DIGITAL DESIGN USING HDL LABORATORY (UE18EC206)

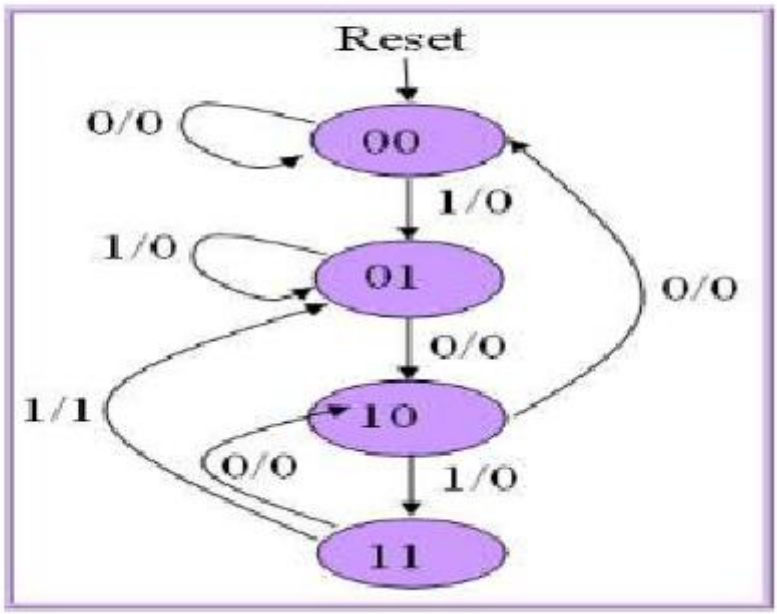
EXPERIMENT-11

SIMULATION OF MEALY AND MOORE MODELS USING XILINX VIVADO TOOL

Aim: Write a verilog code to simulate Mealy and Moore machine Sequence detector for the given sequence

Sequence detector (1011)

- Mealy model



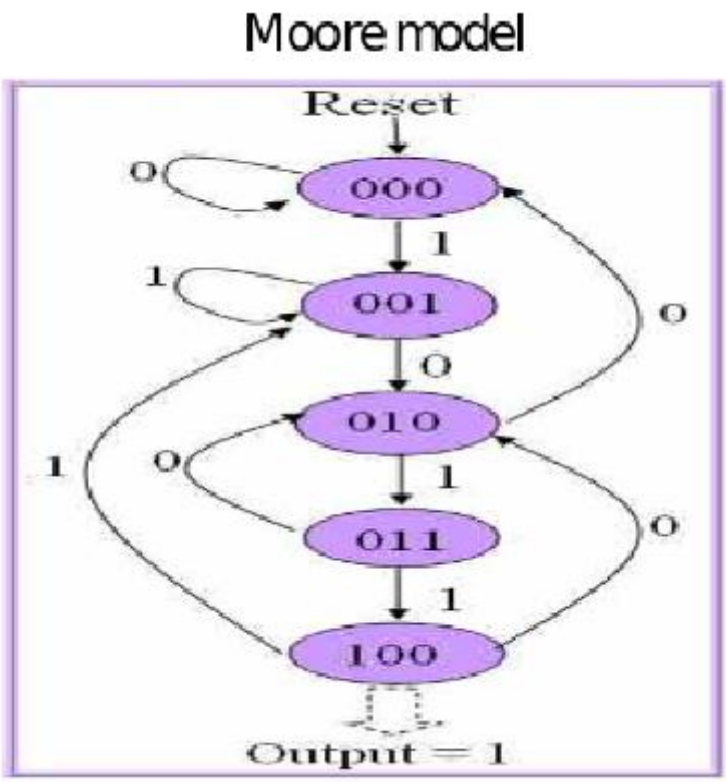

```
module mealy_1011( input clk, reset, inp, output outp);
  reg outp;
  reg [1:0] state;
  reg [1:0] next_state;
always @(posedge clk)
  begin
    if (reset)
 state <= 2'b00;
```

```
    else
      state <= next_state;
  end
always @(state, inp)
  case( state )
    2'b00: begin
             if( inp )
               begin
                 next_state <= 2'b01;
                 outp <= 0;
               end
             else
               begin
                 next_state <= 2'b00;
                 outp <= 0;
               end
             end
2'b01: begin
           if( inp )
             begin
               next_state <= 2'b01;
               outp <= 0;
             end
```

```
            else
               begin
                  next_state <= 2'b10;
                   outp <= 0;
                end
               end
2'b10: begin
            if( inp )
               begin
                  next_state <= 2'b11;
                  outp <= 0;
                end
             else
               begin
                   next_state <= 2'b00;
                   outp <= 0;
               end
          end
2'b11: begin
            if( inp )
              begin
                 next_state <= 2'b01;
                  outp <= 1;
```

```
            end
          else
            begin
                next_state <= 2'b10;
                outp <= 0;
              end
          end
endmodule
```

Moore:




```
module moore_1011( input clk, reset, inp, output outp);
  reg outp;
  reg [2:0] state;
```

```
  reg [2:0] next_state;
always @(posedge clk)
  begin
    if (reset)
        state <= 3b000;
    else
        state = next_state;
  end
always @(state, inp)
  case( state )
    3'b000: begin
              if( inp )
                next_state <= 3'b001;
              else
                next_state <= 3'b000;
            end
    3'b001: begin
              if( inp )
                next_state <= 3'b001;
              else
                next_state <= 3'b010;
            end
3'b010: begin
```

```
                if( inp )
                    next_state <= 3'b011;
                else
                    next_state <= 3'b000;
                    end
        3'b011: begin
                if( inp )
                    next_state <= 3'b100;
                else
                    next_state <= 3'b010;
                    end
        3'b100: begin
                if( inp )
                    next_state <= 3'b001;
                else
                    next_state <= 3'b010;
                    end
            endcase
always @ (state or reset)
        begin
            if (reset)
                outp = 0;
            else
```

```
            case(state)
               3'b000: outp = 0;
               3'b001: outp = 0;
               3'b010: outp = 0;
               3'b011: outp = 0;
               3'b100: outp = 1;
            endcase
        end
endmodule
```

Testbench:

```
module mealy_1011_tb();
        reg clk,reset,inp;
        wire outp;
        mealy_1011 uut(clk,reset,inp,outp);
        initial
                begin
                  clk=0; reset=0; inp=0;
                  #10 reset=1;
                  #15 reset=0;
                  #15 inp=1;
                  #15 inp=0;
                  #10 inp=1;
                  #10 inp=1;
                  #15 inp=1;
                  #15 inp=0;
                  #10 inp=1;
```

```
        #10 inp=1;
        #10 inp=0;
    end
        always #5 clk=~clk;
endmodule
```