Lab 8
UART Receiver

Name___________________________ Class __________ Date ___________________

OBJECTIVES:

Upon completion of this laboratory exercise, you should be able to:

- Create and simulate a shift register that can be used as a receiver in a universal asynchronous receiver/transmitter (UART).
- Create and simulate an 8-bit flip-flop that can be used as a data buffer in a UART receiver.
- Combine the shift register and flip-flop with the counters developed in Lab 7 and other components to create a UART receiver.
- Simulate the operation of the UART receiver.
- Develop a test circuit for the UART receiver that will accept and display an ASCII character from a computer running HyperTerminal. The character will be transmitted at a fixed rate of 9600 baud.

EQUIPMENT REQUIRED:

Altera DE1/DE2 Circuit Board with USB Blaster Download Cable
Quartus II Software
Anti-static wrist strap

REFERENCE:

Application Note:

- Simple UART Receiver for Quartus II Implementation (Class handout; also available on course Sharepoint site.)

PROCEDURE:

Receiver Data Shift Register

1. Create a 10-bit shift register in VHDL, as shown in Figure 1. The shift register should have the following features:

- serial input
- parallel output
- active-HIGH input to enable shift function

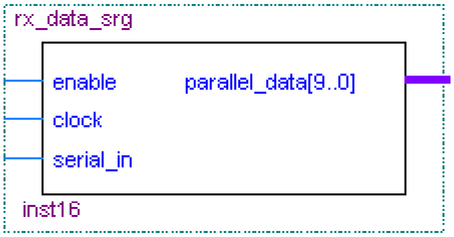


Figure 1 Receiver Shift Register

2. Create a simulation for the shift register that tests all its functions.

Instructor's Initials __________

Receiver Data Buffer

1. Create an 8-bit register (i.e. a parallel flip-flop) in VHDL, as shown in Figure 2, with the following features:

- The register will synchronously transfer the bits at **parallel_data[7..0]** to **rx_buffer[7..0]** if a positive edge is applied to **clock** and **load_rx_buffer** is HIGH.
- When the data is transferred from **parallel_data[7..0]** to **rx_buffer[7..0]**, the same clock edge sets **end_character** to '0'.
- When **load_rx_buffer** goes LOW, **end_character** goes HIGH on the next clock edge.

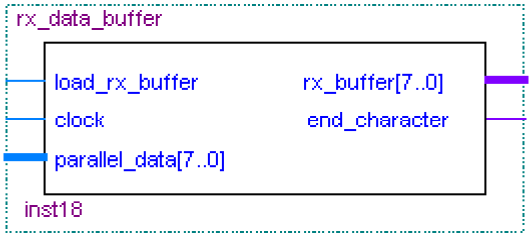


Figure 2 Receiver Data Buffer

2. Create a simulation to test the receiver data buffer.

Lab 8
UART Receiver

UART Receiver

1. Use the Quartus II Block Editor to combine the bit counter and sample clock generator from Lab 6 with the receiver shift register and receiver data buffer created earlier in this lab, as shown in the diagram **UART_rx**. This diagram is available as a separate 11 × 17 sheet or as a **pdf** document on the course Sharepoint site.

2. Create a simulation similar to the one shown in Figure 12 of the application note for the UART receiver. The only two inputs to define are **rx_data** and **rx_clock**. The simulation should have the following characteristics:

- **rx_data** must be set up to approximately emulate an incoming bit rate of 9600 baud. (Calculate the duration of each bit at that rate.)
- **rx_clock** must run at 16 × the incoming bit rate.
- the simulation must show at least two incoming ASCII characters.

Show the simulation to your instructor.

50% Bonus: Create the receiver using only VHDL components and top-level, with no use of the Quartus II Graphic Editor.

Instructor's Initials _________

UART Receiver Data Test Circuit

1. Create a test circuit for your UART receiver, as shown in Figure 3. The clock divider should consist of an LPM counter with a modulus chosen to divide the 50 MHz clock from the Altera DE1/DE2 board to produce an output that is 16 × 9600 Hz. (Refer to Lab 8 for the structure of the clock divider.) Assign pins to the circuit as shown on the table on the second last page of this lab.

