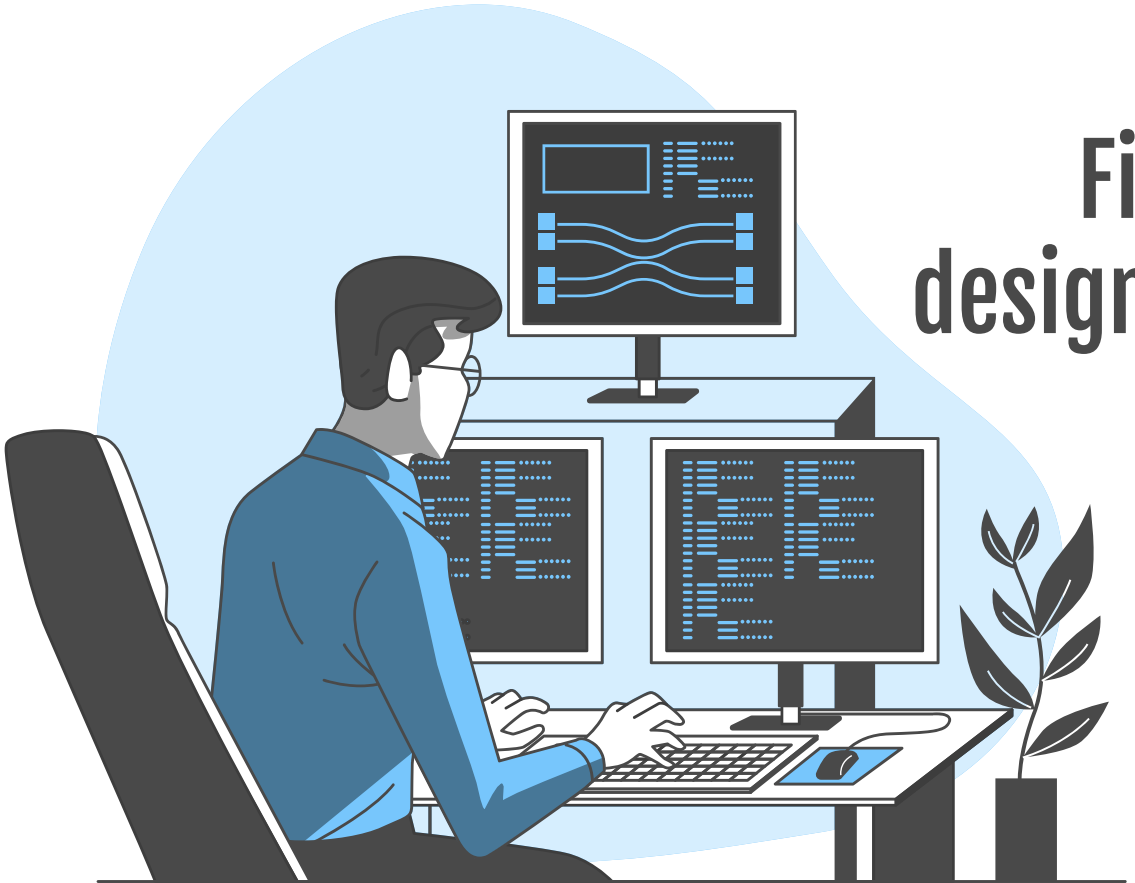
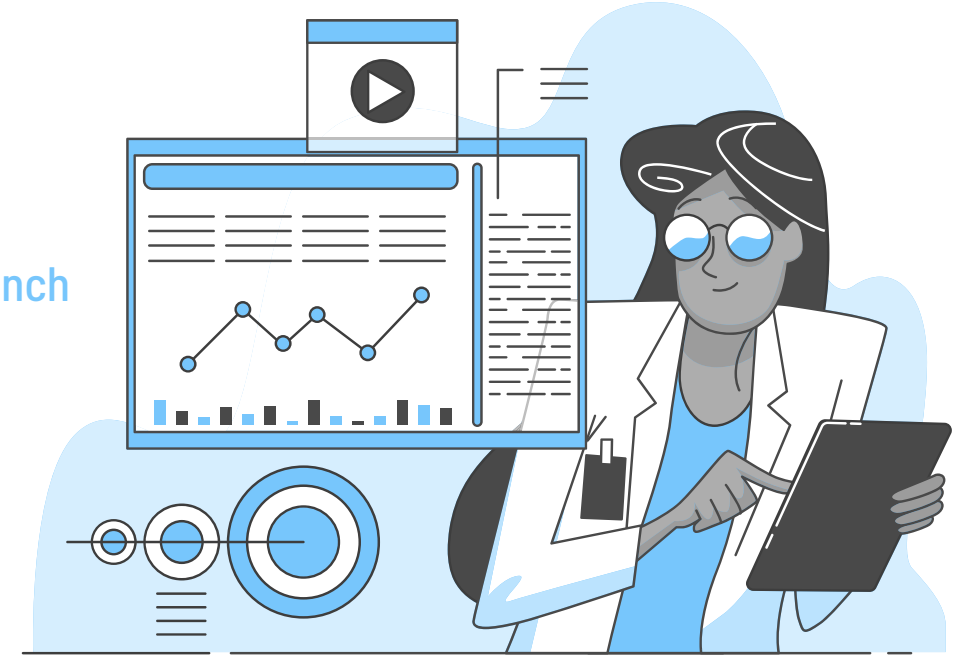


Final project/RTL design: 1. Ultrasound based distance measure

Adam Alakeli

Table of Contents

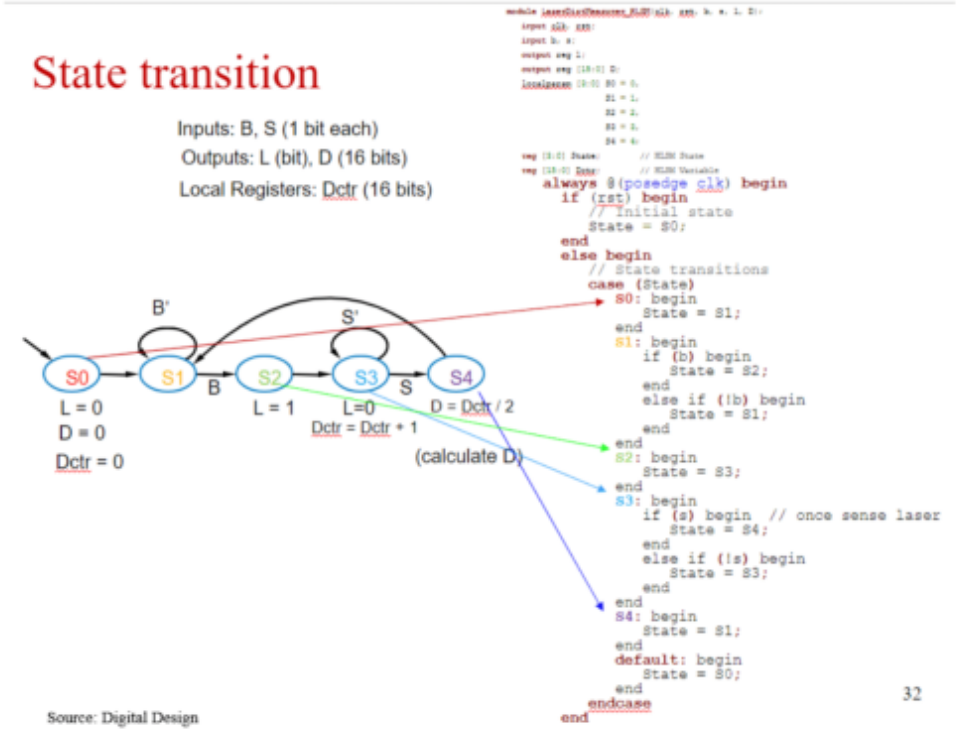


01
Introduction

- Objective: To create a synchronous sequential system using RTL (Register Transfer Level) design.
- Hardware: Utilize the Basys3 FPGA board.
- Inputs: Incorporate at least two different input methods (e.g., buttons, switches).
- Outputs: Implement at least two different output components, including a 7-segment display and an LED.
- Functionality: Develop a logic application that performs specific functions.

02
Design Steps

Make sure that you understand how to convert state diagram into Verilog code, see example below:



03
Verilog Design & TB

calcLaserD is the DUT

```
1  `timescale 1ns / 1ps
2
3  module calcLaserD(
4      input clk,
5      input reset,
6      input startBtn,
7      input stopBtn,
8      output reg [15:0] D, // OUTPUT D
9      output [3:0] an,  // OUTPUT
10     output [6:0] seg
11     );
12
13     localparam [3:0] s0 = 0,
14                      s1 = 1,
15                      s2 = 2,
16                      s3 = 3,
17                      s4 = 4;
18
19     reg [3:0] State;     // HLSM State
20     reg [15:0] Dctr;     // HLSM VARIABLE
21 //    reg [15:0] D;
22     reg [3:0] hundreds;
23     reg [3:0] tens;
24     reg [3:0] ones;
25     reg [3:0] temp;
26     reg l;
27     wire second_elapsed;
28
29     disp_mux uut(.clk(clk), .reset(reset), .in2(hundreds), .in1(tens), .in0(ones), .an(an), .seg(seg));
30
31     second sec(.Y(second_elapsed), .clk(clk), .reset(stopBtn));
32
```

```
32
33      //Add N-bit register code for q_reg here:
34      always @(posedge clk) begin
35          if (reset) begin
36              // Reset logic
37              State <= s0;
38              l <= 0;
39              D <= 0;
40              Dctr <= 0;
41              hundreds <= 0;
42              tens <= 0;
43              ones <= 0;
44          end
45          else begin
46              case (State)
47                  s0: begin
48                      State = s1;
49                  end
50
51                  s1: begin
52                      if (startBtn) begin
53                          State = s2;
54                      end
55                      else if(!startBtn) begin
56                          State = s1;
57                      end
58                  end
59
60                  s2: begin
61                      State = s3;
62                  end
63
64                  s3: begin
65                      if (stopBtn) begin // once sense laser
66                          State = s4;
67                      end
68                      else if (!stopBtn) begin
69                          State = s3;
70                      end
```

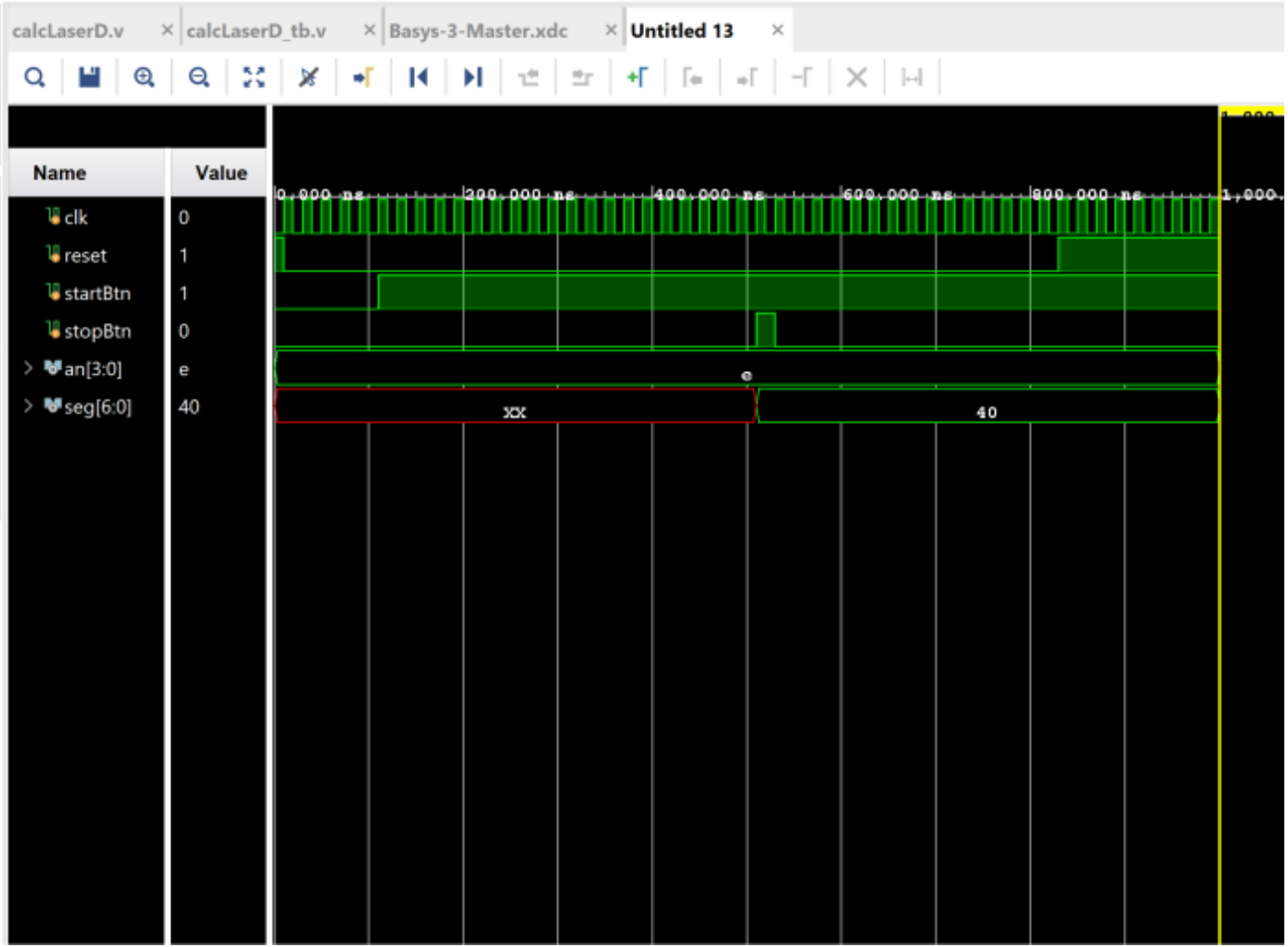
```
                end

                s4: begin
                    State = s1;
                end

                default: begin
                    State = s0;
                end
            endcase
        end

    case (State)
        s0: begin
            l = 0;
            D = 0;
            Dctr = 0;
        end

        s1: begin
            l = 0;
        end

        s2: begin
            l = 1;
        end

        s3: begin
            l = 0;
            if (second_elapsed) begin
                Dctr = Dctr + 1;
            end
        end

```

```
            D = 0;
            Dctr = 0;
        end

        s1: begin
            l = 0;
        end

        s2: begin
            l = 1;
        end

        s3: begin
            l = 0;
            if (second_elapsed) begin
                Dctr = Dctr + 1;
            end
        end

        s4: begin
            D = Dctr >> 1;
            // Split D into hundred tens and ones
            hundreds = D /100;
            tens = (D-hundreds)/10;
            ones = D-(100*hundreds)-(10*tens);
        end

        default: begin
            l = 0;
            D = 0;
            Dctr = 0;
        end
    endcase

end


endmodule

```

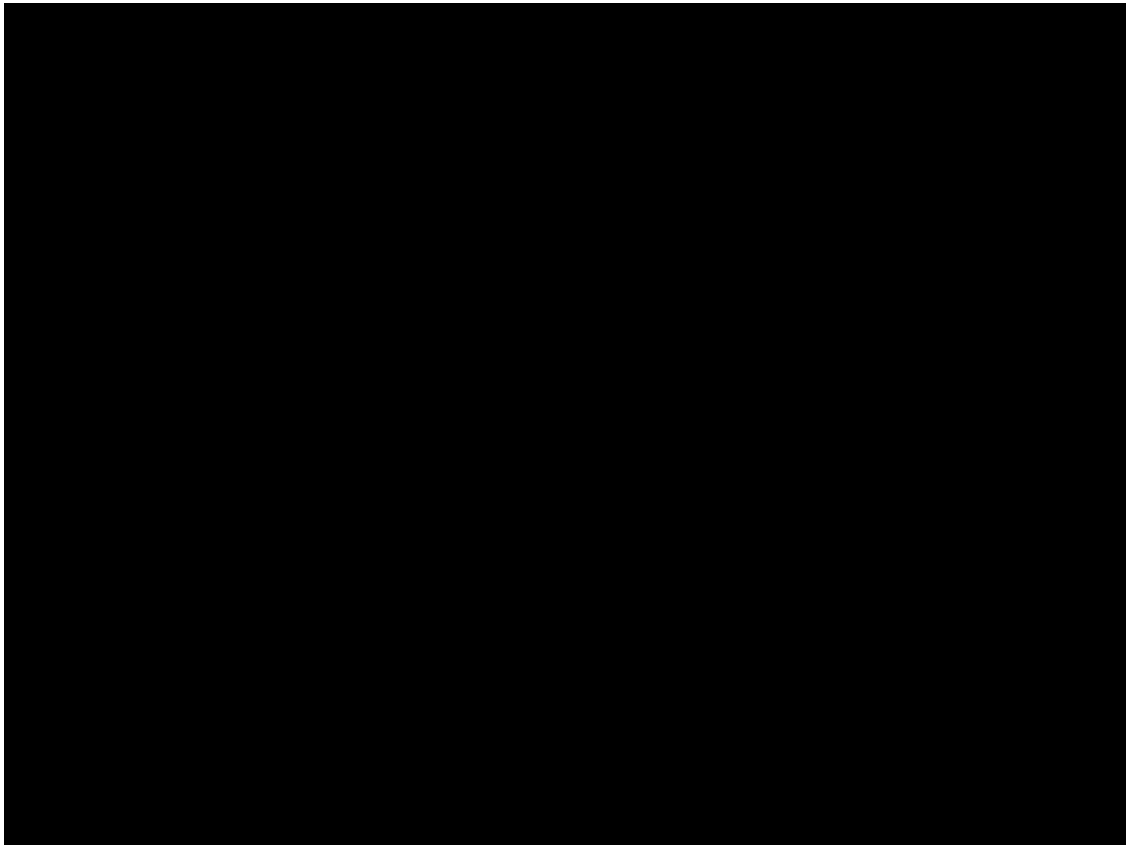
Testbench

```
`timescale 1ns / 1ps

module calcLaserD_tb;

    // Inputs
    reg clk;
    reg reset;
    reg startBtn;
    reg stopBtn;
    reg [15:0] D;

    // Outputs
    wire [3:0] an;
    wire [6:0] seg;

    // Instantiate the Unit Under Test (UUT)
    calcLaserD uut_tb(
        .clk(clk),
        .reset(reset),
        .startBtn(startBtn),
        .stopBtn(stopBtn),
        .an(an), .seg(seg),
        .D(D)
    );

    // Clock generation
    initial begin
        clk = 0;
        forever #10 clk = ~clk; // Generate a clock with a period of 20ns
    end

    // Test procedure
    initial begin
        // Initialize Inputs
        reset = 1;
        startBtn = 0;
        stopBtn = 0;

```

```
31
32      // Test procedure
33      initial begin
34          // Initialize Inputs
35          reset = 1;
36          startBtn = 0;
37          stopBtn = 0;
38
39          // Wait for global reset
40          #10;
41          reset = 0;
42
43          // Add stimulus here
44          // Example: simulate pressing start and stop buttons
45          #100;
46          startBtn = 1;
47          #200;
48          // Wait some time
49          #200;
50          stopBtn = 1;
51          #20;
52          stopBtn = 0;
53          #300
54          reset = 1;
55          #100;
56          startBtn = 1;
57          #20;
58          stopBtn = 0;
59
60          // Add more tests as needed
61
62          // Finish the simulation
63          #500;
64          $finish;
65      end
66
67  endmodule
68
```


calcLaserD.v
calcLaserD_tb.v
Basys-3-Master.xdc
Untitled 13
Name
Value
clk
0
reset
1
startBtn
1
stopBtn
0
an[3:0]
e
seg[6:0]
40
0.000 ns
200.000 ns
400.000 ns
600.000 ns
800.000 ns
1,000.
e
XX
40

Video Time!



Thank You!