LAB 5 TUTORIAL

VHDL FOR COMBINATORIAL CIRCUITS AND STORAGE ELEMENTS

OVERVIEW

- In this lab we will Use VHDL code to construct combinatorial circuits and circuits with basic storage elements:
 - 1. Mux2to1 (2-to-1 Multiplexer).
 - 2. Decode (2-to-4 Decoder).
 - 3. Encod (Priority Encoder).
 - 4. Johns (Johnson Counter).
 - 5. muxModified (4-to-1 Multiplexer).
 - 6. decodModified (3-to-8 Decoder).

PROCEDURE

1. Mux2to1 (2-to-1 Multiplexer)

- Please follow the following instructions to implement the (2-to-1 Multiplexer):
 - 1. Create a new folder "Lab5" in your "BME328" folder.
 - 2. Create a new folder "Mux2to1" in your "Lab5" folder.
 - 3. Open the Quartus II software, and using the new project wizard, create a new project "Mux2to1" in your "Mux2to1" folder.

- 4. Create a new VHDL file in your "Mux2to1" project (File > New > VHDL File).
- 5. Type the following in the Text Editor and save file as "Mux2to1.vhd":

```
1
   library ieee;
2 use ieee.std logic 1164.all;
3
 4
   entity mux2tol is
                                 : in std_logic;
: out std_logic);
    port (w0, w1, s
5
 6
                f
7
8
   end mux2tol;
9
   architecture Behavior of mux2tol is
10
11 begin
      with s select
12
13
           f <= w0 when '0',
14
               wl when others;
15 end Behavior;
```

- 6. Set "Mux2to1.vhd" as the top-level entity. You can do this by right-clicking on "MUX2to1.vhd" in the "Files" section of the project navigator (located at the right of your Quartus II window) and selecting "Set at Top-Level Entity".
 - 7. Start the compiler. Fix any errors and re-compile. Once the compiler compiles without any errors, move to the next step.
 - 8. Create a symbol for your "Mux2to1.vhd" file. You can do this by right clicking on "Mux2to1.vhd" in the "Files" section of the project navigator (located at the left of your Quartus II window) and selecting "Create Symbol Files for Current File".
 - 9. Create a new University Program VWF (File > New > University Program VWF).
 - 10. Simulate "Mux2to1.vhd" and make sure that your simulation results match the results shown below.
 - 11. Save the VWF file as "Mux2to1.vwf" in your "Mux2to1" folder and take screenshots of your code and your simulation results.



2. Decode (2-to-4 Decoder)

- Please follow the following instructions to implement the 2-to-4 Decoder:
 - 1. Create a new folder "dec2to4" in your "Lab5" folder.
 - 2. Open the Quartus II software, and using the new project wizard, create a new project "dec2to4" in your "dec2to4" folder.

- 3. Create a new VHDL file in your "dec2to4" project (File > New > VHDL File).
- 4. Type the following in the Text Editor and save file as "dec2to4.vhd":

```
1 library ieee;
 2 use ieee.std logic 1164.all;
 3
 4 entity dec2to4 is
     port( w : in std_logic_vector (1 downto 0);
En : in std_logic;
 5
 6
                y : out std_logic_vector(0 to 3));
 7
   end dec2to4;
 8
9
10 architecture Behavior of dec2to4 is
11
       signal Enw : std_logic_vector(2 downto 0);
12 begin
13
       Enw <= En & w;
14
        with Enw select
                    "1000" when "100",
15
           У <=
                    "0100" when "101",
16
17
                    "0010" when "110",
                    "0001" when "111",
18
                    "0000" when others;
19
20 end Behavior;
21
```

5. Set "dec2to4.vhd" as the top-level entity. You can do this by right-clicking on "dec2to4.vhd" in the "Files" section of the project navigator (located at the right of your Quartus II window) and selecting "Set at Top-Level Entity".

- 6. Start the compiler. Fix any errors and re-compile. Once the compiler compiles without any errors, move to the next step.
- Create a symbol for your "dec2to4.vhd" file. You can do this by right clicking on "dec2to4.vhd" in the "Files" section of the project navigator (located at the left of your Quartus II window) and selecting "Create Symbol Files for Current File".
- 8. Create a new University Program VWF (File > New > University Program VWF).
- 9. Simulate "dec2to4.vhd" and make sure that your simulation results match the results shown below.
- 10. Save the VWF file as "dec2to4.vwf" in your "dec2to4" folder and take screenshots of your code and your simulation results.



3. Encod (Priority Encoder)

- Please follow the following instructions to implement the Priority Encoder:
 - 1. Create a new folder "encod" in your "Lab5" folder.
 - 2. Open the Quartus II software, and using the new project wizard, create a new project "encod" in your "encod" folder.

