Toronto Metropolitan University

Department of Electrical, Computer and Biomedical Engineering

COE 328 – Digital Systems

Lab 4 - VHDL for Combinational Circuits and Storage Elements (2 weeks)

---

1 Objectives:

To design/implement combinational circuits and circuits with basic storage elements using VHDL.

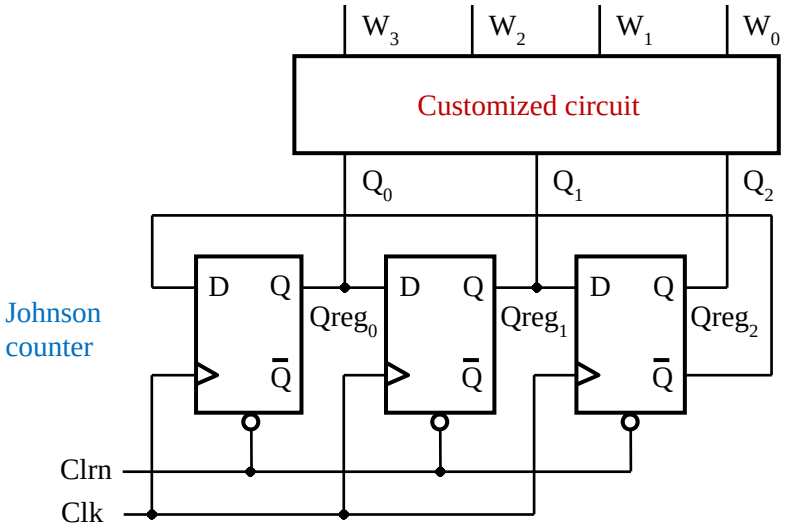
2 Pre-Lab Preparation

1. Start-up Quartus II - an integrated suite of CAD tools.
2. To save files for this lab, create subdirectories **mux**, **decode**, **encod**, and **johns** in your work directory.
3. Enter the name of the first project, **mux**, by clicking on File/Project on the pull-down menu, and then Name on the subsequent pull-down menu. Type the Project Name, and click OK.
4. Open Text Editor and type the VHDL file from Fig. 6.28 of the textbook. Save the file as **mux.vhd**.
5. Start the compiler. Fix any errors and re-compile. Once the file compiles without errors, go to the next step.
6. Repeat steps 3-5 for the remaining examples. Use files from the following figures accordingly (see the textbook):
   - **decod** (Fig. 6.30)
   - **encod** (Fig. 6.41)
   - **johns** (Fig. 1 below)
7. The VHDL code in Fig. 1 demonstrates one of the ways of implementing the Johnson counter. The outputs Q_0, Q_1, Q_2 of the Johson counter are connected to the inputs of a customized combinational circuit. The function of this combinational circuit is defined by the student ID number D = {d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9} as follows. When the input Q of the customized circuit goes through all possible combinations (i.e., 000, 100, 110, 111, 011, 001), its output W sequentially goes through the last 6 digits of the student ID number (i.e., d_4, d_5, d_6, d_7, d_8, d_9). Qreg is an internal signal which can be fed back to the D's or fed out to Q.

3 Laboratory Work

1. Create the subdirectory *lab4* in your work directory, and copy the all the subdirectories created as part of the pre-lab to this subdirectory.
2. Compile your designs, create symbols for respective projects (**mux**, **decode**, **encod**, and **johns**) and save them.
3. Create new subdirectories inside *lab4* folder of your working directory with names **muxModified** and **decodModified**.
4. Start-up Quartus II - an integrated suite of CAD tools.
5. Enter the name of the project, **muxModified**, by clicking on File/Project on the pull-down menu and then Name on the subsequent pull-down menu. Type the Project Name and click OK.

6. Create a block schematic file **muxModified.bdf** for the project defined in (5) and implement a **4-to-1** multiplexer using two **2-to-1** multiplexers (**mux** symbols) as shown in Fig. 6.3 of the text book.
7. Repeat the steps (5) and (6) for the project **decodModified** and implement a **3-to-8** decoder using two **2-to-4** decoders (**decode** symbols) as outlined in Fig. 6.17 of the text book.
8. Assign all Input/Output signals to any dedicated Input/Output pins of the Cyclone-II FPGA on the prototype board (see Pin Assignment Table in Lab3). Re-compile your design.
   NOTE:
   - All the output LEDs are active **HIGH**. (i.e., a logic '1' will turn the LED's on).
   - All the 7-segment displays are active LOW (i.e., a logic '0' will turn the respective segment on).
   - The **resetn** signal must be assigned to the pushbutton[0] switch on the prototype board (PIN_G26).
   - The **clk** signal must be assigned to the to the pushbutton[3] switch on the prototype board (PIN_W26).
9. Implement/program all your designs into the Cyclone II FPGA.
10. Every digit of last 6 digits of the Student ID (signal W) should be displayed on the 7-segment display, while the current state of the Johnson counter (signal Q) is displayed on green LED's.
11. Design your circuits as outlined and demonstrate results to the instructor by displaying both the state and student ID signals.
    NOTE: Re-use the 7-segment module from Lab 3 to display student ID digits.




```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY johns IS
        PORT  (Clrn, Clk        :       IN      STD_LOGIC;
               W                :       OUT     STD_LOGIC_VECTOR (3 DOWNTO 0);
               Q                :       OUT     STD_LOGIC_VECTOR (0 TO 2));
END johns;

ARCHITECTURE Behaivor OF johns IS
        signal  Qreg            :       STD_LOGIC_VECTOR (0 TO 2);
BEGIN
        PROCESS (Clrn, Clk)
        BEGIN
                IF Clrn = '0' THEN
                        Qreg <= "000";
                ELSIF (Clk'EVENT AND Clk = '1') THEN
                        …               -- complete the code
                        Qreg(1) <= Qreg(0);
                        …               -- complete the code
                END IF;
                CASE Qreg IS                            -- d(i), i=4,…,9
                        WHEN "000" => W <= "0100"; -- d4
                        WHEN "100" => W <= "0011"; -- d5
                        …               -- complete the code
                        WHEN "001" => W <= "1001"; -- d9
                        WHEN OTHERS => W <= "----";
                END CASE;
        END PROCESS;                    --      student ID
        Q <= Qreg;                      -- d1,d2,d3,d4,d5,d6,d7,d8,d9
END Behaivor;                           -- 5  0  0  4  3  5  4  2  9
```

Figure 1: Student ID number is 500435429