OMAP5910 Dual-Core Processor DSP Subsystems Interrupts Reference Guide

Literature Number: SPRU923
March 2005

OMAP™
TEXAS INSTRUMENTS TECHNOLOGY

TEXAS INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI 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 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. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

Preface

Read This First

About This Manual

This document describes the OMAP5910 multimedia processor DSP subsystem interrupts.

Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers are shown with the suffix h. For example, the following number is 40 hexadecimal (decimal 64): 40h.

Related Documentation From Texas Instruments

The following documents describe the OMAP5910 device and related peripherals. Copies of these documents are available on the Internet at www.ti.com. *Tip:* Enter the literature number in the search box provided at www.ti.com.

OMAP5910 Dual-Core Processor MPU Subsystem Reference Guide (literature number SPRU671)

OMAP5910 Dual-Core Processor DSP Subsystem Reference Guide (literature number SPRU672)

OMAP5910 Dual-Core Processor Memory Interface Traffic Controller Reference Guide (literature number SPRU673)

OMAP5910 Dual-Core Processor System DMA Controller Reference Guide (literature number SPRU674)

OMAP5910 Dual-Core Processor LCD Controller Reference Guide (literature number SPRU675)

OMAP5910 Dual-Core Processor Universal Asynchronous Receiver/Transmitter (UART) Devices Reference Guide (literature number SPRU676)

OMAP5910 Dual-Core Processor Universal Serial Bus (USB) and Frame Adjustment Counter (FAC) Reference Guide (literature number SPRU677)

OMAP5910 Dual-Core Processor Clock Generation and System Reset Management Reference Guide (literature number SPRU678)

OMAP5910 Dual-Core Processor General-Purpose Input/Output (GPIO) Reference Guide (literature number SPRU679)

OMAP5910 Dual-Core Processor MMC/SD Reference Guide (literature number SPRU680)

OMAP5910 Dual-Core Processor Inter-Integrated Circuit (I2C) Controller Reference Guide (literature number SPRU681)

OMAP5910 Dual-Core Processor Timer Reference Guide (literature number SPRU682)

OMAP5910 Dual-Core Processor Inter-Processor Communication Reference Guide (literature number SPRU683)

OMAP5910 Dual-Core Processor Camera Interface Reference Guide (literature number SPRU684)

OMAP5905 Dual-Core Processor Multichannel Serial Interface (MCSI) Reference Guide (literature number SPRU685)

OMAP5910 Dual-Core Processor Micro-Wire Interface Reference Guide (literature number SPRU686)

OMAP5910 Dual-Core Processor Real-Time Clock (RTC) Reference Guide (literature number SPRU687)

OMAP5910 Dual-Core Processor HDQ/1-Wire Interface Reference Guide (literature number SPRU688)

OMAP5910 Dual-Core Processor PWL, PWT, and LED Peripheral Reference Guide (literature number SPRU689)

OMAP5910 Dual-Core Processor Multichannel Buffered Serial Port (McBSP) Reference Guide (literature number SPRU708)

Trademarks

OMAP and the OMAP symbol are trademarks of Texas Instruments.

Contents

Figures

Tables

This page is intentionally left blank.

DSP Subsystem Interrupts

This document describes the OMAP5910 multimedia processor DSP subsystem interrupts.

1 Architecture Overview

The Digital Signal Processor (DSP) Subsystem is a collection of modules which include the TMS320C55x CPU processor along with its hardware accelerators, tightly coupled memory, instruction cache, and dedicated DMA, the interfaces it uses to communicate with rest of the OMAP device, as well as a number of peripherals.

The TMS320C55x core processor (also referred to as the DSP core) and the peripherals included in the DSP subsystem interface with:

- The MPU core via the microprocessor unit interface (MPUI)
- Various standard memories via the external memory interface (EMIF)
- Various system peripherals via two TI peripheral bus (TIPB) bridges

Figure 1 shows the OMAP5910 device with the DSP subsystem highlighted. Figure 2 shows the subsystem and the modules with which it interfaces.

Figure 1. Highlight of DSP Subsystem

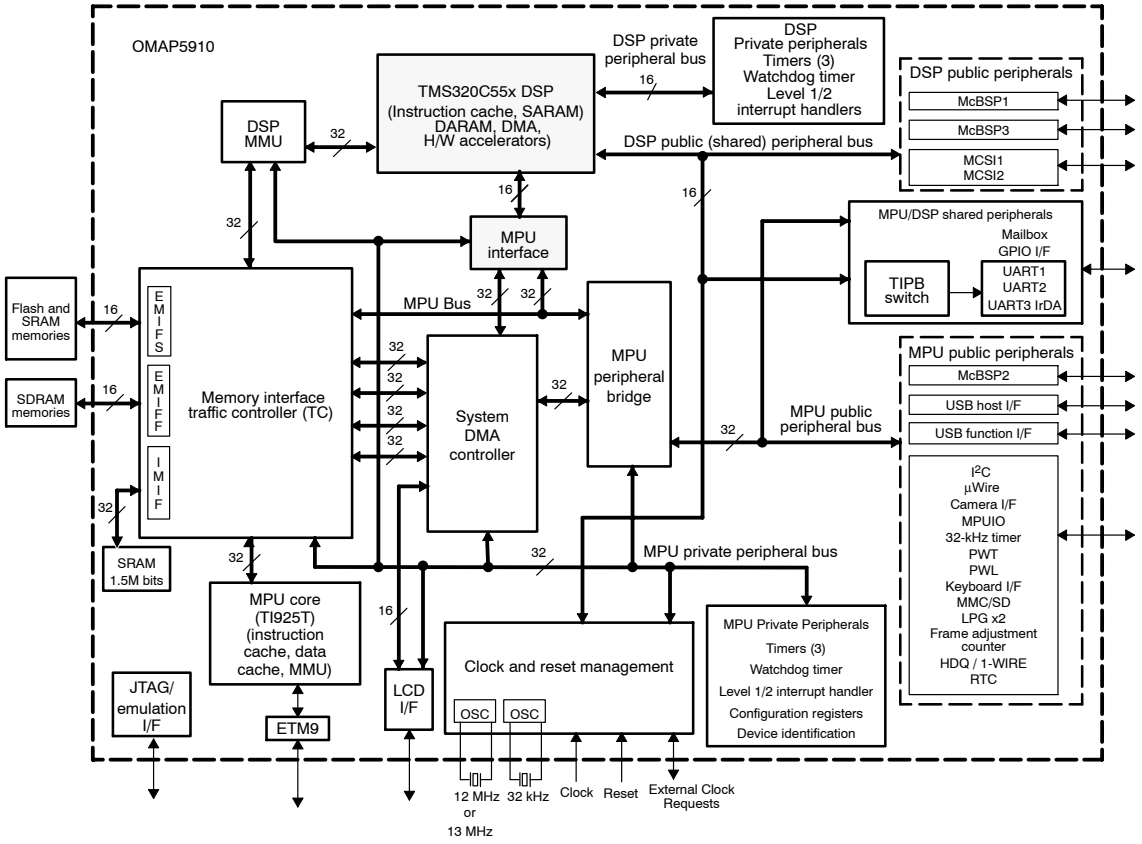
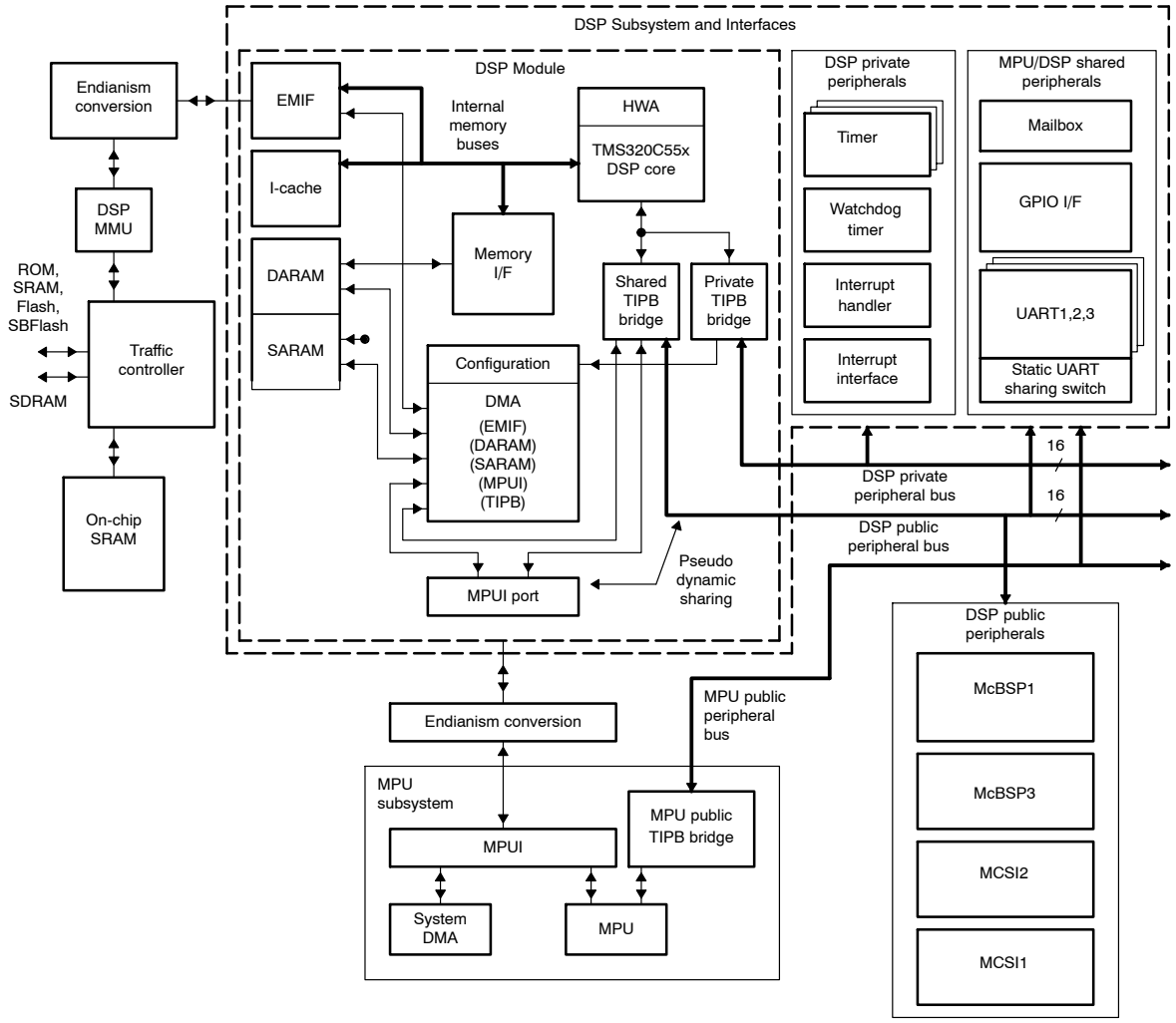


Figure 2. DSP Subsystem and Modules



The DSP subsystem is composed of several portions: the DSP module, the peripherals that surround that module, and several interfaces used to communicate with the rest of the OMAP modules. Each portion has the following components:

- ❏ DSP module:
  - TMS320C55x (C55x) DSP core.
  - Tightly coupled hardware accelerators: discrete cosine transform/inverse discrete cosine transform (DCT/IDCT), motion estimation, and half-pixel interpolation.

- Tightly coupled memories and their interfaces: dual-access RAM (DARAM), single-access RAM (SARAM), programmable dynamic ROM, and instruction cache (I-Cache).
- A six-channel DMA controller that can copy memory contents from one address to another without DSP core intervention.

- DSP subsystem interfaces:
  - External memory interface (EMIF) that connects the DSP core to external and loosely coupled memories.
  - An MPUI port that permits access to DSP resources by the MPU and system DMA.
  - TIPB that provides two external bus interfaces for private and public peripherals.

