TMS320C6457 DSP
Universal Test and Operations PHY Interface for ATM 2 (UTOPIA2)

User's Guide

Literature Number: SPRUGL1
March 2009

Contents

List of Figures

List of Tables

Read This First

About This Manual

This document describes the universal test and operations PHY interface for asynchronous transfer mode (ATM) 2 (UTOPIA2) in the TMS320C6457 digital signal processors (DSPs) of the TMS320C6000™ DSP family.

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 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

The following documents describe the C6000 devices and related support tools. 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.

SPRU189 — *TMS320C6000 DSP CPU and Instruction Set Reference Guide.* Describes the CPU architecture, pipeline, instruction set, and interrupts for the TMS320C6000 digital signal processors (DSPs).

SPRU198 — *TMS320C6000 Programmer's Guide.* Describes ways to optimize C and assembly code for the TMS320C6000™ DSPs and includes application program examples.

SPRU301 — *TMS320C6000 Code Composer Studio Tutorial.* Introduces the Code Composer Studio™ integrated development environment and software tools.

SPRU321 — *Code Composer Studio Application Programming Interface Reference Guide.* Describes the Code Composer Studio™ application programming interface (API), which allows you to program custom plug-ins for Code Composer.

SPRU871 — *TMS320C64x+ DSP Megamodule Reference Guide.* Describes the TMS320C64x+ digital signal processor (DSP) megamodule. Included is a discussion on the internal direct memory access (IDMA) controller, the interrupt controller, the power-down controller, memory protection, bandwidth management, and the memory and cache.

Trademarks

TMS320C6457 UTOPIA2

1 Overview

The UTOPIA is an ATM controller (ATMC) slave device that interfaces to a master ATM controller. The UTOPIA port conforms to the ATM Forum standard specification af-phy-0039.000. Specifically, this interface supports the UTOPIA level 2 interface that allows 8-bit slave operations of up to 50 MHz for both transmit and receive operations.

The UTOPIA slave interface relies on the master ATM controller to provide the necessary control signals such as the clock, enable, and address values. Only cell-level handshaking is supported.

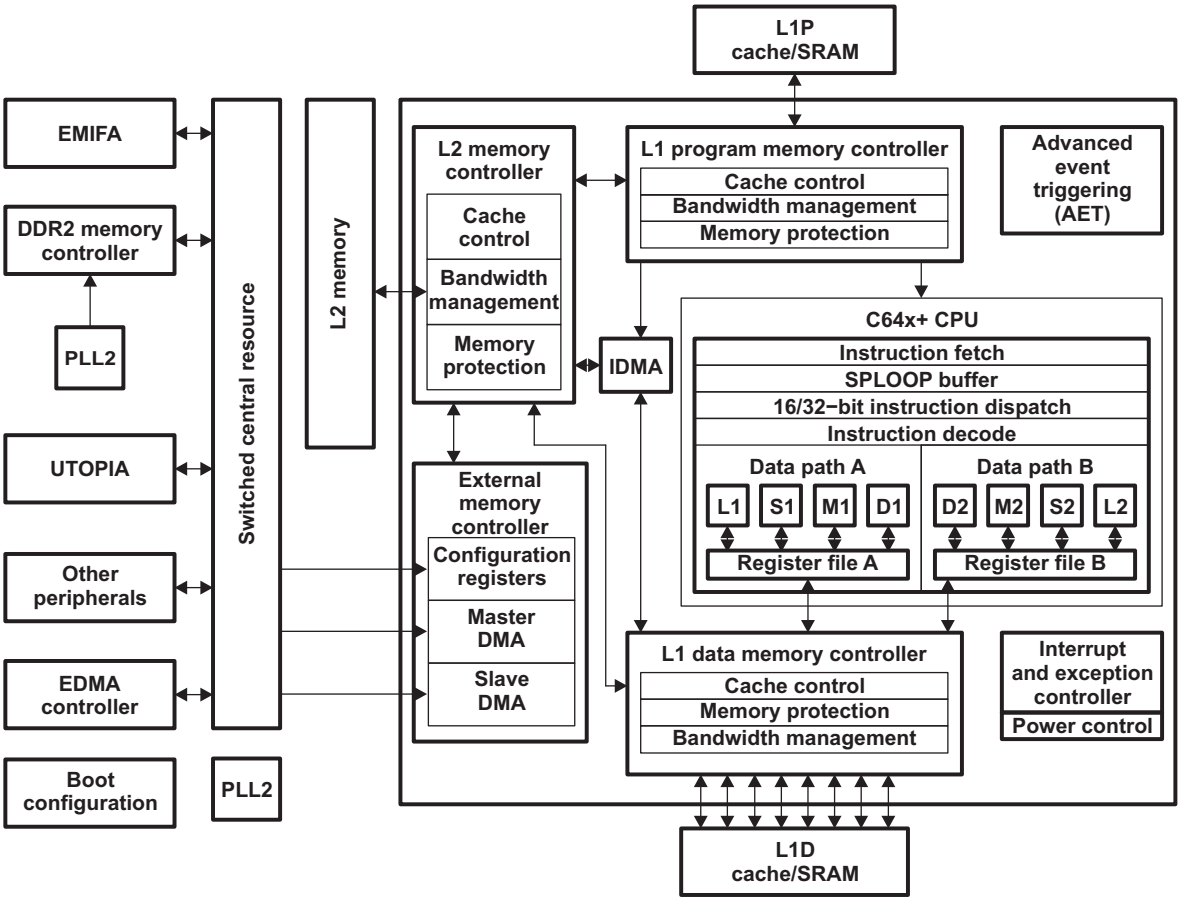
The enhanced DMA (EDMA) controller services the UTOPIA. The ATM adaptation layer (AAL) is commonly called, as segmentation and reassembly (SAR) functions should be performed in software.

All references to the term slave devices are analogous to multi-PHYs (MPHYs) as referenced in the ATM Forum specification. For MPHY systems, refer to the ATM Forum standard specification af-phy-0039.000. For single-PHY (single device) systems, refer to the ATM Forum standard specification af-phy-0017.000.

Figure 1 shows the TMS320C6457 block diagram.

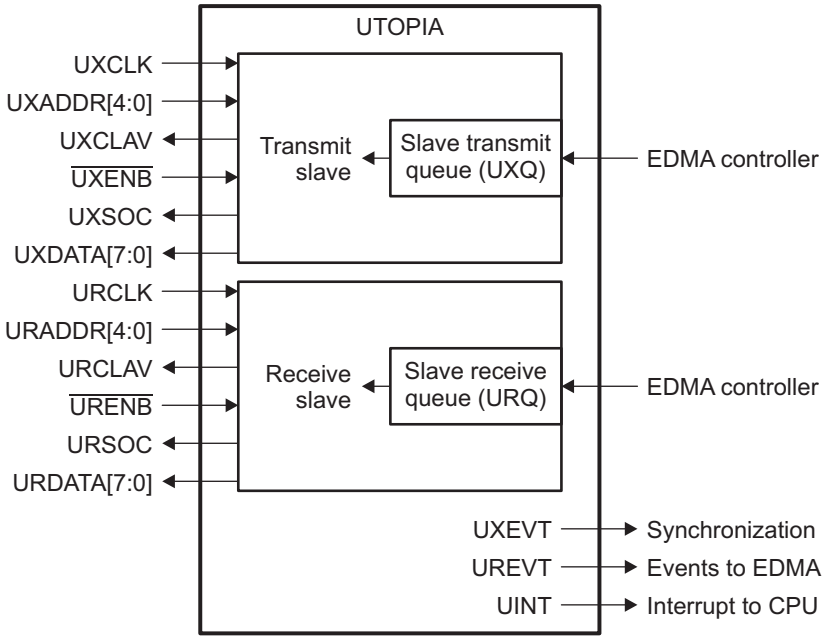
The UTOPIA slave consists of the transmit interface and the receive interface as shown in Figure 2. The interface signals are described in Section 3.1. The UTOPIA sends synchronization events to the EDMA controller through the UXEVT and UREVT signals. An interrupt signal (UINT) is also generated to the CPU to communicate error conditions (see Section 7).

