UART 16750 for Cu-08

Core Databook

SA15-5794-01

Revision 1

March 12, 2007 - IBM Confidential

Printed in the United States of America March 2007

IBM Systems and Technology Group
2070 Route 52, Bldg. 330
Hopewell Junction, NY 12533-6351

The IBM home page can be found at **ibm.com**

The IBM Semiconductor solutions page can be found at **ibm.com**/chips

SA15-5794-01
March 12, 2007 - IBM Confidential

Contents

Preface

About this Book

This book describes the functionality, registers, interfaces, signals, timing specifications, core integration, clocking guidelines, testing and physical design requirements, and the performance and operating environment of the UART 16750 for Cu-08.

This document is available to anyone with access to the Design Kit for the core. If you need access, contact your IBM representative.

Who Should Use this Book

This book is for hardware, software, and application developers who need to understand the UART 16750 for Cu-08.

Revision Log

Note: Changes from previous versions of this document are marked with blue change bars throughout the text. Italicized text and page numbers in this table are hypertext links. Change bars apply only to the most recent version of the document.

Date	Version	Description	Page
March 12, 2007	Revision 1	Removed ‘Preliminary’ designation.	–
February 26, 2004	Preliminary	Initial release	–

1. Overview

1.1 Introduction

The Universal Asynchronous Receiver/Transmitter UART750 core contains a universal asynchronous receiver/transmitter (UART) and an optional interface to the IBM PowerPC® On-Chip Peripheral Bus (OPB). The UART750 cores are available in IBM's Cu-08 library as a synthesizable core. This means that the deliverables include Verilog rtl source files and a synthesis script. It is the customer's responsibility to create the gate level netlist.

UART750 is the base UART function with an optional FIFO size.

UART750OPB is a shell providing an optional interface to the IBM PowerPC OPB.

1.2 General Description

The UART750 performs serial-to-parallel conversion on data received from a peripheral device or a modem, and parallel-to-serial conversion on data received from the processor. The optional UART750OPB is a shell providing interfacing logic for communication to OPB bus, and it instantiates the UART750.

The processor can read the complete status of the UART750 at any time during functional operation. Status information reported includes the type and condition of the transfer operations being performed by the UART750 as well as any error conditions, such as parity, overrun, framing, and break interrupt.

The UART750 core has a configuration similar to the NS16450 in character mode (on power-up, it will be in character mode). The UART750 can run in FIFO mode to relieve the processor of excessive software overhead. In this case, internal FIFOs are activated allowing 16 or 64 bytes (plus three bits per byte of error data in the RCVR FIFO) to be stored in both receive and transmit modes.

In FIFO mode, a selectable auto flow control feature can use the CTS_N and RTS_N input and output signals to reduce software overhead by automatically controlling the flow of serial data.

The UART750 core includes a programmable baud rate generator, part of the UART function, that is capable of dividing the timing reference clock, XTAL, by a divisor of from 1 to (2^{16}-1), and produces a 16x clock for timing the internal transmitter logic. This 16x clock is also used to drive the receiver logic.

The UART750 function has complete modem-control capability and a processor interrupt system. Interrupts can be programmed to the user's requirements, minimizing the computing required to handle the communications link.

The UART750 core can also be used as a peripheral device on the IBM PowerPC OPB. The UART750OPB is a shell that instantiates the UART750 and allows attachment to the OPB. It contains OPB connection functions and also contains optional enhanced DMA support for the receiver and transmitter.

The schematic symbol for the UART750 is shown in *Figure 5* on page 33 (without the OPB Interface and the enhanced DMA options signal lines).

The schematic symbol for the UART750OPB is shown in *Figure 6* on page 37.

1.3 Features

- Choice of 16- or 64-byte FIFOs to buffer transmitter and receiver data
- Programmable auto flow feature where data flow is controlled by RTS_N and CTS_N signals
- Register conformance, after reset, to the configuration of the NS16450 register set
- Complete status reporting capability
- Full prioritized interrupt system controls
- Independently controlled transmit, receive, line status, and data set interrupts
- Programmable baud rate generator divides XTAL input clock by a divisor from 1 to (2^{16}-1) and generates the 16x clock:

  Baud rate (bps) = (XTAL frequency) / (16 x decimal divisor)
- Software modem control functions (CTS, RTS, DSR, DTR, RI, and DCD)
- Ability to add/delete standard asynchronous communication bits such as start, stop, and parity to/from the serial data
- Fully programmable serial-interface characteristics:
  - 5-, 6-, 7-, or 8-bit characters
  - Even, odd, or no-parity bit generation and detection
  - 1-, 1.5-, or 2-stop bit generation
  - Variable baud rate
- Line break generation and detection and false start bit detection
- Internal diagnostic capability:
  - Loop-back controls for communications link fault isolation
  - Break, parity, overrun, framing error simulation
- OPB interface option with further optional DMA support.

1.4 References

The UART750 core supports OPB bus architectures, version 1.2.

2. Functional Description

2.1 UART750 Functional Description

The UART750 block diagram is illustrated in *Figure 1* on page 10. This block diagram illustrates the basic UART750 operation without the OPB interface and enhanced DMA options.

During normal operation, data is received in parallel form on the XDI input bus and fed to the transmit FIFO (in FIFO mode) or the transmit holding register (in character mode). The data is then routed through a data selector and applied to the transmit shift register that converts the parallel data to a serial data stream. The serial data is then applied to interfacing functions via the serial output (SOUT) output line.

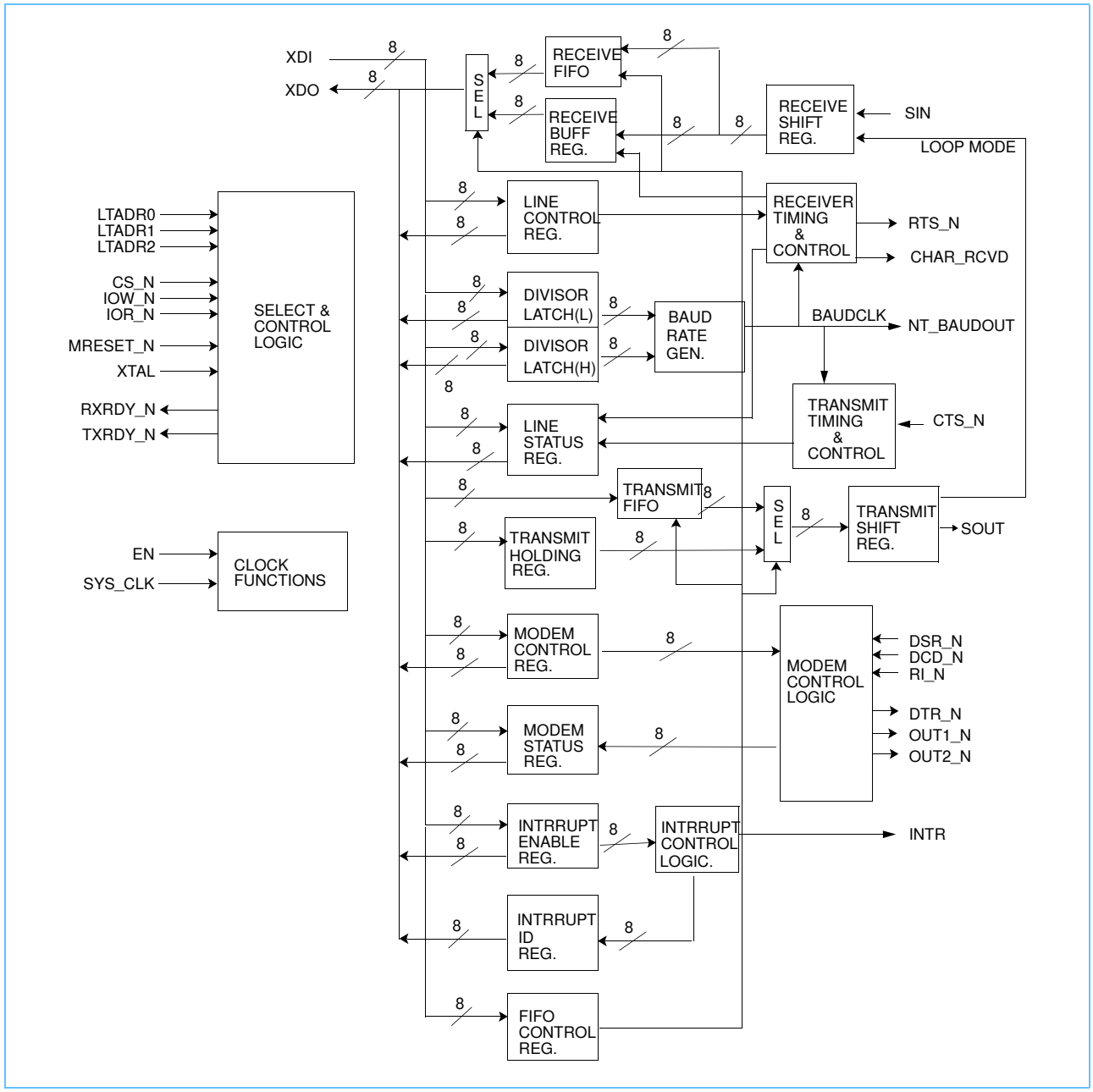
As a complementary function, serial data is received on the serial data in (SIN) line and applied to the receive shift register which converts the serial data stream to parallel format. The parallel data is then applied to the receive FIFO and receive buffer register. Depending on the mode of operation, the data is then fed from the FIFO or receive buffer register to a data selector for forwarding to the interfacing processor.

When the UART750 core is first powered up, register resets place it into character mode (sometimes called 16450 mode). In character mode, single characters are received and transmitted to the interfacing processor. This mode does not use buffering or temporary data storage but processes each character on a one-for-one basis.

In a typical application, following power-up, an initialization routine sets the UART750 core into FIFO mode. In this mode, serial data is received in a data stream consisting of multiple characters, and the data is temporarily stored in FIFO registers and sent out sequentially.

FIFO mode can operate under auto flow control or in either interrupt or polled mode.

Figure 1. UART750 Block Diagram



2.2 Baud Rate Generator

The UART750 core contains a programmable baud rate generator that is capable of dividing the input XTAL clock by any divisor from 1 to (2^{16}-1). The output frequency of the baud rate generator is the XTAL frequency/divisor. The internal generated clock is provided as a primary output pin, NT_BAUDOUT, that outputs a pulse at the divided XTAL clock frequency.

Two 8-bit divisor latch registers store the divisor in a 16-bit binary format. These divisor latch registers must be loaded during initialization to help ensure proper operation of the baud rate generator. The divisor latch registers are called MLL and DLL for the most significant bits and the least significant bits, respectively. The generated Baud rate is:

Baud rate (bps) = (XTAL frequency) / (16 x decimal divisor)

The UART treats the XTAL input as asynchronous data that is sampled by SYS_CLK to determine its period, and it is important that SYS_CLK be able to detect each positive and each negative pulse of XTAL sucessfully. Therefore, the XTAL clock must have a frequency and duty cycle that ensures SYS_CLK rises at least once each time the XTAL is low and once each time XTAL is high. This calculation must include duty cycle and phase jitter on both clocks. Setting XTAL frequency = (1/2) x SYS_CLK frequency is not sufficient to meet this requirement. It is more likely that XTAL frequency = (1/3)x SYS_CLK frequency will be more successful.

2.3 Auto Flow Control

Auto flow control can start and stop transmission of data when operating in FIFO mode. Auto flow control operates through the use of auto clear to send CTS_N and auto request to send RTS_N signals. With auto CTS_N, CTS_N must be active before the transmit FIFO can send data. With auto, RTS_N, RTS_N becomes inactive when the receiver FIFO has reached trigger level and inactive when the FIFO is empty.

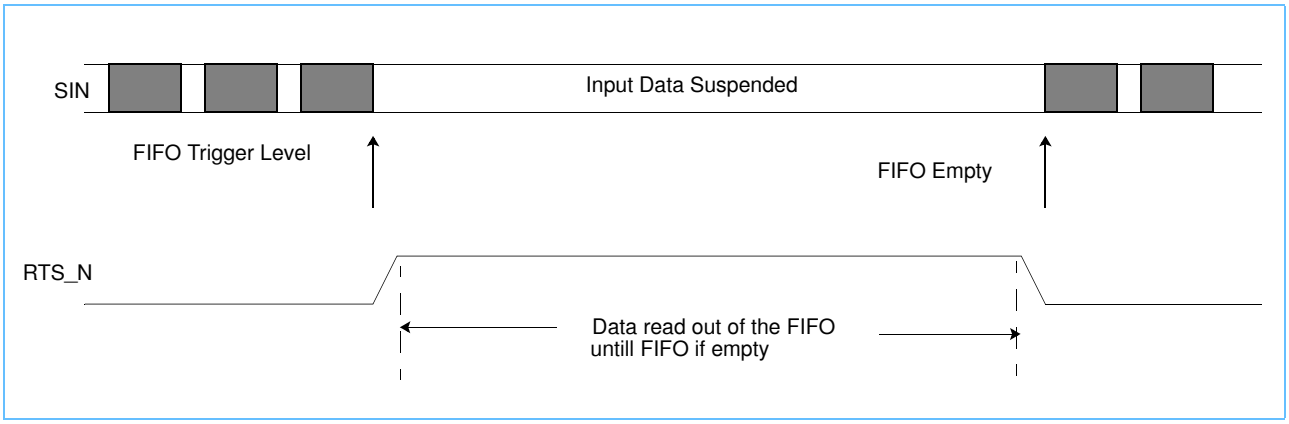
When RTS_N is connected to CTS_N of another UART core, data transmission is stopped when RTS_N becomes inactive and does not resume until the receive FIFO is empty and reasserts RTS_N. Therefore, overrun errors that would normally occur when the data rate exceeds the receiver FIFO/processor latency are eliminated.

2.3.1 Auto RTS_N

Auto RTS_N data flow control, shown in *Figure 2* on page 12 originates in the receiver control circuitry and is linked to the receiver FIFO trigger level. When the receiver FIFO level reaches the programmed trigger level (bits 6 and 7 in FIFO control register), RTS_N is deasserted. The sending UART core could be in the process of sending a byte, so an additional byte might be received after RTS_N is deasserted. When RTS_N is deasserted and the sending UART core stops sending data, an interrupt is generated and the interfacing processor initiates processing to extract accumulated data from the receive FIFO. When all data in the FIFO has been read out and the FIFO is empty, RTS_N is reasserted to initiate further serial input from the interfacing UART.