- DSP subsystem peripherals:
  - Private peripherals are on the DSP private peripheral bus, and as such can only be accessed by the DSP core. DSP private peripherals include:
    - Three 32-bit timers
    - A watchdog timer
    - Interrupt handlers
  - Public peripherals are on the DSP public peripheral bus. These peripherals are directly accessible by the DSP core and DSP DMA. The MPU core can also access these peripherals through the MPUI port. DSP public peripherals include:
    - Two multichannel buffered serial ports (McBSPs)
    - Two multichannel serial interfaces (MCSIs)
  - The DSP CPU and DMA also have access to system peripherals (also referred to as shared peripherals). Shared peripherals are connected to both the MPU public peripheral bus and the DSP public peripheral bus. Shared peripherals include:
    - A mailbox module to permit interrupt-based signaling between the DSP and MPU cores
    - Three universal asynchronous receiver/transmitter (UART) modules
    - A general-purpose input/output (GPIO) module

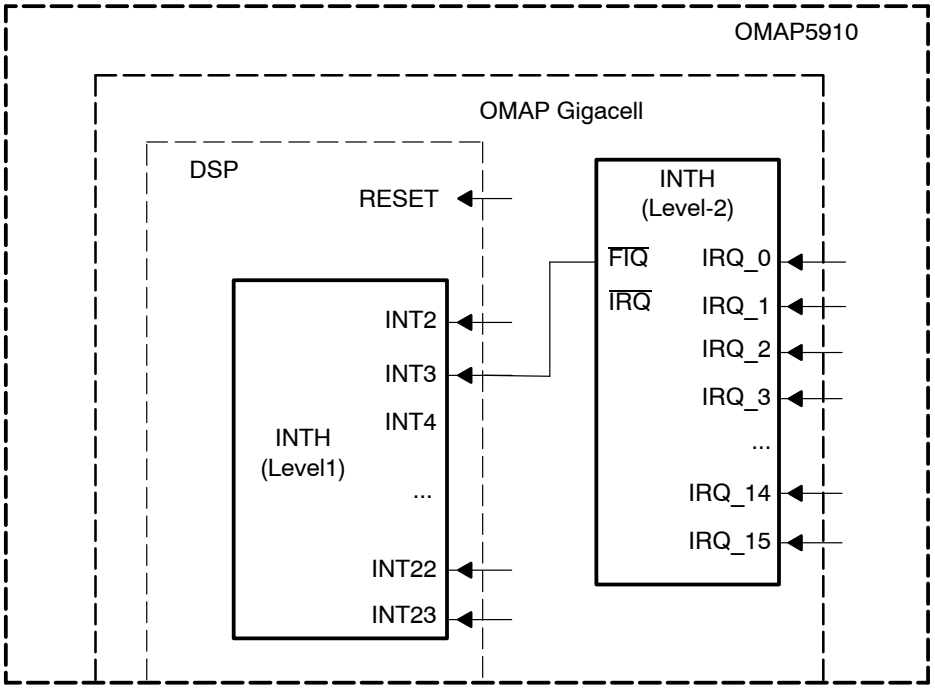
2 Interrupt Handlers

The interrupt handler handles interrupts generated by modules and peripherals (DMA controller, MMU, watchdog timer, timers, software interrupt, etc.). The interrupt handler processes, on a programmable basis, edge-triggered or level-sensitive interrupts. All interrupts are maskable (that is, individually enabled and disabled) with internal registers except for reset. Information regarding the source of the interrupt can be read back. Interrupt priority is programmable to provide flexibility for different applications. All of these interrupts are routed to the DSP core interrupt inputs.

Interrupts are handled through two cascaded interrupt controllers. One is the level 1 handler that is inside the DSP core. The second is the level 2 handler that is external to the DSP and functions similarly to the MPU interrupt handler.

- The 22 level 1 interrupts are handled by the DSP internal interrupt controller provided by the DSP core (see Table 1, *DSP Level 1 Interrupt Mapping*).
- The 16 level 2 interrupts are handled by the external interrupt controller, cascaded into INT3 of the DSP internal interrupt controller.

Figure 3. DSP Interrupt Handler Cascade



2.1 Level 1 Interrupts

The DSP level 1 interrupt controller receives interrupts from peripherals and sends them to the DSP core (see Table 1). The TI peripheral bus is responsible for prioritizing, capturing, and synchronizing interrupts, before sending them to the DSP. The level 1 interrupt controller has 22 maskable interrupts. Of the 22 maskable interrupts, 21 are peripheral interrupts and the remaining one is an MPU interrupt.

The Level 1 DSP interrupts must be at least two DSP_CLK cycles long in order for the DSP to recognize it. To ensure that this requirement is met, the DSP is provided with an internal hardware module called the DSP interrupt interface (see section 3, *DSP Interrupt Interface*).

Table 1. Level 1 Interrupt Mapping

Level 1 Interrupt	Priority	DSP Interrupt	Vector Location	DSP IFR_bit/IMT_bit (26:0)
RESET	0		FFFF00	
NMI†	1		FFFF08	
Emulator/Test	3	INT2	FFFF10	2
Level-2 INTH FIQ	5	INT3	FFFF18	3
TC_ABORT	6	INT4	FFFF20	4
MAILBOX 1	7	INT5	FFFF28	5
Reserved	9	INT6	FFFF30	6
GPIO	10	INT7	FFFF38	7
TIMER3	11	INT8	FFFF40	8
DMA_channel_1	13	INT9	FFFF48	9
MPU	14	INT10	FFFF50	10
Reserved	15	INT11	FFFF58	11
UART	17	INT12	FFFF60	12
WDGTIMER	18	INT13	FFFF68	13
DMA_channel_4	21	INT14	FFFF70	14
DMA_channel_5	22	INT15	FFFF78	15
EMIF	4	INT16	FFFF80	16
Local Bus	8	INT17	FFFF88	17
DMA_channel_0	12	INT18	FFFF90	18
Mailbox 2	16	INT19	FFFF98	19
DMA_channel_2	19	INT20	FFFFA0	20
DMA_channel_3	20	INT21	FFFFA8	21
TIMER2	23	INT22	FFFFB0	22
TIMER1	24	INT23	FFFFB8	23

† NMI is not physically connected on OMAP devices, it is included here for compatibility with other C55x documentation.

