VHDL for Sequential Circuits

COE 328-022 Pei Yang

Nov 14 2022

Introduction:

The objective of this lab was to explore Moore and Mealy machines by designing and simulating an FSM to cycle through our student IDs.

Pre Lab:

The machine given was a mealy FSM # 2.

Diagram

Description automatically generated

Table

Description automatically generatedState/State Assigned table:

A picture containing table

Description automatically generated

Logic Equations:

w = 1 w = 0

Engineering drawing

Description automatically generatedDiagram, engineering drawing

Description automatically generated

A picture containing diagram

Description automatically generated

Engineering drawing

Description automatically generated

Logic Diagram:

Diagram

Description automatically generated

Experiment:

According to the FSM, the states and their outputs should follow the diagram below.

Diagram

Description automatically generated

A skeleton code for the machine was provided in the lab manual. However, the state linking and state assignments would be written depending on our student numbers, FSM machine assigned, and choice of Mealy/Moore. In this lab, the machine used was a #2 FSM Mealy Machine. The code and waveform have been posted below.

The final waveform displays my student number and states in their order. The student number is displayed in 8 bit form using the sseg block, which accepts 4 bit binary input from the FSM and converts the given into the 8 bit output. The due to the way the sseg VHDL was written, the output given by Leds is missing its rightmost value, which is the value of Leds[7]. Since the state waveforms are also the 8 bit output of a sseg block, the waveform generated also needs to be interpreted this way (under the group name Ledss).

In the VHDL, the first process will move from the current state to the next one assigned when the clock is high, otherwise the state will stay the same when clock is low. The second process will assign a 4 bit binary number to the variables current\_state and student\_id which will eventually be fed to the ssegs.

Conclusion:

There were some issues relating to incorrect values displaying in the waveform, however that was due to accidentally writing the incorrect values in the second process of the FSM block.

library ieee;  
use ieee.std\_logic\_1164.all;  
  
entity lab5 is  
port(clk,data\_in,reset:in std\_logic;  
 student\_id,current\_state:out std\_logic\_vector(3 downto 0));  
 end lab5;  
   
 architecture fsm of lab5 is  
 type state\_type is (s0,s1,s2,s3,s4,s5,s6,s7,s8);  
 signal yfsm:state\_type;  
 begin  
 process(clk,reset)  
 begin  
 if reset = '1' then  
 yfsm<=s0;  
 elsif(clk'event and clk='1')then  
   
 case yfsm is  
 when s0 => if data\_in = '1' then yfsm <=s1;  
 else yfsm<=s0;end if;  
 when s1 => if data\_in = '1' then yfsm <=s3;  
 else yfsm<=s1;end if;  
 when s2 => if data\_in = '1' then yfsm <=s0;  
 else yfsm<=s2;end if;  
 when s3 => if data\_in = '1' then yfsm <=s5;  
 else yfsm<=s3;end if;  
 when s4 => if data\_in = '1' then yfsm <=s2;  
 else yfsm<=s4;end if;  
 when s5 => if data\_in = '1' then yfsm <=s7;  
 else yfsm<=s5;end if;  
 when s6 => if data\_in = '1' then yfsm <=s4;  
 else yfsm<=s6;end if;  
 when s7 => if data\_in = '1' then yfsm <=s8;  
 else yfsm<=s7;end if;  
 when s8 => if data\_in = '1' then yfsm <=s6;  
 else yfsm<=s8;end if;  
 end case;  
 end if;  
 end process;  
   
 process (yfsm,data\_in)  
 begin  
 case yfsm is  
 when s0 =>current\_state <="0000";  
 if data\_in='1' then student\_id<="0101";  
 else student\_id<="0000";  
 end if;  
 when s1=>current\_state<="0001";  
 if data\_in='1' then student\_id<="0000";  
 else student\_id<="0001";  
 end if;  
 when s2=>current\_state<="0010";  
 if data\_in='1' then student\_id<="1001";  
 else student\_id<="0010";  
 end if;  
 when s3=>current\_state<="0011";  
 if data\_in='1' then student\_id<="0001";  
 else student\_id<="0011";  
 end if;  
 when s4=>current\_state<="0100";  
 if data\_in='1' then student\_id<="0101";  
 else student\_id<="0100";  
 end if;  
 when s5=>current\_state<="0101";  
 if data\_in='1' then student\_id<="0001";  
 else student\_id<="0101";  
 end if;  
 when s6=>current\_state<="0110";  
 if data\_in='1' then student\_id<="0110";  
 else student\_id<="0110";  
 end if;  
 when s7=>current\_state<="0111";  
 if data\_in='1' then student\_id<="0011";  
 else student\_id<="0111";  
 end if;  
 when s8=>current\_state<="1000";  
 if data\_in='1' then student\_id<="0111";  
 else student\_id<="1000";  
 end if;  
 when others =>current\_state <="1111";  
                Student\_id <="1111";  
  
 end case;  
 end process;  
  
 end fsm;

Graphical user interface, application, table, Excel

Description automatically generated

A picture containing graphical user interface

Description automatically generated

Conclusion: