

Final Project Report

Anirudh C (IMT2017006)

May 8, 2018

Contents

1	Project Report	1
1.1	Moore '01' Sequence Detector	1
1.1.1	Detector Module	1
1.1.2	Detector Test Bench	2
1.1.3	Timing	4
1.1.4	Links	4
1.2	Traffic Light Controller	4
1.2.1	TLC module	4
1.2.2	TLC Test Bench	5
1.2.3	Timing	9
1.2.4	Links	10

1 Project Report

1.1 Moore '01' Sequence Detector

1.1.1 Detector Module

The module for the detector:

```
module zero_one_detector(input A, input clk, input rst, output Y);
    reg [1:0] state, nextstate;
    parameter S0 = 2'b00;
    parameter S1 = 2'b01;
    parameter S2 = 2'b10;
    always_ff @ (posedge clk, posedge rst)
        if (rst) state <= S0;
        else     state <= nextstate;
```

```

always @ (*)
  case(state)
    S0: if (A) nextstate = S0;
        else nextstate = S1;
    S1: if (A) nextstate = S2;
        else nextstate = S0;
    S2: if (A) nextstate = S0;
        else nextstate = S1;
    default: nextstate = S0;
  endcase
  assign Y = (state == S2);
endmodule

```

1.1.2 Detector Test Bench

The test bench:

```

`include "zero_one_detector.vh"
module test_zero_one();
  reg clk, rst; reg A, Yexpected;
  wire Y;
  zero_one_detector dut(A,clk,rst,Y);
  always
  begin
    clk = 1; #5; clk = 0; #5;
  end
  initial begin
    rst = 1;
    #6;
    rst=0;
    A = 0; Yexpected = 0; #10;
    if (Y !== Yexpected) begin
      $display("E: A = %b, Yexpected = %b, Y = %b",A,Yexpected,Y);
    end
    else $display("D: A = %b, Yexpected = %b, Y = %b",A,Yexpected,Y);

    A = 1; Yexpected = 1; #10;
    if (Y !== Yexpected) begin
      $display("E: A = %b, Yexpected = %b, Y = %b",A,Yexpected,Y);
    end
    else $display("D: A = %b, Yexpected = %b, Y = %b",A,Yexpected,Y);
  end
endmodule

```

```

A = 0; Yexpected = 0; #10;
if (Y != Yexpected) begin
    $display("E: A = %b, Yexpected = %b, Y = %b",A,Yexpected,Y);
end
else $display("D: A = %b, Yexpected = %b, Y = %b",A,Yexpected,Y);

A = 1; Yexpected = 1; #10;
if (Y != Yexpected) begin
    $display("E: A = %b, Yexpected = %b, Y = %b",A,Yexpected,Y);
end
else $display("D: A = %b, Yexpected = %b, Y = %b",A,Yexpected,Y);

A = 1; Yexpected = 0; #10;
if (Y != Yexpected) begin
    $display("E: A = %b, Yexpected = %b, Y = %b",A,Yexpected,Y);
end
else $display("D: A = %b, Yexpected = %b, Y = %b",A,Yexpected,Y);

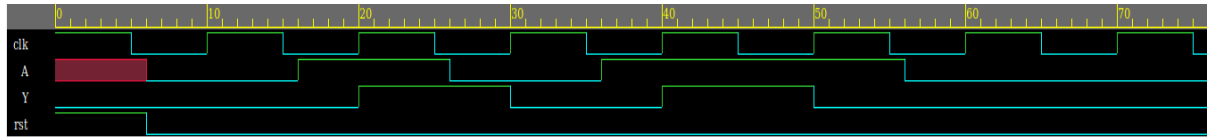
A = 0; Yexpected = 0; #10;
if (Y != Yexpected) begin
    $display("E: A = %b, Yexpected = %b, Y = %b",A,Yexpected,Y);
end
else $display("D: A = %b, Yexpected = %b, Y = %b",A,Yexpected,Y);

A = 0; Yexpected = 0; #10;
if (Y != Yexpected) begin
    $display("E: A = %b, Yexpected = %b, Y = %b",A,Yexpected,Y);
end
else $display("D: A = %b, Yexpected = %b, Y = %b",A,Yexpected,Y);
$finish;

end
endmodule

```

1.1.3 Timing



1.1.4 Links

Module

Waveform

1.2 Traffic Light Controller

1.2.1 TLC module

1. Controller Module

```
module traffic_light_controller(input TA, TB, clk, rst, output RA, YA, GA, RB, YB,
    typedef enum logic [1:0] {S0,S1,S2,S3} statetype;
    statetype state, nextstate;
    always @ (posedge clk, posedge rst)
        if (rst) state <= S0;
        else      state <= nextstate;
    always @ (*)
        case(state)
            S0: if (TA) nextstate = S0;
                else nextstate = S1;
            S1:      nextstate = S2;
            S2: if (TB) nextstate = S2;
                else nextstate = S3;
            S3:      nextstate = S0;
            default: nextstate = S0;
        endcase
    // output logic
    assign RA = (state == S2 | state == S3);
    assign YA = (state == S1);
    assign GA = (state == S0);
    assign RB = (state == S0 | state == S1);
    assign YB = (state == S3);
    assign GB = (state == S2);
```

```
endmodule
```

