EDA PROGECT 1

CSE-215

ZIAD TAREK SALAH

17P3047 CESS

Ziadgyr57@gmail.com

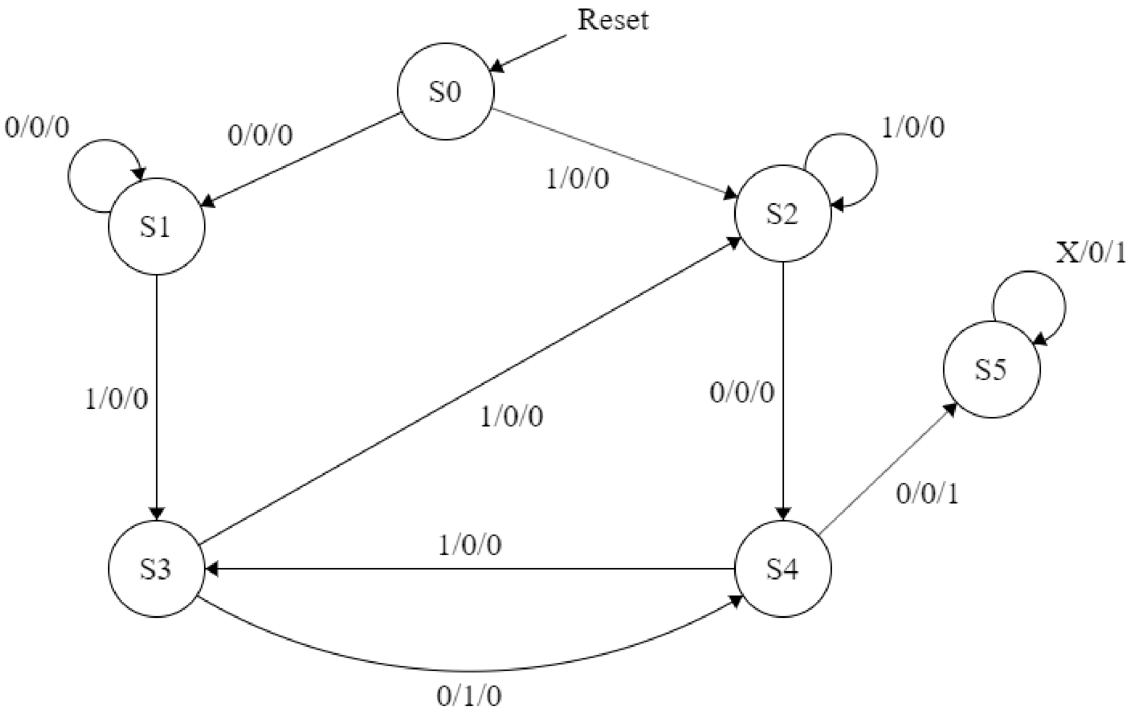
Introduction

A finite state machine (sometimes called a finite state automaton) is a computation model that can be implemented with hardware or software and can be used to simulate sequential logic and some computer programs. Finite state machines can be used to model problems in many fields including mathematics, artificial intelligence, games, and linguistics.

The behavior of state machines can be observed in many devices in modern society that perform a predetermined sequence of actions depending on a sequence of events with which they are presented. Simple examples are vending machines, which dispense products when the proper combination of coins is deposited, elevators, whose sequence of stops is determined by the floors requested by riders, traffic lights, which change sequence when cars are waiting, and combination locks, which require the input of a sequence of numbers in the proper order.

In this project we are going to design a finite state machine that recognizes a string of 010 and terminates when the string 100 appears. The state machine will have only one input and two outputs (output, termination) in addition to clock, VDD, and VSS.

