

***TMS320C64x DSP
Universal Test and Operations
PHY Interface for ATM (UTOPIA)
Reference Guide***

Literature Number: SPRU583
July 2003



IMPORTANT NOTICE

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

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

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

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

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

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

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

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

Copyright © 2003, Texas Instruments Incorporated

Read This First

About This Manual

This document describes the universal test and operations PHY interface for asynchronous transfer mode (UTOPiA) in the TMS320C64x™ 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 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

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.

TMS320C6000 CPU and Instruction Set Reference Guide (literature number SPRU189) describes the TMS320C6000™ CPU architecture, instruction set, pipeline, and interrupts for these digital signal processors.

TMS320C6000 Peripherals Reference Guide (literature number SPRU190) describes the peripherals available on the TMS320C6000™ DSPs.

TMS320C6000 Technical Brief (literature number SPRU197) gives an introduction to the TMS320C62x™ and TMS320C67x™ DSPs, development tools, and third-party support.

TMS320C64x Technical Overview (SPRU395) gives an introduction to the TMS320C64x™ DSP and discusses the application areas that are enhanced by the TMS320C64x VelociTI™.

TMS320C6000 Programmer's Guide (literature number SPRU198) describes ways to optimize C and assembly code for the TMS320C6000™ DSPs and includes application program examples.

TMS320C6000 Code Composer Studio Tutorial (literature number SPRU301) introduces the Code Composer Studio™ integrated development environment and software tools.

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

TMS320C6x Peripheral Support Library Programmer's Reference (literature number SPRU273) describes the contents of the TMS320C6000™ peripheral support library of functions and macros. It lists functions and macros both by header file and alphabetically, provides a complete description of each, and gives code examples to show how they are used.

TMS320C6000 Chip Support Library API Reference Guide (literature number SPRU401) describes a set of application programming interfaces (APIs) used to configure and control the on-chip peripherals.

Trademarks

Code Composer Studio, C6000, C62x, C64x, C67x, TMS320C6000, TMS320C62x, TMS320C64x, TMS320C67x, and VelociTI are trademarks of Texas Instruments.

PowerQUICC II is a trademark of Motorola, Inc.

Contents

1	Overview	9
2	Cell Transfer Format	11
3	UTOPIA Slave as ATM Controller	12
3.1	UTOPIA Slave Pins	12
3.2	Slave-Transmit Operation	14
3.3	Slave-Transmit Queue	15
3.4	Slave-Receive Operation	16
3.5	Slave-Receive Queue	17
3.6	UTOPIA Events Generation	17
3.7	Multi-PHY (MPHY) Operation	18
4	EDMA Servicing UTOPIA	19
4.1	EDMA Setup for UTOPIA Transmitter	19
4.2	EDMA Setup for UTOPIA Receiver	19
5	CPU Servicing UTOPIA	22
6	UTOPIA Clocking and Clock Detection	23
7	Special Transfer Conditions	24
8	Endian Considerations	25
9	UTOPIA Reset	29
10	UTOPIA Slave Initialization Sequence	30
11	ATM Adaptation Layer (AAL) Functions	30
12	Registers	31
12.1	UTOPIA Control Register (UCR)	31
12.2	UTOPIA Interrupt Enable Register (UIER)	34
12.3	UTOPIA Interrupt Pending Register (UIPR)	35
12.4	Clock Detect Register (CDR)	36
12.5	Error Interrupt Enable Register (EIER)	37
12.6	Error Interrupt Pending Register (EIPR)	39

Figures

1	TMS320C64x DSP Block Diagram	10
2	UTOPIA Block Diagram	10
3	Cell Transfer Formats for 8-Bit Mode	11
4	UTOPIA Slave Interfaced to Motorola MPC8260 PowerQUICC II Master in 8-Bit Mode	12
5	ATM Controller Slave Transmit Timing Diagram	14
6	ATM Controller Slave Receive Timing Diagram	16
7	EDMA Setup for UTOPIA Transmitter Code Example	20
8	EDMA Setup for UTOPIA Receiver Code Example	21
9	8-Bit UTOPIA Slave in Little-Endian Mode (BEND = 0) with RUDC/XUDC = 0	26
10	8-Bit UTOPIA Slave in Big-Endian Mode (BEND = 1) with RUDC/XUDC = 0	27
11	8-Bit UTOPIA Slave in Big-Endian Mode (BEND = 1) with RUDC/XUDC = 1	27
12	8-Bit UTOPIA Slave in Little-Endian Mode (BEND = 0) with RUDC/XUDC = 7	27
13	8-Bit UTOPIA Slave in Little-Endian Mode (BEND = 0) with RUDC/XUDC = 11	28
14	8-Bit UTOPIA Slave in Big-Endian Mode (BEND = 1) with RUDC/XUDC = 11	28
15	UTOPIA Control Register (UCR)	31
16	UTOPIA Interrupt Enable Register (UIER)	34
17	UTOPIA Interrupt Pending Register (UIPR)	35
18	Clock Detect Register (CDR)	36
19	Error Interrupt Enable Registers (EIER)	37
20	Error Interrupt Pending Register (EIPR)	39

Tables

1	UTOPIA Slave Pin Descriptions	13
2	EDMA Synchronization Events from UTOPIA	19
3	CPU Interrupt from UTOPIA	22
4	8-Bit UTOPIA Slave Modes Depending on BEND, XUDC, and RUDC Bits	26
5	UTOPIA Pin Reset Values	29
6	UTOPIA Configuration Registers	31
7	UTOPIA Data Ports	31
8	UTOPIA Control Register (UCR) Field Descriptions	32
9	UTOPIA Interrupt Enable Register (UIER) Field Descriptions	34
10	UTOPIA Interrupt Pending Register (UIPR) Field Descriptions	35
11	Clock Detect Register (CDR) Field Descriptions	36
12	Error Interrupt Enable Register (EIER) Field Descriptions	38
13	Error Interrupt Pending Register (EIPR) Field Descriptions	39

This document describes the universal test and operations PHY interface for asynchronous transfer mode [ATM] (UTOPIA) in the TMS320C64x™ digital signal processors (DSPs) of the TMS320C6000™ DSP family.

