VHDL for Sequential Circuits

COE 328-022 Pei Yang

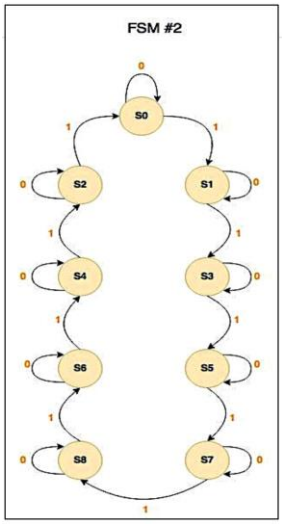
Nov 14 2022

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.



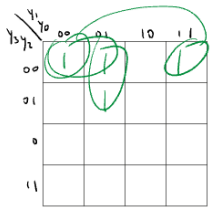
State/State Assigned table:

present state	next state w=0	next state w=1	output w=0	output w=1
S_0	S_0	S_1	d_1	d_2
S_1	S_1	S_3	d_2	d_3
S_2	S_2	S_0	d_9	d_1
S_3	S_3	S_5	d_3	d_4
S_4	S_4	S_2	d_8	d_9
S_5	S_5	S_7	d_4	d_5
S_6	S_6	S_4	d_7	d_8
S_7	S_7	S_8	d_5	d_6
S_8	S_8	S_6	d_6	d_0

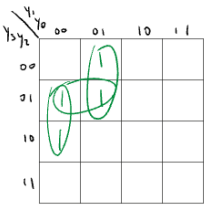
	present state $Y_3Y_2Y_1Y_0$	next state $Y_3Y_2Y_1Y_0$ w=0	next state w=1	output $Z_3Z_2Z_1Z_0$ w=0	output w=1
S_0	0000	0000	0001	d_1	d_2
S_1	0001	0001	0011	d_2	d_3
S_3	0011	0011	0101	d_9	d_1
S_5	0101	0101	0111	d_3	d_4
S_7	0111	0111	1000	d_8	d_9
S_8	1000	1000	0110	d_4	d_5
S_6	0110	0110	0100	d_7	d_8
S_4	0100	0100	0010	d_5	d_6
S_2	0010	0010	0000	d_6	d_0

Logic Equations:

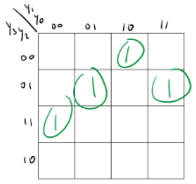
w = 1



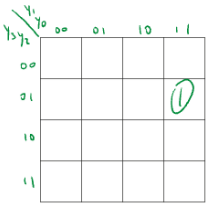
$Y_0 = \bar{Y}_3\bar{Y}_2\bar{Y}_1 + \bar{Y}_3\bar{Y}_2 + \bar{Y}_3\bar{Y}_1Y_0$



$Y_1 = \bar{Y}_3\bar{Y}_1Y_0 + \bar{Y}_3Y_2\bar{Y}_1 + \bar{Y}_1\bar{Y}_0$

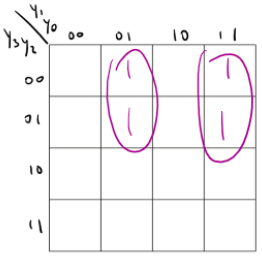


$Y_2 = Y_3Y_2\bar{Y}_1\bar{Y}_0 + \bar{Y}_3Y_2\bar{Y}_1Y_0 + \bar{Y}_3\bar{Y}_2Y_1\bar{Y}_0 + \bar{Y}_3Y_2Y_1Y_0$

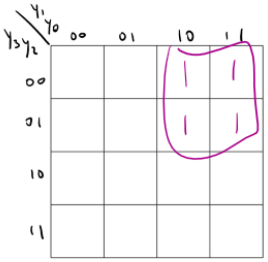


$Y_3 = \bar{Y}_3Y_2Y_1Y_0$

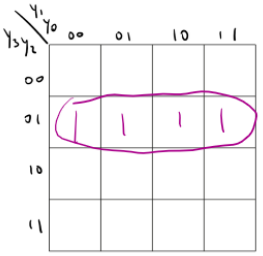
w = 0



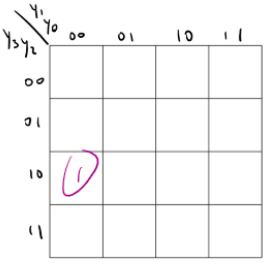
$Y_0 = \bar{Y}_3Y_1\bar{Y}_0 + \bar{Y}_3Y_1Y_0$