2.2 Level 2 Interrupts

The level 2 interrupt controller provides up to 16 prioritized and maskable interrupts to the DSP core.

The level 2 interrupt controller resides on the 16-bit TI peripheral bus. This module is clocked by the DSP_INTH_CK clock, which is fixed at half the CK_GEN2 frequency (see SPRU678, *OMAP5910 Clock Generation and System Reset Management Reference Guide*, for details). Configuration registers configure incoming interrupts as level-sensitive or edge-sensitive interrupts.

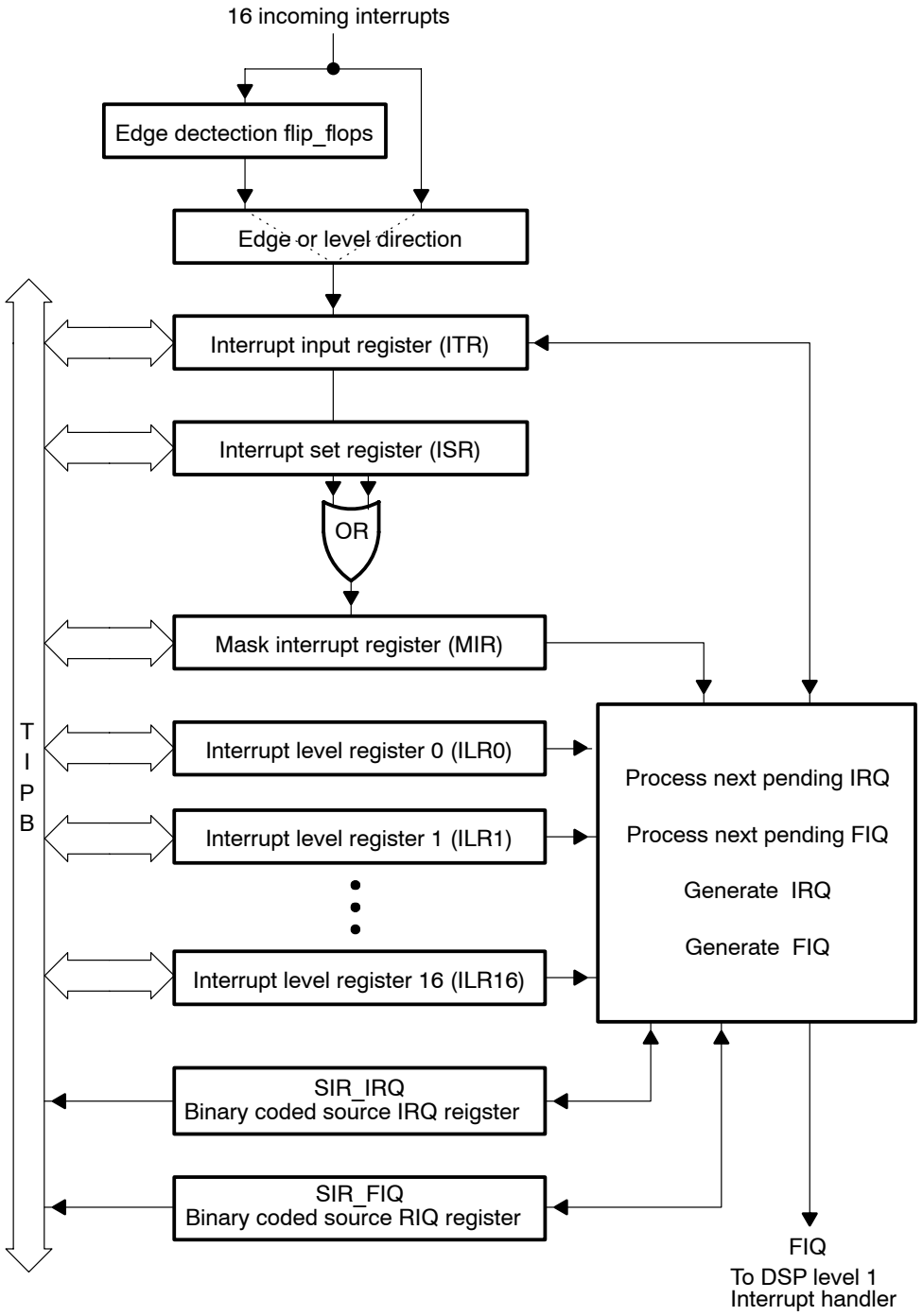
One interrupt-level register (ILR) is associated with each incoming interrupt. It assigns a priority to the corresponding interrupt, determines whether it is to be level- or edge-sensitive, and determines to which DSP interrupt (fast interrupt request (FIQ) or low priority interrupt request (IRQ)) the incoming interrupt goes. If several interrupts have the same priority level assigned, they are serviced in a predefined order.

All level 2 interrupts are routed to FIQ. IRQ output is unconnected.

The interrupt controller also provides a 16-bit software interrupt register. This 16-bit register corresponds to the same 16-bit external interrupt lines. By writing a 1 to the targeted bit, an interrupt is generated if the corresponding ILR is set to edge sensitive; otherwise, no interrupt is generated. External interrupt requests and internal software requests are “OR”ed together first before being sent to the interrupt controller to be serviced. The software interrupt register is always read back with a zero. This allows simulation of external interrupts to test the corresponding interrupt driver by using the software interrupt mechanism at any time.

The FIQ outputs from the interrupt controller are reset by writing a 1 to the corresponding bit of the control register.

Figure 4. Level 2 Interrupt Control Flow



2.2.1 Interrupt Sequence

1) One or several incoming interrupts occcur, setting the corresponding ITR bits.

2) At this time, two possibilities exist:

   - If there is only one active incoming interrupt and FIQ is not already active, the interrupt controller sends an FIQ.
   - When several incoming interrupts are active, the interrupt controller must determine which is the new interrupt to be serviced. It does so by comparing the priority level of an interrupt with the one held in a dedicated register N_FIQ and stores the interrupt with the highest priority in the dedicated register N_FIQ. It performs this comparison until all the active interrupts have been processed. If FIQ is not already active, the interrupt controller sends an FIQ.

3) When an FIQ is sent, the source interrupt encoded register (SIR_FIQ) is updated (indicating the interrupt contained in N_FIQ) and the priority resolver is reset (and restarted if necessary).

4) To determine which incoming interrupt has requested a DSP action, the source interrupt encoded register (SIR_FIQ) must be read. The register contains an encoded number that tells which interrupt lines are being serviced. After that, it runs the corresponding subroutine.

5) To finish the interrupt sequence, the DSP software must first clear the interrupt bit in the interrupt input register (ITR) that is being serviced. This is done by writing a 0 to the corresponding bit in the interrupt input register (ITR) directly or by reading the source interrupt encoded register (SIR_FIQ). For a level-sensitive interrupt, the level must also be removed for the next interrupt to occur. Then set a dedicated bit (NEW_FIQ_AGR bit of the control register) in order to reset the FIQ output and the source encoded register (SIR_FIQ), thus allowing a new FIQ generation.

2.2.2 DSP Accessible Registers

DSP start word address (hex): 0x004800

Bus width: 16 bits

DSP address of a register = Start address + offset TIPB address

Table 2 lists the interrupt handler level 2 registers. Table 3 through Table 9 describe the register bits.

Table 2. Interrupt Handler Level 2 Registers

Register Name	Description	Default Value	Read/ Write	Size	DSP Address
ITR	Interrupt	00000000	R/W	16 bits	0x004800
MIR	Interrupt mask	FFFFFFFF	R/W	16 bits	0x004802
SIR_IRQ	Interrupt encoded source (IRQ)	00	R	4 bits	0x004804
SIR_FIQ	Interrupt encoded source (FIQ)	00	R	4 bits	0x004806
CONTROL_REG	Interrupt control	0	R/W	2 bits	0x004808
ISR	Software interrupt set	00000000	R/W	16 bits	0x00480A
ILR0	Interrupt priority level bit 0	00	R/W	6 bits	0x00480C
ILR1	Interrupt priority level bit 1	00	R/W	6 bits	0x00480E
ILR2	Interrupt priority level bit 2	00	R/W	6 bits	0x004810
ILR3	Interrupt priority level bit 3	00	R/W	6 bits	0x004812
ILR4	Interrupt priority level bit 4	00	R/W	6 bits	0x004814
ILR5	Interrupt priority level bit 5	00	R/W	6 bits	0x004816
ILR6	Interrupt priority level bit 6	00	R/W	6 bits	0x004818
ILR7	Interrupt priority level bit 7	00	R/W	6 bits	0x00481A
ILR8	Interrupt priority level bit 8	00	R/W	6 bits	0x00481C
ILR9	Interrupt priority level bit 9	00	R/W	6 bits	0x00481E
ILR10	Interrupt priority level bit 10	00	R/W	6 bits	0x004820
ILR11	Interrupt priority level bit 11	00	R/W	6 bits	0x004822
ILR12	Interrupt priority level bit 12	00	R/W	6 bits	0x004824
ILR13	Interrupt priority level bit 13	00	R/W	6 bits	0x004826
ILR14	Interrupt priority level bit 14	00	R/W	6 bits	0x004828
ILR15	Interrupt priority level bit 15	00	R/W	6 bits	0x00482A

Table 3. Interrupt Input Register (ITR)

Bit	Name	Type	Reset Value
15	IRQ_15	R/W	0
:::	:::	:::	:::
0	IRQ_0	R/W	0

In the event of an edge-sensitive interrupt, ITR stores an incoming interrupt. When the DSP accesses the SIR_FIQ register, the bit corresponding to the interrupt that has requested the DSP action is reset.

The DSP can also clear each bit individually by writing a 0 to the corresponding bits at the ITR address. A 1 bit keeps its previous value. If the individual bit is cleared just before the DSP unmasks the interrupts, the interrupt is not processed.

The DSP reads this register. If an incoming interrupt is edge sensitive, the read value corresponds to the value held in the storage element.

> IRQ (FIQ) output and SIR_IRQ (SIR_FIQ) registers are reset only if the bit of ITR register corresponding to the interrupt that requested DSP action is already cleared or masked.
>
> The time when this ITR bit is reset depends on the sensitivity of the incoming interrupt. In case of an edge-sensitive interrupt, the IT register bit is cleared when reading SIR_IRQ (SIR_FIQ) register. Otherwise, it is reset when the corresponding interrupt line becomes inactive (low).
>
> For a level-sensitive interrupt, the level must be removed before the write to the control register. Otherwise, the interrupt controller is not reset for a new interrupt.

Table 4. Mask Interrupt Register (MIR)

Bit	Name	Description	Type	Reset Value
15	IRQ_15_MSK	Disable IRQ_15 interrupt	R/W	1
:::	:::	:::	:::	:::
0	IRQ_0_MSK	Disable IRQ_0 interrupt	R/W	1

Each incoming interrupt can be masked individually by this register by setting the corresponding bit to 1.

The mask interrupt register (MIR) operates after interrupt input register (ITR); this means that occurrences of incoming interrupts are always stored in the interrupt input register (ITR).

Table 5. IRQ Binary-Coded Source Register (SIR_IRQ)

Bit	Name	Type	Reset Value
3–0	IRQ_NUM	R	0

This register saves software processing time by recognizing the interrupt number as being either an IRQ or FIQ request. Reading this register clears the corresponding bit in the interrupt input register (ITR) if the interrupt is set as edge-sensitive. This register will not normally be used since all level 2 DSP interrupts must be configured as FIQ to generate the DSP interrupts because the IRQ is not connected.

Table 6. FIQ Binary-Coded Source Register (SIR_FIQ)

Bit	Name	Type	Reset Value
3–0	FIQ_NUM	R	0

In order to save software processing time, this register indicates the interrupt number that has an IRQ or FIQ request. Reading this register clears the corresponding bit in the interrupt input register (ITR) if the interrupt is set as edge-sensitive.