Finite State Machine Diagram



We’ll assume the following Encoding:

S0	→	000
S1	→	001
S2	→	010
S3	→	011
S4	→	100
S5	→	101

Finite State Machine Transition Table

Current State				Next State				
C2	C1	C0	IN	N2	N1	N0	OUT	Ter
0	0	0	0	0	0	1	0	0
0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	1	0	0
0	0	1	1	0	1	1	0	0
0	1	0	0	1	0	0	0	0
0	1	0	1	0	1	0	0	0
0	1	1	0	1	0	0	1	0
0	1	1	1	0	1	0	0	0
1	0	0	0	1	0	1	0	1
1	0	0	1	0	1	1	0	0
1	0	1	X	1	0	1	0	1

Testbench Strategy Table

Test Number	Tested Feature	Input	Delay	Expected Output	
				Output	Termination
1	when reset=1	"000"	20ns	0	0
2	when reset=0 & not terminated	"010"	20ns	1	0
3	when reset=0 & not terminated	"100"	20ns	0	1
4	when reset=0 & not terminated	any sequence	20ns	0	0
5	when reset=0 & terminated	any sequence	20ns	0	1

VHDL Code:

```
Library IEEE;
use IEEE.all;
USE ieee.std_logic_1164.ALL;

entity fsm is
port  (
                  ck    : in        bit;
                  vdd   : in        bit;
                  vss   : in        bit;
                  inp   : in        bit;
                  reset : in        bit;
                  op    : out       bit;
                  tr    : out       bit
     );
end fsm;

architecture archi of fsm is
  type STATE_TYPE is (S0, S1, S2, S3, S4, S5);  --S5 is the termination State
  signal ns, cs : STATE_TYPE;

begin

-- Process (1)
  process (cs, inp, reset)
  begin
      if (reset='1') then
      ns<=S0;
      else
```

```
CASE cs is
   when S0 =>
                      if (inp='1') then
                            op <= '0';
                            tr <= '0';
                            ns <= S2;
   else
                            op <= '0';
                            tr <= '0';
                            ns <= S1;
   end if;

   when S1 =>
   if (inp='1') then
                            op <= '0';
                            tr <= '0';
                            ns <= S3;
   else
                            op <= '0';
                            tr <= '0';
                            ns <= S1;
   end if;

                      when S3 =>
   if (inp='1') then
                            op <= '0';
                            tr <= '0';
                            ns <= S2;
   else
                            op <= '1';
```

```
                       tr <= '0';

                       ns <= S4;

end if;


when S4 =>

if (inp='1') then

                       op <= '0';

                       tr <= '0';

                       ns <= S3;

else

                       op <= '0';

                       tr <= '1';

                       ns <= S5;

end if;


                  when S2 =>

 if (inp='1') then

                       op <= '0';

                       tr <= '0';

                       ns <= S2;

 else

                       op <= '0';

                       tr <= '0';

                       ns <= S4;

 end if;


             when S5 =>

                       op <= '0';

                       tr <= '1';

                       ns <= S5;
```

```
         end case;
     end if;
    end process;
-- Process (2)
  process(ck)
  begin
                    if(ck = '1' and not ck'stable)then

                              cs <= ns;
     end if;
  end process;

end archi;
```