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 operation 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.

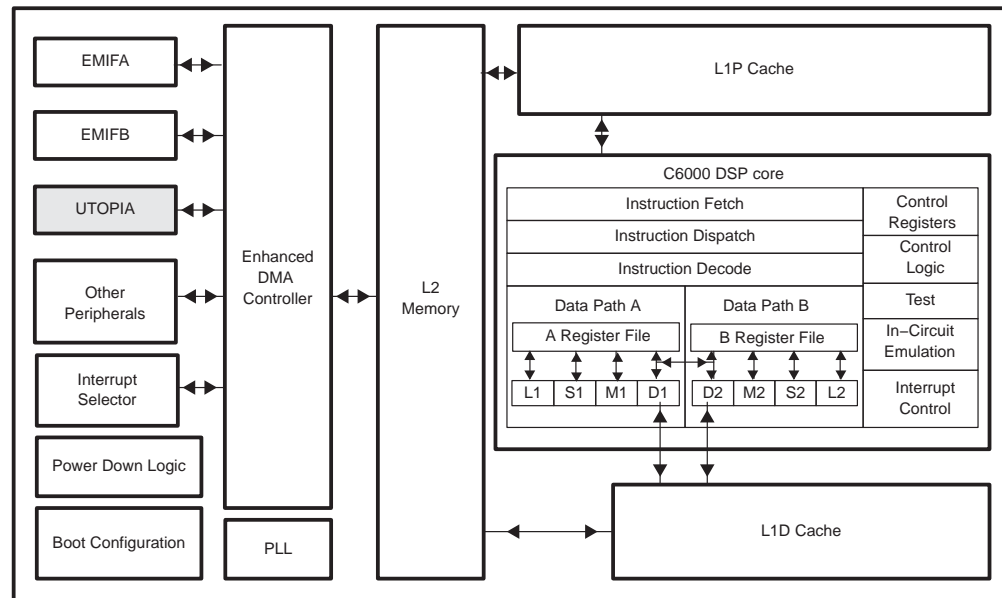
Both the CPU and enhanced DMA (EDMA) controller can service the UTOPIA. The ATM Adaptation Layer (AAL) commonly called as Segmentation and Reassembly (SAR) functions should be performed in software.

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

Figure 1 shows the UTOPIA interface on some C64x™ DSP.

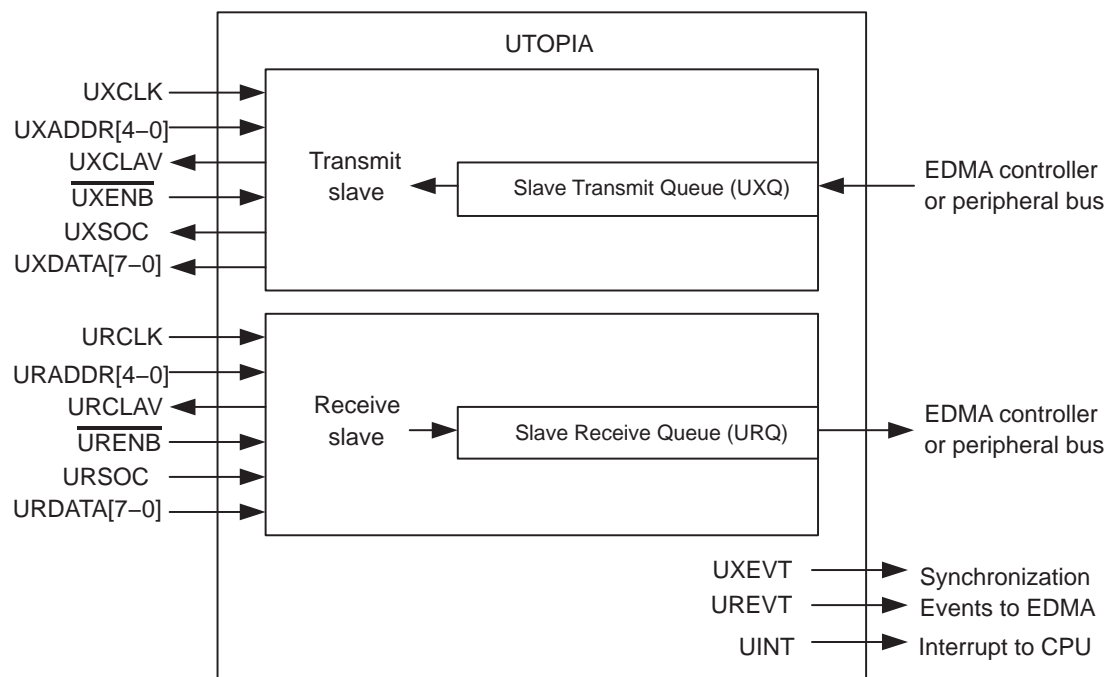
The UTOPIA slave consists of the transmit interface and the receive interface as shown in Figure 2. The interface signals are described in section 3.1. The UTOPIA sends notification of interrupts to the CPU via the UINT signal and synchronization events to the EDMA controller via the UXEVT and UREVT signals.

