BME 328 Lab 5 – VHDL for Combinatorial Circuits and Storage Elements

15 Marks (1 week)
Due Date: Week 9

1. Objectives

- To construct combinational circuits and circuits with basic storage elements using VHDL.

2. Laboratory Work

Part 1

1. Start-up Quartus II. This window gives you access to an integrated suite of CAD tools.
2. Create the subdirectory *lab4* in your work directory.
3. To save files for this lab, create subdirectories **mux**, **decode, encod,** and **johns** in your *lab4* subdirectory.
4. Create a new project **mux** in its corresponding subdirectory. Remember to choose the "EP2C35F672C6" device in the Project Wizard.
5. Open the Text Editor and type the VHDL file from *Figure 6.28* of the textbook. Save the file as **mux.vhd** in its corresponding subdirectory.
6. Start the compiler. Fix any errors and re-compile. Once the file compiles without errors, go to the next step.
7. Simulate your design and verify that it is functioning properly.
8. Take, and save, screenshots, of all your ".vhd" files and your simulation waveforms.
9. Create a symbol for your design.
10. Repeat steps 4-9 for the remaining examples. Use files from the following figures accordingly (see textbook):
    i. **decod –** Figure 6.30
    ii. **encod –** Figure 6.41
    iii. **johns –** Figure 2 (In this Manual).
11. The last example shows one of the ways of implementing the Johnson counter. The last six digits of the student identification number must be represented by a four-bit vector variable **STUDENT_ID** which will be displayed cyclically in sequence with Johnson counter output. **Qreg** is an internal signal which can be fed back to the D's or fed out to **Q**. The circuit design must handle non-valid states and non-valid student identifier cases by outputting an "E". Consider the last 6 digits of the student identifier D = {d_1, d_2, d_3, d_4, d_5, d_6} in its general representation. Then, as an example, a student with identifier: **500435429** will follow the display sequence as **435429**.
12. Prepare a Truth Table for Johnson Counter for 6 clock cycles.

Part 2

1. Start-up Quartus II. This window gives you access to an integrated suite of CAD tools.
2. Create subdirectories **muxModified** and **decodModified** in your *lab4* subdirectory.
3. Create a new project **muxModified** in its corresponding subdirectory. Remember to choose the "EP2C35F672C6" device in the Project Wizard.
4. Create a block schematic file **muxModified.bdf** for the project defined in (3) and implement a **4:1** multiplexer using two **2:1** multiplexer (**mux** symbols) as shown in *Figure 6.3* of the text book.
5. Start the compiler. Fix any errors and re-compile. Once the file compiles without errors, go to the next step.

6. Simulate your design and verify that it is functioning properly.
7. Take, and save, screenshots of all your ".bdf" files and your simulation waveforms.
8. Repeat steps 3-7 for **decodModified**.

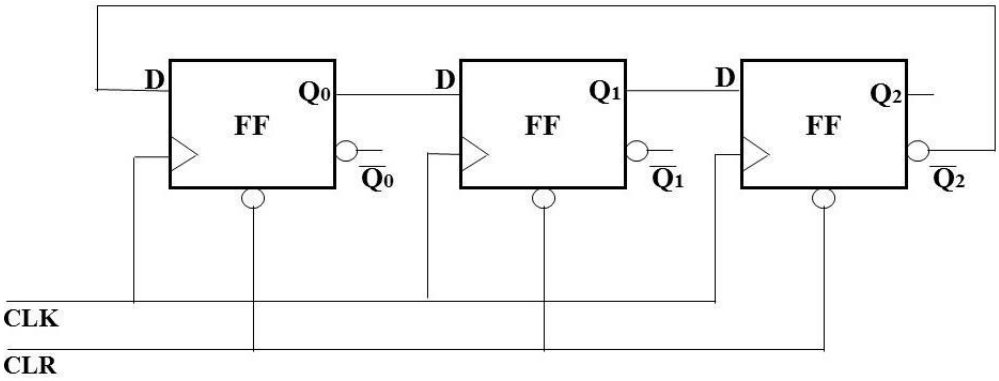

Figure 1: Johnson Counter (Note feedback Connection)

```
LIBRARY ieee;
USE ieee .std_logic_1164.all;
ENTITY johns IS

PORT (Clrn, E, Clkn : IN STD_LOGIC; --clrn is your reset button
 STUDENT_ID : out   std_logic_vector(3 downto 0);
 Q : OUT STD_LOGIC_VECTOR (0 TO 2));
END johns;

ARCHITECTURE Behavior OF johns IS
signal Qreg : STD_LOGIC_VECTOR (0 TO 2);
BEGIN
    PROCESS (Clrn, Clkn)
     BEGIN
         IF Clrn = '0' THEN
            Qreg <= "000";
         ELSIF (Clkn'EVENT AND Clkn = '0') THEN
             IF E = '1' THEN
                 .. -- complete your johns flip-flop outputs here..
                 ..
                 Qreg(1) <= Qreg(0);
                 ..
             ELSE ..
                Qreg <= Qreg;
             END IF;
         END IF;

    -- STUDENT_ID variable represents the last 6 digits of your student ID
    hence d4 is the fourth digit of your
    --student ID in four bits, d5 is the fifth and so on. For example, for
    Student ID 500435429,
    --d4 is 0100, d5 is 0011 and so on
         CASE  Qreg  IS
             WHEN   "000" =>
                 STUDENT ID <= ..... --d1
             WHEN   "100" =>
                 STUDENT ID <= ..... --d2
             .                       --d3
             .                       --d4
             .                       --d5
                                     --d6
             WHEN OTHERS => STUDENT_ID <= "----";--error
         END CASE;
    END PROCESS;
Q <= Qreg;

END Behavior;
```

Figure 2