Table 7. Interrupt Control Register (CONTROL_REG)

Bit	Name	Description	Type	Reset Value
1	NEW_FIQ_AGR	New FIQ agreement Writing a 1 resets FIQ output and clears the source FIQ register. Enables a new FIQ generation, reset by internal logic. Corresponding bit of ITR must be cleared first.	R/W	0
0	NEW_IRQ_AGR	New IRQ agreement Writing a 1 resets IRQ output and clears the source IRQ register. Enables a new IRQ generation, reset by internal logic. Corresponding bit of ITR must be cleared first. Note: All level 2 DSP interrupts must be configured as FIQ to generate DSP interrupts because the IRQ is not connected.	R/W	0

The software interrupt set register is a 16-bit, read/write register. Writing a 1 to any bit generates an interrupt to the DSP if the corresponding ILR register is set as edge-triggered; otherwise, no interrupt is generated. A 0 is always returned from a read to this register. External interrupts are ORed with the software interrupts before they are sent to the mask interrupt register for interrupt masking.

Table 8. Interrupt Level Registers (ILR0...ILR15)

DSP Word Offset Address (hex)	Name	Corresponding Interrupt
0x0C	ILR_IRQ_0	IRQ_0
0x0E	ILR_IRQ_1	IRQ_1
:::	:::	:::
0x0C + (N−1)*2	ILR_IRQ_N−1	IRQ_N−1
:::	:::	:::
0x2A	ILR_IRQ_15	IRQ_15

Table 9. Interrupt Level Registers (ILR0...ILR15)

Bit	Name	Value	Description	Type	Reset Value
5−2	PRIORITY		Define the priority level when the corresponding interrupt is routed to IRQ or FIQ. 0 is the highest priority level. 15 is the lowest priority level.	R/W	0
1	SENS_EDGE	0	The corresponding interrupt is falling-edge-sensitive.	R/W	0
		1	The corresponding interrupt is low-level-sensitive.		
0	FIQ	0	0: The corresponding interrupt is routed to IRQ.	R/W	0
		1	The corresponding interrupt is routed to FIQ. Note: Since IRQ is not connected, only the FIQ setting is useful. This bit must be set to 1 for the corresponding level 2 interrupt to cause a DSP interrupt.		

Note:

Assuming that all interrupts have the same priority level and they are active at the same at the same moment, the order of servicing is as follows: IRQ_15, IRQ_N−1, IRQ_N−2, …, IRQ_0.

2.2.3 Level 2 Interrupt Mapping

Table 10 shows the DSP level 2 interrupt mapping.

Table 10. DSP Level 2 Interrupt Mapping

Incoming Interrupts	Required Sensitivity Setup	Level 2 Interrupt
McBSP3 TX	Edge	IRQ_00
McBSP3 RX	Edge	IRQ_01
McBSP1 TX	Edge	IRQ_02
McBSP1 RX	Edge	IRQ_03
UART2	Level	IRQ_04
UART1	Level	IRQ_05
MCSI1 TX	Level	IRQ_06
MCSI1 RX	Level	IRQ_07
MSCI2 TX	Level	IRQ_08
MCSI2 RX	Level	IRQ_09
MCSI1 frame error	Level	IRQ_10
MCSI2 frame error	Level	IRQ_11
Reserved		IRQ_12
Reserved		IRQ_13
Reserved		IRQ_14
Reserved		IRQ_15

Level-sensitive interrupts are level-active; the interrupt line must remain asserted until it has been acknowledged.

Edge-triggered interrupts are edge-triggered; just an edge is required for generating the interrupt. The interrupt to the DSP is cleared upon reading of the interrupt registers or writing a 0 to the interrupt mask registers in the interrupt handler.

3 DSP Interrupt Interface

The DSP interrupt interface (DSP_INT_IF) augments the capability of the DSP interrupt processing by providing user-definable edge-triggered and level-sensitive implementations for each of the interrupt lines. This is necessary to allow edge-triggered interrupts, since the DSP level 1 interrupts must be active for greater than two DSP_CLK cycles to be recognized as being active. The DSP_INT_IF module is clocked by the DSP_INTH_CK clock, which which is fixed at half the CK_GEN2 frequency (see SPRU678, *OMAP5910 Clock Generation and System Reset Management Reference Guide*).

3.1 Functional Description

The implementation of each of the interrupt channels to the DSP interrupt handler is shown in Figure 5.

Each interrupt channel processes the incoming interrupt as both an edge-triggered interrupt and a level-sensitive interrupt. The decision regarding which process to use is made by the interrupt (N) edge-triggered enable input, which is bit 2^N of the edge-enable control register. If this bit is 1, the edge-triggered process path is chosen. If 0, the level-sensitive process path is chosen.