RTS_N is reasserted once the receiver FIFO is emptied by reading the receiver buffer register. This signals the sending UART core to continue transmitting data.

Figure 2. RTS_N Auto Flow Control

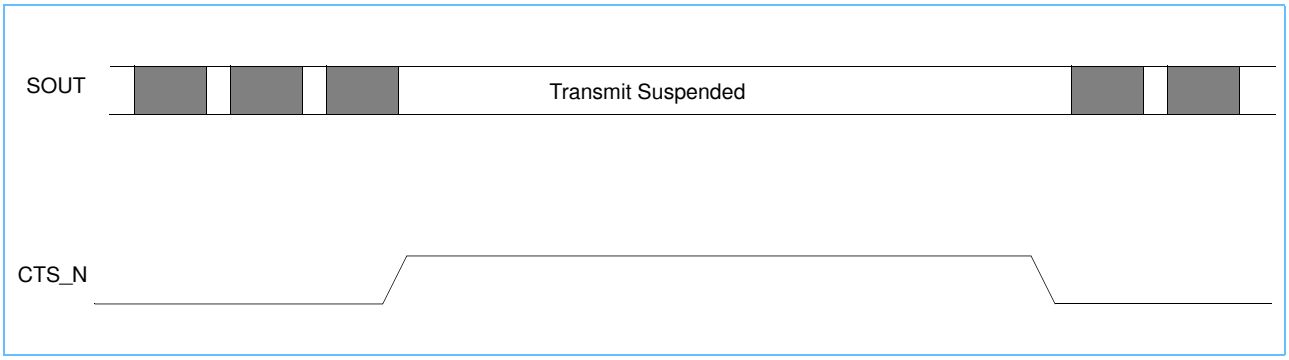


2.3.2 Auto CTS_N

When Auto CTS_N is active (see *Figure 3*), the transmitter checks CTS_N before sending the next data byte. To stop the transmitter, CTS_N must become inactive before the middle of the last stop bit that is currently being sent, otherwise the transmission of the next byte is initiated. Without auto CTS_N, the transmitter sends data until the transmit data is exhausted.

The auto CTS_N function also reduces interrupts to the host system. When auto flow control is enabled, changes in CTS_N do not trigger host interrupts because the transmitter is automatically started and stopped with the flow control function.

Figure 3. CTS_N Auto Flow Control



2.4 FIFO Operation

The transmitter and receiver FIFOs can operate in either interrupt or polled mode of operation. In interrupt mode, signals developed internal to the UART750 core generate interrupts to the interfacing processor, necessitating the processor to provide certain processing tasks required to service the interrupts. In polled mode, the interrupts are disabled and the processor must check the LSR register to see if UART service is required.

2.4.1 Interrupt Mode

2.4.1.1 Receiver

Receiver interrupts occur when the receiver FIFO and receiver interrupts are enabled by setting the FIFO control register (FCR) bit 0 and the interrupt enable register (IER) bit 0 to logic '1'.

The IIR received data available interrupt is issued when the number of characters in the FIFO has reached the trigger level programmed into the FCR. This interrupt will be reset to logic '0' when the FIFO character count drops below this trigger level upon readout of the accumulated FIFO characters.

The receiver line status interrupt (IIR = C6) is a top priority interrupt; the received data available interrupt (IIR = C4) is a second priority interrupt.

The data ready bit (bit 0 of LSR) is set as soon as a character is transferred from the shift register to the receiver FIFO. This bit is reset when the FIFO is empty.

Receiver timeout interrupts will occur as described below when the receiver FIFO and receiver interrupts are enabled by setting FCR bit 0 and IER bit 0 to logic '1'.

A FIFO timeout will occur when at least one character is in the receiver FIFO, no serial characters have been received for four serial character time periods, and the processor has not read the FIFO for four serial character time periods. A serial character time period is:

1/(baud rate) x (# start bits + word length + # parity bits + # stop bits)

For example, the serial character time period for an 8-bit word with one parity bit and two stop bits at 56 K baud is:

1/(56000) x (1 + 8 + 1 + 2) = 214.3 μs

The timeout would occur after 857.1 μs, if the above conditions hold.

When a timeout interrupt has occurred, it is cleared and its timer reset when the processor reads one character from the receiver FIFO.

When a timeout interrupt has *not* occurred, it's timer is reset after a new serial character is received or the processor reads the receiver FIFO.

2.4.1.2 Transmitter

Transmitter interrupts occur when the transmitter FIFO and transmitter interrupts are enabled by setting FCR0 and IER1 to logic '1'.

The transmitter holding register interrupt (IIR = C2) occurs when the transmitter holding register is empty; it is cleared as soon as the transmitter holding register is written to or the IIR is read. One to 16/64 characters may be written to the transmitter FIFO while servicing this interrupt.

The transmitter FIFO empty indications are delayed by one character time minus the last stop bit time whenever the following event occurs: THRE = 1 and there were less than two bytes simultaneously present in the transmit FIFO since the last THRE = 1. If bit 0 of the FCR is 1 (FIFOs enabled), the first transmitter interrupt after changing this bit (bit 0 of the FCR) is immediate.

RCVR FIFO trigger level interrupts, received data available interrupts, and character timeouts all have equivalent second interrupt priority. Current transmitter holding register empty interrupt and XMIT FIFO empty have equivalent third interrupt priority.

2.4.2 Polled Mode

When the FIFOs are enabled (FCR0 = 1) and all interrupts are disabled (IER0 to IER3 = '0000'), the UART750 core is in the FIFO polled mode of operation. Since the receiver and transmitter are controlled separately, either can be in polled mode of operation. In polled mode, the user program must check the LSR to see the status of the receiver and/or transmitter.

Bits 1–4 of the LSR specify if any errors have occurred. In polled mode, character status errors are handled in the same way as when configured for interrupt mode. Since the receiver line status interrupt is disabled in polled mode (IER2 = 0), the IIR is not affected. Bit 0 of the LSR is set as long as there is at least one character in the receiver FIFO. Bit 5 of the LSR indicates whether the transmitter FIFO is empty. Bit 6 of the LSR indicates whether the transmitter FIFO and the transmitter shift register are empty. Bit 7 of the LSR indicates whether there are any errors in the receiver FIFO.

In FIFO polled mode, there are no character timeout or trigger levels, however, the FIFOs are still capable of holding characters.

2.5 Modem Control Signals

There are three delay times associated with the modem control signals:

- Delay from write (WR) to output: Associated with RTS and DTR signals, this is the amount of time required to write from the CPU bus to the modem control register to indicate a request to send or data terminal ready indication. The maximum time required for this to occur is five SYS_CLK cycles. If SYS_CLK runs at 50 MHz, this amounts to 100 ns.
- Delay from read (RD) to reset interrupt: Associated with read MSR, this is the amount of time required to reset an interrupt by reading an appropriate register. The maximum time required for this to occur is four SYS_CLK cycles.
- Delay from Modem input to set interrupt: Associated with CTS, DSR, and DCD, this is the amount of time required to set the interrupt line once one of the input modem signals has gone active. The **maximum** time required for this to occur is four SYS_CLK cycles.

2.6 UART750OPB Shell Functional Description

The UART750OPB block diagram is illustrated in *Figure 6* on page 37. This block diagram shows the UART750OPB operation and enhanced DMA functions.

2.6.1 OPB Base Address Decode and Chip Select

The OPB interface contains a 29-bit base address port, OPB_BASE_ABUS(0:28), which is compared to OPB_ABUS(0:28) as part of the address decode. If these inputs match, the core allows the OPB_SELECT signal to activate the UART750 chip select signal, CS_N, and OPB_ABUS(29:31) are used to generate the UART750 address inputs LTADR2, LTADR1, and LTADR0.

- OPB_ABUS(29) → LTADR2
- OPB_ABUS(30) → LTADR1
- OPB_ABUS(31) → LTADR0

The UART750 chip select CS_N is also activated during DMA cycles using enhanced DMA support. See *Section 2.6.4 Enhanced DMA Support* on page 15.

2.6.2 OPB Bus Operation

The UART750OPB entity has been designed to interface to the OPB with two-cycle latency for read and write transactions. *Figure 12* on page 46 and *Figure 13* on page 46 show that the OPB timing requires two SYS_CLK cycles to complete a read or write transaction. The address and data must be valid for the times shown with respect to SYS_CLK. OPB_XFERACK, and OPB_DBUS_EN (for read) are asserted by the UART750OPB on the second SYS_CLK cycle.

The target registers for read and write are really in the UART750 that is instantiated by the OPB shell. If the OPB address decode is successful, the interface logic assigns values to the UART750 address and data inputs, and then activates the IOR_N or IOW_N input in order to complete the read or write transaction. Because the UART750 read and write operation requires a minimum of three cycles, there must be at least one cycle of separation between consecutive OPB transactions.

Also, once a UART750 read or write is started, it cannot be undone. UART data would be lost or overwritten. Therefore this core cannot support aborted master OPB read or write transactions without loss of UART data.

2.6.3 Other OPB Functions

The master reset, MRESET_N, is synchronized in the OPB shell. The UART750 signal, XTAL, is renamed to SERIAL_CLK by the OPB shell for descriptive purposes. The name XTAL is maintained by the UART750 core to be consistent with the stand alone UART pin name where the serial clock is supplied by a crystal oscillator.

2.6.4 Enhanced DMA Support

During enhanced DMA operation, the user defined core address corresponding to OPB_ABus(0 to 28) is ignored and the register address to the UART750 address is forced to '000'. This accesses the RBR and THR registers.

The UART750 chip select, CS_N, is asserted while the internal signal, DMA_Decode, is asserted.

The following equations represent the logic that generates the read and write inputs to the UART750. The internal signal XFEREND is a delayed version of XFERACK to signal that the transaction is complete.

DMA_Decode <= (TX_DMA_ENABLE & TX_DMA_ACK & TX_DMA_ACK_delayed) or (RX_DMA_ENABLE & RX_DMA_ACK & RX_DMA_ACK_delayed);

cs_n1 <= not (OPB_select) when (OPB_ABus(0 to 28) = *OPB base address[0:28]* else ‘1’;

CS_N <= not (not cs_n1 or dma_decode);

IOR_N <= CS_N or XFEREND or (not (OPB_SELECT and OPB_RNW)) and not (RX_DMA_ACK and RX_DMA_ENABLE);

IOW_N <= CS_N or XFERACK or XFEREND or (not (OPB_SELECT and not OPB_RNW)) and not (TX_DMA_ACK and TX_DMA_ACK_delayed and TX_DMA_ENABLE));

For a detailed description of enhanced DMA support signals, see *Section 4.6 Enhanced DMA Support Interface Signals* on page 40.

3. Software Interface

3.1 Registers

The registers in the UART750 core are accessed for writing from the XDI bus and for reading from the XDO bus.

3.1.1 Address Map

Table 1 lists the programmable registers in the UART750 core. This table shows a bit map summary of the registers with individual bit names identified. For detailed information on each bit, refer to the individual register tables contained in this databook.

When addressing the registers, address bits LTADR2 through LTADR0 received on the address bus from the interfacing processor, are prefixed by the divisor latch access bit (DLAB), bit 7 in the line control register. For example, if the address bits were all zeros (LTADR2:0 = 000) and DLAB was also zero, the addressing, equal to 0000, would point to RBR or THR according to whether a read or write is being performed. If DLAB = 1, the composite address, 1000, would then point to the divisor latch register (LSB).

Table 1. Summary of UART750 Registers (Page 1 of 2)

Register Name	Alias	Address[1]	Bit Name							
			LSb							MSb
			0	1	2	3	4	5	6	7
Receiver Buffer Register (read only)	RBR	0000	Data Bit 0	Data Bit 1	Data Bit 2	Data Bit 3	Data Bit 4	Data Bit 5	Data Bit 6	Data Bit 7
Transmitter Holding Register (write only)	THR	0000	Data Bit 0	Data Bit 1	Data Bit 2	Data Bit 3	Data Bit 4	Data Bit 5	Data Bit 6	Data Bit 7
Interrupt Enable Register (read and write)	IER	0001	Enable Receive Data Available Interrupt (ERBFI)	Enable Transmitter Holding Register Empty Interrupt (ETBEI)	Enable Receiver Line Status Interrupt (ELSI)	Enable Modem Status Interrupt (EDSSI)	0	0	0	0
Interrupt Identification Register (read only)	IIR	X010	"0" if Interrupt Pending	Interrupt ID Bit 0	Interrupt ID Bit 1	Interrupt ID Bit 2	0	0	FIFOs Enabled[3]	FIFOs Enabled[3]
FIFO Control Register (write only)	FCR	X010	FIFO Enable	RCVR FIFO Reset	XMIT FIFO Reset	DMA Mode Select	Reserved	Reserved	RCVR Trigger (LSB)	RCVR Trigger (MSB)
Line Control Register (read and write)	LCR	X011	Word Length Select Bit 0 (WLS0)	Word Length Select Bit 1 (WLS1)	Number of Stop Bits (STB)	Parity Enable (PEN)	Even Parity Select (EPS)	Stick Parity	Set Break	Divisor Latch Access Bit (DLAB)

1. DLAB concatenated with LTADR2 through LTADR0. An x in the register address represents a don't care state for the DLAB bit.
2. Bit 0 is the least significant bit. It is the first bit serially transmitted and received.
3. This bit is always zero in character mode.

Table 1. Summary of UART750 Registers (Page 2 of 2)

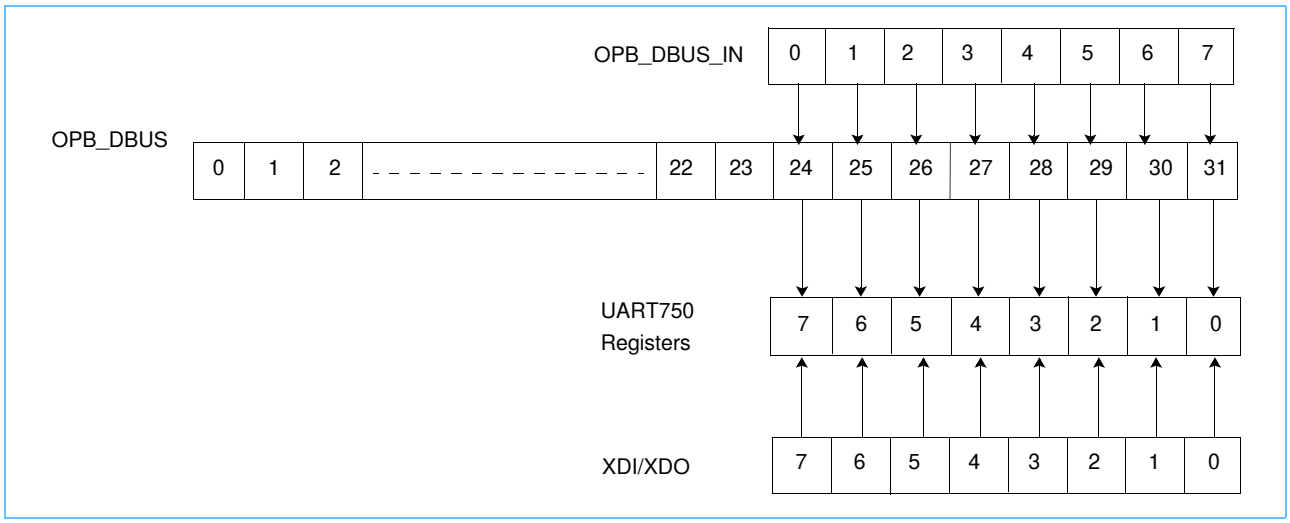
Register Name	Alias	Address[1]	Bit Name							
			LSb							MSb
			0	1	2	3	4	5	6	7
Modem Control Register (read and write)	MCR	X100	Data Terminal Ready (DTR)	Request To Send (RTS)	Out 1	Out 2	Loop	Flow Control Enable (AFC)	0	0
Line Status Register (read and write)	LSR	X101	Data Ready (DR)	Overrun Error (OE)	Parity Error (PE)	Framing Error (FE)	Break Interrupt (BI)	Transmitter Holding Register Empty (THRE)	Transmitter Empty (TEMT)	Error in Receiver FIFO
Modem Status Register (read and write)	MSR	X110	Delta Clear To Send (DCTS)	Delta Data Set Ready (DDSR)	Trailing Edge Ring Indicator (TERI)	Delta Data Carrier Detect (DDCD)	Clear To Send (CTS)	Data Set Ready (DSR)	Ring Indicator (RI)	Data Carrier Detect (DCD)
Scratchpad Register (read and write)	SCR	X111	Data Bit 0	Data Bit 1	Data Bit 2	Data Bit 3	Data Bit 4	Data Bit 5	Data Bit 6	Data Bit 7
Divisor Latch (LSB) (read and write)	DLL	1000	Data Bit 0	Data Bit 1	Data Bit 2	Data Bit 3	Data Bit 4	Data Bit 5	Data Bit 6	Data Bit 7
Divisor Latch (MSB) (read and write)	DLM	1001	Data Bit 8	Data Bit 9	Data Bit 10	Data Bit 11	Data Bit 12	Data Bit 13	Data Bit 14	Data Bit 15

1. DLAB concatenated with LTADR2 through LTADR0. An x in the register address represents a don't care state for the DLAB bit.
2. Bit 0 is the least significant bit. It is the first bit serially transmitted and received.
3. This bit is always zero in character mode.

3.1.2 Bit Organization

In the register summary, *Table 1* on page 17, the bit numbers of the UART input and output data correspond to the same bit numbers of the register table. For example, XDI7 corresponds to register bit 7, and XDI0 corresponds to register bit 0. When connected to the OPB interface, the bit numbers are reversed such that OPB_DBUS_IN(0) corresponds to register bit 7, and OPB_DBUS_IN(7) corresponds to register bit 0. This is illustrated in *Figure 4*. Also see *Section 4.5.1 OPB Interface I/O Signals* on page 38.

Figure 4. Bit Organization



3.1.3 Register Reset and Access Conditions

Table 2 lists the reset condition for registers and selected control signals.

Table 2. Reset Configuration for UART750

Register/Signal	Register Address DLAB[2], LTADR(2:0)	Reset Control	Reset State Bits (7 down to 0)
Interrupt Enable Register	0001	MasterReset	0000 0000
Interrupt Identification Register	x010	MasterReset	0000 0001
FIFO Control Register	x010	MasterReset	0000 0000
Line Control Register	x011	MasterReset	0000 0000
Modem Control Register	x100	MasterReset	0000 0000
Line Status Register	x101	MasterReset	0110 0000
Modem Status Register	x110	MasterReset	xxxx 0000[1]
Divisor Latch LSB	1000	MasterReset	0000 0000
Divisor Latch MSB	1001	MasterReset	0000 0000
SOUT	-	MasterReset	1
INTR (Receiver Errors)	-	MasterReset or Read LSR	0
INTR (Receiver Data Ready)	-	MasterReset or Read RBR	0
INTR (THRE)	-	MasterReset or Read IIR or Write THR	0
INTR (Modem Status Changes)	-	MasterReset or Read MSR	0
OUT1_N	-	MasterReset	1
OUT2_N	-	MasterReset	1
RTS_N	-	MasterReset	1
DTR_N	-	MasterReset	1
Receiver FIFO	-	MasterReset, or FCR bits 0 and 1 = 1, or change on FCR bit 0	Pointers reset to 0, Reg bits - No Effect
Transmitter FIFO	-	MasterReset, or FCR bits 0 and 1 = 1, or change on FCR bit 0	Pointers reset to 0, Reg bits - No Effect
Scratchpad Register	x111	Master Reset	No Effect

1. For the reset state of the modem status register bits, the four high order bits (7 through 4) are controlled by prime input signals to the UART750 (DCD, RI, DSR, and CTS).
2. x in the register address represents a don't care state for the DLAB bit.

3.2 Register Descriptions

3.2.1 Interrupt Enable Register

Address = 0001

Five types of interrupts are enabled via the interrupt enable register (IER). The interrupt (INTR) output signal can be activated by any of the five interrupt types. Resetting bits 0 through 3 of the IER totally disables the interrupt system. Each interrupt type can be enabled by setting one of these bits. Disabling an interrupt prevents it from activating the INTR output signal and to prevents it from being shown as active in the IIR. All other system functions, LSR, and MSR continue to operate in their normal manner.

Table 3. Interrupt Enable Register Description

Bit #	Bit Name	Value	Bit Description
0	Enable Receive Data Available Interrupt (ERBFI)	0	Received data available interrupt (and timeout interrupts in FIFO mode) disabled
		1	Received data available interrupt (and timeout interrupts in FIFO mode) enabled
1	Enable Transmitter Holding Register Empty Interrupt (ETBEI)	0	Transmitter holding register empty interrupt disabled
		1	Transmitter holding register empty interrupt enabled
2	Enable Receiver Line Status Interrupt (ELSI)	0	Receiver line status interrupt disabled
		1	Receiver line status interrupt enabled
3	Enable Modem Status Interrupt (EDSSI)	0	Modem status interrupt disabled
		1	Modem status interrupt enabled
4	Not used	0	Always 0
5	Not used	0	Always 0
6	Not used	0	Always 0
7	Not used	0	Always 0

3.2.2 Interrupt Identification Register

Address = x010

The UART750 core prioritizes interrupts into four levels which are recorded in the interrupt identification register (IIR). The levels of interrupt (bits 1, 2, and 3) in the order of their priority are receiver line status, received data ready, transmitter holding register empty, and modem status. When the processor accesses the IIR, the UART750 core records new interrupts but does not change its current contents until the access by the processor is complete. The UART750 core indicates the highest priority interrupt pending to the processor via the IIR.

Table 4. Interrupt Identification Register Description

Bit	Bit Name	Value			Bit Description			
0	Pending Interrupt	0			Interrupt is pending. IIR contents may be used as a pointer to the appropriate interrupt service routine.			
		1			No interrupt is pending.			
1, 2 and 3	Interrupt ID	Bit 3	Bit 2	Bit 1	Bits 1 and 2 are used to indicate the interrupt priority as shown below. Bit 3 is always 0 in 16450 mode. In FIFO mode, when a timeout interrupt is pending, bits 2 and 3 are set to logic 1.			
					Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
		0	1	1	1st	Receiver line status	Overrun, parity or framing error, or break interrupt	Read LSR
		0	1	0	2nd	Received data available	Receiver data available or trigger level reached.	Read RBR, or FIFO drops below trigger level.
		1	1	0	2nd	Character timeout indication	No characters have been removed from or input to the receiver FIFO during the last four char. times and it contains at least one char. during this time.	Read RBR
		0	0	1	3rd	Transmitter holding register empty	Transmitter holding register empty	Read IIR (if source of interrupt) or write THR
		0	0	0	4th	Modem status	Clear to send, data set ready, ring indicator or data carrier detect	Read MSR
4	Not used	0			always 0			
5	Not used	0			always 0			
6	FIFOs enabled	0			FIFOs disabled (bit 0 of FCR = 0)			
		1			FIFOs enabled (bit 0 of FCR = 1)			
7	FIFOs enabled	0			FIFOs disabled (bit 0 of FCR = 0)			
		1			FIFOs enabled (bit 0 of FCR = 1)			

3.2.3 FIFO Control Register

Address = x010

The FIFO control register (FCR) has the same address as the IIR and is a write-only register. This register is used to perform FIFO control operations such as selecting the type of DMA signaling, setting the receiver FIFO trigger levels, clearing the FIFOs, and enabling the FIFOs.

Table 5. FIFO Control Register Description

Bit	Bit Name	Value	Bit Description	
0	FIFO enable	0	FIFO disabled. Resets both receiver and transmitter FIFOs. FiFO pointers are set to 0 when changing to and from FIFO and 16450 modes.	
		1	FIFO enable. Writing a logic 1 here enables both the receiver and transmitter FIFOs.	
1	RCVR FIFO reset	0	Receiver FIFO reset. The 1 that is written into this position is self-clearing.	
		1	Receiver FIFO reset. A logic 1 written here will reset all of its counter logic to 0. The receiver shift register is not cleared by this bit.	
2	XMIT FIFO reset	0	Transmitter FIFO reset. The 1 that is written into this position is self-clearing.	
		1	Transmitter FIFO reset. A logic 1 written here will reset all of its counter logic to 0. The transmitter shift register is not cleared by this bit.	
3	DMA mode select	0	DMA mode select. If bit 0 of this register is 1, FIFOs are enabled; if set to logic 0, RXRDY and TXRDY outputs will change from DMA mode 1 to DMA mode 0 (single transfer DMA).	
		1	DMA mode select. If bit 0 of this register is 1, FIFOs are enabled; if set to logic 1, RXRDY and TXRDY outputs will change from DMA mode 0 to DMA mode 1 (multiple transfer DMA).	
4	Reserved	-	Reserved	
5	Reserved	-	Reserved	
		Bit 7:6	16 Byte Receiver FIFO Trigger Level	64 Byte Receiver FIFO Trigger Level
6:7	RCVR trigger Bit 6=LSB, Bit 7=MSB	00	01	01
		01	04	16
		10	08	32
		11	14	56

3.2.4 Line Control Register

Address = x011

The system programmer uses the line control register (LCR) to specify the format of the asynchronous data communications exchange and to set the divisor latch access bit. The contents of the LCR can also be read by the processor. The read capability simplifies system programming and eliminates the need for separate storage of the line characteristics in system memory.

Table 6. Line Control Register Description

Bit	Bit Name	Value	Bit Description
0:1	Word Length	Bit 1:0	
		00	Character Word Length = 5
		01	Character Word Length = 6
		10	Character Word Length = 7
		11	Character Word Length = 8
2	Number of stop bits (STB)	0	Specifies one stop bit transmitted and received in each serial character. The receiver checks the first stop bit only, regardless of how many stop bits are selected.
		1	If the character word length = 5, then one and one half stop bits are generated. If any other character length is selected, two stop bits are generated. The receiver checks the first stop bit only, regardless of how many stop bits are selected.
3	Parity enable (PEN)	0	Parity enable bit. Parity checking is disabled.
		1	Parity enabled. A parity bit is generated during transmission of data (or checked during the reception of data) between the last data word bit and the stop bit of the serial data. The parity bit is used to produce an odd or even number of 1's when the data word bits and the parity bit are summed.
4	Even parity select (EPS)	0	Odd parity select. When bit 3 is a logic 1, an odd number of logic 1's is transmitted or checked in the data word and the parity bits.
		1	Even parity select. When bit 3 is a logic 1, an even number of logic 1's is transmitted or checked in the data word and the parity bits.
5	Stick parity	0	Sticky parity bit, stick parity is disabled.
		1	Sticky parity bit. When bits 3 and 4 are logic 1, the parity bit is transmitted and checked as logic 0. If bit 3 is logic 1 and bit 4 is a logic 0, then the parity bit is transmitted and checked as logic 1.
6	Set break	0	Break control bit, the break is disabled.
		1	Break control bit. When the core is transmitting, it causes a break condition to be transmitted from the UART750 core. SOUT is forced to the spacing state (logic 0). This bit acts only on SOUT and has no effect on the transmitter logic.
7	Divisor latch access bit (DLAB)	0	Divisor latch access bit, DLAB. Set = 0 to address RBR, THR, and IER.
		1	Divisor latch access bit. Set = 1 to address divisor latches.

3.2.5 Modem Control Register

Address = x100

The interface between the modem, data set, or peripheral device emulating a modem, and the UART750, is controlled by the modem control register (MCR).

Table 7. Modem Control Register Description

<table>
<tr><th>Bit</th><th>Bit Name</th><th>Value</th><th>Bit Description</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">Data terminal ready (DTR)</td><td>0</td><td>DTR_N (Data Terminal Ready, active low) output set to 1.</td></tr>
<tr><td>1</td><td>DTR_N (Data Terminal Ready, active low) output set to 0.</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Request to send (RTS)</td><td>0</td><td>RTS_N (Request To Send, active low) output set to 1.</td></tr>
<tr><td>1</td><td>RTS_N (Request To Send, active low) output set to 0.</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Out 1</td><td>0</td><td>OUT1_N (auxiliary user designated output) set to 1. When bit 2 is 0, this indicates the OUT1_N signal is not active.</td></tr>
<tr><td>1</td><td>OUT1_N (auxiliary user designated output) set to 0. When bit 2 is 1, this indicates the OUT1_N signal is active.</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">Out 2</td><td>0</td><td>OUT2_N (auxiliary user designated output) set to 1. When bit 3 is 0, this indicates the OUT2_N signal is not active.</td></tr>
<tr><td>1</td><td>OUT2_N (auxiliary user designated output) set to 0. When bit 3 is 1, this indicates the OUT2_N signal is active.</td></tr>
<tr><td rowspan="2">4</td><td rowspan="2">Loop</td><td>0</td><td>Loopback mode deactivated.</td></tr>
<tr><td>1</td><td>Loop-back mode activated. Provides a local loop-back feature for diagnostic testing of the UART750 core. The following occurs: SOUT is set to the marking state (logic 1); SIN is disconnected; the output of the transmitter shift register feeds the input of the receiver shift register (this provides the loop-back); the four modem control inputs DSR_N, CTS_N, RI_N, and DCD_N, are disconnected; the four modem control outputs, DTR_N, RTS_N, OUT1_N, and OUT2_N, are set to a logic 1 (their inactive state); the four modem control outputs are connected internally to the four modem control inputs, therefore, the data that is transmitted is immediately received. This allows the verification of the UART750 transmit and received data paths.
The receiver and transmitter interrupts are fully operational. Their sources are external to the UART750 core. Also operational are the modem control interrupts, but their source is now the lower four bits of the MCR instead of the modem control inputs to the UART750 core. The IER still controls the interrupts.</td></tr>
<tr><td>5</td><td>Flow control enable (AFC)</td><td></td><td>Auto Flow Control Enable: See table below. (Effective only if FIFOs are enabled)
<table>
<tr><th>MCR Bit 5 (AFC)</th><th>MCR Bit 1 (RTS_N)</th><th>Auto Flow Control</th></tr>
<tr><td>1</td><td>1</td><td>Auto RTS and auto CTS enabled</td></tr>
<tr><td>1</td><td>0</td><td>Auto CTS only enabled (RTS_N is inactive)</td></tr>
<tr><td>0</td><td>x</td><td>Auto CTS and auto CTS disabled (AFC disabled)</td></tr>
</table></td></tr>
<tr><td>6</td><td>Not used</td><td>0</td><td>Always 0</td></tr>
<tr><td>7</td><td>Not used</td><td>0</td><td>Always 0</td></tr>
</table>

3.2.6 Line Status Register

Address = x101

Information concerning the data transfer is held for the processor in the line status register. Bits 1 through 4 are conditions that produce a receiver line status interrupt whenever the condition corresponding to the active bit is detected and the interrupt is enabled. This register is intended for read operations only; writing is not recommended.

Table 8. Line Status Register Description (Page 1 of 2)

Bit	Bit Name	Value	Bit Description
0	Data ready (DR)	0	Receiver data ready (DR) indicator. Reset to 0 when all data has been read from the receiver FIFO or the receiver buffer register.
		1	Receiver data ready (DR) indicator. An entire incoming character has been received into the RBR or receiver FIFO.
1	Overrun error (OE)	0	Overrun error (OE) indicator. Reset to 0 whenever processor reads LSR.
		1	Overrun error (OE) indicator. Data in the RBR was not read by the processor before the next character was transferred into the RBR, hence the original data was lost. OE detected when this bit is 1. In FIFO mode, if the incoming data continues to fill the FIFO beyond the trigger level, an OE occurs only after the FIFO is completely full and the entire next character has been received in the receiver shift register. The processor is informed of the OE immediately upon occurrence. The character in the shift register is overwritten and is not transferred to the FIFO.
2	Parity error (PE)	0	Parity error (PE) indicator. Reset to 0 whenever processor reads LSR.
		1	Parity error (PE) indicator. Indicates that the received data character does not have the correct parity as determined by the even parity select bit of the LCR. Set to logic 1 upon detection of a parity error. In FIFO mode, this error is revealed to the processor when the character this error is associated with is at the top of the FIFO.
3	Framing error (FE)	0	Framing error (FE) indicator. Reset to 0 whenever processor reads LSR.
		1	Framing error (FE) indicator. Indicates that a valid stop bit was not found in the received character. Set to logic 1 whenever stop bit following the last data bit or parity bit is detected as logic 0 (spacing level). In FIFO mode, this error is revealed to the processor when the character this error is associated with is at the top of the FIFO. To resynchronize after a framing error, the UART750 assumes that the framing error was due to the next start bit, so it samples this “start bit” twice, then takes in the data.
4	Break interrupt (BI)	0	Break interrupt (BI) indicator. Reset to 0 whenever processor reads LSR.
		1	Break interrupt (BI) indicator. Set to logic 1 whenever the received data input is held at the spacing level (logic 0) for longer than a full word transmission time. The full word transmission time is the time required for the start bit, data bits (can be 5-8 bits), parity and stop bits. In FIFO mode, this error is revealed to the processor when the character this error is associated with is at the top of the FIFO. Only one zero character is loaded into the receiver FIFO when a break occurs. After SIN receives the next valid start bit, and has gone into the marking state, the next character transfer is enabled.
5	Transmitting holding register empty (THRE)	0	Transmitter holding register empty (THRE) indicator. Reset to 0 with the loading of the THR by the processor. In FIFO mode it is reset to 0 when at least one byte is written to the transmitter FIFO.
		1	Transmitter holding register empty (THRE) indicator. Logic 1 when the UART750 is ready to accept a new character for transmission. When the THRE enable (bit 1 in the IER) is set to logic 1, the UART750 will issue an interrupt to the processor. This bit is set to logic 1 when a character is transferred from the THR to the transmitter shift register. In FIFO mode, this bit is set when the transmitter FIFO is empty.

Table 8. Line Status Register Description (Page 2 of 2)

Bit	Bit Name	Value	Bit Description
6	Transmitter empty (TEMT)	0	Transmitter empty (TEMT) indicator. Reset to logic 0 whenever the THR or the transmitter shift register contain a character. In FIFO mode, it is reset to logic 0 whenever the transmitter FIFO or the transmitter shift register contain a character.
		1	Transmitter empty (TEMT) indicator. Set to logic 1 when the THR and the Transmitter shift register are both empty. In FIFO mode, it is set to logic 1 when the transmitter FIFO and the transmitter shift register are both empty.
7	Error in receiver FIFO	0	Receiver FIFO Error indicator. Always 0 in 16450 mode. In FIFO mode, it is reset to 0 whenever the processor reads the LSR, provided there are no subsequent errors in the FIFO.
		1	Receiver FIFO Error indicator. Set to 1 when there are one or more instances of parity error, framing error or break indication in the FIFO.

3.2.7 Modem Status Register

Address = x110

The processor can monitor the present state of the modem (or peripheral device) control lines by reading the modem status register (MSR). In addition, the MSR has four bits (delta bits) to indicate whether any of the modem's (or peripheral device's) control lines have changed state.

Table 9. Modem Status Register Description

Bit	Bit Name	Value	Bit Description
0	Delta clear to send (DCTS)	0	Reset to 0 whenever the processor reads the MSR.
		1	Delta clear to send (DCTS). Indicates that the Clear to Send (CTS_N) input to the UART750 has changed state since the processor last read the MSR. A modem status interrupt is generated. When Auto Flow Control in enabled (MCR5 = '1'), changes in DCD_N do not create interrupts.
1	Delta data set ready (DDSR)	0	Reset to 0 whenever the processor reads the MSR.
		1	Delta data set ready (DDSR). Indicates that the Data Set Ready (DSR_N) input to the UART750 has changed state since the processor last read the MSR. A modem status interrupt is generated.
2	Trailing edge ring indicator (TERI)	0	Reset to 0 whenever the processor reads the MSR.
		1	Trailing edge of ring indicator (TERI) detector. Indicates that the ring indicator (RI_N) input to the UART750 changed from 0 to 1 since the processor last read the MSR. A modem status interrupt is generated.
3	Delta data carrier detect (DDCD)	0	Reset to 0 whenever the processor reads the MSR.
		1	Delta data carrier detect (DDCD). Indicates that the data carrier detect (DCD_N) input to the UART750 has changed state since the processor last read the MSR. A modem status interrupt is generated.
4	Clear to send (CTS)	-	Complement of clear to send (CTS_N) input to the UART750. In loopback mode (bit 4 of MCR is 1), it is equivalent to RTS (bit 1) in the MCR.
5	Data set ready (DSR)	-	Complement of data set ready (DSR_N) input to the UART750. In loopback mode (bit 4 of MCR is 1), it is equivalent to DTR (bit 0) in the MCR.
6	Ring indicator (RI)	-	Complement of ring indicator (RI_N) input to the UART750. In loopback mode (bit 4 of MCR is 1), it is equivalent to OUT 1 (bit 2) in the MCR.
7	Data carrier detect (DCD)	-	Complement of data carrier detect (DCD_N) input to the UART750. In loopback mode (bit 4 of MCR is 1), it is equivalent to OUT 2 (bit 3) in the MCR.

3.2.8 Scratchpad Register

A scratchpad register, intended for use by the programmer as a temporary data holder, is provided in this UART750 core. It does not control the UART750 operation in any way.

3.2.9 Divisor Latch Low Register

The divisor latch low register (DLL, address 1000) concatenated with the divisor latch high register (DLH + DLL) combine to determine the relationship between the baud rate and the XTAL clock by the following equation:

Baud rate (bps) = (XTAL frequency) / (16 x (DLM+DLL))

In order to access this register, the write or read operation must be preceded by writing a '1' to bit 7 of the line control register, LCR(7), the divisor latch access bit.

DLL7 is the high order bit of the DLL register.

3.2.10 Divisor Latch High Register

The Divisor Latch High Register (DLH, address 1001), concatenated with the divisor latch low register (DLH + DLL) combine to determine the relationship between the baud rate and the XTAL clock by the following equation:

Baud rate (bps) = (XTAL frequency) / (16 x (DLM+DLL))

In order to access this register, a write or read operation must be preceded by writing a ‘1’ to bit 7 of the line control register, LCR(7), the divisor latch access bit.

DLM7 is the high order bit of the DLM register.

3.3 Programming Examples

If the OPB interface option is used, the UART750 registers are programmed via OPB hexadecimal inputs organized as most significant (MS) HEX digit and a least significant (LS) HEX digit.

3.3.1 UART750 Register Hex Codes

Table 10. Hex Codes for Programming UART750 (Page 1 of 2)

Reg	MS Hex Digit		LS Hex Digit	
(r/w)	#	Description	#	Description
IER (r/w)	0	Always 0	8	Enable modem status register
			4	Enable RX line status register
			2	Enable THR empty status register
			1	Enable RX data available interrupt * (and timeout interrupts in FIFO mode)
IIR (r)	0	FIFOs disabled	C	Character timeout interrupt * (reset when RBR read)
			6	RX line status interrupt * (reset when LSR read)
	C	FIFOs enabled	4	RX data available interrupt * (reset when RBR read or trigger level)
			2	THR empty interrupt * (reset when IIR read or THR write)
			0	Modem status interrupt * (reset when MSR read)
FCR (w)	C	14/56 byte RX FIFO trigger level	8	Multiple Xfer DMA (DMA1)
	8	8/32 byte RX FIFO trigger level	4	Reset TX FIFO
	4	4/16 byte RX FIFO trigger level	2	Reset RX FIFO
	0	1/1 byte RX FIFO trigger level	1	Enable both FIFOs
LCR (r/w)	8	DLAB bit enabled	8	Parity enable
	4	Send break	4	2 stop bit (1.5 if 5 bit word)
	2	Stick parity * (parity bit 0 if parity set to even)	3	8 bit character length
			2	7 bit character length
	1	Even parity (Odd parity if 0)	1	6 bit character length
			0	5 bit character length

1. In this table, the most significant bit corresponds to XDI(7)/XDO(7) or OPB_DBUS_IN(0). See *Figure 4* on page 19. The data bit numbers of the OPB bus are reversed (see *Table 1* on page 17).
2. Some elements are mutually exclusive, that is, for LCR least significant hex digit, one of four character lengths must be chosen.
3. Elements not mutually exclusive can be added together in each digit, that is, if both DCD and RI indicators in MSR register are set, first hex digit will be C.
4. If not otherwise indicated, the absence of particular setting corresponds to 0.

Table 10. Hex Codes for Programming UART750 (Page 2 of 2)

Reg	MS Hex Digit		LS Hex Digit	
MCR (r/w)	1	Loopback mode	8	OUT2_N set to 0
			4	OUT1_N set to 0
			2	RTS_N set to 0
			1	DTR_N set to 0
LSR (r/w)	8	RX FIFO error * (> 1 Parity, framing or break errors)	8	Framing error indicator
	4	TX empty indicator	4	Parity error indicator
	2	TX holding register empty	2	Overrun error indicator
	1	RX break interrupt indicator	1	RX data ready indicator
MSR (r/w)	8	DCD (not DCD_N)	8	Delta DCD
	4	RI - Ring indicator	4	TERI - Trailing edge ring indicator
	2	DSR - Data set ready	2	Delta DSR
	1	CTS - Clear to send	1	Delta CTS

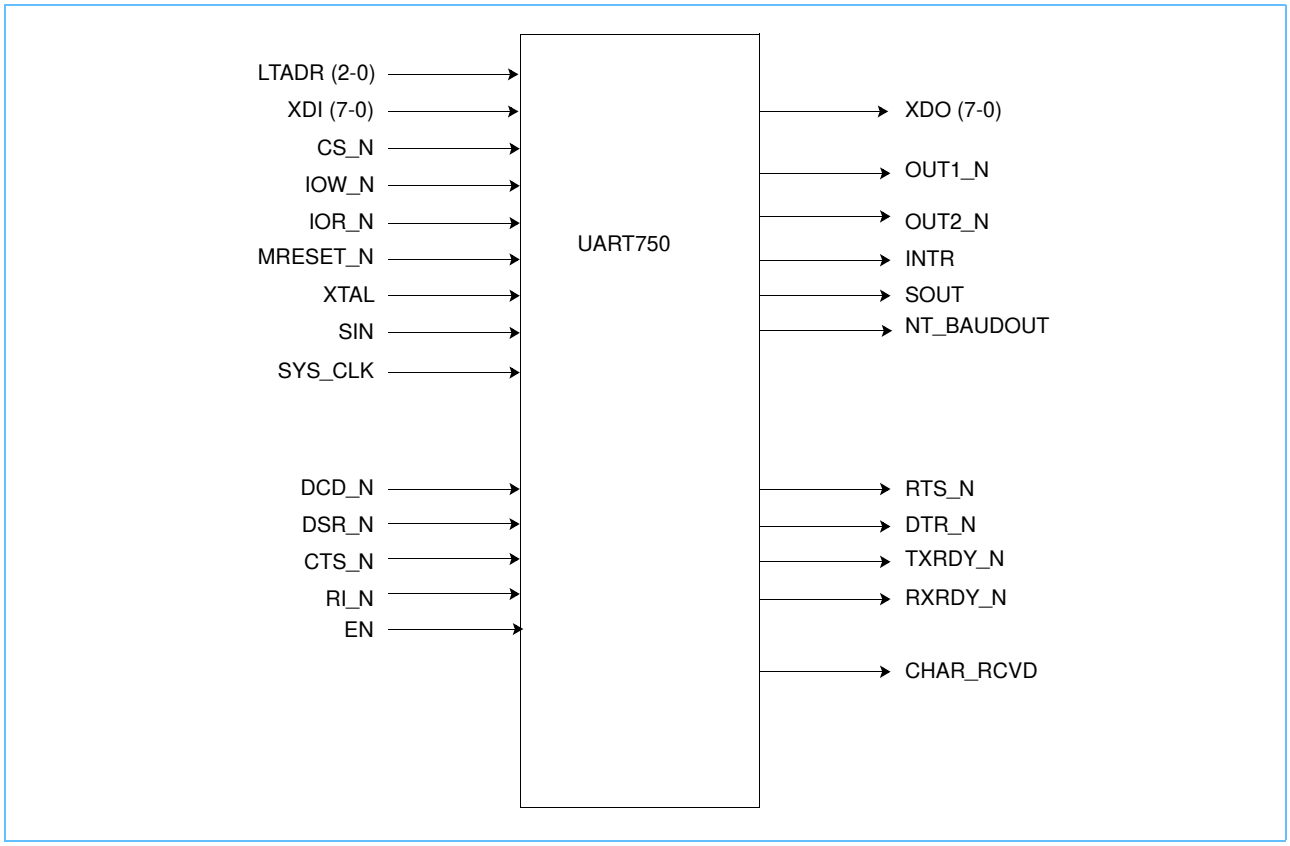
1. In this table, the most significant bit corresponds to XDI(7)/XDO(7) or OPB_DBUS_IN(0). See *Figure 4* on page 19. The data bit numbers of the OPB bus are reversed (see *Table 1* on page 17).
2. Some elements are mutually exclusive, that is, for LCR least significant hex digit, one of four character lengths must be chosen.
3. Elements not mutually exclusive can be added together in each digit, that is, if both DCD and RI indicators in MSR register are set, first hex digit will be C.
4. If not otherwise indicated, the absence of particular setting corresponds to 0.

4. Hardware Interface

4.1 UART750 Signal Diagram

The UART750 signal diagram is illustrated in *Figure 5*.

Figure 5. UART750 Signal Diagram



4.2 Signal List and Descriptions

Signals that constitute the UART750 interface are listed in the following sections. Separate coverage is provided for interface signals used for the OPB and DMA options.

4.3 UART750 Input/Output Description

Table 11. UART750 I/O Signals (Page 1 of 3)

Signal Name	I/O	Active	Description
LTADR[2:0]	I	0	Address [2:0]. The processor uses this address along with CS_N to select the UART750 internal registers. It must be valid to perform an I/O read or I/O white Bit 2 is the MSB. The register addresses are provided in *Section 3.1.1 Address Map* on page 17.
CS_N	I	0	Chip select. The active-low chip select signal. It must be active to perform I/O read or I/O write.
XDI[7:0]	I	N/A	Data input [7:0]. The processor data to be written to the UART750 internal registers. Bit 7 is the MSB.
IOW_N	I	0	I/O write. Active IOW_N writes data to internal registers.
IOR_N	I	0	I/O read. Active IOR_N reads data from internal registers.
MRESET_N	I	0	Master reset. When this input is low, it clears the control logic of the UART750 and all registers (except the receiver buffer, transmitter holding and divisor latches).
XTAL	I	N/A	XTAL in. XTAL is the basic clock for baud rate generation. The frequency is calculated as follows: XTAL frequency = Baud frequency x 16 x value in divisor latch SYS_CLK frequency > 2 x XTAL frequency The XTAL clock input is treated as asynchronous data and sampled by the rising edge of SYS_CLK, therefore, the XTAL clock must have a frequency and duty cycle that guarantees SYS_CLK rises at least once each time the XTAL is low and once each time XTAL is high. This calculation must include duty cycle and phase jitter on both of these clock inputs. Setting XTAL frequency = 1/2 x SYS_CLK frequency is *not* sufficient to meet this requirement.
SIN	I	N/A	Serial input. Serial data input from the communications link (peripheral device, modem, or data set).
DCD_N	I	0	Data carrier detect. When low, indicates that the data carrier was detected by the modem or data set. The DCD_N signal is a status input from the modem whose condition can be tested by the processor reading bit 7 (DCD) of the modem status register. Bit 7 is the complement of the DCD_N signal. Bit 3 (DDCD) of the modem status register indicates whether the DCD_N input changed state since the previous reading of the modem status register. DCD_N has no effect on the receiver.
DSR_N	I	0	Data set ready. When low, indicates that the modem or data set is ready to establish a communications link with the UART750. The DSR_N signal is a status input from the modem whose condition can be tested by the processor reading bit 5 (DSR) of the modem status register. Bit 5 is the complement of the DSR_N signal. Bit 1 (DDSR) of the modem status register indicates whether the DSR_N input has changed state since the previous reading of the modem status register. Tie high, if not in use.
CTS_N	I	0	Clear to send. When low, indicates that the modem or data set is ready to exchange data. The CTS_N signal is a status input from the modem whose condition can be tested by the processor reading bit 4 (CTS) of the modem status register. Bit 4 is the complement of the CTS_N signal. Bit 0 (DCTS) of the modem status register indicates whether the CTS_N input has changed state since the previous reading of the modem status register. CTS_N has no effect on the transmitter when auto flow control is not enabled. Tie high, if not in use. When auto flow control is enabled and FIFOs are enabled, CTS_N is also used to control the start and stop of the transmitter.

Table 11. UART750 I/O Signals (Page 2 of 3)

Signal Name	I/O	Active	Description
RI_N	I	0	Ring indicator. When low, indicates that a telephone ringing signal has been received by the modem or data set. The RI_N signal is a status input from the modem whose condition can be tested by the processor reading bit 6 (RI) of the modem status register. Bit 6 is the complement of the RI_N signal. Bit 2 (TERI) of the modem status register indicates whether the RI_N input signal has changed from a low to a high state since the previous reading of the modem status register. Whenever the RI bit of the modem status register changes from a high to a low state, an interrupt is generated if the modem status interrupt is enabled. Tie high, if not in use.
XDO[7:0]	O	N/A	Data Output [7:0]. Data output to the processor.
OUT1_N	O	0	Output 1. This is a user-designated output that can be set to an active low by programming bit 2 (OUT1) of the modem control register to a high level. A master reset operation sets this signal to its inactive (high) state. Loop mode operation holds this signal in its inactive state.
OUT2_N	O	0	Output 2. This is a user-designated output that can be set to an active low by programming bit 3 (OUT2) of the modem control register to a high level. A Master Reset operation sets this signal to its inactive (high) state. Loop mode operation holds this signal in its inactive state.
INTR	O	N/A	Interrupt. This pin goes high whenever any one of the following interrupt types have an active high condition and is enabled via the IER (receiver error flag), received data available (timeout, in FIFO mode only), transmitter holding register empty, and modem status. The INTR signal is reset low upon the appropriate interrupt service or a master reset operation.
NT_BAUDOUT	O	N/A	Divided XTAL clock. This is the 16x baud clock generated by dividing the XTAL frequency by the decimal divisor. This pin shows the synchronization between the baud clock and the system clock and shows the baud sampling timing of SIN.
SOUT	O	N/A	Serial output. Composite serial data output to the communications link (peripheral, modem, or data set). The SOUT signal is set to the marking (logic 1) state upon a master reset operation.
DTR_N	O	0	Data terminal ready. When low, this informs the modem or data set that the UART750 is ready to establish a communications link. The DTR_N output signal can be set to an active low by programming bit 0 (DTR) of the modem control register to a high level. A Master Reset operation sets this signal to its inactive (high) state. Loop mode operation holds this signal in its inactive state.
RTS_N	O	0	Request to send. When low, this informs the modem or data set that the UART750 is ready to exchange data. The RTS_N output signal can be set to an active low by programming bit 1 (RTS) of the modem control register. A Master Reset operation sets this signal to its inactive (high) state. Loop mode operation holds this signal in its inactive state. When auto flow_control is enabled, RTS_N is activated and deactivated by the receiver threshold logic.

Table 11. UART750 I/O Signals (Page 3 of 3)

Signal Name	I/O	Active	Description
TXRDY_N, RXRDY_N	O	0	Transmitter/receiver ready. Transmitter and receiver DMA signalling is available through two pins. When FIFOs are enabled, DMA signalling can be selected via FCR3. When FIFOs are disabled, only DMA mode 0 is allowed. Mode 0 supports single transfer DMA where a transfer is made between processor bus cycles. Mode 1 supports multiple transfer DMA where multiple transfers are made continuously until the RCVR FIFO has been emptied or the XMIT FIFO has been filled. DMA mode 0: FCR3 = 0 RXRDY mode 0 When the FIFOs are disabled (16450 mode) or the FIFO are enabled and there is at least one character in the RCVR FIFO or RCVR buffer register, the RXRDY_N pin will be low active. Once activated, the RXRDY_N pin goes inactive when there are no more characters in the FIFO or buffer register. TXRDY mode 0 When the FIFOs are disabled or the FIFOs are enabled and there are no characters in the XMIT FIFO or XMIT holding register, the TXRDY_N pin will be low active. Once activated, the TXRDY_N pin goes inactive after the first character is loaded into the XMIT FIFO or transmit holding register. DMA mode 1: FCR3 = 1 RXRDY mode 1 When the FIFOs are enabled and the trigger level or the timeout has been reached, the RXRDY_N pin goes active low. Once activated, it will go inactive high when there are no more characters in the FIFO or holding register. TXRDY mode 1 When FIFOs are enabled TXRDY_N becomes inactive when the TXFIFO is completely full, and becomes active when the TXFIFO is completely empty.
CHAR_RCVD	O	1	Character received. Active high for 1 system clock upon detection of the stop bit of the current received character. A diagnostic pin that can be used by external logic to count time between received characters.
SYS_CLK	I	N/A	System clock, drives clock splitters to generate B and C clocks for IBM Full-Scan DFT.
EN	I	1	EN is an enable signal to the latches. When EN is held to ‘0’ the latches will not accept new data. Optionally, EN can be used to generate a clock gating structure during synthesis. If this option is not chosen, EN will most likely become a data gate into the register. This depends on the synthesis tool.

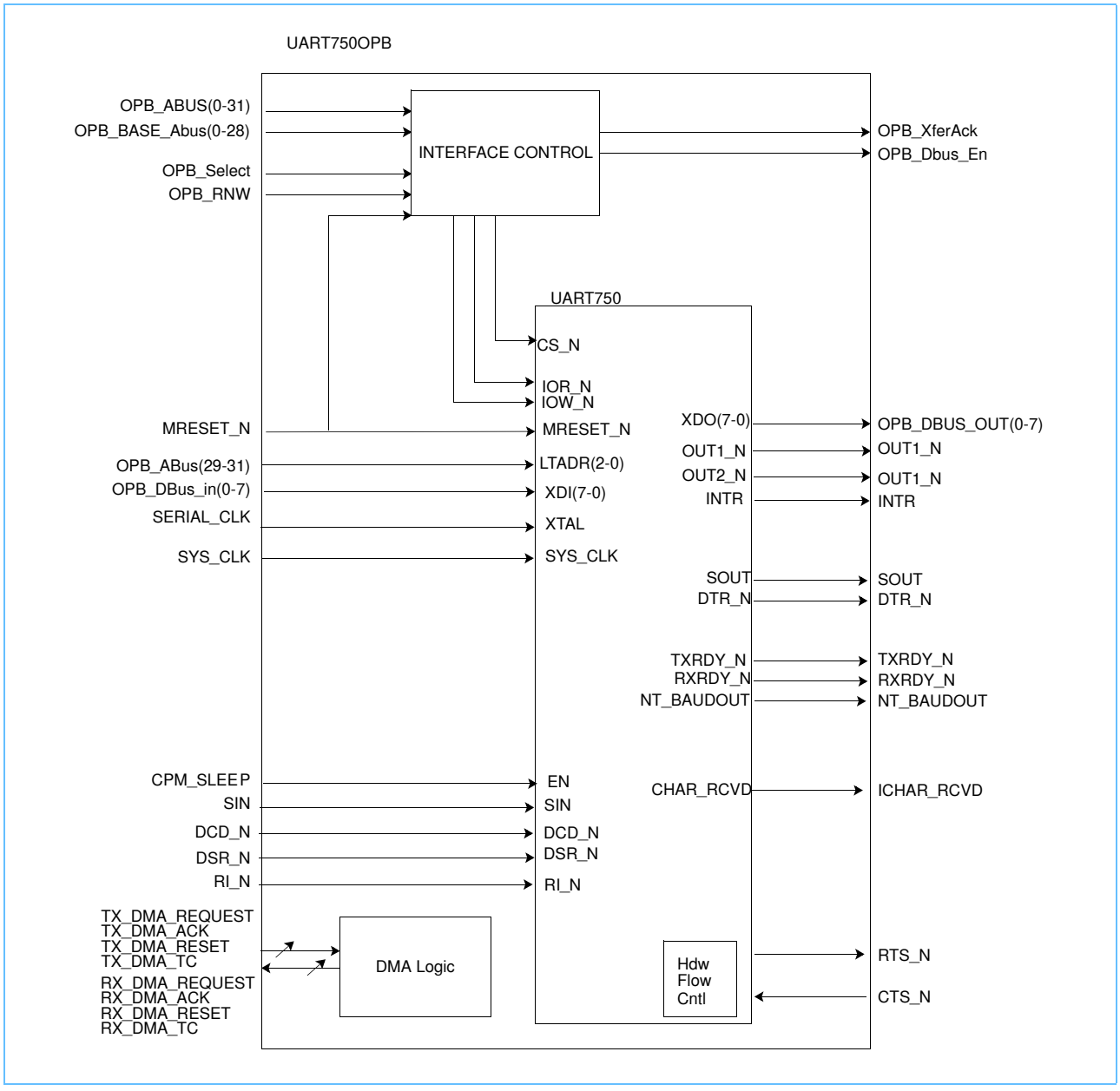
4.4 UART750 Interface Timing Specifications

For interface timing information, see *Section 5.4 Signal Timing Specifications* on page 42.

