Core16550 v3.3

HandBook

Table of Contents

Introduction

Core16550 is a standard UART providing software compatibility with the popular 16550 device. It performs serial-to-parallel conversion on data originating from modems or other serial devices, and performs parallel-to-serial conversion on data from a CPU to these devices. When transmitting, the data is written in parallel into the transmit FIFO of the UART. The data is then transmitted in serial form. When receiving data, the UART transforms the serial input data into a parallel form to facilitate reading by the processor. A typical 16550 application is shown in Figure 1.

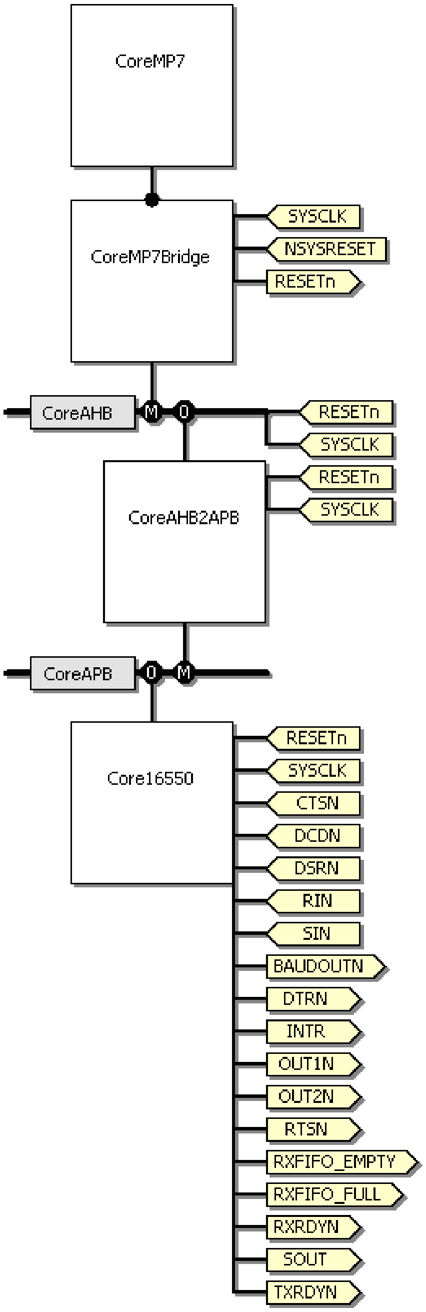


Figure 1 **• Typical 16550 Application**

Supported Device Families

Fusion
IGLOO®
IGLOOe
ProASIC®3L
ProASIC3
ProASIC3E
SmartFusion®
SmartFusion®2
IGLOO®2
RTG4™

Core Version

This handbook applies to Core16550 v3.3. The release notes provided with the core list known discrepancies between this handbook and the core release associated with the release notes.

Device Utilization and Performance

Core16550 utilization and performance data are summarized in Table 1.

***Table 1* • Core16550 Utilization and Performance**

Family	Cell or Tiles			RAM Blocks	Utilization	
	Sequential	Combinatorial	Total		Device	Total
RTG4™	349	874	1,223	2	RT4G150-1CG1657M	0.8%
IGLOO®2	271	754	1,021	2	M2GL050TFBGA896STD	1.82%
SmartFusion®2	271	754	1,021	2	M2S050TFBGA896STD	1.82%
SmartFusion®	243	1,163	1,406	2	A2F500M3G-STD	12%
IGLOO™/IGLOOE	237	1,010	1,247	2	AGL600-STD/AGLE600V2	9%
Fusion	237	1,010	1,247	2	AFS600-STD	9%
ProASIC®3/E	237	1,010	1,247	2	A3P600-STD	9%
ProASIC$^{\underline{PLUS}}$®	233	1,209	1,442	2	APA075-STD	47%
RTAX-S	229	608	837	2	RTAX250S-STD	20%
Axcelerator®	229	608	837	2	AX125-STD	42%

Note: The above data were obtained by typical synthesis and place-and-route flow.

1 – Functional Block Description

This section provides a short description for each element of the internal block diagram in Figure 1-1.

