MOXSYN®

C16550 Universal Asynchronous Receiver/Transmitter with FIFOs

Function Description

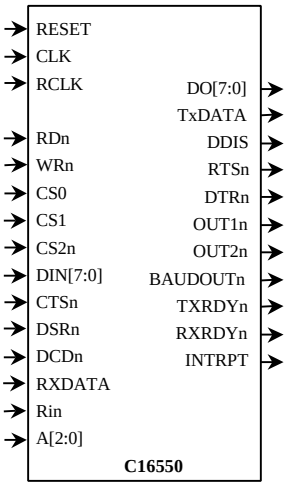
The C16550 programmable asynchronous communications interface (UART) core provides data formatting and control to a serial communication channel.

The core has select, read/write, interrupt and bus interface logic features that allow data transfers over an 8-bit bi-directional parallel data bus system. With proper formatting and error checking, the core can transmit and receive serial data, supporting asynchronous operation.

Features

- Capable of running with all existing 16450 and 16550A Software
- Asynchronous operation
- In FIFO mode, Transmitter and Receiver are each buffered with 16-byte FIFOs to reduce the number of interrupts of the CPU
- Programmable data word length (5 - 8 bit), parity and stop bits
- Parity, overrun and framing error checking
- Supports up to 3 Mbps transmission rates
- (Typical value: silicon dependent)
- Programmable Baud Rate Generator allows division of any reference clock by 1 to (2^{16}-1) and generates an internal 16 X Clock
- False start bit detection
- Automatic break generation and detection
- Internal diagnostic capabilities
- Peripheral modem control functions
- The C16550 is available in VHDL or Verilog and synthesizes to approximately 6,500 gates depending on the process used

Symbol



Pin Description

Name	Type	Polarity	Description
RESET	In	High	External reset
CLK	In	-	Master clock
RCLK	In	-	Receive clock
RDn	In	Low	Read control
WRn	In	Low	Write control
CS0	In	High	Chip Select 0
CS1	In	High	Chip Select 1
CS2n	In	Low	Chip Select 2
DIN[7:0]	In	-	Data Input Bus
CTSn	In	Low	Clear-to -Send
DSRn	In	Low	Data Set Ready
DCDn	In	Low	Data Carrier Detect
RXDATA	In	-	Receive Data
RIn	In	Low	Ring Indicator
A[2:0]	In	-	Register Select
D0[7:0]	Out	-	Data Output Bus
TXDATA	Out	-	Transmit Data
DDIS	Out	High	Driver Disable
RTSn	Out	Low	Request-to-Send
DTRn	Out	Low	Data Terminal Ready
OUT1n	Out	Low	Output 1
OUT1n	Out	Low	Output 2
TxRDYn	Out	Low	Transmit ready
RxRDYn	Out	Low	Receiver ready
INTRPT	Out	High	Interrupt
BAUDOUTn	Out	Low	Baud Out