4.5 UART750OPB Signal Diagram

The UART750 signal diagram is illustrated in *Figure 6*.

Figure 6. UART750OPB Signal Diagram



4.5.1 OPB Interface I/O Signals

When the UART750 core is connected to the OPB, the OPB_DBUS_in(0) corresponds to XDI(7) of the UART750 input bus. Therefore, if the line control register is to be programmed to a word length of seven bits and no parity, then OPB_DBUS_in(0-7) = '00000010'.

The UART750 OPB interface I/O signals are shown in *Table 12*.

Table 12. UART750 OPB Interface I/O Signals (Page 1 of 2)

Signal Name	I/O	Description
OPB_ABUS[0:31]	I	OPB_ABus[0:28] corresponds to the OPB_BASE_ABUS[0:28]. OPB_ABus[29:31] corresponds to UART inputs LTADR[2:0]. Note: OPB_ABus[29] is the most significant UART address bit, LTADR[2].
OPB_BASE_ABUS[0: 28]	I	OPB base address bits, used to decode OPB_ABUS[0:28].
OPB_DBus_in[0:7]	I	OPB data input [0:7], see OPB specification v1.2. Bit 0 is the MSB, corresponding to XDI[7] of UART750.
OPB_DBus_out[0:7]	O	OPB data output [0:7], see OPB specification v1.2. Bit 0 is the MSB, corresponding to XDO[7] of UART750.
OPB_Select	I	OPB data transfer control signal see OPB specification v1.2.
OPB_RNW	I	OPB read not write, see OPB specification v1.2.
OPB_DBus_en	O	OPB data bus enable, see OPB specification v1.2.
OPB_XferAck	O	OPB transfer acknowledge, see OPB specification v1.2.
SYS_CLK	I	System clock, corresponds to UART750 SYS_CLK.
SERIAL_CLK	I	Xtal in, corresponds to UART750 XTAL.
NT_BAUDOUT	O	Divided XTAL clock, corresponds to UART750 NT_BAUDOUT.
MRESET_N	I	Master reset, corresponds to UART750 MRESET_N.
SIN	I	Serial input, corresponds to UART750 SIN.
DCD_N	I	Data carrier detect, corresponds to UART750 DCD_N.
DSR_N	I	Data set ready, corresponds to UART750 DSR_N.
CTS_N	I	Clear to send, corresponds to UART750 CTS_N.
RI_N	I	Ring indicator, corresponds to UART750 RI_N.
DTR_N	O	Data terminal ready, corresponds to UART750 DTR_N.
RTS_N	O	Request to send, corresponds to UART750 RTS_N.
OUT1_N	O	Output 1, corresponds to UART750 OUT1_N.
OUT2_N	O	Output 2, corresponds to UART750 OUT2_N.
INTR	O	Interrupt, corresponds to UART750 INTR.
RXRDY_N	O	Receiver ready, corresponds to UART750 RXRDY_N.
TXRDY_N	O	Transmitter ready, corresponds to UART750 RXRDY_N.
CPM_SLEEP_N	I	CPM_SLEEP_N is an enable signal to the latches. When CPM_SLEEP_N is held to ‘0’ the latches will not accept new data. Optionally, CPM_SLEEP_N can be used to generate a clock gating structure during synthesis. If this option is not chosen, CPM_SLEEP_N will most likely become a data gate into the register. This depends on the synthesis tool.

Table 12. UART750 OPB Interface I/O Signals (Page 2 of 2)

Signal Name	I/O	Description
CHAR_RCVD	O	Goes active high for 1 system clock upon detection of the stop bit of the current received character, can be used by external logic to count time between received characters.
SOUT	O	Serial Output -corresponds to UART750 SOUT
TX_DMA_ENABLE	I	Transmitter DMA enable.
TX_DMA_REQUEST	O	Transmitter DMA request.
TX_DMA_ACK	I	Transmitter DMA acknowledge.
TX_DMA_TC	I	TX terminal count used to generate TX_DMA_RESET.
TX_DMA_RESET	O	DMA RESET signal, can be used to reset external DMA control latches.
RX_DMA_ENABLE	I	Receiver DMA enable.
RX_DMA_REQUEST	O	Receiver DMA request.
RX_DMA_ACK	I	Receiver DMA acknowledge.
RX_DMA_TC	I	RX terminal count used to generate RX_DMA_RESET.
RX_DMA_RESET	O	DMA RESET signal, used to reset external DMA control latches.

4.6 Enhanced DMA Support Interface Signals

The enhanced DMA signals are described in *Table 13*.

Table 13. Enhanced DMA Signals

Signal	In/Out	Signal Description
TX_DMA_ENABLE	In	This input enables the TX DMA channel. The latch for TX_DMA_ENABLE is external to this core.
TX_DMA_REQUEST	O	The UART750 asserts TX_DMA_REQUEST to the DMA controller to indicate its readiness to be given transmitter data. Before enabling the transmitter DMA channel, the channel should be programmed to indicate the UART's data width, that it is an internal device, the transfer direction, the target address, and the timing parameters associated with the UART. TX_DMA_REQUEST is gated inactive as soon as TX_DMA_ACK is received. Logically this is: TX_DMA_REQUEST = not TXRDY_N & TX_DMA_ENABLE & not TX_DMA_ACK & not TX_DMA_ACK_delayed;
TX_DMA_ACK	I	After receiving a DMA request from the UART, the DMA channel arbitrates for control of the PLB. After it gets the PLB, the DMA controller places the memory address on the PLB and asserts TX_DMA_ACK acknowledge to the UART. The UART will receive data from the OPB, and the data should be valid on the last two cycles of TX_DMA_ACK. The DMA channel should be should be programmed to have TX_DMA_ACK active for 3 cycles. This can be controlled by setting the Peripheral Wait Time field to 000010 in the DMA channel control register.
RX_DMA_ENABLE	I	This input enables the RX DMA channel. The latch for RX_DMA_ENABLE is external to this core.
RX_DMA_REQUEST	O	The UART asserts RX_DMA_REQUEST to the DMA controller to indicate its readiness to be present receiver data. Before enabling the receiver DMA channel, the channel should be programmed to indicate the UART's width, that it is an internal device, the transfer direction, the target address, and the timing parameters associated with the UART. RX_DMA_REQUEST is gated inactive as soon as RX_DMA_ACK is received. Logically this is: RX_DMA_REQUEST = not RXRDY_N & RX_DMA_ENABLE & not RX_DMA_ACK;
RX_DMA_ACK	I	After receiving a DMA request from the UART, the DMA channel arbitrates for control of the PLB. After it gets the PLB, the DMA controller places the memory address on the PLB and asserts RX_DMA_ACK to the UART. The UART will send data to the OPB, and the data should be valid on the last cycle of RX_DMA_ACK. The DMA channel should be should be programmed to have RX_DMA_ACK active for 3 cycles. This can be controlled by setting the Peripheral Wait Time field to 000010 in the DMA channel control register.
TX_DMA_TC	I	See Tx_DMA_RESET.
RX_DMA_TC	I	See Rx_DMA_RESET.
TX_DMA_RESET	O	If TX DMA channel is enabled by TX_DMA_ENABLE, the TX_DMA_TC terminal count input will generate a TX_DMA_RESET output signal that can be used to reset the external TX DMA enable latch (external to core). The TX_DMA_RESET output will remain active as long as TX_DMA_TC input is active and TX_DMA_ENABLE is active. Logically this is: TX_DMA_RESET <= TX_DMA_TC and TX_DMA_ENABLE;
RX_DMA_RESET	O	If the RX DMA channel is enabled by RX_DMA_ENABLE, the RX_DMA_TC terminal count input will generate an RX_DMA_RESET output signal that can be used to reset the external RX DMA enable latch (external to core). The RX_DMA_RESET output will remain active as long as RX_DMA_TC input is active and RX_DMA_ENABLE is active. Logically this is: RX_DMA_RESET <= RX_DMA_TC and RX_DMA_ENABLE;

5. Core Integration

5.1 Configurability

5.1.1 UART750

The UART750 and UART750OPB are two cells delivered as synthesizable rtl. The UART750 is configurable to choose a 16-byte or 64_byte FIFO. This is done by defining a parameter in the UART750.defines.v and UART750OPB.defines.v files.

- ‘define FIFO16 - results in 16-byte FIFOs
- ‘define FIFO64 - results in 64-byte FIFOs

5.1.2 UART750OPB

The UART750OPB entity allows attachment of the UART750 to the IBM PowerPC OPB. The UART750OPB interface also contains optional enhanced DMA support for receiver and transmitter transactions. UART750OPB also instantiates the UART750.

5.2 Initialization Sequence

The UART750 requires a reset for at least 10 clock cycles before any registers are written to. A typical initialization sequence follows:

Table 14. Initialization Sequence

Command	Function
reset	At least 10 system clocks
write lcr 80	dlab = 1 enable write to dll
write dll 01	Set baud = xtal / (16 x XX01)
write dlm 00	Set baud = xtal/(16 x 0001)
write lcr 0C	Configure 5 bit odd parity, 2 stop bits
write fcr C1	Enable both FIFOs, 14 byte rx FIFO trigger level
write ier 0F	Enable all interrupts
read lsr	Read The lsr to clear any false status interrupts
read msr	Read the msr to clear any false modem interrupts

At this point, the UART750 is ready for both sending and receiving data.

5.3 Clocking Guidelines

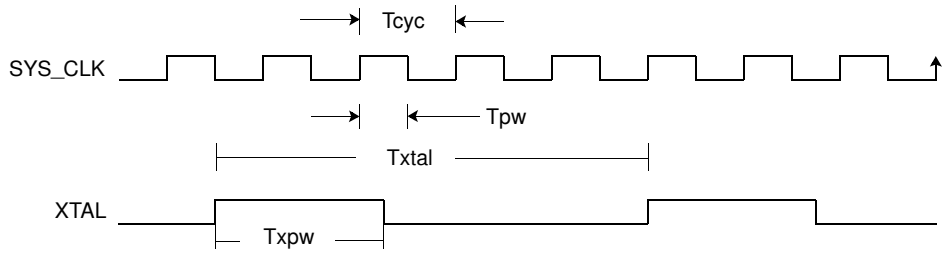
For input XTAL requirements, see *Section 5.4.1 XTAL Clock Input versus SYS_CLK* on page 42.

5.4 Signal Timing Specifications

5.4.1 XTAL Clock Input versus SYS_CLK

The XTAL clock timing is illustrated in *Figure 7*.

Figure 7. XTAL Clock



XTAL is the basic clock for baud rate generation. The frequency is calculated as follows:

XTAL frequency = Baud frequency x 16 x value in divisor latch
SYS_CLK frequency > 2 x XTAL frequency

The XTAL clock input is treated as asynchronous data and sampled by the rising edge of SYS_CLK. Therefore the XTAL clock must have a frequency and duty cycle that ensures SYS_CLK rises at least once each time the XTAL is low and once each time XTAL is high. This calculation must include duty cycle and phase jitter on both of these clock inputs. Setting Txtal = 2 * Tcyc is NOT sufficient to meet this requirement. A better choice is setting Txtal >= 3 * Tcyc.

5.4.2 UART750 Write Operation

1. Inputs are synchronous to the rising edge of SYS_CLK, and therefore have setup and hold times to SYS_CLK.
2. IOW_N is a data enable signal for latching of the address and input data.
3. The target register is updated in the cycle following the rise of IOW_N as shown.
4. The last value of address and data latched during IOW_N is used to update the target register.

Figure 8. UART750 Write Operation

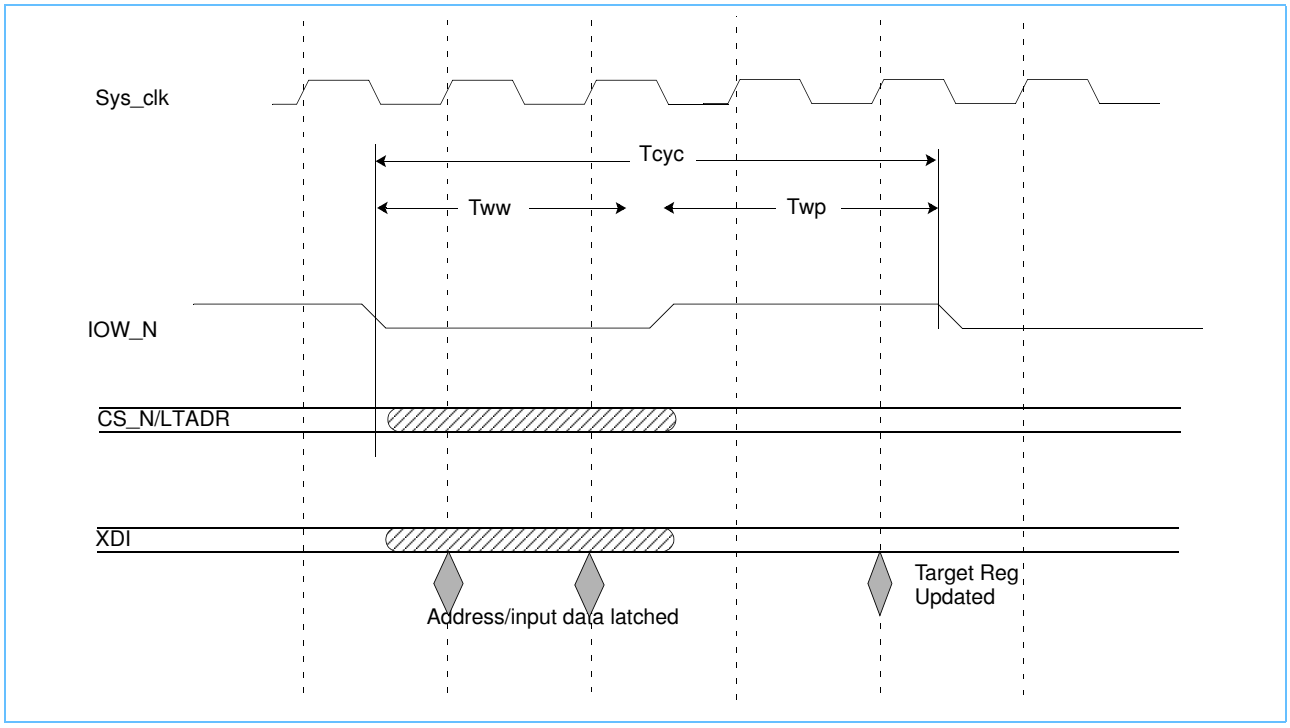


Table 15. Write Timing Parameters

Parameter	Description	Min (SYS_CLK cycles)	Max (SYS_CLK cycles
Tcyc	Read cycle time	3 cycle	–
Tww	Write width	1 cycle	–
Twp	Write pulse	2 cycle	–

Note: Address and data must be valid for the last cycle the IOW_N is active.

5.4.3 UART750 Read Operation

1. Inputs are synchronous to the rising edge of SYS_CLK, and therefore have setup and hold times to SYS_CLK.
2. IOR_N is a data enable signal for latching of the address and output data.
3. The address is decoded and the output data is latched on the first cycle that IOR_N is active.
4. Some registers are updated after a read. Therefore the address is latched while IOR_N is active and used to do the update.
5. The read address must be valid for each cycle that IOR_N is active.

Figure 9. UART750 Read Operation

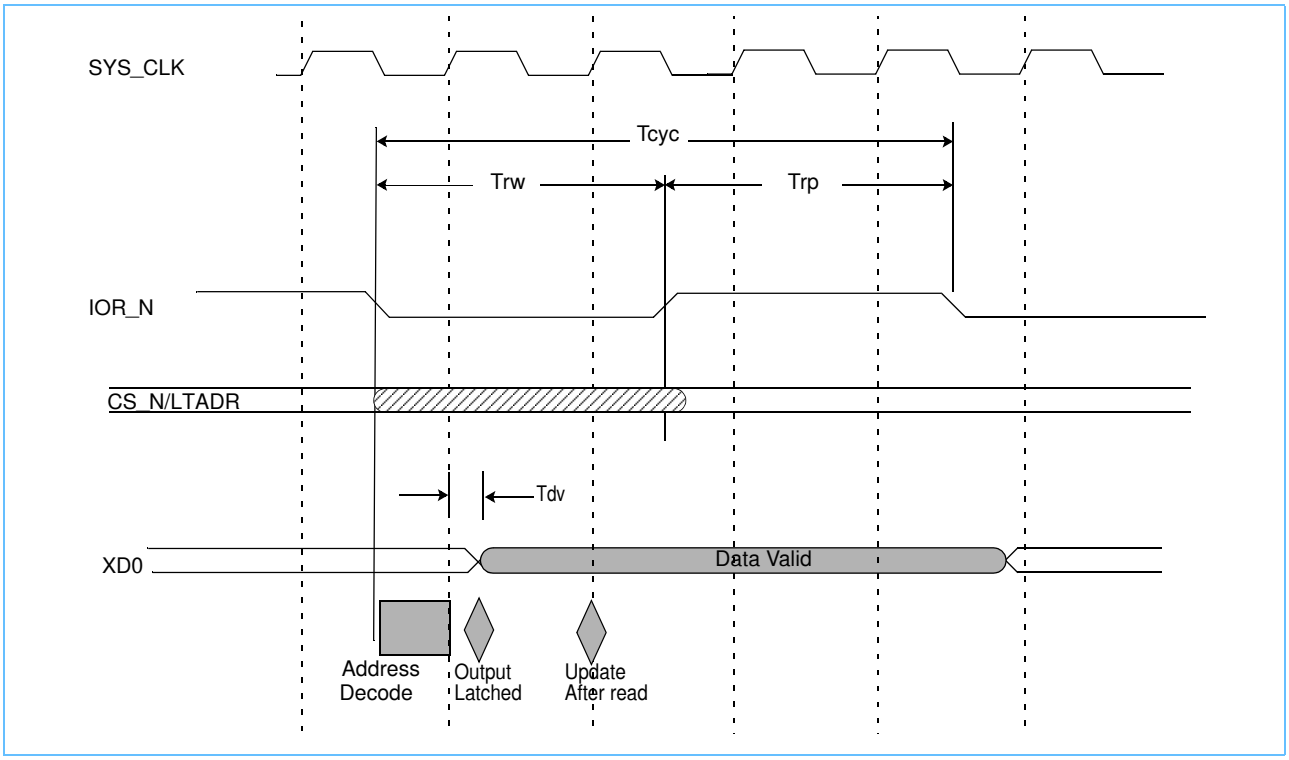


Table 16. Read Timing Parameters

Parameter	Description	Min (SYS_CLK cycles)	Max
Tcyc	Read cycle time	3 cycles	–
Trw	Read width	2 cycle	–
Trp	Read pulse	1 cycle	–

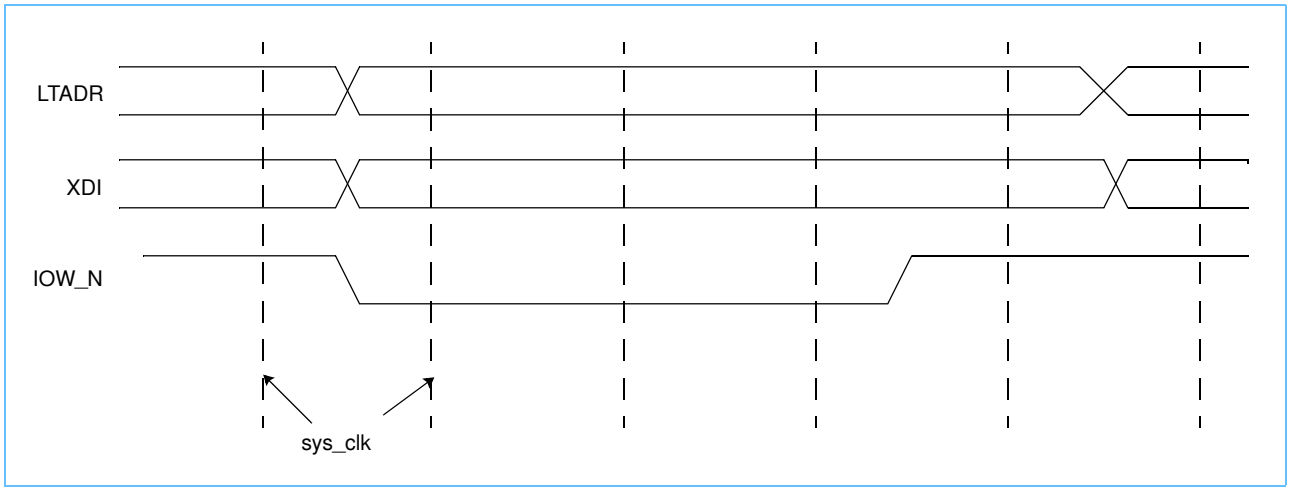
Note: The output remains valid until the next read.

5.4.3.1 Recommended UART750 Read/Write Protocol

Timing assertions alone cannot guarantee correct read and write operation to the UART750. The diagrams above show the functional timing requirements, but the safest way to meet these requirements is for address and data to be valid for each SYS_CLK cycle that IOW_N or IOR_N is active.

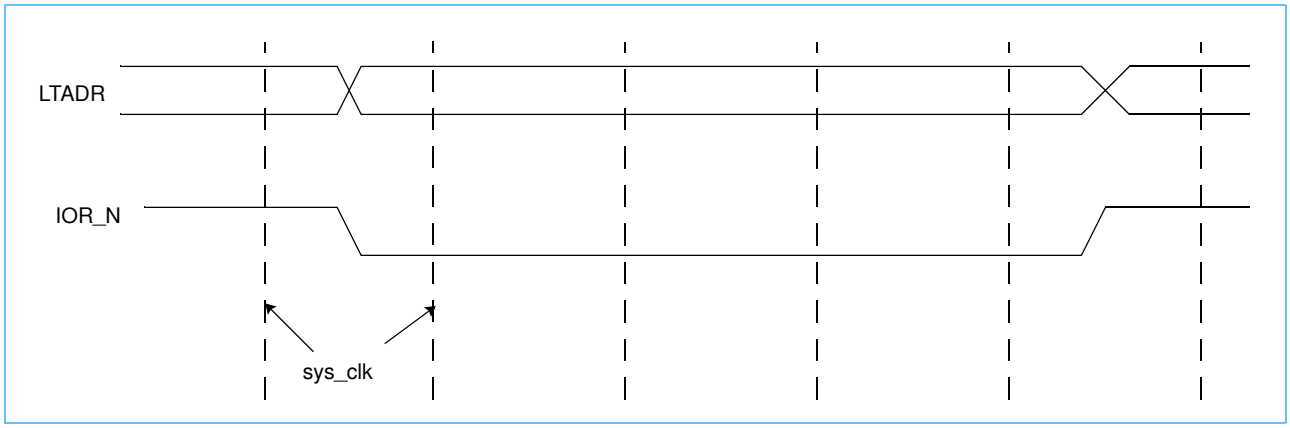
The OPB shell that is available with this core is designed to operate exactly this way. This is illustrated in *Figure 10* and *Figure 11*.

Figure 10. Recommended Write Protocol



5.4.3.2 Recommended Write Protocol

Figure 11. Recommended Read Protocol



5.4.3.3 OPB Bus Timing Interface Diagrams

Figure 12 and *Figure 13* show the expected OPB read and write timing.

Figure 12. Write Cycle Timing Diagram on OPB Bus

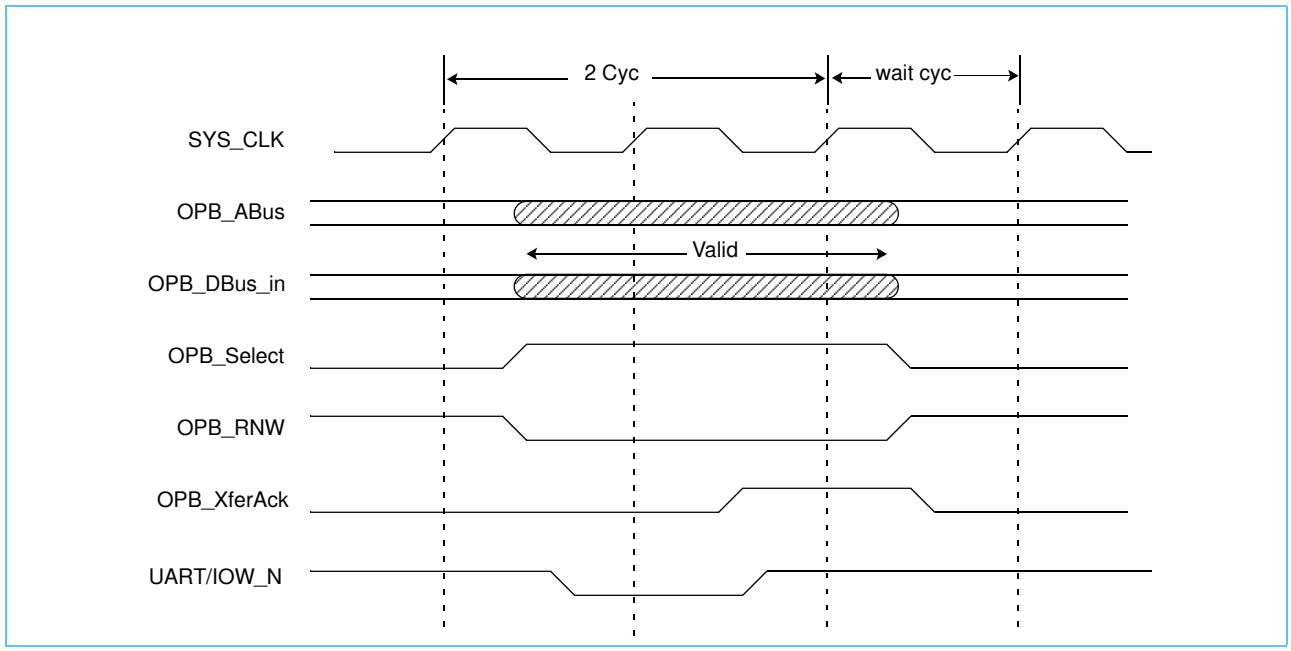
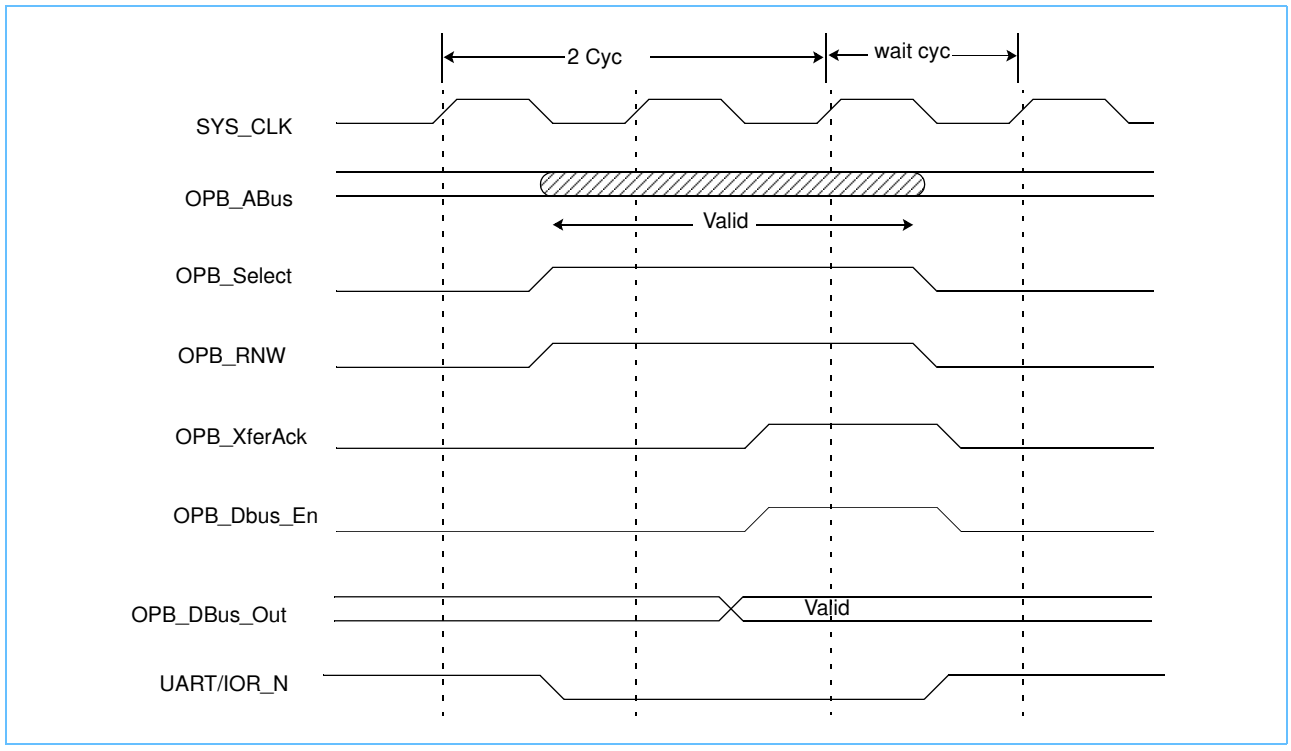


Figure 13. Read Cycle Timing Diagram on OPB Bus



5.4.4 Optional DMA Function Timing Diagrams

Figure 14 through *Figure 17* on page 48 present the timing diagrams for the optional DMA function in the UART750OPB shell along with transaction examples.

Figure 14. DMA Write Cycle timing

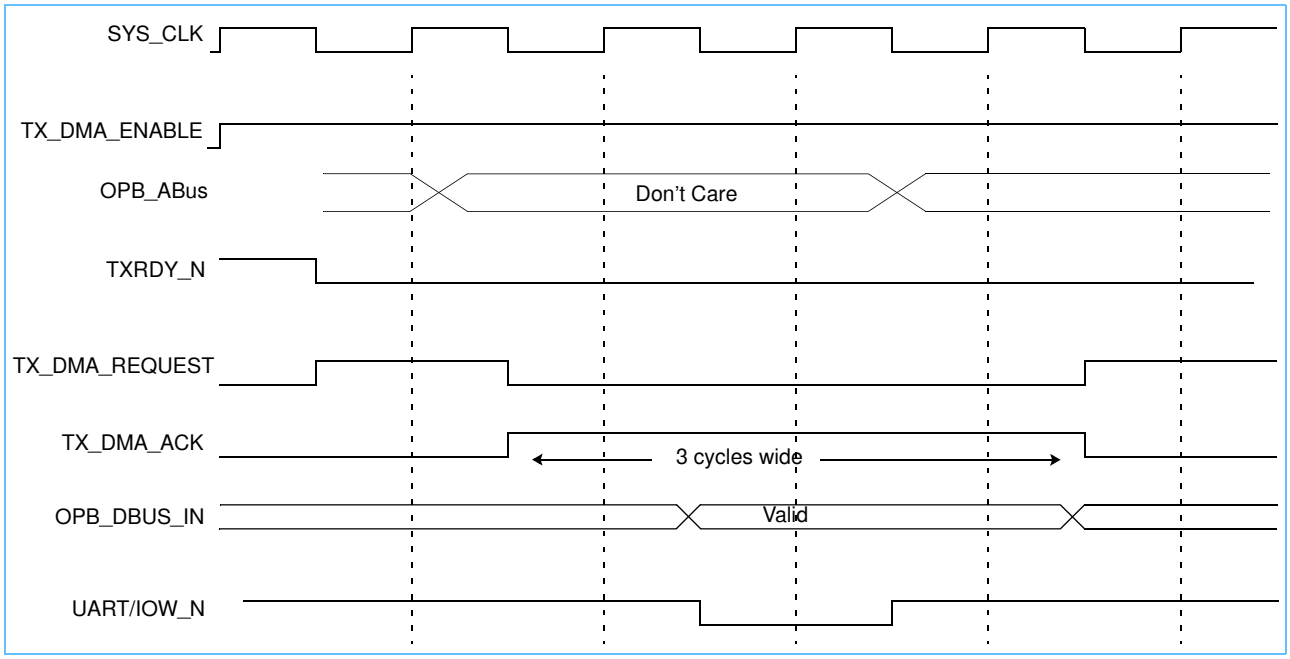


Figure 15. DMA Read Cycle timing

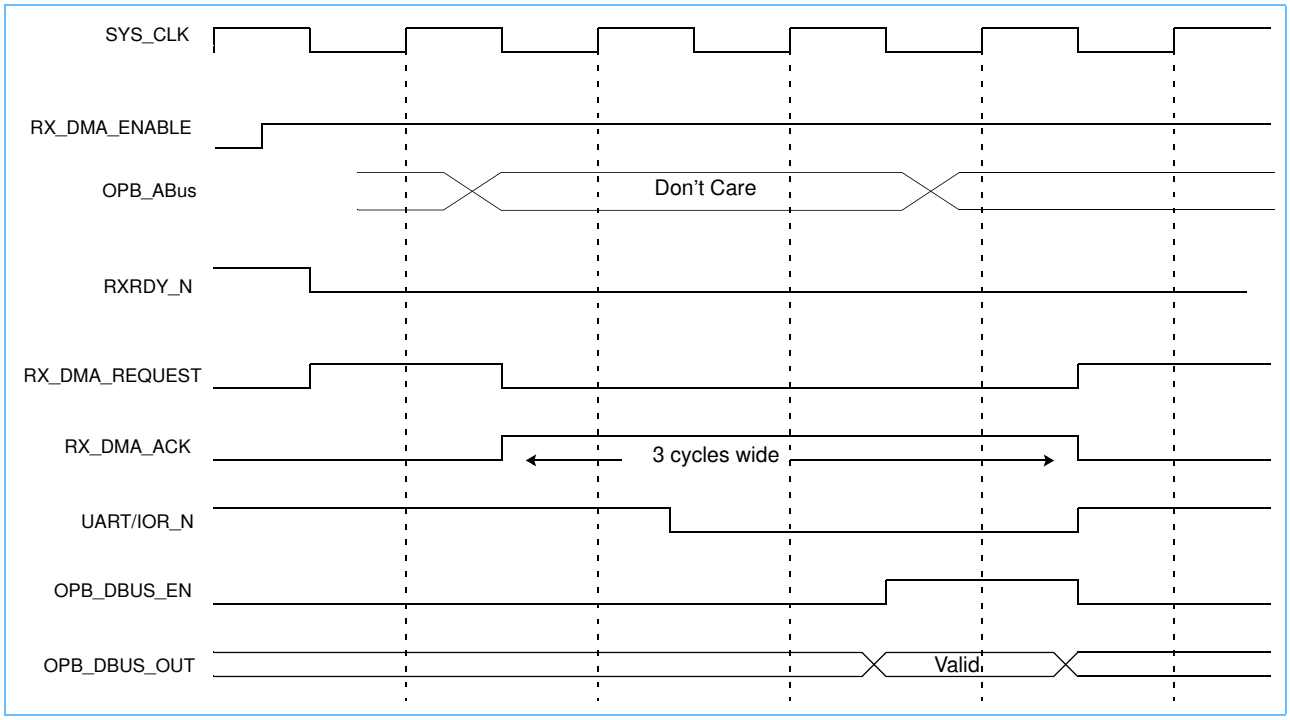


Figure 16. DMA Write Transaction Example

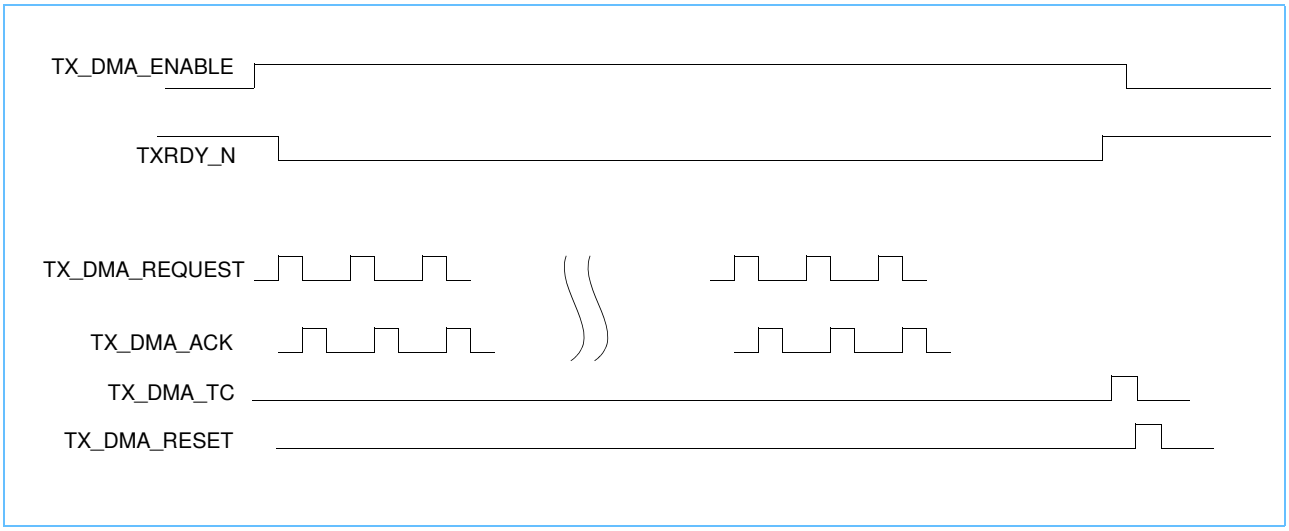
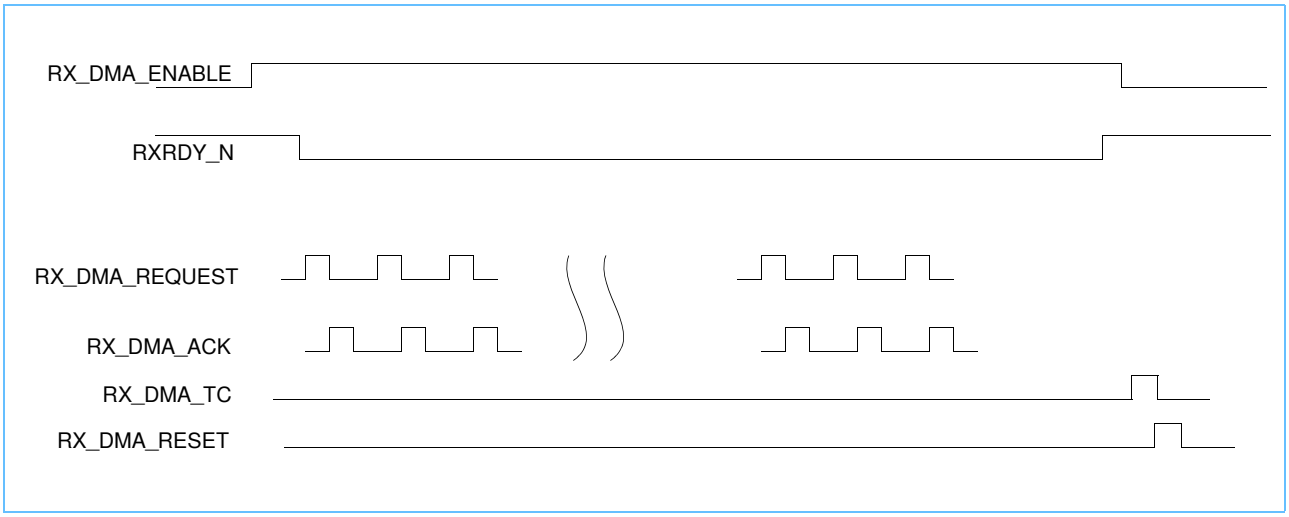


Figure 17. DMA Read Transaction Example



5.4.5 Fifo Latency

The UART750 FIFOs are implemented with shift registers. When data is written to the TX_FIFO, or serial data is received and loaded into the RX_FIFO, the data must first shift to the top of the FIFOs before it can be transmitted or received. Therefore, there is a latency of approximately 16 cycles (64 cycles for 64-byte FIFOs) that must be absorbed before the very first character loaded into each FIFO can be processed. If the FIFOs are loaded with more than one character, subsequent characters are processed with no delay.

5.5 Synthesis

Synthesis scripts and timing assertions are supplied with this core for both Synopsys and IBM BooleDozer™ to convert the Verilog source rtl to gate level net lists in a target technology. The BooleDozer script assumes that the Verilog rtl source has already been run through HDL synthesis in order to create HDLvim.

Both synthesis scripts are technology independent. It is assumed that the customer will provide the technology-dependent instructions and timing requirements that identify the target library and any other requirements specific to the target technology.

5.6 Boolean Equivalency Checking

After the gate level netlist is generated in the target technology, it is recommended that the gate level netlist is compared to the Verilog rtl source code with formal verification. This will insure that the logic function has not been changed during synthesis and mapping.

Appendix A. IBM ASIC-Specific Information

OPB device compliance is shown in *Table 17*.

Table 17. OPB Device Compliance List

Item	Protocol	Description	Supported?
OPB Slave Compliance			
S1	Addressing	Can coexist with other OPB slaves	Y
		OPB address decode size	32
		Programmable address location	29-31
		Address decoding	0-31
		Supports 32-bit addressing	Y
		Supports 64-bit addressing	N
S2	Read operations	Single byte	Y
		Halfword transfers	N
		Fullword transfers	N
		Doubleword transfers	N
S3	Write operations	Single byte	Y
		Halfword transfers	N
		Fullword transfers	N
		Doubleword transfers	N
S4	Aborts	Support for master transaction aborts	N
S5	Bus locking	Support for locked_bus transfers	N
S6	Sequential addressing	Support for sequential addressing (OPB_seqAddr signal)	N
S7	Data steering	Support for writes from 8-bit masters	N
		Support for writes from 16-bit masters	N
		Support for writes from 32-bit masters	N
		Support for writes from 64-bit masters	N
		Support for reads from 8-bit masters	N
		Support for reads from 16-bit masters	N
		Support for reads from 32-bit masters	N
		Support for reads from 64-bit masters	N
S8	Retry	Without bus locked	N
		With bus locked	N
S9	Wait/Timeout	Slave provides for “timeout suppression (Sln_toutSup signal)	N
S10	Transfer Errors	Slave provides error acknowledgment when transfer errors occur (Sln_errAck signal)	N
S11	Byte Enable	Support for transactions with byte enables	N
		Provide acknowledgements for byte-enabled transfers (use of the Sln_beAck signal)	N