Faculty of Engineering and Technology
Electrical and Computer Engineering Department

ENCS5332 - VLSI

Hw3: Sequence detector design Assignment

Prepared by

Arwa Doha - 1190324

Instructor

Dr. Khader Mohammad

BIRZEIT

January – 2024

•Sequence detector:

The Moore FSM keeps detecting a binary sequence from a digital input and the output of the FSM goes high only when a "1101" sequence is detected.

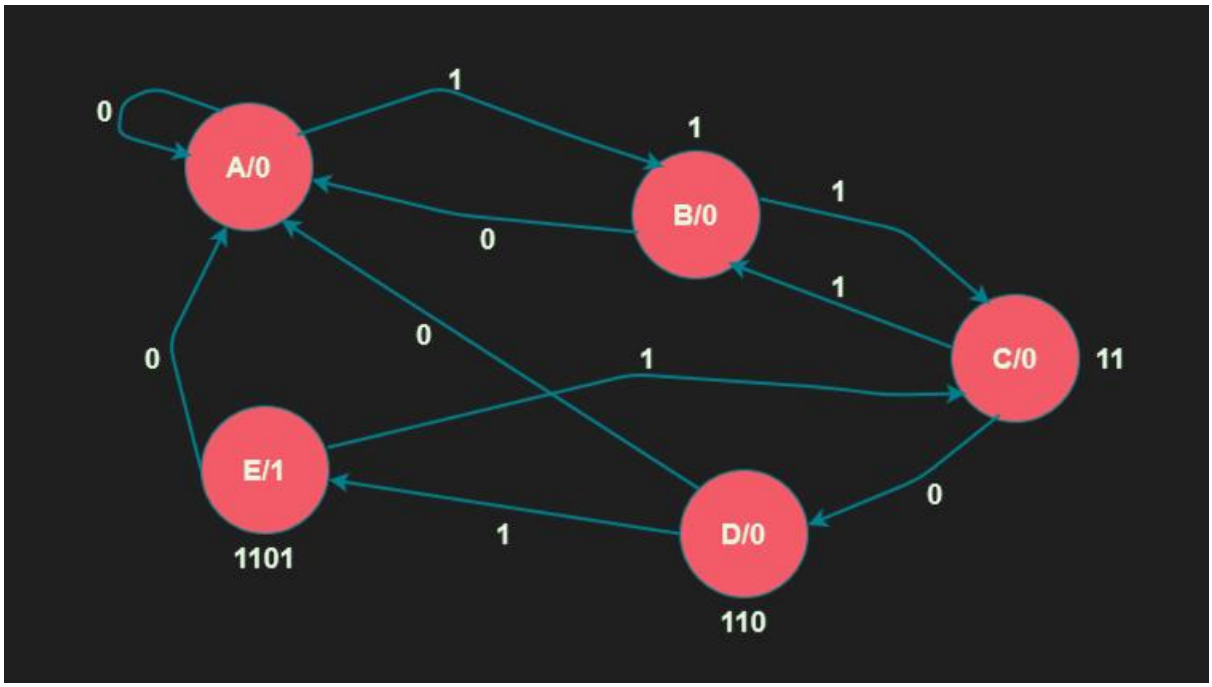
•We need to: -Draw state diagram (flow chart) of the Moore FSM for the sequence detector (see below requirements). -Implement Verilog project to present a full Verilog code for Sequence Detector using Moore FSM. -Implement Verilog Testbench for the Moore FSM sequence detector.

I. Introduction

In the realm of digital design, the creation of reliable and efficient sequence detectors plays a pivotal role in numerous applications, ranging from data communication to embedded systems. This report delves into the design and implementation of a Moore Finite State Machine (FSM) for a sequence detector tasked with recognizing the specific binary sequence "1101." The significance of such detectors lies in their ability to discern predetermined patterns within digital input streams, enabling applications to respond intelligently to specific sequences of interest.