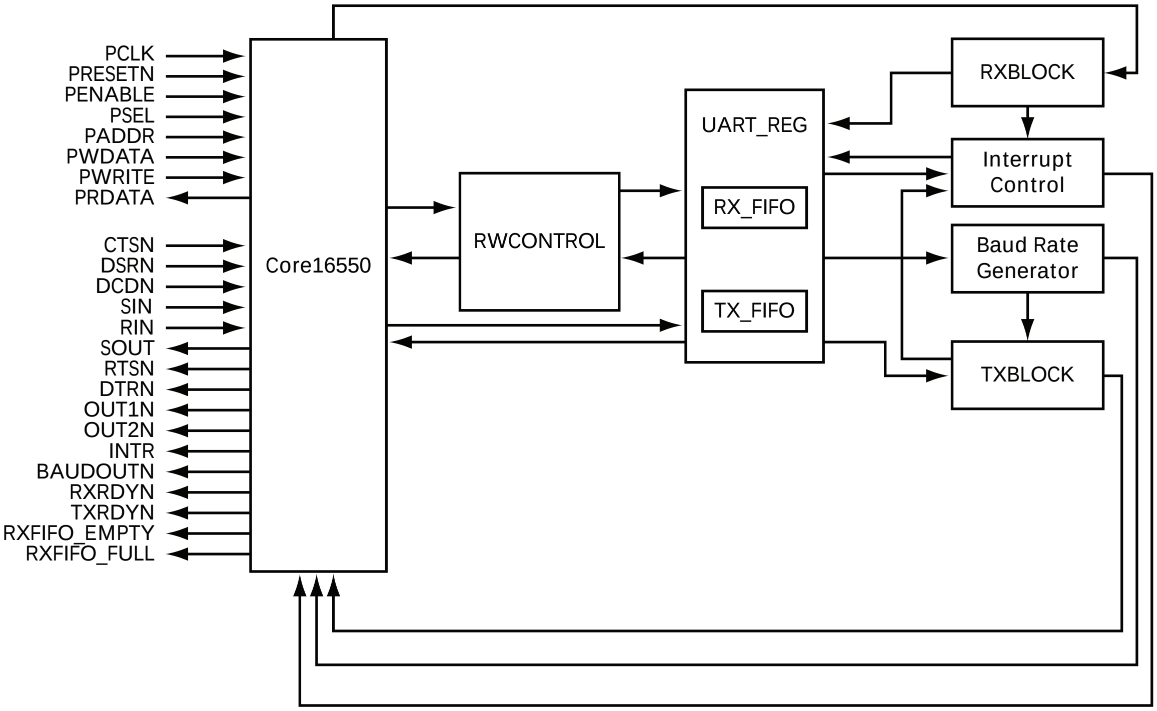


Figure 1-1 **• Core16550 Block Diagram**

Elements of the Internal Block Diagram

RWControl

The RWControl block is responsible for handling the communications with the processor (parallel) side of the system. All writing and reading of internal registers is accomplished through this block.

UART_Reg

The UART_Reg block holds all of the device internal registers.

RXBlock

This is the Receiver block. RXBlock receives the incoming serial word. It is programmable to recognize data widths, such as 5, 6, 7, or 8 bits; various parity settings, such as even, odd, or no-parity; and different stop bits, such as 1, 1½, and 2 bits. RXBlock checks for errors in the input data stream, such as overrun errors, frame errors, parity errors, and break errors. If the incoming word has no problems, it is placed in the Receiver FIFO.

Interrupt Control

The Interrupt Control block sends an interrupt signal back to the processor, depending on the state of the FIFO and its received and transmitted data. The Interrupt Identification Register provides the level of the interrupt. Interrupts are sent for empty transmission/receipt buffers (or FIFOs), an error in receiving a character, or other conditions requiring the attention of the processor.

Baud Rate Generator

This block takes the input PCLK and divides it by a programmed value (from 1 to $2^{16} – 1$). The result is divided by 16 to create the transmission clock (BAUDOUT).

TXBlock

The Transmit block handles the transmission of data written to the Transmit FIFO. It adds the required start, parity, and stop bits to the data being transmitted so the receiving device can do the proper error handling and receiving.

Software Interface

The Core16550 register definitions and address mappings are described in this section. The Core16550 register summary is shown in Table 1-1.

***Table 1-1* • Core16550 Register Summary**

PADDR[4:0] (Address)	Divisor Latch Access Bit* (DLAB)	Name	Symbol	Default (reset) Value	No. of Bits	Read/Write
00	0	Receiver Buffer Register	RBR	XX	8	R
00	0	Transmitter Holding Register	THR	XX	8	W
00	1	Divisor Latch (LSB)	DLR	01h	8	R/W
04	1	Divisor Latch (MSB)	DMR	00h	8	R/W
04	0	Interrupt Enable Register	IER	00h	8	R/W
08	X	Interrupt Identification Register	IIR	C1h	8	R
08	X	FIFO Control Register	FCR	01h	8	W
0C	X	Line Control Register	LCR	00h	8	R/W
10	X	Modem Control Register	MCR	00h	8	R/W
14	X	Line Status Register	LSR	60h	8	R
18	X	Modem Status Register	MSR	00h	8	R
1C	X	Scratch Register	SR	00h	8	R/W

*Note: *DLAB is the MSB of the Line Control Register (LCR bit 7).*

Receiver Buffer Register

The Receiver Buffer Register is defined in Table 1-2.

Table 1-2 • **Receiver Buffer Register (read only) — Address 0 DLAB 0**

Bits	Name	Default State	Valid States	Function
7..0	RBR	XX	0..FFh	Received data bits. Bit 0 is the LSB, and is the first received bit.