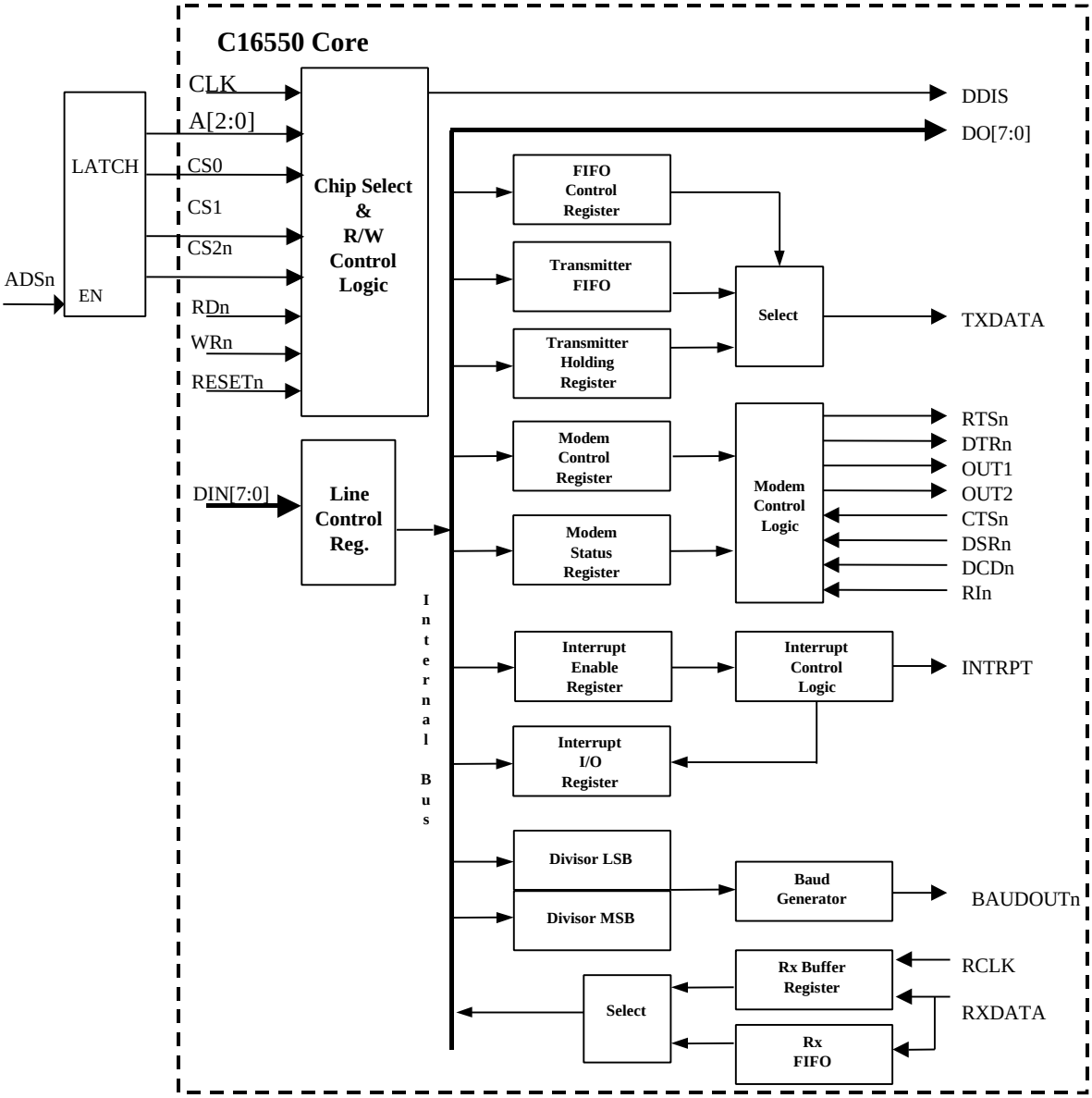
Register description

The C16550 contains the following registers:

1. Line Control
2. Line Status
3. Interrupt Enable
4. Modem status
5. Modem Control
6. Transmitter Holding buffer
7. Receiver buffer
8. Interrupt Identification
9. FIFO control
10. Scratch

Block Diagram

Figure 1: C16550 UART Block Diagram



Applications

- Serial data communications applications
- Modem interface

Functional Description

Line Control (LCR)

The Line Control Register is used to specify the data communication format. The break feature, parity, stop bits and word length can be changed by writing to the appropriate bits in LSR.

Line Status (LSR)

This register provides information on the status of data transfers between the C16550 and the CPU.

Interrupt Enable (IER)

The Interrupt Enable Register masks interrupts from the modem status registers, line status, transmitter empty and receiver ready to the INTRPT output pin.

Modem Status (MSR)

This register provides the current state of modem control lines.

Modem Control (MCR)

This register controls the interface lines with the MODEM and changes the status of the C16550 from normal operating mode and local loop-back mode (diagnostics mode).

Transmitter Holding Buffer

The transmitter section is composed of a Transmit Holding Register (THR) and a Transmit Shift Register (TSR). Writing to THR will transfer the contents of the data bus (DIN 7-0) to the Transmit Holding Register every time that the THR or TSR is empty. This write operation should be done when Transmit Holding Register Empty (THRE) is set.

Receiver Buffer

This register contains the assembled received data. On the falling edge of the start bit, the receiver section starts its operations. The start bit is valid if the RXDATA is still low at the middle sample of Start bit, thus preventing the receiver from assembling a false data character.