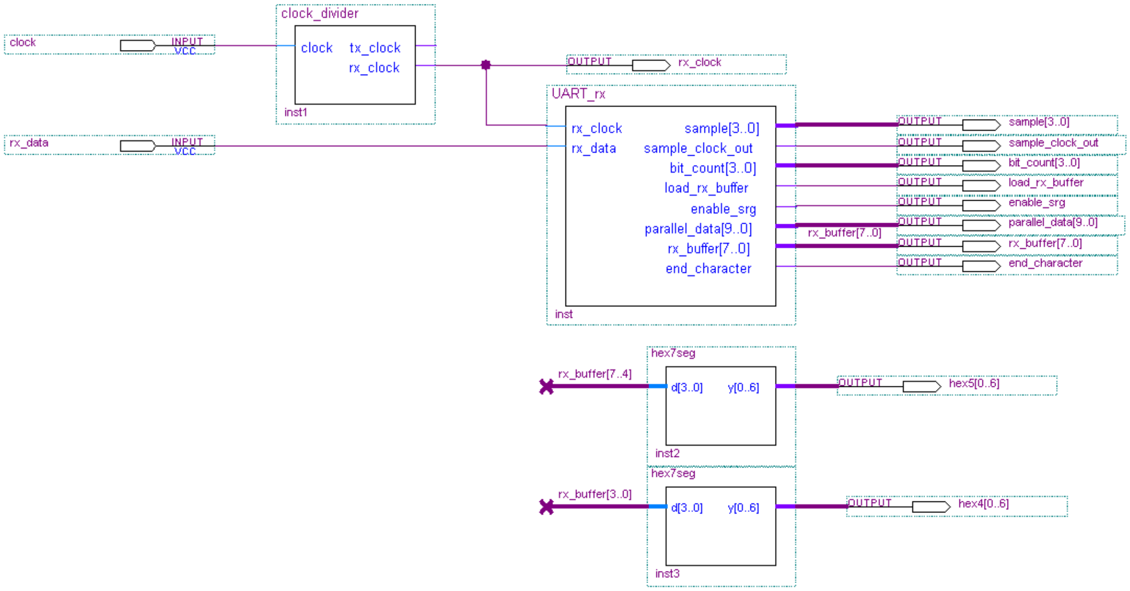
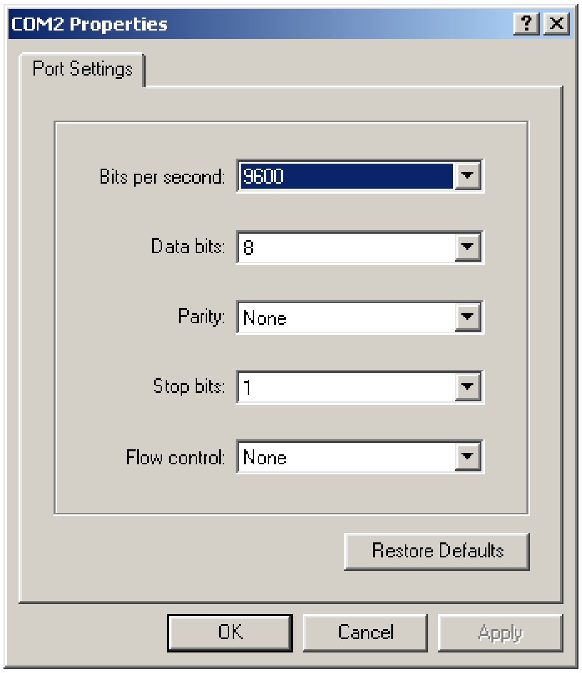


Figure 3 UART Rx Test Circuit

Testing the Receiver using HyperTerminal

1. Connect one of the serial ports of your PC to the 9-pin D connector on the Altera DE1/DE2 board.

2. Open HyperTerminal in Windows and create a session with the settings shown in Figure 5. Make sure that the COM port selected in HyperTerminal (probably COM1 or COM2) is the one you connected to the DE2.



Bits per second: 9600
Data bits: 8
Parity: None
Stop bits: 1
Flow control: None

Figure 4 HyperTerminal Settings

5. Type a character on the keyboard. Compare the values on LEDs 1 to 8 and the two-digit hexadecimal number on the DE1/DE2 board display with values in the ASCII table. Make sure that the key you typed corresponds to the received data. Demonstrate the operation of the board to your instructor.

Instructor's Initials _________

DE1/DE2 Pin Assignments
Device: Cyclone II, EP2C35F672C6

Signal	Pin	Device
clock		50 MHz clock (internal)
rx_data		On-board RS-232 Rx
rx_buffer[7]		LEDR[7]
rx_buffer[6]		LEDR[6]
rx_buffer[5]		LEDR[5]
rx_buffer[4]		LEDR[4]
rx_buffer[3]		LEDR[3]
rx_buffer[2]		LEDR[2]
rx_buffer[1]		LEDR[1]
rx_buffer[0]		LEDR[0]
ao1		HEX4[0]
bo1		HEX4[1]
co1		HEX4[2]
do1		HEX4[3]
eo1		HEX4[4]
fo1		HEX4[5]
go1		HEX4[6]
ao2		HEX5[0]
bo2		HEX5[1]
co2		HEX5[2]
do2		HEX5[3]
eo2		HEX5[4]
fo2		HEX5[5]
go2		HEX5[6]

ASCII Table

	MSBs							
	000	001	010	011	100	101	110	111
LSBs	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
0000 (0)	NUL	DLE	SP	0	@	P	’	p
0001 (1)	SOH	DC1	!	1	A	Q	a	q
0010 (2)	STX	DC2	″	2	B	R	b	r
0011 (3)	ETX	DC3	#	3	C	S	c	s
0100 (4)	EOT	DC4	$	4	D	T	d	t
0101 (5)	ENQ	NAK	%	5	E	U	e	u
0110 (6)	ACK	SYN	&	6	F	V	f	v
0111 (7)	BEL	ETB	′	7	G	W	g	w
1000 (8)	BS	CAN	(	8	H	X	h	x
1001 (9)	HT	EM	)	9	I	Y	i	y
1010 (A)	LF	SUB	*	:	J	Z	j	z
1011 (B)	VT	ESC	+	;	K	[	k	{
1100 (C)	FF	FS	,	<	L	\	l	|
1101 (D)	CR	GS	-	=	M	]	m	}
1110 (E)	SO	RS	.	>	N	^	n	~
1111 (F)	SI	US	/	?	O	_	o	DEL