Figure 1. TMS320C64x DSP Block Diagram



Note: Refer to your device-specific datasheet for your peripheral set.

Figure 2. UTOPIA Block Diagram

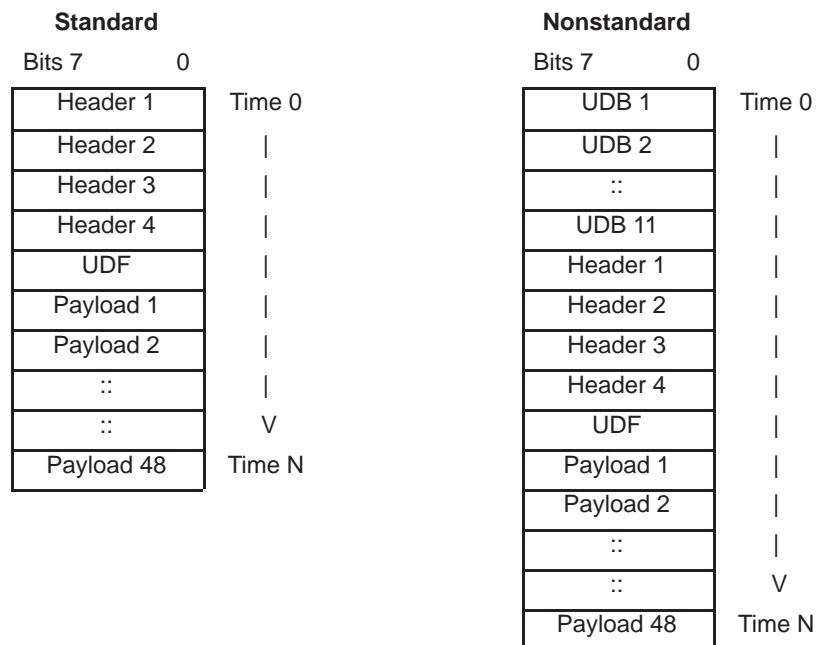


2 Cell Transfer Format

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 C6000™ 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. See also section 8.

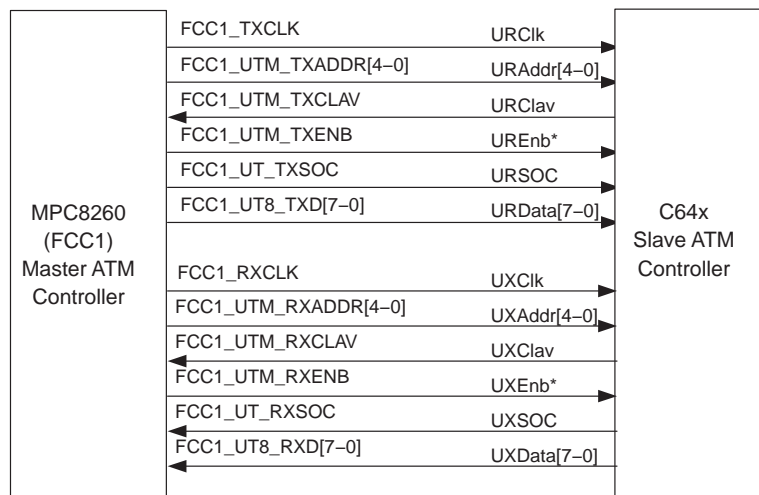
Figure 3. Cell Transfer Formats for 8-Bit Mode



3 UTOPIA Slave as ATM Controller

The UTOPIA interface can be 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 PowerQUICC II™ 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 shows the pins and their direction relevant 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 Slave Pin Descriptions

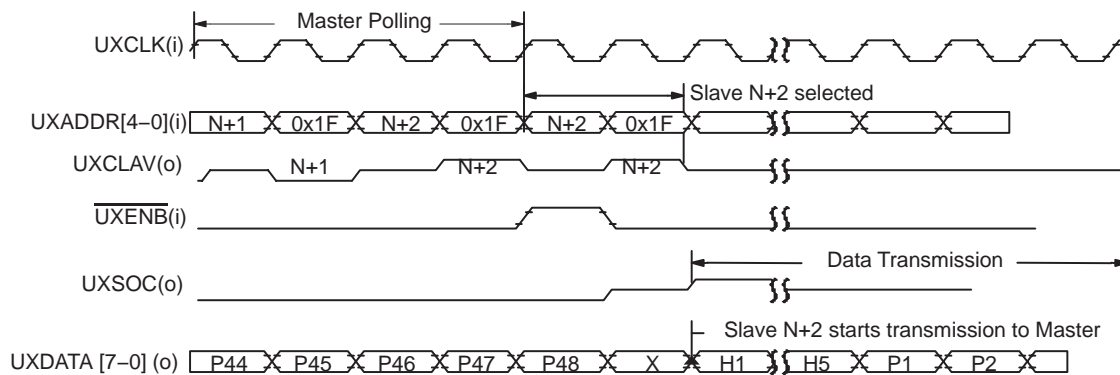
Pin	Direction	Description
UTOPIA Transmit Interface (Slave mode)		
UXCLK	In	UTOPIA Transmit Clock. An input driven by the master in the system. Transmit data and transmit control signals are synchronous to this clock.
UXADDR[4–0]	In	5-bit address input driven by the master ATM Controller to identify 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.
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.
UTOPIA Receive Interface (Slave mode)		
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.
URENB	In	UTOPIA Receive Interface Enable. An active-low signal driven by the master to enable the receive interface of the slave. It indicates to the slave to sample Receive Data and SOC signal 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. The UTOPIA slave-transmit block diagram is shown in Figure 2 (page 10). The slave-transmit queue can be accessed via the UXQ data port. The CPU/EDMA services the slave-transmit queue with 32-bit writes when a transmit interrupt/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 on-going data transfers) to get 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 facilitates the UTOPIA interface to be ready to transmit data whenever the master requests one. 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) 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 CPU can also be used to service the UTOPIA via the UINT signal (see section 4).

