Digital Systems **Lab 6**

Decoded Counters

NAME_____

OBJECTIVES:

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

- Write VHDL descriptions of two synchronous counters with an enable input, a defined modulus, and a decoded output.
- Simulate each counter design to verify the correctness of the design entry.
- Test each counter on an FPGA board.
- Combine the two counters in a higher-level VHDL file and simulate the combination of counters to verify correct design.

EQUIPMENT REQUIRED:

FPGA Trainer: Altera DE1 or DE2 Board with USB Programming Cable and AC Adapter
Quartus II Software
Anti-static wrist strap

EXPERIMENTAL NOTES:

The two VHDL-based counters created in this lab exercise will be used as components in a larger design – a Universal Asynchronous Receiver/Transmitter (UART) – in Lab 9 of this course.

The purpose of the first counter we will create (**bit_counter.vhd**) is to count incoming data bits in a UART receiver. Its operation is shown in the simulation in Figure 1, where the counter counts to 10 (A in hexadecimal) and recycles, with two decoded outputs at the last stage.

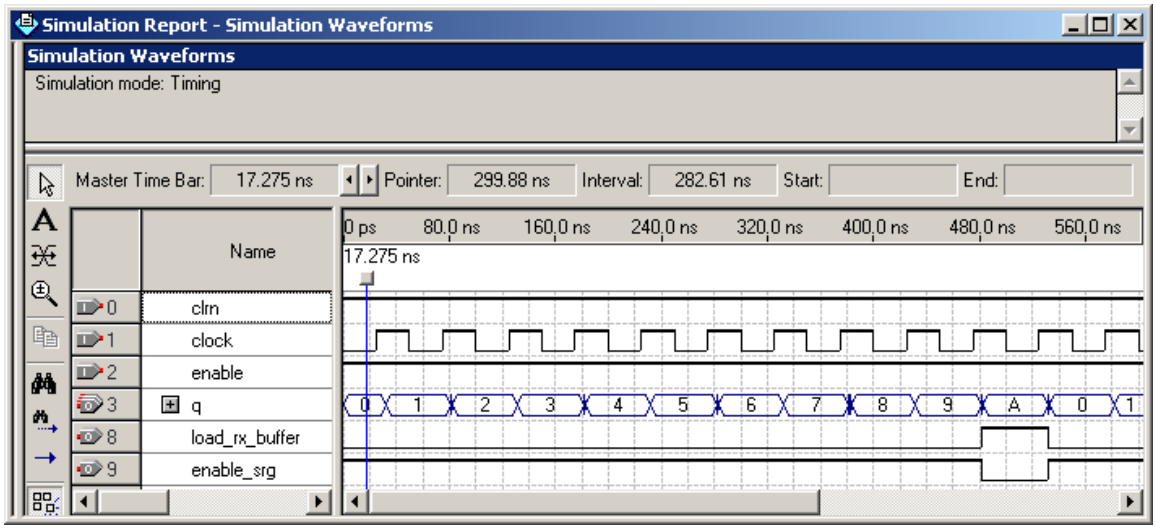

Figure 1 Simulation for a bit counter

The second counter (**sample_clock.vhd**) is used to generate a sample clock for a UART receiver, which clocks in the incoming data bits in the middle of the bit time. It does so by dividing the incoming receiver clock signal by 16 and using a binary decoder to produce a sample clock pulse when the count value is halfway through its sequence at hexadecimal 8. The operation of this counter is shown in Figure 2. The two glitches following states 9 and B can be ignored.

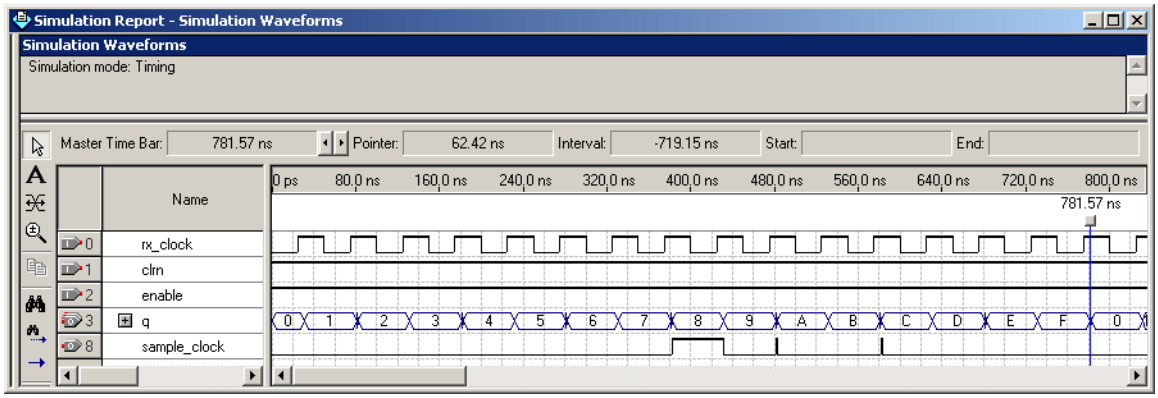

Figure 2 Simulation for a sample clock counter

The counters are later combined to make a counter system. One counter generates a sample clock which enables the bit counter whenever it goes HIGH. Therefore, the bit counter advances by one when the sample clock is HIGH and a positive edge is applied from **rx_clock**. Figure 3 shows the combined operation of the counters.

