLSCGRBL	RW	32	0x0000 00F8	0x5505 10F8	0x5C01 10F8
ISIF_2DLSCGROF	RW	32	0x0000 00FC	0x5505 10FC	0x5C01 10FC
ISIF_2DLSCORBU	RW	32	0x0000 0100	0x5505 1100	0x5C01 1100
ISIF_2DLSCORBL	RW	32	0x0000 0104	0x5505 1104	0x5C01 1104
ISIF_2DLSCOROF	RW	32	0x0000 0108	0x5505 1108	0x5C01 1108
ISIF_2DLSCIRQEN	RW	32	0x0000 010C	0x5505 110C	0x5C01 110C
ISIF_2DLSCIRQST	RW	32	0x0000 0110	0x5505 1110	0x5C01 1110
ISIF_FMTCFG	RW	32	0x0000 0114	0x5505 1114	0x5C01 1114
ISIF_FMTPLEN	RW	32	0x0000 0118	0x5505 1118	0x5C01 1118
ISIF_FMTSPH	RW	32	0x0000 011C	0x5505 111C	0x5C01 111C
ISIF_FMTLNH	RW	32	0x0000 0120	0x5505 1120	0x5C01 1120
ISIF_FMTLSV	RW	32	0x0000 0124	0x5505 1124	0x5C01 1124
ISIF_FMTLNV	RW	32	0x0000 0128	0x5505 1128	0x5C01 1128
ISIF_FMTRLEN	RW	32	0x0000 012C	0x5505 112C	0x5C01 112C
ISIF_FMTHCNT	RW	32	0x0000 0130	0x5505 1130	0x5C01 1130
ISIF_FMTAPTR0	RW	32	0x0000 0134	0x5505 1134	0x5C01 1134
ISIF_FMTAPTR1	RW	32	0x0000 0138	0x5505 1138	0x5C01 1138
ISIF_FMTAPTR2	RW	32	0x0000 013C	0x5505 113C	0x5C01 113C
ISIF_FMTAPTR3	RW	32	0x0000 0140	0x5505 1140	0x5C01 1140
ISIF_FMTAPTR4	RW	32	0x0000 0144	0x5505 1144	0x5C01 1144
ISIF_FMTAPTR5	RW	32	0x0000 0148	0x5505 1148	0x5C01 1148
ISIF_FMTAPTR6	RW	32	0x0000 014C	0x5505 114C	0x5C01 114C
ISIF_FMTAPTR7	RW	32	0x0000 0150	0x5505 1150	0x5C01 1150

Table 644. ISS ISIF Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	ISS_ISIF Base Address Cortex-M3 Private Access	ISS_ISIF Base Address L3 Interconnect
ISIF_FMTAPTR8	RW	32	0x0000 0154	0x5505 1154	0x5C01 1154
ISIF_FMTAPTR9	RW	32	0x0000 0158	0x5505 1158	0x5C01 1158
ISIF_FMTAPTR10	RW	32	0x0000 015C	0x5505 115C	0x5C01 115C
ISIF_FMTAPTR11	RW	32	0x0000 0160	0x5505 1160	0x5C01 1160
ISIF_FMTAPTR12	RW	32	0x0000 0164	0x5505 1164	0x5C01 1164
ISIF_FMTAPTR13	RW	32	0x0000 0168	0x5505 1168	0x5C01 1168
ISIF_FMTAPTR14	RW	32	0x0000 016C	0x5505 116C	0x5C01 116C
ISIF_FMTAPTR15	RW	32	0x0000 0170	0x5505 1170	0x5C01 1170
ISIF_FMTPGMVF0	RW	32	0x0000 0174	0x5505 1174	0x5C01 1174
ISIF_FMTPGMVF1	RW	32	0x0000 0178	0x5505 1178	0x5C01 1178
ISIF_FMTPGMAPU0	RW	32	0x0000 017C	0x5505 117C	0x5C01 117C
ISIF_FMTPGMAPU1	RW	32	0x0000 0180	0x5505 1180	0x5C01 1180
ISIF_FMTPGMAPS0	RW	32	0x0000 0184	0x5505 1184	0x5C01 1184
ISIF_FMTPGMAPS1	RW	32	0x0000 0188	0x5505 1188	0x5C01 1188
ISIF_FMTPGMAPS2	RW	32	0x0000 018C	0x5505 118C	0x5C01 118C
ISIF_FMTPGMAPS3	RW	32	0x0000 0190	0x5505 1190	0x5C01 1190
ISIF_FMTPGMAPS4	RW	32	0x0000 0194	0x5505 1194	0x5C01 1194
ISIF_FMTPGMAPS5	RW	32	0x0000 0198	0x5505 1198	0x5C01 1198
ISIF_FMTPGMAPS6	RW	32	0x0000 019C	0x5505 119C	0x5C01 119C
ISIF_FMTPGMAPS7	RW	32	0x0000 01A0	0x5505 11A0	0x5C01 11A0
ISIF_CSCCTL	RW	32	0x0000 01A4	0x5505 11A4	0x5C01 11A4
ISIF_CSCM0	RW	32	0x0000 01A8	0x5505 11A8	0x5C01 11A8
ISIF_CSCM1	RW	32	0x0000 01AC	0x5505 11AC	0x5C01 11AC
ISIF_CSCM2	RW	32	0x0000 01B0	0x5505 11B0	0x5C01 11B0
ISIF_CSCM3	RW	32	0x0000 01B4	0x5505 11B4	0x5C01 11B4
ISIF_CSCM4	RW	32	0x0000 01B8	0x5505 11B8	0x5C01 11B8
ISIF_CSCM5	RW	32	0x0000 01BC	0x5505 11BC	0x5C01 11BC
ISIF_CSCM6	RW	32	0x0000 01C0	0x5505 11C0	0x5C01 11C0
ISIF_CSCM7	RW	32	0x0000 01C4	0x5505 11C4	0x5C01 11C4
ISIF_CLKCTL	RW	32	0x0000 01F8	0x5505 11F8	0x5C01 11F8

3.5.5.1 ISIF_SYNCEN

Table 645. ISIF_SYNCEN

Address Offset	0x0000 0000		
Physical Address	0x5505 1000 0x5C01 1000	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														DWEN	SYEN

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000
1	DWEN	Controls the storage of image sensor RAW data in memory. This bit is loaded with the timing of the internal VD signal: it becomes active starting at the lead of the VD signal that comes after 1 is written in this bit. 0x0: Disable 0x1: Enable	RW	0
0	SYEN	Controls ON/OFF of VD/HD output. Internal timing generator becomes active and VD/HD output starts when 1 is written in this bit. In case if input, VD/HD loading begins. 0: Disable 1: Enable	RW	0

3.5.5.2 ISIF_MODESET

Table 646. ISIF_MODESET

Address Offset	0x0000 0004		
Physical Address	0x5505 1004 0x5C01 1004	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	HLPF	INPMOD		OVF	CCDW			CCDMD	DPOL	SWEN	RESERVED	HDPOL	VDPOL	FIDD	HDVDD

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	RESERVED	Read returns reset value	R	0
14	HLPF	Low pass filter enable. When this bit is enabled, a 3-tap (1/4 + 1/2 Z^{-2} + 1/4 Z^{4}) filtering process is performed on the sensor data. 0x0: Disable 0x1: Enable	RW	0
13:12	INPMOD	Data input mode: 0x0: RAW data 0x1: YCbCr 16bit 0x2: YCbCr 8bit 0x3: Reserved	RW	0x2
11	OVF	ISIF module write port overflow status bit If the write port of the ISIF module overflows when writing data to SDRAM, this bit will toggle. 0x0: No overflow pending (r) No action (w) 0x1: Overflow pending (r) Clear overflow (w)	RW	0
10:8	CCDW	This bit enables to shift right (divide) the up-to-12-bit RAW data value when writing out to SDRAM. The effect is that the dynamic of the output signal is decreased. The ISIF_MODESET.CCDW, ISIF_HSIZE.ADCR, ISIF_HSIZE.HSIZE, ISIF_CCDCFG.BSWD, ISIF_CCDCFG.MSBINV, ISIF_CCDCFG.SDRPACK bit fields control how pixel data are stored to SDRAM. 0x0: No shift out[15:0] = 0000 data[11:0] 0x1: 1-bit right shift out[15:0] = 00000 data[11:1] 0x2: 2-bit right shift out[15:0] = 000000 data[11:2] 0x3: 3-bit right shift out[15:0] = 0000000 data[11:3] 0x4: 4-bit right shift out[15:0] = 00000000 data[11:4] 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x0

Bits	Field Name	Description	Type	Reset
7	CCDMD	Field mode: This bit selects the type of image sensor: interlaced or progressive 0x0: Progressive image sensor 0x1: Interlaced image sensor	RW	0
6	DPOL	Image sensor input data polarity 0x0: No change 0x1: One's complement	RW	0
5	SWEN	External WEN selection In case this bit and SYNCEN.DWEN are set to 1, the external WEN signal is used to store image sensor data to memory. 0x0: WEN not used 0x1: Use external WEN	RW	0
4	RESERVED	Read returns reset value	RW	0
3	HDPOL	HD Sync Signal Polarity 0x0: Positive 0x1: Negative	RW	0
2	VDPOL	VD Sync Signal Polarity 0x0: Positive 0x1: Negative	RW	0
1	FIDD	FLD Signal Direction. There must be at least three clock cycles between the time this bit is modified and the HD/VD pulse for the start of frame comes. 0x0: Input 0x1: Output	RW	0
0	HDVDD	VD,HD Sync Signal Direction. There must be at least three clock cycles between the time this bit is modified and the HD/VD pulse for the start of frame comes. 0x0: Input 0x1: Output	RW	0

3.5.5.3 ISIF_HDW

Table 647. ISIF_HDW

Address Offset	0x0000 0008		
Physical Address	0x5505 1008 0x5C01 1008	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				HDW											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	HDW	HD width: Sets width of HD. HD width = HDW + 1 clock	RW	0x000

3.5.5.4 ISIF_VDW

Table 648. ISIF_VDW

Address Offset	0x0000 000C		
Physical Address	0x5505 100C 0x5C01 100C	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VDW											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VDW	VD width: Sets width of VD. VD width = VDW + 1 line	RW	0x000

3.5.5.5 ISIF_PPLN

Table 649. ISIF_PPLN

Address Offset	0x0000 0010		
Physical Address	0x5505 1010 0x5C01 1010	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PPLN															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	PPLN	Pixels per line Number of pixel clock periods in one line HD period = PPLN+1 pixel clocks. PPLN is not used when HD and VD are inputs, that is, when VDHDOUT in MODESET is cleared to 0. *This bit field is latched by VD.	RW	0x0000

3.5.5.6 ISIF_LPFR

Table 650. ISIF_LPFR

Address Offset	0x0000 0014		
Physical Address	0x5505 1014 0x5C01 1014	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Line per Frame/Field		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LPFR															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	LPFR	Half lines per filed or frame Sets number of half lines per frame or field. VD period = (LPFR+1)/2 lines. LPFR is not used when HD and are inputs, that is, when VDHDOUT in MODESET is cleared to 0. *This bit field is latched by VD.	RW	0x0000

3.5.5.7 ISIF_SPH

Table 651. ISIF_SPH

Address Offset	0x0000 0018		
Physical Address	0x5505 1018 0x5C01 1018	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Start Pixel Horizontal		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	SPH														

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	RESERVED		R	0
14:0	SPH	The first pixel in a line to be stored to memory.	RW	0x0000

3.5.5.8 ISIF_LNH

Table 652. ISIF_LNH

Address Offset	0x0000 001C		
Physical Address	0x5505 101C 0x5C01 101C	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	LNH														

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	RESERVED		R	0
14:0	LNH	Number of pixels in an line to be stored to memory. Number of pixels = LNH + 1.	RW	0x0000

3.5.5.9 ISIF_LNV

Table 653. ISIF_LNV

Address Offset	0x0000 0028		
Physical Address	0x5505 1028 0x5C01 1028	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	LNV														

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	RESERVED	The number of lines to be stored to SDRAM.	R	0
14:0	LNV	The number of lines to be stored to memory. Number of lines = LNV + 1	RW	0x0000

3.5.5.10 ISIF_CULH

Table 654. ISIF_CULH

Address Offset	0x0000 002C		
Physical Address	0x5505 102C 0x5C01 102C	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLHO								CLHE							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	CLHO	Culling Pattern in ODD Line: Sets culling pattern when data is loaded into memory (odd lines). Example: 0xAA: 1 / 2 horizontal direction culling. LSB becomes left side on screen. 0x0: Pixel invalid 0x1: Pixel valid	RW	0xFF
7:0	CLHE	Culling Pattern in Even Line: Sets culling pattern when data is loaded into memory (even lines). 0x0: Pixel invalid 0x1: Pixel valid	RW	0xFF

3.5.5.11 ISIF_CULV

Table 655. ISIF_CULV

Address Offset	0x0000 0030		
Physical Address	0x5505 1030 0x5C01 1030	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								CULV							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	CULV	Culling Pattern in Vertical Line Example: 0x88: 1/4 vertical direction culling. LSB becomes top side on screen. 0x0: Pixel invalid 0x1: Pixel valid	RW	0xFF

3.5.5.12 ISIF_HSIZE

Table 656. ISIF_HSIZE

Address Offset	0x0000 0034		
Physical Address	0x5505 1034 0x5C01 1034	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	SDRAM OUTPUT CTRL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			ADCR	HSIZE											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12	ADCR	SDRAM address decrement. By setting this bit, memory address in a line is automatically decreased so that a line can be Horizontally flipped in memory. The ISIF_MODESET.CCDW, ISIF_HSIZE.ADCR, ISIF_HSIZE.HSIZE, ISIF_CCDCFG.BSWD, ISIF_CCDCFG.MSBINV, ISIF_CCDCFG.SDRPACK bit fields control how pixel data are stored to SDRAM. 0x0: Address increment. 0x1: Address decrement.	RW	0
11:0	HSIZE	Memory address offset between the lines. Specify the offset in 32-byte units.	RW	0x000

3.5.5.13 ISIF_CADU

Table 657. ISIF_CADU

Address Offset	0x0000 003C		
Physical Address	0x5505 103C 0x5C01 103C	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	SDRAM OUTPUT CTRL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								CADU							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	CADU	Memory Address (Upper 11-bits): Specifies the memory destination address. The actual address is the value set here multiplied by 32bytes.	RW	0x000

3.5.5.14 ISIF_CADL

Table 658. ISIF_CADL

Address Offset	0x0000 0040		
Physical Address	0x5505 1040 0x5C01 1040	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	SDRAM OUTPUT CTRL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CADL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	CADL	Memory Address (Lower 16-bits): Specifies the memory destination address. The actual address is the value set here multiplied by 32bytes.	RW	0x0000

3.5.5.15 ISIF_LINCFG0

Table 659. ISIF_LINCFG0

Address Offset	0x0000 0044		
Physical Address	0x5505 1044 0x5C01 1044	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	INPUT LINEARIZATION CTRL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								CORRSFT				RESERVED		LINMD	LINEN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:7	RESERVED		R	0x000
6:4	CORRSFT	Shift up value for the correction value (S10). 0x0: No shift 0x1: 1-bit left shift 0x2: 2-bit left shift 0x3: 3-bit left shift 0x4: 4-bit left shift 0x5: 5-bit left shift 0x6: 6-bit left shift 0x7: Reserved	RW	0x0
3:2	RESERVED		R	0x0
1	LINMD	Linearization Mode: 0x0: Uniform sampling 0x1: Non-uniform sampling	RW	0
0	LINEN	Linearization Enable: 0x0: Disable 0x1: Enable	RW	0

3.5.5.16 ISIF_LINCFG1

Table 660. ISIF_LINCFG1

Address Offset	0x0000 0048		
Physical Address	0x5505 1048 0x5C01 1048	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	INPUT LINEARIZATION CTRL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								LUTSCL							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	LUTSCL	Scale factor (U11Q10) for LUT input. Range: 0 - 1+1023/1024 It is applied to the Input Data before looking up the correction factor. The scale factor is only applied to the table input. It is not applied when using the input value to compute the output.	RW	0x400

3.5.5.17 ISIF_CCOLP

Table 661. ISIF_CCOLP

Address Offset	0x0000 004C		
Physical Address	0x5505 104C 0x5C01 104C	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CP0_F1		CP1_F1		CP2_F1		CP3_F1		CP0_F0		CP1_F0		CP2_F0		CP3_F0	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:14	CP0_F1	Specifies color pattern for pixel position 0 (Field 1) Pixel position 0 corresponds to pixel count=0 at even line in case of CFAP= 0, and to pixel count=0 in case of CFAP= 1. 0x0: R / Ye 0x1: Gr / Cy 0x2: Gb / G 0x3: B / Mg	RW	0x0
13:12	CP1_F1	Specifies color pattern for pixel position 1 (Field 1) Pixel position 1 corresponds to pixel count=1 at even line in case of CFAP= 0, and to pixel count=1 in case of CFAP= 1. 0x0: R / Ye 0x1: Gr / Cy 0x2: Gb / G 0x3: B / Mg	RW	0x0
11:10	CP2_F1	Specifies color pattern for pixel position 2 (Field 1) Pixel position 2 corresponds to pixel count=0 at odd line in case of CFAP= 0, and to pixel count=2 in case of CFAP= 1. 0x0: R / Ye 0x1: Gr / Cy 0x2: Gb / G 0x3: B / Mg	RW	0x0
9:8	CP3_F1	Specifies color pattern for pixel position 3 (Field 1) Pixel position 3 corresponds to pixel count=1 at odd line in case of CFAP= 0. Not applicable for CFAP= 1. 0x0: R / Ye 0x1: Gr / Cy 0x2: Gb / G 0x3: B / Mg	RW	0x0
7:6	CP0_F0	Specifies color pattern for pixel position 0 (Field 0) Pixel position 0 corresponds to pixel count=0 at even line in case of CFAP= 0, and to pixel count=0 in case of CFAP= 1. 0x0: R / Ye 0x1: Gr / Cy 0x2: Gb / G 0x3: B / Mg	RW	0x0

Bits	Field Name	Description	Type	Reset
5:4	CP1_F0	Specifies color pattern for pixel position 1 (Field 0) Pixel position 1 corresponds to pixel count=1 at even line in case of CFAP= 0, and to pixel count=1 in case of CFAP= 1. 0x0: R / Ye 0x1: Gr / Cy 0x2: Gb / G 0x3: B / Mg	RW	0x0
3:2	CP2_F0	Specifies color pattern for pixel position 2 (Field 0) Pixel position 2 corresponds to pixel count=0 at odd line in case of CFAP= 0, and to pixel count=2 in case of CFAP= 1. 0x0: R / Ye 0x1: Gr / Cy 0x2: Gb / G 0x3: B / Mg	RW	0x0
1:0	CP3_F0	Specifies color pattern for pixel position 3 (Field 0) Pixel position 3 corresponds to pixel count=1 at odd line in case of CFAP= 0. Not applicable for CFAP= 1. 0x0: R / Ye 0x1: Gr / Cy 0x2: Gb / G 0x3: B / Mg	RW	0x0

3.5.5.18 ISIF_CRGAIN

Table 662. ISIF_CRGAIN

Address Offset	0x0000 0050		
Physical Address	0x5505 1050 0x5C01 1050	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				CGR											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	CGR	R/Ye gain: Performs gain adjustment on image sensor data. U12Q9. Range: 0 - 7+511/512	RW	0x200

3.5.5.19 ISIF_CGRGAIN

Table 663. ISIF_CGRGAIN

Address Offset	0x0000 0054		
Physical Address	0x5505 1054 0x5C01 1054	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				CGGR											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	CGGR	Gr/Cy gain: Performs gain adjustment on image sensor data. U12Q9. Range: 0 - 7+511/512	RW	0x200

3.5.5.20 ISIF_CGBGAIN

Table 664. ISIF_CGBGAIN

Address Offset	0x0000 0058		
Physical Address	0x5505 1058 0x5C01 1058	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				CGGB											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	CGGB	Gb/Cy gain: Performs gain adjustment on image sensor data. U12Q9. Range: 0 - 7+511/512	RW	0x200

3.5.5.21 ISIF_CBGAIN

Table 665. ISIF_CBGAIN

Address Offset	0x0000 005C		
Physical Address	0x5505 105C 0x5C01 105C	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				CGB											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	CGB	B/Mg gain: Performs gain adjustment on image sensor data. U12Q9. Range: 0 - 7+511/512	RW	0x200

3.5.5.22 ISIF_COFSTA

Table 666. ISIF_COFSTA

Address Offset	0x0000 0060		
Physical Address	0x5505 1060 0x5C01 1060	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				COFT											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	COFT	Image sensor offset: Performs offset value adjustment on image sensor data (0~4095).	RW	0x000

3.5.5.23 ISIF_VDINT0

Table 667. ISIF_VDINT0

Address Offset	0x0000 0070		
Physical Address	0x5505 1070 0x5C01 1070	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	CVD0														

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	RESERVED		R	0
14:0	CVD0	VD0 Interrupt timing in a field (line number).	RW	0x0000

3.5.5.24 ISIF_VDINT1

Table 668. ISIF_VDINT1

Address Offset	0x0000 0074		
Physical Address	0x5505 1074 0x5C01 1074	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	CVD1														

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	RESERVED		R	0
14:0	CVD1	VD1 Interrupt timing in a field (line number).	RW	0x0000

3.5.5.25 ISIF_VDINT2

Table 669. ISIF_VDINT2

Address Offset	0x0000 0078		
Physical Address	0x5505 1078 0x5C01 1078	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	CVD2														

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	RESERVED		R	0
14:0	CVD2	VD2 Interrupt timing in a field (line number).	RW	0x0000

3.5.5.26 ISIF_MISC

Table 670. ISIF_MISC

Address Offset	0x0000 007C		
Physical Address	0x5505 107C 0x5C01 107C	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		DPCMPRE	DPCMEN	RESERVED											RESERVED

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:14	RESERVED		R	0x0
13	DPCMPRE	Selects Predictor for DPCM Encoder (12-8) 0x0: Predictor 1 0x1: Predictor 2	RW	0
12	DPCMEN	Enables DPCM Encoding (12-8) 0x0: Disable 0x1: Enable	RW	0
11:1	RESERVED		R	0x000
0	RESERVED		RW	0

3.5.5.27 ISIF_CGAMMAWD

Table 671. ISIF_CGAMMAWD

Address Offset	0x0000 0080		
Physical Address	0x5505 1080 0x5C01 1080	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	WBEN2	WBEN1	WBEN0	RESERVED	OFSTEN2	OFSTEN1	OFSTEN0	RESERVED		CFAP	GWDI				CCDTBL

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	RESERVED		R	0
14	WBEN2	White Balance Enable for H3A 0x0: Disable 0x1: Enable	RW	0
13	WBEN1	White Balance Enable for IPIPE 0x0: Disable 0x1: Enable	RW	0
12	WBEN0	White Balance Enable for memory capture 0x0: Disable 0x1: Enable	RW	0
11	RESERVED		R	0
10	OFSTEN2	Offset control Enable for H3A 0x0: Disable 0x1: Enable	RW	0
9	OFSTEN1	Offset control Enable for IPIPE 0x0: Disable 0x1: Enable	RW	0
8	OFSTEN0	Offset control Enable for SDRAM capture 0x0: Disable 0x1: Enable	RW	0
7:6	RESERVED		R	0x0
5	CFAP	Selects CFA pattern 0x0: Mosaic color pattern. It should look like this. G R G R G R G R G R . . . B G B G B G B G B G . . . G R G R G R G R G R . . . . 0x1: Stripe color pattern. It should look like this. R G B R G B R G B . . . R G B R G B R G B . . . R G B R G B R G B	RW	0

Bits	Field Name	Description	Type	Reset
4:1	GWDI	Selects MSB position of Input Data 0x0: bit 15 0x1: bit 14 0x2: bit 13 0x3: bit 12 0x4: bit 11 0x5: bit 10 0x6: bit 9 0x7: bit 8 0x8: bit 7 0x9: Reserved 0xA: Reserved 0xB: Reserved 0xC: Reserved 0xD: Reserved 0xE: Reserved 0xF: Reserved	RW	0x0
0	CCDTBL	On/Off control of A-law table for SDRAM capture 0x0: Disable 0x1: Enable	RW	0

3.5.5.28 ISIF_REC656IF

Table 672. ISIF_REC656IF

Address Offset	0x0000 0084		
Physical Address	0x5505 1084 0x5C01 1084	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	INPUT CONFIG REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														R656ON	ECCFVH

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1	R656ON	CCIR Rec.656 interface mode 0x0: Disable 0x1: Enable	RW	0
0	ECCFVH	Error correction of FVH code 0x0: Disable 0x1: Enable	RW	0

3.5.5.29 ISIF_CCDCFG

Table 673. ISIF_CCDCFG

Address Offset	0x0000 0088		
Physical Address	0x5505 1088 0x5C01 1088	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VLDC	RESERVED	MSBINVI	BSWD	Y8POS	EXTRG	TRGSEL	WENLOG	FIDMD		BT656	YCINSWP	RESERVED	RESERVED	SDRPACK	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	VLDC	On/off control of CPU registers resynchronize function by VSYNC. All the others are shadowed registers, where register values are updated at V-sync timing by default. If VDLC=1, ISIF register values are updated immediately after register write just like non-shadowed registers. 0x0: Enable 0x1: Disable	RW	0
14	RESERVED	Reserved. must always be set to 0.	RW	0
13	MSBINVI	MSB inverse of CIN port when the data are captured to SDRAM. The ISIF_MODESET.CCDW, ISIF_HSIZE.ADCR, ISIF_HSIZE.HSIZE, ISIF_CCDCFG.BSWD, ISIF_CCDCFG.MSBINV, ISIF_CCDCFG.SDRPACK bit fields control how pixel data are stored to SDRAM. 0x0: Disable 0x1: Enable	RW	0
12	BSWD	On/off control of Byte SWAP function when SDRAM capturing. The ISIF_MODESET.CCDW, ISIF_HSIZE.ADCR, ISIF_HSIZE.HSIZE, ISIF_CCDCFG.BSWD, ISIF_CCDCFG.MSBINV, ISIF_CCDCFG.SDRPACK bit fields control how pixel data are stored to SDRAM. 0x0: Disable 0x1: Enable (swap)	RW	0
11	Y8POS	Selects Y signal position when in 8bit input mode 0x0: even pixel 0x1: odd pixel	RW	0
10	EXTRG	Setting 1 to this register, the SDRAM address is initialized at the rising edge of FID input signal or DWEN register.	RW	0
9	TRGSEL	Select trigger source signal of SDRAM address initializing in case EXTRG=1. 0x0: DWEN register 0x1: FID input port	RW	0

Bits	Field Name	Description	Type	Reset
8	WENLOG	Specifies the CCD valid area. 0x0: internal valid signal and WEN signal is ANDed logically. 0x1: internal valid signal and WEN signal is ORed logically.	RW	0
7:6	FIDMD	Specifies FID detection mode 0x0: latch the FID at the VSYNC timing 0x1: no latch the FID 0x2: Reserved 0x3: Reserved	RW	0x0
5	BT656	Selects bit width of CCIR656. This bit applies only if ISIF_REC656IF.R656ON = 1. 0x0: 8 bits 0x1: 10 bits	RW	0
4	YCINSWP	The ISIF module has a 16-bit interface. When 16-bit YUV data are input, the luma data (YIN7-0) are expected to be on the 8 MS bits and the chroma (CIN7-0) data are expected to be on the LS bits. This bit enables to swap the 8 MS bits with the 8 LS bits of the interface in case the luma and chroma do not come in the correct order. See the top-level ISIF block diagram. 0x0: YIN7-0 = Y signal / CIN7-0 = C signal 0x1: YIN7-0 = C signal / CIN7-0 = Y signal	RW	0
3	RESERVED	Reserved. must always be set to 0.	RW	0
2	RESERVED	Reserved. must always be set to 0.	RW	0
1:0	SDRPACK	This bit field selects how the data are stored to SDRAM. There can be 8, 12 or 16 bits per pixel. The ISIF_MODESET.CCDW, ISIF_HSIZE.ADCR, ISIF_HSIZE.HSIZE, ISIF_CCDCFG.BSWD, ISIF_CCDCFG.MSBINV, ISIF_CCDCFG.SDRPACK bit fields control how pixel data are stored to SDRAM. 0x0: 16 bits / pixel 0x1: 12 bits / pixel 0x2: 8 bits / pixel 0x3: Reserved	RW	0x0

3.5.5.30 ISIF_DFCCTL

Table 674. ISIF_DFCCTL

Address Offset	0x0000 008C		
Physical Address	0x5505 108C 0x5C01 108C	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	VERTICAL LINE DEFCT CTRL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED					VDFLSFT			VDFCUDA	VDFCSL		VDFCEN	RESERVED			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:8	VDFLSFT	Vertical line Defect level shift value Defect Level (value to be subtracted from the data) is 8bit width, but can be up-shifted up to 6bits by VDFLSFT. Left shift value = VDFLSFT (Range: 0-6) Setting 7 to VDFLSFT is not allowed.	RW	0x0
7	VDFCUDA	Vertical line Defect Correction upper pixels disable. 0x0: The whole line is corrected. 0x1: Pixels upper than the defect are not corrected.	RW	0
6:5	VDFCSL	Vertical line Defect Correction mode select. 0x0: Defect level subtraction. Just fed through if data are saturating. 0x1: Defect level subtraction. Horizontal interpolation ((i-2)+(i+2))/2 if data are saturating. 0x2: Horizontal interpolation ((i-2)+(i+2))/2. 0x3: Reserved	RW	0x0
4	VDFCEN	Vertical line Defect Correction enable. This bit field is latched by VD. 0x0: Disable 0x1: Enable	RW	0
3:0	RESERVED		R	0x0

3.5.5.31 ISIF_VDFSATLV

Table 675. ISIF_VDFSATLV

Address Offset	0x0000 0090		
Physical Address	0x5505 1090 0x5C01 1090	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	VERTICAL LINE DEFCT CTRL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VDFSLV											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VDFSLV	Vertical line Defect Correction saturation level. VDFSLV is U12 (Range: 0 - 4,095).	RW	0x000

3.5.5.32 ISIF_DFCMEMCTL

Table 676. ISIF_DFCMEMCTL

Address Offset	0x0000 0094		
Physical Address	0x5505 1094 0x5C01 1094	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	VERTICAL LINE DEFCT CTRL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED											DFCMCLR	RESERVED	DFCMARST	DFCMRD	DFCMWR

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4	DFCMCLR	Defect correction. Memory clear. Writing 1 to this bit clears the memory contents to all zero. It will be automatically cleared to `0` when the memory clear is completed.	RW	0
3	RESERVED		R	0
2	DFCMARST	Defect correction. Memory address reset. Setting DFCMWR or DFCMRD with LSCMARST set starts memory access to address offset 0. DFCMARST is automatically cleared if data transfer completes. Setting DFCMWR or DFCMRD with LSCMARST cleared starts memory access to the next address. 0x0: Increment the memory address 0x1: Clear the memory address to offset 0	RW	0
1	DFCMRD	Defect correction. Memory read [for debug purpose] Writing 1 to this bit starts reading from the memory. It will be automatically cleared when the data transfer is completed, and the data can be read from DFCMEM4-0.	RW	0
0	DFCMWR	Defect correction. Memory write Writing 1 to this bit starts writing to the memory. It will be automatically cleared when the data transfer is completed. DFCMEM4-0 should be set prior to the memory access.	RW	0

3.5.5.33 ISIF_DFCMEM0

Table 677. ISIF_DFCMEM0

Address Offset	0x0000 0098		
Physical Address	0x5505 1098 0x5C01 1098	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Defect correction memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								DFCMEM0							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	DFCMEM0	Defect correction Memory 0 Sets V position of the defects.	RW	0x0000

3.5.5.34 ISIF_DFCMEM1

Table 678. ISIF_DFCMEM1

Address Offset	0x0000 009C		
Physical Address	0x5505 109C 0x5C01 109C	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Defect correction memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								DFCMEM1							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	DFCMEM1	Defect correction Memory 1 Sets H position of the defects.	RW	0x0000

3.5.5.35 ISIF_DFCMEM2

Table 679. ISIF_DFCMEM2

Address Offset	0x0000 00A0		
Physical Address	0x5505 10A0 0x5C01 10A0	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Defect correction memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								DFCMEM2							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	DFCMEM2	Defect correction Memory 2 Set SUB1: Defect level of the Vertical line defect position (V = Vdefect). DFCMEM2 can be up shifted according to VDFLSFT, and subtracted from the data for Vertical line defect correction.	RW	0x00

3.5.5.36 ISIF_DFCMEM3

Table 680. ISIF_DFCMEM3

Address Offset	0x0000 00A4		
Physical Address	0x5505 10A4 0x5C01 10A4	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Defect correction memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								DFCMEM3							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	DFCMEM3	Defect correction Memory 3 Set SUB2: Defect level of the pixels upper than the Vertical line defect (V Vdefect). DFCMEM3 can be up shifted according to VDFLSFT, and subtracted from the data for Vertical line defect correction.	RW	0x00

3.5.5.37 ISIF_DFCMEM4

Table 681. ISIF_DFCMEM4

Address Offset	0x0000 00A8		
Physical Address	0x5505 10A8 0x5C01 10A8	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Defect correction memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								DFCMEM4							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	DFCMEM4	Defect correction Memory 4 Set SUB3: Defect level of the pixels lower than the Vertical line defect (V Vdefect). DFCMEM4 can be up shifted according to VDFLSFT, and subtracted from the data for Vertical line defect correction.	RW	0x00

3.5.5.38 ISIF_CLAMPCFG

Table 682. ISIF_CLAMPCFG

Address Offset	0x0000 00AC		
Physical Address	0x5505 10AC 0x5C01 10AC	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	BLACK CLAMP CTRL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED											CLMD	RESERVED		CLHMD		CLEN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4	CLMD	Black clamp mode Clamp value can be calculated regardless of the color or can be calculated separately for each 4 colors. 0x0: Clamp value calculated regardless of the pixel color. 0x1: Clamp value calculated separately for each 4 colors.	RW	0
3	RESERVED		R	0
2:1	CLHMD	Horizontal Clamp mode 0x0: Horizontal clamp disabled. Only the Vertical clamp value is subtracted from the Image data. 0x1: Horizontal clamp value calculation enabled. The calculated Horizontal clamp value is subtracted from the Image data along with the Vertical clamp value. 0x2: Horizontal clamp value not updated. The Horizontal clamp value used for the previous image is subtracted from the Image data along with the Vertical clamp value. 0x3: Reserved	RW	0x0
0	CLEN	Black Clamp Enable Enables clamp value to be subtracted from Image data. 0x0: Disable 0x1: Enable	RW	0

3.5.5.39 ISIF_CLDCOFST

Table 683. ISIF_CLDCOFST

Address Offset	0x0000 00B0		
Physical Address	0x5505 10B0 0x5C01 10B0	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	BLACK CLAMP CTRL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								CLDC							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	CLDC	DC offset for black clamp This value is added to the incoming pixels regardless whether optical black clamp is enabled (ISIF_CLAMPCFG.CLEN). This value is in S13Q0 format.	RW	0x0000

3.5.5.40 ISIF_CLSV

Table 684. ISIF_CLSV

Address Offset	0x0000 00B4		
Physical Address	0x5505 10B4 0x5C01 10B4	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	BLACK CLAMP CTRL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								CLSV							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	CLSV	Black Clamp Start position (V). Sets the line number where clamp value subtraction starts. Range: 0 - 8191	RW	0x0000

3.5.5.41 ISIF_CLHWIN0

Table 685. ISIF_CLHWIN0

Address Offset	0x0000 00B8		
Physical Address	0x5505 10B8 0x5C01 10B8	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	BLACK CLAMP CTRL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		CLHWN		RESERVED		CLHWM		RESERVED	CLHLMT	CLHWBS	CLHWC				

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:14	RESERVED		R	0x0
13:12	CLHWN	Horizontal Black clamp - Vertical dimension of a Window (2^N). 0x0: Window is 2 pixels tall (N=1) 0x1: Window is 4 pixels tall (N=2) 0x2: Window is 8 pixels tall (N=3) 0x3: Window is 16 pixels tall (N=4)	RW	0x0
11:10	RESERVED		R	0x0
9:8	CLHWM	Horizontal Black clamp - Horizontal dimension of a Window (2^M). 0x0: Window is 32 pixels wide (M=5) 0x1: Window is 64 pixels wide (M=6) 0x2: Window is 128 pixels wide (M=7) 0x3: Window is 256 pixels wide (M=8)	RW	0x0
7	RESERVED		R	0
6	CLHLMT	Horizontal Black clamp - Pixel value limitation for the Horizontal clamp value calculation. If this bit is set, the maximum pixel value to be used for the clamp value calculation would be limited to 1023. By setting this bit, the pixel value greater than 1023 will be replaced by the last pixel value which was equal to or less than 1023. In case ISIF_CLAMPCFG.CLMD=1 (4-color mode), the pixel value greater than 1023 will be replaced by the last pixel value of the same color which was equal to or less than 1023. 0x0: Limitation disabled 0x1: Limitation enabled	RW	0
5	CLHWBS	Horizontal Black clamp - Base Window select 0x0: The most left window 0x1: The most right window	RW	0
4:0	CLHWC	Horizontal Black clamp - Window count per color Window count = CLHWC+1 Range: 1 - 32	RW	0x00

3.5.5.42 ISIF_CLHWIN1

Table 686. ISIF_CLHWIN1

Address Offset	0x0000 00BC		
Physical Address	0x5505 10BC 0x5C01 10BC	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	BLACK CLAMP CTRL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								CLHSH							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	CLHSH	Horizontal black clamp. Window Start position (H). Range: 0 - 8191	RW	0x0000

3.5.5.43 ISIF_CLHWIN2

Table 687. ISIF_CLHWIN2

Address Offset	0x0000 00C0		
Physical Address	0x5505 10C0 0x5C01 10C0	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	BLACK CLAMP CTRL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								CLHSV							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	CLHSV	Horizontal black clamp. Window Start position (V). Range: 0 - 8191	RW	0x0000

3.5.5.44 ISIF_CLVRV

Table 688. ISIF_CLVRV

Address Offset	0x0000 00C4		
Physical Address	0x5505 10C4 0x5C01 10C4	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	BLACK CLAMP CTRL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				CLVRV											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	CLVRV	Vertical black clamp reset value. (U12) Range: 0 to 4095	RW	0x000

3.5.5.45 ISIF_CLVWIN0

Table 689. ISIF_CLVWIN0

Address Offset	0x0000 00C8		
Physical Address	0x5505 10C8 0x5C01 10C8	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	BLACK CLAMP CTRL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLVCOEF								RESERVED		CLVRVSL		RESERVED		CLVOBH	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	CLVCOEF	Vertical Black clamp - Line average coefficient (k). Set a coefficient that is applied to the line average for clamp value calculation. (1-k) is applied to the clamp value of the previous line. Value in the U8Q8 format, the range is 0 to 255/256.	RW	0x00
7:6	RESERVED		R	0x0
5:4	CLVRVSL	Vertical Black clamp - reset value selection Select the reset value for the clamp value of the previous line 0x0: The base value calculated for Horizontal direction 0x1: Value set via the configuration register 0x2: No update (same as the previous image) 0x3: Reserved	RW	0x0
3	RESERVED		R	0
2:0	CLVOBH	Vertical Black clamp - Optical Black H valid (2^L). 0x0: 2 pixels wide (L=1) 0x1: 4 pixels wide (L=2) 0x2: 8 pixels wide (L=3) 0x3: 16 pixels wide (L=4) 0x4: 32 pixels wide (L=5) 0x5: 64 pixels wide (L=6) 0x6: Reserved 0x7: Reserved	RW	0x0

3.5.5.46 ISIF_CLVWIN1

Table 690. ISIF_CLVWIN1

Address Offset	0x0000 00CC		
Physical Address	0x5505 10CC 0x5C01 10CC	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	BLACK CLAMP CTRL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								CLVSH							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	CLVSH	Vertical black clamp. Window Start position (H). Range: 0 - 8191	RW	0x0000

3.5.5.47 ISIF_CLVWIN2

Table 691. ISIF_CLVWIN2

Address Offset	0x0000 00D0		
Physical Address	0x5505 10D0 0x5C01 10D0	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	BLACK CLAMP CTRL REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								CLVSV							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	CLVSV	Vertical black clamp. Window Start position (V). Range: 0 - 8191	RW	0x0000

3.5.5.48 ISIF_CLVWIN3

Table 692. ISIF_CLVWIN3

Address Offset	0x0000 00D4		
Physical Address	0x5505 10D4 0x5C01 10D4	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	BLACK CLAMP CTRL REGISTER		
Type	RW		

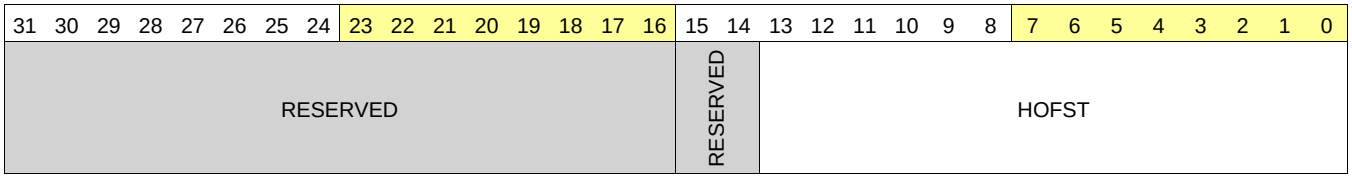
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								CLVOBV							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	CLVOBV	Vertical black clamp. Optical black V valid (V). Range: 0 - 8191	RW	0x0000

3.5.5.49 ISIF_LSCHOFST

Table 693. ISIF_LSCHOFST

Address Offset	0x0000 00D8		
Physical Address	0x5505 10D8 0x5C01 10D8	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	2D Lens Shading Correction Register		
Type	RW		

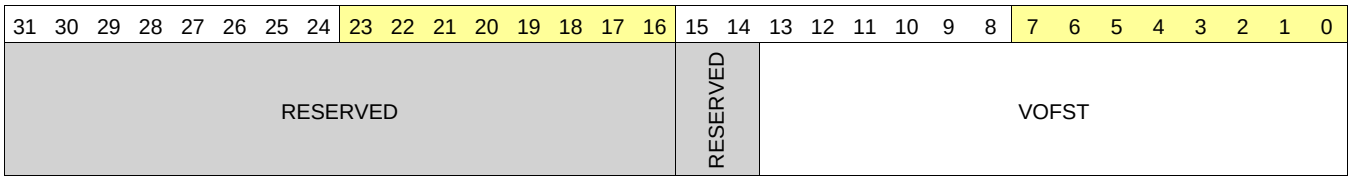


Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:14	RESERVED		R	0x0
13:0	HOFST	H direction Data offset for Lens Shading Correction. Not valid if the Formatter is enabled. Range: 0-16,383	RW	0x0000

3.5.5.50 ISIF_LSCVOFST

Table 694. ISIF_LSCVOFST

Address Offset	0x0000 00DC		
Physical Address	0x5505 10DC 0x5C01 10DC	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	2D Lens Shading Correction Register		
Type	RW		

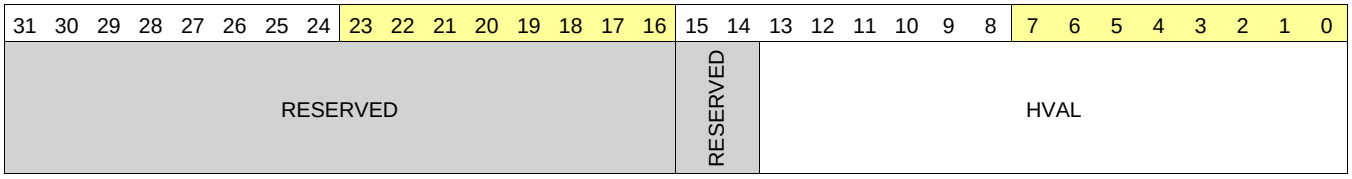


Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:14	RESERVED		R	0x0
13:0	VOFST	V direction Data offset for Lens Shading Correction. Range: 0-16,383	RW	0x0000

3.5.5.51 ISIF_LSCHVAL

Table 695.

Address Offset	0x0000 00E0		
Physical Address	0x5505 10E0 0x5C01 10E0	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	2D Lens Shading Correction Register		
Type	RW		

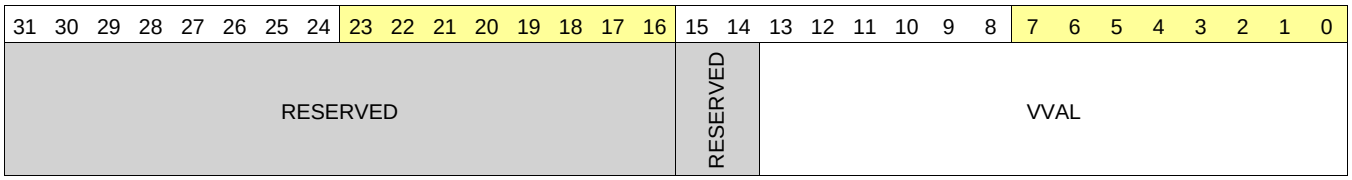


Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:14	RESERVED		R	0x0
13:0	HVAL	Number of valid pixels in H direction. HVAL is for LSC. Number of valid pixels = HVAL+ 1	RW	0x0000

3.5.5.52 ISIF_LSCVVAL

Table 696. ISIF_LSCVVAL

Address Offset	0x0000 00E4		
Physical Address	0x5505 10E4 0x5C01 10E4	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	2D Lens Shading Correction Register		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:14	RESERVED		R	0x0
13:0	VVAL	Number of valid lines in V direction. VVAL is for LSC. Number of valid lines = VVAL+ 1	RW	0x0000

3.5.5.53 ISIF_2DLSCCFG

Table 697. ISIF_2DLSCCFG

Address Offset	0x0000 00E8		
Physical Address	0x5505 10E8 0x5C01 10E8	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	2D Lens Shading Correction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	GAIN_MODE_M			RESERVED	GAIN_MODE_N			BUSY	RESERVED				GAIN_FORMAT		ENABLE

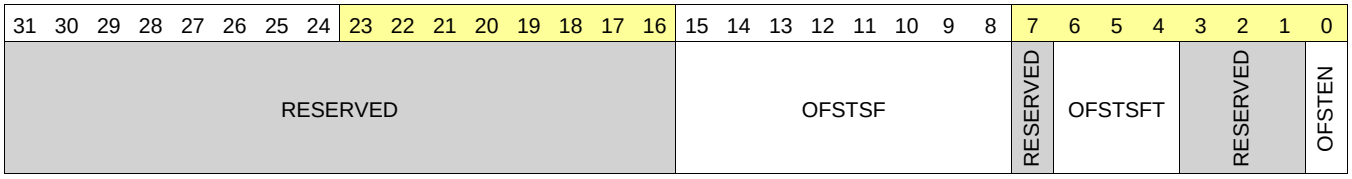
Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	RESERVED		R	0
14:12	GAIN_MODE_M	Define the horizontal dimension of a paxel. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Paxel is 8 pixels tall (M=8) 0x4: Paxel is 16 pixels tall (M=16) 0x5: Paxel is 32 pixels tall (M=32) 0x6: Paxel is 64 pixels tall (M=64) 0x7: Paxel is 128 pixels tall (M=128)	RW	0x6
11	RESERVED		R	0
10:8	GAIN_MODE_N	Define the vertical dimension of a paxel. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Paxel is 8 pixels tall (N=8) 0x4: Paxel is 16 pixels tall (N=16) 0x5: Paxel is 32 pixels tall (N=32) 0x6: Paxel is 64 pixels tall (N=64) 0x7: Paxel is 128 pixels tall (N=128)	RW	0x6
7	BUSY	Busy bit Read 0x1: Busy Read 0x0: Idle	R	0
6:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3:1	GAIN_FORMAT	Sets gain table format 0x0: Coded as 8-bit fraction Range from 0 to 255/256 0x1: Coded as 8-bit fraction + 1.0 of base Range from 1 to 1+255/256 0x2: Coded as 1-bit integer, 7-bit fraction Range from 0 to 1+127/128 0x3: Coded as 1-bit integer, 7-bit fraction + 1.0 Range from 1 to 2+127/128 0x4: Coded as 2-bit integer, 6-bit fraction Range from 0 to 3+63/64 0x5: Coded as 2-bit integer, 6-bit fraction + 1.0 Range from 1 to 4+63/64 0x6: Coded as 3-bit integer, 5-bit fraction Range from 0 to 7+31/32 0x7: Coded as 3-bit integer, 5-bit fraction + 1.0 Range from 1 to 8+31/32	RW	0x0
0	ENABLE	Enables/disables LSC 0x0: Disables the module at the end of the current frame. 0x1: Enables the module.	RW	0

3.5.5.54 ISIF_2DLSCOFST

Table 698. ISIF_2DLSCOFST

Address Offset	0x0000 00EC		
Physical Address	0x5505 10EC 0x5C01 10EC	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	2D Lens Shading Correction Register		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	OFSTSF	Scaling factor for Offsets (U8Q7) Range: 0 to 1+127/128	RW	0x80
7	RESERVED		R	0
6:4	OFSTSFT	Shift up value for Offsets (S8Q0) 0x0: No shift 0x1: 1 bit left shift 0x2: 2 bits left shift 0x3: 3 bits left shift 0x4: 4 bits left shift 0x5: 5 bits left shift 0x6: Reserved 0x7: Reserved	RW	0x0
3:1	RESERVED		R	0x0
0	OFSTEN	Enables/disables Offset control in LSC 0x0: Disable 0x1: Enable	RW	0

3.5.5.55 ISIF_2DLSCINI

Table 699. ISIF_2DLSCINI

Address Offset	0x0000 00F0		
Physical Address	0x5505 10F0 0x5C01 10F0	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	2D Lens Shading Correction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	Y							RESERVED	X						

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	RESERVED		R	0
14:8	Y	Initial Y Y position, in pixels, of the first active pixel in reference to the first active paxel. Must be an even number.	RW	0x00
7	RESERVED		R	0
6:0	X	Initial X X position, in pixels, of the first active pixel in reference to the first active paxel. Must be an even number.	RW	0x00

3.5.5.56 ISIF_2DLSCGRBU

Table 700. ISIF_2DLSCGRBU

Address Offset	0x0000 00F4		
Physical Address	0x5505 10F4 0x5505 10F4	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	2D Lens Shading Correction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BASE31_16															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	BASE31_16	Gain Table address base (Upper 16-bits) Table address in bytes. Table is 32-bit aligned so this register must be a multiple of 4. This bit field sets the address of the gain table in memory.	RW	0x0000

3.5.5.57 ISIF_2DLSCGRBL

Table 701. ISIF_2DLSCGRBL

Address Offset	0x0000 00F8		
Physical Address	0x5505 10F8 0x5C01 10F8	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	2D Lens Shading Correction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BASE15_0															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	BASE15_0	Gain Table address base (Lower 16-bits) Table address in bytes. Table is 32-bit aligned so this register must be a multiple of 4. This bit field sets the address of the gain table in memory.	RW	0x0000

3.5.5.58 ISIF_2DLSCGROF

Table 702. ISIF_2DLSCGROF

Address Offset	0x0000 00FC		
Physical Address	0x5505 10FC 0x5C01 10FC	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	2D Lens Shading Correction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																OFFSET															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	OFFSET	Gain Table offset Defines the length, in bytes, of one row of the table. Table is 32-bit aligned, so this value must be a multiple of 4. Note that the row in memory could be longer than what LSC uses.	RW	0x0000

3.5.5.59 ISIF_2DLSCORBU

Table 703. ISIF_2DLSCORBU

Address Offset	0x0000 0100		
Physical Address	0x5505 1100 0x5C01 1100	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	2D Lens Shading Correction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BASE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	BASE	Offset Table address base (Upper 16-bits) Table address in bytes. Table is 32-bit aligned so this register must be a multiple of 4. This bit field sets the address of the gain table in memory.	RW	0x0000

3.5.5.60 ISIF_2DLSCORBL

Table 704. ISIF_2DLSCORBL

Address Offset	0x0000 0104		
Physical Address	0x5505 1104 0x5C01 1104	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	2D Lens Shading Correction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BASE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	BASE	Offset Table address base (Lower 16-bits) Table address in bytes. Table is 32-bit aligned so this register must be a multiple of 4. This bit field sets the address of the gain table in memory.	RW	0x0000

3.5.5.61 ISIF_2DLSCOROF

Table 705. ISIF_2DLSCOROF

Address Offset	0x0000 0108		
Physical Address	0x5C01 1104 0x5C01 1108	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	2D Lens Shading Correction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																OFFSET															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	OFFSET	Offset Table offset Defines the length, in bytes, of one row of the table. Table is 32-bit aligned, so this value must be a multiple of 4. Note that the row in memory could be longer than what LSC uses.	RW	0x0000

3.5.5.62 ISIF_2DLSCIRQEN

Table 706. ISIF_2DLSCIRQEN

Address Offset	0x0000 010C		
Physical Address	0x5505 110C 0x5C01 110C	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED												SOF	PREFETCH_COMPLETED	PREFETCH_ERROR	DONE

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:4	RESERVED		R	0x000
3	SOF	Interrupt status for LSC SOF Indicates the start of the LSC valid region. LSC configuration registers can be updated after LSC SOF for the next frame. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
2	PREFETCH_COMPLETED	Interrupt enable for Prefetch Complete Indicates current state of the prefetch buffer. Could be used to start sending the data once the buffer is full to minimize the risk of an underflow. This event is triggered when the buffer contains 3 full paxel rows. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
1	PREFETCH_ERROR	Interrupt enable for Prefetch Error The prefetch error indicates when the gain table was read to slowly from SDRAM. When this event is pending the module goes into transparent mode (output=input). Normal operation can be resumed at the start of the next frame after 1) clearing this event 2) disabling the LSC module 3) enabling it 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
0	DONE	Interrupt enable for LSC Done The event is triggered when the internal state of LSC toggles from BUSY to IDLE. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0

3.5.5.63 ISIF_2DLSCIRQST

Table 707. ISIF_2DLSCIRQST

Address Offset	0x0000 0110		
Physical Address	0x5505 1110 0x5C01 1110	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	2D Lens Shading Correction Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED												SOF	PREFETCH_COMPLETED	PREFETCH_ERROR	DONE

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:4	RESERVED		R	0x000
3	SOF	Interrupt status for LSC SOF Indicates the start of the LSC valid region. LSC configuration registers can be updated after LSC SOF for the next frame. 0x0: Event is not pending (r) Bit remains unchanged (w) 0x1: Event is pending (r) Event is cleared (w)	RW	0
2	PREFETCH_COMPLETED	Interrupt status for Prefetch Complete Indicates current state of the prefetch buffer. Could be used to start sending the data once the buffer is full to minimize the risk of an underflow. This event is triggered when the buffer contains 3 full paxel rows. It could be used to minimize buffer underflow risks. 0x0: Event is not pending (r) Bit remains unchanged (w) 0x1: Event is pending (r) Event is cleared (w)	RW	0
1	PREFETCH_ERROR	Interrupt status for Prefetch Error The prefetch error indicates when the gain table was read to slowly from SDRAM. When this event is pending the module goes into transparent mode (output=input). Normal operation can be resumed at the start of the next frame after 1) clearing this event 2) disabling the LSC module 3) enabling it 0x0: Event is not pending (r) Bit remains unchanged (w) 0x1: Event is pending (r) Event is cleared (w)	RW	0

Bits	Field Name	Description	Type	Reset
0	DONE	Interrupt status for LSC Done The event is triggered when the internal state of LSC toggles from BUSY to IDLE. 0x0: Event is not pending (r) Bit remains unchanged (w) 0x1: Event is pending (r) Event is cleared (w)	RW	0

3.5.5.64 ISIF_FMTCFG

Table 708. ISIF_FMTCFG

Address Offset	0x0000 0114		
Physical Address	0x5505 1114 0x5C01 1114	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				FMTAINC				RESERVED		LNUM		RESERVED	LNALT	FMTCBL	FMTEN

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:8	FMTAINC	Address increment Address increment = (FMTAINC + 1) Range (1-16) *This bit is latched by VD.	RW	0x0
7:6	RESERVED		R	0x0
5:4	LNUM	Split/Combine number of lines *This bit is latched by VD. 0x0: 1 output line 0x1: 1 input line - 2 output lines (FMTCBL=0) 2 input lines - 1 output line (FMTCBL=1) 0x2: 1 input line - 3 output lines (FMTCBL=0) 3 input lines - 1 output line (FMTCBL=1) 0x3: 1 input line - 4 output lines (FMTCBL=0) 4 input lines - 1 output line (FMTCBL=1)	RW	0x0
3	RESERVED		R	0
2	LNALT	Line alternating *This bit is latched by VD. 0x0: Normal mode 0x1: Line alternative mode	RW	0
1	FMTCBL	Combine Input lines *This bit is latched by VD. 0x0: Split 1 input line into multiple output lines 0x1: Combine multiple input lines into 1 output line	RW	0
0	FMTEN	CCD Formatter enable *This bit is latched by VD. 0x0: Disable 0x1: Enable	RW	0

3.5.5.65 ISIF_FMTPLEN

Table 709. ISIF_FMTPLEN

Address Offset	0x0000 0118		
Physical Address	0x5505 1118 0x5C01 1118	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	FMTPLEN3			RESERVED		FMTPLEN2		FMTPLEN1				FMTPLEN0			

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:12	FMTPLEN3	Number of program entries for SET3 Number of entries = (FMTPLEN3 + 1) Range: 1-8 Valid only if FMTCBL is set *This bit is latched by VD.	RW	0x0
11	RESERVED		R	0
10:8	FMTPLEN2	Number of program entries for SET2 Number of entries = (FMTPLEN2 + 1) Range: 1-8 Valid only if FMTCBL is set *This bit is latched by VD.	RW	0x0
7:4	FMTPLEN1	Number of program entries for SET1 Number of entries = (FMTPLEN1 + 1) Range: 1-16 (FMTCBL = 0) 1-8 (FMTCBL = 1) Setting a value greater than 7 to FMTPLEN1 is not allowed if FMTCBL is set *This bit is latched by VD.	RW	0x0
3:0	FMTPLEN0	Number of program entries for SET0 Number of entries = (PLEN0 + 1) Range: 1-16 (FMTCBL = 0) 1-8 (FMTCBL = 1) Setting a value greater than 7 to FMTPLEN1 is not allowed if FMTCBL is set *This bit is latched by VD.	RW	0x0

3.5.5.66 ISIF_FMTSPH

Table 710. ISIF_FMTSPH

Address Offset	0x0000 011C		
Physical Address	0x5505 111C 0x5C01 111C	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								FMTSPH							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	FMTSPH	The first pixel in a line fed into the formatter	RW	0x0000

3.5.5.67 ISIF_FMTLNH

Table 711. ISIF_FMTLNH

Address Offset	0x0000 0120		
Physical Address	0x5505 1120 0x5C01 1120	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								FMTLNH							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	FMTLNH	Number of pixels in a line fed to the formatter. Number of pixels = FMTLNH + 1	RW	0x0000

3.5.5.68 ISIF_FMTLSV

Table 712. ISIF_FMTLSV

Address Offset	0x0000 0124		
Physical Address	0x5505 1124 0x5C01 1124	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FMTSLV															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	FMTSLV	Start line vertical	RW	0x0000

3.5.5.69 ISIF_FMTLNV

Table 713. ISIF_FMTLNV

Address Offset	0x0000 0128		
Physical Address	0x5505 1128 0x5C01 1128	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FMTLNV															

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:0	FMTLNV	Number of lines in vertical Number of lines = FMTLNV + 1	RW	0x0000

3.5.5.70 ISIF_FMTRLEN

Table 714. ISIF_FMTRLEN

Address Offset	0x0000 012C		
Physical Address	0x5505 112C 0x5C01 112C	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								FMTRLEN							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	FMTRLEN	Number of pixels in an output line Maximum value = 4480	RW	0x0000

3.5.5.71 ISIF_FMTHCNT

Table 715. ISIF_FMTHCNT

Address Offset	0x0000 0130		
Physical Address	0x5505 1130 0x5C01 1130	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								FMTHCNT							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	FMTHCNT	HD interval for output lines Set all 0 to this register if combining multiple lines into a single line.	RW	0x0000

3.5.5.72 ISIF_FMTAPTR0

Table 716. ISIF_FMTAPTR0

Address Offset	0x0000 0134		
Physical Address	0x5505 1134 0x5C01 1134	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE			INIT												

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared 0x0: 1st line 0x1: 2nd line 0x2: 3rd line 0x3: 4th line	RW	0x0
12:0	INIT	Initial address value for address pointer 0 This address can not exceed FMTRLEN - 1	RW	0x0000

3.5.5.73 ISIF_FMTAPTR1

Table 717. ISIF_FMTAPTR1

Address Offset	0x0000 0138		
Physical Address	0x5505 1138 0x5C01 1138	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE			INIT												

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared 0x0: 1st line 0x1: 2nd line 0x2: 3rd line 0x3: 4th line	RW	0x0
12:0	INIT	Initial address value for address pointer 1 This address can not exceed FMTRLEN - 1	RW	0x0000

3.5.5.74 ISIF_FMTAPTR2

Table 718. ISIF_FMTAPTR2

Address Offset	0x0000 013C		
Physical Address	0x5505 113C 0x5C01 113C	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE								INIT							

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared 0x0: 1st line 0x1: 2nd line 0x2: 3rd line 0x3: 4th line	RW	0x0
12:0	INIT	Initial address value for address pointer 2 This address can not exceed FMTRLEN - 1	RW	0x0000

3.5.5.75 ISIF_FMTAPTR3

Table 719. ISIF_FMTAPTR3

Address Offset	0x0000 0140		
Physical Address	0x5505 1140 0x5C01 1140	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE								INIT							

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared 0x0: 1st line 0x1: 2nd line 0x2: 3rd line 0x3: 4th line	RW	0x0
12:0	INIT	Initial address value for address pointer 3 This address can not exceed FMTRLEN - 1	RW	0x0000

3.5.5.76 ISIF_FMTAPTR4

Table 720. ISIF_FMTAPTR4

Address Offset	0x0000 0144		
Physical Address	0x5505 1144 0x5C01 1144	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE								INIT							

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared 0x0: 1st line 0x1: 2nd line 0x2: 3rd line 0x3: 4th line	RW	0x0
12:0	INIT	Initial address value for address pointer 4 This address can not exceed FMTRLEN - 1	RW	0x0000

3.5.5.77 ISIF_FMTAPTR5

Table 721. ISIF_FMTAPTR5

Address Offset	0x0000 0148		
Physical Address	0x5505 1148 0x5C01 1148	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE								INIT							

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared 0x0: 1st line 0x1: 2nd line 0x2: 3rd line 0x3: 4th line	RW	0x0
12:0	INIT	Initial address value for address pointer 5 This address can not exceed FMTRLEN - 1	RW	0x0000

3.5.5.78 ISIF_FMTAPTR6

Table 722. ISIF_FMTAPTR6

Address Offset	0x0000 014C		
Physical Address	0x5505 114C 0x5C01 114C	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE								INIT							

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared 0x0: 1st line 0x1: 2nd line 0x2: 3rd line 0x3: 4th line	RW	0x0
12:0	INIT	Initial address value for address pointer 6 This address can not exceed FMTRLEN - 1	RW	0x0000

3.5.5.79 ISIF_FMTAPTR7

Table 723. ISIF_FMTAPTR7

Address Offset	0x0000 0150		
Physical Address	0x5505 1150 0x5C01 1150	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE								INIT							

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared 0x0: 1st line 0x1: 2nd line 0x2: 3rd line 0x3: 4th line	RW	0x0
12:0	INIT	Initial address value for address pointer 7 This address can not exceed FMTRLEN - 1	RW	0x0000

3.5.5.80 ISIF_FMTAPTR8

Table 724. ISIF_FMTAPTR8

Address Offset	0x0000 0154		
Physical Address	0x5505 1154 0x5C01 1154	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE								INIT							

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared 0x0: 1st line 0x1: 2nd line 0x2: 3rd line 0x3: 4th line	RW	0x0
12:0	INIT	Initial address value for address pointer 8 This address can not exceed FMTRLEN - 1	RW	0x0000

3.5.5.81 ISIF_FMTAPTR9

Table 725. ISIF_FMTAPTR9

Address Offset	0x0000 0158		
Physical Address	0x5505 1158 0x5C01 1158	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE								INIT							

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared 0x0: 1st line 0x1: 2nd line 0x2: 3rd line 0x3: 4th line	RW	0x0
12:0	INIT	Initial address value for address pointer 9 This address can not exceed FMTRLEN - 1	RW	0x0000

3.5.5.82 ISIF_FMTAPTR10

Table 726. ISIF_FMTAPTR10

Address Offset	0x0000 015C		
Physical Address	0x5505 115C 0x5C01 115C	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE			INIT												

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared 0x0: 1st line 0x1: 2nd line 0x2: 3rd line 0x3: 4th line	RW	0x0
12:0	INIT	Initial address value for address pointer 10 This address can not exceed FMTRLEN - 1	RW	0x0000

3.5.5.83 ISIF_FMTAPTR11

Table 727. ISIF_FMTAPTR11

Address Offset	0x0000 0160		
Physical Address	0x5505 1160 0x5C01 1160	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE			INIT												

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared 0x0: 1st line 0x1: 2nd line 0x2: 3rd line 0x3: 4th line	RW	0x0
12:0	INIT	Initial address value for address pointer 11 This address can not exceed FMTRLEN - 1	RW	0x0000

3.5.5.84 ISIF_FMTAPTR12

Table 728. ISIF_FMTAPTR12

Address Offset	0x0000 0164		
Physical Address	0x5505 1164 0x5C01 1164	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE								INIT							

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared 0x0: 1st line 0x1: 2nd line 0x2: 3rd line 0x3: 4th line	RW	0x0
12:0	INIT	Initial address value for address pointer 12 This address can not exceed FMTRLEN - 1	RW	0x0000

3.5.5.85 ISIF_FMTAPTR13

Table 729. ISIF_FMTAPTR13

Address Offset	0x0000 0168		
Physical Address	0x5505 1168 0x5C01 1168	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE								INIT							

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared 0x0: 1st line 0x1: 2nd line 0x2: 3rd line 0x3: 4th line	RW	0x0
12:0	INIT	Initial address value for address pointer 13 This address can not exceed FMTRLEN - 1	RW	0x0000

3.5.5.86 ISIF_FMTAPTR14

Table 730. ISIF_FMTAPTR14

Address Offset	0x0000 016C		
Physical Address	0x5505 116C 0x5C01 116C	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE			INIT												

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared 0x0: 1st line 0x1: 2nd line 0x2: 3rd line 0x3: 4th line	RW	0x0
12:0	INIT	Initial address value for address pointer 14 This address can not exceed FMTRLEN - 1	RW	0x0000

3.5.5.87 ISIF_FMTAPTR15

Table 731. ISIF_FMTAPTR15

Address Offset	0x0000 0170		
Physical Address	0x5505 1170 0x5C01 1170	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE			INIT												

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared 0x0: 1st line 0x1: 2nd line 0x2: 3rd line 0x3: 4th line	RW	0x0
12:0	INIT	Initial address value for address pointer 15 This address can not exceed FMTRLEN - 1	RW	0x0000

3.5.5.88 ISIF_FMTPGMVF0

Table 732. ISIF_FMTPGMVF0

Address Offset	0x0000 0174		
Physical Address	0x5505 1174 0x5C01 1174	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM15EN	PGM14EN	PGM13EN	PGM12EN	PGM11EN	PGM10EN	PGM09EN	PGM08EN	PGM07EN	PGM06EN	PGM05EN	PGM04EN	PGM03EN	PGM02EN	PGM01EN	PGM00EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	PGM15EN	Program 15 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
14	PGM14EN	Program 14 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
13	PGM13EN	Program 13 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
12	PGM12EN	Program 12 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
11	PGM11EN	Program 11 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
10	PGM10EN	Program 10 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
9	PGM09EN	Program 9 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
8	PGM08EN	Program 8 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
7	PGM07EN	Program 7 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
6	PGM06EN	Program 6 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
5	PGM05EN	Program 5 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0

Bits	Field Name	Description	Type	Reset
4	PGM04EN	Program 4 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
3	PGM03EN	Program 3 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
2	PGM02EN	Program 2 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
1	PGM01EN	Program 1 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
0	PGM00EN	Program 0 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0

3.5.5.89 ISIF_FMTPGMVF1

Table 733. ISIF_FMTPGMVF1

Address Offset	0x0000 0178		
Physical Address	0x5505 1178 0x5C01 1178	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM31EN	PGM30EN	PGM29EN	PGM28EN	PGM27EN	PGM26EN	PGM25EN	PGM24EN	PGM23EN	PGM22EN	PGM21EN	PGM20EN	PGM19EN	PGM18EN	PGM17EN	PGM16EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	PGM31EN	Program 31 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
14	PGM30EN	Program 30 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
13	PGM29EN	Program 29 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
12	PGM28EN	Program 28 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
11	PGM27EN	Program 27 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
10	PGM26EN	Program 26 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
9	PGM25EN	Program 25 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
8	PGM24EN	Program 24 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
7	PGM23EN	Program 23 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
6	PGM22EN	Program 22 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
5	PGM21EN	Program 21 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0

Bits	Field Name	Description	Type	Reset
4	PGM20EN	Program 20 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
3	PGM19EN	Program 19 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
2	PGM18EN	Program 18 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
1	PGM17EN	Program 17 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
0	PGM16EN	Program 16 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0

3.5.5.90 ISIF_FMTPGMAPU0

Table 734. ISIF_FMTPGMAPU0

Address Offset	0x0000 017C		
Physical Address	0x5505 117C 0x5C01 117C	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM15UPDT	PGM14UPDT	PGM13UPDT	PGM12UPDT	PGM11UPDT	PGM10UPDT	PGM9UPDT	PGM8UPDT	PGM7UPDT	PGM6UPDT	PGM5UPDT	PGM4UPDT	PGM3UPDT	PGM2UPDT	PGM1UPDT	PGM0UPDT

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	PGM15UPDT	Program 15 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
14	PGM14UPDT	Program 14 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
13	PGM13UPDT	Program 13 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
12	PGM12UPDT	Program 12 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
11	PGM11UPDT	Program 11 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
10	PGM10UPDT	Program 10 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
9	PGM9UPDT	Program 9 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
8	PGM8UPDT	Program 8 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
7	PGM7UPDT	Program 7 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
6	PGM6UPDT	Program 6 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
5	PGM5UPDT	Program 5 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0

Bits	Field Name	Description	Type	Reset
4	PGM4UPDT	Program 4 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
3	PGM3UPDT	Program 3 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
2	PGM2UPDT	Program 2 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
1	PGM1UPDT	Program 1 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
0	PGM0UPDT	Program 0 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0

3.5.5.91 ISIF_FMTPGMAPU1

Table 735. ISIF_FMTPGMAPU1

Address Offset	0x0000 0180		
Physical Address	0x5505 1180 0x5C01 1180	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM31UPDT	PGM30UPDT	PGM29UPDT	PGM28UPDT	PGM27UPDT	PGM26UPDT	PGM25UPDT	PGM24UPDT	PGM23UPDT	PGM22UPDT	PGM21UPDT	PGM20UPDT	PGM19UPDT	PGM18UPDT	PGM17UPDT	PGM16UPDT

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	PGM31UPDT	Program 31 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
14	PGM30UPDT	Program 30 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
13	PGM29UPDT	Program 29 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
12	PGM28UPDT	Program 28 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
11	PGM27UPDT	Program 27 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
10	PGM26UPDT	Program 26 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
9	PGM25UPDT	Program 25 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
8	PGM24UPDT	Program 24 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
7	PGM23UPDT	Program 23 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
6	PGM22UPDT	Program 22 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
5	PGM21UPDT	Program 21 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0

Bits	Field Name	Description	Type	Reset
4	PGM20UPDT	Program 20 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
3	PGM19UPDT	Program 19 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
2	PGM18UPDT	Program 18 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
1	PGM17UPDT	Program 17 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
0	PGM16UPDT	Program 16 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0

3.5.5.92 ISIF_FMTPGMAPS0

Table 736. ISIF_FMTPGMAPS0

Address Offset	0x0000 0184		
Physical Address	0x5505 1184 0x5C01 1184	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM3APTR				PGM2APTR				PGM1APTR				PGM0APTR			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	PGM3APTR	Program 3 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
11:8	PGM2APTR	Program 2 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
7:4	PGM1APTR	Program 1 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
3:0	PGM0APTR	Program 0 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0

3.5.5.93 ISIF_FMTPGMAPS1

Table 737. ISIF_FMTPGMAPS1

Address Offset	0x0000 0188		
Physical Address	0x5505 1188 0x5C01 1188	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM7APTR				PGM6APTR				PGM5APTR				PGM4APTR			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	PGM7APTR	Program 7 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
11:8	PGM6APTR	Program 6 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
7:4	PGM5APTR	Program 5 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
3:0	PGM4APTR	Program 0 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0

3.5.5.94 ISIF_FMTPGMAPS2

Table 738. ISIF_FMTPGMAPS2

Address Offset	0x0000 018C		
Physical Address	0x5505 118C 0x5C01 118C	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM11APTR				PGM10APTR				PGM9APTR				PGM8APTR			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	PGM11APTR	Program 11 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
11:8	PGM10APTR	Program 10 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
7:4	PGM9APTR	Program 9 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
3:0	PGM8APTR	Program 8 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0

3.5.5.95 ISIF_FMTPGMAPS3

Table 739. ISIF_FMTPGMAPS3

Address Offset	0x0000 0190		
Physical Address	0x5505 1190 0x5C01 1190	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM15APTR				PGM14APTR				PGM13APTR				PGM12APTR			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	PGM15APTR	Program 15 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
11:8	PGM14APTR	Program 14 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
7:4	PGM13APTR	Program 13 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
3:0	PGM12APTR	Program 12 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0

3.5.5.96 ISIF_FMTPGMAPS4

Table 740. ISIF_FMTPGMAPS4

Address Offset	0x0000 0194		
Physical Address	0x5505 1194 0x5C01 1194	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM19APTR				PGM18APTR				PGM17APTR				PGM16APTR			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	PGM19APTR	Program 19 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
11:8	PGM18APTR	Program 18 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
7:4	PGM17APTR	Program 17 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
3:0	PGM16APTR	Program 16 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0

3.5.5.97 ISIF_FMTPGMAPS5

Table 741. ISIF_FMTPGMAPS5

Address Offset	0x0000 0198		
Physical Address	0x5505 1198 0x5C01 1198	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM23APTR				PGM22APTR				PGM21APTR				PGM20APTR			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	PGM23APTR	Program 23 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
11:8	PGM22APTR	Program 22 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
7:4	PGM21APTR	Program 21 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
3:0	PGM20APTR	Program 20 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0

3.5.5.98 ISIF_FMTPGMAPS6

Table 742. ISIF_FMTPGMAPS6

Address Offset	0x0000 019C		
Physical Address	0x5505 119C 0x5C01 119C	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM27APTR				PGM26APTR				PGM25APTR				PGM24APTR			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	PGM27APTR	Program 27 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
11:8	PGM26APTR	Program 26 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
7:4	PGM25APTR	Program 25 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
3:0	PGM24APTR	Program 24 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0

3.5.5.99 ISIF_FMTPGMAPS7

Table 743. ISIF_FMTPGMAPS7

Address Offset	0x0000 01A0		
Physical Address	0x5505 11A0 0x5C01 11A0	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Input Data Formatter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM31APTR				PGM30APTR				PGM29APTR				PGM28APTR			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	PGM31APTR	Program 31 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
11:8	PGM30APTR	Program 30 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
7:4	PGM29APTR	Program 29 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
3:0	PGM28APTR	Program 28 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0

3.5.5.100 ISIF_CSCCTL

Table 744. ISIF_CSCCTL

Address Offset	0x0000 01A4		
Physical Address	0x5505 11A4 0x5C01 11A4	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Color Space Converter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															CSCEN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	CSCEN	Controls ON/OFF of Color Space converter. 0x0: Disable 0x1: Enable	RW	0

3.5.5.101 ISIF_CSCM0

Table 745. ISIF_CSCM0

Address Offset	0x0000 01A8		
Physical Address	0x5505 11A8 0x5C01 11A8	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Color Space Converter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CSCM01								CSCM00							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	CSCM01	Color Space convert coefficient value M01: This value is signed 8-bit with the 5-bits decimal.	RW	0x00
7:0	CSCM00	Color Space convert coefficient value M00: This value is signed 8-bit with the 5-bits decimal.	RW	0x00

3.5.5.102 ISIF_CSCM1

Table 746. ISIF_CSCM1

Address Offset	0x0000 01AC		
Physical Address	0x5505 11AC 0x5C01 11AC	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Color Space Converter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CSCM03								CSCM02							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	CSCM03	Color Space convert coefficient value M03: This value is signed 8-bit with the 5-bits decimal.	RW	0x00
7:0	CSCM02	Color Space convert coefficient value M02: This value is signed 8-bit with the 5-bits decimal.	RW	0x00

3.5.5.103 ISIF_CSCM2

Table 747. ISIF_CSCM2

Address Offset	0x0000 01B0		
Physical Address	0x5505 11B0 0x5C01 11B0	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Color Space Converter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CSCM11								CSCM10							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	CSCM11	Color Space convert coefficient value M11: This value is signed 8-bit with the 5-bits decimal.	RW	0x00
7:0	CSCM10	Color Space convert coefficient value M10: This value is signed 8-bit with the 5-bits decimal.	RW	0x00

3.5.5.104 ISIF_CSCM3

Table 748. ISIF_CSCM3

Address Offset	0x0000 01B4		
Physical Address	0x5505 11B4 0x5C01 11B4	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Color Space Converter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CSCM13								CSCM12							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	CSCM13	Color Space convert coefficient value M13: This value is signed 8-bit with the 5-bits decimal.	RW	0x00
7:0	CSCM12	Color Space convert coefficient value M12: This value is signed 8-bit with the 5-bits decimal.	RW	0x00

3.5.5.105 ISIF_CSCM4

Table 749. ISIF_CSCM4

Address Offset	0x0000 01B8		
Physical Address	0x5505 11B8 0x5C01 11B8	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Color Space Converter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CSCM21								CSCM20							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	CSCM21	Color Space convert coefficient value M21: This value is signed 8-bit with the 5-bits decimal.	RW	0x00
7:0	CSCM20	Color Space convert coefficient value M20: This value is signed 8-bit with the 5-bits decimal.	RW	0x00

3.5.5.106 ISIF_CSCM5

Table 750. ISIF_CSCM5

Address Offset	0x0000 01BC		
Physical Address	0x5505 11BC 0x5C01 11BC	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Color Space Converter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CSCM23								CSCM22							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	CSCM23	Color Space convert coefficient value M23: This value is signed 8-bit with the 5-bits decimal.	RW	0x00
7:0	CSCM22	Color Space convert coefficient value M22: This value is signed 8-bit with the 5-bits decimal.	RW	0x00

3.5.5.107 ISIF_CSCM6

Table 751. ISIF_CSCM6

Address Offset	0x0000 01C0		
Physical Address	0x5505 11C0 0x5C01 11C0	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Color Space Converter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CSCM31								CSCM30							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	CSCM31	Color Space convert coefficient value M31: This value is signed 8-bit with the 5-bits decimal.	RW	0x00
7:0	CSCM30	Color Space convert coefficient value M30: This value is signed 8-bit with the 5-bits decimal.	RW	0x00

3.5.5.108 ISIF_CSCM7

Table 752. ISIF_CSCM7

Address Offset	0x0000 01C4		
Physical Address	0x5505 11C4 0x5C01 11C4	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description	Color Space Converter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CSCM33								CSCM32							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	CSCM33	Color Space convert coefficient value M33: This value is signed 8-bit with the 5-bits decimal.	RW	0x00
7:0	CSCM32	Color Space convert coefficient value M32: This value is signed 8-bit with the 5-bits decimal.	RW	0x00

3.5.5.109 ISIF_CLKCTL

Table 753. ISIF_CLKCTL

Address Offset	0x0000 01F8		
Physical Address	0x5505 11F8 0x5C01 11F8	**Instance**	ISS_ISIF_CORTEX-M3 ISS_ISIF_L3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED														CLKEN1	CLKEN2

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1	CLKEN1	Forces isif_clken1 to be active. (Test mode) 0x0: normal mode 0x1: force isif_clken1 to be active	RW	0
0	CLKEN2	Forces isif_clken2 to be active. (Test mode) 0x0: normal mode 0x1: force isif_clken2 to be active	RW	0

3.5.6 ISS IPIPEIF Registers

Table 754 summarizes the ISS IPIPEIF registers.

Table 754. ISS IPIPEIF Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ISS_IPIPEIF Base Address Cortex-M3 Private Access	ISS_IPIPEIF Base Address L3 Interconnect
IPIPEIF_ENABLE	RW	32	0x0000 0000	0x5505 1200	0x5C01 1200
IPIPEIF_CFG1	RW	32	0x0000 0004	0x5505 1204	0x5C01 1204
IPIPEIF_PPLN	RW	32	0x0000 0008	0x5505 1208	0x5C01 1208
IPIPEIF_LPFR	RW	32	0x0000 000C	0x5505 120C	0x5C01 120C
IPIPEIF_HNUM	RW	32	0x0000 0010	0x5505 1210	0x5C01 1210
IPIPEIF_VNUM	RW	32	0x0000 0014	0x5505 1214	0x5C01 1214
IPIPEIF_ADDRU	RW	32	0x0000 0018	0x5505 1218	0x5C01 1218
IPIPEIF_ADDRL	RW	32	0x0000 001C	0x5505 121C	0x5C01 121C
IPIPEIF_ADOFS	RW	32	0x0000 0020	0x5505 1220	0x5C01 1220
IPIPEIF_RSZ	RW	32	0x0000 0024	0x5505 1224	0x5C01 1224
IPIPEIF_GAIN	RW	32	0x0000 0028	0x5505 1228	0x5C01 1228
IPIPEIF_DPCM	RW	32	0x0000 002C	0x5505 122C	0x5C01 122C
IPIPEIF_CFG2	RW	32	0x0000 0030	0x5505 1230	0x5C01 1230
IPIPEIF_INIRSZ	RW	32	0x0000 0034	0x5505 1234	0x5C01 1234
IPIPEIF_OCLIP	RW	32	0x0000 0038	0x5505 1238	0x5C01 1238
IPIPEIF_DTUDF	RW	32	0x0000 003C	0x5505 123C	0x5C01 123C
IPIPEIF_CLKDIV	RW	32	0x0000 0040	0x5505 1240	0x5C01 1240
IPIPEIF_DPC1	RW	32	0x0000 0044	0x5505 1244	0x5C01 1244
IPIPEIF_DPC2	RW	32	0x0000 0048	0x5505 1248	0x5C01 1248
IPIPEIF_RSZ3A	RW	32	0x0000 0054	0x5505 1254	0x5C01 1254
IPIPEIF_INIRSZ3A	RW	32	0x0000 0058	0x5505 1258	0x5C01 1258

3.5.6.1 IPIPEIF_ENABLE

Table 755. IPIPEIF_ENABLE

Address Offset	0x0000 0000		
Physical Address	0x5505 1200 0x5C01 1200	**Instance**	ISS_IPIPEIF_CORTEX-M3 ISS_IPIPEIF_L3
Description	IPIPEIF Enable.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														SYNCOFF	ENABLE

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	SYNCOFF	VD output mask This register masks the VD output to the IPIPE module. This can be useful when one wants to read data from SDRAM which are stored in a double buffer. If the VD is not masked each time we start the module an new VD will be generated to the IPIPEIF module. Let's consider two buffers A and B of N lines each. *This bit field is latched by VD. 0x0: VD output mask is disabled. 0x1: VD output mask is enabled.	RW	0
0	ENABLE	IPIPE I/F Enable This register is used to start the operation of SDRAM buffer memory read and generates SYNC signals. This register is available when INPSRC1 or INPSCR2 = 1, 2 or 3. 0x0: disable 0x1: enable	RW	0

3.5.6.2 IPIPEIF_CFG1

Table 756. IPIPEIF_CFG1

Address Offset	0x0000 0004		
Physical Address	0x5505 1204 0x5C01 1204	**Instance**	ISS_IPIPEIF_CORTEX-M3 ISS_IPIPEIF_L3
Description	IPIPEIF Configuration #1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																INPSRC1		DATASFT			CLKSEL	UNPACK		AVGFILT	RESERVED			INPSRC2		DECIM	ONESHOT

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:14	INPSRC1	Selects the source for the mux (VPORT / ISIF / SDRAM) as well as the data format type. 0x0: VPORT_RAW 0x1: SDRAM_RAW 0x2: ISIF_DARKFM Input ports to DFS submodule are ISIF and SDRAM. 0x3: SDRAM_YUV	RW	0x0
13:11	DATASFT	SDRAM read data shift This register is available when INPSRC1 or INPSRC2 = 1 or 2, that is, when data are read from SDRAM. 0x0: Output data[11:0] = (read data[11:0]) 0x0FFF 0x1: Output data[11:0] = (read data[11:0] 1) 0x0FFF 0x2: Output data[11:0] = (read data[11:0] 2) 0x0FFF 0x3: Output data[11:0] = (read data[11:0] 3) 0x0FFF 0x4: Output data[11:0] = (read data[11:0] 4) 0x0FFF 0x5: Output data[11:0] = (read data[15:4] 4) 0x0FFF 0x6: Output data[11:0] = (read data[15:4] 4) 0x0FFF 0x7: Output data[11:0] = (read data[15:4] 4) 0x0FFF	RW	0x0
10	CLKSEL	IPIPEIF IPIPE module pixel clock selection. This register must be set to 1 when INPSRC1 or INPSRC2 = 1 or 3, that is, data are solely read from SDRAM (VPORT inactive). 0x0: Selects the pixel clock from the VPORT. 0x1: Selects the pixel clock from the fractional clock divider. The fractional clock divider value is setup with the IPIPEIF_CLKDIV register.	RW	0
9:8	UNPACK	8-Bit, 12-bit Packed Mode When sensor raw data are stored in 8-bit packed mode or 12-bit packed mode, this register should code 1 or 3. This register is effective when INPSRC = 1 or 2. 0x0: 16 bits / pixel 0x1: 8 bits / pixel 0x2: 8 bits / pixel + inverse A law (8 bits to 10 bits) 0x3: 12 bits / pixel	RW	0x0

Bits	Field Name	Description	Type	Reset
7	AVGFILT	Averaging Filter It applies (1,2,1) filter for the RGB/YCbCr data. *This bit field is latched by VD. 0x0: disable 0x1: enable	RW	0
6:4	RESERVED		R	0x0
3:2	INPSRC2	Selects the source for the mux (ISIF / SDRAM) as well as the data format type. 0x0: ISIF 0x1: SDRAM_RAW 0x2: ISIF_DARKFM Input ports to DFS submodule are VPORT and SDRAM. 0x3: SDRAM_YUV	RW	0x0
1	DECIM	Pixel Decimation The decimation rate defined by RSZ register. *This bit field is latched by VD. 0x0: No decimation 0x1: Decimation	RW	0
0	ONESHOT	One Shot Mode This register is available when INPSRC = 1 or 3. 0x0: Continuous mode 0x1: One shot mode	RW	0

3.5.6.3 IPIPEIF_PPLN

Table 757. IPIPEIF_PPLN

Address Offset	0x0000 0008		
Physical Address	0x5505 1208 0x5C01 1208	**Instance**	ISS_IPIPEIF_CORTEX-M3 ISS_IPIPEIF_L3
Description	IPIPEIF Interval of HD / Start pixel in HD		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								PPLN							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	PPLN	Case-1: Interval of Horizontal Sync (HD) Specifies the interval of horizontal sync. This register is available when INPSRC = 1 or 3. Case-2: Start Pixel in Horizontal Sync (HD) Specifies the start pixel in horizontal sync. This register is available when INPSRC = 2 *This bit field is latched by VD.	RW	0x0000

3.5.6.4 IPIPEIF_LPFR

Table 758. IPIPEIF_LPFR

Address Offset	0x0000 000C		
Physical Address	0x5505 120C 0x5C01 120C	**Instance**	ISS_IPIPEIF_CORTEX-M3 ISS_IPIPEIF_L3
Description	IPIPEIF Interval of VD / Start line in VD		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								LPFR							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	LPFR	Case-1: Interval of Vertical Sync (VD) Specifies the interval of vertical sync. This register is available when INPSRC = 1 or 3. Case-2: Start Pixel in Vertical Sync (VD) Specifies the start line in vertical sync. This register is available when INPSRC = 2 *This bit field is latched by VD.	RW	0x0000

3.5.6.5 IPIPEIF_HNUM

Table 759. IPIPEIF_HNUM

Address Offset	0x0000 0010		
Physical Address	0x5505 1210 0x5C01 1210	**Instance**	ISS_IPIPEIF_CORTEX-M3 ISS_IPIPEIF_L3
Description	IPIPEIF Number of valid pixels per line		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED								HNUM							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	HNUM	The Number of Valid Pixels in a Line Specifies the number of valid pixels in a horizontal line. This register is available when INPSRC = 1, 2, or 3 *This bit field is latched by VD.	RW	0x0000

3.5.6.6 IPIPEIF_VNUM

Table 760. IPIPEIF_VNUM

Address Offset	0x0000 0014		
Physical Address	0x5505 1214 0x5C01 1214	**Instance**	ISS_IPIPEIF_CORTEX-M3 ISS_IPIPEIF_L3
Description	IPIPEIF Number of valid lines per frame		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								VNUM							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	VNUM	The Number of Valid Line in a Vertical Specifies the number of valid line in a vertical. This register is available when INPSRC = 1, 2, or 3 *This bit field is latched by VD.	RW	0x0000

3.5.6.7 IPIPEIF_ADDRU

Table 761. IPIPEIF_ADDRU

Address Offset	0x0000 0018		
Physical Address	0x5505 1218 0x5C01 1218	**Instance**	ISS_IPIPEIF_CORTEX-M3 ISS_IPIPEIF_L3
Description	IPIPEIF Memory Address (Upper)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					ADDRU										

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x000000
10:0	ADDRU	Memory Address - Upper Memory address upper 11-bits are specified in units of 32-bytes This register is available when INPSRC = 1, 2, or 3. *This bit field is latched by VD.	RW	0x000

3.5.6.8 IPIPEIF_ADDRL

Table 762. IPIPEIF_ADDRL

Address Offset	0x0000 001C		
Physical Address	0x5505 121C 0x5C01 121C	**Instance**	ISS_IPIPEIF_CORTEX-M3 ISS_IPIPEIF_L3
Description	IPIPEIF Memory Address (Lower)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ADDRL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	ADDRL	Memory Address - Lower Memory address lower 16-bits are specified in units of 32-bytes. This register is available when INPSRC = 1, 2, or 3. *This bit field is latched by VD.	RW	0x0000

3.5.6.9 IPIPEIF_ADOFS

Table 763. IPIPEIF_ADOFS

Address Offset	0x0000 0020		
Physical Address	0x5505 1220 0x5C01 1220	**Instance**	ISS_IPIPEIF_CORTEX-M3 ISS_IPIPEIF_L3
Description	IPIPEIF Address offset		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				ADOFS											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:0	ADOFS	Specifies the SDRAM stride for each line in units of 32-bytes. This register is available when reading data from SDRAM: INPSRC1 or INPSRC2 = 1, 2, or 3. Assuming that the first line is at position ADDR, the second line is at address ADDR+ ADOFS, and so on. *This bit field is latched by VD.	RW	0x000

3.5.6.10 IPIPEIF_RSZ

Table 764. IPIPEIF_RSZ

Address Offset	0x0000 0024		
Physical Address	0x5505 1224 0x5C01 1224	**Instance**	ISS_IPIPEIF_CORTEX-M3 ISS_IPIPEIF_L3
Description	IPIPEIF Horizontal Resizing Parameter on IPIPE data path		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RSZ							

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0000000
6:0	RSZ	Horizontal Resizing Parameter for IPIPE data path Specifies the horizontal resizing parameter. The RSZ register can be configured within 16 to 112 range. This resizing ratio is determined by 16/RSZ (= 1/1 to 1/7) *This bit field is latched by VD.	RW	0x10

3.5.6.11 IPIPEIF_GAIN

Table 765. IPIPEIF_GAIN

Address Offset	0x0000 0028		
Physical Address	0x5505 1228 0x5C01 1228	**Instance**	ISS_IPIPEIF_CORTEX-M3 ISS_IPIPEIF_L3
Description	IPIPEIF Gain Parameter		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								GAIN							

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9:0	GAIN	Gain Parameter Specifies the gain applied to RAW data before it is forwarded to the IPIPE module. The gain value is expressed using the U10Q9 fractional format. The range is from 0.00195 (1/512) to 1.99805(1023/512). By default the unity gain is applied, that is, IPIPEIF_GAIN.GAIN = 0x200. The gain is applied to RAW data only (IPIPEIF_CFG1.INPSRC2 != 3): the gain is not applied if the input data is YCbCr. *This bit field is latched by VD.	RW	0x200

3.5.6.12 IPIPEIF_DPCM

Table 766. IPIPEIF_DPCM

Address Offset	0x0000 002C		
Physical Address	0x5505 122C 0x5C01 122C	**Instance**	ISS_IPIPEIF_CORTEX-M3 ISS_IPIPEIF_L3
Description	IPIPEIF DPCM configuration This register applies only if IPIPEIF_CFG1.UNPACK = 1, that is, RAW8 data is read from SDRAM.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													BITS	PRED	ENA

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	BITS	DPCM bit mode for SDRAM data 0x0: 8bit to 10bit DPCM decompression 0x1: 8bit to 12bit DPCM decompression	RW	0
1	PRED	DPCM prediction mode for SDRAM data 0x0: Simple predictor 0x1: Advanced predictor	RW	0
0	ENA	DPCM decompression enable for SDRAM data. 0x0: DPCM off (no decompression) 0x1: DPCM on	RW	0

3.5.6.13 IPIPEIF_CFG2

Table 767. IPIPEIF_CFG2

Address Offset	0x0000 0030		
Physical Address	0x5505 1230 0x5C01 1230	**Instance**	ISS_IPIPEIF_CORTEX-M3 ISS_IPIPEIF_L3
Description	IPIPEIF Configuration #2		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								YUV8P	YUV8	DFSDIR	RESERVED	YUV16	VDPOL	HDPOL	INTSW

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7	YUV8P	8-bit YUV data unpacking to 16 bits When IPIPEIF_CFG1.INPSRC2 = 0 and IPIPEIF_CFG2.YUV16 = 1, the 8-bit YUV data are transformed into 16-bit YUV data. The way the data are unpacked from 8 bits to 16 bits is controlled by the IPIPEIF_CFG2.YUV8P register. The upper 8 bits of the 16-bit output are set to 0. 0x0: Y output on even pixels C output on odd pixels 0x1: C output on even pixels Y output on odd pixels	RW	0
6	YUV8	YUV 8bit mode When ISIF_CFG1.INPSRC2 = 0 and YUV16 = 1, setting this bit to 1 enables the conversion from 8-bit YUV input to 16-bit YUV. This register is used when the input data from the ISIF module is 8-bit YUV data. 0x0: YUV16 0x1: YUV8 to 16	RW	0
5	DFSDIR	DFS direction Selects the direction of dark frame subtraction. 0x0: VPORT IF(capture frame) - SDRAM (dark frame) 0x1: SDRAM (capture frame) - VPORT IF(dark frame)	RW	0
4	RESERVED	Read returns reset value	RW	0

Bits	Field Name	Description	Type	Reset

3	YUV16	Data type selection. The behavior of this bit field depends upon other register settings. The functionality is best explained with the following pseudo code:
		if ((CFG1.INPSRC2==0 CFG2.YUV16) ||
		CFG1.INPSRC2==3) {
		data_out[15:0] = yuv[15:0]
		}
		else if (CFG1.INPSRC2==1 CFG2.YUV16
		CFG1.UNPACK=1) {
		data_out[15:8] = gain_clip[7:0];
		data_out[7:0] = 0;
		}
		else {
		data_out[15:12] = 0;
		data_out[11: 0] = gain_clip[11:0];
		}
		where:
		o data_out[15:0] = 16-bit YUV or 12-bit RAW data to ipipe
		o yuv[15:0] = 16-bit YUV data from "horizontal pixel decimator" block.
		o gain_clip[11:0] = 12-bit RAW data from "gain" block.
		0x0: 12-bit RAW data
		0x1: 16-bit YUV data
			RW	0

2	VDPOL	VD Sync Polarity When input VD is active low SYNC pulse, this bit needs to be set to 1. 0x0: Positive 0x1: Negative	RW	0
1	HDPOL	HD Sync Polarity When input HD is active low SYNC pulse, this bit needs to be set to 1. 0x0: Positive 0x1: Negative	RW	0
0	INTSW	IPIPEIF interrupt source selection. This register select the interrupt source. 0x0: Start position of VD from VPORT interface 0x1: Start position of VD from ISIF module	RW	0

3.5.6.14 IPIPEIF_INIRSZ

Table 768. IPIPEIF_INIRSZ

Address Offset	0x0000 0034		
Physical Address	0x5505 1234 0x5C01 1234	**Instance**	ISS_IPIPEIF_CORTEX-M3 ISS_IPIPEIF_L3
Description	IPIPEIF resize initial position - IPIPE data path.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		ALNSYNC	INIRSZ												

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13	ALNSYNC	Align the HSYNC, VSYNC to initial position defined by INIRSZ. 0x0: Disable 0x1: Enable	RW	0
12:0	INIRSZ	Offset used to re-initialize the HD/VD position after resizer. From 0 to 8191 PCLK cycles. Skips INIRSZ pixels for every line.	RW	0x0000

3.5.6.15 IPIPEIF_OCLIP

Table 769. IPIPEIF_OCLIP

Address Offset	0x0000 0038		
Physical Address	0x5505 1238 0x5C01 1238	**Instance**	ISS_IPIPEIF_CORTEX-M3 ISS_IPIPEIF_L3
Description	IPIPEIF output clipping value		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				OCLIP											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:0	OCLIP	Output clipping value after gain control on IPIPE data path. This value is in U12Q0 data format.	RW	0xFFF

3.5.6.16 IPIPEIF_DTUDF

Table 770. IPIPEIF_DTUDF

Address Offset	0x0000 003C		
Physical Address	0x5505 123C 0x5C01 123C	**Instance**	ISS_IPIPEIF_CORTEX-M3 ISS_IPIPEIF_L3
Description	IPIPEIF data underflow detection		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															DTUDF

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	DTUDF	Data under flow error status register. Reading 1 shows there is data under flow and at least one data is corrupted while reading from SDRAM. Writing 1 to this register clears (=0) the error (=1) status. Underflow errors are non recoverable at ISP level, need to do a soft reset at ISS level. The IPIPEIF_UDF interrupt is generated when an underflow happens. The interrupt avoids polling this register for errors.	RW	0

3.5.6.17 IPIPEIF_CLKDIV

Table 771. IPIPEIF_CLKDIV

Address Offset	0x0000 0040		
Physical Address	0x5505 1240 0x5C01 1240	**Instance**	ISS_IPIPEIF_CORTEX-M3 ISS_IPIPEIF_L3
Description	IPIPEIF CLOCK DIVIDER		
Type	RW		

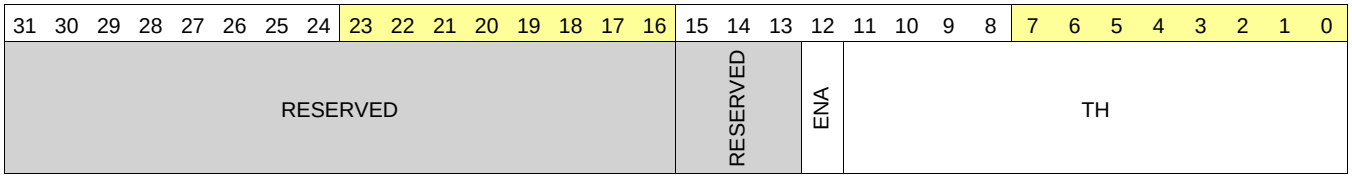
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKDIV															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	CLKDIV	IPIPEIF clock rate configuration IPIPE/IPIPEIF clock frequency = M/N x clk_vpss clock frequency. We have M = CLKDIV[15:8] + 1 and N = CLKDIV[7:0] + 1 When clk_vpss 121.5 MHz, M/N must not be greater that 1/2 value This register is available when IPIPEIF_CFG1.CLKSEL = 1.	RW	0x0001

3.5.6.18 IPIPEIF_DPC1

Table 772. IPIPEIF_DPC1

Address Offset	0x0000 0044		
Physical Address	0x5505 1244 0x5C01 1244	**Instance**	ISS_IPIPEIF_CORTEX-M3 ISS_IPIPEIF_L3
Description	IPIPEIF defect pixel correction #1		
Type	RW		

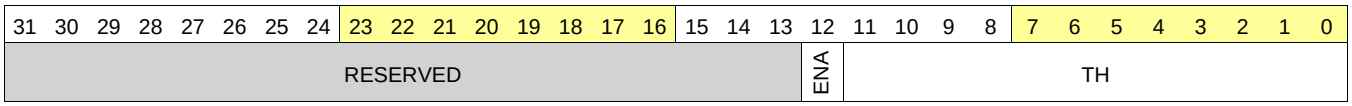


Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12	ENA	DPC enable. Applies DPC for video port data, ISIF input path. 0x0: Disable 0x1: Enable	RW	0
11:0	TH	DPC threshold value	RW	0x000

3.5.6.19 IPIPEIF_DPC2

Table 773. IPIPEIF_DPC2

Address Offset	0x0000 0048		
Physical Address	0x5505 1248 0x5C01 1248	**Instance**	ISS_IPIPEIF_CORTEX-M3 ISS_IPIPEIF_L3
Description	IPIPEIF defect pixel correction #2		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12	ENA	DPC enable. Applies DPC for SDRAM input path. 0x0: Disable 0x1: Enable	RW	0
11:0	TH	DPC threshold value	RW	0x000

3.5.6.20 IPIPEIF_RSZ3A

Table 774. IPIPEIF_RSZ3A

Address Offset	0x0000 0054		
Physical Address	0x5505 1254 0x5C01 1254	**Instance**	ISS_IPIPEIF_CORTEX-M3 ISS_IPIPEIF_L3
Description	IPIPEIF HORIZONTAL RESIZING PARAMETER FOR H3A		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED						DECIM	AVGFILT	RESERVED	RSZ						

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9	DECIM	Pixel Decimation Enable The decimation rate defined by the RSZ bit field. *This bit field is latched by VD. 0x0: No Decimation 0x1: Decimate	RW	0
8	AVGFILT	Averaging Filter It applies a (1, 2, 1) filter for the RGB/YCbCr data. *This bit field is latched by VD. 0x0: Disable 0x1: Enable	RW	0
7	RESERVED		R	0
6:0	RSZ	Horizontal Resizing Parameter for H3A data path Specifies the horizontal resizing parameter. The RSZ register can be configured within 16 to 112 range. This resizing ratio is determined by 16/RSZ (= 1/1 to 1/7) *This bit field is latched by VD.	RW	0x10

3.5.6.21 IPIPEIF_INIRSZ3A

Table 775. IPIPEIF_INIRSZ3A

Address Offset	0x0000 0058		
Physical Address	0x5505 1258 0x5C01 1258	**Instance**	ISS_IPIPEIF_CORTEX-M3 ISS_IPIPEIF_L3
Description	IPIPEIF resize initial position - H3A data path.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		ALNSYNC	INIRSZ												

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13	ALNSYNC	Align the HD, VD to initial position defined by the INIRSZ bit field. It means that HD and VD are effectively shifted by INIRSZ pixel clock cycles. 0x0: Disable 0x1: Enable	RW	0
12:0	INIRSZ	Offset used to re-initialize the HD/VD position after resizer. From 0 to 8191 PCLK cycles. Skips INIRSZ pixels for every line.	RW	0x0000

3.5.7 ISS H3A Registers

Table 776 summarizes the ISS H3A registers.

Table 776. ISS H3A Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ISS_H3A Base Address Cortex-M3 Private Access	ISS_H3A Base Address L3 Interconnect
H3A_PID	R	32	0x0000 0000	0x5505 1400	0x5C01 1400
H3A_PCR	RW	32	0x0000 0004	0x5505 1404	0x5C01 1404
H3A_AFPAX1	RW	32	0x0000 0008	0x5505 1408	0x5C01 1408
H3A_AFPAX2	RW	32	0x0000 000C	0x5505 140C	0x5C01 140C
H3A_AFPAXSTART	RW	32	0x0000 0010	0x5505 1410	0x5C01 1410
H3A_AFIIRSH	RW	32	0x0000 0014	0x5505 1414	0x5C01 1414
H3A_AFBUFST	RW	32	0x0000 0018	0x5505 1418	0x5C01 1418
H3A_AFCOEF010	RW	32	0x0000 001C	0x5505 141C	0x5C01 141C
H3A_AFCOEF032	RW	32	0x0000 0020	0x5505 1420	0x5C01 1420
H3A_AFCOEF054	RW	32	0x0000 0024	0x5505 1424	0x5C01 1424
H3A_AFCOEF076	RW	32	0x0000 0028	0x5505 1428	0x5C01 1428
H3A_AFCOEF098	RW	32	0x0000 002C	0x5505 142C	0x5C01 142C
H3A_AFCOEF0010	RW	32	0x0000 0030	0x5505 1430	0x5C01 1430
H3A_AFCOEF110	RW	32	0x0000 0034	0x5505 1434	0x5C01 1434
H3A_AFCOEF132	RW	32	0x0000 0038	0x5505 1438	0x5C01 1438
H3A_AFCOEF154	RW	32	0x0000 003C	0x5505 143C	0x5C01 143C
H3A_AFCOEF176	RW	32	0x0000 0040	0x5505 1440	0x5C01 1440
H3A_AFCOEF198	RW	32	0x0000 0044	0x5505 1444	0x5C01 1444
H3A_AFCOEF1010	RW	32	0x0000 0048	0x5505 1448	0x5C01 1448
H3A_AEWWIN1	RW	32	0x0000 004C	0x5505 144C	0x5C01 144C
H3A_AEWINSTART	RW	32	0x0000 0050	0x5505 1450	0x5C01 1450
H3A_AEWINBLK	RW	32	0x0000 0054	0x5505 1454	0x5C01 1454
H3A_AEWSUBWIN	RW	32	0x0000 0058	0x5505 1458	0x5C01 1458
H3A_AEWBUFST	RW	32	0x0000 005C	0x5505 145C	0x5C01 145C
H3A_AEWCFG	RW	32	0x0000 0060	0x5505 1460	0x5C01 1460
H3A_LINE_START	RW	32	0x0000 0064	0x5505 1464	0x5C01 1464
H3A_VFV_CFG1	RW	32	0x0000 0068	0x5505 1468	0x5C01 1468
H3A_VFV_CFG2	RW	32	0x0000 006C	0x5505 146C	0x5C01 146C
H3A_VFV_CFG3	RW	32	0x0000 0070	0x5505 1470	0x5C01 1470
H3A_VFV_CFG4	RW	32	0x0000 0074	0x5505 1474	0x5C01 1474
H3A_HVF_THR	RW	32	0x0000 0078	0x5505 1478	0x5C01 1478
H3A_ADVANCED	RW	32	0x0000 007C	0x5505 147C	0x5C01 147C

3.5.7.1 H3A_PID

Table 777. H3A_PID

Address Offset	0x0000 0000		
Physical Address	0x5505 1400 0x5C01 1400	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	Peripheral Revision and Class Information		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCHEME		RESERVED		FUNC												RTL					MAJOR			RESERVED					MINOR			

Bits	Field Name	Description	Type	Reset
31:30	SCHEME		R	0x1
29:28	RESERVED		R	0x0
27:16	FUNC		R	0xD01
15:11	RTL		R	0x00
10:8	MAJOR		R	0x0
7:6	RESERVED		R	0x0
5:0	MINOR		R	0x00

3.5.7.2 H3A_PCR

Table 778. H3A_PCR

Address Offset	0x0000 0004		
Physical Address	0x5505 1404 0x5C01 1404	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	Peripheral Control Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AVE2LMT										OVF	AF_VF_EN	AEW_MED_EN	BUSYAEAWB	AEW_ALAW_EN	AEW_EN	BUSYAF	FVMODE	RGBPOS			MED_TH									AF_MED_EN	AF_ALAW_EN	AF_EN

Bits	Field Name	Description	Type	Reset
31:22	AVE2LMT	AE/AWB Saturation Limit This is the value that all sub sampled pixels in the AE/AWB engine are compared to. If the data is greater or equal to this data then the block is considered saturated.	RW	0x3FF
21	OVF	H3A module overflow status bit. If the H3A module overflows it will keep sending data. The software can read this status bit during vertical blanking period to ensure that no overflow happened while writing out the data to SDRAM. There is also an interrupt at ISP level (H3A_OVF) which can be used to monitor this. 0x0: Read 0: No overflow pending Write 0: Status bit unchanged 0x1: Read 1: Overflow happened while writing out the data. Output data likely to be corrupted. Write 1: Clear the status bit.	RW	0
20	AF_VF_EN	AF Vertical Focus Enable 0x0: 4 Color Horizontal FV only 0x1: 1 Color Horizontal FV and 1 Color Vertical FV	RW	0
19	AEW_MED_EN	AE/AWB Median filter Enable If the median filter is enabled, then the 1st 2 and last 2 pixels in the frame are not filtered. 0x0: Disable Auto Focus median filter 0x1: Enable Auto Focus median filter	RW	0
18	BUSYAEAWB	Busy bit for AE/AWB	R	0
17	AEW_ALAW_EN	AE/AWB A-law Enable 0x0: Disable Auto exposure/white balance A-law table 0x1: Enable Auto exposure/white balance A-law table.	RW	0
16	AEW_EN	AE/AWB enable 0x0: Disable Auto exposure/white balance 0x1: Enable Auto exposure/white balance	RW	0
15	BUSYAF	Busy bit for AF.	R	0
14	FVMODE	Focus Value Accumulation Mode 0x0: Sum mode. 0x1: Peak mode.	RW	0

Bits	Field Name	Description	Type	Reset
13:11	RGBPOS	Red, Green, and blue pixel location in the AF windows This Value is only used if VF is disabled. RGBPOS(0): GR and GB as Bayer pattern RGBPOS(1): RG and GB as Bayer pattern RGBPOS(2): GR and BG as Bayer pattern RGBPOS(3): RG and BG as Bayer pattern RGBPOS(4): GG and RB as custom pattern RGBPOS(5): RB and GG as custom pattern RGBPOS(6): reserved RGBPOS(7): reserved	RW	0x0
10:3	MED_TH	Median filter threshold.	RW	0xFF
2	AF_MED_EN	Auto Focus Median filter Enable If the median filter is enabled, then the 1st 2 and last 2 pixels in the frame are not in the valid region. Therefore the paxel start/end and IIR filter start positions should not be set within the 1st and last 2 pixels. 0x0: Disable AF median filter. 0x1: Enable AF median filter.	RW	0
1	AF_ALAW_EN	AF A-law table enable 0x0: Disable Auto Focus A-law table 0x1: Enable Auto Focus A-law table	RW	0
0	AF_EN	AF enable 0x0: Disable Auto Focus Engine 0x1: Enable Auto Focus Engine	RW	0

3.5.7.3 H3A_AFPAX1

Table 779. H3A_AFPAX1

Address Offset	0x0000 0008		
Physical Address	0x5505 1408 0x5C01 1408	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	Setup for the AF Engine Paxel Configuration		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PAXW								RESERVED								PAXH							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:16	PAXW	AF Engine Paxel Width The width of the paxel is the value of this register plus 1 multiplied by 2. The minimum width is expected to be 8 pixels. * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x00
15:8	RESERVED		R	0x00
7:0	PAXH	AF Engine Paxel Height The height of the paxel is the value of this register plus 1 multiplied by 2 with a final value of 2-256 (even). * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x00

3.5.7.4 H3A_AFPAX2

Table 780. H3A_AFPAX2

Address Offset	0x0000 000C		
Physical Address	0x5505 140C 0x5C01 140C	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	Setup for the AF Engine Paxel Configuration		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								AFINCH								AFINCV								PAXVC							
PAXHC																															

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x000
20:17	AFINCH	AF Engine Column Increments Number of columns to increment in a paxel plus 1 multiplied by 2. Thus, the number of columns that can be skipped between two processed line pairs is 2-32 (even). The starting two columns in a paxel are first processed before this field is applied. This must be set so that there are at least 4 samples on a line when combined with the number of horizontal paxels. * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x0
16:13	AFINCV	AF Engine Line Increments Number of lines to increment in a Paxel plus 1 multiplied by 2. Incrementing the line in a paxel is always done on a line pair due to the fact that the RGB pattern falls in two lines. If all the lines are to be processed, this field should be set to zero, and thus line count is incremented by 2 following a line pair. Thus, the number of lines that can be skipped between two processed line pairs is 0-30 (even). The starting two lines in a paxel are first processed before this field is applied. * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x0
12:6	PAXVC	AF Engine Vertical Paxel Count The number of paxels in the vertical direction plus 1. The maximum number of vertical paxels in a frame should not exceed 128. The value should be set to ensure that the bandwidth requirements and buffer size are not exceeded. * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x00
5:0	PAXHC	AF Engine Horizontal Paxel Count The number of paxels in the horizontal direction plus 1. It is illegal to set a number that is greater than 35 (total of 36 paxels in the horizontal direction). The minimum number of paxels should be 2 (valid range for the field is 1-35). * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x00

3.5.7.5 H3A_AFPAXSTART

Table 781. H3A_AFPAXSTART

Address Offset	0x0000 0010		
Physical Address	0x5505 1410 0x5C01 1410	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	Start Position for AF Engine Paxels		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PAXSH								RESERVED								PAXSV							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	PAXSH	AF Engine Paxel Horizontal start position Range: 2-4094 PAXSH must be equal to or greater than (IIRSH + 2) This value must be even, if Vertical mode is not enabled. If Vertical mode is enabled, then the lower bit of PAXSH and IIRSH must be equal. * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x000
15:12	RESERVED		R	0x0
11:0	PAXSV	AF Engine Paxel Vertical start position Range: 0-4095 Sets the vertical line for the first paxel. This value must be greater then or equal to 8, if the vertical mode is enabled. * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x000

3.5.7.6 H3A_AFIIRSH

Table 782. H3A_AFIIRSH

Address Offset	0x0000 0014		
Physical Address	0x5505 1414 0x5C01 1414	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	Start Position for IIRSH		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				IIRSH											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:0	IIRSH	AF Engine IIR Horizontal Start Position Range from 0-4094. When the horizontal position of a line equals this value the shift registers are cleared on the next pixel. This value must be even if Vertical mode is not enabled. If vertical mode is enabled then the lower bit must match the paxel horizontal start. position. * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x000

3.5.7.7 H3A_AFBUFST

Table 783. H3A_AFBUFST

Address Offset	0x0000 0018		
Physical Address	0x5505 1418 0x5C01 1418	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	SDRAM destination address for AF engine statistics		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFBUFST																											RESERVED			

Bits	Field Name	Description	Type	Reset
31:5	AFBUFST	SDRAM destination address for AF engine statistics The SDRAM destination address for the AF statistics. The 6 LSBs are ignored, address must be on a 64-byte boundary. This field can be altered even when the AF is busy. Change will take place only for the next frame. However, note that reading this register will always give the latest value.	RW	0x0000000
4:0	RESERVED		R	0x00

3.5.7.8 H3A_AFCOEF010

Table 784. H3A_AFCOEF010

Address Offset	0x0000 001C		
Physical Address	0x5505 141C 0x5C01 141C	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	IIR filter coefficient data for SET 0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COEFF1								RESERVED								COEFF0							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	COEFF1	AF Engine IIR filter Coefficient #1 (Set 0) The range is signed -32 = value = 31 +63/64	RW	0x000
15:12	RESERVED		R	0x0
11:0	COEFF0	AF Engine IIR filter Coefficient #0 (Set 0) The range is signed -32 = value = 31 +63/64	RW	0x000

3.5.7.9 H3A_AFCOEF032

Table 785. H3A_AFCOEF032

Address Offset	0x0000 0020		
Physical Address	0x5505 1420 0x5C01 1420	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	IIR filter coefficient data for SET 0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COEFF3								RESERVED								COEFF2							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	COEFF3	AF Engine IIR filter Coefficient #3 (Set 0) The range is signed -32 = value = 31 +63/64	RW	0x000
15:12	RESERVED		R	0x0
11:0	COEFF2	AF Engine IIR filter Coefficient #2 (Set 0) The range is signed -32 = value = 31 +63/64	RW	0x000

3.5.7.10 H3A_AFCOEF054

Table 786. H3A_AFCOEF054

Address Offset	0x0000 0024		
Physical Address	0x5505 1424 0x5C01 1424	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	IIR filter coefficient data for SET 0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COEFF5								RESERVED								COEFF4							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	COEFF5	AF Engine IIR filter Coefficient #5 (Set 0) The range is signed -32 = value = 31 +63/64	RW	0x000
15:12	RESERVED		R	0x0
11:0	COEFF4	AF Engine IIR filter Coefficient #4 (Set 0) The range is signed -32 = value = 31 +63/64	RW	0x000

3.5.7.11 H3A_AFCOEF076

Table 787. H3A_AFCOEF076

Address Offset	0x0000 0028		
Physical Address	0x5505 1428 0x5C01 1428	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	IIR filter coefficient data for SET 0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COEFF7								RESERVED								COEFF6							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	COEFF7	AF Engine IIR filter Coefficient #7 (Set 0) The range is signed -32 = value = 31 +63/64	RW	0x000
15:12	RESERVED		R	0x0
11:0	COEFF6	AF Engine IIR filter Coefficient #6 (Set 0) The range is signed -32 = value = 31 +63/64	RW	0x000

3.5.7.12 H3A_AFCOEF098

Table 788. H3A_AFCOEF098

Address Offset	0x0000 002C		
Physical Address	0x5505 142C 0x5C01 142C	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	IIR filter coefficient data for SET 0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COEFF9								RESERVED								COEFF8							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	COEFF9	AF Engine IIR filter Coefficient #9 (Set 0) The range is signed -32 = value = 31 +63/64	RW	0x000
15:12	RESERVED		R	0x0
11:0	COEFF8	AF Engine IIR filter Coefficient #8 (Set 0) The range is signed -32 = value = 31 +63/64	RW	0x000

3.5.7.13 H3A_AFCOEF0010

Table 789. H3A_AFCOEF0010

Address Offset	0x0000 0030		
Physical Address	0x5505 1430 0x5C01 1430	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	IIR filter coefficient data for SET 0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				COEFF10											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:0	COEFF10	AF Engine IIR filter Coefficient #10 (Set 0) The range is signed -32 = value = 31 +63/64	RW	0x000

3.5.7.14 H3A_AFCOEF110

Table 790. H3A_AFCOEF110

Address Offset	0x0000 0034		
Physical Address	0x5505 1434 0x5C01 1434	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	IIR filter coefficient data for SET 1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COEFF1								RESERVED								COEFF0							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	COEFF1	AF Engine IIR filter Coefficient #1 (Set 1) The range is signed -32 = value = 31 +63/64	RW	0x000
15:12	RESERVED		R	0x0
11:0	COEFF0	AF Engine IIR filter Coefficient #0 (Set 1) The range is signed -32 = value = 31 +63/64	RW	0x000

3.5.7.15 H3A_AFCOEF132

Table 791. H3A_AFCOEF132

Address Offset	0x0000 0038		
Physical Address	0x5505 1438 0x5C01 1438	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	IIR filter coefficient data for SET 1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COEFF3								RESERVED								COEFF2							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	COEFF3	AF Engine IIR filter Coefficient #3 (Set 1) The range is signed -32 = value = 31 +63/64	RW	0x000
15:12	RESERVED		R	0x0
11:0	COEFF2	AF Engine IIR filter Coefficient #2 (Set 1) The range is signed -32 = value = 31 +63/64	RW	0x000

3.5.7.16 H3A_AFCOEF154

Table 792. H3A_AFCOEF154

Address Offset	0x0000 003C		
Physical Address	0x5505 143C 0x5C01 143C	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	IIR filter coefficient data for SET 1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COEFF5								RESERVED								COEFF4							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	COEFF5	AF Engine IIR filter Coefficient #5 (Set 1) The range is signed -32 = value = 31 +63/64	RW	0x000
15:12	RESERVED		R	0x0
11:0	COEFF4	AF Engine IIR filter Coefficient #4 (Set 1) The range is signed -32 = value = 31 +63/64	RW	0x000

3.5.7.17 H3A_AFCOEF176

Table 793. H3A_AFCOEF176

Address Offset	0x0000 0040		
Physical Address	0x5505 1440 0x5C01 1440	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	IIR filter coefficient data for SET 1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COEFF7								RESERVED								COEFF6							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	COEFF7	AF Engine IIR filter Coefficient #7 (Set 1) The range is signed -32 = value = 31 +63/64	RW	0x000
15:12	RESERVED		R	0x0
11:0	COEFF6	AF Engine IIR filter Coefficient #6 (Set 1) The range is signed -32 = value = 31 +63/64	RW	0x000

3.5.7.18 H3A_AFCOEF198

Table 794. H3A_AFCOEF198

Address Offset	0x0000 0044		
Physical Address	0x5505 1444 0x5C01 1444	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	IIR filter coefficient data for SET 1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COEFF9								RESERVED				COEFF8											

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	COEFF9	AF Engine IIR filter Coefficient #9 (Set 1) The range is signed -32 = value = 31 +63/64	RW	0x000
15:12	RESERVED		R	0x0
11:0	COEFF8	AF Engine IIR filter Coefficient #8 (Set 1) The range is signed -32 = value = 31 +63/64	RW	0x000

3.5.7.19 H3A_AFCOEF1010

Table 795. H3A_AFCOEF1010

Address Offset	0x0000 0048		
Physical Address	0x5505 1448 0x5C01 1448	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	IIR filter coefficient data for SET 1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				COEFF10											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:0	COEFF10	AF Engine IIR filter Coefficient #10 (Set 1) The range is signed -32 = value = 31 +63/64	RW	0x000

3.5.7.20 H3A_AEWWIN1

Table 796. H3A_AEWWIN1

Address Offset	0x0000 004C		
Physical Address	0x5505 144C 0x5C01 144C	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	Configuration for AE/AWB Windows.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WINH								RESERVED			WINW										WINVC								WINHC			

Bits	Field Name	Description	Type	Reset
31:24	WINH	AE/AWB Engine Window Height This specifies the window height in an even number of pixels, the window height is the value plus 1 multiplied by 2. The final value can be from 2-512 (even). * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x00
23:21	RESERVED		R	0x0
20:13	WINW	AE/AWB Engine Window Width This specifies the window width in an even number of pixels, the window width is the value plus 1 multiplied by 2. The minimum width is expected to be 8 pixels. * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x00
12:6	WINVC	AE/AWB Engine Vertical Window Count The number of windows in the vertical direction plus 1. The maximum number of vertical windows in a frame should not exceed 128. The value should be set to ensure that the bandwidth requirements and buffer size are not exceeded. * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x00
5:0	WINHC	AE/AWB Engine Horizontal Window Count The number of horizontal windows plus 1. The maximum number of horizontal windows is 35 plus 1 (36). The minimum number of windows should be 2 (valid range for the field is 1-35). * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x00

3.5.7.21 H3A_AEWINSTART

Table 797. H3A_AEWINSTART

Address Offset	0x0000 0050		
Physical Address	0x5505 1450 0x5C01 1450	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	Start position for AE/AWB Windows.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								WINSV								RESERVED								WINSH							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	WINSV	AE/AWB Engine Vertical Window Start Position Sets the first line for the first window. Range 0-4095 * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x000
15:12	RESERVED		R	0x0
11:0	WINSH	AE/AWB Engine Horizontal Window Start Position Sets the horizontal position for the first window on each line. Range 0-4095 * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x000

3.5.7.22 H3A_AEWINBLK

Table 798. H3A_AEWINBLK

Address Offset	0x0000 0054		
Physical Address	0x5505 1454 0x5C01 1454	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	Start position and height for black line of AE/AWB Windows		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								WINSV								RESERVED								WINH							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	WINSV	AE/AWB Engine Vertical Window Start Position for single black line of windows Sets the first line for the single black line of windows. * This value is shadowed and latched on the rising edge of VSYNC. Range 0-4095 Note that the horizontal start and the horizontal number of windows will be similar to the regular windows	RW	0x000
15:7	RESERVED		R	0x000
6:0	WINH	AE/AWB Engine Window Height for the single black line of windows This specifies the window height in an even number of pixels, the window height is the value plus 1 multiplied by 2. The final value can be from 2-256 (even) * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x00

3.5.7.23 H3A_AEWSUBWIN

Table 799. H3A_AEWSUBWIN

Address Offset	0x0000 0058		
Physical Address	0x5505 1458 0x5C01 1458	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	Configuration for subsample data in AE/AWB window.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				AEWINCV				RESERVED				AEWINCH			

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:8	AEWINCV	AE/AWB Engine Vertical Sampling Point Increment Sets vertical distance between sub-samples within a window plus 1 multiplied by 2. The final range is 2-32. * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x0
7:4	RESERVED		R	0x0
3:0	AEWINCH	AE/AWB Engine Horizontal Sampling Point Increment Sets horizontal distance between sub-samples within a window plus 1 multiplied by 2. The final range is 2-32. * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x0

3.5.7.24 H3A_AEWBUFST

Table 800. H3A_AEWBUFST

Address Offset	0x0000 005C		
Physical Address	0x5505 145C 0x5C01 145C	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	SDRAM destination address for AE/AWB engine statistics		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AEWBUFST																													RESERVED		

Bits	Field Name	Description	Type	Reset
31:5	AEWBUFST	SDRAM destination address for AE/AWB engine statistics The start location in SDRAM for the AE/AWB statistics. The 6 LSB are ignored, address should be on a 64-byte boundary This field can be altered even when the AE/AWB is busy. Change will take place only for the next frame. However, note that reading this register will always give the latest value.	RW	0x0000000
4:0	RESERVED		R	0x00

3.5.7.25 H3A_AEWCFG

Table 801. H3A_AEWCFG

Address Offset	0x0000 0060		
Physical Address	0x5505 1460 0x5C01 1460	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	Configuration for AE/AWB		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						AEFMT		RESERVED				SUMSHFT			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9:8	AEFMT	AE/AWB output format 0 = sum of squares 1 = min/max 2 = sum only; no sum of squares or min/max * This value is shadowed and latched on the rising edge of VSYNC	RW	0x0
7:4	RESERVED		R	0x0
3:0	SUMSHFT	AE/AWB engine shift value for the accumulation of pixel values This bit field sets the right shift value that is applied on the result of the pixel accumulation before it is stored in the packet. The accumulation takes place on 26 bits, which is enough for 10-bit data and a maximum widow size of 512 x 512 that results into the accumulation of 256 x 256 pixels of the same color. The shift value must be set such that the result fits on 16 bits. SUMSHFT = right shift value. Range: 0 -15 * This value is shadowed and latched on the rising edge of VSYNC	RW	0x0

3.5.7.26 H3A_LINE_START

Table 802. H3A_LINE_START

Address Offset	0x0000 0064		
Physical Address	0x5505 1464 0x5C01 1464	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	Line Framing Logic Register In certain cases the number of clock cycles between HD pulses will be greater than the line buffer included in the H3A module. The framing module prior to the line buffer enables to control the data which is input to the line buffer.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLV																LINE_START															

Bits	Field Name	Description	Type	Reset
31:16	SLV	Start Line Vertical Specifies how many lines after the VD rising edge the real frame starts.	W	0x0000
15:0	LINE_START	Line Start The framing module uses the LINE_START bit field to find the position of the first pixel to place into the line buffer. Range: 0-65535	RW	0x0000

3.5.7.27 H3A_VFV_CFG1

Table 803. H3A_VFV_CFG1

Address Offset	0x0000 0068		
Physical Address	0x5505 1468 0x5C01 1468	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	Vertical focus value configuration 1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VCOEF1_3								VCOEF1_2								VCOEF1_1								VCOEF1_0							

Bits	Field Name	Description	Type	Reset
31:24	VCOEF1_3	Vertical FV FIR 1 coefficient 3	RW	0x00
23:16	VCOEF1_2	Vertical FV FIR 1 coefficient 2	RW	0x00
15:8	VCOEF1_1	Vertical FV FIR 1 coefficient 1	RW	0x00
7:0	VCOEF1_0	Vertical FV FIR 1 coefficient 0	RW	0x00

3.5.7.28 H3A_VFV_CFG2

Table 804. H3A_VFV_CFG2

Address Offset	0x0000 006C		
Physical Address	0x5505 146C 0x5C01 146C	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	Vertical focus value configuration 2.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VTHR1																RESERVED								VCOEF1_4							

Bits	Field Name	Description	Type	Reset
31:16	VTHR1	Threshold for vertical FV FIR 1	RW	0x0000
15:8	RESERVED		R	0x00
7:0	VCOEF1_4	Vertical FV FIR 1 coefficient 4	RW	0x00

3.5.7.29 H3A_VFV_CFG3

Table 805. H3A_VFV_CFG3

Address Offset	0x0000 0070		
Physical Address	0x5505 1470 0x5C01 1470	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	Vertical focus value configuration 4.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VCOEF2_3								VCOEF2_2								VCOEF2_1								VCOEF2_0							

Bits	Field Name	Description	Type	Reset
31:24	VCOEF2_3	Vertical FV FIR 2 coefficient 3	RW	0x00
23:16	VCOEF2_2	Vertical FV FIR 2 coefficient 2	RW	0x00
15:8	VCOEF2_1	Vertical FV FIR 2 coefficient 1	RW	0x00
7:0	VCOEF2_0	Vertical FV FIR 2 coefficient 0	RW	0x00

3.5.7.30 H3A_VFV_CFG4

Table 806. H3A_VFV_CFG4

Address Offset	0x0000 0074		
Physical Address	0x5505 1474 0x5C01 1474	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	Vertical focus value configuration 4.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VTHR2																RESERVED								VCOEF2_4							

Bits	Field Name	Description	Type	Reset
31:16	VTHR2	Threshold for vertical FV FIR 2	RW	0x0000
15:8	RESERVED		R	0x00
7:0	VCOEF2_4	Vertical FV FIR 2 coefficient 4	RW	0x00

3.5.7.31 H3A_HVF_THR

Table 807. H3A_HVF_THR

Address Offset	0x0000 0078		
Physical Address	0x5505 1478 0x5C01 1478	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	Horizontal Focus Value Threshold		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HTHR2																HTHR1															

Bits	Field Name	Description	Type	Reset
31:16	HTHR2	Threshold for horizontal FV IIR 2	RW	0x0000
15:0	HTHR1	Threshold for horizontal FV IIR 1	RW	0x0000

3.5.7.32 H3A_ADVANCED

Table 808. H3A_ADVANCED

Address Offset	0x0000 007C		
Physical Address	0x5505 147C 0x5C01 147C	**Instance**	ISS_H3A_CORTEX-M3 ISS_H3A_L3
Description	Normal and Advanced AF stats collection mode		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																RESERVED															AF_MODE

Bits	Field Name	Description	Type	Reset
31:15	ID	0x0: This bit field must be set to 0xCA00 to enable AF advanced mode.	RW	0x00000
14:1	RESERVED		R	0x0000
0	AF_MODE	AF engine mode. 0x0: Normal Mode 0x1: Advanced mode. H3A_ADVANCED.ID must be set to 0xCA00 to enable this functionality.	RW	0

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265