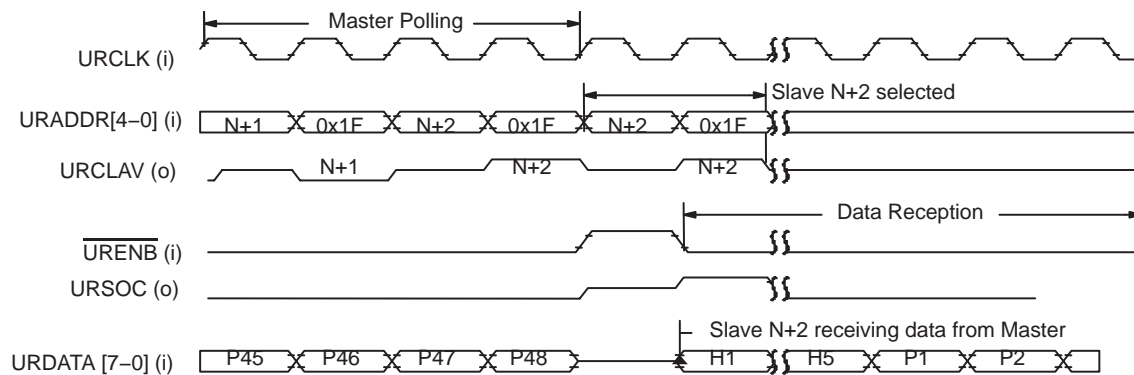
The UTOPIA slave will agree for transmission to the master by asserting its UXCLAV signal when there is at least one cell available in the slave-transmit queue. If the slave cannot provide the next cell in a contiguous fashion, it deasserts its UXCLAV signal in the cycle following the completion of the current cell transmission. The UXCLAV signal remains asserted if the slave has another cell available to transfer to the master. The UXCLAV signal is not available to the CPU, so the CPU cannot determine if the transmit queue is empty. The master may 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 it.

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 (page 10). The slave-receive queue can be accessed via the URQ data port. The CPU/EDMA controller services the slave-receive queue with 32-bit reads when a receive interrupt/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 can choose to transmit to this slave or continue to poll to find a suitable slave for its data. In any case, 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 or UINT to the CPU (if desired) when at least one cell worth of data is available. As soon as the first read is performed by the EDMA or CPU, 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) of that cell occurs.
- 3) UREVT is regenerated when the next complete cell is available in the slave-receive queue.
- 4) Go to step 2.

The CPU can also be used to service the UTOPIA via the UINT signal (see section 4).

The UTOPIA slave will agree 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 deasserts its URCLAV signal at least 4 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 may 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. 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 can be continuously transferred without having to wait for a full or empty queue. Refer to section 3.3 and section 3.5 for details on the generation of these events. Either the EDMA or the CPU can be used to service the UTOPIA in response to these events, as discussed in section 4 and section 5, respectively. Typically, the EDMA is used to service the UTOPIA for performance considerations.

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. The programming interface can either be the HPI or the PCI.

MPHY operation is based on cell-level handshaking. 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 Servicing UTOPIA

Typically, the EDMA is used to service the UTOPIA interface. Table 2 lists the UTOPIA synchronization events to the EDMA and their corresponding EDMA channel.

Table 2. EDMA Synchronization Events from UTOPIA

EDMA Event	EDMA Channel	Synchronization Event Description
UXEVT	40	Transmit event from the UTOPIA to the EDMA. UXEVT is asserted if the transmit queue has space for at least one cell packet.
UREVT	32	Receive event from the UTOPIA to the EDMA. UREVT is asserted if a complete cell packet is available in the receive queue.

4.1 EDMA Setup for UTOPIA Transmitter

As mentioned in section 3.6, the UTOPIA transmitter generates a UXEVT synchronization event to the EDMA when at least one cell-packet space is available in the slave-transmit queue. EDMA channel 40 is dedicated to the UXEVT event. Per 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.

An example of an EDMA setup to service a UTOPIA transmitter is shown in Figure 7.

4.2 EDMA Setup for UTOPIA Receiver

As mentioned in section 3.6, the UTOPIA receiver generates a UREVT synchronization event to the EDMA when the slave-receive queue has space for at least one cell-packet. EDMA channel 32 is dedicated to the UREVT event. Per UREVT synchronization event, one frame of cell-packet data is read from the slave-receive queue via 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).

An example of an EDMA setup to service a UTOPIA receiver is shown in Figure 8.

