AM17x/AM18x ARM Microprocessor Universal Parallel Port (uPP)

User's Guide

Literature Number: SPRUFW4
March 2010

List of Figures

List of Tables

Read This First

About This Manual

Describes the universal parallel port (uPP) peripheral.

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.
- 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 below. A legend explains the notation used for the properties.
  - Reserved bits in a register figure designate a bit that is used for future device expansion.

Related Documentation From Texas Instruments

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.

The current documentation that describes the DSP, related peripherals, and other technical collateral, is available in the C6000 DSP product folder at: www.ti.com/c6000.

SPRUGU4 — ***AM1806 ARM Microprocessor System Reference Guide.*** Describes the ARM subsystem, system memory, memory protection unit (MPU), device clocking, phase-locked loop controller (PLLC), power and sleep controller (PSC), power management, ARM interrupt controller (AINTC), and system configuration module.

SPRUGM9 — ***AM1808 ARM Microprocessor System Reference Guide.*** Describes the ARM subsystem, system memory, memory protection unit (MPU), device clocking, phase-locked loop controller (PLLC), power and sleep controller (PSC), power management, ARM interrupt controller (AINTC), and system configuration module.

SPRUFU0 — ***AM17x/AM18x ARM Microprocessor Peripherals Overview Reference Guide.*** Provides an overview and briefly describes the peripherals available on the AM17x/AM18x ARM Microprocessors.

Universal Parallel Port (uPP)

1 Introduction

1.1 Purpose of the Peripheral

The universal parallel port (uPP) peripheral is a multichannel, high-speed parallel interface with dedicated data lines and minimal control signals. It is designed to interface cleanly with high-speed analog-to-digital converters (ADCs) or digital-to-analog converters (DACs) with up to 16-bit data width (per channel). It may also be interconnected with field-programmable gate arrays (FPGAs) or other uPP devices to achieve high-speed digital data transfer. It can operate in receive mode, transmit mode, or duplex mode, in which its individual channels operate in opposite directions.

The uPP peripheral includes an internal DMA controller to maximize throughput and minimize CPU overhead during high-speed data transmission. All uPP transactions use the internal DMA to feed data to or retrieve data from the I/O channels. The DMA controller includes two DMA channels, which typically service separate I/O channels. The uPP peripheral also supports data interleave mode, in which all DMA resources service a single I/O channel. In this mode, only one I/O channel may be used.

1.2 Features

For more information on the features and performance supported by the uPP peripheral, see your device-specific data manual.

1.3 Functional Block Diagram

Figure 1 provides a high-level view of the uPP peripheral internal logic. Note that this figure shows one particular configuration: Channel A receives and Channel B transmits. In general, each channel may operate in either direction.

Figure 2, Figure 3, Figure 4, and Figure 5 show simplified data paths through the uPP peripheral for various configurations. Note that these figures are examples and do not represent all possible configurations. More information on these and other modes of operation is given in subsequent sections.

Figure 1. uPP Functional Block Diagram

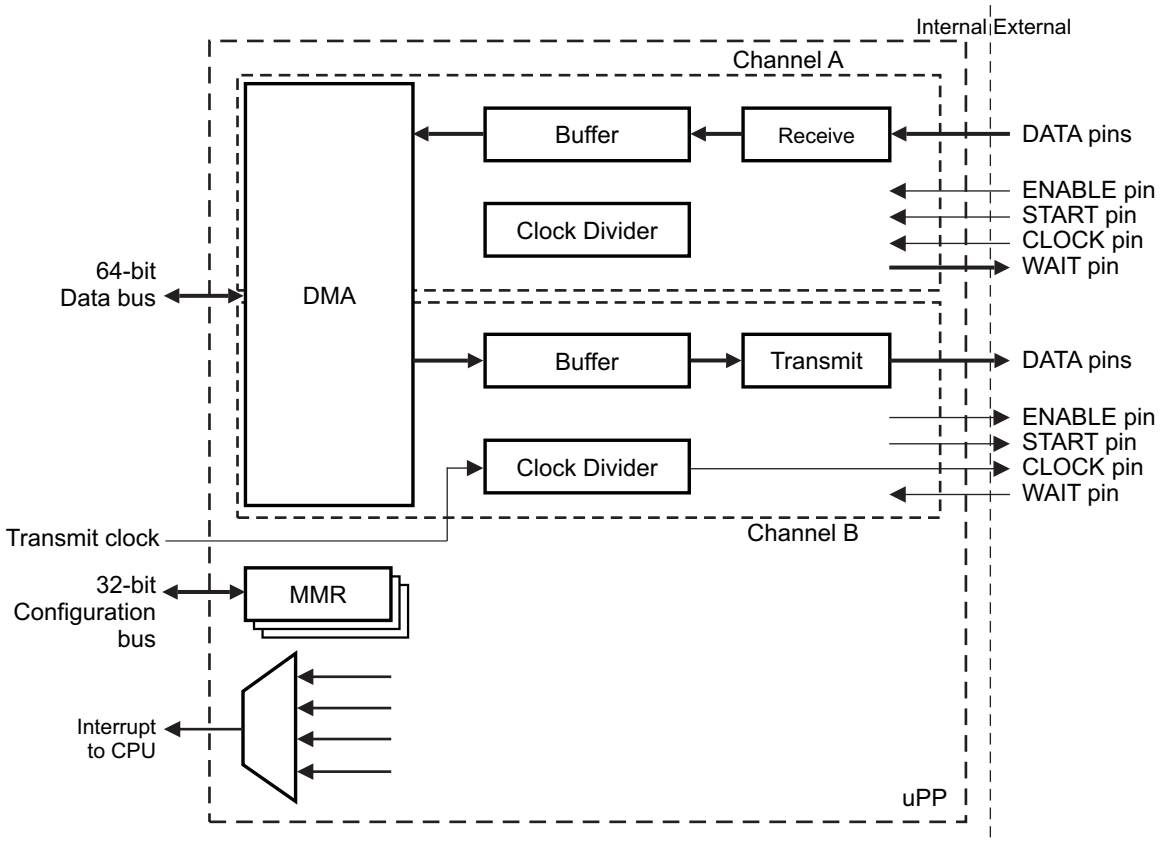


Figure 2. Data Flow for Single-Channel Receive Mode

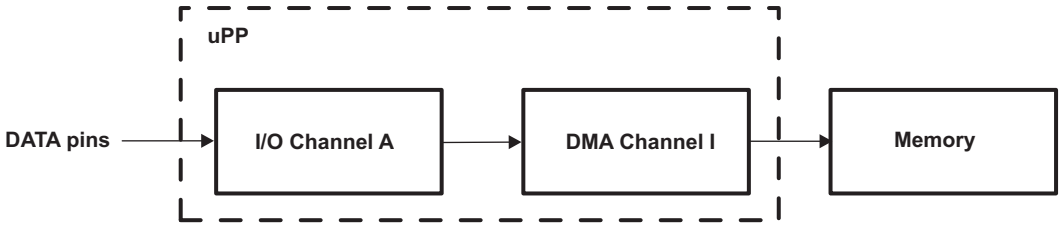


Figure 3. Data Flow for Single-Channel Transmit Mode

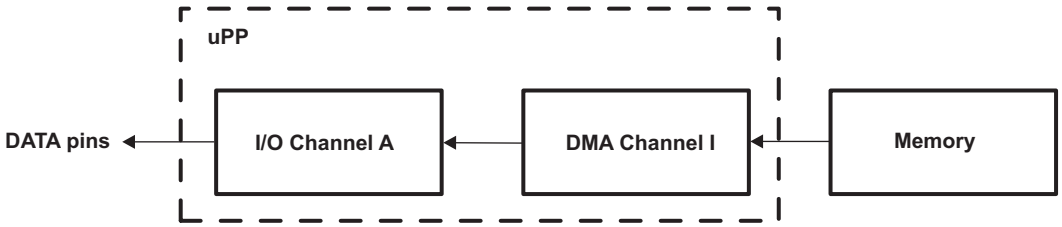


Figure 4. Data Flow for Digital Loopback (DLB) Mode (Duplex Mode 0)

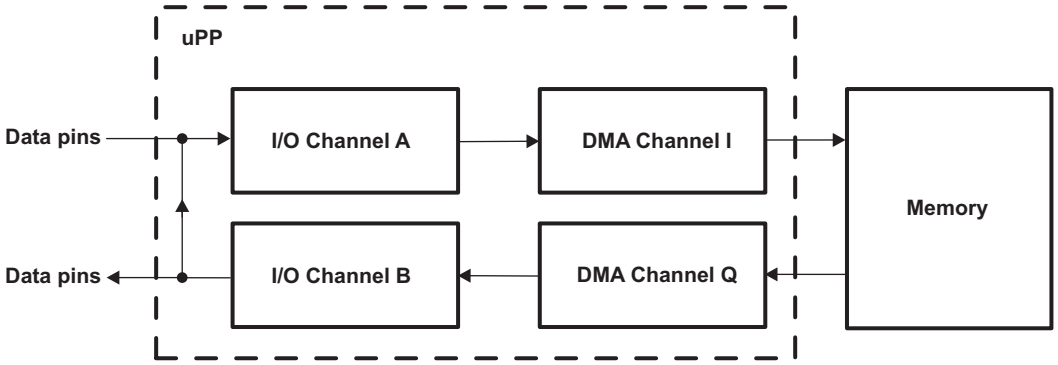
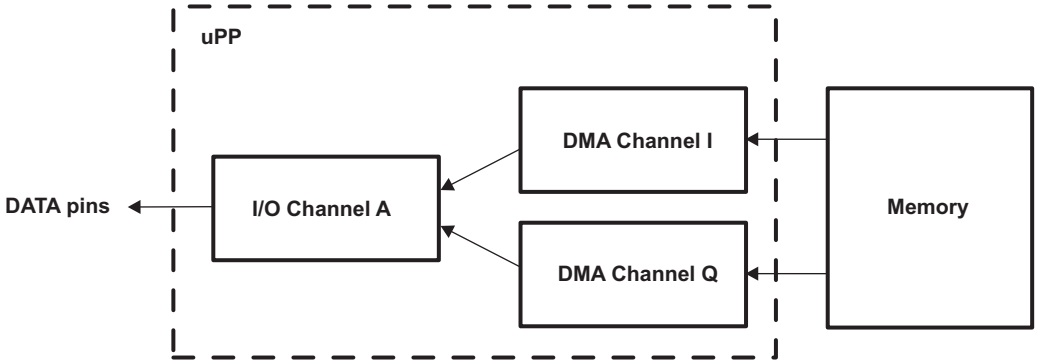


Figure 5. Data Flow for Single-Channel Transmit with Data Interleave



2 Architecture

2.1 Clock Generation and Control

The uPP peripheral uses two separate clocks: a module clock that controls its internal logic, and a transmit clock that runs either interface channel in transmit mode. The source for each of these clocks may be configurable; see your device-specific *System Reference Guide* for more information. Neither the module clock nor the transmit clock can be faster than one-half the device CPU clock speed.

Each channel's CLOCK pin, or I/O clock, is obtained independently based on its operating direction.

2.1.1 Transmit Mode (Single Data Rate)

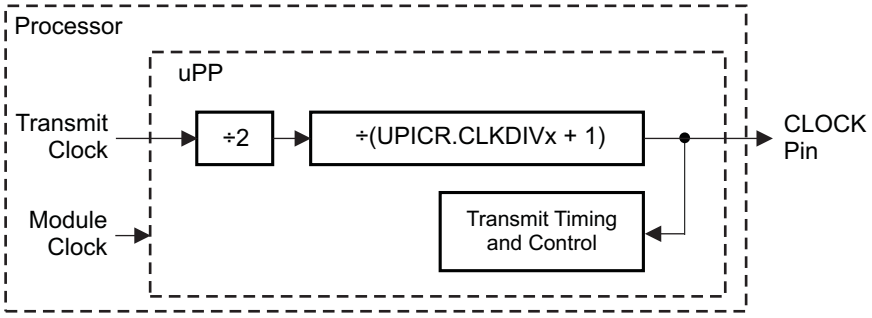
The channel drives a clock signal on its CLOCK pin. The uPP transmit clock is divided by a fixed value of 2, then divided again by a user-specified value between 1 and 16 (UPICR.CLKDIV*n* + 1). The resulting signal then drives the CLOCK pin. The following formula determines the final I/O clock speed:

I/O Clock = Transmit Clock/(2 × (UPICR.CLKDIVn + 1))

The fixed divisor restricts the maximum speed of the I/O clock to one-fourth the device CPU clock speed.

Figure 6 shows the clock generation system for a channel configured in transmit mode.

Figure 6. Clock Generation for a Channel Configured in Transmit Mode

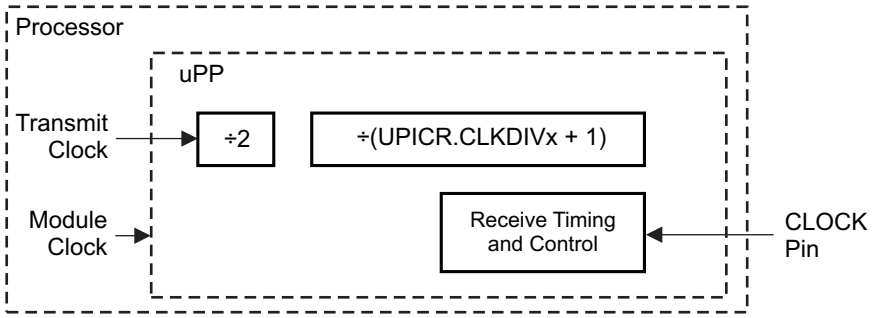


2.1.2 Receive Mode (Single Data Rate)

The channel requires an external clock to drive its CLOCK pin. The incoming clock is not divided, and its maximum allowed speed is one fourth (¼) the device CPU clock speed.

Figure 7 shows the clock generation system for a channel configured in receive mode.

Figure 7. Clock Generation for a Channel Configured in Receive Mode



2.1.3 Double Data Rate

The uPP peripheral supports two I/O clocking schemes. The first, single data rate (SDR), clocks data from the DATA pins on either the rising edge or the falling edge (depending on UPICR.CLKINV*n*) of the I/O clock.

The second clocking scheme is double data rate (DDR). In this mode, data is clocked on both the rising and falling edges of the I/O clock. However, DDR mode imposes a lower I/O clock speed limit of one eighth (1/8) the device CPU clock for both transmit and receive modes. The operating speed for transmit mode with various divisors in each data rate are summarized in Table 1 (in this table, a data word is defined as the data represented on the DATA pins; uPP supports data words in the 8-bit to 16-bit range). In receive mode, a channel I/O clock is generated by an external source, but the same speed limit applies.

Table 1. I/O Clock Speeds for Channel in Transmit Mode Given 150 MHz Transmit Clock

		Word Rate (Mw/s)	
UPICR.CLKDIV*n*	I/O Clock (MHz)	Single Data Rate	Double Data Rate
0	75.00	75.00	...
1	37.50	37.50	75.00
2	25.00	25.00	50.00
3	18.75	18.75	37.50
...	...	...	...
15	4.69	4.69	9.38

Additional restrictions may apply, check the device datasheet to see if your particular uPP peripheral has any additional clock requirements.

2.2 Signal Description

Each uPP channel has its own set of control and data signals. Table 2 lists every signal and briefly describes their functions. Section 2.5 explains the uPP protocol.

Table 2. uPP Signal Descriptions

Signal	I/O Channel	Type (Transmit)	Type (Receive)	Description
DATA[15:0]	—	Output	Input	Parallel data bus
XDATA[15:0]	—	Output	Input	Extended parallel data bus
CHA_START	A	Output	Input	Indicates first data word per line of data
CHA_ENABLE	A	Output	Input	Indicates data transmission active
CHA_WAIT	A	Input	Output	Requests transmitter halt temporarily
CHA_CLOCK	A	Output	Input	Source-synchronous clock signal
CHB_START	B	Output	Input	Indicates first data word per line of data
CHB_ENABLE	B	Output	Input	Indicates data transmission active
CHB_WAIT	B	Input	Output	Requests transmitter halt temporarily
CHB_CLOCK	B	Output	Input	Source-synchronous clock signal
UPP_2xTXCLK	—	Input	—	Optional external source for transmit clock(1)

(1) This clock can only be used in transmit mode, and must be twice the speed of your desired I/O clock. See Section 2.1 and your device-specific *System Reference Guide* for more information.

Note that the DATA and XDATA pins are not dedicated to a single I/O channel in the same way as the control signals. For practical reasons, uPP data pin channel assignments are not static. Instead, the data pins used by each I/O channel (A, B) depend on the operating mode of the uPP peripheral. Table 3 summarizes the assignment of the DATA and XDATA pins to each channel for various operating modes, along with the relevant register settings. For more information on these pins, see your device-specific data manual.

Table 3. DATA and XDATA Pin Assignments to Channels A and B According to Operating Mode

uPP Channel Control Register (UPCTL) Bit			Assigned Channel			
CHN	IWA	IWB	DATA[15:8]	DATA[7:0]	XDATA[15:8]	XDATA[7:0]
0	0	x	—	A[7:0]	—	—
0	1	x	A[15:8]	A[7:0]	—	—
1	0	0	B[7:0]	A[7:0]	—	—
1	0	1	B[7:0]	A[7:0]	B[15:8]	—
1	1	0	B[7:0]	A[7:0]	—	A[15:8]
1	1	1	B[7:0]	A[7:0]	B[15:8]	A[15:8]

2.3 Pin Multiplexing

Extensive pin multiplexing is used to accommodate the largest number of peripheral functions in the smallest possible package. Pin multiplexing is controlled using a combination of hardware configuration at device reset and software programmable register settings. To determine how pin multiplexing affects the uPP peripheral, see your device-specific data manual.

2.4 Internal DMA Controller Description

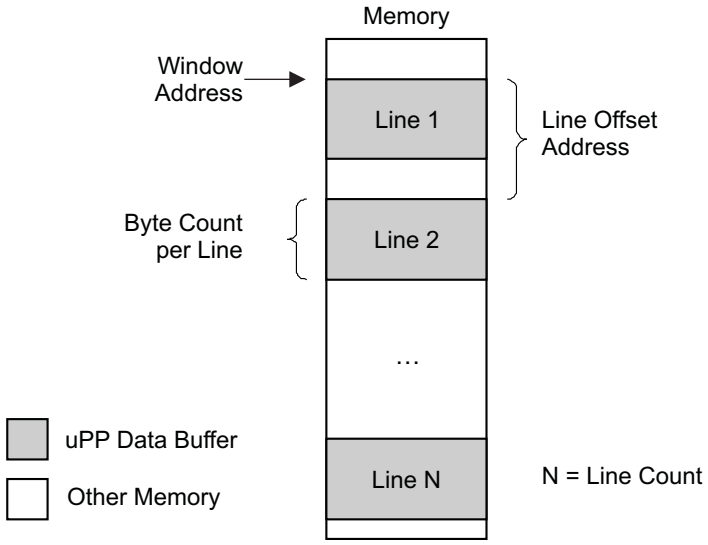
The uPP peripheral includes an internal DMA controller separate from any device-level DMA, such as EDMA. The internal DMA controller consists of two DMA channels, I and Q, which move data to/from the uPP peripheral interface (I/O) channels in all operating modes. This section describes how to program the internal DMA channels.

2.4.1 DMA Programming Concepts

The uPP internal DMA controller uses a simplified programming model similar to 2D transfers performed by the EDMA. (see the *AM18x ARM Microprocessor Enhanced Direct Memory Access (EDMA3) Controller User's Guide* (SPRUFU5) for more information). Each DMA channel may be configured with four parameters: window address, byte count, line count, and line offset address. Figure 8 shows a typical DMA window defined by these parameters.

- **Window Address** (UPxD0.ADDR) – The location in memory of the first byte in the data buffer. When the uPP operates in receive mode, the DMA channel begins writing to this address as it takes incoming data from the uPP I/O channel. When the uPP operates in transmit mode, the DMA channel begins reading from this address and pass the data to the uPP I/O channel. The window address can reside in any available memory space (including EMIF), but it must be aligned to a 64-bit boundary (that is, the 3 LSBs must equal 0). Nonaligned addresses are automatically adjusted to a properly aligned value when written to UPxD0.
- **Byte Count** (UPxD1.BCNT) – The number of bytes per line. The byte count must be an even number.
- **Line Count** (UPxD1.LNCNT) – The number of lines per window. The total number of bytes transferred equals B × L, where B is the byte count per line and L is the line count.
- **Line Offset Address** (UPxD2.LNOFFSET) – The offset address between the first byte in successive lines. The line offset address cannot exceed 65528 (FFF8h) bytes, and must be aligned to a 64-bit boundary in memory (that is, the 3 LSBs must equal 0).

Figure 8. Structure of DMA Window and Lines in Memory



Certain values of the line offset address have special implications on the structure of the data buffer:

- **Line Offset Address = Byte Count** – Data buffer is a contiguous block in memory with size equal to (Line Count) × (Byte Count).
- **Line Offset Address = 0** – Data buffer consists of a single line, with total size equal to Byte Count. If the I/O channel is configured in transmit mode, this line is transmitted (Line Count) consecutive times before the DMA transfer completes. If the I/O channel is configured in receive mode, the buffer is repeatedly written and overwritten by incoming data.

To program a DMA transfer, write the appropriate fields in the DMA channel descriptor registers, UPID*n* for DMA Channel I or UPQD*n* for DMA Channel Q. If the associated I/O channel is initialized and idle, the DMA transfer and I/O transaction begins immediately. Section 2.6 describes a step-by-step process for configuring the I/O and DMA channels and starting a uPP transfer.

Each DMA channel allows a second descriptor to be queued while the previously programmed DMA transfer is still running. The UPxS2.PEND bit reports whether a new set of DMA parameters may be written to the DMA descriptor registers. Each DMA channel can have at most one active transfer and one queued transfer. This allows each I/O channel to perform uninterrupted, consecutive transactions across DMA transfer boundaries.

The internal DMA controller does not support automatically reloading DMA transfer descriptors. Each successive descriptor set must be explicitly written to the UPxD*n* registers by software.

All uPP interrupt events originate in the internal DMA controller. Section 2.8 lists and explains all uPP interrupt events.

The internal DMA controller always writes data in bursts of 64 bytes. However, DMA read operations have configurable burst size, which may be set per channel using the RDSIZEI and RDSIZEQ bits in the uPP threshold configuration register (UPTCR). A DMA channel waits until the specified number of bytes leaves its internal buffer before performing another burst read from memory.

Note that the TXSIZEA and TXSIZEB bits in UPTCR are not DMA parameters; instead, they control transmit thresholds for the uPP interface channels.

2.4.2 Data Interleave Mode

The data interleave mode is a special configuration that maps both DMA channels to a single interface channel. Since there are only two DMA channels in the uPP peripheral, data interleave mode can only be used when the uPP peripheral is operated in single-channel mode. There are two variants on data interleave mode, each with special conditions:

- Single Data Rate (SDR) Interleave – Transmit Only
  - UPCTL.CHN = 0 (single-channel mode)
  - UPCTL.DRA = 0 (single data rate)
  - UPCTL.MODE = 1 (transmit mode)
  - UPCTL.SDRTXIL = 1 (enable SDR transmit interleave)
  - UPCTL.DDRDEMUX = 0 (disable DDR interleave)
- Double Data Rate (DDR) Interleave – Transmit or Receive
  - UPCTL.CHN = 0 (single-channel mode)
  - UPCTL.DRA = 1 (double data rate)
  - UPCTL.MODE = 0 or 1 (receive or transmit mode; not duplex)
  - UPCTL.SDRTXIL = 0 (disable SDR transmit interleave)
  - UPCTL.DDRDEMUX = 1 (enable DDR interleave)

Section 2.1 describes the differences between single data rate (SDR) and double data rate (DDR).

In data interleave mode, only I/O Channel A is used. This single channel is associated with two data buffers, each serviced by its own DMA channel (I and Q). In SDR interleave mode, the START signal is used as a buffer selection line: START = 1 indicates that the current word comes from DMA Channel I; START = 0 indicates that the current word comes from DMA Channel Q. In DDR Interleave mode, the data buffers alternate every word beginning with Channel I: Channel I Word 0, Channel Q Word 0, Channel I Word 1, Channel Q Word 1, etc. Section 2.5 shows signal diagrams for both data interleave modes.

2.4.3 Interface and DMA Channel Mapping

Typically, DMA Channels I and Q are mapped to interface Channels A and B, respectively. Data interleave mode is the exception, since it allocates both DMA channels to service interface Channel A. Table 4 summarizes DMA channel mapping for various modes of operation.

Table 4. Interface and DMA Channel Mapping for Various Operating Modes

	I/O Channel Serviced	
Operating Mode	**DMA I**	**DMA Q**
1-Channel Receive	A	—
1-Channel Transmit	A	—
2-Channel Receive	A	B
2-Channel Transmit	A	B
2-Channel Duplex	A	B
1-Channel Transmit (Interleave)	A	A
1-Channel Receive (Interleave)	A	A

2.5 Protocol Description

The uPP peripheral consists of two independent channels, each possessing its own data lines and control signals. A channel may be configured to run in transmit or receive mode and to use either 8 or 16 data lines (8-bit or 16-bit mode) using the uPP channel control register (UPCTL). A channel may also be configured to ignore certain control signals using the uPP interface configuration register (UPICR). Each uPP defaults to 8-bit mode and uses all four control signals, unless otherwise configured. Table 5 summarizes the signals that are required for basic operation in receive and transmit modes. The following subsections describe the role of each signal.

Table 5. Required Signals for Various Modes

	Signal Required?	
Signal Name	**Transmit Mode**	**Receive Mode**
DATA[7:0]	√	√
DATA[15:8]		
START	√	
ENABLE	√	
WAIT		√
CLOCK	√	√

2.5.1 DATA[7:0] Signals

In 8-bit mode, DATA[7:0] comprise the channel's entire data bus. In 16-bit mode, DATA[7:0] comprise the least-significant bits of the 16-bit word. The channel's data width is selected using the IWx bit in UPCTL.

In transmit mode, these pins are outputs that transmit data supplied by the channel's associated DMA channel. While the channel is idle, their behavior depends on the TRISx bit in UPICR. These pins can be configured to drive an idle value (TRISx = 0, VALx field in the uPP interface idle value register (UPIVR)) or be in a high-impedance state while idle (TRISx = 1).

In receive mode, these pins are inputs that provide data to the channel's associated DMA channel.

Note that the DATA signals map differently to the DATA and XDATA pins for various uPP configurations, see Section 2.2 for more information.

2.5.2 DATA[15:8] Signals

In 8-bit mode, DATA[15:8] are not used. In 16-bit mode, DATA[15:8] comprise the most-significant bits of the 16-bit word. The channel's data width is selected using the IWx bit in UPCTL. A channel may be further configured to use only part of its DATA[15:8] pins, which allows any total data width from 8 to 16 bits. Section 2.5.8 describes data format and packing in the 9-bit to 15-bit configurations.

While in use, the direction and behavior of DATA[15:8] in transmit and receive modes are the same as the direction and behavior of DATA[7:0].

Note that the DATA signals map differently to the DATA and XDATA pins for various uPP configurations, see Section 2.2 for more information.

2.5.3 START Signal

The uPP transmitter asserts the START signal when it transfers the first word of a data line. A line is defined in terms of the channel's associated DMA channel; for more on DMA programming concepts, see Section 2.4. The START signal is active-high by default, but its polarity is controlled by the STARTPOLx bit in UPICR.

In transmit mode, START is an output signal and is always driven; in receive mode, START is an input signal and may be disabled using the STARTx bit in UPICR.

When the channel is configured in transmit mode with data interleave enabled (SDRTXIL = 1 in UPCTL), the START signal function changes completely. The START signal now asserts on every data word that is provided by DMA Channel I. See Section 2.5.7 for this alternative behavior.

2.5.4 ENABLE Signal

The uPP transmitter asserts the ENABLE signal when it transfers a valid data word. The ENABLE signal is active-high by default, but its polarity is controlled by the ENAPOLx bit in UPICR.

In transmit mode, ENABLE is an output signal and is always driven; in receive mode, ENABLE is an input signal and may be disabled using the ENAx bit in UPICR.

2.5.5 WAIT Signal

The WAIT signal allows the receiver to request a temporary halt in data transmission. When the receiver asserts WAIT, the transmitter responds by stopping transmission (starting with the next word) until WAIT is released. The receiver ignores all incoming data until WAIT is released. Once WAIT is released, the transmitter can resume transmission on the next word. Figure 10 shows WAIT signal timing. The WAIT signal is active-high by default, but its polarity is controlled by the WAITPOLx bit in UPICR.

In transmit mode, WAIT is an input signal and may be disabled using the WAITx bit in UPICR; in receive mode, WAIT is an output signal.

2.5.6 CLOCK Signal

The uPP transmitter drives the CLOCK signal to align all other uPP signals. By default, other signals align on the rising edge of CLOCK, but its polarity is controlled by the CLKINVx bit in UPICR. The active edge(s) of CLOCK should always slightly precede transitions of other uPP signals.

In transmit mode, CLOCK is an output signal; in receive mode, CLOCK is an input signal. See Section 2.1 for more information on clock generation and allowed frequencies.

2.5.7 Signal Timing Diagrams

In the following diagrams, signals are marked (I) to indicate that they are inputs to the uPP peripheral and (o) to indicate that they are outputs from the uPP peripheral. Data words from a single DMA channel are designated Dx, while data words that must come from a specific DMA channel are designated Ix or Qx to indicate DMA Channel I or Q, respectively. For more information on DMA channels and data interleave mode, see Section 2.4.

All signal diagrams are drawn with signal polarities in their default states. All signals except DATA are independently configurable in the uPP interface configuration register (UPICR).

Figure 9. Signal Timing for uPP Channel in Receive Mode with Single Data Rate

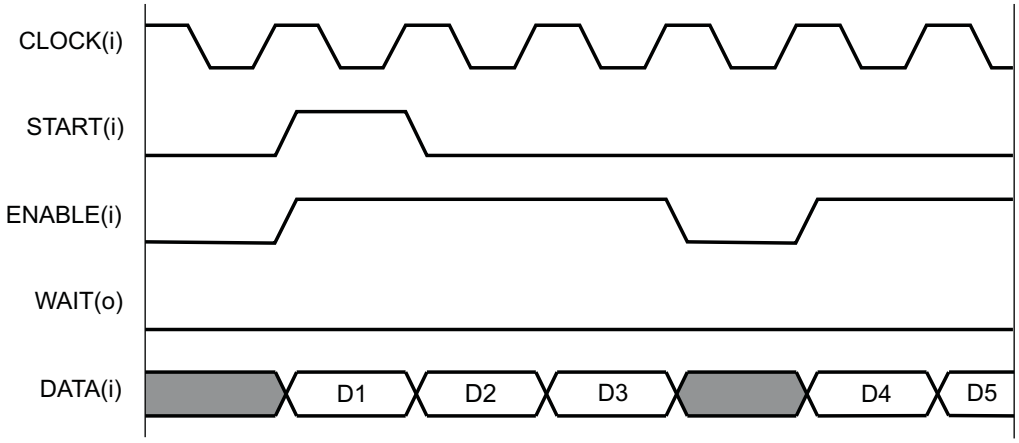


Figure 10. Signal Timing for uPP Channel in Transmit Mode with Single Data Rate

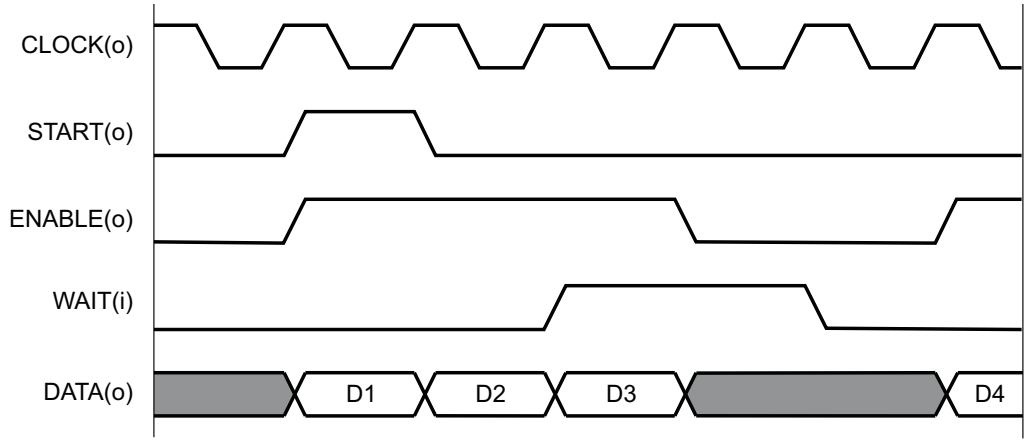


Figure 11. Signal Timing for uPP Channel in Receive Mode with Double Data Rate

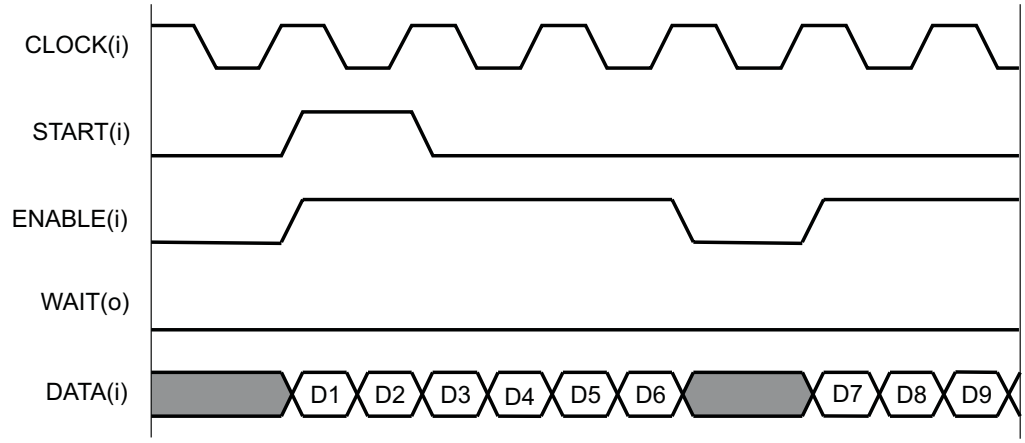


Figure 12. Signal Timing for uPP Channel in Transmit Mode with Double Data Rate

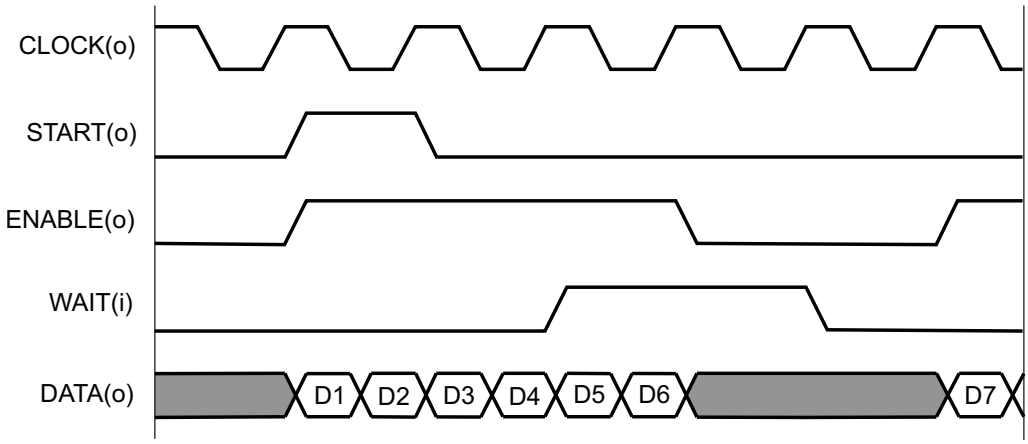


Figure 13. Signal Timing for uPP Channel in Receive Mode with Double Data Rate and Data Interleave Enabled (via UPCTL.DDRDEMUX)

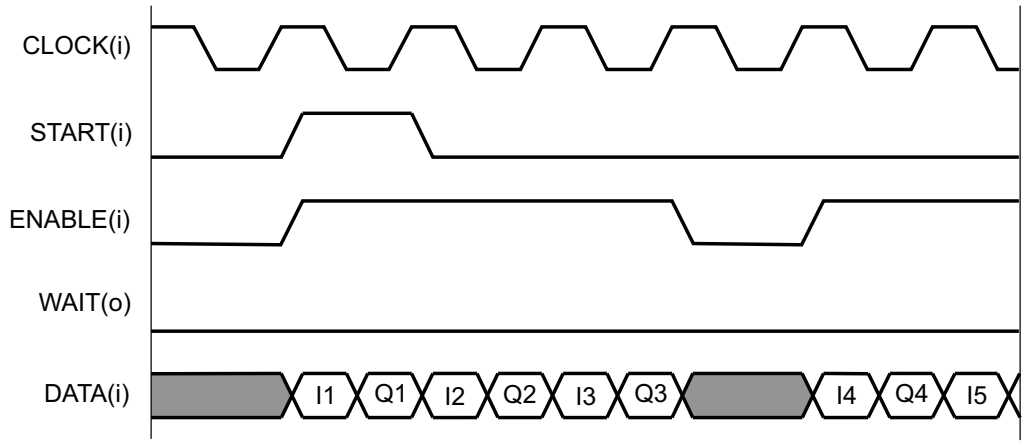


Figure 14. Signal Timing for uPP Channel in Transmit Mode with Double Data Rate and Data Interleave Enabled (via UPCTL.DDRDEMUX)

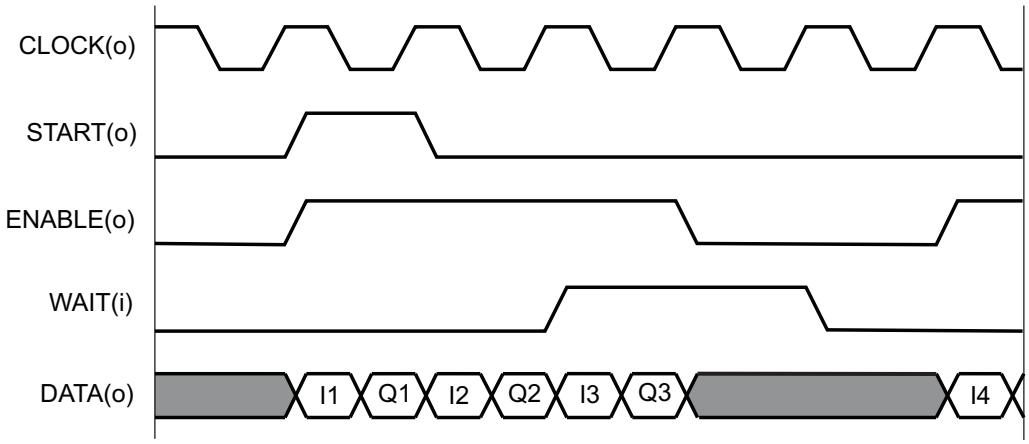
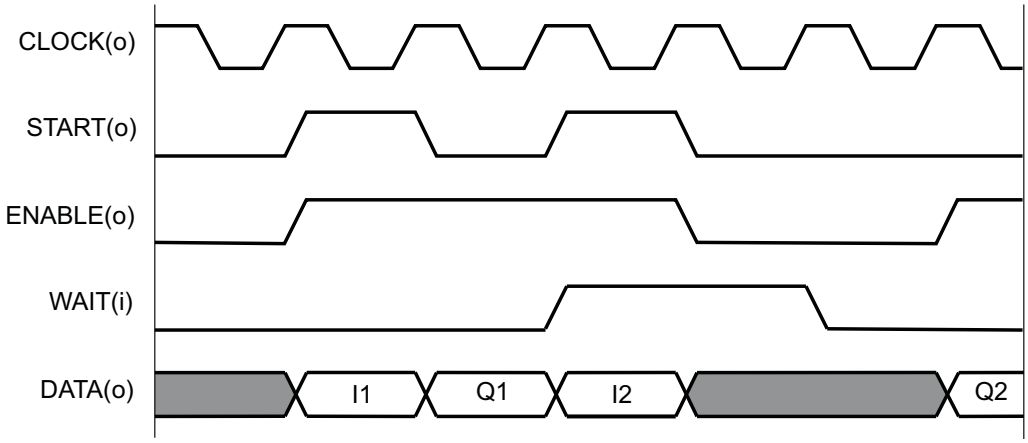


Figure 15. Signal Timing for uPP Channel in Transmit Mode with Single Data Rate and Data Interleave Enabled (via UPCTL.SDRTXIL)



NOTE: START asserts on every data word from DMA Channel I.