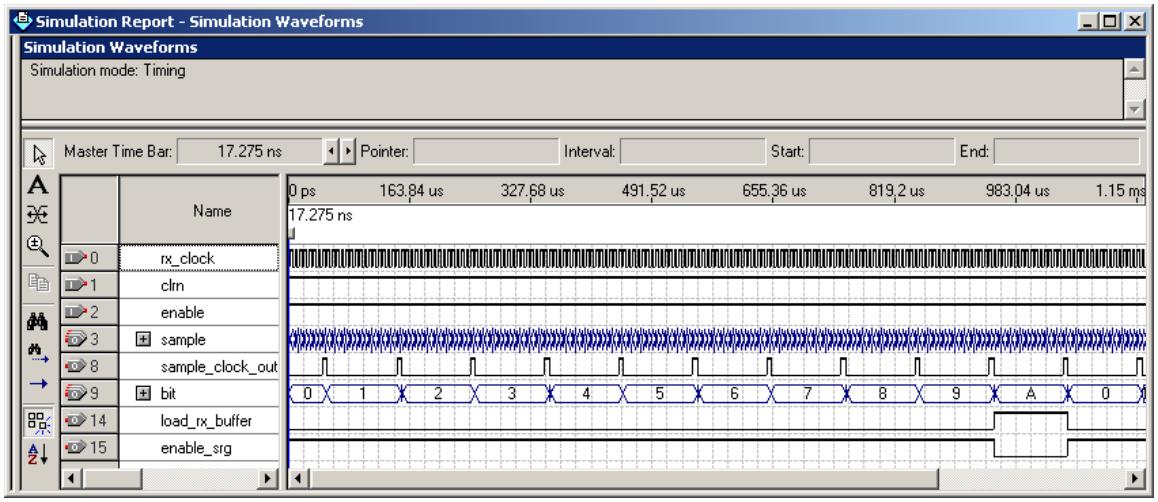

Figure 3 Simulation for combined counters

Figures 4 and 5 show greater detail of the combined counters' simulation. In Figure 4, we see that the sample clock goes HIGH when the sample clock counter has a value of 8. Notice that the bit counter advances at the *end* of this period, on the first positive edge of the clock *after* the sample clock goes HIGH.

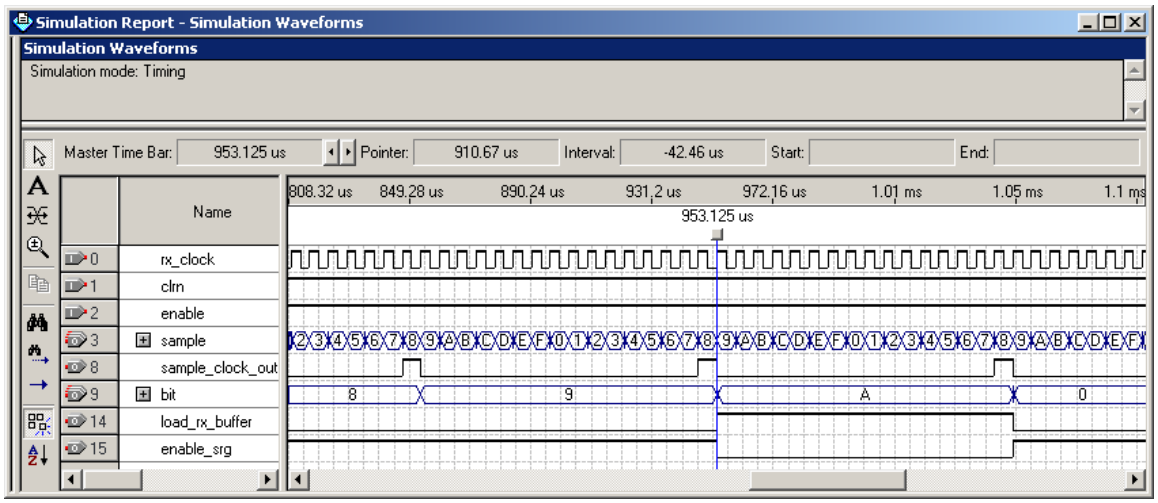

Figure 4 Combined counter simulation showing sample clock counter detail

Figure 5 shows the combined counters' simulation at high magnification. Note that the sample clock output remains HIGH long enough for it to overlap the positive edge **rx_clock**, thus enabling the bit counter to advance. The bit counter is not enabled again until the next time the sample clock output goes HIGH.