2. Sensor Module

```
module Traffic_sensor(T1, T2, clk, rst);
    output reg [4:0] T1, T2;
    input clk, rst;
    wire feedback1, feedback2;
    assign feedback1 = {(~(T1[4] ^ T1[3])),(~(T1[3] ^ T1[2]))};
    assign feedback2 = {(~(T1[4] ^ T1[3])),(~(T1[3] ^ T1[2]))};
    always @ (posedge clk, posedge rst)
        begin
            if (rst)
                begin
                    T1 = 5'b01101;
                    T2 = 5'b10110;
                end
            else
                begin
                    T1 = {T1[2:0],feedback1};
                    T2 = {T2[2:0],feedback2};
                end
            end
        end
    endmodule
```

1.2.2 TLC Test Bench

1. Controller Test Bench (without sensor)

```
'include "traffic_light_controller.vh"
module test_TLC();
    reg clk, rst;
    reg TA, TB;
    wire RA, YA, GA, RB, YB, GB;
    traffic_light_controller dut(TA,TB,clk,rst,RA,YA,GA,RB,YB,GB);
    always
        begin
            clk = 1; #5; clk = 0; #5;
        end
    initial begin
```

```

rst = 1; #10; rst = 0;
$display("Initially traffic in both lanes A and B");
TA = 1; TB = 1; #10;
$display("RA = %b, YA = %b, GA = %b", RA, YA, GA);
$display("RB = %b, YB = %b, GB = %b\n", RB, YB, GB);
#10;
$display("RA = %b, YA = %b, GA = %b", RA, YA, GA);
$display("RB = %b, YB = %b, GB = %b\n", RB, YB, GB);
$display("-----");

$display("Now traffic in A but not in B");
TA = 1; TB = 0; #10;
$display("RA = %b, YA = %b, GA = %b", RA, YA, GA);
$display("RB = %b, YB = %b, GB = %b\n", RB, YB, GB);
#10;
$display("RA = %b, YA = %b, GA = %b", RA, YA, GA);
$display("RB = %b, YB = %b, GB = %b\n", RB, YB, GB);
$display("-----");

$display("Now traffic in B but not in A");
TA = 0; TB = 1; #10;
$display("RA = %b, YA = %b, GA = %b", RA, YA, GA);
$display("RB = %b, YB = %b, GB = %b\n", RB, YB, GB);
#10;
$display("RA = %b, YA = %b, GA = %b", RA, YA, GA);
$display("RB = %b, YB = %b, GB = %b\n", RB, YB, GB);
$display("-----");

$display("Now traffic in neither");
TA = 0; TB = 0; #10;
$display("RA = %b, YA = %b, GA = %b", RA, YA, GA);
$display("RB = %b, YB = %b, GB = %b\n", RB, YB, GB);
#10;
$display("RA = %b, YA = %b, GA = %b", RA, YA, GA);
$display("RB = %b, YB = %b, GB = %b\n", RB, YB, GB);

$finish;
end
endmodule

```

2. Sensor Test Bench

```
'include "traffic_light_controller.vh"
module test_lfsr();
    reg clk, rst;
    wire [4:0] T;
    reg [4:0] index;
    initial
        begin
            index = 4'b0;
            clk = 0;
            rst = 1;
            #15;
            rst = 0;
            #200;
        end
    always
        begin
            #5;
            clk = ~clk;
            index = index + 1;
            $display("T = %b", T);
            if(index === 5'b11111) begin
                $finish;
            end
        end
    end
    Traffic_sensor dut(T,clk,rst);
endmodule
```