$Y_1 = \bar{Y}_3Y_1$

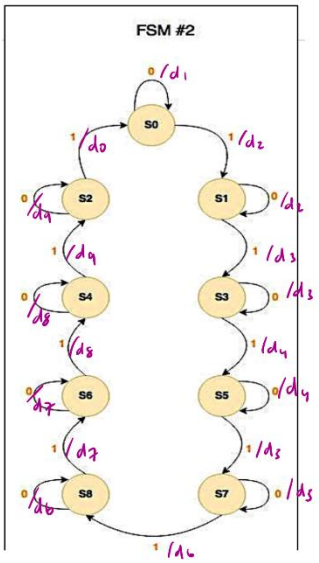


$Y_2 = \bar{Y}_3Y_2$



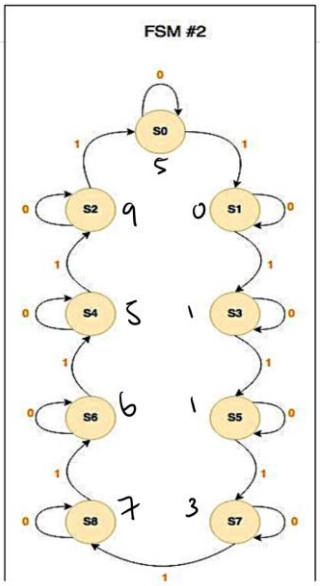
$Y_3 = Y_3\bar{Y}_2\bar{Y}_1\bar{Y}_0$

Logic Diagram:



Experiment:

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



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).

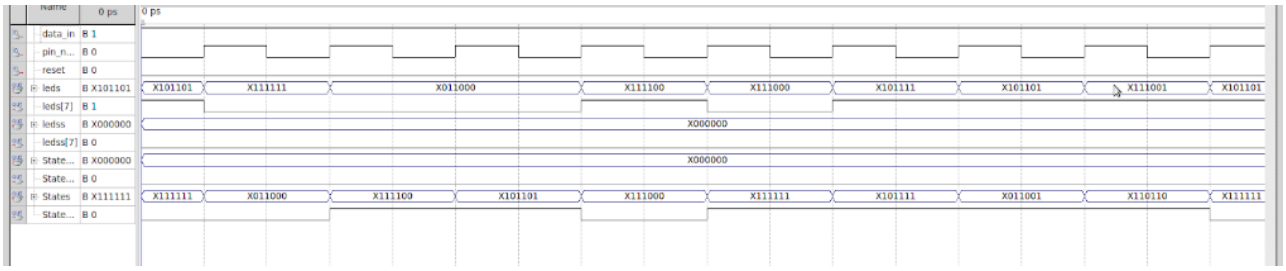
In the VHDL, the first process will move from the current state to the next one assigned when the clock is high, otherwise the state will stay the same when clock is low. The second process will assign a 4 bit binary number to the variables current_state and student_id which will eventually be fed to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

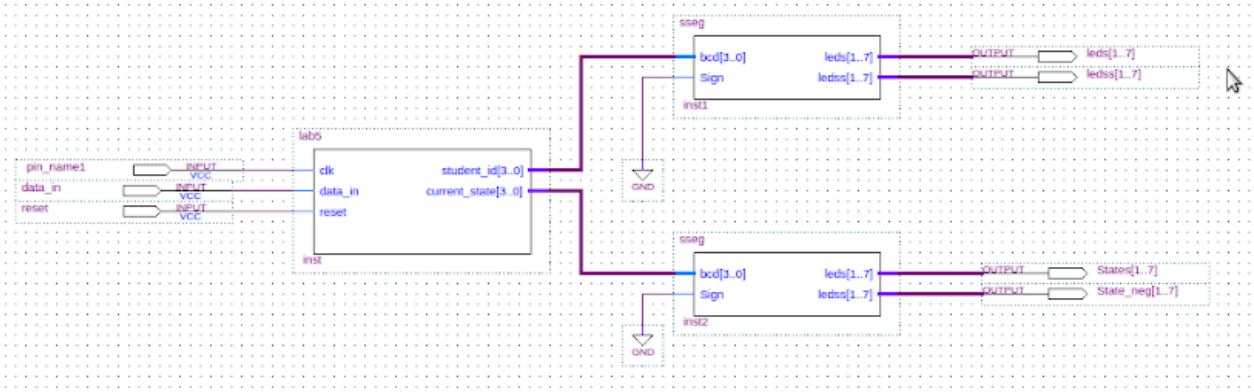
Conclusion:

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;
```





Conclusion: