A VHDL 16550 UART core

An
www.OpenCores.org
Project

hlefevre@opencores.org

Revision History

Revision	Date	Author	Description
0.1	18 Feb 2006	H LeFevre	Release of TX and RX modules
0.2	25 Feb 2006	H LeFevre	Fist Alpha release of full project design, TX and RX modules updated
0.5	18 Mar 2006	H LeFevre	Beta release Update documentation, Transmit Hold Register Empty Interrupt will clear on read of IIR (in addition to clear on a FIFO write)
1.0	8 April 2006	H LeFevre	Add time out interrupt
1.1	6 May 2006	H. LeFevre	Add AMBA APB 2.0 wrapper
2.0	20 Jan 2007	H. LeFevre	Removes frequency limits on BR_clk for gh_uart_16550.vhd logic version 2.2 and above.
2.1	14 July 2007	H. LeFevre	Correct FCR bit 3 information (DMA Mode control)
2.2	13 Oct 2007	H. LeFevre	add note about THRE Interrupt

Table of Contents

1 Introduction

This core is designed to be a compatible with the National Semiconductor PC16550D UART (Universal Asynchronous Receiver/Transmitter).

Some differences:

The FIFO's are always enabled
Sticky Parity is not supported

1.1 Purpose

- Educational – to learn more about SOC design (at lest for me…)
- To provide an example how to make use of the GH VHDL Standard Parts Library.

1.2 The VHDL 16550 UART core License

2 The UART Core

Until the document is done (and perhaps later as well), those interested in understanding the UART better should consult the data sheets for the PC16550D.

The gh_vhdl_library (another www.opencores.org project- version 3.17 or later) is used extensively in this project, and needs to be downloaded separately.

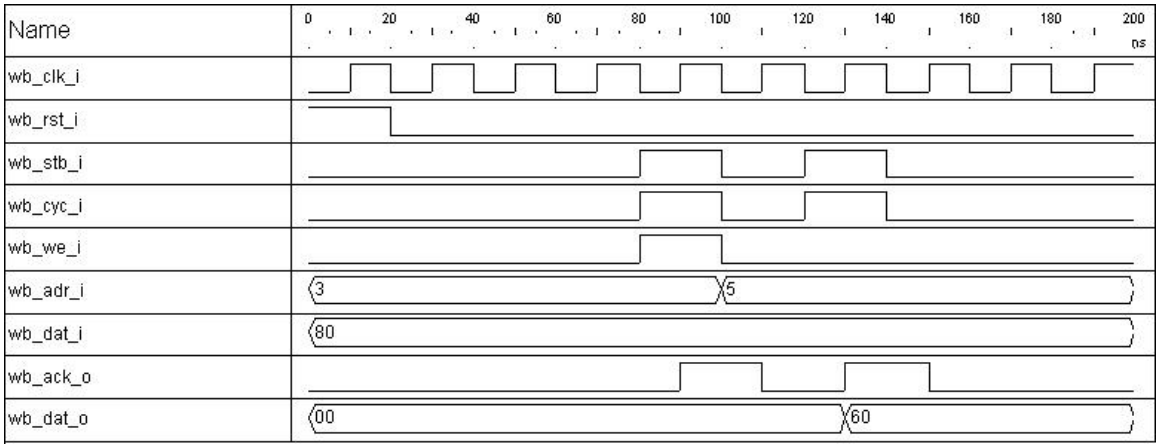
2.1 A Wishbone Wrapper

A Wishbone Wrapper has been added as an example of how to interface with the UART core.

I/O		Function
wb_clk_i	I	Clock, primary clock used in core – not used for the baud rate generator, the Tx module, or the Rx module
wb_rst_i	I	Asynchronous Reset, active high
wb_stb_i	I	Data strobe
wb_cyc_i	I	Cycle in process
wb_we_i	I	Write enable
wb_adr_i (2 downto 0)	I	Address bus
wb_dat_i (7 downto 0)	I	Input data bus
wb_ack_o	O	Data transfer acknowledge
wb_dat_o (7 downto 0)	O	Output Data bus

File name: gh_uart_16550_wb_wrapper.vhd

NOTE: The Wishbone Wrapper does not allow “burst mode” operation. The data strobe or cycle in process signals have to go low between transfers.