3. Controller Test Bench (with sensor)

```
'include "traffic_light_controller.vh"
module test_TLC();
    reg clk, rst;
    reg TA, TB;
    wire RA, YA, GA, RB, YB, GB;
    wire [4:0] A, B;
    Traffic_sensor input_string (A,B,clk,rst);
    traffic_light_controller dut(TA,TB,clk,rst,RA,YA,GA,RB,YB,GB);
    always
```

```

begin
    clk = 1; #5; clk = 0; #5;
end
initial begin
    rst = 1; #10; rst = 0;
    TA = A[0]; TB = B[0]; #10;
    $display("Input String from Traffic Sensors:");
    $display("TA = %b; TB = %b\n", A, B);
    $display("RA = %b, YA = %b, GA = %b", RA, YA, GA);
    $display("RB = %b, YB = %b, GB = %b\n", RB, YB, GB);
    #10;
    $display("RA = %b, YA = %b, GA = %b", RA, YA, GA);
    $display("RB = %b, YB = %b, GB = %b\n", RB, YB, GB);
    $display("-----");

    TA = A[1]; TB = B[1]; #10;
    $display("RA = %b, YA = %b, GA = %b", RA, YA, GA);
    $display("RB = %b, YB = %b, GB = %b\n", RB, YB, GB);
    #10;
    $display("RA = %b, YA = %b, GA = %b", RA, YA, GA);
    $display("RB = %b, YB = %b, GB = %b\n", RB, YB, GB);
    $display("-----");

    TA = A[2]; TB = B[2]; #10;
    $display("RA = %b, YA = %b, GA = %b", RA, YA, GA);
    $display("RB = %b, YB = %b, GB = %b\n", RB, YB, GB);
    #10;
    $display("RA = %b, YA = %b, GA = %b", RA, YA, GA);
    $display("RB = %b, YB = %b, GB = %b\n", RB, YB, GB);
    $display("-----");

    TA = A[3]; TB = B[3]; #10;
    $display("RA = %b, YA = %b, GA = %b", RA, YA, GA);
    $display("RB = %b, YB = %b, GB = %b\n", RB, YB, GB);
    #10;
    $display("RA = %b, YA = %b, GA = %b", RA, YA, GA);
    $display("RB = %b, YB = %b, GB = %b\n", RB, YB, GB);
    $display("-----");

    TA = A[4]; TB = B[4]; #10;

```



```

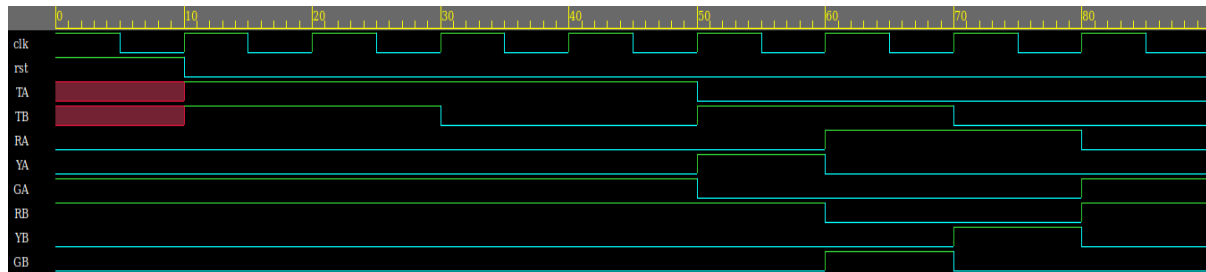
    $display("RA = %b, YA = %b, GA = %b", RA, YA, GA);
    $display("RB = %b, YB = %b, GB = %b\n", RB, YB, GB);
    #10;
    $display("RA = %b, YA = %b, GA = %b", RA, YA, GA);
    $display("RB = %b, YB = %b, GB = %b\n", RB, YB, GB);

    $finish;
end
endmodule

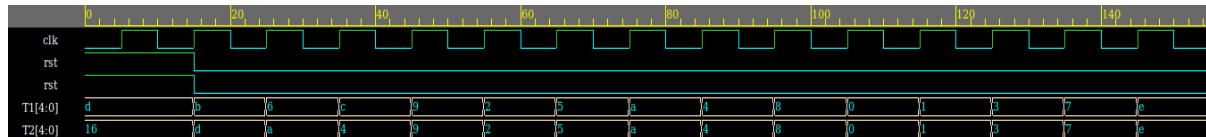
```

1.2.3 Timing

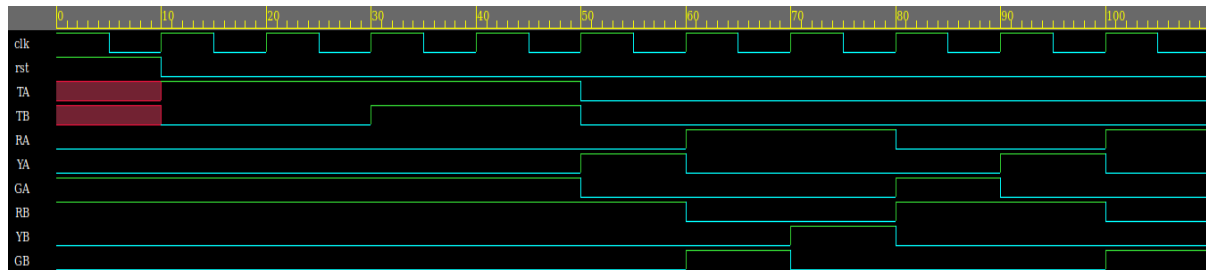
1. Waveform 1 (TLC without sensor)



2. Waveform 2 (sensor)



3. Waveform 3 (TLC with sensor)



1.2.4 Links

1. Waveform 1 (TLC without sensor)
Module
Waveform
2. Waveform 2 (sensor)
Module
Waveform
3. Waveform 3 (TLC with sensor)
Module
Waveform