2.5.8 Data Format

The uPP peripheral supports any data word width between 8 to 16 bits using the IWx and DPWx bits in the uPP channel control register (UPCTL). For 8-bit operation, uPP reads/writes 8-bit words in memory; for 16-bit operation, uPP reads/writes 16-bit words in memory.

For N-bit operation (8 < N < 16), the uPP peripheral reads/writes 16-bit words in memory. The “extra” bits are filled by the uPP according to a data packing scheme, selected by the DFWx bit in UPCTL. There are three selectable data packing modes:

- Right-Justify, Zero Extend – Data occupies N LSBs. The (16 – N) MSBs are cleared to 0.
- Right-Justify, Sign Extend – Data occupies N LSBs. The (16 – N) MSBs are the same value as the (N – 1) bit.
- Left-Justify, Zero Fill – Data occupies N MSBs. The (16 – N) LSBs are cleared to 0.

Table 6 lists some example data for N = 12 (that is, 12-bit operation). In transmit mode, the packed version of each data word from memory is transmitted using all 16 data pins allotted to the uPP channel. In receive mode, the packed version of each incoming data word (using only N data pins) is stored in memory.

Table 6. Data Packing Examples for 12-Bit Data Words

12-Bit Data Word	Right-Justify, Zero Extend	Right-Justify, Sign Extend	Left-Justify, Zero Fill
123h	0123h	0123h	1230h
ABCh	0ABCh	FABCh	ABC0h
000h	0000h	0000h	0000h
800h	0800h	F800h	8000h
FFFh	0FFFh	FFFFh	FFF0h

2.6 Initialization and Operation

2.6.1 Step-by-Step Procedure

NOTE: When initializing the uPP peripheral, the uPP interface configuration register (UPICR) must be programmed using a single, 32-bit write. Writing UPICR fields one-by-one can lead to unexpected results.

This section provides step-by-step instructions for initializing and running the uPP peripheral in various modes. These instructions are given assuming that the device has just come out of a power-on reset (POR) state.

1. Apply the appropriate pin multiplexing settings. See your device-specific *System Reference Guide*, your device-specific data manual, and/or pin multiplexing utility for more information.
2. Enable the power and clocks to the uPP peripheral. See your device-specific *System Reference Guide* for more information.
3. Set the SWRST bit in the uPP peripheral control register (UPPCR) to 1 to place uPP in software reset.
4. Wait at least 200 device clock cycles, then clear the SWRST bit to 0 to bring the module out of reset.
5. Program the uPP configuration registers: UPCTL, UPICR, UPIVR, UPTCR, and UPDLB. The basic function of each register is summarized here; for more information, see Section 3.
   (a) UPCTL – Transmit/receive selection (see Table 7), data width, data format, data rate, data interleave enable
   (b) UPICR – Signal enable, signal inversion, clock divisor (transmit only)
   (c) UPIVR – Idle value (transmit only)
   (d) UPTCR – I/O transmit threshold (transmit only), DMA read burst size
   (e) UPDLB – Digital loopback (see Table 7)
6. Program the uPP interrupt enable set register (UPIES) to interrupt generation for the desired events. Register an interrupt service routine (ISR) if desired; otherwise, polling is required.
7. Set the EN bit in the uPP peripheral control register (UPPCR) to 1 to turn on the uPP peripheral.
8. Allocate and/or initialize data buffers for use with uPP.
9. Program the DMA channels with their first transfers using the uPP DMA channel descriptor registers: UPID0-2 and/or UPQD0-2.
10. Watch for interrupt events. Reprogram the DMA as necessary (checking that the PEND bit in the uPP DMA channel status register (UPxS2) is 0).
    (a) If polling, check UPIES. Reading a bit as 1 indicates the corresponding event has occurred. Write the corresponding bit with 1 to clear.
    (b) If using ISR, check UPIES inside your ISR. Structure your ISR according to the pseudo-code in Section 2.6.4.

2.6.2 Sample Configuration Settings

The uPP peripheral is flexible, with several orthogonal configuration choices. Table 7 summarizes selecting the fundamental operating mode of the module.

NOTE: Digital loopback (DLB) mode is a configuration that the uPP peripheral internally routes data and control signals from one channel to the other. DLB can only be used when the peripheral is configured in duplex mode (that is, UPCTL.MODE = 2h or 3h). DLB is primarily useful for debug purposes, and requires no physical connections between channels. The standard uPP pin multiplexing must be applied, however, even though the pins are not used.

Table 7. Basic Operating Mode Selection

	uPP Channel Control Register (UPCTL) Bit		uPP Digital Loopback Register (UPDLB) Bit	
Operating Mode	**CHN**	**MODE**	**AB**	**BA**
1-Channel Transmit	0	1	0	0
1-Channel Receive	0	0	0	0
2-Channel Transmit	1	1	0	0
2-Channel Receive	1	0	0	0
2-Channel Duplex 0	1	2h	0	0
2-Channel Duplex 1	1	3h	0	0
2-Channel Duplex 0 (DLB)	1	2h	0	1
2-Channel Duplex 1 (DLB)	1	3h	1	0

Other than Table 7, there are several more choices to make (per channel):

- Data width – 8-bit, 9-bit to 16-bit
- Data packing – 9-bit to 15-bit data width only
- Data rate – single, double
- Data interleave –single channel only
- Clock divisor – transmit only
- Individual control signal enable
- Individual control signal polarity
- Idle value – transmit only
- Transmit threshold – transmit only
- DMA read burst size

Table 8 lists an example set of uPP parameters for duplex mode 0. This configuration places the uPP peripheral in duplex mode with Channel A receiving and Channel B transmitting. Each channel uses a 16-bit interface with a different data format.

Table 8. Sample uPP Parameters for Duplex Mode 0

Register	Register Field [1]	Setting	Description
UPCTL	DPFB	2h	Data packing: left-justified, zero fill
	DPWB	4h	12-bit data format
	IWB	1	16-bit
	DRB	0	Single data rate
	DPFA	—	Unused
	DPWA	0	16-bit data format
	IWA	1	16-bit
	DRA	0	Single data rate
	CHN	1	2-Channel
	MODE	2h	Duplex 0: A receive, B transmit
UPICR	CLKDIVB	1	Divide by 2 (total division of transmit clock: 4)
	CLKDIVA	-—	Unused
UPIVR	VALB	0BBBh	Note idle value is 12-bit data format; 4 MSBs unused
	VALA	—	Unused
UPIES	EOLQ	1	Turn on EOL interrupt for Channel B (DMA Channel Q)
	EOWQ	1	Turn on EOW interrupt for Channel B (DMA Channel Q)
	EOLI	1	Turn on EOL interrupt for Channel A (DMA Channel I)
	EOWI	1	Turn on EOW interrupt for Channel A (DMA Channel I)

(1) Unlisted register fields are left at their default values (typically 0), see Section 3.

2.6.3 System Tuning Tips

The uPP peripheral can operate at high speed and transfer data at a very high rate. When operating the uPP near its upper limits, tuning certain parameters can help decrease the incidence of errors and the software overhead incurred servicing uPP data. Table 9 lists several parameters that can be useful in system tuning. A parameter is defined as a “coarse” adjustment, if changing the parameter directly alters the peripheral throughput. A “fine” adjustment does not change the peripheral throughput, but it does affect general system performance.

Table 9. uPP Parameters Useful for System Tuning

Parameter	Register	Register Field	Edge Value	Safe Value	Description
Data Rate	UPCTL	DRB DRA	1	0	Double data rate increases data transfer by a factor of 2 and greatly increases system loading for the same clock divisor. This is a coarse adjustment and is probably fixed due to design constraints.
Clock Division	UPICR	CLKDIVB CLKDIVA	0	1+	Increasing clock division is the most straight-forward way to decrease system loading. This is a coarse adjustment; the difference between CLKDIVx = 0 and 1 is the same (in terms of data rate) as the difference between single and double data rate.
DMA Read Burst Size	UPTCR	RDSIZEQ RDSIZEI	0	3h	Increasing the DMA read threshold decreases system loading by generating fewer, larger DMA events. This is a fine adjustment.
DMA Line Size, Count	UPxD1	LNCNT BCNT		(1)(1)	Condensing uPP transfers into fewer, larger lines generates fewer end-of-line interrupts and, thus, invokes fewer ISR calls. This is a fine adjustment.
Total Transfer Size	UPxD1	LNCNT BCNT	(1)	(1)	Performing many small uPP transfers can require excessive software overhead (programming DMA descriptors, handling interrupts, etc.) at high data rates. This is a fine adjustment.
System Priority			(2)(2)	(2)	When the uPP operates in parallel with other data masters, such as EDMA, assigning higher priority to the uPP may help the uPP avoid underflow or overflow conditions. This is a fine adjustment.

(1) These values vary per application. One example could be a 16-KB transfer. The same total data could be transferred as 16 1-KB lines or 2 8-KB lines.

(2) System priority settings are not set within the uPP peripheral. See your device-specific *System Reference Guide* for more information.

2.6.4 Sample Interrupt Service Routine

The following pseudo-code serves as a template for writing a uPP interrupt service routing (ISR) function. Note that the uPP combines all events into a single CPU interrupt, and a new interrupt does not call the ISR if the previous interrupt still has not returned from the ISR. To allow future ISR calls, the uPP end-of-interrupt register (UPEOI) must be written with a zero value. Thus, the ISR should check for multiple events, and should continue rechecking after handling each individual event until no more events are found. Then, it must write UPEOI = 0 before returning.

```
Function upp_isr
{
      interrupt_status = UPIER

      while (interrupt_status != 0)
      {
            if (interrupt_status.EOLI)
            {
                  UPIER.EOLI = 1      // clear EOLI
                  // Handle EOLI...
            }
            if (interrupt_status.EOWI)
            {
                  UPIER.EOWI = 1      // clear EOWI
                  // Handle EOWI...
            }
            if (interrupt_status.ERRI)
            {
                  UPIER.ERRI = 1      // clear ERRI
                  // Handle ERRI...
            }
            if (interrupt_status.UORI)
            {
                  UPIER.UORI = 1      // clear UORI
                  // Handle UORI...
            }
            if (interrupt_status.DPEI)
            {
                  UPIER.DPEI = 1      // clear DPEI
                  // Handle DPEI...
            }

            if (interrupt_status.EOLQ)
            {
                  UPIER.EOLQ = 1      // clear EOLQ
                  // Handle EOLQ...
            }
            if (interrupt_status.EOWQ)
            {
                  UPIER.EOWQ = 1      // clear EOWQ
                  // Handle EOWQ...
            }
            if (interrupt_status.ERRQ)
            {
                  UPIER.ERRQ = 1      // clear ERRQ
                  // Handle ERRQ...
            }
            if (interrupt_status.UORQ)
            {
                  UPIER.UORQ = 1      // clear UORQ
                  // Handle UORQ...
            }
            if (interrupt_status.DPEQ)
            {
                  UPIER.DPEQ = 1      // clear DPEQ
                  // Handle DPEQ...
            }

            // loop again if any interrupts are left
            interrupt_status = UPIER
      } // end of while

      // write end of interrupt vector to allow future calls
      UPEOI = 0
} // end of function
```

2.7 Reset Considerations

2.7.1 Software Reset

Software reset clears the uPP internal state machines but does not reset the contents of the uPP registers. The following procedure performs a software reset on the uPP peripheral.

1. Write the EN bit in the uPP peripheral control register (UPPCR) to 0 (disables the uPP).
2. Poll the DB bit in UPPCR for activity; wait until DMA controller is inactive/idle.
3. Write the SWRST bit UPPCR to 1 (places uPP in software reset).
4. Write the SWRST bit UPPCR to 0 to (brings uPP out of software reset).

2.7.2 Hardware Reset

When the processor RESET pin is asserted, the entire processor is reset and is held in the reset state until the RESET pin is released. As part of a device reset, the uPP state machines are reset, and the uPP registers are forced to their default states (see Section 3).

2.8 Interrupt Support

The uPP peripheral generates eight interrupt events, all tied to internal DMA Channels I and Q. The uPP peripheral automatically combines all interrupt events into a single chip-level (CPU) interrupt. Individual events may be enabled using the uPP interrupt enable set register (UPIES) and disabled using the uPP interrupt enable clear register (UPIEC). Only enabled events generate interrupts and assert bits in the interrupt enabled status register (UPIER). Disabled events do not generate interrupts but do assert bits in the interrupt raw status register (UPISR).

An interrupt service routine (ISR) may be assigned to handle uPP CPU-level interrupts using the interrupt controller module. If uPP events occur in close proximity to one another, a single CPU interrupt (and a single call to the ISR) may represent multiple interrupt events. Thus, the uPP ISR must meet certain structural requirements:

- The ISR must be able to handle multiple events before returning.
- The ISR must handle any subsequent events that occur after it is called but before it returns.
- The ISR must write 00h to the uPP end-of-interrupt register (UPEOI) just before it returns. This allows future uPP events to generate a CPU interrupt.

Section 2.6 provides a sample ISR that demonstrates these requirements.

2.8.1 End of Line (EOL) Event

This event occurs each time that the DMA channel reaches the end of a line in the data window. Note that if the associated uPP interface channel is operating in transmit mode, this event may occur before the line's final bytes are actually transmitted over the data pins.

For small line size and fast data transfer, it is possible to "miss" EOL events if they occur faster than the user's code can handle them. This does not hinder uPP operation; the uPP peripheral continue processing data uninterrupted until the EOW event or some error condition is encountered.

