LAB 6 TUTORIAL

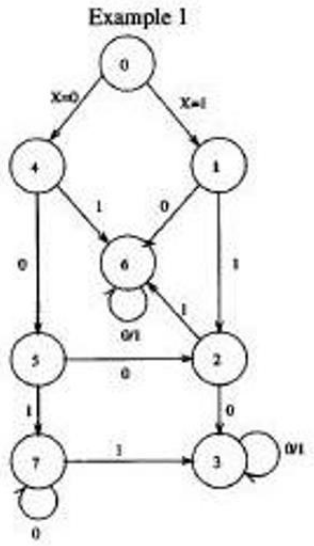
VHDL FOR SEQUENTIAL CIRCUITS: IMPLEMENTING A CUSTOMIZED STATE MACHINE

OVERVIEW

In this lab we will design a finite state machine that cycles through the individual digits of your student ID using the assigned state diagram below.

Each state of the assigned state diagram corresponds to a digit of your student ID. I will use my student ID as an example: 500414487

State 0: 5
State 1: 0
State 2: 0
State 3: 4
State 4: 1
State 5: 4
State 6: 4
State 7: 8
State 8: 7



PROCEDURE

Please follow the following instructions to implement Lab 6:

1. Create a new folder “Lab6” in your “BME328” folder.
2. Open the Quartus II software, and using the new project wizard, create a new project “Lab6” in your “Lab6” folder.

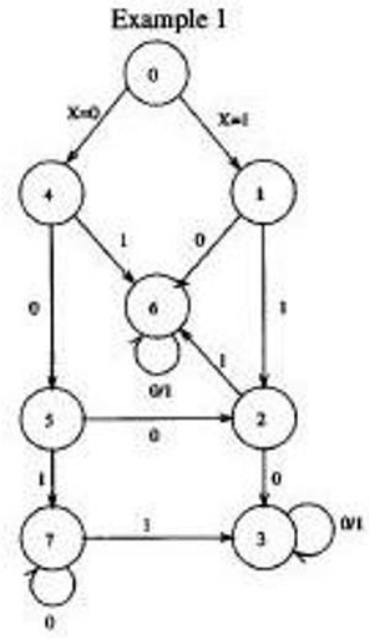
   MAKE SURE THAT YOU CHOOSE THE “EP2C35F672C6” DEVICE IN THE PROJECT WIZARD.
3. Create a new VHDL file in your “Lab6.vhd” project (File > New > VHDL File).
4. Type the following in the Text Editor and save file as “Lab6.vhd”. Remember to modified your file according to your student ID.

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity Lab6 is
5       port(
6           Clock, Resetn    : in    std_logic;
7           data_in          : in    std_logic;
8           student_id       : out   std_logic_vector(3 downto 0);
9           current_state    : out   std_logic_vector(3 downto 0)
10          );
11  end Lab6;
12
13  architecture Behavior of Lab6 is
14      type    state_type              is (s0, s1, s2, s3, s4, s5, s6, s7);
15      signal  y:                      state_type;
16      --signal     control_signal:          std_logic_vector(9 downto 0);
17      begin
18          process (Clock, Resetn)
19          begin
20              if  (Resetn = '0') then
21                  y <= s0;
22              elsif   (Clock'EVENT and Clock = '1') then
23                  case y is
24                      when s0 =>
25                          case data_in is
26                              when '0'    => y <= s4;
27                              when '1'    => y <= s1;
28                              when others => y <= s0;
29                          end case;
30                      when s1 =>
31                          case data_in is
32                              when '0'    => y <= s6;
33                              when '1'    => y <= s2;
34                              when others => y <= s1;
35                          end case;
36                      when s2 =>
37                          case data_in is
38                              when '0'    => y <= s3;
39                              when '1'    => y <= s6;
40                              when others => y <= s2;
41                          end case;
42                      when s3 =>
43                          case data_in is
44                              when '0'    => y <= s3;
45                              when '1'    => y <= s3;
46                              when others => y <= s3;
47                          end case;
48                      when s4 =>
49                          case data_in is
50                              when '0'    => y <= s5;
51                              when '1'    => y <= s6;
52                              when others => y <= s4;
53                          end case;
54                      when s5 =>
55                          case data_in is
56                              when '0'    => y <= s2;
57                              when '1'    => y <= s7;
58                              when others => y <= s5;
59                          end case;
60                      when s6 =>
61                          case data_in is
62                              when '0'    => y <= s6;
63                              when '1'    => y <= s6;
64                              when others => y <= s6;
65                          end case;
66                      when s7 =>
```

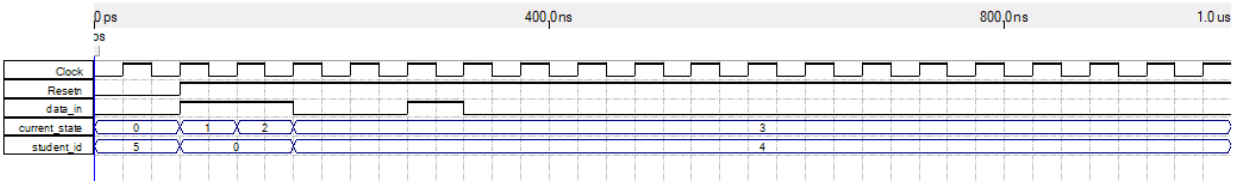
CONTINUED ON NEXT PAGE

```
67                          case data_in is
68                              when '0'    => y <= s7;
69                              when '1'    => y <= s3;
70                              when others => y <= s7;
71                          end case;
72                  end case;
73              end if;
74          end process;
75
76          process (y, data_in)
77          begin
78              case y is
79                  when s0 =>
80                      student_id <= X"5";
81                      current_state <= X"0";
82                  when s1 =>
83                      student_id <= X"0";
84                      current_state <= X"1";
85                  when s2 =>
86                      student_id <= X"0";
87                      current_state <= X"2";
88                  when s3 =>
89                      student_id <= X"4";
90                      current_state <= X"3";
91                  when s4 =>
92                      student_id <= X"1";
93                      current_state <= X"4";
94                  when s5 =>
95                      student_id <= X"4";
96                      current_state <= X"5";
97                  when s6 =>
98                      student_id <= X"4";
99                      current_state <= X"6";
100                 when s7 =>
101                     student_id <= X"8";
102                     current_state <= X"7";
103                 when others =>
104                     student_id <= X"E";
105                     current_state <= X"E";
106             end case;
107         end process;
108 end Behavior;
109
```

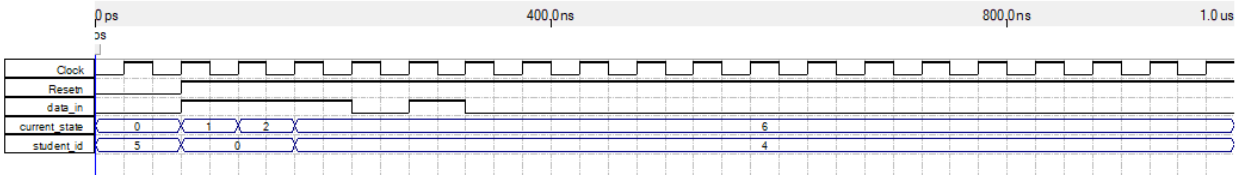
5. Set “Lab6.vhd” as the top-level entity. You can do this by right-clicking on “Lab6.vhd” in the “Files” section of the project navigator (located at the right of your Quartus II window) and selecting “Set at Top-Level Entity”.
6. Start the compiler. Fix any errors and re-compile. Once the compiler compiles without any errors, move to the next step.
7. Create a symbol for your “Lab6.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”.
8. Create a new University Program VWF (File > New > University Program VWF).
9. Simulate “Lab6.vhd” and make sure that your design functions correctly for all cases.
10. Take screenshots of your code and your simulation results.



TEST 1	
Data In Sequence	1 1 0 0 1
Current State Sequence	S0 S1 S2 S3 S3 S3
Student ID Sequence	5 0 0 4 4 4

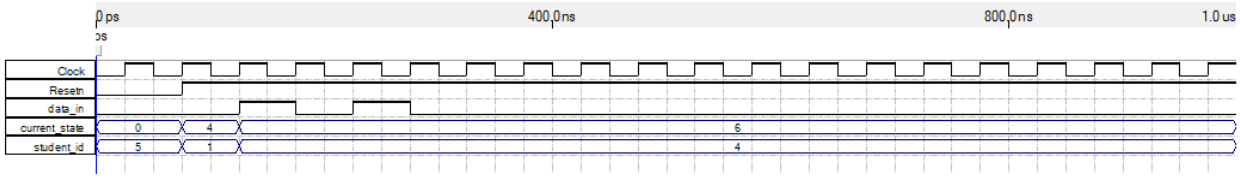


TEST 2	
Data In Sequence	1 1 1 0 1
Current State Sequence	S0 S1 S2 S6 S6 S6
Student ID Sequence	5 0 0 4 4 4

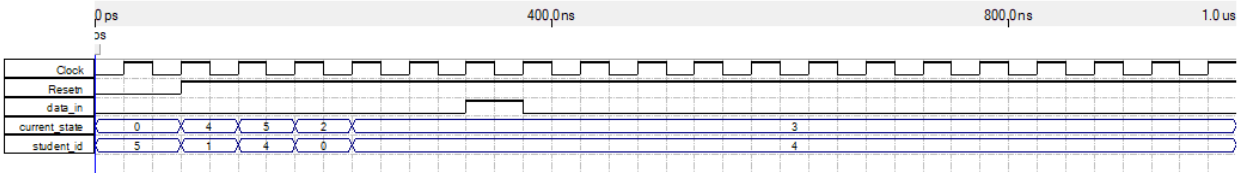


BME 328 – DIGITAL SYSTEMS (FALL 2020)

TEST 3	
Data In Sequence	0 1 0 1
Current State Sequence	S0 S4 S6 S6 S6
Student ID Sequence	5 1 4 4 4



TEST 4	
Data In Sequence	0 0 0 0 0 1
Current State Sequence	S0 S4 S5 S2 S3 S3 S3
Student ID Sequence	5 1 4 0 4 4 4



TEST 5	
Data In Sequence	0 0 1 0 1 0 1
Current State Sequence	S0 S4 S5 S7 S7 S3 S3 S3
Student ID Sequence	5 1 4 8 8 4 4 4