Transmitter Holding Register

The Transmitter Holding Register is defined in Table 1-3.

Table 1-3 • **Transmitter Holding Register (write only)**

Bits	Name	Default State	Valid States	Function
7..0	THR	XX	0..FFh	Data bits to be transmitted. Bit 0 is the LSB, and is transmitted first.

FIFO Control Register

The FIFO Control Register is defined in Table 1-4.

Table 1-4 • **FIFO Control Register (write only)**

Bits (7:0)	Default State	Valid States	Function
0	1	0, 1	Enables both the TX and RX FIFOs. This bit must be set to 1 when other FCR bits are written to or they will not be programmed. 0 – Disabled 1 – Enabled
1	0	0, 1	Clears all bytes in the RX FIFO and resets its counter logic. The shift register is not cleared. 0 – Disabled 1 – Enabled
2	0	0, 1	Clears all bytes in the TX FIFO and resets its counter logic. The shift register is not cleared. 0 – Disabled 1 – Enabled
3	0	0, 1	0 – Single transfer DMA:Transfer made between CPU bus cycles 1 – Multi-transfer DMA:Transfers made until RCVR FIFO empty or XMIT FIFO filled. FCR[0] must be set to 1 to set FCR[3] to 1.
4, 5	0	0, 1	Reserved for future use.
6, 7	0	0, 1	These bits are used to set the trigger level for the RX FIFO interrupt. 7 6 RX FIFO Trigger Level (bytes) 0 0 01 0 1 04 1 0 08 1 1 14

The Divisor Control Registers

The baud rate (BR) clock is generated by dividing the input reference clock (PCLK) by 16 and the Divisor value. The exact formula is shown in EQ 1-1:

$$BR = \frac{PCLK}{16 \cdot DivisorValue}$$

EQ 1-1

Table 1-5 gives an example of divisor values for desired baud rates when using an 18.432 MHz reference clock.

***Table 1-5* • Divisor Latch (LS), (MS)**

Bits	Name	Default State	Valid States	Function
7..0	DLR	01h	01..FFh	The LSB of Divisor value
7..0	DMR	00h	00..FFh	The MSB of Divisor value

***Table 1-6* • Baud Rates and Divisor Values for 18.432 MHz Reference Clock**

Baud Rate	Decimal Divisor (Divisor Value)	Percent Error
50	23,040	0.0000%
75	15,360	0.0000%
110	10,473	–0.2865%
134.5	8,565	0.0876%
150	7,680	0.0000%
300	3,840	0.0000%
600	1,920	0.0000%
1,200	920	4.3478%
1,800	640	0.0000%
2,000	576	0.0000%
2,400	480	0.0000%
3,600	320	0.0000%
4,800	240	0.0000%
7,200	160	0.0000%
9,600	120	0.0000%
19,200	60	0.0000%
38,400	30	0.0000%
56,000	21	–2.0408%

Interrupt Enable Register

The Interrupt Enable Register is defined in Table 1-7.

Table 1-7 **• Interrupt Enable Register**

Bits	Name	Default State	Valid State	Function
0	ERBFI	0	0, 1	Enables Received Data Available Interrupt 0 – Disabled 1 – Enabled
1	ETBEI	0	0, 1	Enables the Transmitter Holding Register Empty Interrupt 0 – Disabled 1 – Enabled
2	ELSI	0	0, 1	Enables the Receiver Line Status Interrupt 0 – Disabled 1 – Enabled
3	EDSSI	0	0, 1	Enables the Modem Status Interrupt 0 – Disabled 1 – Enabled
7..4	Reserved	0	0	Always 0

Interrupt Identification Register

The Interrupt Identification Register is defined in Table 1-8.

Table 1-8 **• Interrupt Identification Register**

Bits	Name	Default State	Valid States	Function
3..0	IIR	1h	0..Ch	Interrupt Identification bits. Described in Table 1-9.
5..4	Reserved	00	00	Always 00
7..6	Mode	11	11	11 – FIFO mode

The Interrupt Identification Register field is defined in Table 1-9.

Table 1-9 **• Interrupt Identification Register Field (IIR)**

IIR Value[3:0)]	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
0110	Highest	Receiver Line Status	Overrun error, parity error, framing error, or break interrupt	Reading the Line Status Register
0100	Second	Received Data Available	Receiver Data Available	Reading the Receiver Buffer Register or the FIFO drops below the trigger level
1100	Second	Character Timeout Indication	No characters have been read from the RX FIFO during the last four character times and there was at least one character in it during this time.	Reading the Receiver Buffer Register
0010	Third	Transmitter Holding Register Empty	Transmitter Holding Register Empty	Reading the IIR or writing into the Transmitter Holding Register
0000	Fourth	Modem Status	Clear to Send, Data Set Ready, Ring Indicator, or Data Carrier Detect	Reading the modern Status Register

Line Control Register

The Line Control Register is defined in Table 1-10.