- 3. Create a new VHDL file in your "encod" project (File > New > VHDL File).
- 4. Type the following in the Text Editor and save file as "encod.vhd":

```
l library ieee;
   use ieee.std logic 1164.all;
 2
 3
 4 entity encod is
                         std_logic_vector(3 downto 0);
 5
       port (w : in
             y : out std_logic_vector(1 downto 0);
 6
                z : out std logic);
 7
   end encod;
 8
9
   architecture Behavior of encod is
10
11 begin
12
       process (w)
13
       begin
            y <= "00";
14
15
            if w(1) = '1' then y <= "01"; end if;</pre>
            if w(2) = '1' then y <= "10"; end if;
16
            if w(3) = 'l' then y <= "ll"; end if;
17
18
19
            z <= '1';
20
21
            if w = "0000" then z<= '0'; end if;
22
       end process;
23
   end Behavior;
24
25
26
27
```

5. Set "encod.vhd" as the top-level entity. You can do this by right-clicking on "encod.vhd" in the "Files" section of the project navigator (located at the right of your Quartus II window) and selecting "Set at Top-Level Entity".

- 6. Start the compiler. Fix any errors and re-compile. Once the compiler compiles without any errors, move to the next step.
- 7. Create a symbol for your "encod.vhd" file. You can do this by right clicking on "encod.vhd" in the "Files" section of the project navigator (located at the left of your Quartus II window) and selecting "Create Symbol Files for Current File".
- 8. Create a new University Program VWF (File > New > University Program VWF).
- 9. Simulate "encod.vhd" and make sure that your simulation results match the results shown below.
- 10. Save the VWF file as "encod.vwf" in your "encod" folder and take screenshots of your code and your simulation results.



4. Johns (Johnson Counter)

- Please follow the following instructions to implement the Priority Encoder:
 - 1. Create a new folder "johns" in your "Lab5" folder.
 - 2. Open the Quartus II software, and using the new project wizard, create a new project "johns" in your "johns" folder.

- 3. Create a new VHDL file in your "johns" project (File > New > VHDL File).
- 4. Type the following in the Text Editor and save file as "johns":

```
1
    library ieee;
 2
    use ieee.std logic 1164.all;
3
 4
    entity johns is
      port (Clrn, E, Clkn : in std_logic;
 5
                 STUDENT_ID : out std_logic_vector(3 downto 0);
 6
                                     : out std logic vector(0 to 2));
 7
                 0
    end johns;
8
G
10
   architecture Behavior of johns is
        signal Qreg : std logic vector (0 to 2);
11
  begin
12
13
        process (Clrn, Clkn)
14
        begin
            if (Clrn = '0') then
15
                 Qreg <= "000";
16
             elsif (Clkn 'event and Clkn = '0') then
17
                 if E = 'l' then
18
19
                     Qreg (0) <= not Qreg (2);</pre>
20
                     Qreg (1) <= Qreg (0);
                     Qreg (2) <= Qreg (1);
21
22
                 else
                     Qreg <= Qreg;</pre>
23
24
                 end if:
            end if:
25
                                  -- the last 6 digits of my student ID are 414487
26
27
                                  -- modify the next case statement so that it gives
28
                                  -- the last 6 digits of your student ID based on the
                                  -- corresponding states of the johnson counter.
29
30
            case Qreg is
31
                 when "000" => STUDENT ID <= "0100"; -- 4
32
                 when "100" => STUDENT_ID <= "0001"; -- 1
when "110" => STUDENT_ID <= "0100"; -- 4</pre>
33
34
                 when "111" => STUDENT_ID <= "0100"; -- 4
35
                 when "011" => STUDENT ID <= "1000"; -- 8
36
                 when "001" => STUDENT_ID <= "0111"; -- 7
37
                 when others => STUDENT ID <= "----"; -- error
38
39
             end case;
40
        end process;
41
        Q <= Qreg;
   End Behavior;
42
43
```

5. Set "johns.vhd" as the top-level entity. You can do this by right-clicking on "johns.vhd" in the "Files" section of the project navigator (located at the right of your Quartus II window) and selecting "Set at Top-Level Entity".

- 6. Start the compiler. Fix any errors and re-compile. Once the compiler compiles without any errors, move to the next step.
- 7. Create a symbol for your "johns.vhd" file. You can do this by right clicking on "johns.vhd" in the "Files" section of the project navigator (located at the left of your Quartus II window) and selecting "Create Symbol Files for Current File".
- 8. Create a new University Program VWF (File > New > University Program VWF).
- 9. Simulate "johns.vhd" and make sure that your simulation results match the results shown below.
- 10. Save the VWF file as "johns.vwf" in your "johns" folder and take screenshots of your code and your simulation results.