2.8.2 End of Window (EOW) Event

This event occurs when the DMA channel reaches the end of its current data window. Note that if the associated uPP interface channel is operating in transmit mode, this event may occur shortly before the window's final bytes are actually transmitted over the data pins.

When an EOW event occurs, the DMA channel automatically begins the next DMA transfer if one has been pre-programmed into the channel descriptor registers. If no new transfer is preprogrammed, the DMA channel becomes idle. For small window size and fast data transfer, code overhead may make it impossible to maintain a constant flow of data through the uPP interface channel. This problem can be solved by increasing the DMA window size or decreasing the peripheral clock speed.

2.8.3 Internal Bus Error (ERR) Event

This event occurs when the uPP peripheral or its internal DMA controller encounters an internal bus error. After encountering this error, the uPP peripheral should be reset to avoid further problem.

2.8.4 Underrun or Overflow (UOR) Event

This event occurs when the DMA channel fails to keep up with incoming or outgoing data on its associated interface channel. Typically, this error indicates that background system activity has interfered with normal operation of the peripheral. It does not occur simply when a channel is allowed to idle. After encountering this error, the uPP peripheral should be reset when this event occurs.

This error should primarily occur when operating the uPP at high speed with significant system loading. To avoid this error, run the uPP at slower speeds or reduce background activity, such as non-uPP peripheral or DMA transactions. Additional tuning tips are given in Section 2.6.3.

2.8.5 DMA Programming Error (DPE) Event

This event occurs when the DMA channel descriptors are programmed while its PEND bit in the uPP DMA channel status register (UPxS2) is set to 1. A channel's descriptors should only be programmed while the channel's PEND bit is cleared to 0.

2.9 Power Management

The uPP peripheral can be placed in reduced-power modes to conserve power during periods of inactivity. For information on power management, see your device-specific *System Reference Guide*.

2.10 Emulation Considerations

The uPP peripheral stops running if any of three conditions are met:

- Peripheral Disable – EN bit in the uPP peripheral control register (UPPCR) is 0.
- Clock Stop – uPP acknowledges a clock stop request from the device power management module.
- Emulation Suspend – JTAG emulator halts chip while FREE = 0 and SOFT = 1 in UPPCR.

For other settings of FREE and SOFT, the uPP peripheral continues running during emulation halt.

When the uPP encounters a stop condition, it completes the current DMA burst transaction (if one is active) before stopping.

An I/O channel configured in transmit mode immediately places its pins in a high-impedance state and preserves the state of its internal state machines. Unless some reset event occurs (see Section 2.7), the channel can resume where it left off when the stop condition is cleared.

An I/O channel configured in receive mode immediately asserts its WAIT signal (see Section 2.5.5) and captures one additional data word. Further incoming data words are ignored as long as the stop condition persists.

2.11 Transmit and Receive FIFOs

Each of the uPP peripheral I/O channels has a 512-byte FIFO. In receive mode, the FIFO is divided into eight 64-byte blocks. In transmit mode, the FIFO is divided into blocks that can be set to 64, 128, or 256 bytes, configured by the TXSIZEA or TXSIZEB field in the uPP threshold configuration register (UPTCR). Transmission will not begin until the channel has loaded enough data to fill at least one full FIFO block.

The internal DMA channels may also be configured to use a read threshold of 64, 128, or 256 bytes using the RDSIZEI or RDSIZEQ field in UPTCR. The DMA write threshold is fixed at 64 bytes.

3 Registers

The system programmer has access to and control over any of the uPP registers that are listed in Table 10. These registers, which control uPP I/O and DMA operations, are available at 32-bit addresses in the device memory-map. The device-specific data sheet lists the base memory address of these registers.

Table 10. uPP Registers

Address Offset	Acronym	Register Description	Section
0	UPPID	uPP Peripheral Identification Register	Section 3.1
4h	UPPCR	uPP Peripheral Control Register	Section 3.2
8h	UPDLB	uPP Digital Loopback Register	Section 3.3
10h	UPCTL	uPP Channel Control Register	Section 3.4
14h	UPICR	uPP Interface Configuration Register	Section 3.5
18h	UPIVR	uPP Interface Idle Value Register	Section 3.6
1Ch	UPTCR	uPP Threshold Configuration Register	Section 3.7
20h	UPISR	uPP Interrupt Raw Status Register	Section 3.8
24h	UPIER	uPP Interrupt Enabled Status Register	Section 3.9
28h	UPIES	uPP Interrupt Enable Set Register	Section 3.10
2Ch	UPIEC	uPP Interrupt Enable Clear Register	Section 3.11
30h	UPEOI	uPP End-of-Interrupt Register	Section 3.12
40h	UPID0	uPP DMA Channel I Descriptor 0 Register	Section 3.13
44h	UPID1	uPP DMA Channel I Descriptor 1 Register	Section 3.14
48h	UPID2	uPP DMA Channel I Descriptor 2 Register	Section 3.15
50h	UPIS0	uPP DMA Channel I Status 0 Register	Section 3.16
54h	UPIS1	uPP DMA Channel I Status 1 Register	Section 3.17
58h	UPIS2	uPP DMA Channel I Status 2 Register	Section 3.18
60h	UPQD0	uPP DMA Channel Q Descriptor 0 Register	Section 3.19
64h	UPQD1	uPP DMA Channel Q Descriptor 1 Register	Section 3.20
68h	UPQD2	uPP DMA Channel Q Descriptor 2 Register	Section 3.21
70h	UPQS0	uPP DMA Channel Q Status 0 Register	Section 3.22
74h	UPQS1	uPP DMA Channel Q Status 1 Register	Section 3.23
78h	UPQS2	uPP DMA Channel Q Status 2 Register	Section 3.24

3.1 uPP Peripheral Identification Register (UPPID)

The uPP peripheral identification register (UPPID) reports the revision ID of the uPP peripheral. The UPPID is shown in Figure 16 and described in Table 11.

Figure 16. uPP Peripheral Identification Register (UPPID)

31	0
REVID	
R-4423 0100h	

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

Table 11. uPP Peripheral Identification Register (UPPID) Field Descriptions

Bit	Field	Value	Description
31-0	REVID	4423 0100h	Peripheral identification number.

3.2 uPP Peripheral Control Register (UPPCR)

The uPP peripheral control register (UPPCR) controls certain peripheral-level configuration settings for the uPP peripheral. Among these are global enable and reset states and rules governing its behavior during CPU emulation halt. This register also reports the current activity state of the uPP internal DMA controller. The UPPCR is shown in Figure 17 and described in Table 12.

Figure 17. uPP Peripheral Control Register (UPPCR)

31	16
Reserved	
R-0	

15	8	7	6	5	4	3	2	1	0
Reserved		DB	Reserved		SWRST	EN	RTEMU	SOFT	FREE
R-0		R-0	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 12. uPP Peripheral Control Register (UPPCR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7	DB		DMA burst status. Used to poll whether internal DMA is currently active. Writes to this field have no effect.
		0	Internal DMA is idle.
		1	Internal DMA is active.
6-5	Reserved	0	Reserved
4	SWRST		Software reset control. Asserting reset clears internal state machines and prevents device from running. For graceful reset, you should first clear the EN bit and poll the DB bit to make sure the DMA is idle, then assert the SWRST bit.
		0	Peripheral running (out of reset)
		1	Peripheral in reset
3	EN		Peripheral enable control. When transitioning to disabled status, peripheral completes any DMA transactions already in progress before stopping.
		0	Peripheral is disabled.
		1	Peripheral is enabled.
2	RTEMU		Real-time emulation control. When asserted, emulation halts/breakpoints halts pending transactions.
		0	Real-time emulation disabled. Peripheral continues pending transactions while program is halted.
		1	Real-time emulation enabled. Peripheral halts transactions while program is halted.
1	SOFT		Soft stop enable. Must be enabled to allow emulation to halt the peripheral.
		0	Soft stop is disabled.
		1	Soft stop is enabled.
0	FREE		Enable free run. This must be disabled to allow emulation to halt the peripheral.
		0	Free run is disabled.
		1	Free run is enabled.

3.3 uPP Digital Loopback Register (UPDLB)

The uPP digital loopback register (UPDLB) enables or disables the use of internal digital loopback in the uPP peripheral. Internal loopback may be used to transfer data from one uPP interface channel to another when the peripheral is configured in duplex mode. The interface Channel A and Channel B pins do not need to be connected for this operation, but the proper pin multiplexing must still be applied. The UPDLB is shown in Figure 18 and described in Table 13.

Figure 18. uPP Digital Loopback Register (UPDLB)

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	7	6	5	4	3	2	1	0
Reserved		BA	AB	Reserved											
R-0		R/W-0	R/W-0	R-0											

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

Table 13. uPP Digital Loopback Register (UPDLB) Field Descriptions

Bit	Field	Value	Description
31-14	Reserved	0	Reserved
13	BA		B-to-A digital loopback control. Assert to enable digital loopback, transmitting from Channel B to Channel A.
		0	Disable B-to-A digital loopback. Allows AB to be asserted.
		1	Enable B-to-A digital loopback. Requires AB = 0 and the MODE bit in the uPP channel control register (UPCTL) to be set to 2h.
12	AB		A-to-B digital loopback control. Set to enable digital loopback, transmitting from Channel A to Channel B.
		0	Disable A-to-B digital loopback. Allows BA to be asserted.
		1	Enable A-to-B digital loopback. Requires BA = 0 and the MODE bit in the uPP channel control register (UPCTL) to be set to 3h.
11-0	Reserved	0	Reserved

3.4 uPP Channel Control Register (UPCTL)

The uPP channel control register (UPCTL) controls uPP interface channel settings. This includes global settings, such as the channel count and data interleave, and channel-specific settings such as bit width and data rate.

The UPCTL also controls data formatting for 9-bit to 15-bit operating modes. For a channel with an N-bit interface, the 16 – N MSBs are written or replaced (in receive or transmit mode, respectively) according to the DPFB and DPFA fields. The uPP peripheral performs no data formatting for 8-bit and 16-bit modes. The UPCTL is shown in Figure 19 and described in Table 14.

Figure 19. uPP Channel Control Register (UPCTL)

31	30 29	28 26	25	24	23	22 21	20 18	17	16
Rsvd	DPFB	DPWB	IWB	DRB	Rsvd	DPFA	DPWA	IWA	DRA
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0

15 8
Reserved
R-0

7 5	4	3	2	1 0
Reserved	DDRDEMUX	SDRTXIL	CHN	MODE
R-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 14. uPP Channel Control Register (UPCTL) Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved
30-29	DPFB	0-3h	Channel B data packing format. Applies only to 9-bit to 15-bit modes (IWB = 1 and DPWB != 0).
		0	Right-justified, zero extended
		1h	Right-justified, sign extended
		2h	Left-justified, zero filled
		3h	Reserved
28-26	DPWB	0-7h	Channel B bit width. Applies only if IWB = 1.
		0	No data packing (8-bit or 16-bit case)
		1h	9-bit data format
		2h	10-bit data format
		3h	11-bit data format
		4h	12-bit data format
		5h	13-bit data format
		6h	14-bit data format
		7h	15-bit data format
25	IWB		Channel B interface width. Controls whether Channel B performs 8-bit or 16-bit transactions.
		0	8-bit interface
		1	16-bit interface
24	DRB		Channel B data rate. Controls whether Channel B operates at single or double rate.
		0	Single data rate
		1	Double data rate
23	Reserved	0	Reserved
22-21	DPFA	0-3h	Channel A data packing format. Applies only to 9-bit to 15-bit modes (IWA = 1 and DPWA != 0).
		0	Right-justified, zero extended
		1h	Right-justified, sign extended
		2h	Left-justified, zero filled
		3h	Reserved

Table 14. uPP Channel Control Register (UPCTL) Field Descriptions (continued)

Bit	Field	Value	Description
20-18	DPWA	0-7h	Channel A bit width. Applies only if IWA = 1.
		0	No data packing (8-bit or 16-bit case)
		1h	9-bit data format
		2h	10-bit data format
		3h	11-bit data format
		4h	12-bit data format
		5h	13-bit data format
		6h	14-bit data format
		7h	15-bit data format
17	IWA		Channel A interface width. Controls whether Channel A performs 8-bit or 16-bit transactions.
		0	8-bit interface
		1	16-bit interface
16	DRA		Channel A data rate. Controls whether Channel A operates at single or double rate.
		0	Single data rate
		1	Double data rate
15-5	Reserved	0	Reserved
4	DDRDEMUX		Double data rate demultiplexing mode. Only applies when DRA = 1.
		0	Disable. Each peripheral channel is associated with its own DMA channel.
		1	Enable. Both DMA channels service peripheral Channel A. Requires CHN = 0 and MODE = 0 or 1.
3	SDRTXIL		Single data rate transmit interleave mode. Only applies when DRA = 0.
		0	Disable. Each peripheral channel is associated with its own DMA channel.
		1	Enable. Both DMA channels service peripheral Channel A. Requires CHN = 0 and MODE = 1.
2	CHN		Interface channel number. Controls whether one or both interface channels (A, B) are active.
		0	Single channel mode. Only Channel A is active.
		1	Dual channel mode. Channel A and Channel B are both active.
1-0	MODE	0-3h	Operating mode. Controls the direction each active interface channel operates.
		0	All receive mode
		1h	All transmit mode
		2h	Duplex Mode 0. Channel A receives and Channel B transmits. Requires CHN = 1.
		3h	Duplex Mode 1. Channel A transmits and Channel B receives. Requires CHN = 1.

3.5 uPP Interface Configuration Register (UPICR)

The uPP interface configuration register (UPICR) controls the enable state and polarity of each uPP interface channel's pins. The polarity selection applies regardless of channel direction. The signal enable states only apply in either receive or transmit mode, but never both. The UPICR is shown in Figure 20 and described in Table 15.