Interrupt Identification (IIR)

The Interrupt Identification Register provides the source of interrupt among four levels of prioritized interrupt conditions in order to minimize the CPU overhead during data transfers.

FIFO control register (FCR)

The FCR is a write-only register at the same location as the IIR, which is a read-only register. The FCR enables the FIFOs, clears the FIFOs, sets the receiver FIFO trigger level, and selects the type of DMA signaling.

Scratchpad (SR)

This register stores the temporary byte for variable use.

Core Assumptions

The functionality of the C16550 core was based on the Texas Instruments TL16C550A. The following characteristics differentiate the C16550 from the Texas Instruments device:

- This core does not support connection of a crystal directly to the device. It was replaced by an external master clock signal (CLK).
- The bi-directional data bus has been split into two separate buses: DIN[7:0] and DO[7:0]. If required these buses can be combined into a single 8-bit bi-directional bus.
- Signals rd2, wr2, xin and xout have been eliminated from the interface.
- Signal ADSn and address latch have been removed.
- The 1½ stop bit mode (for 5-bit word length) is not supported.
- Divisor latch low (DLL), divisor latch high (DLM), THR register and SCR register are reset to all zeros.
- Both DLL and DLM registers must be loaded before BAUDOUTn has valid output.
- DO (output data bus) always shows the last selected register; after reset DO is "00".
- MSR register is reset during the clock cycle following a MSR read.
- Transmitter Empty status (TEMT), bit 6 of Line Status Register (LSR), is reset during the clock cycle following a TBR register write.
- Between 2 rising edges of WRn or RDn the core needs at least 2 clock (CLK) cycles. (Tcycle)

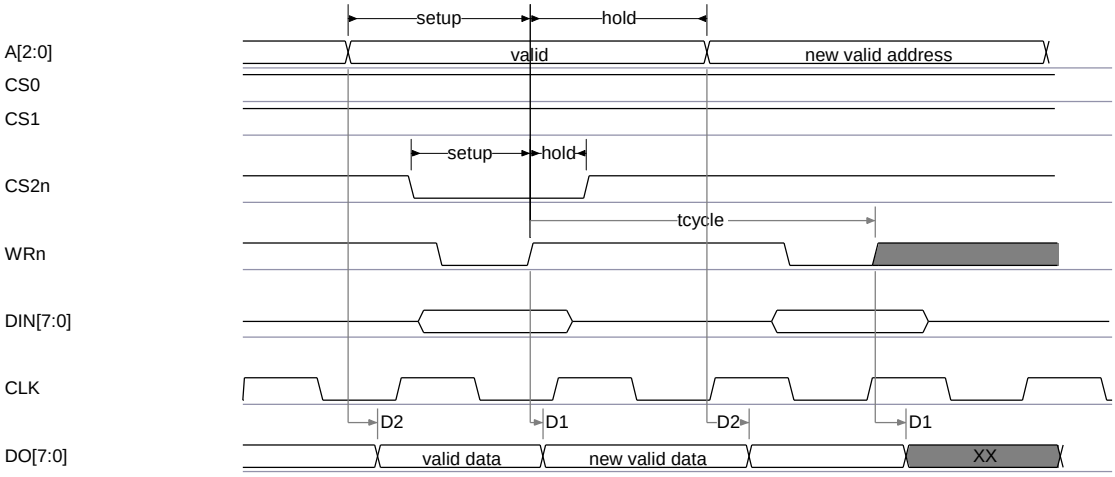
Timing variations differences from the Texas Instruments device:

Symbol	Parameter	C16550		TL16C550A	
		Min	Max	Min	Max
td_1	Delay from WRn MCR to Output		tpd		100 ns
td_2	Delay from MODEM Input to Set Interrupt		2 CLK Cycles		170 ns
td_3	Delay from RDn to Reset Interrupt(RD MDR)		tpd after first CLK rising edge		140 ns
td_4	Delay from WRn to Reset Interrupt(WR THR)		2 CLK Cycles		140 ns
td_5	Delay from RDn to Reset Interrupt THRE (RD IIR)		tpd		140 ns
td_6	Delay from initial INTR Reset to Transmit Start	1 Baudout Cycles	4 Baudout Cycles	8 Baudout Cycles	24 Baudout Cycles
td_7	Delay from initial Write to Interrupt	4 Baudout Cycles	4 Baudout Cycles	16 Baudout Cycles	32 Baudout Cycles
td_8	Delay from STOP to Interrupt (THRE)	1 CLK Cycle	1 CLK Cycle	8 Baudout Cycles	8 Baudout Cycles
td_9	Delay from START to TXRDYn active		1 CLK Cycle		8 Baudout Cycles
td_{10}	Delay from Write to TXRDYn inactive	1 CLK Cycle	2 CLK Cycles		195 ns
td_{11}	Delay from STOP to Set Interrupt		tpd		1-3 RCLK Cycles
td_{12}	Delay from RD RBR to RXRDYn inactive		tpd		150 ns
td_{13}	Address, ADSn, Chip Select setup time [1]			15 ns	
td_{14}	Address, ADSn, Chip Select hold time [1]			5 ns	

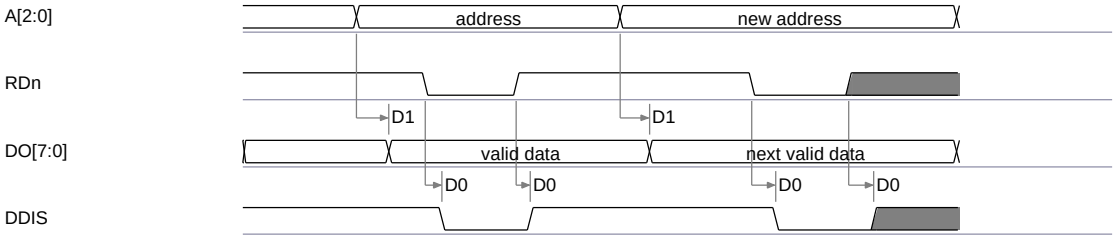
[1] Typical values: silicon dependent

All delays : setup, hold, D0, D1, D2 are silicon dependent

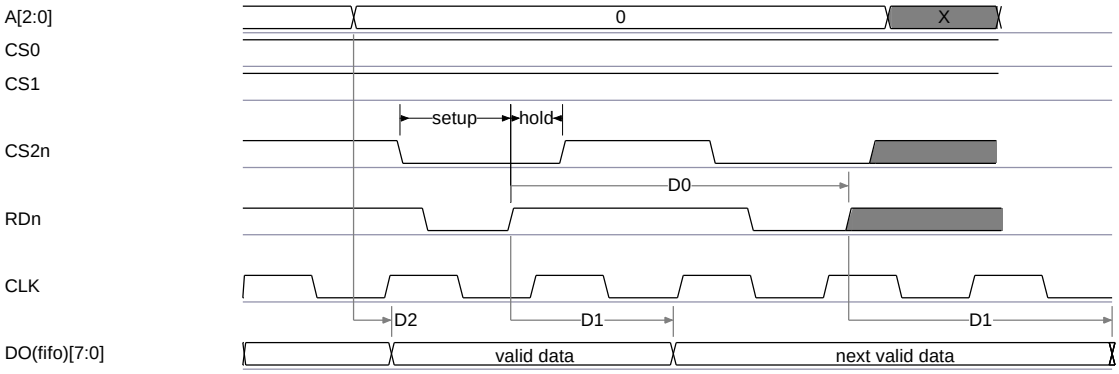
Timing Diagrams



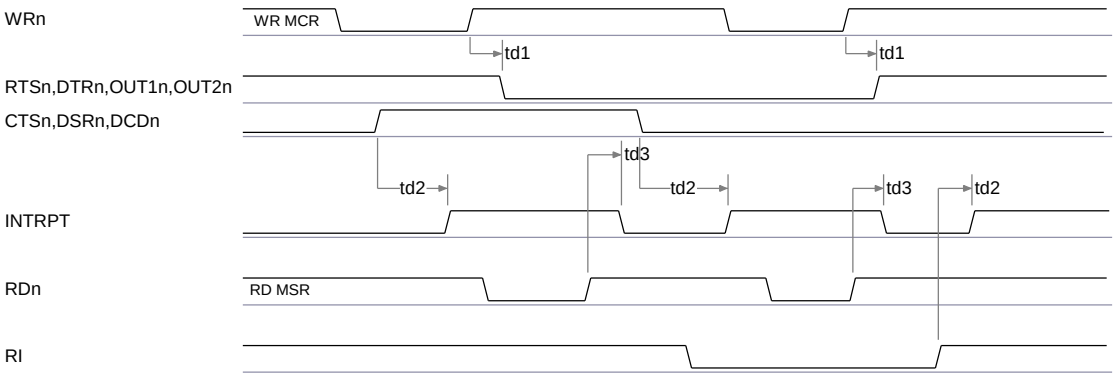
Write Cycle Timing



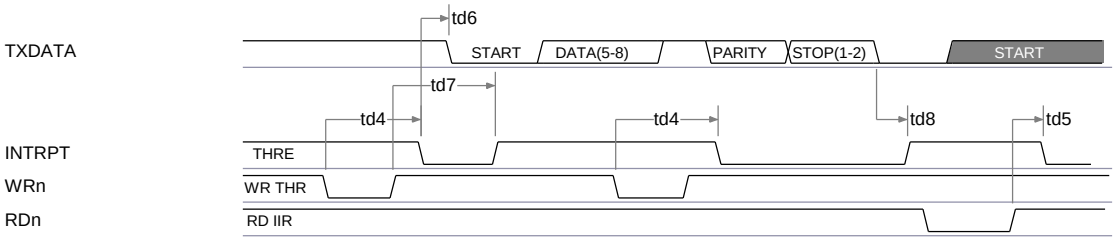
Read Cycle Timing



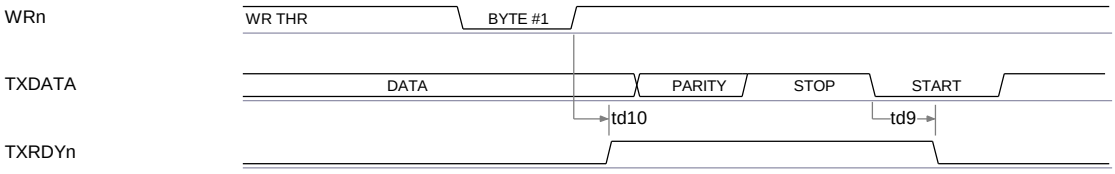
Read of FIFO Timing



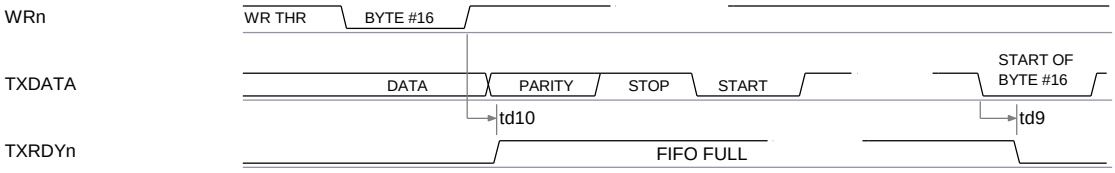
MODEM Control Timing



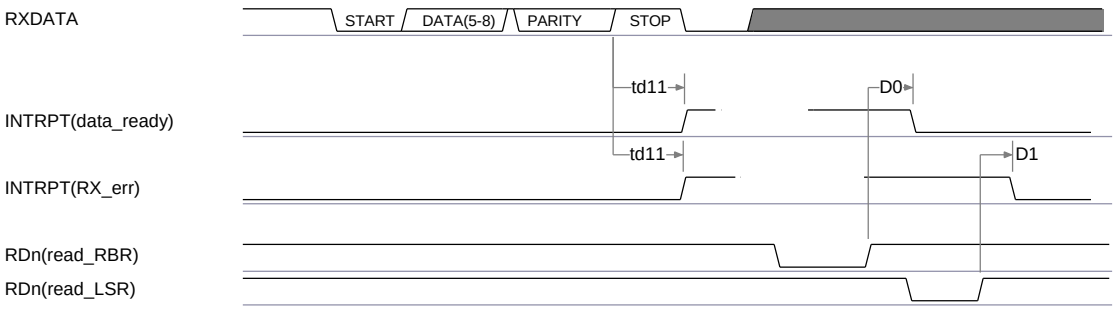
Transmitter Timing



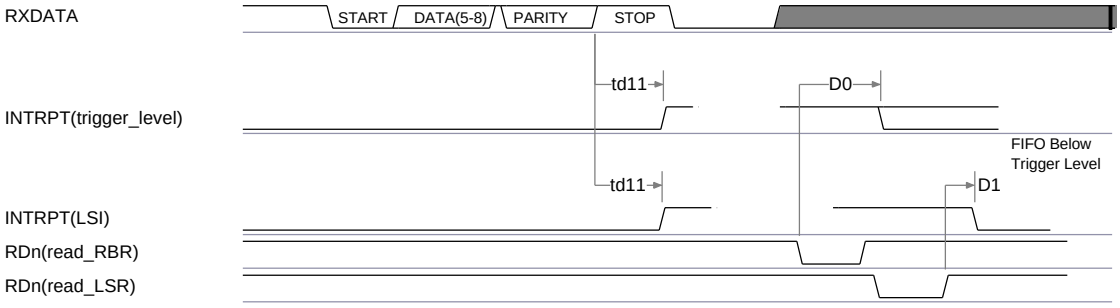
Transmitter Ready, FCR0 = 0 or FCR0 = 1 and FCR3 = 0 (Mode 0)



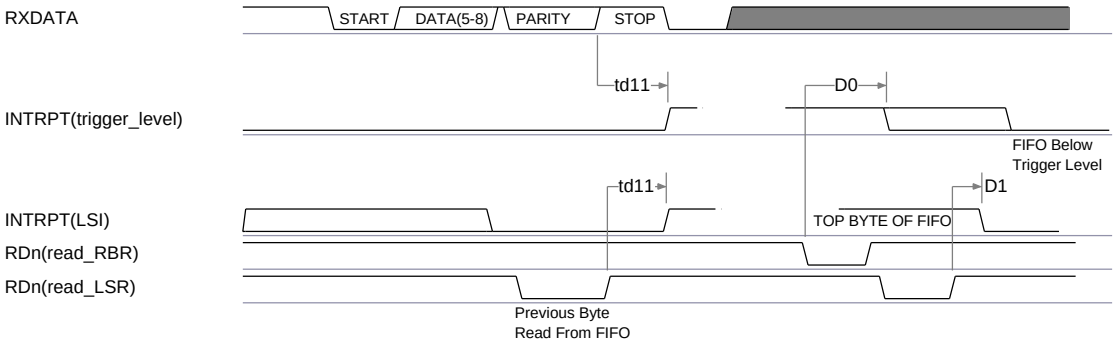
Transmitter Ready, FCR0 = 1 and FCR3 = 1 (Mode 1)



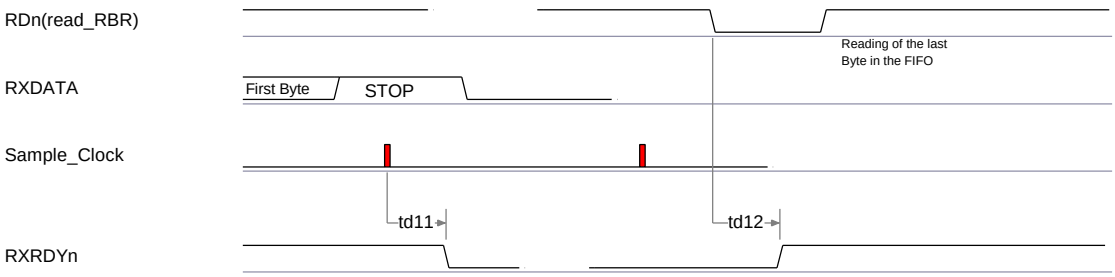
Receiver Timing



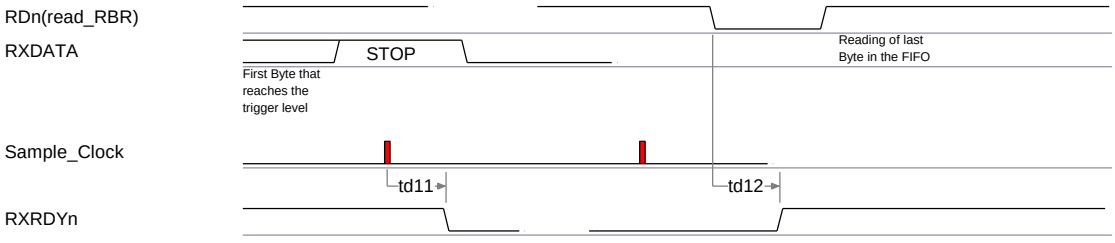
Receiver FIFO first Byte(Sets DR bit)



Receiver FIFO other than first Byte(DR internal bit already set)



Receiver Ready, FCR0 =0 or FCR0=1 and FCR3 = 0 (Mode 0)



Receiver Ready, FCR0 =1 and FCR3 = 1 (Mode 1)

Supplied Files

readme.txt	General information about the core and installation
doc/	
C16550.pdf	This document
release.txt	Release notes
src/	
C16550.v	Top Level module (Verilog version)
C16550.vhd	Top Level entity/architecture (VHDL version)
tb/	
C16550tb.v	Testfixture (Verilog version)
C16550ts.vhd	Testbench (VHDL version)
C16550.stm	Stimulus file used by the testbench (VHDL version)
C16550.ref	Expected testbench results (VHDL version)
C16550_v.ref	Expected testbench results (Verilog version)
scripts/	
compile.do	Sample compilation script for ModelSim
simulatate.do	Sample simulation script for ModelSim
synopsys.scr	Sample Synopsys synthesis script for Synopsys

Testbench Notes (VHDL version)

The testbench provided can be used for functional and post-synthesis simulation, and is used to test the functionality of the core.

The stimulus file, *C16550.stm* and the expected results file *C16550.ref* should be in the local simulation directory.

Functional Simulation

Once the testbench is loaded into the simulator, simulate for 1900 us. At the end of the simulation the following message should appear:

"Note: test completed with no errors"

Post-Synthesis Simulation

If your synthesis tool renames the top-level pins after routing, it may be necessary to create a dummy wrapper in order to use the supplied testbench.

Edit the file *C16550ts.vhd* and go to line 405 (SECTION 4) of the testbench
Add the actual maximum expected delays of your silicon
Compile your post-synthesis or post-layout VHDL netlist file
Recompile the *C16550ts.vhd* file just edited
Simulate as above

Testbench Notes (Verilog version)

The testfixture provided can be used for functional and post-synthesis simulation, and is used to test the functionality of the core.

The expected results file *C16550_v.ref* should be in the local simulation directory.

Functional Simulation

Once the testfixture is loaded into the simulator, simulate for 1900 us. At the end of the simulation the following message should appear:

"Note: test completed with no errors"

Post-Synthesis Simulation

If your synthesis tool renames the top-level pins after routing, it may be necessary to create a dummy wrapper in order to use the supplied testfixture.

Edit the file *C16550tb.v* and go to line 1100 of the textfixture
Add the actual maximum expected delays of your silicon
Compile your post-synthesis or post-layout Verilog netlist file
Recompile the *C16550tb.v* file just edited
Simulate as above

Synthesis Notes

A Synopsys sample synthesis script is provided in the *scripts* directory. The script will create VHDL and Verilog gate-level netlists, and gate/performance reports at the completion of the synthesis process. It is recommended that a directory *syn* be created where the source files reside. The script expects to find all HDL source files in "*../*". If this is not the case, then edit *synopsys.scr* and update with the correct path.

Core Modifications

The C16550 core can be customized to include:

- Different FIFOs size (separately for Transmitter and Receiver)
- Removal of internal baud rate generator
- Different CPU interface

Support

Every effort has been made to ensure that this core functions correctly. If a problem is encountered please contact MOXSYN:

Technical Support Hotline: +39-335-65-74-399
Fax: +39-736-43125
E-mail: mailto:support@moxsyn.com
URL: www.moxsyn.com

Related Information

Data Transmission Circuits 1993 Data Book

Contact:
Texas Instruments
Literature Response Center
P.O. Box 809066
Dallas, Texas 75380-9066
URL: http://www.ti.com

The C16550 core is licensed from Moxsyn S.r.l.