Figure 1. TMS320C6457 DSP Block Diagram



Note: For your peripheral set, see the *TMS320C6457 Fixed-Point Digital Signal Processor* data manual (SPRS582).

Figure 2. UTOPIA Block Diagram

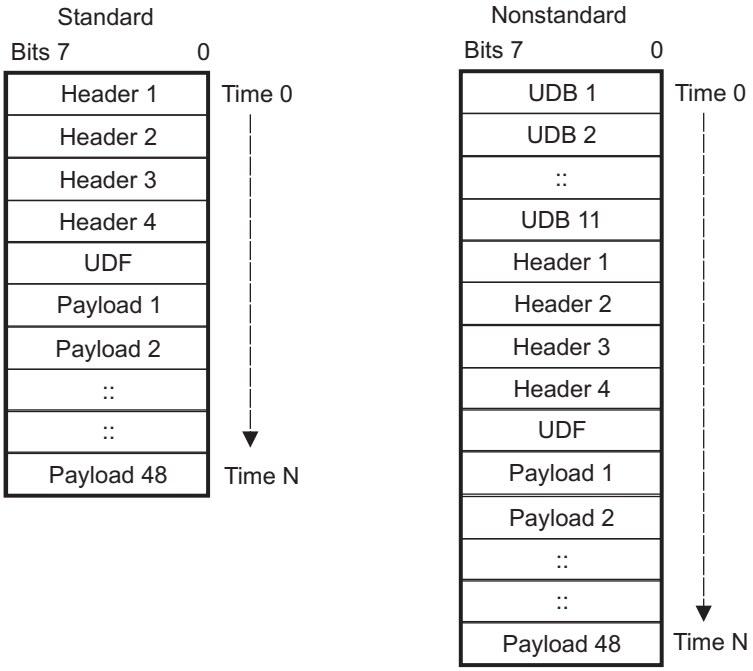


2 Cell Transfer Format

The ATM Forum specification for UTOPIA level 2 specifies the order in which header and payload information is sent across the ATM-PHY interface. The header information is sent first, followed by the 48-byte payload. A standard ATM cell is 53 bytes (5-byte header + 48-byte payload), as shown in Figure 3. The UTOPIA also supports a nonstandard ATM cell of size 54 to 64 bytes (R/XUDC = 1 to 11 + 5-byte header + 48-byte payload), as shown in Figure 3. The UTOPIA transmit queue and receive queue each accommodate two cells. The number of cells each queue accommodates is not dependent upon cell size.

For the TMS320C6457 DSP, each ATM cell must be aligned on a word-boundary. Therefore, each ATM cell (53 bytes) in the DSP memory (internal or external) and in the UTOPIA transmit/receive queues is padded with dummy bytes, as necessary, before the ATM header. The standard 56-byte cell-packet consists of the 53-byte ATM cell, plus 3 bytes of dummy data before the ATM header. This 56-byte packet is referred to as a cell packet. For more details, see Section 8.

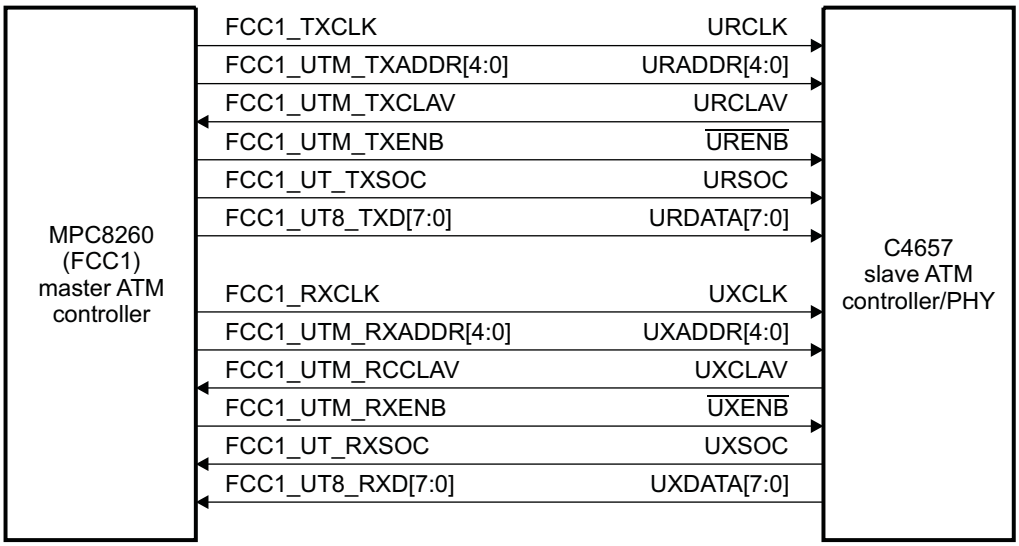
Figure 3. Cell Transfer Formats for 8-Bit Mode



3 UTOPIA Slave as ATM Controller

The UTOPIA interface is used as an ATM controller slave in either a single-PHY or multi-PHY (MPHY) configuration. As a slave, the clock, address, and enable signals of the transmit and receive interfaces are driven by the master. An example configuration is shown in Figure 4.

Figure 4. UTOPIA Slave Interfaced to Motorola MPC8260 Power QUICC IIE™ Master in 8-Bit Mode



3.1 UTOPIA Slave Pins

As a slave device in an ATM system, the UTOPIA performs all ATM cell transfers when directed by the master. The clock, address, and enable signals are inputs. The master can configure the slave's address in the UTOPIA control register (UCR) through the HPI/PCI interface. Table 1 and Table 2 show the pins and their direction in relation to the UTOPIA slave interface.

The slave responds when it detects its assigned address on the address bus by asserting its UXCLAV or URCLAV signal, if a cell is available for transmit or receive, respectively. If the slave does not have a cell to transmit or cell space to receive, it does not assert the relevant CLAV signal, but the master continues to poll the remaining slaves/PHYs in the system on the address bus.

Table 1. UTOPIA Transmit Interface Slave Pin Descriptions

Pin	Direction	Value	Description
UXCLK	In		UTOPIA Transmit Clock. An input driven by the master in the system. Transmit data and transmit control signals are synchronous to this clock.
UXADDR[4:0]	In		5-bit address input driven by the master ATM controller to identify each of the slave devices (up to 31) in the ATM system.
UXCLAV	Out		Transmit Cell Available status output signal of the slave. For cell-level handshake, the following is true:
		0	Indicates that the slave does not have a complete cell available for transmit.
		1	Indicates that the slave has a complete cell available to transmit.
$\overline{\text{UXENB}}$	In		UTOPIA Transmit Interface Enable input signal. Asserted active low by the master to indicate that the slave should put first byte of valid data and assert SoC signal in the next clock cycle.
UXSOC	Out		Transmit Start-Of-Cell signal (active high) output by the slave on rising edge of UXCLK to indicate that the first valid byte of the cell is available on the Transmit Data Bus UXDATA[7:0].
UXDATA[7:0]	Out		8-bit Transmit Data Bus. Slave transmits ATM cells to the master using this bus on rising edge of UXCLK.

Table 2. UTOPIA Receive Interface Slave Pin Descriptions

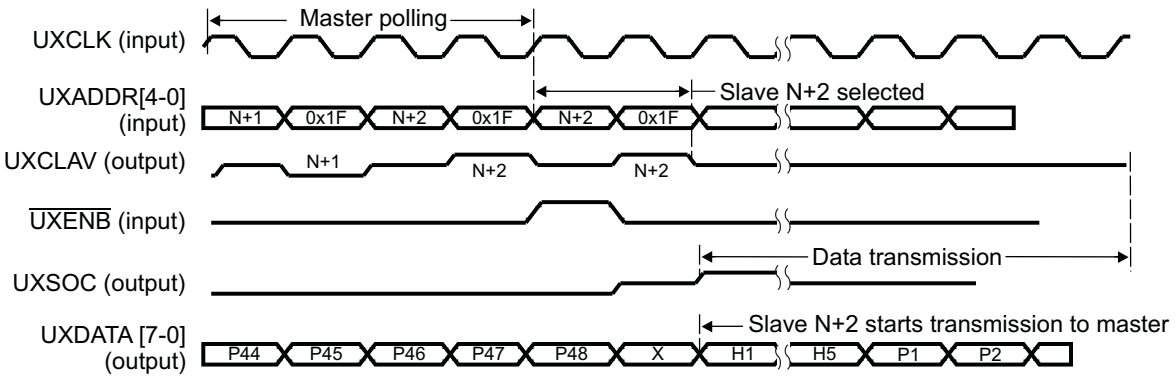
Pin	Direction	Value	Description
URCLK	In		UTOPIA Receive Clock is an input signal driven by the ATM master. Receive data and control signals are sampled and synchronous to this clock.
URADDR[4:0]	In		5-bit address bus input driven by the master to select a slave.
URCLAV	Out		Receive Cell Available status signal is an output from the slave to indicate that it has space available to receive a cell from the master. For cell-level handshake, the following is true:
		0	No space is available to receive a cell from the master.
		1	Space is available to receive a cell from the master.
$\overline{\text{URENB}}$	In		UTOPIA Receive Interface Enable. An active-low signal driven by the master to enable the receive interface of the slave. It tells the slave to sample Receive Data and SoC signals in the next clock cycle or thereafter.
URSOC	In		Receive Start-Of-Cell signal driven by the master to indicate that the first valid byte of the cell is available on the Receive Data Bus for the slave to sample.
URDATA[7:0]	In		8-bit UTOPIA Receive Data Bus. Data from the master is received on this bus. Data is sampled on the rising edge of URCLK.

3.2 Slave-Transmit Operation

The UTOPIA slave-transmit block consists of a UTOPIA level 2 pin interface that interfaces internally to the slave-transmit queue. Figure 2 shows the UTOPIA slave-transmit block diagram. The slave-transmit queue is accessed through the UXQ data port. The EDMA controller services the slave-transmit queue with 32-bit writes when a transmit event is generated by the UTOPIA transmit section.

When the UTOPIA slave interface detects its address on the transmit address bus, UXADDR[4:0], it drives the UXCLAV signal to indicate to the master that a cell is or is not available for transmit. In the following cycles, when the master chooses (after completion of any ongoing data transfers) to receive the data from this UTOPIA slave, the master asserts the slave address along with the enable signal, $\overline{\text{UXENB}}$. Next, the slave starts transmitting the data on its UXDATA[7:0] pins, by asserting the start-of-cell signal, UXSOC. Figure 5 shows the UTOPIA slave-transmit interface timing. The clock for the UTOPIA slave-transmit interface, UXCLK, is an input driven by the external master.

Figure 5. ATM Controller Slave Transmit Timing Diagram



3.3 Slave-Transmit Queue

The slave-transmit queue prepares the UTOPIA interface to be ready to transmit data whenever the master requests a transmit. The slave-transmit queue generates a transmit event (UXEVT) when it is not full. This transmit event triggers the EDMA controller to perform 32-bit writes to the slave-transmit queue. A total of 14 word writes are required to fill one standard ATM cell packet in the queue.

As soon as the first write to the queue occurs, the transmit event is cleared. The next transmit event is generated if the queue is not full. This allows the EDMA controller to begin the next cell-packet write without having to wait for the current cell to be fully written to the queue. This process repeats as described below.

The transmit event is generated and cleared as follows:

1. UXEVT is generated when the queue is not full. The queue is not full when there is space for at least one cell packet (56B).
2. UXEVT is cleared when the first write (by the EDMA controller) of that cell occurs.
3. UXEVT is regenerated immediately (without waiting for the previous cell to be fully written) if the queue is not full.
4. Go to step 2.

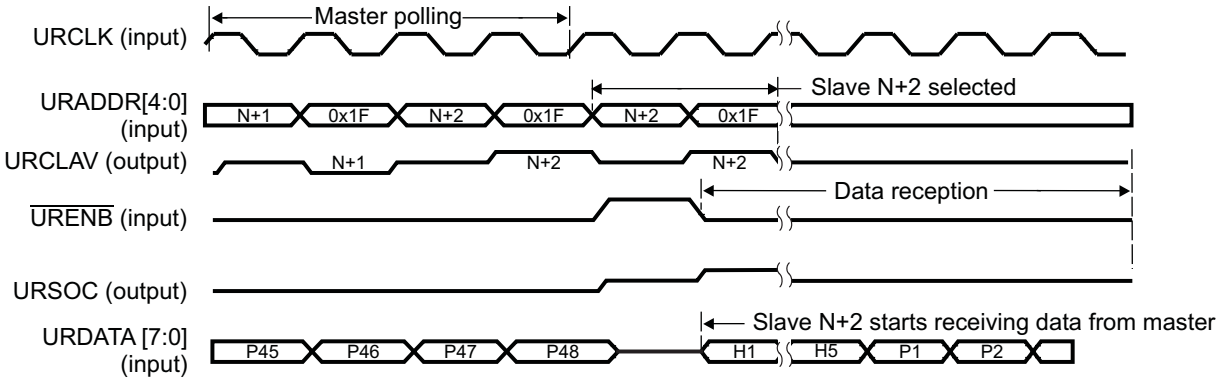
The UTOPIA slave indicates the availability of a cell to transmit to the master by asserting its UXCLAV signal when there is at least one cell available in the slave-transmit queue. If the slave cannot provide the next cell in a contiguous fashion, it de-asserts its UXCLAV signal in the cycle following the completion of the current cell transmission. The UXCLAV signal remains asserted if the slave has another cell available to transfer to the master. The master can disable RXENB on its side (connected to the UXENB pin for this ATMC slave), which causes the UTOPIA slave to hold off the next cell transfer until the master commands the transfer.

3.4 Slave-Receive Operation

The UTOPIA slave-receive block consists of a UTOPIA level 2 pin interface that interfaces internally to a slave-receive queue. The UTOPIA slave-receive block diagram is shown in Figure 2. The slave-receive queue is accessed through the URQ data port. The EDMA controller services the slave-receive queue with 32-bit reads when a receive event is generated by the UTOPIA receive section.