5. muxModified (4-to-1 Multiplexer)

- Please follow the following instructions to implement the 4-to-1 Multiplexer:
 - 1. Create a new folder "muxModified" in your "Lab5" folder.
 - 2. Open the Quartus II software, and using the new project wizard, create a new project "muxModified" in your "muxModified" folder.

- 3. Open "mux2to1.vhd" that you created earlier. SAVE "mux2to1.vhd" AS "mux2to1.vhd" IN YOUR "muxModified" FOLDER.
- 4. Set "mux2to1.vhd" as the top-level entity. You can do this by right-clicking on "mux2to1.vhd" in the "Files" section of the project navigator (located at the right of your Quartus II window) and selecting "Set at Top-Level Entity".
- 5. Start the compiler. Fix any errors and re-compile. Once the compiler compiles without any errors, move to the next step.
- 6. Create a symbol for your "mux2to1.vhd" file. You can do this by right clicking on "mux2to1.vhd" in the "Files" section of the project navigator (located at the left of your Quartus II window) and selecting "Create Symbol Files for Current File".
- 7. Create a new Block Diagram File in your "muxModified" project (File > New > Block Diagram/Schematic File).
- 8. Connect the components as follows:



- 9. Set "muxModified.bdf" as the top-level entity. You can do this by right-clicking on "muxModified.bdf" in the "Files" section of the project navigator (located at the right of your Quartus II window) and selecting "Set at Top-Level Entity".
 - 10. Start the compiler. Fix any errors and re-compile. Once the compiler compiles without any errors, move to the next step.
 - 11. Create a symbol for your "muxModified.bdf" file. You can do this by right clicking on "muxModified.bdf" in the "Files" section of the project navigator (located at the left of your Quartus II window) and selecting "Create Symbol Files for Current File".
 - 12. Create a new University Program VWF (File > New > University Program VWF).
 - 13. Simulate "muxModified.bdf" and make sure that your simulation results match the results shown below.
 - 14. Save the VWF file as "muxModified.vwf" in your "muxModified" folder and take screenshots of your code and your simulation results.



6. decodModified (3-to-8 Decoder)

- Please follow the following instructions to implement the 3-to-8 Decoder:
 - 1. Create a new folder "decodModified" in your "Lab5" folder.
 - 2. Open the Quartus II software, and using the new project wizard, create a new project "decodModified" in your "decodModified" folder.

- 3. Open "dec2to4.vhd" that you created earlier. SAVE "dec2to4.vhd" AS "dec2to4.vhd" IN YOUR "decodModified" FOLDER.
- 4. Set "dec2to4.vhd" as the top-level entity. You can do this by right-clicking on "dec2to4.vhd" in the "Files" section of the project navigator (located at the right of your Quartus II window) and selecting "Set at Top-Level Entity".
- 5. Start the compiler. Fix any errors and re-compile. Once the compiler compiles without any errors, move to the next step.
- Create a symbol for your "dec2to4.vhd" file. You can do this by right clicking on "dec2to4.vhd" in the "Files" section of the project navigator (located at the left of your Quartus II window) and selecting "Create Symbol Files for Current File".
- 7. Create a new VHDL file in your "decodModified" project (File > New > VHDL File).
- 8. Type the following in the Text Editor and save file as "decodModified":

```
library ieee;
 1
2
    use ieee.std logic 1164.all;
 3
    entity decodModified is
 4
 5
        port (w : in
                                 std logic vector (2 downto 0);
                       : in std_logic;
 6
                En
 7
                               out std logic vector (0 to 7));
                v
                       .
    end decodModified;
8
9
10
    architecture Behavior of decodModified is
11
       component dec2to4
                          in std_logic_vector (1 downto 0);
in std_logic;
out std_logic_vector (3 downto 0));
12
           port (w :
13
                   En :
                    у :
14
15
        end component;
16
17
   begin
18
      dec0: dec2to4 port map (w(1 downto 0), not w(2) and En, y(0 to 3));
19
        decl: dec2to4   port map (w(1 downto 0), En and w(2), y(4 to 7));
20
21
    end Behavior;
22
```

- Set "decodModified.vhd" as the top-level entity. You can do this by right-clicking on "decodModified.vhd" in the "Files" section of the project navigator (located at the right of your Quartus II window) and selecting "Set at Top-Level Entity".
 - 10. Start the compiler. Fix any errors and re-compile. Once the compiler compiles without any errors, move to the next step.
 - 11. Create a symbol for your "decodModified.vhd" file. You can do this by right clicking on "decodModified.vhd" in the "Files" section of the project navigator (located at the left of your Quartus II window) and selecting "Create Symbol Files for Current File".
 - 12. Create a new University Program VWF (File > New > University Program VWF).
 - 13. Simulate "decodModified.vhd" and make sure that your simulation results match the results shown below.
 - 14. Save the VWF file as "decodModified.vwf" in your "decodModified" folder and take screenshots of your code and your simulation results.