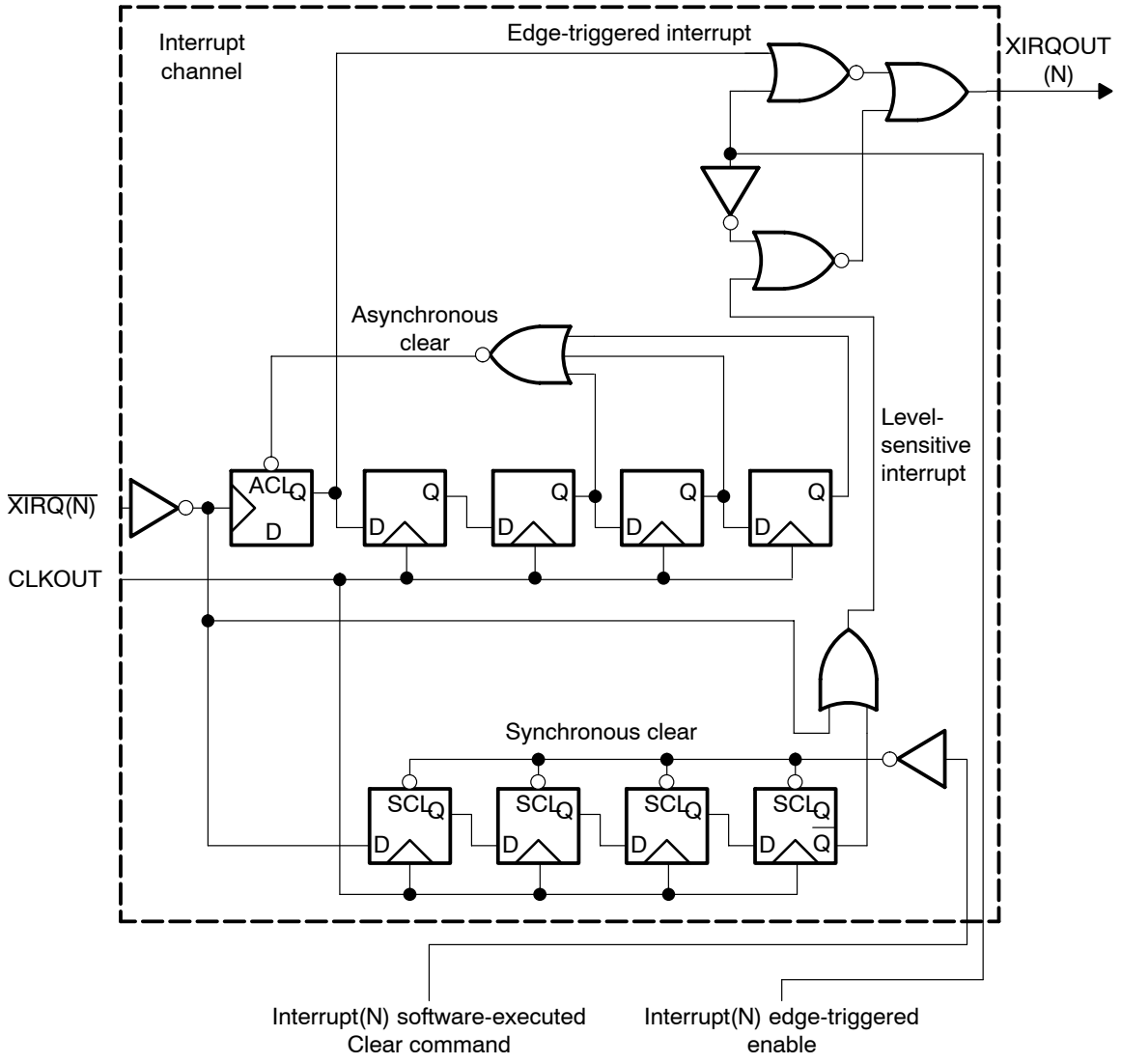
3.2 Edge-Triggered Interrupts

The edge-triggered interrupt process consists of an edge-registration flip-flop and a chain of four positive-edge triggered timing flip-flops. A negative transition (falling edge) on the incoming nXIRQ(N) interrupt line sets the edge-registration flip-flop to 1, and the output of this flip-flop is the edge-triggered interrupt. In addition to activating the output interrupt line nIRQ(N), this output also propagates through the four timing flip-flops. When the 1 output of the edge-registration flip-flop has propagated to the fourth flip-flop, an asynchronous reset is generated, clearing the edge-registration flip-flop and deactivating nIRQ(N).

nIRQ(N) then lasts between three and four DSP_INTH_CK clock periods, depending on when the asynchronous falling edge of nXIRQ(N) occurs with respect to the rising edge of DSP_INTH_CK clock. In OMAP, the frequency of DSP_INTH_CK clock is set to half the DSP_CLK frequency. The DSP requires the nIRQ(N) to transition from high to low and to be low for at least two DSP_CLK cycles, so that nIRQ(N) can be recognized. Also the DSP requires that between two back-to-back interrupts on nIRQ(N), nIRQ(N) be high for at least one DSP_CLK cycle. nIRQ(N) is generated by the rising edge of DSP_INTH_CK clock and lasts for a minimum of three DSP_INTH_CK clock periods (which is actually six DSP_CLK cycles). The requirements on nIRQ(N) are clearly met.

When the edge-registration flip-flop is cleared by the asynchronous reset, two DSP_INTH_CK clock periods must expire before another negative edge transition can be registered. Thus successive negative transitions must be a minimum of six DSP_INTH_CK clock periods apart in time to be ensured of being recognized as two separate incidents. This minimal time does not take into account the processing time of the interrupts once recognized by the DSP processor, and this time must be taken into account to derive the minimum time between interrupts from a system perspective.

Figure 5. Interrupt Channel Implementation



3.3 Level-Sensitive Interrupts

The level-sensitive interrupt process is, in many ways, identical to the edge-triggered interrupt process. This process also uses a chain of four positive-edge triggered timing flip-flops, but this chain is driven by the inverted representation of the incoming interrupt nXIRQ(N). A negative transition (falling edge) on the incoming nXIRQ(N) line activates the output interrupt nIRQ(N) and must be held low for one DSP_INTH_CK cycle (which is equivalent to two DSP_CLK cycles) to be recognized by the DSP. Even when nXIRQ(N) remains at 0 for a time period exceeding three to four DSP_INTH_CK periods (depending on when the asynchronous falling edge of nXIRQ(N) occurs with respect to the rising edge of DSP_INTH_CK), the interrupt nIRQ(N) remains activated until the nXIRQ(N) is deactivated. However, the DSP recognizes this as only one interrupt, because there was only one falling edge of nIRQ(N).

3.4 Internal Registers

DSP word start address: 0x003800

Bit width: 16 bits

DSP word address of a register = start word address + offset address

The DSP_INT_IF has two control registers (one 16-bit and one 7-bit) and two clear command registers (one 16-bit and one 7-bit). The control registers are used exclusively for assigning edge-triggered/level-sensitive status to each of the 23 interrupt channels. The clear command registers are not actual physical registers, but rather a block of decoding logic that issues clear commands to the level-sensitive logic in each interrupt channel upon detecting a TIPB write transaction to an address that falls within the required address range.

The bit-alignment of interrupt channel assignments within the control register, the definition of the assignments, the default values at power turn-on, the address used to write to the control register, and the address used to read the content of the register are all presented in Table 11 through Table 14.