When the master polls for slaves in the system that can receive its cells, the UTOPIA slave responds with an active cell-available signal on its URCLAV pin if it has space in the slave-receive queue to receive a complete cell. The master transmits to this slave or continues to poll to find a suitable slave for its data. The UTOPIA slave responds to its assigned address by asserting its appropriate URCLAV state a cycle after its address is detected on the receive address bus, URADDR[4:0]. The master then outputs the slave address that has an active RCLAV signal and also provides the enable signal ($\overline{\text{URENB}}$ on slave) to enable slave-receive operation. The UTOPIA receive slave starts receiving data in the cycle when the master asserts its start-of-cell signal, URSOC. The bytes are assembled into words and written into the slave-receive queue. Figure 6 shows the UTOPIA slave-receive interface timing. The clock for the UTOPIA slave-receive interface, URCLK, is an input driven by the external master.

Figure 6. ATM Controller Slave Receive Timing Diagram



3.5 Slave-Receive Queue

When the master initiates the transfers to the slave, the slave-receive queue generates a receive event (UREVT) to the EDMA controller when at least one cell worth of data is available. As soon as the first read is performed by the EDMA controller, the receive event is cleared. The next receive event is generated when the next cell is fully available and the process repeats.

The receive event is generated and cleared as follows:

1. UREVT is generated when a complete cell is available.
2. UREVT is cleared when the first read (by the EDMA controller) of that cell occurs.
3. UREVT is regenerated when the next complete cell is available in the slave-receive queue.
4. Go to Step 1.

The UTOPIA slave agrees for reception from the master by asserting its URCLAV signal when there is at least one cell space available in the slave-receive queue. If the slave cannot receive the next cell immediately, it de-asserts its URCLAV signal at least four URCLK cycles before the end of this cell transfer. If it remains asserted, it indicates that the slave can receive another cell from the master. The master can disable TXENB on its side ($\overline{\text{URENB}}$ for this ATMC slave) at its discretion.

3.6 UTOPIA Events Generation

The UTOPIA transmit and receive queues generate not-full and not-empty events to the EDMA controller. The events are generated when the queues have space available for at least one cell and not when the queues are completely full or empty. This allows for more throughput and better performance because the transmit and receive data are continuously transferred without having to wait for a full or empty queue. For details on the generation of these events, see Section 3.3 and Section 3.5. The EDMA controller services the UTOPIA in response to these events, as discussed in Section 4.

3.7 Multi-PHY (MPHY) Operation

The UTOPIA interface supports multi-PHY operation as per UTOPIA level 2 specification. The MPHY mode is enabled when the MPHY bit in the UTOPIA control register (UCR) is set to 1 (default state). In MPHY mode, the slave ID (SLID) bit in UCR indicates the PHY address of the UTOPIA. Either the DSP or the external master programs the SLID bits. MPHY operation is based on cell-level handshaking.The programming interface can either be the HPI or the PCI.

As shown in Figure 5 and Figure 6, the external ATM master polls for available slave devices before the beginning of the actual data transaction. The UTOPIA output signals URCLAV, UXCLAV, UXSOC, and UXDATA[7:0] are in high-impedance state when the UTOPIA slave is not selected by the master. When the UTOPIA slave detects its address at the UXADDR[4:0] or URADDR[4:0] pins, it asserts the UXCLAV or URCLAV signal, respectively.

When used in single-PHY mode (MPHY = 0 in UCR), there is no need to program the address.

4 EDMA Controller Servicing UTOPIA

The EDMA controller is used to service the UTOPIA interface. Table 3 lists the UTOPIA synchronization events to the EDMA controller and their corresponding EDMA channel.

Table 3. EDMA Synchronization Events from UTOPIA

EDMA Event	EDMA Channel	Synchronization Event Description
UXEVT	42	Transmit event from the UTOPIA to the EDMA controller. UXEVT is asserted if the transmit queue has space for at least one cell packet.
UREVT	30	Receive event from the UTOPIA to the EDMA controller. UREVT is asserted if a complete cell packet is available in the receive queue.

4.1 EDMA Setup for UTOPIA Transmitter

The UTOPIA transmitter generates a UXEVT synchronization event to the EDMA controller when at least one cell-packet space is available in the slave-transmit queue. EDMA event 40 is dedicated to the UXEVT event. Per the UXEVT synchronization event, one frame of cell-packet data is transferred to the slave-transmit queue. A standard cell-packet consists of 14 words (56 bytes), while a nonstandard cell-packet consists of 14, 15, or 16 words (56, 60, or 64 bytes). The EDMA access to the UTOPIA is always 32 bits. The EDMA source address should point to the UTOPIA source buffer in the DSP memory (internal or external). The EDMA destination address should point to the slave-transmit queue data port UXQ.

Example 1 provides a code example of an EDMA setup to service a UTOPIA transmitter for a single cell-packet.

Example 1. EDMA3 Setup for UTOPIA Transmitter Code Example

```
//************************************************\
* EDMA Channel Configuration
\************************************************/
void Channel_Setup(Uint32 srcTxAddr,Uint32 TxQue_Addr,Uint32 RxQue_Addr,Uint32 dstRxAddr,Uint32
TxACnt,Uint32 RxACnt, Uint32 TxBCnt,Uint32 RxBCnt,Uint32 TxCCnt,Uint32 RxCCnt,Int32
TxBIdx,Int32 RxBIdx ,Int32 TxCIdx,Int32 RxCIdx){

    chParamTx.regionNum = CSL_EDMA3_REGION_GLOBAL;
    chParamTx.chaNum = CSL_EDMA3_CHA_UXEVT;   //CSL_EDMA3_CHA_UXEVT
    hChTx = CSL_edma3ChannelOpen(&chObjTx,
                                 CSL_EDMA3,
                                 &chParamTx,
                                 &Status);

    CSL_edma3HwChannelSetupQue (hChTx, CSL_EDMA3_QUE_2);
    CSL_edma3HwChannelSetupParam (hChTx, PARAM_NUMTx);

paramSetupTx.option =
CSL_EDMA_OPT_MAKE(
FALSE,      /* ITCCEN = 0 (Intermediate transfer complete chaining) */
FALSE,      /* TCCEN = 0 (Transfer complete chaining */
FALSE,      /* ITCINTEN = 0 (Intermediate transfer complete interrupt */
TRUE,         /* TCINTEN = 1 (Transfer complete interrupt */
40,         /* TCC = 1 to 63 (Transfer Complete Code)*/
CSL_EDMA_TCC_NORMAL, /* TCCMODE = 0 ( Normal Completion) */
CSL_EDMA_FIFOWIDTH_NONE, /* FWid = NONE(Fifo Width) */
FALSE,       /* STAT = 0 (Entry is updated as normal) */
CSL_EDMA3CC_OPT_SYNCDIM_ASYNC, /* SYNCDIM = 0 (A-Sync. Each event triggers
                                  the transfer ACNT elements) */
CSL_EDMA_ADDRMODE_INCR,    /* DAM = 0 (Dst addressing within an array
                           increments. Dst is not a FIFO) */
CSL_EDMA_ADDRMODE_INCR     /*SAM = 0 (Src addressing within an array
                           increments. Source is not a FIFO) */
);
paramSetupTx.srcAddr = srcTxAddr;
                    /* Source Address : start address of the source buffer
                    (must be double-word aligned) */
paramSetupTx.aCntbCnt = CSL_EDMA_CNT_MAKE(TxACnt, TxBCnt);
                    /* ACNT = 56 (Number of bytes in an array, should be equal
                    to number of bytes in one receive cell) It can have only
                    values 56, 60 and 64 */
                    /* BCNT = 1 (Number of arrays of length ACNT, should be
                    equal to number of cell-packets to be transferred) */
paramSetupTx.dstAddr = TxQue_Addr;
                    /* Utopia Transmit Queue Base Address (0x3D000400)/*
paramSetupTx.srcDstBidx = CSL_EDMA_BIDX_MAKE(TxBIdx, 0);
                    /* SRCBIDX = TxBIdx = ACNT (Source 2nd Dimension Index)*/
                    /* DSTBIDX = 0 (Destination 2nd Dimension Index) */
paramSetupTx.linkBcntrld = CSL_EDMA_LINKBCNTRLD_MAKE(CSL_EDMA_LINK_NULL, 0);
paramSetupTx.srcDstCidx = CSL_EDMA_CIDX_MAKE(TxCIdx, TxCIdx);
                   / * SRCCCIDX = TxCIdx = 0 (Source 3rd Dimension Index)*/
                   /* DSTCIDX = 0 (Destination 3rd Dimension Index) */
paramSetupTx.cCnt = TxCCnt; /* CCNT = 1 (Number of frames in a block)*/
hParamTx = CSL_edmaGetParamHandle(hChTx,PARAM_NUMTx,NULL);
CSL_edmaParamSetup(hParamTx,&paramSetupTx);
```

