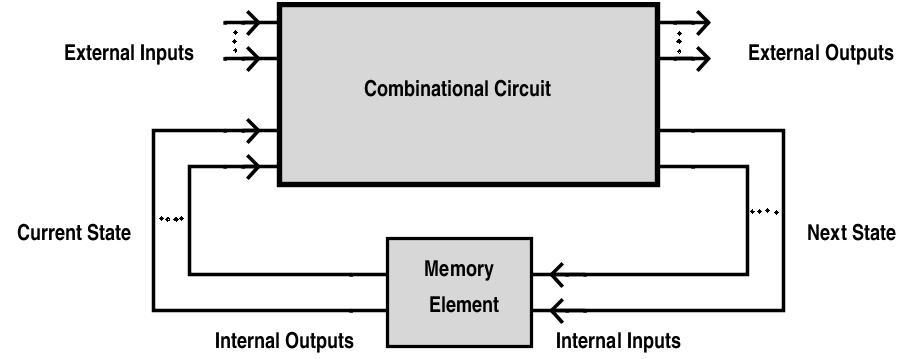
1010 Sequence Detector



**Sequential Circuits:**

Sequential circuits works on a clock cycle which may be synchronous or asynchronous. The figure shows a basic diagram of sequential circuits. Sequential circuits use current inputs and previous inputs by storing the information and putting back into the circuit on the next clock cycle.



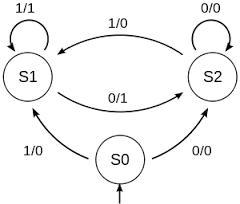
**Finite State Machine (FSM):**

A FSM is a model used to design sequential logic circuits. It is conceived as an abstract machine that can be in one of a finite number of states. The machine is in only one state at a time; the state it is in at any given time is called the current state. It can change from one state to another when initiated by a triggering event or condition, this is called a transition. A particular FSM is defined by a list of its states, and the triggering condition for each transition. It can be implemented using models like Mealy and Moore machine. For this expt., we will use Mealy model implementation.

**Mealy Machine:**

In this model of FSM, the output values are determined both by its current state and the current inputs. The state diagram of a mealy machine associates an output value with each transition edge.

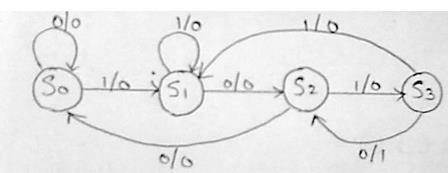
Example:



In the above diagram 1/1, 1/0, 0/1, 0/0 represent input/output and S0, S1, S2 represent the states. Consider a part of the state diagram S1 -> S2 where “0/1” is written on the arrow. This can be interpreted as, when the current state of the system is S1 and when input “0” is applied, the system goes into next state - S2 and the output of the system is “1”.

**Sequence detector (1010):**

Suppose a sequence detector is to be designed to detect a sequence 1010. Then the state diagram will be:



Note that this state diagram is considering overlap i.e. if we have input 11010101 we will have output 00001010. For same input, non-overlap case will have output 0001000. Either cases are correct but we will consider only overlap case henceforth.

**Tesbench code:**

Library IEEE;

use IEEE.std\_logic\_1164.all;

Entity seq\_det\_tb is

end seq\_det\_tb;

Architecture test of seq\_det\_tb is

Component seq\_detec is

port( clk : in BIT;

data\_in : in std\_logic;

rst : in BIT;

seq\_det : out BIT);

end Component seq\_detec;

signal clk : BIT;

signal data\_in : BIT;

signal rst : BIT;

signal seq\_det : BIT;

signal test\_seq : std\_logic\_vector(31 downto 0) := "11111010010111011010110011101001";

signal i : integer range 0 to 31;

begin

SQD : seq\_detec port map (clk, data\_in, rst, seq\_det);

tb\_process : process

begin

clk <= '0';

wait for 5 ns;

clk <= '1';

if(i = 31) then

i <= 0;

else

i <= i + 1;

end if;

data\_in <= test\_seq(i);

wait for 5 ns;

end process;

end test;

configuration config\_tb of seq\_det\_tb is

for test

for SQD : seq\_detec

use entity work.seq\_detec(sequence\_detector);

end for;

end for;

end config\_tb;

The above code is given an input of 32 bits which are fed to the VHDL entity one after the other. When the input consists of sequence “1010” or more “10” overlapping, it gives an output ‘1’. In the sequence given in the above testbench, three overlapping “10” and a single sequence of “1010” has been included.

**GROUP MEMBERS:**

KARLAPATI RAJA PRADYUMNA - EE16BTECH11017

|  |  |
| --- | --- |
| CHOPPA ARAVIND REDDY | - EE16BTECH11004 |
| AELLA ABHILASH REDDDY | - EE16BTECH11001 |
| PAMULA BHARGAV RAM | -EE16BTECH11024 |