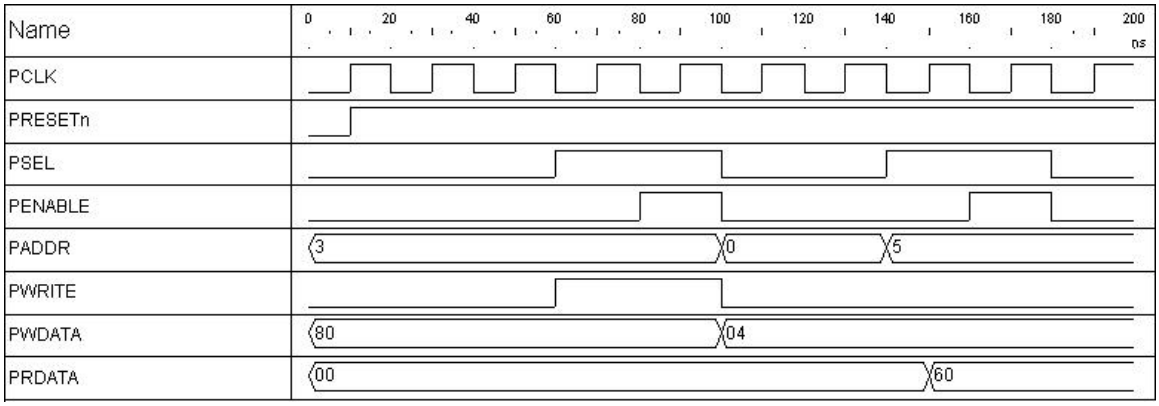
A Write cycle, followed by a Read cycle. The output data (wb_dat_o) is valid when wb_ack_o is active, and it is held until the next data transfer cycle.

2.2 A AMBA APB Wrapper

The AMBA APB bus 2.0 Wrapper is included as an example of interfacing the UART with the Peripheral bus used by the ARM family of processors.

I/O		Function
PCLK	I	Clock, primary clock used in core – not used for the baud rate generator, the Tx module, or the Rx module
PRESETn	I	Asynchronous Reset, active low
PSEL	I	Peripheral Select
PENABLE	I	Peripheral transfer Enable
PWRITE	I	Peripheral Write / READn Control
PADDR (2 downto 0)	I	Peripheral Address bus
PWDATA (7 downto 0)	I	Peripheral Data Write bus
PRDATA (7 downto 0)	O	Peripheral Data Read bus

File name: gh_uart_16550_AMBA_APB_wrapper.vhd



A write cycle, followed by a read cycle.

2.3 Non Wishbone/AMBA APB Signals

The following signals are not part of a standard bus, but are part of the UART core.

I/O		Function
BR_clk	I	The Baud rate generator Clock – used for the baud rate generator, Tx module and the Rx module.
sRX	I	Serial receiver input
CTSn	I	Clear to Send, active low
DSRn	I	Data Set Ready, active low
RIn	I	Ring Indicator, active low
DCDn	I	Data Carrier Detect, active low
sTX	O	Serial Transmit data
DTRn	O	Data Terminal Ready, active low
RTSn	O	Request to Send, active low
OUT1n	O	User-designated output1, active low
OUT2n	O	User-designated output2, active low
TXRDYn	O	Transmit DMA signaling
RXRDYn	O	Receiver DMA signaling
IRQ	O	Interrupt Output
B_CLK	O	16x Baud Rate Clock output

2.4 Address Map

DLAB LCR(7)	Address	R/W	Register Mnemonic	Function
0	0	R	RBR	Receiver FIFO (Receiver Buffer Register)
0	0	W	THR	Transmitter FIFO (Transmitter Holding Register)
0	1	R/W	IER	Interrupt Enable Register
X	2	R	IIR	Interrupt Identification Register
X	2	W	FCR	FIFO Control Register
X	3	R/W	LCR	Line Control Register
X	4	R/W	MCR	Modem Control Register
X	5	R	LSR	Line Status Register
X	6	R	MSR	Modem Status Register
X	7	R/W	SCR	Scratch Register (No UART Control or Status)
1	0	R/W	DLL	Divisor Latch LSB (Baud Rate Generator)
1	1	R/W	DLM	Divisor Latch MSB (Baud Rate Generator)

2.4.1 Interrupt Enable Register (IER)

This register controls which, if any, interrupts are enabled.