Example 1. EDMA3 Setup for UTOPIA Transmitter Code Example (continued)

```
/* RECIEVE CHANNEL SETUP */
    chParamRx.regionNum = CSL_EDMA3_REGION_GLOBAL;
    chParamRx.chaNum = CSL_EDMA3_CHA_UREVT;
    hChRx = CSL_edma3ChannelOpen(&chObjRx,
                                  CSL_EDMA3,
                                  &chParamRx,
                                 &Status);

    CSL_edma3HwChannelSetupQue (hChRx, CSL_EDMA3_QUE_2);
    CSL_edma3HwChannelSetupParam (hChRx, PARAM_NUMRx);

paramSetupRx.option = CSL_EDMA_OPT_MAKE (
FALSE, /* ITCCEN = 0 (Intermediate transfer complete chaining) */
FALSE, /* TCCEN = 0 (Transfer complete chaining */
FALSE, /* ITCINTEN = 0 (Intermediate transfer complete interrupt */
TRUE,  /* TCINTEN = 1 (Transfer complete interrupt */
32,    /* TCC = 1 to 63 (Transfer Complete Code)*/
CSL_EDMA_TCC_NORMAL, /* TCCMODE = 0 ( Normal Completion) */
CSL_EDMA_FIFOWIDTH_NONE, /* FWid = NONE(Fifo Width) */
FALSE,                   /* STAT = 0 (Entry is updated as normal) */
CSL_EDMA3CC_OPT_SYNCDIM_ASYNC, /* SYNCDIM = 0 (A-Sync. Each event triggers
                                  the transfer ACNT elements) */
CSL_EDMA_ADDRMODE_INCR, /* DAM = 0 (Dst addressing within an array
                           increments. Dst is not a FIFO) */
CSL_EDMA_ADDRMODE_INCR /*SAM = 0 (Src addressing within an array
                           increments. Source is not a FIFO) */
);


    paramSetupRx.srcAddr = RxQue_Addr;
    paramSetupRx.aCntbCnt = CSL_EDMA3_CNT_MAKE(RxACnt, RxBCnt);
    paramSetupRx.dstAddr = dstRxAddr;
    paramSetupRx.srcDstBidx = CSL_EDMA3_BIDX_MAKE(0, RxBIdx);
    paramSetupRx.linkBcntrld = CSL_EDMA3_LINKBCNTRLD_MAKE(CSL_EDMA3_LINK_NULL, 0);
    paramSetupRx.srcDstCidx = CSL_EDMA3_CIDX_MAKE(RxCIdx, RxCIdx);
    paramSetupRx.cCnt = RxCCnt;
    hParamRx = CSL_edma3GetParamHandle(hChRx,PARAM_NUMRx,&Status);
    CSL_edma3ParamSetup(hParamRx,&paramSetupRx);
}
}
```

4.2 EDMA Setup for UTOPIA Receiver

The UTOPIA receiver generates a UREVT synchronization event to the EDMA controller when the slave-receive queue has space for at least one cell-packet. EDMA event 32 is dedicated to the UREVT event. Per UREVT synchronization event, one frame of cell-packet data is read from the slave-receive queue by way of the data port URQ. A standard cell-packet consists of 14 words (56 bytes), while a nonstandard cell-packet consists of 14, 15, or 16 words (56, 60, or 64 bytes). The EDMA destination address should point to the destination buffer in the DSP memory (internal or external).

Example 2 shows a code example of an EDMA3 setup to service a UTOPIA receiver for a single cell-packet.

Example 2. EDMA3 Setup for UTOPIA Receiver Code Example

```
//************************************************\
* EDMA RX Channel Configuration
\************************************************/
void RXChannelParamSetup()
{
CSL_EDMA_OPT_MAKE(paramSetupRx.option
    FALSE,            /* ITCCEN = 0 (Intermediate transfer complete
                      chaining) */
    FALSE,            /* TCCEN = 0 (Transfer complete chaining */
    FALSE,            /* ITCINTEN = 0 (Intermediate transfer
                      complete interrupt */
    TRUE,             /* TCINTEN = 1 (Transfer complete interrupt */
    CSL_EDMA_CHA32,   /* TCC = 1 to 63 (Transfer Complete Code)*/
    CSL_EDMA_TCC_NORMAL, /* TCCMODE = 0 ( Normal Completion) */
    CSL_EDMA_FIFOWIDTH_NONE, /* FWid = NONE(Fifo Width) */
    FALSE,            /* STAT = 0 (Entry is updated as normal) */
    CSL_EDMACC_OPT_SYNCDIM_ASYNC, /* SYNCDIM = 0 (A-Sync. Each event triggers
                                  the transfer ACNT elements) */
    CSL_EDMA_ADDRMODE_INCR, /* DAM = 0 (Dst addressing within an array
                            increments. Dst is not a FIFO) */
    CSL_EDMA_ADDRMODE_INCR  /*SAM = 0 (Src addressing within an array
                            increments. Source is not a FIFO) */
    );
paramSetupRx.srcAddr = UTOPIA_RXQ_ADDR
                       /* UTOPIA Receive Queue base Address = (0x3C00 0000h); */
                       /* srcRxAddr */
paramSetupRx.aCntbCnt = CSL_EDMA_CNT_MAKE(RxACnt, RxBCnt);
                       /* RxAcnt = 56 (Number of bytes in an array, should be
                       equal to number of bytes in one receive cell) It can have
                       only values 56, 60 and 64. */
                       /* RxBCnt = 1 (Number of arrays of length ACNT, should be
                       equal to number of cell-packets to be transferred) */
paramSetupRx.dstAddr = dstRxAddr
                       /* DESTINATION ADDRESS: start address of the destination
                       array (must be double-word aligned) */
paramSetupRx.srcDstBidx = CSL_EDMA_BIDX_MAKE(RxBIdx, 0);
                       /* RxBIdx = 0 (Source 2nd Dimension Index)*/
                       /* DstBIdx = RxBIdx = ACNT (Destination 2nd Dimension
                       Index) */
paramSetupRx.linkBcntrld = CSL_EDMA_LINKBCNTRLD_MAKE(CSL_EDMA_LINK_NULL, 0);
paramSetupRx.srcDstCidx = CSL_EDMA_CIDX_MAKE(RxCIdx, RxCIdx);
                       / * SRCCCIdx = 0 (Source 3rd Dimension Index)*/
                       /* DSTCIdx = 0 (Destination 3rd Dimension Index) */
paramSetupRx.cCnt = RxCCnt; /* RxCCNT = 1 (Number of frames in a block)*/
paramSetupRx.triggerWord = CSL_EDMA_TRIGWORD_NONE;
hParamRx = CSL_edmaGetParamHandle(hChRx,PARAM_NUMTx,NULL);
CSL_edmaParamSetup(hParamRx,&paramSetupRx,CSL_EDMA_PARAM_BASIC);
}
```

5 CPU Servicing UTOPIA

The CPU should not be used to service the UTOPIA peripheral on the C6457 device.

6 UTOPIA Clocking and Clock Detection

The transmit and receive clock for the UTOPIA interface is supplied by an external clock source such as the master ATM controller. This allows for accurate clocks as required by most applications. Internal to the DSP, the UTOPIA registers and queues are synchronized to the DSP peripheral clock at a CPU/6 rate.

Clock detection logic has been implemented to facilitate proper reset of the UTOPIA interface when the clock(s) is removed from the system. It uses a user-programmable Clock Detect register to trigger the clock detection based on the (R/X) CCNT bit-field parameters. Receive and transmit logic are independent of each other.

7 Special Transfer Conditions

This section explains how the UTOPIA slave interface handles some of the error conditions.

- **Absence of UTOPIA clocks**: If, for any reason during a cell transfer in either direction, the receive clock (URCLK) or transmit clock (UXCLK) stop toggling, the corresponding section of the UTOPIA may be reset depending on the duration of absence. The queues are returned to their reset state, and all control registers are reset. In addition, the receive clock failed (RCFP) or transmit clock failed (XCFP) bit, respectively, is set in the error interrupt pending register (EIPR). An interrupt UINT is generated to the CPU if the corresponding bits in the error interrupt enable register (EIER) are set. This helps when recovering from conditions where the master card (that supplies the clocks) is pulled from the system.

  The receive clock present (RCPP) and transmit clock present (XCPP) bits in EIPR are set if the URCLK and UXCLK are detected, regardless of the state of the UTOPIA port. If the corresponding bits are enabled in EIER, an interrupt UINT is generated. This is useful in reenabling the UTOPIA ports once the UTOPIA clocks are detected.
- **Abrupt reset**: In slave mode, typically the master issues a reset command through the management interface to ensure a graceful stop of transfers. But if this is violated, data corruption occurs and the system software should comprehend this. In short, abrupt reset causes data loss/corruption. It is your responsibility to avoid such conditions.
- **FIFO read/write stall conditions**: There are two potential stall conditions when a read or a write to the transmit/receive queues is attempted when the transmit/receive queues are not ready (when UXEVT and UREVT are not active):
  - Writes to a FULL slave-transmit queue: Writing to the transmit queue that is full renders the queue not ready. In other words, writes are stalled until the queue is drained and space is available for further writes. Therefore, data is not overwritten. When such a condition occurs, the transmit queue stall interrupt pending (XQSP) status bit in EIPR is set to indicate the stall condition. The XQSP bit is a read-only bit and is cleared once the queue has space available, and writes can continue.
  - Reads from an EMPTY slave-receive queue: Attempting to read a queue that has no data results in stalling that operation until valid data is available. This also sets the receive queue stall interrupt pending (RQSP) bit in EIPR to indicate the read stall condition. This is a user error, because a read access is performed when there is no active UREVT. The RQSP bit is a read-only bit and is cleared as soon as valid data is available in the receive queue and the read is performed.

8 Endian Considerations

For 8-bit operation, shown in Table 5 through Table 10, bytes are assembled into words in the UTOPIA queues. Device endian configuration is selected during reset. Pin-level endian configuration ensures the desired endian mode for the DSP/CPU. For the UTOPIA interface to present the data transferred across its interface to the DSP in accordance with the device endian mode, the big-endian mode enable bit (BEND) in the UTOPIA control register (UCR) has to be programmed. By default/reset, the UTOPIA data is presented to the DSP in little-endian format (BEND = 0). If big-endian is preferred, the BEND bit should be programmed to 1. The data bytes are swapped in hardware based on the BEND value.

When the DSP is a UTOPIA slave in a system, it only communicates to the master. Therefore, the communication from the slave's perspective is always point-to-point. The cell-packet transfer formats are shown in Figure 3.

Depending on the user-defined cell (RUDC and XUDC bits) and endian format (BEND bit), bytes are placed in the transmit and receive queues as shown in Table 5 through .

The cell-packet formats in Table 5 through indicate the data stored in the DSP memory (internal or external) and in the transmit/receive queues. Only the ATM data, including the header and payload information but not the Dummy bytes, is actually sent or received across the UTOPIA pins.

Table 4. 8-Bit UTOPIA Slave Modes Depending on BEND, XUDC, and RUDC Bits

	UCR Bits			
Endian Mode	**BEND bit**	**XUDC value**	**RUDC value**	**See**
Little endian	0	0 (disabled)	0 (disabled)	Table 5
Big endian	1	0 (disabled)	0 (disabled)	Table 6
Big endian	1	1	1	Table 7
Little endian	0	7	7	Table 8
Little endian	0	11	11	Table 9
Big endian	1	11	11	Table 10

Table 5. 8-Bit UTOPIA Slave in Little-Endian Mode (BEND = 0) With RUDC/XUDC = 0

Queue Bits	**31-24**	**23-16**	**15-8**	**7-0**	**Order**
Address n	Header 1	Dummy	Dummy	Dummy	Word 0
Address n+4	UDF	Header 4	Header 3	Header 2	Word 1
Address n+8	Payload 4	Payload 3	Payload 2	Payload 1	:
::	::	::	::	::	:
Address n+48	Payload 44	Payload 43	Payload 42	Payload 41	:
Address n+52	Payload 48	Payload 47	Payload 46	Payload 45	Word 13

Table 6. 8-Bit UTOPIA Slave in Big-Endian Mode (BEND = 1) With RUDC/XUDC = 0

Queue Bits	**31-24**	**23-16**	**15-8**	**7-0**	**Order**
Address n	Dummy	Dummy	Dummy	Header 1	Word 0
Address n+4	Header 2	Header 3	Header 4	UDF	Word 1
Address n+8	Payload 1	Payload 2	Payload 3	Payload 4	:
::	::	::	::	::	:
Address n+48	Payload 41	Payload 42	Payload 43	Payload 44	:
Address n+52	Payload 45	Payload 46	Payload 47	Payload 48	Word 13

Table 7. 8-Bit UTOPIA Slave in Big-Endian Mode (BEND = 1) With RUDC/XUDC = 1

Queue Bits	31-24	23-16	15-8	7-0	Order
Address n	Dummy	Dummy	UDB 1	Header 1	Word 0
Address n+4	Header 2	Header 3	Header 4	UDF	Word 1
Address n+8	Payload 1	Payload 2	Payload 3	Payload 4	:
::	::	::	::	::	:
Address n+48	Payload 41	Payload 42	Payload 43	Payload 44	:
Address n+52	Payload 45	Payload 46	Payload 47	Payload 48	Word 13

Table 8. 8-Bit UTOPIA Slave in Little-Endian Mode (BEND = 0) With RUDC/XUDC = 7

Queue Bits	31-24	23-16	15-8	7-0	Order
Address n	UDB 4	UDB 3	UDB 2	UDB 1	Word 0
Address n+4	Header 1	UDB 7	UDB 6	UDB 5	Word 1
Address n+8	UDF	Header 4	Header 3	Header 2	:
Address n+12	Payload 4	Payload 3	Payload 2	Payload 1	:
::	::	::	::	::	:
Address n+56	Payload 48	Payload 47	Payload 46	Payload 45	Word 14

