CPET-233 Digital Systems Design
Fall 2019

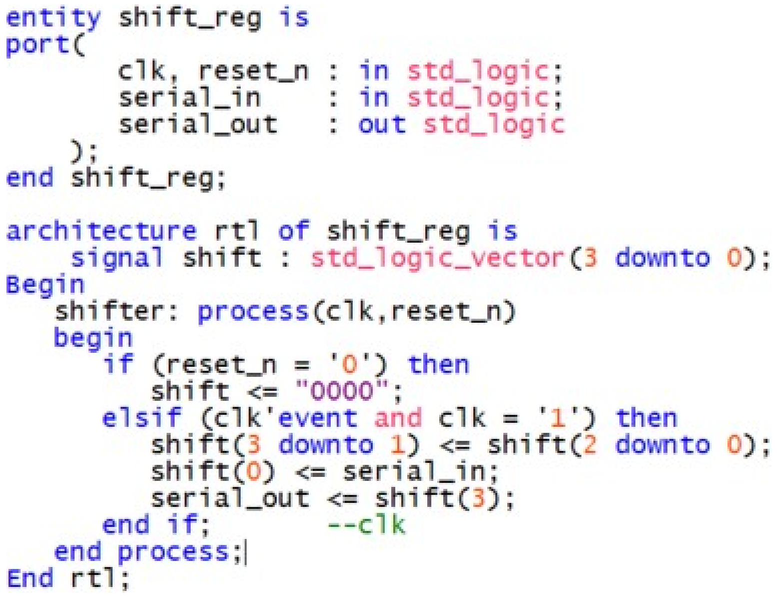
Homework #9 – Due 11/06/19
Please submit a hard copy or put in the Dropbox in MyCourses

1. Write the VHDL for a 4-bit up/down counter that does not overflow or underflow. When the count reaches its maximum value it remains at that value if the count mode is up. When the count reaches zero, it remains zero if the count mode is down. Use a control input named U_D to control the direction of the count (1 means count up and 0 means count down). Remember all counters have a clock and reset as well. Submit your code.

2. Use the Quartus waveform editor to simulate your counter. Your simulation should show the following:
   a. Counting up
   b. Holding at its max value
   c. Counting down
   d. Holding at its min value

   Submit your output waveform.

3. Create a project and enter the code below:



```
entity shift_reg is
port(
       clk, reset_n : in std_logic;
       serial_in    : in std_logic;
       serial_out   : out std_logic
    );
end shift_reg;

architecture rtl of shift_reg is
    signal shift : std_logic_vector(3 downto 0);
Begin
   shifter: process(clk,reset_n)
   begin
      if (reset_n = '0') then
         shift <= "0000";
      elsif (clk'event and clk = '1') then
         shift(3 downto 1) <= shift(2 downto 0);
         shift(0) <= serial_in;
         serial_out <= shift(3);
      end if;       --clk
   end process;
End rtl;
```

Compile the code then choose Tools > Netlist Viewers > RTL Viewer. Take a screen shot. Now move the serial_out assignment outside of the process. Recompile and look at the RTL. Take a screen shot. What is the difference between the two implementations? What general conclusion can you make about assignments made within a clock statement? Submit both screen shots and your answers.

4. Write the VHDL code for a 16-bit universal shift register. This shift register has a SH/$\overline{LD}$ input. When SH/$\overline{LD}$ is 1, the shift register will shift. When it is 0 the shift register will perform a parallel load of the 16-bit data input called Din. There is also an R/$\overline{L}$ input that determines the direction of shift. When it is a 1, the shift is right (msb to lsb). When it is a 0 the shift is left (lsb to msb). Don't forget to include a serial input and serial output.

5. The schematic below is a pseudo-random number generator. It has 4 inputs as follows:

   Seed : an 8-bit number that seeds the random number generator when load is 1.

   Load : when load = 1, the number generator is seeded. When load = 0, it shifts on each clock cycle. The load is synchronous.

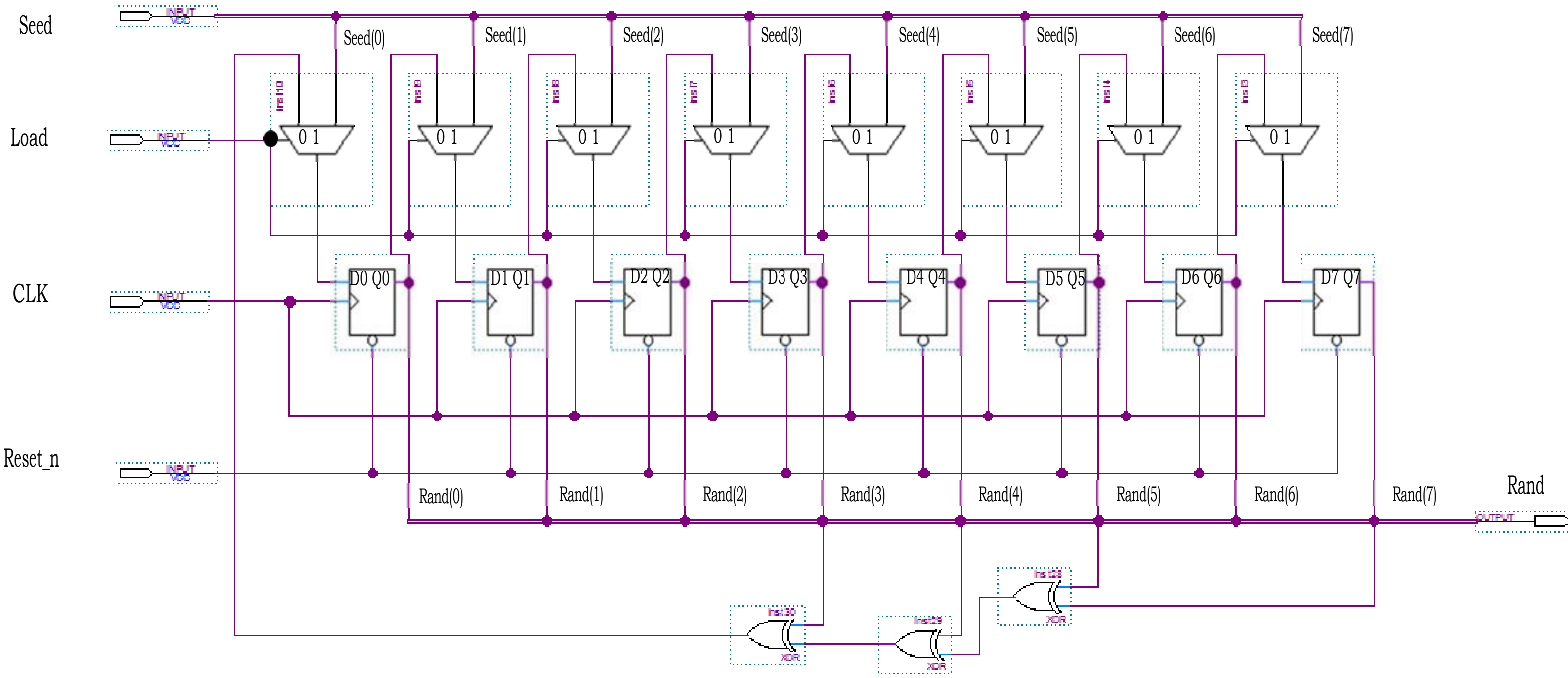
   CLK : A standard clk input

   Reset_n : An asynchronous reset input, active low

The random number generator only has one output

Rand : An 8-bit number.

Write the complete VHDL (entity and architecture) for the circuit. Note that this is a shift register. You can use a behavioural architecture for a shift register, you do not have to use components.


Seed
Load
CLK
Reset_n
Seed(0)
Seed(1)
Seed(2)
Seed(3)
Seed(4)
Seed(5)
Seed(6)
Seed(7)
0 1
D0 Q0
D1 Q1
D2 Q2
D3 Q3
D4 Q4
D5 Q5
D6 Q6
D7 Q7
Rand(0)
Rand(1)
Rand(2)
Rand(3)
Rand(4)
Rand(5)
Rand(6)
Rand(7)
Rand
INPUT
VCC
OUTPUT
XOR