Bit	R/W	Bit Description
0	R/W	Enable Received Data Available Interrupt 0 = disabled 1 = enabled
1	R/W	Enable Transmitter Holding Register Empty Interrupt 0 = disabled 1 = enabled
2	R/W	Enable Receiver Line Status Interrupt 0 = disabled 1 = enabled
3	R/W	Enable Modem Status Interrupt 0 = disabled 1 = enabled
7-4	R	0

2.4.2 Interrupt Identification Register (IIR)

Bit	R/W	Bit Description	Interrupt Cleared when:
0	R	Interrupt Pending 0 = Interrupt Pending 1 = No Interrupt Pending	No Interrupt Pending
3-1	R	3 = Receiver Line Status Interrupt Parity, Data overrun, or Framing error, or Break Interrupt	Reading LSR
		2 = Receiver Data Available Receiver FIFO trigger level reached	FIFO continents drops below trigger level
		6 = Timeout Indication After a receiver Data Available interrupt, when no Characters read from the receive FIFO for the time of four characters transfer time, and FIFO is not empty	Reading the Receiver Buffer Register
		1 = Transmit Hold Register Empty	Writing to the Transmitter Holding Register (the Write FIFO) or reading IIR
		0 = Modem Status Interrupt CTS, SDR RI or DCD change state	Reading MSR
		4, 5, 7 – will not occur	
7-4	R	C	

2.4.3 FIFO Control Register (FCR)

Bit	R/W	Bit Description
0	W	Enable FIFO's Don't care (FIFO's always enabled)
1	W	Receive FIFO Reset 0 = nothing 1 = reset FIFO
2	W	Transmit FIFO Reset 0 = nothing 1 = reset FIFO
3	W	DMA Mode Select 0 = Mode 0, 1 = Mode 1
5-4	W	Reserved (ignored)
7-6	W	Receive FIFO trigger level 00 = 1 byte 01 = 4 bytes 10 = 8 bytes 11 = 14 bytes

2.4.4 Line Control Register (LCR)

Bit	R/W	Bit Description
1-0	R/W	Word Length Select 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits
2	R/W	Number of Stop bits 0 = 1 stop bit 1 = 2 stop bits (1.5 stop bits with 5 data bits) Note: the receiver checks for the 1st stop bit only
3	R/W	Parity Enable 0 = no parity 1 = enable parity
4	R/W	Even Parity (used only when parity is enabled) 0 = odd parity 1 = even parity
5	R/W	Stick Parity Ignored
6	R/W	Set Brake 0 = normal operation 1 = The serial output is forced to logic 0 (Spacing State, which will cause a Break interrupt in the receiver)
7	R/W	Divisor Latch (baud rate generator) Access bit This bit must be set to 1 to set the baud rate, it Must be set to 0 to access the FIFO's

2.4.5 Modem Control Register (MCR)

Bit	R/W	Bit Description
0	R/W	DTRn bit is inverted to drive pin
1	R/W	RTSn bit is inverted to drive pin
2	R/W	Out 1n bit is inverted to drive pin
3	R/W	Out 2n bit is inverted to drive pin
4	R/W	Loop Back mode
7-5	R	0 (ignored)

2.4.6 Line Status Register (LSR)

Bit	R	Bit Description
0	R	Data Ready 0 = Receive FIFO is empty 1 = at lest one character is in the receive FIFO
1	R	Overrun Error 0 = no error 1 = Receive FIFO was full, and an additional character was received, but was lost
2	R	Parity Error 0 = no error 1 = top character in FIFO was received with a parity error. If enabled, generates a Receiver Line Status Interrupt
3	R	Framing Error 0 = no error 1 = top character in FIFO was received without a valid stop bit. If enabled, generates a Receiver Line Status Interrupt
4	R	Break Interrupt 0 = No Interrupt 1 = a break condition has been reached (the receive serial input has a logic 0 for a character period). If enabled, generates a Receiver Line Status Interrupt
5	R	Transmitter Holding Register 0 = Transmitter FIFO is not Empty 1 = Transmitter FIFO is Empty If enabled, generates a Transmitter Holding Empty Interrupt
6	R	Transmitter Empty 0 = not 1 1 = Transmitter FIFO and Transmitter Shift Register is Empty.
7	R	Error in receive FIFO 0 = not 1 1 = at least one error (parity, framing or break) in receive FIFO. This bit is cleared upon reading this register.

2.4.7 Modem Status Register (MSR)

Bit	R	Bit Description
0	R	Delta Clear to Send 0 = no change in CTSn input line 1 = CTSn has changed state since last register read
1	R	Delta Data Set Ready 0 = no change in DSRn input line 1 = DSRn has changed state since last register read
2	R	Trailing Edge Ring Indicator 0 = no rising edge in RIn input line 1 = rising edge in RIn input line since last register read
3	R	Delta Data Carrier Detect 0 = no change in DCDn input line 1 = DCDn has changed state since last register read
4	R	Clear to Send Complement of the CTSn input In loop back, this bit is equivalent to RTS in the MCR (bit 1)
5	R	Data Set Ready Complement of the DSRn input In loop back, this bit is equivalent to DTR in the MCR (bit 0)
6	R	Ring Indicator Complement of the RIn input In loop back, this bit is equivalent to Out 1 in the MCR (bit 2)
7	R	Data Carrier Detect Complement of the DCDn input In loop back, this bit is equivalent to Out 2 in the MCR (bit 3)

2.4.8 Scratch Register (SCR)

This byte is readable and writeable. It has no active UART function. Software may use it for temporary storage, or ignore it completely.

2.4.9 Baud Rate Generator (DLL,DLM)

The Baud Rate Generator will divide the BR_clk input by any divisor from 2 to 2^16-1. The output frequency of the Baud Rate Generator is 16X the serial data baud rate. These two registers store the divisor in a 16 bit binary format. When either of these registers are loaded, the 16 bit counter is immediately loaded.

2.5 The UART I/O

I/O		Function
clk	I	Clock – the processor interface is synchronous with this clock
BR_clk	I	The Baud rate generator Clock – asynchronous FIFO's are used to transfer data between clock domains.
rst	I	Asynchronous Reset, active high
CS	I	Chip select, active high – when this signal is high, a data transfer will take place on each clock cycle (The wrappers will limit this to one clock period wide, if a wrapper is not used, it should be high for only one clock period- unless the burst mode is intended)
WR	I	Write – when high with CS, a write cycle When low with CS, a read cycle
A(2 downto 0)	I	Address bus
D(7 downto 0)	I	Input Data Bus
sRX	I	Serial receiver input
CTSn	I	Clear to Send, active low
DSRn	I	Data Set Ready, active low
RIn	I	Ring Indicator, active low
DCDn	I	Data Carrier Detect, active low
sTX	O	Serial Transmit data
DTRn	O	Data Terminal Ready, active low
RTSn	O	Request to Send, active low
OUT1n	O	User-designated output1, active low
OUT2n	O	User-designated output2, active low
TXRDYn	O	Transmit DMA signaling
RXRDYn	O	Receiver DMA signaling
IRQ	O	Interrupt Output
B_CLK	O	16x Baud Rate Clock output
RD(7 downto 0)	O	Read data bus

File name: gh_uart_16550.vhd

DMA Modes affect the operation of the TXRDYn and RXRDYn signals.
Mode 0: Supports single transfer DMA
Mode 1: Supports multiple transfers (once active, signals stay active until the Read FIFO is empty, or the Transmit FIFO is full)

2.6 The Tx Module

I/O		Function
clk	I	Clock, the same clock as used by the baud rate generator
rst	I	Asynchronous Reset, active high
xBRC	I	A clock enable, 16 times the baud rate
D_RDYn	I	Data ready, active low (expected to be tied to the FIFO EMPTY pin)
D(7 downto 0)	I	Input Data
num_bits	I	The number of data bits in a transfer (an integer), expected to be 5, 6, 7 or 8.
Break_CB	I	Forces the output to the Break level (logic low)
stop_B	I	When low, one stop bit is sent, when high, two (or 1.5 when num_bits = 5) stop bits are sent
Parity_EN	I	Parity Enable
Parity_EV	I	When parity is enabled, low means ODD parity is generated, when high, EVEN parity is enabled
sTX	O	Serial Transmit data
BUSYn	O	Active low while busy with Transmitting
read	O	The hand shake companion of D_RDYn (expected to be tied to the FIFO RD pin)