Table 9. 8-Bit UTOPIA Slave in Little-Endian Mode (BEND = 0) With RUDC/XUDC = 11

Queue Bits	31-24	23-16	15-8	7-0	Order
Address n	UDB 4	UDB 3	UDB 2	UDB 1	Word 0
Address n+4	UDB 8	UDB 7	UDB 6	UDB 5	Word 1
Address n+8	Header 1	UDB 11	UDB 10	UDB 9	Word 2
Address n+12	UDF	Header 4	Header 3	Header 2	:
Address n+16	Payload 4	Payload 3	Payload 2	Payload 1	:
::	::	::	::	::	:
Address n+60	Payload 48	Payload 47	Payload 46	Payload 45	Word 15

Table 10. 8-Bit UTOPIA Slave in Big-Endian Mode (BEND = 1) With RUDC/XUDC = 11

Queue Bits	31-24	23-16	15-8	7-0	Order
Address n	UDB 1	UDB 2	UDB 3	UDB 4	Word 0
Address n+4	UDB 5	UDB 6	UDB 7	UDB 8	Word 1
Address n+8	UDB 9	UDB 10	UDB 11	Header 1	Word 2
Address n+12	Header 2	Header 3	Header 4	UDF	:
Address n+16	Payload 1	Payload 2	Payload 3	Payload 4	
::	::	::	::	::	:
Address n+60	Payload 45	Payload 46	Payload 47	Payload 48	Word 15

9 UTOPIA Reset

The UTOPIA interface is in reset state during device reset. The UTOPIA interface can also be reset through software by programming the UREN and UXEN bits in the UTOPIA control register (UCR) when the device is out of reset. The peripheral reset through software ensures that the state machine is in a known state and there is no activity in the corresponding section. The software reset does not reset registers to their default values.

The UTOPIA pins have no internal pull-up or pull-down resistors; therefore, all input pins should be pulled externally to bring inputs to a known state. The Reset Value column of Table 11 shows the recommended state for the UTOPIA pins during reset. The address pins are pulled high to give the address of a null PHY/slave (address 11111b) during reset. At reset, all outputs are driven to a high-impedance state to facilitate MPHY operation.

Table 11. UTOPIA Pin Reset Values

UTOPIA Pin	ATM Controller SLAVE (Direction)	Reset Value
UXCLK	In	Low
UXADDR[4:0]	In	High
UXCLAV	Out	High impedance
$\overline{\text{UXENB}}$	In	High
UXSOC	Out	High impedance
UXDATA[7:0]	Out	High impedance
URCLK	In	Low
URADDR[4:0]	In	High
URCLAV	Out	High impedance
$\overline{\text{URENB}}$	In	High
URSOC	In	Low
URDATA[7:0]	In	Low

10 UTOPIA Slave Initialization Sequence

A device reset or a programmable reset through the UTOPIA control register (UCR) resets the UTOPIA interface. To initialize the UTOPIA interface for slave operation, the following steps are required:

- The UTOPIA master device in the system provides the clock input to URCLK and UXCLK. The UTOPIA port cannot be initialized without these clocks.
- Ensure that the DSP is out of reset.
- Program the EDMA channel(s) for data transmission and reception to/from the UTOPIA interface.
- Set up the UTOPIA configuration registers, as required.
- Either the DSP or the external ATM master can write the address of this slave/PHY in UCR. The external ATM master can write to UCR through the HPI or PCI interfaces.
- Take the interface out of reset by setting the UREN bit to enable the receive interface and setting the UXEN bit to enable the transmit interface. Note that the transmit and receive interfaces are independent of each other. In typical systems, both are used.

11 ATM Adaptation Layer (AAL) Functions

The UTOPIA interface provides a standard hardware interface between an ATM layer device (master) and a PHY device (slave). The ATM adaptation layer functions such as segmentation and re-assembly (SAR) for AAL2, AAL5 should be implemented in software.

12 Registers

12.1 UTOPIA2 Global Registers

The UTOPIA port is configured through the configuration registers listed in Table 12. The data for transmit and receive queues are accessible through the EDMA controller or CPU at the data port listed in Table 13. For the memory address of these registers, see the device-specific data manual.

Table 12. UTOPIA Configuration Registers

Offset	Acronym	Register Name	See
0000	UCR	UTOPIA Control Register	Section 12.1.1
0014	CDR	Clock Detect Register	Section 12.1.2
0018	EIER	Error Interrupt Enable Register	Section 12.1.3
001C	EIPR	Error Interrupt Pending Register	Section 12.1.4

Table 13. UTOPIA Data Ports

Offset	Acronym	Register Name	See
0000	URQ	UTOPIA Receive Queue	Section 3.4
0004	UXQ	UTOPIA Transmit Queue	Section 3.2

12.1.1 UTOPIA Control Register (UCR)

The UTOPIA interface is configured through the UTOPIA control register (UCR) and contains UTOPIA status and control bits. The UCR is shown in Figure 7 and described in Table 14.

Figure 7. UTOPIA Control Register (UCR)

31	30	29	28	24
BEND	Reserved		SLID	
R/W-0	R-1		R/W-0	

23	22	21	18	17	16
Reserved		XUDC		Reserved	UXEN
R-0		R/W-0		R-0	R/W-0

15	14	13	8
Reserved	MPHY	Reserved	
R-0	R/W-1	R-0	

7	6	5	2	1	0
Reserved		RUDC		Reserved	UREN
R-0		R/W-0		R-0	R/W-0

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

Table 14. UTOPIA Control Register (UCR) Field Descriptions

Bit	Field	Value	Description
31	BEND		Big-endian mode enable bit for data transferred by way of the UTOPIA interface.
		0	Data is assembled to conform to little-endian format.
		1	Data is assembled to conform to big-endian format.
30-29	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
28-24	SLID		Slave ID bits. Applicable in multi-PHY mode (MPHY = 1). This 5-bit value is used to identify the UTOPIA in a MPHY setup. Does not apply to single-PHY slave operation (MPHY = 0).
		0	
		1Fh	Null PHY/slave address.
23-22	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
21-18	XUDC		Transmit user-defined cell bits. Valid values are 0 to 11, the remaining values are reserved.
		0	XUDC feature is disabled. The UTOPIA interface transmits a normal ATM cell of 53 bytes.
		1h-Bh	UTOPIA interface transmits the programmed number (1 to 11) of bytes as an extra header. A UDC can have a minimum of 54 bytes (XUDC = 1h) up to a maximum of 64 bytes (XUDC = Bh).
		Ch-Fh	Reserved
17	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
16	UXEN		UTOPIA transmitter enable bit.
		0	UTOPIA port transmitter is disabled and in reset state.
		1	UTOPIA port transmitter is enabled.
15	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
14	MPHY		UTOPIA receive/transmit multi-PHY mode enable bit. MPHY mode is the default.
		0	Single PHY mode is selected for receive and transmit UTOPIA.
		1	Multi-PHY mode is selected for receive and transmit UTOPIA.

Table 14. UTOPIA Control Register (UCR) Field Descriptions (continued)

Bit	Field	Value	Description
13-6	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
5-2	RUDC		Receive user-defined cell bits. Valid values are 0 to 11, the remaining values are reserved.
		0	RUDC feature is disabled. The UTOPIA interface expects a normal ATM cell of 53 bytes.
		1h-Bh	UTOPIA interface expects to receive the programmed number (1 to 11) of bytes as an extra header. A UDC can have a minimum of 54 bytes (RUDC = 1h) to a maximum of 64 bytes (RUDC = Bh). Receives with header bytes ranging from 1 to 10 bytes.
		Ch-Fh	Reserved
1	Reserved		Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
0	UREN		UTOPIA receiver enable bit.
		0	UTOPIA port receiver is disabled and in reset state.
		1	UTOPIA port receiver is enabled.

12.1.2 Clock Detect Register (CDR)

The clock detect register (CDR) and the UTOPIA clock detection feature allow the DSP to detect the presence of the URCLK and/or UXCLK. The CDR is shown in Figure 8 and described in Table 15.

If a URCLK or a UXCLK edge is not detected within the respective time period specified in CDR, an error bit, RCFP or XCFP, respectively, is set in the error interrupt pending register (EIPR). In addition, the RCPP and XCPP bits in EIPR indicate the presence of the URCLK and UXCLK, respectively. For usage of these interrupts to the CPU, see Section 7.

Figure 8. Clock Detect Register (CDR)

31	24	23	16
Reserved		XCCNT	
R-0		R/W-FFh	

15	8	7	0
Reserved		RCCNT	
R-0		R/W-FFh	

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

Table 15. Clock Detect Register (CDR) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
23-16	XCCNT		Transmit clock count bits specify the number of peripheral clock cycles for which the external UTOPIA transmit clock (UXCLK) must have a low-to-high transition to avoid a reset of the transmit interface. If a UXCLK clock edge is undetected within XCCNT peripheral clock cycles, the transmit UTOPIA port is reset by hardware. The XCF error bit (XCFP) in the error interrupt pending register (EIPR) is set.
		0	Transmit clock detect feature is disabled.
		1h-FFh	Transmit clock detect feature is enabled. This 8-bit value is the number of peripheral clock cycles before the next UTOPIA clock edge (UXCLK) must be present. There must be 255 peripheral clock cycles before the next UTOPIA clock edge (UXCLK).
15-8	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
7-0	RCCNT		Receive clock count bits specify the number of peripheral clock cycles for which the external UTOPIA receive clock must have a low-to-high transition to avoid a reset of the receive interface. If a URCLK clock edge is undetected within RCCNT peripheral clock cycles, the receive UTOPIA port is reset by hardware. The RCF error bit (RCFP) in the error interrupt pending register (EIPR) is set.
		0	Receive clock detect feature is disabled.
		1h-FFh	Receive clock detect feature is enabled. This 8-bit value is the number of peripheral clock cycles before the next UTOPIA clock edge (URCLK) must be present. There must be 255 peripheral clock cycles before the next UTOPIA clock edge (URCLK).

12.1.3 Error Interrupt Enable Register (EIER)

If an error condition is set in the error interrupt enable register (EIER) and the corresponding error is set in the error interrupt pending register (EIPR), an interrupt is generated to the CPU. The EIER is shown in Figure 9 and described in Table 16.

Figure 9. Error Interrupt Enable Register (EIER)

31	19	18	17	16
Reserved		XCPE	XCFE	XQSE
R-0		R/W-0	R/W-0	R/W-0

15	3	2	1	0
Reserved		RCPE	RCFE	RQSE
R-0		R/W-0	R/W-0	R/W-0

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

Table 16. Error Interrupt Enable Register (EIER) Field Descriptions

Bit	Field	Value	Description
31-19	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
18	XCPE		Transmit clock present interrupt enable bit
		0	Transmit clock present interrupt is disabled.
		1	Transmit clock present interrupt is enabled.
17	XCFE		Transmit clock failed interrupt enable bit
		0	Transmit clock failed interrupt is disabled.
		1	Transmit clock failed interrupt is enabled.
16	XQSE		Transmit queue stall interrupt enable bit
		0	Transmit queue stall interrupt is disabled.
		1	Transmit queue stall interrupt is enabled.
15-3	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
2	RCPE		Receive clock present interrupt enable bit
		0	Receive clock present interrupt is disabled.
		1	Receive clock present interrupt is enabled.
1	RCFE		Receive clock failed interrupt enable bit
		0	Receive clock failed interrupt is disabled.
		1	Receive clock failed interrupt is enabled.
0	RQSE		Receive queue stall interrupt enable bit
		0	Receive queue stall interrupt is disabled.
		1	Receive queue stall interrupt is enabled.

12.1.4 Error Interrupt Pending Register (EIPR)

The UTOPIA error conditions are recorded in the error interrupt pending register (EIPR). The EIPR is shown in Figure 10 and described in Table 17. A write of 1 to the XCPP, XCFP, RCPP, or RCFP bit clears the corresponding bit. A write of 0 has no effect. The XQSP and RQSP bits are read-only bits and are cleared automatically by the UTOPIA interface once the error conditions cease. The error conditions in EIPR can generate an interrupt to the CPU, if the corresponding bits are set in the error interrupt enable register (EIER).

The EIPR is shown in Figure 10 and described in Table 17.

Figure 10. Error Interrupt Pending Register (EIPR)

31	19	18	17	16
Reserved		XCPP	XCFP	XQSP
R-0		R/W-0	R/W-0	R/W-0

15	3	2	1	0
Reserved		RCPP	RCFP	RQSP
R-0		R/W-0	R/W-0	R/W-0

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

Table 17. Error Interrupt Pending Register (EIPR)

Bit	Field	Value	Description
31-19	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
18	XCPP		Transmit clock present interrupt pending bit indicates if the UTOPIA transmit clock (UXCLK) is present. XCPP is valid regardless if the transmit interface is enabled or disabled.
		0	UXCLK is not present.
		1	UXCLK is present. If the corresponding bit in EIER is set, an interrupt UINT is sent to the CPU.
17	XCFP		Transmit clock failed interrupt pending bit is activated only when the UTOPIA transmit interface is enabled (UXEN in UCR = 1).
		0	UXCLK is present.
		1	UXCLK failed. No UXCLK is detected for a period longer than that specified in the XCCNT field of CDR. If the corresponding bit in EIER is set, an interrupt UINT is sent to the CPU.
16	XQSP		Transmit queue stall interrupt pending bit
		0	No transmit queue stall condition
		1	Transmit queue stalled, a write is performed to a full transmit queue. The write is stalled until the queue is drained and space is available. Data is not overwritten. XQSP is cleared once the queue has space available and writes can continue.
15-3	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
2	RCPP		Receive clock present interrupt pending bit indicates if the UTOPIA receive clock (URCLK) is present. RCPP is valid regardless if the receive interface is enabled or disabled.
		0	URCLK is not present.
		1	URCLK is present. If the corresponding bit in EIER is set, an interrupt UINT is sent to the CPU.
1	RCFP		Receive clock failed interrupt pending bit is activated only when the UTOPIA receive interface is enabled (UREN in UCR = 1).
		0	URCLK is present.
		1	URCLK failed. No URCLK is detected for a period longer than that specified in the RCCNT field of CDR. If the corresponding bit in EIER is set, an interrupt UINT is sent to the CPU.
0	RQSP		Receive queue stall interrupt pending bit
		0	No receive queue stall condition
		1	Receive queue stalled, a read is performed from an empty receive queue. The read is stalled until valid data is available in the queue. RQSP is cleared as soon as valid data is available and the read is performed.

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	Broadband	www.ti.com/broadband
DSP	dsp.ti.com	Digital Control	www.ti.com/digitalcontrol
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Military	www.ti.com/military
Logic	logic.ti.com	Optical Networking	www.ti.com/opticalnetwork
Power Mgmt	power.ti.com	Security	www.ti.com/security
Microcontrollers	microcontroller.ti.com	Telephony	www.ti.com/telephony
RFID	www.ti-rfid.com	Video & Imaging	www.ti.com/video
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Wireless	www.ti.com/wireless

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