Figure 7. EDMA Setup for UTOPIA Transmitter Code Example

```

//*****\
* EDMA XMIT Channel Configuration
\*****/
void
setup_EdmaXmit() {
    /* Setup EDMA registers */
    cfgEdmaOut0.opt = EDMA_OPT_RMK(
        EDMA_OPT_PRI_HIGH,
        EDMA_OPT_ESIZE_32BIT,
        EDMA_OPT_2DS_YES,
        EDMA_OPT_SUM_INC,
        EDMA_OPT_2DD_NO,
        EDMA_OPT_DUM_NONE,
        EDMA_OPT_TCINT_YES,
        EDMA_OPT_TCC_OF(TCCXMIT0NUM),
        EDMA_OPT_TCCM_OF(TCCXMIT0NUM>>4),
        EDMA_OPT_ATCINT_NO,
        EDMA_OPT_ATCC_OF(0),
        EDMA_OPT_PDTS_DISABLE,
        EDMA_OPT_PDTD_DISABLE,
        EDMA_OPT_LINK_YES,
        EDMA_OPT_FS_NO
    );
    cfgEdmaOut0.src = EDMA_SRC_RMK(DSP_out0);
    cfgEdmaOut0.dst = EDMA_DST_RMK(UTOP_XMTQ_ADDR);
    cfgEdmaOut0.cnt = EDMA_CNT_RMK((NUM_XMIT_CELL-1), xcell_sz);
    cfgEdmaOut0.idx = EDMA_IDX_RMK((xcell_sz*4), 0);
    /* xcell_sz = # of 32-b words in each transmit cell */
    cfgEdmaOut0.rld = EDMA_RLD_RMK(0, hEdmaNullTbl);
    /* Copy above setup to the EDMA Handle */
    EDMA_config(hEdmaOut0, &cfgEdmaOut0);
}

```

Figure 8. EDMA Setup for UTOPIA Receiver Code Example

```

/*****\
* EDMA RECV Channel Configuration
\*****/
void
setup_EdmaRecv() {
    /* Setup EDMA registers */
    cfgEdmaIn0.opt = EDMA_OPT_RMK(
        EDMA_OPT_PRI_MEDIUM,
        EDMA_OPT_ESIZE_32BIT,
        EDMA_OPT_2DS_NO,
        EDMA_OPT_SUM_NONE,
        EDMA_OPT_2DD_YES,
        EDMA_OPT_DUM_INC,
        EDMA_OPT_TCINT_YES,
        EDMA_OPT_TCC_OF(TCCRECV0NUM),
        EDMA_OPT_TCCM_OF(TCCRECV0NUM>>4),
        EDMA_OPT_ATCINT_NO,
        EDMA_OPT_ATCC_OF(0),
        EDMA_OPT_PDTS_DISABLE,
        EDMA_OPT_PDTD_DISABLE,
        EDMA_OPT_LINK_YES,
        EDMA_OPT_FS_NO
    );
    cfgEdmaIn0.src = EDMA_SRC_RMK(UTOP_RCVQ_ADDR);
    cfgEdmaIn0.dst = EDMA_DST_RMK(DSP_in0);
    cfgEdmaIn0.cnt = EDMA_CNT_RMK((NUM_RECV_CELL-1), rcell_sz);
    cfgEdmaIn0.idx = EDMA_IDX_RMK((rcell_sz*4), 0);
    /* rcell_sz = # of 32-b words in each receive cell */
    cfgEdmaIn0.rld = EDMA_RLD_RMK(0, hEdmaNullTbl);
    /* Copy above setup to the EDMA Handle */
    EDMA_config(hEdmaIn0, &cfgEdmaIn0);
}

```

5 CPU Servicing UTOPIA

The CPU can also be used to service the UTOPIA interface, but is not recommended. The CPU can access the slave-transmit queue and the slave-receive queue via the data ports UXQ and URQ, respectively. The UTOPIA interface generates a single CPU interrupt, UINT, for both the transmit and receive interface.

The conditions that generate transmit and receive events to the EDMA are exactly the conditions that generate transmit and receive interrupts to the CPU. Therefore, the relevant interrupt pending bit in the UTOPIA interrupt pending register (UIPR) is equivalent to the UXEVT/UREVT to the EDMA. But the interrupt to the CPU is generated only if the relevant enable bit is set in the UTOPIA interrupt enable register (UIER). For better system performance, the EDMA should be used to service the UTOPIA (see section 4). Table 3 lists the UTOPIA events that cause the CPU interrupt, UINT.

The transmit interrupt and receive interrupt are generated on the same conditions that a UXEVT or UREVT is generated to the EDMA, see sections 3.3 and 3.5 for details.

Table 3. CPU Interrupt from UTOPIA

UTOPIA Event	CPU Interrupt	Interrupt Number	Interrupt Description
UXEVT	UINT	23	Transmit interrupt from the slave-transmit queue to the CPU.
UREVT	UINT	23	Receive interrupt from the slave-receive queue to the CPU.

Note: A single UTOPIA Interrupt is generated to the CPU, if the corresponding bit is set in the UTOPIA interrupt enable register (UIER). The interrupt from individual queues for both transmit and receive is read from the UTOPIA interrupt pending register (UIPR).