NOTE: When initializing the uPP peripheral, the uPP interface configuration register (UPICR) must be programmed using a single, 32-bit write. Writing UPICR fields one-by-one can lead to unexpected results.

Figure 20. uPP Interface Configuration Register (UPICR)

31	30	29	28	27			24
Reserved		TRISB	CLKINVB	CLKDIVB			
R-0		R/W-0	R/W-0	R/W-0			
23	22	21	20	19	18	17	16
Reserved		WAITB	ENAB	STARTB	WAITPOLB	ENAPOLB	STARTPOLB
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15	14	13	12	11			8
Reserved		TRISA	CLKINVA	CLKDIVA			
R-0		R/W-0	R/W-0	R/W-0			
7	6	5	4	3	2	1	0
Reserved		WAITA	ENAA	STARTA	WAITPOLA	ENAPOLA	STARTPOLA
R-0		R/W-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 15. uPP Interface Configuration Register (UPICR) Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved
29	TRISB		Channel B high-impedance state. Controls interface Channel B while idle in transmit mode. Only applies when Channel B is configured in transmit mode using the MODE bit in the uPP channel control register (UPCTL).
		0	Channel B drives value from the VALB bit in the uPP interface idle value register (UPIVR) while idle.
		1	Channel B data pins are in a high-impedance state while idle.
28	CLKINVB		Channel B clock inversion. Controls clock signal polarity for interface Channel B.
		0	Clock is not inverted. Channel B signals align on rising edge of clock.
		1	Clock is inverted. Channel B signals align on falling edge of clock.
27-24	CLKDIVB	0-Fh	Clock divisor for Channel B. Only used when interface Channel B is configured in transmit mode using the MODE bit in the uPP channel control register (UPCTL). Applied divisor equals CLKDIVB + 1.
23-22	Reserved	0	Reserved
21	WAITB		Channel B WAIT signal enable. Controls use of WAIT signal for interface Channel B. Only applied when Channel B is configured in transmit mode using the MODE bit in the uPP channel control register (UPCTL). In receive mode, WAIT is always driven low.
		0	WAIT signal is disabled. Channel B ignores WAIT in transmit mode.
		1	WAIT signal is enabled. Channel B honors WAIT in transmit mode.
20	ENAB		Channel B ENABLE Signal Enable. Controls use of ENABLE signal for interface Channel B. Only applied when Channel B is configured in receive mode using the MODE bit in the uPP channel control register (UPCTL). In transmit mode, ENABLE is always driven.
		0	ENABLE signal is disabled. Channel B ignores ENABLE in receive mode.
		1	ENABLE signal is enabled. Channel B honors ENABLE in receive mode.

Table 15. uPP Interface Configuration Register (UPICR) Field Descriptions (continued)

Bit	Field	Value	Description
19	STARTB		Channel B START Signal Enable. Controls use of START signal for interface Channel B. Only applied when Channel B is configured in receive mode using the MODE bit in the uPP channel control register (UPCTL). In transmit mode, STARTB is always driven.
		0	START signal is disabled. Channel B ignores START in receive mode.
		1	START signal is enabled. Channel B honors START in receive mode.
18	WAITPOLB		Channel B WAIT Signal Polarity. Controls polarity of WAIT signal for interface Channel B.
		0	WAIT is active-high for Channel B.
		1	WAIT is active-low for Channel B.
17	ENAPOLB		Channel B ENABLE Signal Polarity. Controls polarity of ENABLE signal for interface Channel B.
		0	ENABLE is active-high for Channel B.
		1	ENABLE is active-low for Channel B.
16	STARTPOLB		Channel B START Signal Polarity. Controls polarity of START signal for interface Channel B.
		0	START is active-high for Channel B.
		1	START is active-low for Channel B.
15-14	Reserved	0	Reserved
13	TRISA		Channel A high-impedance state. Controls interface Channel A while idle in transmit mode. Only applies when Channel A is configured in transmit mode using the MODE bit in the uPP channel control register (UPCTL).
		0	Channel A drives value from the VALA bit in the uPP interface idle value register (UPIVR) while idle.
		1	Channel A data pins are in a high-impedance state while idle.
12	CLKINVA		Channel A clock inversion. Controls clock signal polarity for interface Channel A.
		0	Clock is not inverted. Channel A signals align on rising edge of clock.
		1	Clock is inverted. Channel A signals align on falling edge of clock.
11-8	CLKDIVA	0-Fh	Clock divisor for Channel A. Only used when interface Channel A is configured in transmit mode using the MODE bit in the uPP channel control register (UPCTL). Applied divisor equals CLKDIVA + 1.
7-6	Reserved	0	Reserved
5	WAITA		Channel A WAIT signal enable. Controls use of WAIT signal for interface Channel A. Only applied when Channel A is configured in transmit mode using the MODE bit in the uPP channel control register (UPCTL). In receive mode, WAIT is always driven low.
		0	WAIT signal is disabled. Channel A ignores WAIT in transmit mode.
		1	WAIT signal is enabled. Channel A honors WAIT in transmit mode.
4	ENAA		Channel A ENABLE Signal Enable. Controls use of ENABLE signal for interface Channel B. Only applied when Channel A is configured in receive mode using the MODE bit in the uPP channel control register (UPCTL). In transmit mode, ENABLE is always driven.
		0	ENABLE signal is disabled. Channel A ignores ENABLE in receive mode.
		1	ENABLE signal is enabled. Channel A honors ENABLE in receive mode.
3	STARTA		Channel A START Signal Enable. Controls use of START signal for interface Channel A. Only applied when Channel A is configured in receive mode using the MODE bit in the uPP channel control register (UPCTL). In transmit mode, STARTA is always driven.
		0	START signal is disabled. Channel A ignores START in receive mode.
		1	START signal is enabled. Channel A honors START in receive mode.
2	WAITPOLA		Channel A WAIT Signal Polarity. Controls polarity of WAIT signal for interface Channel A.
		0	WAIT is active-high for Channel A.
		1	WAIT is active-low for Channel A.
1	ENAPOLA		Channel A ENABLE Signal Polarity. Controls polarity of ENABLE signal for interface Channel A.
		0	ENABLE is active-high for Channel A.
		1	ENABLE is active-low for Channel A.
0	STARTPOLA		Channel A START Signal Polarity. Controls polarity of START signal for interface Channel A.
		0	START is active-high for Channel A.
		1	START is active-low for Channel A.

3.6 uPP Interface Idle Value Register (UPIVR)

The uPP interface idle value register (UPIVR) controls the value that each interface channel transmits while idle. Note that this only applies when the associated channel is configured in transmit mode and also depends on the value of the TRISA and TRISB bits in the uPP interface configuration register (UPICR). The UPIVR is shown in Figure 21 and described in Table 16.

Figure 21. uPP Interface Idle Value Register (UPIVR)

31	16
VALB	
R/W-0	
15	0
VALA	
R/W-0	

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

Table 16. uPP Interface Idle Value Register (UPIVR) Field Descriptions

Bit	Field	Value	Description
31-16	VALB	0-FFFFh	Channel B idle value. Sets idle value for interface Channel B. This value is output on the Channel B data pins when the channel is idle, configured in transmit mode, and the TRISB bit in the uPP interface configuration register (UPICR) is cleared to 0.
15-0	VALA	0-FFFFh	Channel A idle value. Sets idle value for interface Channel A. This value is output on the Channel A data pins when the channel is idle, configured in transmit mode, and the TRISA bit in the uPP interface configuration register (UPICR) is cleared to 0.

3.7 uPP Threshold Configuration Register (UPTCR)

The uPP threshold configuration register (UPTCR) controls the transmit threshold for each interface channel and the read threshold for each DMA channel. Using larger thresholds can decrease internal bus traffic and increase performance, especially when the uPP is operating in a loaded system. The UPTCR is shown in Figure 22 and described in Table 17.

Figure 22. uPP Threshold Configuration Register (UPTCR)

31	26	25	24	23	18	17	16
Reserved		TXSIZEB		Reserved		TXSIZEA	
R-0		R/W-0		R-0		R/W-0	

15	10	9	8	7	2	1	0
Reserved		RDSIZEQ		Reserved		RDSIZEI	
R-0		R/W-0		R-0		R/W-0	

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

Table 17. uPP Threshold Configuration Register (UPTCR) Field Descriptions

Bit	Field	Value	Description
31-26	Reserved	0	Reserved
25-24	TXSIZEB	0-3h	Transmit threshold for Channel B. Controls the number of bytes that interface Channel B waits before beginning transmission. Only applies when Channel B is configured in transmit mode using the MODE bit in the uPP channel control register (UPCTL).
		0	64 bytes
		1	128 bytes (requires DMA descriptor byte count greater than 64)
		2h	Reserved
		3h	256 bytes (requires DMA descriptor byte count greater than 192)
23-18	Reserved	0	Reserved
17-16	TXSIZEA	0-3h	Transmit threshold for Channel A. Controls the number of bytes that interface Channel A waits before beginning transmission. Only applies when Channel A is configured in transmit mode using the MODE bit in the uPP channel control register (UPCTL).
		0	64 bytes
		1	128 bytes (requires DMA descriptor byte count greater than 64)
		2h	Reserved
		3h	256 bytes (requires DMA descriptor byte count greater than 192)
15-10	Reserved	0	Reserved
9-8	RDSIZEQ	0-3h	Read threshold for DMA Channel Q. Controls burst size for DMA Channel Q.
		0	64 bytes
		1	128 bytes
		2h	Reserved
		3h	256 bytes
7-2	Reserved	0	Reserved
1-0	RDSIZEI	0-3h	Read threshold for DMA Channel I. Controls burst size for DMA Channel I.
		0	64 bytes
		1	128 bytes
		2h	Reserved
		3h	256 bytes

3.8 uPP Interrupt Raw Status Register (UPISR)

The uPP interrupt raw status register (UPISR) reports the raw interrupt status for various conditions. Each status bit reads 1 when the associated event occurs, even if that interrupt event is disabled in the uPP interrupt enable clear register (UPIEC). Writing 1 to any bit simulates the associated interrupt event; writing 0 has no effect. The UPISR is shown in Figure 23 and described in Table 18.

Figure 23. uPP Interrupt Raw Status Register (UPISR)

31	16
Reserved	
R-0	

15	13	12	11	10	9	8
Reserved		EOLQ	EOWQ	ERRQ	UORQ	DPEQ
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	5	4	3	2	1	0
Reserved		EOLI	EOWI	ERRI	UORI	DPEI
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 18. uPP Interrupt Raw Status Register (UPISR) Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0	Reserved
12	EOLQ		Reports raw interrupt status for end-of-line condition (EOL) on DMA Channel Q.
		0	No EOL occurred
		1	EOL occurred
11	EOWQ		Reports raw interrupt status for end-of-window condition (EOW) on DMA Channel Q.
		0	No EOW occurred
		1	EOW occurred
10	ERRQ		Reports raw interrupt status for internal bus error condition (ERR) on DMA Channel Q.
		0	No error occurred
		1	Error occurred
9	UORQ		Reports raw interrupt status for underrun or overflow condition (UOR) on DMA Channel Q.
		0	No underrun or overflow occurred
		1	Underrun or overflow occurred
8	DPEQ		Reports raw interrupt status for programming error condition (DPE) on DMA Channel Q.
		0	No error occurred
		1	Error occurred
7-5	Reserved	0	Reserved
4	EOLI		Reports raw interrupt status for end-of-line condition (EOL) on DMA Channel I.
		0	No EOL occurred
		1	EOL occurred
3	EOWI		Reports raw interrupt status for end-of-window condition (EOW) on DMA Channel I.
		0	No EOW occurred
		1	EOW occurred
2	ERRI		Reports raw interrupt status for internal bus error condition (ERR) on DMA Channel I.
		0	No error occurred
		1	Error occurred
1	UORI		Reports raw interrupt status for underrun or overflow condition (UOR) on DMA Channel I.
		0	No underrun or overflow occurred
		1	Underrun or overflow occurred

Table 18. uPP Interrupt Raw Status Register (UPISR) Field Descriptions (continued)

Bit	Field	Value	Description
0	DPEI		Reports raw interrupt status for programming error condition (DPE) on DMA Channel I.
		0	No error occurred
		1	Error occurred

3.9 uPP Interrupt Enabled Status Register (UPIER)

The uPP interrupt enabled status register (UPIER) reports the enabled interrupt status for various conditions. Each status bit reads 1, if the associated event occurs while that event is enabled in the uPP interrupt enabled set register (UPIES). If the interrupt event is disabled in the uPP interrupt enable clear register (UPIEC), the associated status bit always reads 0. Writing 1 to any bit clears the associated interrupt event; writing 0 has no effect. The UPIER is shown in Figure 24 and described in Table 19.