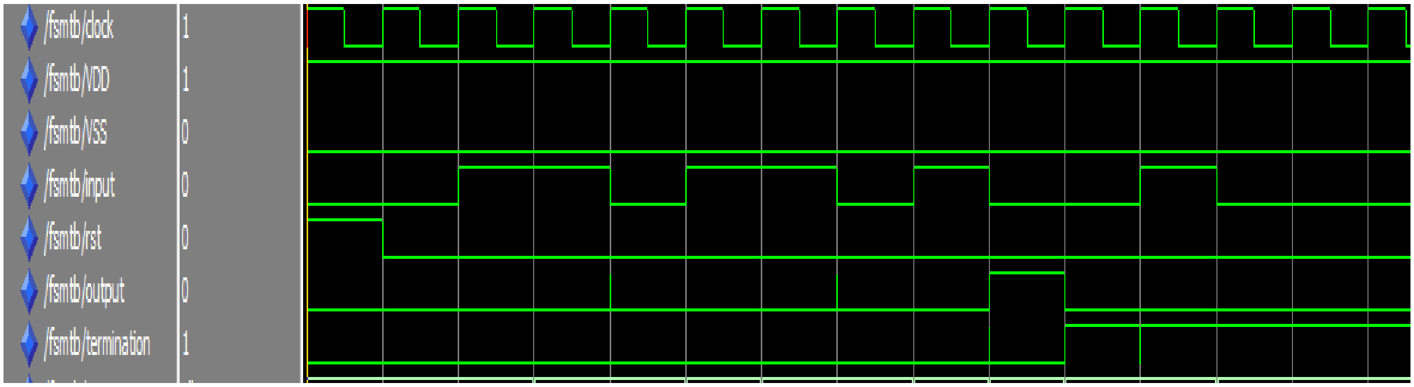
VHDL TestBench Code:

```
Library IEEE;
use IEEE.all;
USE ieee.std_logic_1164.ALL;

Entity fsmTb is
End Entity fsmTb;

Architecture archiTb of fsmTb is
Component fsm is
port  (
                ck    : in      bit;
                vdd   : in      bit;
                vss   : in      bit;
                inp   : in      bit;
                reset : in      bit;
                op    : out     bit;
                tr    : out     bit
    );
end Component fsm;

For dut: fsm use entity work.fsm(archi);



SIGNAL clock : bit := '0';
SIGNAL VDD   : bit := '1';
SIGNAL VSS   : bit := '0';
SIGNAL input : bit := '0';
SIGNAL rst : bit := '1';
SIGNAL output  : bit := '0';
SIGNAL termination  : bit := '0';


constant clk_period : time := 20 ns;
constant sequence1 : bit_vector := "11011010010";
constant sequence2 : bit_vector := "00101010010";
SIGNAL string_Recognized : bit_vector (2 downto 0);


Begin
        dut : fsm PORT MAP ( clock, VDD, VSS, input, rst, output, termination);


        clk_Process: process
        begin
            clock <= '1';
            wait for clk_period/2;
            clock <= '0';
            wait for clk_period/2;
        end process;

        stim_proc: PROCESS IS
        BEGIN
        rst <= '1';
        wait for clk_period;
        rst <= '0';

        string_Recognized <= "000";
        for i in 0 to sequence1'length-1 loop
            wait for clk_period;
            if(rst <= '0') then
                string_Recognized <= string_Recognized(1) & string_Recognized(2) & input;
            else string_Recognized <= "000";
            end if;
        input <= sequence1(i);
        end loop;
        string_Recognized <= string_Recognized(1) & string_Recognized(2) & input;



        if(rst = '1') then
            wait for clk_period;
            Assert output = '0' and termination = '0'
            Report "output=0 & termination=0"
            Severity error;

        elsif(termination = '0' and rst = '0') then
            if    (string_Recognized = "010") then
                wait for clk_period;
                Assert output = '1' and termination = '0'
                Report "output=1 & termination=0"
                Severity error;
            elsif (string_Recognized = "100") then
                wait for clk_period;
                Assert output = '0' and termination = '1'
                Report "output=0 & termination=1"
                Severity error;
            else    wait for clk_period;
```

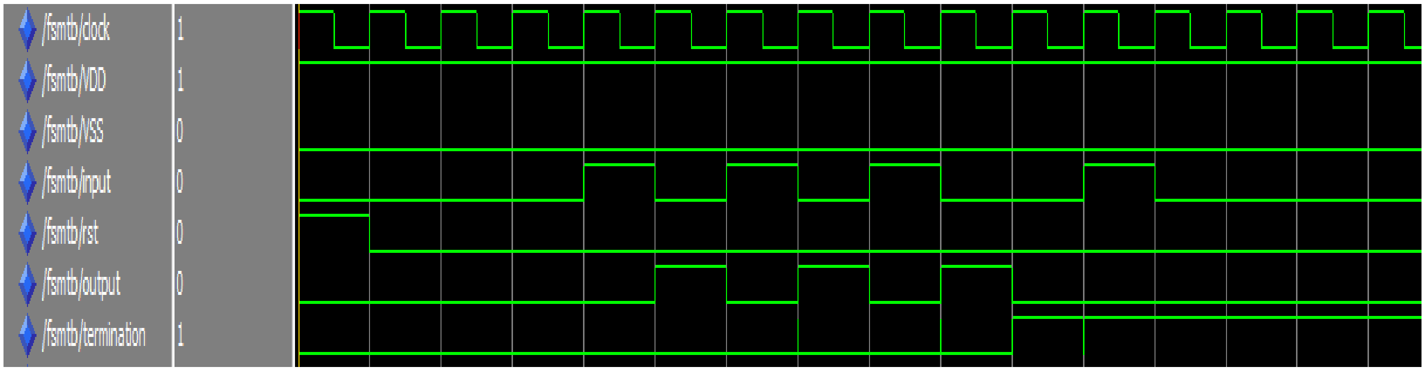
```
                Assert output = '0' and termination = '0'
                Report "output=0 & termination=0"
                Severity error;
            end if;


        elsif (termination = '1' and rst = '0') then
                wait for clk_period;
                Assert output = '0' and termination = '1'
                Report "output=0 & termination=1"
                Severity error;

        end if;

WAIT; -- stop process simulation run
END PROCESS;
END ARCHITECTURE archiTb;
```