Table 1-10 **• Line Control Register**

Bits	Name	Default State	Valid States	Function
1..0	WLS	0	0..3h	Word Length Select 00 – 5 bits 01 - 6 bits 10 – 7 bits 11 – 8 bits
2	STB	0	0, 1	Number of Stop Bits 0 – 1 stop bit 1 – 1½ stop bits when WLS = 00 2 stop bits in other cases
3	PEN	0	0, 1	Parity Enable 0 – Disabled 1 – Enabled. Parity is added in transmission and checked in receiving.
4	EPS	0	0, 1	Even Parity Select 0 – Odd parity 1 – Even parity

***Table 1-10* • Line Control Register (continued)**

5	SP	0	0, 1	Stick Parity 0 – Disabled 1 – Enabled When stick parity is enabled, it works as follows: Bits 4..3 11 – 0 will be sent as a parity bit, and checked in receiving. 01 – 1 will be sent as a parity bit, and checked in receiving.
6	SB	0	0, 1	Set Break 0 – Disabled 1 – Set break. SOUT is forced to 0. This does not have any effect on transmitter logic. The break is disabled by setting the bit to 0.
7	DLAB	0	0, 1	Divisor Latch Access Bit 0 – Disabled. Normal addressing mode in use 1 – Enabled. Enables access to the Divisor Latch registers during read or write operation to addresses 0 and 1.

Modem Control Register

The Modem Control Register is defined in Table 1-11.

***Table 1-11* • Modem Control Register**

Bits	Name	Default State	Valid States	Function
0	DTR	0	0, 1	Controls the Data Terminal Ready (DTRn) output. 0 – DTRn <= 1 1 – DTRn <= 0
1	RTS	0	0, 1	Controls the Request to Send (RTSn) output. 0 – RTSn <= 1 1 – RTSn <= 0
2	Out1	0	0, 1	Controls the Output1 (OUT1n) signal. 0 – OUT1n <= 1 1 – OUT1n <= 0
3	Out2	0	0, 1	Controls the Output2 (OUT2n) signal. 0 – OUT2n <=1 1 – OUT2n <=0

Table 1-11 • Modem Control Register

4	Loop	0	0, 1	Loop enable bit 0 – Disabled 1 – Enabled. The following happens in loop mode: SOUT is set to 1. The SIN, DSRn, CTSn, RIn, and DCDn inputs are disconnected. The output of the Transmitter Shift Register is looped back into the Receiver Shift Register. The modem control outputs (DTRn, RTSn, OUT1n, and OUT2n) are connected internally to the modem control inputs, and the modem control output pins are set at 1. In loopback mode, the transmitted data is immediately received, allowing the CPU to check the operation of the UART. The interrupts are operating in loop mode.
7..4	Reserve d	0h	0	Reserved

Line Status Register

The Line Status Register is defined in Table 1-12.

Table 1-12 • Line Status Register (read only)

Bits	Name	Default State	Valid States	Function
0	DR	0	0, 1	Data Ready indicator 1 when a data byte has been received and stored in the receive buffer or the FIFO. DR is cleared to 0 when the CPU reads the data from the receive buffer or the FIFO.
1	OE	0	0, 1	Overrun Error indicator Indicates that the new byte was received before the CPU read the byte from the receive buffer, and that the earlier data byte was destroyed. OE is cleared when the CPU reads the Line Status Register. If the data continues to fill the FIFO beyond the trigger level, an overrun error will occur once the FIFO is full and the next character has been completely received in the shift register. The character in the shift register is overwritten, but it is not transferred to the FIFO.
2	PE	0	0, 1	Parity Error indicator Indicates that the received byte had a parity error. PE is cleared when the CPU reads the Line Status Register. This error is revealed to the CPU when its associated character is at the top of the FIFO.
3	FE	0	0, 1	Framing Error indicator Indicates that the received byte did not have a valid Stop bit. FE is cleared when the CPU reads the Line Status Register. The UART will try to resynchronize after a framing error. To do this, it assumes that the framing error was due to the next start bit, so it samples this start bit twice, and then starts receiving the data. This error is revealed to the CPU when its associated character is at the top of the FIFO.

Table 1-12 **• Line Status Register (read only)**

4	BI	0	0, 1	Break Interrupt indicator Indicates that the received data is at 0 longer than a full word transmission time (start bit + data bits + parity + stop bits). BI is cleared when the CPU reads the Line Status Register. This error is revealed to the CPU when its associated character is at the top of the FIFO. When break occurs, only one zero character is loaded into the FIFO.
5	THRE	1	0, 1	Transmitter Holding Register Empty indicator Indicates that the UART is ready to transmit a new data byte. THRE causes an interrupt to the CPU when bit 1 (ETBEI) in the Interrupt Enable Register is 1. This bit is set when the TX FIFO is empty. It is cleared when at least one byte is written to the TX FIFO.
6	TEMT	1	0, 1	Transmitter Empty indicator This bit is set to 1 when both the transmitter FIFO and shift registers are empty.
7	FIER	0	1	This bit is set when there is at least one parity error, framing error, or break indication in the FIFO. FIER is cleared when the CPU reads the LSR if there are no subsequent errors in the FIFO.

Modem Status Register

The Modem Status Register is defined in Table 1-13.

Table 1-13 **• Modem Status Register (read only)**

Bits	Name	Default State	Valid States	Function
0	DCTS	0	0, 1	Delta Clear to Send indicator. Indicates that the CTSn input has changed state since the last time it was read by the CPU.
1	DDSR	0	0, 1	Delta Data Set Ready indicator Indicates that the DSRn input has changed state since the last time it was read by the CPU.
2	TERI	0	0, 1	Trailing Edge of Ring Indicator detector. Indicates that RI input has changed from 0 to 1.
3	DDCD	0	0, 1	Delta Data Carrier Detect indicator Indicates that DCD input has changed state. NOTE: Whenever bit 0, 1, 2, or 3 is set to 1, a Modem Status Interrupt is generated.
4	CTS	0	0, 1	Clear to Send The complement of the CTSn input. When bit 4 of the Modem Control Register (MCR) is set to 1 (loop), this bit is equivalent to DTR in the MCR.

Table 1-13 **• Modem Status Register (read only)**

5	DSR	0	0, 1	Data Set Ready The complement of the DSR input. When bit 4 of the MCR is set to 1 (loop), this bit is equivalent to RTSn in the MCR.
6	RI	0	0, 1	Ring Indicator The complement of the RIn input. When bit 4 of the MCR is set to 1 (loop), this bit is equivalent to OUT1 in the MCR.
7	DCD	0	0, 1	Data Carrier Detect The complement of DCDn input. When bit 4 of the MCR is set to 1 (loop), this bit is equivalent to OUT2 in the MCR.

Scratch Register

The Scratch Register is defined in Table 1-14.

Table 1-14 **• Scratch Register**

Bits	Name	Default State	Function
7..0	SCR	00h	Read/write register for CPU. No effects on UART operation.

2 – Tool Flows

Licenses

Core16550 is licensed in two ways. Depending on your license, tool flow functionality may be limited.

Obfuscated

Complete RTL code is provided for the core, enabling the core to be instantiated with Libero SoC. Simulation, Synthesis, and Layout can be performed with Libero SoC. The RTL code for the core is obfuscated, and some of the testbench source files are not provided. They are pre-compiled into the compiled simulation library instead.

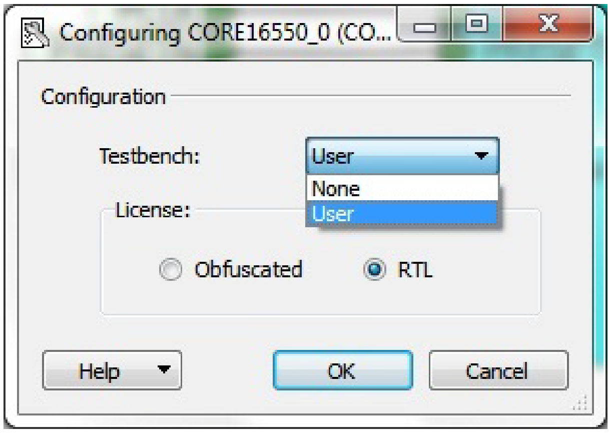
RTL

Complete RTL source code is provided for the core and testbenches.

SmartDesign

Core16550 is available for download in the SmartDesign IP deployment design environment. The core can be configured using the configuration GUI within SmartDesign, as shown in Figure 2-1.

For more information on using SmartDesign to instantiate and generate cores, refer to the *Using DirectCore in Libero SoC User's Guide*.



Testbench: User

License: ○ Obfuscated ● RTL

Figure 2-1 **• Core16550 Configuration**

Simulation Flows

The user testbench for Core16550 is included in all releases.

To run simulations, select the user testbench flow within SmartDesign and click **Generate Design** under the SmartDesign menu. The user testbench is selected through the Core Testbench Configuration GUI.

When SmartDesign generates the Libero SoC project, it will install the user testbench files.

To run the user testbench, set the design root to the Core16550 instantiation in the Libero SoC **Design Hierarchy** pane and click the **Simulation** icon in the SoC Design Flow window. This will invoke ModelSim® and automatically run the simulation.

Synthesis in Libero SoC

Click the **Synthesis** icon in Libero SoC. The Synthesis window appears, displaying the Synplify® project. Set Synplify to use the Verilog 2001 standard if Verilog is being used. To run Synthesis, click the **Run** icon.

Place-and-Route in Libero SoC

Having set the design route appropriately and run Synthesis, click the **Layout** icon in Libero SoC to invoke Designer. Core16550 requires no special place-and-route settings.

3 – Core16550

Parameters

Core16550 has a top-level parameter/generic that is used to select the device family (Table 3-1).

***Table 3-1* • Core Parameters**

Parameter Name	Description	Allowed Values	Default
FAMILY	Must be set to match the supported FPGA family. 11 – Axcelerator 12 – RTAX-S 14 – ProASIC$^{\underline{\text{PLUS}}}$ 15 – ProASIC3 16 – ProASIC3E 17 – Fusion 18 – SmartFusion 19 – SmartFusion2 20 – IGLOO 21 – IGLOOe 24 – IGLOO2 25 – RTG4	11 to 25	15

4 – Core Interfaces

I/O Signal Description

Core16550 I/O definitions are given in Table 4-1.

Table 4-1 **• I/O Signal Summary**

Name	Type	Polarity	Description
PRESETN	In	LOW	Master reset
PCLK	In	–	Master clock PCLK is divided by the value of the Divisor Registers. The result is then divided again by 16 to produce the baud rate. The resultant signal is the BAUDOUT signal. The rising edge of this pin is used to strobe all input and output signals.
PWRITE	In	HIGH	APB write/read enable, active high When HIGH, data is written to the specified address location. When LOW, data is read from the specified address location.
PADDR[4:0]	In		APB Address This bus provides the link for the CPU to the address of the register of Core16550 to be read from or written to.
PSEL	In	HIGH	APB select When this is HIGH along with PENABLE, reading and writing to Core16550 is enabled.
PWDATA[7:0]	In	–	Data input bus Data on this bus will be written into the addressed register during a write cycle.
PENABLE	In	HIGH	APB enable When this is HIGH along with PSEL, reading and writing to Core16550 is enabled.
PRDATA[7:0]	Out	–	Data output bus This bus will hold the value of the addressed register during a read cycle.
CTSn	In	LOW	Clear to Send This active low signal is an input showing when the attached device (modem) is ready to accept data. Core16550 passes this information to the CPU via the Modem Status register. This register also indicates if the CTSn signal has changed since the last time the register was read.
DSRn	In	LOW	Data Set Ready This active low signal is an input indicating when the attached device (modem) is ready to set up a link with Core16550. Core16550 passes this information to the CPU via the Modem Status Register. This register also indicates if the DSRn signal has changed since the last time the register was read.
DCDn	In	LOW	Data Carrier Detect This active low signal is an input indicating when the attached device (modem) has detected a carrier. Core16550 passes this information to the CPU via the Modem Status Register. This register also indicates if the DCDn signal has changed since the last time the register was read.

Table 4-1 • **I/O Signal Summary (continued)**

Name	Type	Polarity	Description
SIN	In	–	Serial Input Data This is the data that will be transmitted into Core16550. It is synchronized with the PCLK input pin.
RIn	In	LOW	Ring Indicator This active low signal is an input showing when the attached device (modem) has sensed a ring signal on the telephone line. Core16550 passes this information to the CPU via the Modem Status Register. This register also indicates when the RIn trailing edge was sensed.
SOUT	Out	–	Serial output data This is the data that will be transmitted from Core16550. It is synchronized with the BAUDOUT output pin.
RTSn	Out	LOW	Request to Send This active low output signal is used to inform the attached device (modem) that Core16550 is ready to send data. It is programmed by the CPU via the Modem Control Register.
DTRn	Out	LOW	Data Terminal Ready This active low output signal informs the attached device (modem) that Core16550 is ready to establish a communications link. It is programmed by the CPU via the Modem Control Register.
OUT1n	Out	LOW	Output 1 This active low output is a user-defined signal. It is programmed by the CPU via the Modem Control Register and is set to the opposite value. programmed.
OUT2n	Out	LOW	Output 2 This active low output signal is a user-defined signal. It is programmed by the CPU via the Modem Control Register and is set to the opposite value. programmed.
INTR	Out	HIGH	Interrupt Pending This active high output signal is the interrupt output signal from Core16550. It can be programmed to become active on certain events, informing the CPU that such an event has occurred (see "Interrupt Identification Register" on page 9 for details). The CPU can then take appropriate action.
BAUDOUTn	Out	LOW	Baud out This is an output clock signal derived from the input clock (see PCLK description) for synchronizing the data output stream from SOUT.
RXRDYN	Out	LOW	Receiver ready to receive transmissions This active low output signal indicates to the CPU that the receiver section of Core16550 is available for data to be read.
TXRDYN	Out	LOW	Transmitter ready to transmit data This active low signal indicates to the CPU that the transmitter section of Core16550 has space to write data for transmission.
rxfifo_empty	Out	High	Receive FIFO empty This signal goes HIGH when the receive FIFO is empty.
rxfifo_full	Out	High	Receive FIFO full This signal goes HIGH when the receive FIFO is full.

5 – Timing Diagrams

Data Write Cycle and Data Read Cycle

Figure 5-1 and Figure 5-2 depict write cycle and read cycle timing relationships relative to the APB system clock, PCLK.

Register Write

As shown in Figure 5-1, the Address, Select, and Enable signals are latched and must be valid prior to the rising edge of PCLK. Writing occurs at the rising edge of the PCLK signal.

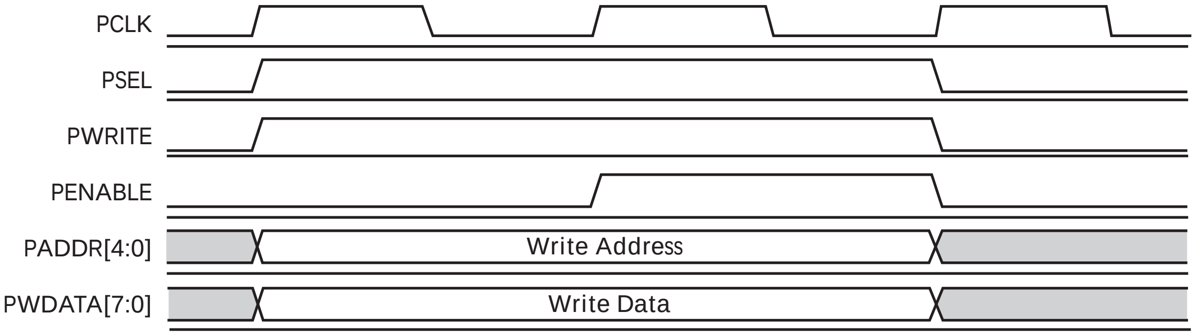


Figure 5-1 **• Data Write Cycle**

Register Read

As shown in Figure 5-2, the Address, Select, and Enable signals are latched and must be valid prior to the rising edge of PCLK. Reading occurs at the rising edge of the PCLK signal.

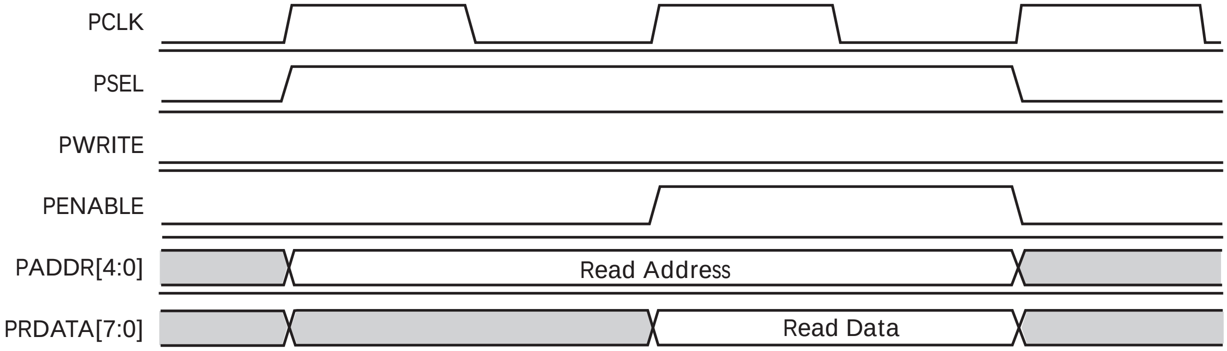


Figure 5-2 **• Data Read Cycle**

More detailed descriptions and timing waveforms can be found in the AMBA specification: http://www.amba.com/products/solutions/AMBA_Spec.html.

Receiver Synchronization

When the Receiver detects a LOW state in the incoming data stream, it will synchronize to it. After the start edge, the UART will wait 1.5 × (the normal bit length). This causes each subsequent bit to be read at the middle of its width. Figure 5-3 depicts this synchronization process.

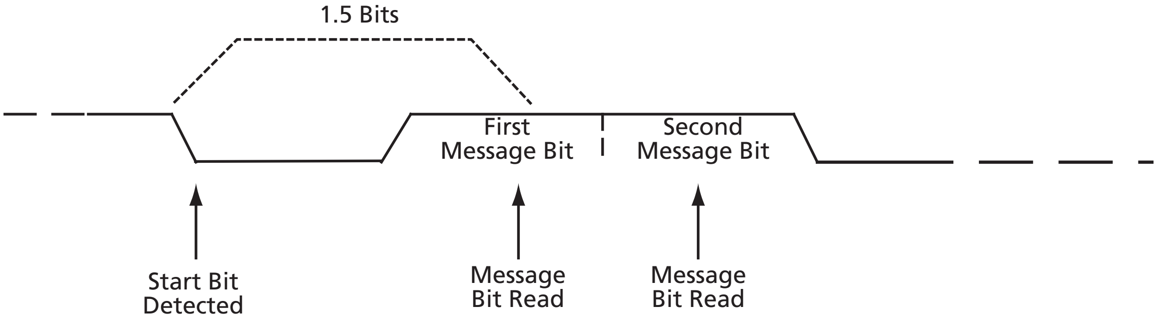


***Figure 5-3* • Receiver Synchronization**

6 – Testbench Operation

Two testbenches are provided with Core16550:

- VHDL user testbench: Simple-to-use testbench written in VHDL. This testbench is intended for customer modification.
- Verilog user testbench: Simple-to-use testbench written in Verilog. This testbench is intended for customer modification.

User Testbench

A block diagram of the example user design and testbench is shown in Figure 6-1.

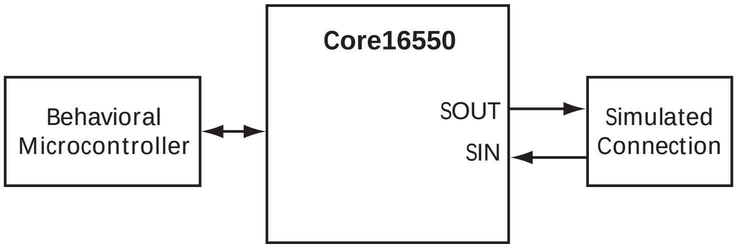


Figure 6-1 **• Core16550 User Testbench**

The user testbench includes a simple example design that serves as a reference for users who want to implement their own designs.

The testbench for the example user design implements a subset of the functionality tested in the verification testbench, described in "User Testbench" on page 23. Conceptually, as shown in Figure 6-1, instantiation of Core16550 is simulated using a behavioral microcontroller and a simulated loopback connection. Example transmit and receive by the same Core16550 unit are demonstrated by the user testbench so you can gain a basic understanding of how to use this core.

The user testbench demonstrates the basic setup, transmit, and receive operations of Core16550. The user testbench exercises the following:

1. Write to the control registers.
2. Read the control registers.
3. Turn on transmit and receive.
4. Transmit and receive one byte.
5. Check received data.

7 – List of Changes

The following table lists critical changes that were made in each revision of the chapter.

Date	Changes	Page
50200110-4/2.15	Added 'IGLOO2' and 'RTG4' to the "Supported Device Families" section and to Table 1 in the "Device Utilization and Performance" section.	4
50200110-3/12.12	Added 'SmartFusion2' to the "Supported Device Families" section and to Table 1 in the "Device Utilization and Performance" section (SAR 43254).	4
50200110-2/4.12	Added 'SmartFusion' to the "Supported Device Families" section and to Table 1 in the "Device Utilization and Performance" section (SAR 36238).	4
	Added 'AGLE600V2' to Table 1 in the "Device Utilization and Performance" section, as also modified the '**Cell or Tiles**' values for SmartFusion (SAR 11944).	4
	Updated Default (reset) Value for IIR and FCR in Table 1-1, **Core16550 Register Summary** (SAR 37805).	6
	Updated FCR[3] definition in Table 1-4 (SAR 22657).	7
	Updated **Default State** for 7..6 Bit definition in Table 1-8 (SAR 36238).	9
	Replaced Figure 2-1 with a new image (SAR 36238).	15
	Deleted the 'Evaluation' section and modified the Obfuscated section (SAR 36238).	15
	Added references to Libero SoC, and the SmartDesign and Simulation Flows sections (SAR 36238).	15, 16
	Removed all references to CoreConsole and Libero IDE, deleted the 'CoreConsole' and 'Importing into Libero IDE' sections (SAR 36238).	15, 16
	Renamed the Place-and-Route in Libero IDE section to "Place-and-Route in Libero SoC" section(SAR 36238).	16
	Added 'SmartFusion' Core Parameter to the Table 3-1 (SAR 36238).	17
	Deleted the 'Verification Testbench' section (SAR 37805). Evaluation license/RTL no longer relevant as Obfuscated licenses are available free.	23
50200110-1/1.08	The "Supported Device Families" section was added in v2.1.	4

Note: *The part number is located on the last page of the document. The digits following the slash indicate the month and year of publication.

8 – Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call 800.262.1060
From the rest of the world, call 650.318.4460
Fax, from anywhere in the world, 650.318.8044

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues, and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

For Microsemi Technical Support, visit http://www.microsemi.com/design-support/technical-support.

Website

You can browse a variety of technical and non-technical information on the SoC home page, at www.microsemi.com/soc.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Sales office listings can be found at www.microsemi.com/soc/company/contact/default.aspx.

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech_itar@microsemi.com. Alternatively, within My Cases, select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.

Index

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; security technologies and scalable anti-tamper products; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 3,400 employees globally. Learn more at **www.microsemi.com**.

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

E-mail: sales.support@microsemi.com

© 2015 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.