Figure 24. uPP Interrupt Enabled Status Register (UPIER)

31	16
Reserved	
R-0	

15	13	12	11	10	9	8
Reserved		EOLQ	EOWQ	ERRQ	UORQ	DPEQ
R-0		R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0

7	5	4	3	2	1	0
Reserved		EOLI	EOWI	ERRI	UORI	DPEI
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; W1C = Write 1 to clear bit (writing 0 has no effect); -*n* = value after reset

Table 19. uPP Interrupt Enabled Status Register (UPIER) Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0	Reserved
12	EOLQ		Interrupt Status for Channel Q End-of-Line. Reports enabled interrupt status for end-of-line condition (EOL) on DMA Channel Q.
		0	No EOL
		1	EOL occurred
11	EOWQ		Interrupt Status for Channel Q End-of-Window. Reports enabled interrupt status for end-of-window condition (EOW) on DMA Channel Q.
		0	No EOW
		1	EOW occurred
10	ERRQ		Interrupt Status for Channel Q Error. Reports enabled interrupt status for internal bus error condition on DMA Channel Q.
		0	No error
		1	Error occurred
9	UORQ		Interrupt Status for Channel Q Underrun/Overflow condition. Reports enabled interrupt status for underrun or overflow condition on DMA Channel Q.
		0	No underrun or overflow
		1	Underrun or overflow occurred
8	DPEQ		Interrupt Status for Channel Q Programming Error. Reports enabled interrupt status for programming error condition on DMA Channel Q.
		0	No error
		1	Error occurred
7-5	Reserved	0	Reserved
4	EOLI		Interrupt Status for Channel I End-of-Line. Reports enabled interrupt status for end-of-line condition (EOL) on DMA Channel I.
		0	No EOL
		1	EOW occurred
3	EOWI		Interrupt Status for Channel I End-of-Window. Reports enabled interrupt status for end-of-window condition (EOW) on DMA Channel I.
		0	No EOW
		1	EOW occurred

Table 19. uPP Interrupt Enabled Status Register (UPIER) Field Descriptions (continued)

Bit	Field	Value	Description
2	ERRI		Interrupt Status for Channel I Error. Reports enabled interrupt status for internal bus error condition on DMA Channel I.
		0	No error
		1	Error occurred
1	UORI		Interrupt Status for Channel I Underrun/Overflow condition. Reports enabled interrupt status for underrun or overflow condition on DMA Channel I.
		0	No underrun or overflow
		1	Underrun or overflow occurred
0	DPEI		Interrupt Status for Channel I Programming Error. Reports enabled interrupt status for programming error condition on DMA Channel I.
		0	No error
		1	Error occurred

3.10 uPP Interrupt Enable Set Register (UPIES)

The uPP interrupt enabled set register (UPIES) controls whether individual interrupt events generate a CPU interrupt. Writing 1 to any bit enables CPU interrupt generation for the associated uPP event; writing 0 has no effect.

Reads from both UPIES and the uPP interrupt enable clear register (UPIEC) access the same internal interrupt enable register, and a value of 1 indicates that the corresponding interrupt is enabled. The UPIES is shown in Figure 25 and described in Table 20.

Figure 25. uPP Interrupt Enable Set Register (UPIES)

31	16
Reserved	
R-0	

15	13	12	11	10	9	8
Reserved		EOLQ	EOWQ	ERRQ	UORQ	DPEQ
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	5	4	3	2	1	0
Reserved		EOLI	EOWI	ERRI	UORI	DPEI
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 20. uPP Interrupt Enable Set Register (UPIES) Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0	Reserved
12	EOLQ		Interrupt Enable Set for Channel Q End-of-Line. Reports interrupt enable for end-of-line condition (EOL) on DMA Channel Q.
		0	Read: EOL interrupt is disabled. Write: no effect.
		1	Read: EOL interrupt is enabled. Write: enable EOL interrupt.
11	EOWQ		Interrupt Enable Set for Channel Q End-of-Window. Reports interrupt enable for end-of-window condition (EOW) on DMA Channel Q.
		0	Read: EOW interrupt is disabled. Write: no effect.
		1	Read: EOW interrupt is enabled. Write: enable EOW interrupt.
10	ERRQ		Interrupt Enable Set for Channel Q Error. Reports interrupt enable for internal bus error condition on DMA Channel Q.
		0	Read: Error interrupt is disabled. Write: no effect.
		1	Read: Error interrupt is enabled. Write: enable ERR interrupt.
9	UORQ		Interrupt Enable Set for Channel Q Underrun/Overflow condition. Reports interrupt enable for underrun or overflow condition on DMA Channel Q.
		0	Read: Underrun or overflow interrupt is disabled. Write: no effect.
		1	Read: Underrun or overflow interrupt is enabled. Write: enable UOR interrupt.
8	DPEQ		Interrupt Enable Set for Channel Q Programming Error. Reports interrupt enable for programming error condition on DMA Channel Q.
		0	Read: Programming error interrupt is disabled. Write: no effect.
		1	Read: Programming error interrupt is enabled. Write: enable DPE interrupt.
7-5	Reserved	0	Reserved
4	EOLI		Interrupt Enable Set for Channel I End-of-Line. Reports interrupt enable for end-of-line condition (EOL) on DMA Channel I.
		0	Read: EOL interrupt is disabled. Write: no effect.
		1	Read: EOL interrupt is enabled. Write: enable EOL interrupt.
3	EOWI		Interrupt Enable Set for Channel I End-of-Window. Reports interrupt enable for end-of-window condition (EOW) on DMA Channel I.
		0	Read: EOW interrupt is disabled. Write: no effect.
		1	Read: EOW interrupt is enabled. Write: enable EOW interrupt.

Table 20. uPP Interrupt Enable Set Register (UPIES) Field Descriptions (continued)

Bit	Field	Value	Description
2	ERRI		Interrupt Enable Set for Channel I Error. Reports interrupt enable for internal bus error condition on DMA Channel I.
		0	Read: Error interrupt is disabled. Write: no effect.
		1	Read: Error interrupt is enabled. Write: enable ERR interrupt.
1	UORI		Interrupt Enable Set for Channel I Underrun/Overflow condition. Reports interrupt enable for underrun or overflow condition on DMA Channel I.
		0	Read: Underrun or overflow interrupt is disabled. Write: no effect.
		1	Read: Underrun or overflow interrupt is enabled. Write: enable UOR interrupt.
0	DPEI		Interrupt Enable Set for Channel I Programming Error. Reports interrupt enable for programming error condition on DMA Channel I.
		0	Read: Programming error interrupt is disabled. Write: no effect.
		1	Read: Programming error interrupt is enabled. Write: enable DPE interrupt.

3.11 uPP Interrupt Enable Clear Register (UPIEC)

The uPP interrupt enable clear register (UPIEC) controls whether individual interrupt events generate a CPU interrupt. Writing 1 to any bit disables CPU interrupt generation for the associated uPP event; writing 0 has no effect.

Reads from both UPIEC and the uPP interrupt enabled set register (UPIES) access the same internal interrupt enable register, and a value of 1 indicates that the corresponding interrupt is enabled. The UPIEC is shown in Figure 26 and described in Table 21.

Figure 26. uPP Interrupt Enable Clear Register (UPIEC)

31	16
Reserved	
R-0	

15	13	12	11	10	9	8
Reserved		EOLQ	EOWQ	ERRQ	UORQ	DPEQ
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	5	4	3	2	1	0
Reserved		EOLI	EOWI	ERRI	UORI	DPEI
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. uPP Interrupt Enable Clear Register (UPIEC) Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0	Reserved
12	EOLQ		Interrupt Enable Clear for Channel Q End-of-Line. Reports interrupt enable for end-of-line condition (EOL) on DMA Channel Q.
		0	Read: EOL interrupt is disabled. Write: no effect.
		1	Read: EOL interrupt is enabled. Write: disable EOL interrupt
11	EOWQ		Interrupt Enable Clear for Channel Q End-of-Window. Reports interrupt enable for end-of-window condition (EOW) on DMA Channel Q.
		0	Read: EOW interrupt is disabled. Write: no effect.
		1	Read: EOW interrupt is enabled. Write: disable EOW interrupt
10	ERRQ		Interrupt Enable Clear for Channel Q Error. Reports interrupt enable for internal bus error condition on DMA Channel Q.
		0	Read: Error interrupt is disabled. Write: no effect.
		1	Read: Error interrupt is enabled. Write: disable ERR interrupt.
9	UORQ		Interrupt Enable Clear for Channel Q Underrun/Overflow condition. Reports interrupt enable for underrun or overflow condition on DMA Channel Q.
		0	Read: Underrun or overflow interrupt is disabled. Write: no effect.
		1	Read: Underrun or overflow interrupt is enabled. Write: disable UOR interrupt.
8	DPEQ		Interrupt Enable Clear for Channel Q Programming Error. Reports interrupt enable for programming error condition on DMA Channel Q.
		0	Read: Programming error interrupt is disabled. Write: no effect.
		1	Read: Programming error interrupt is enabled. Write: disable DPE interrupt.
7-5	Reserved	0	Reserved
4	EOLI		Interrupt Enable Clear for Channel I End-of-Line. Reports interrupt enable for end-of-line condition (EOL) on DMA Channel I.
		0	Read: EOL interrupt is disabled. Write: no effect.
		1	Read: EOL interrupt is enabled. Write: disable EOL interrupt.
3	EOWI		Interrupt Enable Clear for Channel I End-of-Window. Reports interrupt enable for end-of-window condition (EOW) on DMA Channel I.
		0	Read: EOW interrupt is disabled. Write: no effect.
		1	Read: EOW interrupt is enabled. Write: disable EOW interrupt.

Table 21. uPP Interrupt Enable Clear Register (UPIEC) Field Descriptions (continued)

Bit	Field	Value	Description
2	ERRI		Interrupt Enable Clear for Channel I Error. Reports interrupt enable for internal bus error condition on DMA Channel I.
		0	Read: Error interrupt is disabled. Write: no effect.
		1	Read: Error interrupt is enabled. Write: disable ERR interrupt.
1	UORI		Interrupt Enable Clear for Channel I Underrun/Overflow condition. Reports interrupt enable for underrun or overflow condition on DMA Channel I.
		0	Read: Underrun or overflow interrupt is disabled. Write: no effect.
		1	Read: Underrun or overflow interrupt is enabled. Write: disable UOR interrupt.
0	DPEI		Interrupt Enable Clear for Channel I Programming Error. Reports interrupt enable for programming error condition on DMA Channel I.
		0	Read: Programming error interrupt is disabled. Write: no effect.
		1	Read: Programming error interrupt is enabled. Write: disable DPE interrupt.

3.12 uPP End of Interrupt Register (UPEOI)

The uPP end of interrupt register (UPEOI) acknowledges CPU interrupts generated from uPP events. The EOI bit field must be written with 00h at the end of the interrupt service routine (ISR) that handles uPP interrupts. Until this acknowledgement occurs, no uPP event can generate another CPU interrupt. The UPEOI is shown in Figure 27 and described in Table 22.

Figure 27. uPP End of Interrupt Register (UPEOI)

31	16
Reserved	
R-0	

15	8	7	0
Reserved		EOI	
R-0		R/W-0	

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

Table 22. uPP End of Interrupt Register (UPEOI) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	EOI	0-FFh	End of interrupt value. Write 00h after uPP interrupt to allow interrupt generation from subsequent uPP events.

3.13 uPP DMA Channel I Descriptor 0 Register (UPID0)

The uPP DMA channel I descriptor 0 register (UPID0) programs the starting address of the data buffer, or window, for DMA Channel I. The address is programmed by writing a 32-bit value to the entire register. Note that the 3 lower bits are read-only and always equal 0, so that data buffers are properly aligned in memory. The UPID0 is shown in Figure 28 and described in Table 23.

Figure 28. uPP DMA Channel I Descriptor 0 Register (UPID0)

31	16
ADDRH	
R/W-0	

15	3	2	0
ADDRH		ADDR	
R/W-0		R-0	

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

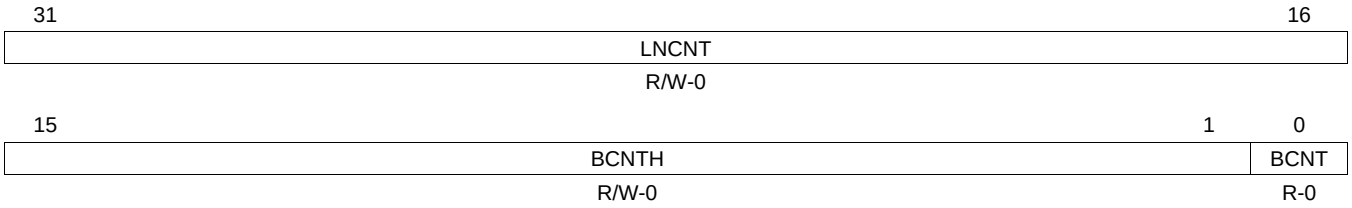
Table 23. uPP DMA Channel I Descriptor 0 Register (UPID0) Field Descriptions

Bit	Field	Value	Description
31-3	ADDRH	0-1FFF FFFFh	Window Address MSBs. Sets the 29 most-significant bits of starting address for DMA Channel I window.
2-0	ADDR	0	Window Address LSBs. Forces window address to align to multiple of 8 bytes (64-bit buffer alignment).

3.14 uPP DMA Channel I Descriptor 1 Register (UPID1)

The uPP DMA channel I descriptor 1 register (UPID1) programs the line count per window and byte count per line for DMA Channel I. The line count (LNCNT) may be set to any number from 1 to 65 535 (FFFFh), but must not be cleared to 0. The byte count (BCNT) may only be set to an even number. For a simple transfer, LNCNT may be set to 1, and BCNT may be set to N >> 1, where N is the desired byte count of the entire DMA transfer. Note that the lower bit is read-only and is always equal 0, so that N is an even number. The UPID1 is shown in Figure 29 and described in Table 24.

Figure 29. uPP DMA Channel I Descriptor 1 Register (UPID1)



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

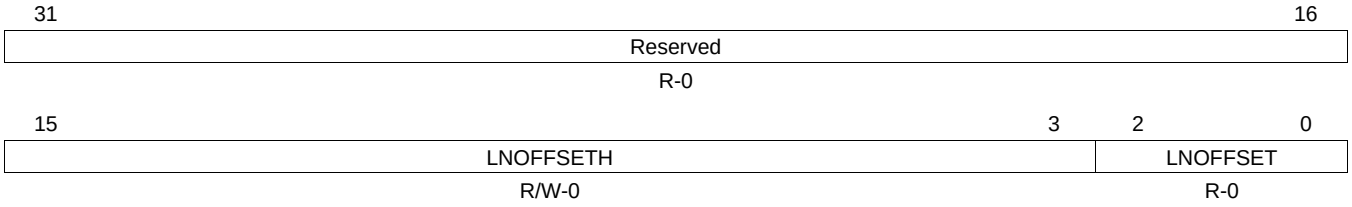
Table 24. uPP DMA Channel I Descriptor 1 Register (UPID1) Field Descriptions

Bit	Field	Value	Description
31-16	LNCNT	1-FFFFh	Line Count. Sets the number of lines in the DMA Channel I window.
		0	Invalid value
15-1	BCNTH	1-7FFFh	Byte Count MSBs. Sets the 15 most-significant bits of the number of bytes per line in the DMA Channel I window.
		0	Invalid value
0	BCNT	0	Byte Count LSB. Forces the number of bytes per line to an even value (multiple of 2 bytes).

3.15 uPP DMA Channel I Descriptor 2 Register (UPID2)

The uPP DMA channel I descriptor 2 register (UPID2) programs the offset address between lines within the DMA Channel I window. Note that the 3 lower bits are read-only and always equal 0, so that the line offset address is aligned to a multiple of 8 bytes, similar to the window address in UPID0. Writing a value of 0 to UPID2 effectively repeats the same (first) line *UPID1.LNCNT* times. The UPID2 is shown in Figure 30 and described in Table 25.

Figure 30. uPP DMA Channel I Descriptor 2 Register (UPID2)



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

Table 25. uPP DMA Channel I Descriptor 2 Register (UPID2) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-3	LNOFFSETH	0-1FFFh	Line Offset Address MSBs. Sets the 13 most-significant bits of the offset address (in bytes) between lines in the DMA Channel I window.
2-0	LNOFFSET	0	Line Offset Address LSBs. Forces the line offset address to align to a multiple of 8 bytes (64-bit alignment).

3.16 uPP DMA Channel I Status 0 Register (UPIS0)

The uPP DMA channel I status 0 register (UPIS0) reports the current address of the DMA Channel I transfer. The UPIS0 is shown in Figure 31 and described in Table 26.

Figure 31. uPP DMA Channel I Status 0 Register (UPIS0)

31	0
ADDR	
R-0	

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

Table 26. uPP DMA Channel I Status 0 Register (UPIS0) Field Descriptions

Bit	Field	Value	Description
31-0	ADDR	0-FFFF FFFFh	DMA Current Address. Reports the current address of the DMA Channel I transfer.

3.17 uPP DMA Channel I Status 1 Register (UPIS1)

The uPP DMA channel I status 1 register (UPIS1) reports the current line number and the byte position within the current line of the DMA Channel I transfer. The UPIS1 is shown in Figure 32 and described in Table 27.

Figure 32. uPP DMA Channel I Status 1 Register (UPIS1)

31	16
LNCNT	
R-0	

15	0
BCNT	
R-0	

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

Table 27. uPP DMA Channel I Status 1 Register (UPIS1) Field Descriptions

Bit	Field	Value	Description
31-16	LNCNT	0-FFFFh	DMA Current Line Number. Reports the current line number of the DMA Channel I transfer.
15-0	BCNT	0-FFFFh	DMA Byte Number. Reports the current byte position within the current line of the DMA Channel I transfer.

3.18 uPP DMA Channel I Status 2 Register (UPIS2)

The uPP DMA channel I status register 2 (UPIS2) reports the status of the current DMA Channel I transfer. The PEND bit is used to determine when a new transfer may be programmed into the uPP DMA channel I descriptor registers (UPID*n*). The UPIS2 is shown in Figure 33 and described in Table 28.

Figure 33. uPP DMA Channel I Status 2 Register (UPIS2)

31	16
Reserved	
R-0	

15	8	7	4	3	2	1	0
Reserved		WM		Reserved		PEND	ACT
R-0		R-0		R-0		R-0	R-0

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

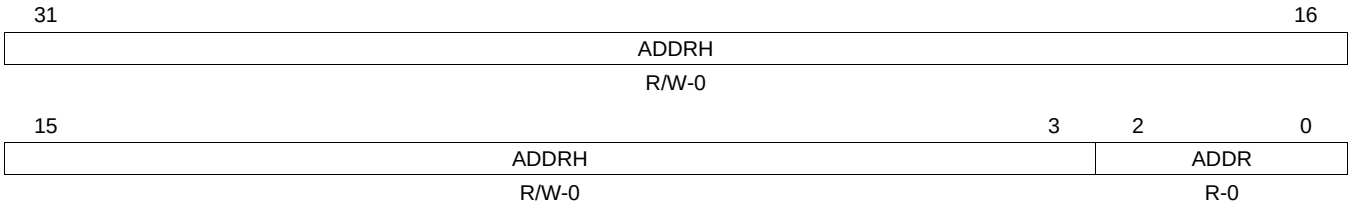
Table 28. uPP DMA Channel I Status 2 Register (UPIS2) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-4	WM	0-Fh	DMA Watermark. When the associated interface channel operates in receive mode, this field records the maximum FIFO block occupancy reached during any transaction. When the associated interface channel operates in transmit mode, this field records the FIFO block emptiness and is overwritten every uPP interface clock cycle.
3-2	Reserved	0	Reserved
1	PEND		DMA Transfer Pending. Reports whether another DMA transfer is pending for DMA Channel I. This field must be low before another transfer may be programmed.
		0	No transfer pending. Channel I descriptors may be written.
		1	Transfer pending. Channel I descriptors may not be written.
0	ACT		DMA Active. Reports the current status of DMA Channel I. This field should not be used to determine whether the DMA Channel I descriptors are programmable; use the PEND bit instead.
		0	DMA is inactive.
		1	DMA is active.

3.19 uPP DMA Channel Q Descriptor 0 Register (UPQD0)

The uPP DMA channel Q descriptor 0 register (UPQD0) programs the starting address of the data buffer, or window, for DMA Channel Q. The address can be programmed by simply writing a 32-bit value to the entire register. Note that the 3 lower bits are read-only and always equal to 0, so that data buffers must be properly aligned in memory. The UPQD0 is shown in Figure 34 and described in Table 29.

Figure 34. uPP DMA Channel Q Descriptor 0 Register (UPQD0)



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

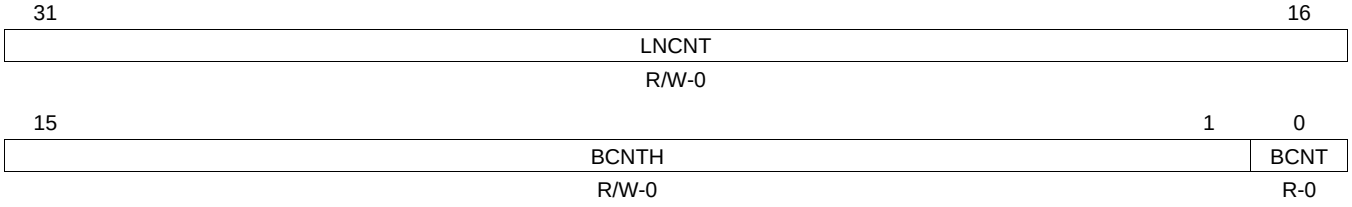
Table 29. uPP DMA Channel Q Descriptor 0 Register (UPQD0) Field Descriptions

Bit	Field	Value	Description
31-3	ADDRH	0-1FFF FFFFh	Window Address MSBs. Sets the 29 most-significant bits of starting address for DMA Channel Q window.
2-0	ADDR	0	Window Address LSBs. Forces window address to align to multiple of 8 bytes (64-bit buffer alignment).

3.20 uPP DMA Channel Q Descriptor 1 Register (UPQD1)

The uPP DMA channel Q descriptor 1 register (UPQD1) programs the line count per window and byte count per line for DMA Channel Q. The line count (LNCNT) may be set to any number from 1 to 65 535 (FFFFh), but must not be cleared to 0. The byte count (BCNT) may only be set to an even number. For a simple transfer, LNCNT may be set to 1, and BCNT may be set to N >> 1, where N is the desired byte count of the entire DMA transfer. Note that the lower bit is read-only and is always equal 0, so that N is an even number. The UPQD1 is shown in Figure 35 and described in Table 30.

Figure 35. uPP DMA Channel Q Descriptor 1 Register (UPQD1)



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

Table 30. uPP DMA Channel Q Descriptor 1 Register (UPQD1) Field Descriptions

Bit	Field	Value	Description
31-16	LNCNT	1-FFFFh	Line Count. Sets the number of lines in the DMA Channel Q window.
		0	Invalid value
15-1	BCNTH	1-7FFFh	Byte Count MSBs. Sets the 15 most-significant bits of the number of bytes per line in the DMA Channel Q window.
		0	Invalid value
0	BCNT	0	Byte Count LSB. Forces the number of bytes per line to an even value (multiple of 2 bytes).

3.21 uPP DMA Channel Q Descriptor 2 Register (UPQD2)

The uPP DMA channel Q descriptor 2 register (UPQD2) programs the offset address between lines within the DMA Channel Q window. Note that the 3 lower bits are read-only and always equal 0, so that the line offset address is aligned to a multiple of 8 bytes, similar to the window address in UPQD0. Writing a value of 0 to UPQD2 effectively repeats the same (first) line *UPQD1.LNCNT* times. The UPQD2 is shown in Figure 36 and described in Table 31.

Figure 36. uPP DMA Channel Q Descriptor 2 Register (UPID2)

31	16
Reserved	
R-0	

15	3	2	0
LNOFFSETH		LNOFFSET	
R/W-0		R-0	

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

Table 31. uPP DMA Channel Q Descriptor 2 Register (UPID2) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-3	LNOFFSETH	0-1FFFh	Line Offset Address MSBs. Sets the 13 most-significant bits of the offset address (in bytes) between lines in the DMA Channel Q window.
2-0	LNOFFSET	0	Line Offset Address LSBs. Forces the line offset address to align to a multiple of 8 bytes (64-bit alignment).

3.22 uPP DMA Channel Q Status 0 Register (UPQS0)

The uPP DMA channel Q status 0 register (UPQS0) reports the current address of the DMA Channel Q transfer. The UPQS0 is shown in Figure 37 and described in Table 32.

Figure 37. uPP DMA Channel Q Status 0 Register (UPQS0)

31	0
ADDR	
R-0	

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

Table 32. uPP DMA Channel Q Status 0 Register (UPQS0) Field Descriptions

Bit	Field	Value	Description
31-0	ADDR	0-FFFF FFFFh	DMA Current Address. Reports the current address of the DMA Channel Q transfer.

3.23 uPP DMA Channel Q Status 1 Register (UPQS1)

The uPP DMA channel Q status 1 register (UPQS1) reports the current line number and the byte position within the current line of the DMA Channel Q transfer. The UPQS1 is shown in Figure 38 and described in Table 33.

Figure 38. uPP DMA Channel Q Status 1 Register (UPQS1)

31	16
LNCNT	
R-0	

15	0
BCNT	
R-0	

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

Table 33. uPP DMA Channel Q Status 1 Register (UPQS1) Field Descriptions

Bit	Field	Value	Description
31-16	LNCNT	0-FFFFh	DMA Current Line Number. Reports the current line number of the DMA Channel Q transfer.
15-0	BCNT	0-FFFFh	DMA Byte Number. Reports the current byte position within the current line of the DMA Channel Q transfer.

3.24 uPP DMA Channel Q Status 2 Register (UPQS2)

The uPP DMA channel Q status 2 register (UPQS2) reports the status of the current DMA Channel Q transfer. The PEND bit is used to determine when a new transfer may be programmed into the into the uPP DMA channel Q descriptor registers (UPQD*n*). The UPQS2 is shown in Figure 39 and described in Table 34.

Figure 39. uPP DMA Channel Q Status 2 Register (UPQS2)

31	16
Reserved	
R-0	

15	8	7	4	3	2	1	0
Reserved		WM		Reserved		PEND	ACT
R-0		R-0		R-0		R-0	R-0

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

Table 34. uPP DMA Channel Q Status 2 Register (UPQS2) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-4	WM	0-Fh	DMA Watermark. When the associated interface channel operates in receive mode, this field records the maximum FIFO block occupancy reached during any transaction. When the associated interface channel operates in transmit mode, this field records the FIFO block emptiness and is overwritten every uPP interface clock cycle.
3-2	Reserved	0	Reserved
1	PEND		DMA Transfer Pending. Reports whether another DMA transfer is pending for DMA Channel Q. This field must be low before another transfer may be programmed.
		0	No transfer pending. Channel Q descriptors may be written.
		1	Transfer pending. Channel Q descriptors may not be written.
0	ACT		DMA Active. Reports the current status of DMA Channel Q. This field should not be used to determine whether the DMA Channel Q descriptors are programmable; use the PEND bit instead.
		0	DMA is inactive.
		1	DMA is active.

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 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. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

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.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

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
DLP® Products	www.dlp.com	Communications and Telecom	www.ti.com/communications
DSP	dsp.ti.com	Computers and Peripherals	www.ti.com/computers
Clocks and Timers	www.ti.com/clocks	Consumer Electronics	www.ti.com/consumer-apps
Interface	interface.ti.com	Energy	www.ti.com/energy
Logic	logic.ti.com	Industrial	www.ti.com/industrial
Power Mgmt	power.ti.com	Medical	www.ti.com/medical
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Space, Avionics & Defense	www.ti.com/space-avionics-defense
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video and Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless-apps

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