SIMPE VOTING MACHINE

```
BUTTON CONTROL
     `timescale 1ns / 1ps
  4 | input clock,
 5 | input reset,
  6 input button,
  7 output reg valid_vote
  8 : );
  9
 10 | reg [30:0] counter;
 11
 12 : //1 sec / 10ms = 100000000
 13
 14 

□ always @ (posedge clock)
 15 ⊝ begin
16  if (reset)
 17
            counter <= 0;
18 :
       else
 19 🖯
       begin
 20 🖯
            if (button & counter < 100000001)
 21
                counter <= counter + 1;
 22 🖯
           else if(!button)
 23 🖨
               counter <= 0;
24 🖨 end
25 🖨 end
 26
 27
 28 palways @(posedge clock)
 29 🖯 begin
30 ⊖ if(reset)
 31
             valid vote <= 1'b0;</pre>
 32
        else
 33 🖨
       begin
 34 ⊖
            if(counter == 100000000)
35 ¦
                 valid_vote <= 1'b1;</pre>
 36
            else
 37 🖨
                valid vote <= 1'b0;</pre>
 39 🖒 end
 40
 41 \(\hat{\text{\text{o}}}\) endmodule
 42
```

VOTE LOGGER

```
`timescale 1ns / 1ps
 2 module voteLogger(
 3 | input clock,
 4 | input reset,
 5 input mode,
 6 input cand1_vote_valid,
 7 input cand2_vote_valid,
 8 input cand3_vote_valid,
 9 | input cand4_vote_valid,
10 output reg [7:0] cand1_vote_recvd,
11
    output reg [7:0] cand2_vote_recvd,
    output reg [7:0] cand3_vote_recvd,
12
13
    output reg [7:0] cand4_vote_recvd
14
15
16
17
    always @(posedge clock)
18
    begin
19
        if(reset)
20
        begin
21
            cand1_vote_recvd <= 0;</pre>
22
           cand2_vote_recvd <= 0;
23
           cand3_vote_recvd <= 0;
24
            cand4_vote_recvd <= 0;
25
        end
25 ¦
        end
26
        else
27
        begin
28
            if(cand1_vote_valid & mode==0)
29
                 cand1 vote recvd <= cand1 vote recvd + 1;
30
            else if(cand2_vote_valid & mode==0)
31
                 cand2_vote_recvd <= cand2_vote_recvd + 1;</pre>
32
            else if(cand3_vote_valid & mode==0)
33
                cand3_vote_recvd <= cand3_vote_recvd + 1;</pre>
34
            else if(cand4 vote valid & mode==0)
35
                 cand4_vote_recvd <= cand4_vote_recvd + 1;</pre>
36
        end
37 end
38
39
40
   endmodule
41
```

MODE CONTROL

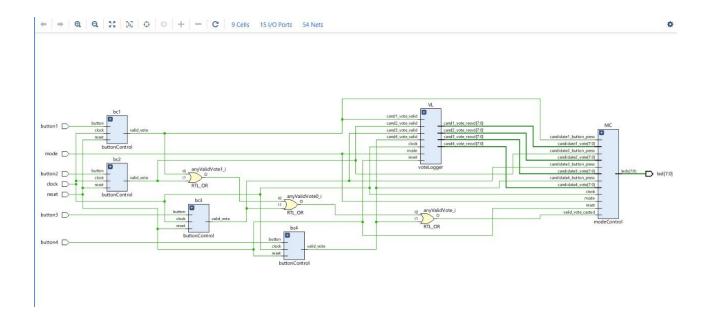
```
1 'timescale 1ns / 1ps
3 ⊕ module modeControl(
4 | input clock,
5 input reset,
 6 input mode,
    input valid_vote_casted,
8 input [7:0] candidate1_vote,
    input [7:0] candidate2_vote,
9 :
10 input [7:0] cand input [7:0] ote,
11 input [7:0] cand input [7:0] ote,
    input candidate1_button_press,
13 input candidate2_button_press,
14
    input candidate3_button_press,
15
    input candidate4_button_press,
16
    output reg [7:0] leds
17
       );
19 reg [30:0] counter;
20
21 \ominus always @(posedge clock)
22 🖯 begin
23 👨
24
            counter <= 0;  //Whenever reset is pressed, counter started from 0</pre>
25 🖨
      else if(valid vote casted) //If a valid vote is casted, counter becomes 1
25 🖨
     else if (valid vote casted) //If a valid vote is casted, counter becomes 1
26
           counter <= counter + 1;
27 🖯
       else if(counter !=0 & counter < 100000000)//If counter is not 0, increment it til
28
           counter <= counter + 1;
        else //Once counter becomes 100000000, reset it to zero
30 ♠
           counter <= 0;
31 🖨 end
32 ;
33 always @(posedge clock)
34 ⇔ begin
      if(reset)
36
           leds <= 0;
37 ¦
       else
38 🖨
       begin
          if (mode == 0 & counter > 0 ) //mode0 -> voting mode, mode 1 -> result mode
39 ⊜
               leds <= 8'hFF;</pre>
40 :
41 ⊖
          else if(mode == 0)
              leds <= 8'h00;
42
43 ₽
           else if(mode == 1) //result mode
44 🖯
           begin
45 ⊜
               if (candidate1 button press)
46
                   leds <= candidate1 vote;
47 🖯
              else if(candidate2 button press)
47 🖨
                   else if (candidate2 button press)
48
                        leds <= candidate2_vote;</pre>
49 ⊜
                    else if (candidate3_button_press)
50
                         leds <= candidate3 vote;</pre>
51 👨
                    else if(candidate4 button press)
52 🖒
                         leds <= candidate4 vote;</pre>
53 🖒
               end
54 🖨
          end
55   end
56
57 △ endmodule
```

```
TOP MODULE VOTING MACHINE
    `timescale 1ns / 1ps
 3 module votingMachine(
 4 input clock,
 5 input reset,
 6 input mode,
    input button1,
 8
    input button2,
 9 input button3,
10 input button4,
11 output [7:0] led
12 );
13
14 wire valid_vote_1;
15 wire valid vote 2;
16 wire valid_vote_3;
17 wire valid_vote_4;
18
    wire [7:0] cand1_vote_recvd;
19 wire [7:0] cand2_vote_recvd;
20 wire [7:0] cand3 vote recvd;
21 | wire [7:0] cand4 vote recvd;
22 wire anyValidVote;
23
    assign anyValidVote = valid_vote_1|valid_vote_2|valid_vote_3|valid_vote_4;
25
 25
 26 buttonControl bc1(
 27 : .clock(clock),
 28 : .reset (reset),
 29 .button(button1),
 30
     .valid vote(valid vote 1)
 31
     );
 32
 33 | buttonControl bc2(
 34 .clock(clock),
 35 | .reset(reset),
 36
     .button(button2),
 37
     .valid vote(valid vote 2)
 38
     );
 39
 40 buttonControl bc3(
 41 : .clock(clock),
 42
     .reset(reset),
 43
     .button(button3),//
 44
     .valid_vote(valid_vote_3)
 45
     );
 46
 47 | buttonControl bc4(
 48 .clock(clock),
 49 .reset (reset),
```

TOP MODULE VOTING MACHINE

```
49 .reset(reset),
50 .button(button4),
51
    .valid vote(valid vote 4)
52
53
54
55
56 | voteLogger VL(
57 | .clock(clock),
58 : reset (reset),
59 .mode (mode),
    .cand1_vote_valid(valid_vote 1),
 61 | .cand2 vote valid(valid vote 2),
62 | .cand3 vote valid(valid vote 3),
63
    .cand4 vote valid(valid vote 4),
    .cand1_vote_recvd(cand1_vote recvd),
64
65
   .cand2_vote_recvd(cand2_vote_recvd),
66 : .cand3 vote recvd(cand3 vote recvd),
67
    .cand4_vote_recvd(cand4_vote_recvd)
68
 69
70
71 | modeControl MC(
72
    .clock(clock),
    .reset(reset),
73
 74 .mode (mode),
 75
    .valid vote casted(anyValidVote),
 76 : .candidate1 vote(cand1 vote recvd),
 77 | .candidate2 vote(cand2 vote recvd),
 78
    .candidate3 vote(cand3 vote recvd),
 79 .candidate4 vote(cand4 vote recvd),
 80
    .candidate1 button press(valid vote 1),
 81
    .candidate2 button press(valid vote 2),
     .candidate3 button press(valid vote 3),
 82
 83
     .candidate4 button press(valid vote 4),
 84
    .leds(led)
 85
          );
 86
 87 : endmodule
```

• RTL SCHEMATIC

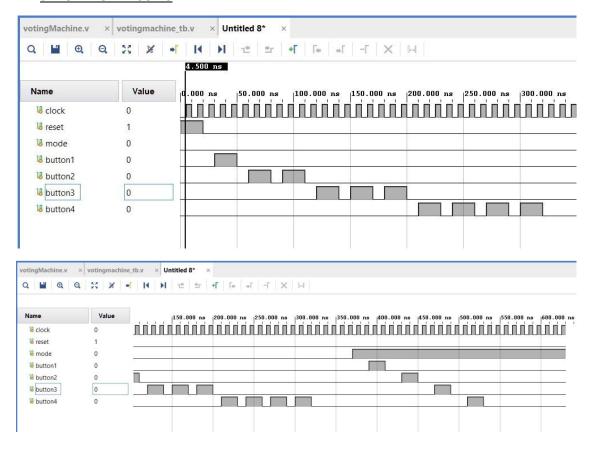


• <u>TESTBENCH</u>

```
module votingMachine tb;
 3
 4
        // Inputs
 5
 6
        reg clock;
 7
        reg reset;
 8
        reg mode;
 9
        reg button1;
10
        reg button2;
11
        reg button3;
12
        reg button4;
13
        // Outputs
14
        wire [7:0] led;
15
16
         // Instantiate the Unit Under Test (UUT)
17
        votingMachine uut (
18
19
             .clock(clock),
             .reset (reset),
20
             .mode (mode),
21
22
             .button1 (button1),
23
             .button2 (button2),
24
             .button3 (button3),
25
             .button4 (button4),
```

```
26
            .led(led)
27
        );
28
29
        // Clock generation
30
        initial begin
           clock = 0;
31
            forever #5 clock = ~clock; // 10ns clock period
32
33
        end
34
35
        // Stimulus generation
        initial begin
36
37
           // Initialization
           reset = 1;
38
39
            mode = 0;
           button1 = 0:
40
41
            button2 = 0;
42
            button3 = 0;
43
            button4 = 0;
44
45
           #20 reset = 0; // Release reset
46
47
            // Voting phase
48
            #10 button1 = 1; // Candidate A gets 1 vote
49
            #20 button1 = 0;
50
51
           #10 button2 = 1; // Candidate B gets 2 votes
52
          #20 \text{ button2} = 0;
53
           #10 button2 = 1;
54
           #20 button2 = 0;
55
56
          #10 button3 = 1; // Candidate C gets 3 votes
57
           #20 button3 = 0;
           #10 button3 = 1;
58
59
          #20 button3 = 0;
60
           #10 button3 = 1;
           #20 button3 = 0;
61
62
           #10 button4 = 1; // Candidate D gets 4 votes
63
64
           #20 button4 = 0;
65
          #10 button4 = 1;
66 :
           #20 button4 = 0;
67
           #10 button4 = 1;
68
           #20 button4 = 0;
69
           #10 button4 = 1;
70
           #20 button4 = 0;
71
72
           // Switch to result mode
73
           #50 \mod = 1;
74
75
             // Show votes for each candidate
            #20 button1 = 1; // Display votes for Candidate A
76
77
            #20 button1 = 0;
78
79
            #20 button2 = 1; // Display votes for Candidate B
80
            #20 button2 = 0;
81
82
            #20 button3 = 1; // Display votes for Candidate C
83
            #20 button3 = 0;
84
85
            #20 button4 = 1; // Display votes for Candidate D
86
             #20 button4 = 0;
87
             // End simulation
88
             #100 $finish;
89
90 🖨
         end
91
92 🖒 endmodule
```

• SIMULATION RESULTS



• HARDWARE RESULTS

CANDIDATE	NO. OF VOTES (INPUT)
А	1
В	2
С	3
D	4