File name: gh_uart_Tx_8bit.vhd

2.7 The Rx Module

I/O		Function
clk	I	Clock, the same clock as used by the baud rate generator
rst	I	Asynchronous Reset, active high
BRCx16	I	A clock enable, 16 times the baud rate
sRX	I	Serial Receive data
num_bits	I	The number of data bits in a transfer (an integer), expected to be 5, 6, 7 or 8.
Parity_EN	I	Parity Enable
Parity_EV	I	When parity is enabled, low means ODD parity is expected, when high, EVEN parity is enabled
Parity_ER	O	Parity check failed
Frame_ER	O	A frame error was detected
Break_ITR	O	Break error, generate a break interrupt
D_RDY	O	Received data ready pulse
D(7 downto 0)	O	Output Data

File name: gh_uart_Rx_8bit.vhd

2.8 The Transmit FIFO

I/O		Function
clk_WR	I	Clock for write port, active on rising edge
clk_RD	I	Clock for read port, active on rising edge
rst	I	Reset, active high – resets counters, flags
srst	I	Sync Reset, active high – resets counters, flags defaults to 0 (synchronous with clk_WR, internally re-synchronized to clk_RD)
WR	I	Write command, synchronous with clk_WR, advances the write counter after write
RD	I	Read command, synchronous with clk_RD, advances read counter to read next word
D(data_width -1 DOWNTO 0)	I	Input data
Q(data_width -1 DOWNTO 0)	O	Output data
empty	O	When low, output data is valid, synchronous with clk_RD
full	O	When low, WR is sampled for write command, synchronous with clk_WR

File name: gh_fifo_async16_sr.vhd

2.9 The Receive FIFO

I/O		Function
clk_WR	I	Clock for write port, active on rising edge
clk_RD	I	Clock for read port, active on rising edge
rst	I	Reset, active high – resets counters, flags
rc_srst	I	Sync Reset, active high – resets counters, flags defaults to 0 (synchronous with clk_RD, internally re-synchronized to clk_WR)
WR	I	Write command, synchronous with clk_WR, advances the write counter after write
RD	I	Read command, synchronous with clk_RD, advances read counter to read next word
D(data_width -1 DOWNTO 0)	I	Input data
Q(data_width -1 DOWNTO 0)	O	Output data
empty	O	When low, output data is valid, synchronous with clk_RD
q_full	O	Quarter Full Flag, synchronous with clk_RD
h_full	O	Half Full Flag, synchronous with clk_RD
a_full	O	almost Full Flag, synchronous with clk_RD (fourteen or more data words in FIFO)
full	O	When low, WR is sampled for write command, synchronous with clk_WR

File name: gh_fifo_async16_rcsr_wf.vhd

3 Using the UART Core

Using the UART core is the similar to using the standard 16550 UART, expect that the FIFO's are always enabled, and there is no sticky parity.

3.1 Initialization

For normal operation, software initialization needs to do the following operations:

- Initialize the LCR register (with bit 7 = 1)
- Initialize the Baud Rate Divisor Registers to set the desired transfer rate
- Reinitialize the LCR register (with bit 7 = 0)
- Set the FIFO trigger level (in the FCR register)
- Enable the desired interrupts

3.2 Baud Rate Divisor calculation

The baud rate divisor value can be calculated with this formula:

Baud rate divisor value = (frequency of BR_clk) / (desired baud rate x 16)

Although the asynchronous protocol is highly immune to small differences in clock frequency, the baud rate divisor should be set as precisely as possible. This way, the accuracy of the desired baud rate is dependent on the oscillator frequency chosen.

The two baud rate divisor registers may be written in either order.

3.3 Transmit Hold Register Empty Interrupt

The Transmit Hold Register Empty Interrupt will be generated two ways:

1. If the transmit FIFO is empty when the interrupt enable bit is changed from disabled to enabled, an interrupt is generated.
2. Once the interrupt is enabled, when the transmit FIFO becomes empty, an interrupt is generated.

Once an interrupt is cleared, one of the two above conditions will need to take place before the next THRI interrupt is generated.

4 Core Notes

It could be argued the modules in the core should have a prefix of hl_. But, since it has a close relationship with the GH VHDL Standard Parts Library, it seems reasonable to use the gh_ prefix, and save the ego trip for better use elsewhere.