The generation and clearing of interrupts is as follows:

- ☐ The slave-transmit queue and slave-receive queue generate ready interrupts that set the relevant bit in UIPR.
- ☐ If the relevant bit in UIER is set and the UTOPIA interrupt to the CPU, UINT (mapped to interrupt number 23), is enabled in the CPU interrupt enable register (IER), the CPU will be interrupted.
- ☐ Within the interrupt service routine (ISR):
 - Read UIPR to determine which queue(s) generated the interrupt.
 - Service the queue(s) as required by performing a cell-packet read from the URQ or a cell-packet write to the UXQ (14 words for a standard cell-packet; 14, 15, or 16 words for a nonstandard cell-packet). To service the receiver, a URQ read should always start at address 3C00 0000h. To service the transmitter, a UXQ write should always start at address 3D00 0000h.
 - The ISR clears the bit in UIPR upon servicing that interrupt, enabling recognition of further interrupts from the same queue. Writing a 1 to the relevant bit clears that interrupt in UIPR; writing a 0 has no effect.

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/4 rate.

7 Special Transfer Conditions

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

- ❑ **Runt cells:** Runt cells are those cells that are shorter than the standard ATM cell (53 bytes for 8-bit mode). This occurs when the device sending data asserts SOC in the middle of a cell transfer. In the C64x UTOPIA, runt cells are handled in hardware. If the receive section of the UTOPIA detects a SOC before a complete cell is received, the byte count is reset and the runt cell is overwritten by the next new data. In the transmit direction, you cannot force SOC; intentional runt cell generation is not supported.
- ❑ **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 via 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.

- Queue 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, 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. In order for the UTOPIA interface to present the data transferred across its interface to the DSP/CPU 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 (page 11).

Depending on the user-defined cell (RUDC and XUDC bits) and endian format (BEND bit) as listed in Table 4, bytes are placed in the transmit and receive queues as shown in Figure 9 through Figure 14.

The cell-packet formats in Figure 9 through Figure 14 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

Endian mode	UCR Bits			See ...
	BEND bit	XUDC value	RUDC value	
Little endian	0	0 (disabled)	0 (disabled)	Figure 9
Big endian	1	0 (disabled)	0 (disabled)	Figure 10
Big endian	1	1	1	Figure 11
Little endian	0	7	7	Figure 12
Little endian	0	11	11	Figure 13
Big endian	1	11	11	Figure 14

Figure 9. 8-Bit UTOPIA Slave in Little-Endian Mode (BEND = 0) with RUDC/XUDC = 0

Queue Bits	31–24	23–16	15–8	7–0	
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

Figure 10. 8-Bit UTOPIA Slave in Big-Endian Mode ($BEND = 1$) with $RUDC/XUDC = 0$

Queue Bits	31–24	23–16	15–8	7–0	
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

Figure 11. 8-Bit UTOPIA Slave in Big-Endian Mode ($BEND = 1$) with $RUDC/XUDC = 1$

Queue Bits	31–24	23–16	15–8	7–0	
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

Figure 12. 8-Bit UTOPIA Slave in Little-Endian Mode ($BEND = 0$) with $RUDC/XUDC = 7$

Queue Bits	31–24	23–16	15–8	7–0	
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

Figure 13. 8-Bit UTOPIA Slave in Little-Endian Mode ($BEND = 0$) with $RUDC/XUDC = 11$

Queue Bits	31–24	23–16	15–8	7–0	
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

Figure 14. 8-Bit UTOPIA Slave in Big-Endian Mode ($BEND = 1$) with $RUDC/XUDC = 11$

Queue Bits	31–24	23–16	15–8	7–0	
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.

Table 5 shows the recommended reset values of the UTOPIA pins. The UTOPIA pins have no internal pull-up or pull-down resistors. At reset, all outputs are driven to a high-impedance state to facilitate MPHY operation. All input pins should be pulled externally to bring inputs to a known state when not driven as shown in Table 5.

Table 5. 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 via 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 via the HPI/PCI or McBSP interfaces.
- ☐ Take the interface out of reset by setting the UREN bit to 1 to enable the receive interface and setting the UXEN bit to 1 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

The UTOPIA port is configured via the configuration registers listed in Table 6. The data for transmit and receive queues are accessible via the EDMA controller or CPU at the data port listed in Table 7. See the device-specific datasheet for the memory address of these registers.

Table 6. UTOPIA Configuration Registers

Acronym	Register Name	Section
UCR	UTOPIA Control Register	12.1
UIER	UTOPIA Interrupt Enable Register	12.2
UIPR	UTOPIA Interrupt Pending Register	12.3
CDR	Clock Detect Register	12.4
EIER	Error Interrupt Enable Register	12.5
EIPR	Error Interrupt Pending Register	12.6

Table 7. UTOPIA Data Ports

Acronym	Queue Name	Section
URQ	UTOPIA Receive Queue	3.5
UXQ	UTOPIA Transmit Queue	3.3

12.1 UTOPIA Control Register (UCR)

The UTOPIA interface is configured via the UTOPIA control register (UCR) and contains UTOPIA status and control bits. The UCR is shown in Figure 15 and described in Table 8.

Figure 15. UTOPIA Control Register (UCR)

31	30	29	28	24	23	22	21	18	17	16
BEND	Reserved	SLID	Reserved	XUDC	—	UXEN				
R/W-0	R-0	R/W-0	R-0	R/W-0	R-0	R/W-0		R-0		R/W-0
15	14	13		6	5			2	1	0
—	MPHY	Reserved		RUDC	—	UREN				
R-0	R/W-0	R-0		R/W-0	R-0	R/W-0		R-0		R/W-0

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

Table 8. UTOPIA Control Register (UCR) Field Descriptions

Bit	field [†]	symval [†]	Value	Description
31	BEND			Big-endian mode enable bit for data transferred by way of the UTOPIA interface.
		LITTLE	0	Data is assembled to conform to little-endian format.
		BIG	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	OF(value)	0–1Fh	Slave ID bits. Applicable in MPHY mode. This 5-bit value is used to identify the UTOPIA in a MPHY set up. Does not apply to single-PHY slave operation.
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	OF(value)	0–Fh	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 extra header. A UDC may 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.
		DISABLE	0	UTOPIA port transmitter is disabled and in reset state.
		ENABLE	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.

[†] For CSL implementation, use the notation UTOP_UCR_field_symval

Table 8. UTOPIA Control Register (UCR) Field Descriptions (Continued)

Bit	field [†]	symval [†]	Value	Description
14	MPHY			UTOPIA receive/transmit multi-PHY mode enable bit.
		SINGLE	0	Single PHY mode is selected for receive and transmit UTOPIA.
		MULTI	1	Multi-PHY mode is selected for receive and transmit UTOPIA.
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	OF(<i>value</i>)	0–Fh	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 extra header. A UDC may have a minimum of 54 bytes (RUDC = 1h) up to a maximum of 64 bytes (RUDC = Bh).
			Ch–Fh	Reserved
1	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.
0	UREN			UTOPIA receiver enable bit.
		DISABLE	0	UTOPIA port receiver is disabled and in reset state.
		ENABLE	1	UTOPIA port receiver is enabled.

[†] For CSL implementation, use the notation `UTOP_UCR_field_symval`

12.2 UTOPIA Interrupt Enable Register (UIER)

The relevant interrupts for each queue are enabled in the UTOPIA interrupt enable register (UIER). The UIER is shown in Figure 16 and described in Table 9.

Figure 16. UTOPIA Interrupt Enable Register (UIER)

31	17	16
Reserved		RQIE
R-0		R/W-0
15	1	0
Reserved		XQIE
R-0		R/W-0

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

Table 9. UTOPIA Interrupt Enable Register (UIER) Field Descriptions

Bit	field†	symval†	Value	Description
31–17	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	RQIE	OF(value)	0	Receive queue interrupt enable bit.
			0	Receive queue interrupt is disabled. No interrupts are sent to the CPU upon the UREVT event.
			1	Receive queue interrupt is enabled. Upon UREVT, interrupt UINT is sent to the CPU interrupt selector.
15–1	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.
0	XQIE	OF(value)	0	Transmit queue interrupt enable bit.
			0	Transmit queue interrupt is disabled. No interrupts are sent to the CPU upon the UXEVT event.
			1	Transmit queue interrupt is enabled. Upon UXEVT, interrupt UINT is sent to the CPU interrupt selector.

† For CSL implementation, use the notation `UTOP_UIER_field_symval`

12.3 UTOPIA Interrupt Pending Register (UIPR)

Interrupts are captured in the UTOPIA interrupt pending register (UIPR). The UIPR is shown in Figure 17 and described in Table 10.

Figure 17. UTOPIA Interrupt Pending Register (UIPR)

31		17	16
Reserved			RQIP
R-0			R/W-0
15		1	0
Reserved			XQIP
R-0			R/W-0

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

Table 10. UTOPIA Interrupt Pending Register (UIPR) Field Descriptions

Bit	field [†]	symval [†]	Value	Description
31–17	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	RQIP			Receive queue interrupt pending bit.
		–	0	No receive queue interrupt is pending.
		CLEAR	1	Receive queue interrupt is pending.
15–1	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.
0	XQIP			Transmit queue interrupt pending bit.
		–	0	No transmit queue interrupt is pending.
		CLEAR	1	Transmit queue interrupt is pending.

[†] For CSL implementation, use the notation `UTOP_UIPR_field_symval`

12.4 Clock Detect Register (CDR)

The clock detect register (CDR) and the UTOPIA clock detection feature allows the DSP to detect the presence of the URCLK and/or UXCLK. The CDR is shown in Figure 18 and described in Table 11.

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. See section 7 for usage of these interrupts to the CPU.

Figure 18. 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 = Read only; R/W = Read/Write; -n = value after reset

Table 11. Clock Detect Register (CDR) Field Descriptions

Bit	field [†]	symval [†]	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	OF(value)	0–FFh	Transmit clock count bits specify the number of peripheral clock cycles that the external UTOPIA transmit clock (UXCLK) must have a low-to-high transition to avoid a reset of the transmit interface.
			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. 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.

[†] For CSL implementation, use the notation UTOP_CDR_field_symval

Table 11. Clock Detect Register (CDR) Field Descriptions (Continued)

Bit	field [†]	symval [†]	Value	Description
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	OF(value)	0–FFh	Receive clock count bits specify the number of peripheral clock cycles that the external UTOPIA receive clock must have a low-to-high transition to avoid a reset of the receive interface.
			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. 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.

[†] For CSL implementation, use the notation `UTOP_CDR_field_symval`

12.5 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 19 and described in Table 12.

Figure 19. Error Interrupt Enable Registers (EIER)

31		19	18	17	16
Reserved			XCPE	XCPE	XQSE
R-0			R/W-0	R/W-0	R/W-0
15		3	2	1	0
Reserved			RCPE	RCPE	RQSE
R-0			R/W-0	R/W-0	R/W-0

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

Table 12. Error Interrupt Enable Register (EIER) Field Descriptions

Bit	field [†]	symval [†]	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.
		DISABLE	0	Transmit clock present interrupt is disabled.
		ENABLE	1	Transmit clock present interrupt is enabled.
17	XCFE			Transmit clock failed interrupt enable bit.
		DISABLE	0	Transmit clock failed interrupt is disabled.
		ENABLE	1	Transmit clock failed interrupt is enabled.
16	XQSE			Transmit queue stall interrupt enable bit.
		DISABLE	0	Transmit queue stall interrupt is disabled.
		ENABLE	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.
		DISABLE	0	Receive clock present interrupt is disabled.
		ENABLE	1	Receive clock present interrupt is enabled.
1	RCFE			Receive clock failed interrupt enable bit.
		DISABLE	0	Receive clock failed interrupt is disabled.
		ENABLE	1	Receive clock failed interrupt is enabled.
0	RQSE			Receive queue stall interrupt enable bit.
		DISABLE	0	Receive queue stall interrupt is disabled.
		ENABLE	1	Receive queue stall interrupt is enabled.

[†] For CSL implementation, use the notation `UTOP_EIER_field_symval`

12.6 Error Interrupt Pending Register (EIPR)

The UTOPIA error conditions are recorded in the error interrupt pending register (EIPR). The EIPR is shown in Figure 20 and described in Table 13. 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).

Figure 20. Error Interrupt Pending Register (EIPR)

31		19	18	17	16
Reserved			XCPP	XCFP	XQSP
R-0			R/W-0	R/W-0	R-0
15		3	2	1	0
Reserved			RCPP	RCFP	RQSP
R-0			R/W-0	R/W-0	R-0

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

Table 13. Error Interrupt Pending Register (EIPR) Field Descriptions

Bit	field [†]	symval [†]	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.
		CLEAR	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.
		CLEAR	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.

[†] For CSL implementation, use the notation `UTOP_EIPR_field_symval`

Table 13. Error Interrupt Pending Register (EIPR) Field Descriptions (Continued)

Bit	field [†]	symval [†]	Value	Description
16	XQSP	OF(value)		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	RCPD			Receive clock present interrupt pending bit indicates if the UTOPIA receive clock (URCLK) is present. RCPD is valid regardless if the receive interface is enabled or disabled.
		–	0	URCLK is not present.
		CLEAR	1	URCLK is present. If the corresponding bit in EIER is set, an interrupt UINT is sent to the CPU.
1	RCLP			Receive clock failed interrupt pending bit is activated only when the UTOPIA receive interface is enabled (UREN in UCR = 1).
		–	0	URCLK is present.
		CLEAR	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	OF(value)		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.

[†] For CSL implementation, use the notation `UTOP_EIPR_field_symval`

Index

A

ATM adaption layer interface 30

B

BEND bit 31

block diagram

nonstandard cell transfer format 11

standard cell transfer format 11

TMS320C64x DSP 10

UTOPIA 10

UTOPIA slave interfaced to Motorola MPC8260

PowerQUICC II master 12

C

CDR 36

cell transfer format 11

clock detect register (CDR) 36

clocking 23

CPU interrupt (UINT) 22

CPU servicing UTOPIA 22

E

EDMA servicing UTOPIA 19

EDMA setup for UTOPIA receiver 19

EDMA setup for UTOPIA transmitter 19

EIER 37

EIPR 39

endian considerations 25

error interrupt enable register (EIER) 37

error interrupt pending register (EIPR) 39

events generation 17

I

interface signals 9

interrupts 22

M

MPHY bit 31

multi-PHY operation 18

N

notational conventions 3

O

operation

multi-PHY 18

slave-receive 16

slave-transmit 14

overview 9

R

RCCNT bits 36

RCFE bit 37

RCFP bit 39

RCPE bit 37

RCPP bit 39

receive event from UTOPIA to EDMA (UREVT) 19

registers 31

clock detect register (CDR) 36

error interrupt enable register (EIER) 37

error interrupt pending register (EIPR) 39

UTOPIA control register (UCR) 31

UTOPIA interrupt enable register (UIER) 34

UTOPIA interrupt pending register (UIPR) 35

related documentation from Texas Instruments 3

reset 29

RQIE bit 34

RQIP bit 35

RQSE bit 37

RQSP bit 39

RUDC bits 31

S

servicing UTOPIA

CPA 22

EDMA 19

slave-receive operation 16

slave-receive queue 17

slave-transmit operation 14

slave-transmit queue 15

SLID bits 31

special transfers 24

T

trademarks 4

transmit event from UTOPIA to EDMA (UXEVT) 19

U

UCR 31

UIER 34

UINT 22

UIPR 35

UREN bit 31

UREVT 19

UTOPIA control register (UCR) 31

UTOPIA interrupt enable register (UIER) 34

UTOPIA interrupt pending register (UIPR) 35

UTOPIA slave as ATM controller 12

UTOPIA slave initialization sequence 30

UTOPIA slave pins 12

UXEN bit 31

UXEVT 19

X

XCCNT bits 36

XCFE bit 37

XCFP bit 39

XCPE bit 37

XCPP bit 39

XQIE bit 34

XQIP bit 35

XQSE bit 37

XQSP bit 39

XUDC bits 31