Table 11. Edge-Triggered/Level-Sensitive Control Register Low

Bit	Name	Value	Description	Type	Reset Value
15–0	CHx Trig/Level		This bit defines whether channel CHx is edge- or level-sensitive where CHx corresponds to interrupt channels nXIRQ[15:0]. Channels nXIRQ[15:0] correspond to the DSP level 1 interrupts IRQ17:2, respectively.	R/W	0
		0	CHx is level-sensitive.		
		1	CHx is edge-sensitive.		

Table 12. Edge-Triggered/Level-Sensitive Control Register High

Bit	Name	Value	Description	Type	Reset Value
15–8	Reserved				0
7	Host Interrupt Trig/Level		This bit defines whether the host interrupt is edge or level-sensitive.	R/W	
		0	NHOSTINT is level-sensitive.		
		1	NHOSTINT is edge-sensitive.		
6	NMI Trig/Level		This bit defines whether the nonmaskable interrupt is edge or level-sensitive. The NMI channel corresponds to the DSP NMI interrupt.	R/W	0
		0	NMI is level-sensitive.		
		1	NMI is edge-sensitive.		
5–0	CHx Trig/Level		This bit defines whether channel CHx is edge or level-sensitive, where CHx corresponds to interrupt channels nXIRQ[21:16]. Channels nXIRQ[21:16] correspond to the DSP level 1 interrupts IRQ23:18, respectively.	R/W	0
		0	CHx is level-sensitive.		
		1	CHx is edge-sensitive.		

3.4.1 Level-Sensitive Clear Commands (Write Only)

A write transaction issues a clear to those interrupt channels whose assigned bit in the 16-bit word being written is 1. Commands to clear interrupt channels are necessary for those channels assigned as level-sensitive interrupt channels. Figure 6 illustrates the alignment of the channel clear assignments within the 16-bit word written to the XIO interrupt processor and gives the permissible range of addresses over which the write can take place.

A write to the level-sensitive clear low register (RST_LVL_LO), whose offset address is 02, clears interrupts corresponding to nXIRQ[15:0].

Table 13. Level-Sensitive Clear Low Register (RST_LVL_LO)

Bit	Name	Value	Description	Type	Reset Value
15–0	Reset_CHx		Reset CHx if a 1 is written into RST_LVL_LO[x] and CHx is configured as level-sensitive interrupt, where CHx corresponds to interrupt channels nXIRQ[15:0]	0	0
		0	Do not reset CHx.		
		1	Reset interrupt channel CHx if level is configured as level sensitive.		

A write to the level-sensitive clear high register (RST_LVL_HI), whose offset address is 03, clears interrupts from interrupt channels [20:16], NMI, and HOSTINT.

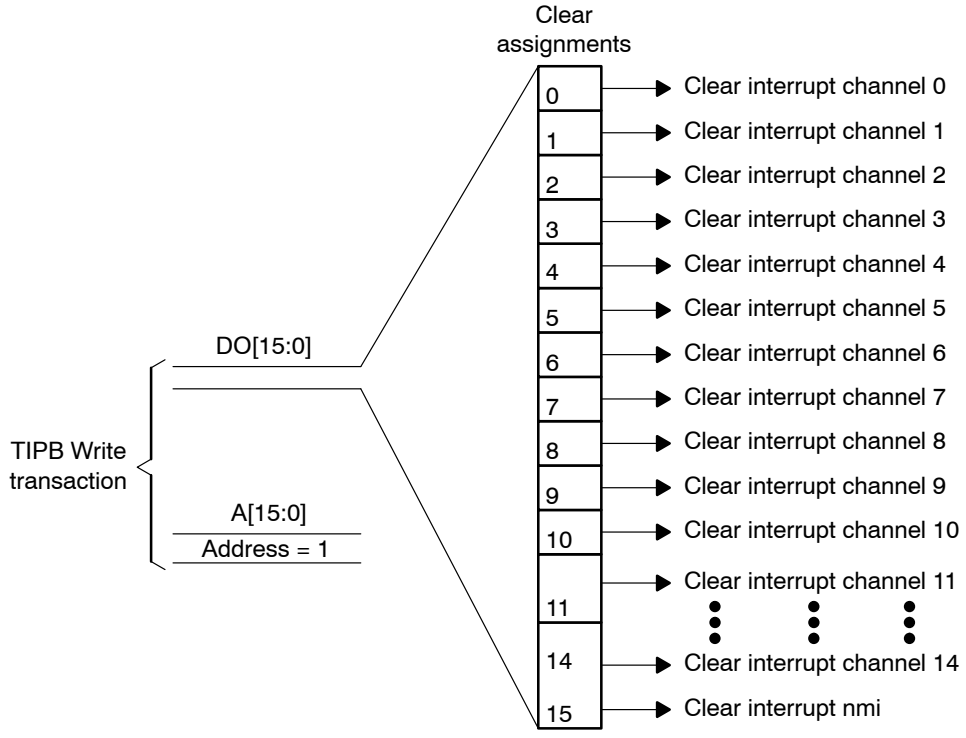
Table 14. Level-Sensitive Clear High Register (RST_LVL_HI)

Bit	Name	Value	Description	Type	Reset Value
7	Reset_NHOSTINT		Reset NHOSTINT channel if a 1 is written into this bit and NHOSTINT is configured as level-sensitive interrupt.		
		0	Do not reset NHOSTINT.		
		1	Reset NHOSTINT interrupt channel, if configured as level-sensitive interrupt.		
6	Reset_NMI		Reset NMI channel if a 1 is written into this bit and NMI is configured as level-sensitive interrupt.		
		0	Do not reset CHx.		
		1	Reset NMI interrupt channel if configured as level-sensitive interrupt.		

Table 14. Level-Sensitive Clear High Register (RST_LVL_HI) (Continued)

Bit	Name	Value	Description	Type	Reset Value
5–0	Reset_CHx		Reset CHx if a 1 is written into RST_LVL_LO[x] and CHx is configured as level-sensitive interrupt, where CHx corresponds to interrupt channels nXIRQ[20:16].		0
		0	Do not reset CHx.		
		1	Reset interrupt channel CHx if level is configured as level-sensitive.		

Figure 6. Level-Sensitive Interrupt Clear Commands



Index