Finite State Machines (FSMs) are powerful tools in digital design, providing a systematic approach to modeling complex behaviors. In this context, the Moore FSM is employed to construct a sequence detector that transitions through distinct states based on the input stream, with the ultimate goal of signaling the detection of the target sequence. The report outlines the development process, beginning with the formulation of a state diagram to guide the design and culminating in the Verilog implementation of the sequence detector.

II. State Diagram



III. Verilog code for "Moore FSM Sequence Detector"

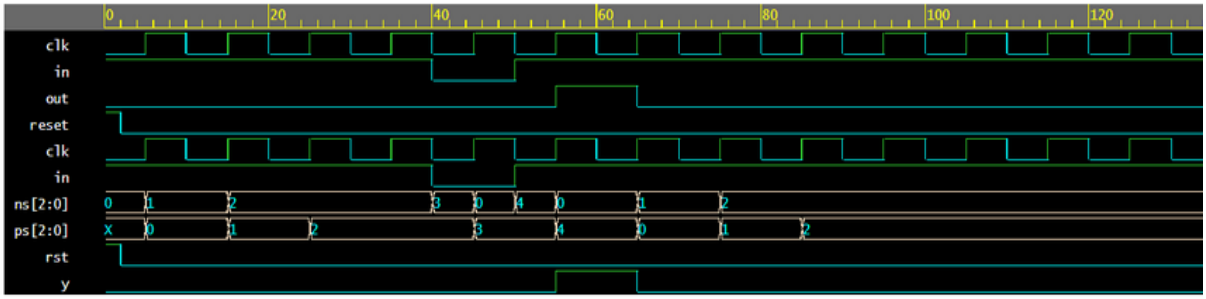
```
module tbw_1101;
  reg clk,reset,in;
  wire out;

  SequenceDetector uut(in,clk,reset,out);

  initial begin
    $dumpfile("dump.vcd"); $dumpvars;
  end

  initial begin
  clk=0;
    #200 $finish;
     end
   always #5 clk=~clk;

  initial begin
    reset=1;    //in=10'b011010110100;
    #2 reset=0;
    #150;
  end
  initial begin
    in=1;
    #10 in=1;
    #10 in=1;
    #10 in=1;
    #10 in=0;
    #10 in=1;
    #10 in=1;
    #10 in=1;
    #10;

  end
endmodule
```

IV. Test Bench

```
module SequenceDetector (in,clk,rst,y);

input in,clk,rst;
output reg y;
  reg [2:0] ps,ns;

parameter s0 = 3'b000 ,s1 = 3'b001,s2 = 3'b010,s3 = 3'b011, s4 = 3'b100;

always @ (posedge clk)
begin
  if (rst)
    ps<= s0;
  else
    ps <= ns;
end

  always @ (ps or in)
begin
  case (ps)
        s0: begin
          y = 0;
          if (in)
              ns = s1;
          else
              ns = ps; //ps or s0
      end
    s1:  begin
          y = 0;
          if (in)
            ns = s2;
          else
            ns = s0;
       end
    s2:  begin
        y = 0;

        if (in)
          ns = s2;
        else
          ns = s3;
      end
    s3:  begin
        y = 0;
        if (in)
          ns = s4;
        else
          ns = s0;
        end
    s4: begin
        y=1;
        if (in)
          ns = s0;

        else
          ns = s0;
        end

    default: begin
      y=0;
      ns = s0;
    end
  endcase
end
endmodule
```

V. Simulation Results

After rigorous simulation, the sequence detector consistently and accurately identified the binary sequence '1101,' confirming the robust performance of the Moore Finite State Machine design.



VI. Conclusion

In conclusion, this assignment has not only provided hands-on experience in digital design and Verilog implementation but has also underscored the importance of systematic approaches, such as Finite State Machines, in solving pattern recognition challenges. The successful creation of the Moore FSM sequence detector serves as a testament to the efficacy of such design methodologies in real-world applications. Moving forward, the insights gained from this project can be applied to more complex designs and integrated into broader digital systems, further solidifying the understanding of digital design principles.