Lab 9
State Machines and UART Integration

Name_____ Class _____ Date _____

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

- Create and simulate a state machine that will generate a single pulse from a pushbutton input.
- Integrate the single-pulse state machine with the UART transmitter from Lab 8 to make a UART that can transmit a single character when a pushbutton input is pressed.

EQUIPMENT REQUIRED:

Altera DE1/DE2 Circuit Board with USB Blaster Download Cable
Quartus II Software
AC Adapter, minimum output: 7 VDC, 250 mA DC
Anti-static wrist strap

REFERENCE:

Lab 7: UART Transmitter
Lab 8: UART Receiver
11 × 17 Diagram: UART_Block_Diagram_2009.pdf

Lab 9
State Machines and UART Integration

PROCEDURE:

State Machine (Single Pulser)

1. Draw a state diagram for a state machine that acts as a single-pulser. That is, the machine will behave as follows:

- The machine has an input called **pulse_in** and an output called **pulse_out** and two states, **s0** and **s1**.
- When the machine is in **s0** and **pulse_in** is HIGH, the machine remains in **s0**, with **pulse_out** = 0.
- When the machine is in **s0** and **pulse_in** is LOW, the machine makes a transition to **s1** and **pulse_out** goes HIGH.
- When the machine is in **s1** and **pulse_in** is LOW, the machine remains in **s1**, with **pulse_out** = 0.
- When the machine is in **s1** and **pulse_in** is HIGH, the machine goes to **s0**, with **pulse_out** = 0.

2. Use Quartus II to create the state machine described by the state diagram drawn in step 1. Create a simulation that shows the operation of the circuit, including any potential response to asynchronous input behavior.

Show the simulation to your instructor.

Instructor's Initials _____

Lab 9
State Machines and UART Integration

Single pulser in an integrated UART

1. Create a Quartus II Block Diagram File like the one shown in the 11×17 sheet called **UART_Block_Block_Diagram_2009.pdf**, that combines the UART Transmitter of Lab 7, the UART Receiver of Lab 8, and the single pulser state machine created in this lab. The idea is to be able to have a full duplex transmitter and receiver in which the transmitter sends a single character when a pushbutton switch on the DE2 board is pressed.

In addition to the UART Tx, Rx, and single pulser, also include:

- a clock divider that divides a 50 MHz input to produce a 153.6 kHz output for the receiver clock and a 9600 Hz output for the transmitter clock;
- a pair of hex-to-seven-segment decoders for the received data.

2. Assign pin numbers as shown in the table on the following page. Compile the project and download it to your board. Connect the board to the serial port of your PC and start a HyperTerminal session. Use a baud rate of 9600, with settings 8N1, no hardware control.

3. Demonstrate the following functions to your instructor:

- Receive a character from the PC.
- Send the message: "Hi there!<LF><CR>" to the PC. (Do not include the quotation marks. <LF> stands for the ASCII code for Line Feed. <CR> stands for the ASCII code for Carriage Return.)

Instructor's Initials _____

Lab 9
State Machines and UART Integration

Pin Name	Pin Number	Device	
hex4[0]		HEX4[0]	Rx LSD
hex4[1]			
hex4[2]			
hex4[3]			
hex4[4]			
hex4[5]			
hex4[6]		HEX4[6]	
hex5[0]		HEX5[0]	Rx MSD
hex5[1]			
hex5[2]			
hex5[3]			
hex5[4]			
hex5[5]			
hex5[6]		HEX5[6]	
hex6[0]		HEX6[0]	Tx LSD
hex6[1]			
hex6[2]			
hex6[3]			
hex6[4]			
hex6[5]			
hex6[6]		HEX6[6]	
hex7[0]		HEX7[0]	Tx MSD
hex7[1]			
hex7[2]			
hex7[3]			
hex7[4]			
hex7[5]			
hex7[6]		HEX7[6]	

Pin Name	Pin Number	Device
parallel_in[7]		SW[7]
parallel_in[6]		SW[6]
parallel_in[5]		SW[5]
parallel_in[4]		SW[4]
parallel_in[3]		SW[3]
parallel_in[2]		SW[2]
parallel_in[1]		SW[1]
parallel_in[0]		SW[0]
parallel_in_leds[7]		LEDR[7]
parallel_in_leds[6]		LEDR[6]
parallel_in_leds[5]		LEDR[5]
parallel_in_leds[4]		LEDR[4]
parallel_in_leds[3]		LEDR[3]
parallel_in_leds[2]		LEDR[2]
parallel_in_leds[1]		LEDR[1]
parallel_in_leds[0]		LEDR[0]
rx_data		RS-232 Rx
tx_out		RS-232 Tx
send		KEY[3]
set		KEY[0]
sys_clock		50 MHz Clock

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