Simulation:

For input "11011010010" the output is:



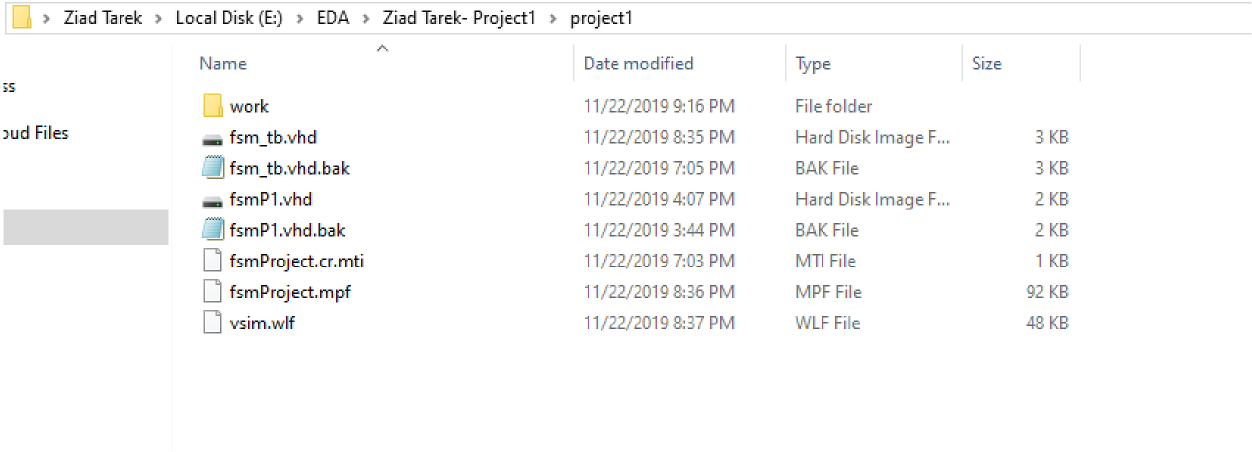
For input "00101010010" the output is:



Appendices

E:\EDA\Ziad Tarek- Project1\project1\fsmP1.vhd

E:\EDA\Ziad Tarek- Project1\project1\fsm_tb.vhd



› Ziad Tarek › Local Disk (E:) › EDA › Ziad Tarek- Project1 › project1

Name	Date modified	Type	Size
work	11/22/2019 9:16 PM	File folder	
fsm_tb.vhd	11/22/2019 8:35 PM	Hard Disk Image F...	3 KB
fsm_tb.vhd.bak	11/22/2019 7:05 PM	BAK File	3 KB
fsmP1.vhd	11/22/2019 4:07 PM	Hard Disk Image F...	2 KB
fsmP1.vhd.bak	11/22/2019 3:44 PM	BAK File	2 KB
fsmProject.cr.mti	11/22/2019 7:03 PM	MTI File	1 KB
fsmProject.mpf	11/22/2019 8:36 PM	MPF File	92 KB
vsim.wlf	11/22/2019 8:37 PM	WLF File	48 KB