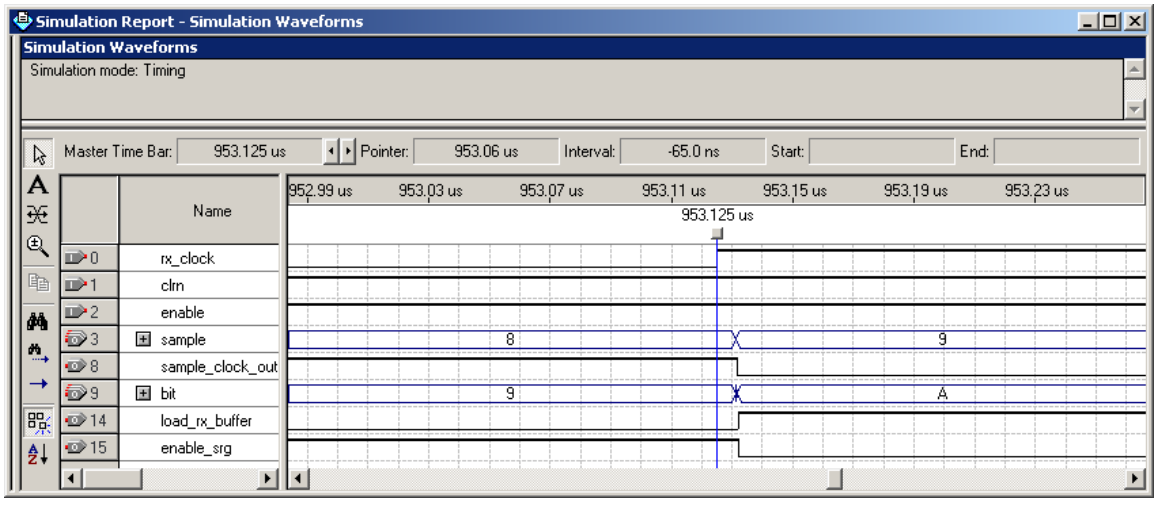

Figure 5 Combined counter simulation showing enabling of the bit counter by the sample clock

PROCEDURE:

COUNTER 1 (BIT COUNTER FOR A UART RECEIVER)

1. Create a counter like the one shown in Figure 6, using VHDL design entry. ***Do not use the Quartus II Block Diagram Editor.***

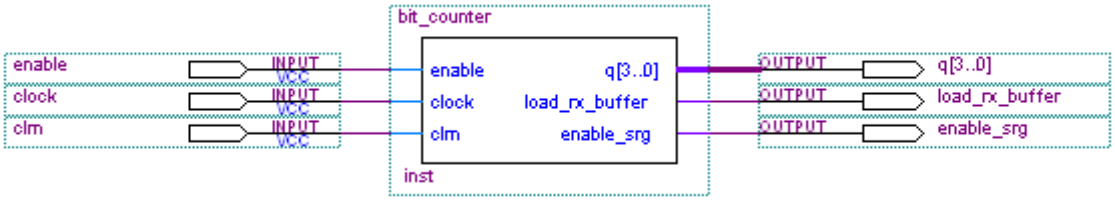

Figure 6 Decoded Counter (bit counter)

Bit counter features:

- Clock is applied to an input called **clock**.
- The counter clears asynchronously when **clrn** is LOW.
- Counter output **q[3..0]** proceeds from 0000 to 1010, one state per clock pulse, when **enable** is HIGH and **clrn** is HIGH.
- The counter returns to 0000 one clock pulse after it reaches 1010.
- Outputs called **load_rx_buffer** and **enable_srg** are decoded outputs. **Enable_srg** goes LOW when **q[3..0]** = 1010. **Load_rx_buffer** is the opposite of **enable_srg.**

2. Create a VHDL testbench for the counter, testing all functions. Run the testbench in ModelSim.

Instructor's Initials _____

3. Assign pins as follows. Recompile the design and program it into your DE1/DE2 board. Test the effect of the **clock**, **enable**, and **clrn** inputs on the output LEDs.

Signal	Pin Number	Device
clrn		KEY[0]
clock		KEY[3]
enable		SW[0]
q[0]		LEDG[0]
q[1]		LEDG[1]
q[2]		LEDG[2]
q[3]		LEDG[3]
enable_srg		LEDG[7]
load_rx_buffer		LEDR[0]

Instructor's Initials _____

COUNTER 2 (SAMPLE CLOCK GENERATOR FOR A UART RECEIVER)

1. Create a counter like the one shown in 7, using VHDL design entry. ***Do not use the Quartus II Block Diagram Editor***.

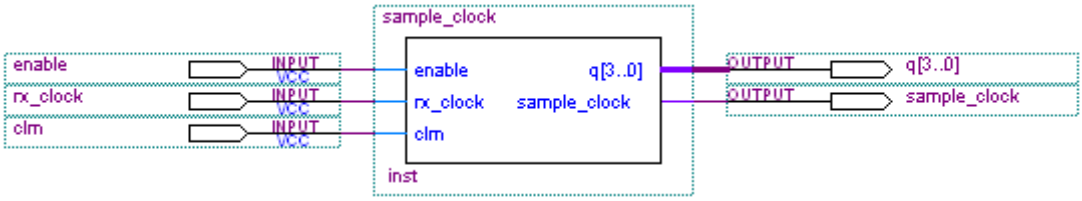

Figure 7 Decoded counter (sample clock generator)

Sample clock counter features:

- Clock is applied to an input called **rx_clock**.
- The counter clears asynchronously when **clrn** is LOW.
- Counter output **q[3..0]** proceeds from 0000 to 1111, one state per clock pulse, when **enable** is HIGH and **clrn** is HIGH.
- Output called **sample_clock** is a decoded output. It goes HIGH when **q[3..0]** = 1000.

2. Create a VHDL testbench for the counter, testing all functions. Run the testbench in ModelSim.

Instructor's Initials _____

3. Assign pins as follows. Recompile the design and program it into your DE1/DE2 board.

Signal	Pin Number	Device
clrn		KEY[0]
rx_clock		KEY[3]
enable		SW[0]
q[0]		LEDG[0]
q[1]		LEDG[1]
q[2]		LEDG[2]
q[3]		LEDG[3]
sample_clock		LEDR[0]

Instructor's Initials _____

COMBINED COUNTERS (VHDL ONLY)

Create a VHDL File in the Quartus II Text Editor called **combined_counters.vhd**. ***This design must be done entirely in VHDL. Do not use the Quartus II Block Editor.*** The diagram in Figure 8 shows a graphical representation of the design for reference only.

Use the VHDL files **bit_counter** and **sample_clock** as components in the **combined_counters** file. Connect the sample clock generator and the bit counter, as shown in Figure 8. Note that both counters are clocked by the same clock line and the sample clock is used to enable the bit counter for one clock cycle out of 16.

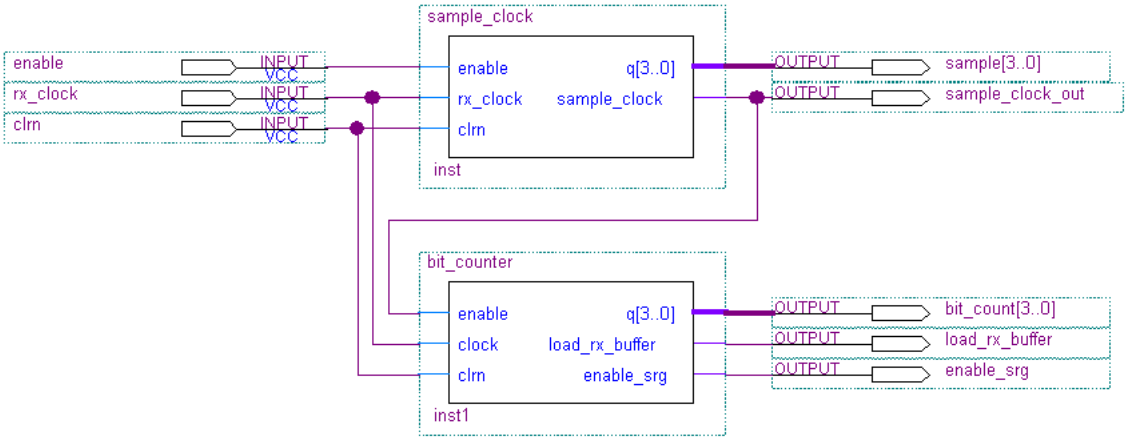

Figure 8 Combined Counters

Create a VHDL testbench for the combined counters, testing all functions. Set the the end time of the simulation to **1.5 ms**. Set the Clock Period (**rx_clock**) to **6.25 us**. Run the testbench in ModelSim.

Show the simulation to your instructor and explain what it shows about the circuit operation.

Instructor's Initials _____

SAVE THE VHDL FILES FROM THIS LAB EXERCISE IN A PLACE WHERE YOU CAN LOCATE THEM LATER. THEY WILL BE USED AS PART OF A DESIGN IN A FUTURE LAB.