

SIMPE VOTING MACHINE

BUTTON CONTROL

```
1  `timescale 1ns / 1ps
2
3  module buttonControl(
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     always @(posedge clock)
29     begin
30         if(reset)
31             valid_vote <= 1'b0;
32         else
33             begin
34                 if(counter == 100000000)
35                     valid_vote <= 1'b1;
36                 else
37                     valid_vote <= 1'b0;
38             end
39     end
40
41 endmodule
42
```

VOTE LOGGER

```
1  `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,
12     output reg [7:0] cand3_vote_recvd,
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;
22         cand2_vote_recvd <= 0;
23         cand3_vote_recvd <= 0;
24         cand4_vote_recvd <= 0;
25     end
26
27     end
28     else
29     begin
30         if(cand1_vote_valid & mode==0)
31             cand1_vote_recvd <= cand1_vote_recvd + 1;
32         else if(cand2_vote_valid & mode==0)
33             cand2_vote_recvd <= cand2_vote_recvd + 1;
34         else if(cand3_vote_valid & mode==0)
35             cand3_vote_recvd <= cand3_vote_recvd + 1;
36         else if(cand4_vote_valid & mode==0)
37             cand4_vote_recvd <= cand4_vote_recvd + 1;
38     end
39 end
40
41 endmodule
```

MODE CONTROL

```
1  `timescale 1ns / 1ps
2
3  module modeControl(
4  input clock,
5  input reset,
6  input mode,
7  input valid_vote_casted,
8  input [7:0] candidatel_vote,
9  input [7:0] candidate2_vote,
10 input [7:0] candidate3_vote,
11 input [7:0] candidate4_vote,
12 input candidatel_button_press,
13 input candidate2_button_press,
14 input candidate3_button_press,
15 input candidate4_button_press,
16 output reg [7:0] leds
17 );
18
19 reg [30:0] counter;
20
21 always @(posedge clock)
22 begin
23     if(reset)
24         counter <= 0;    //Whenever reset is pressed, counter started from 0
25     else if(valid_vote_casted) //If a valid vote is casted, counter becomes 1
26
27     else if(valid_vote_casted) //If a valid vote is casted, counter becomes 1
28         counter <= counter + 1;
29     else if(counter !=0 & counter < 1000000000) //If counter is not 0, increment it til
30         counter <= counter + 1;
31     else //Once counter becomes 1000000000, reset it to zero
32         counter <= 0;
33 end
34
35 always @(posedge clock)
36 begin
37     if(reset)
38         leds <= 0;
39     else
40     begin
41         if(mode == 0 & counter > 0 ) //mode0 -> voting mode, mode 1 -> result mode
42             leds <= 8'hFF;
43         else if(mode == 0)
44             leds <= 8'h00;
45         else if(mode == 1) //result mode
46         begin
47             if(candidate1_button_press)
48                 leds <= candidatel_vote;
49             else if(candidate2_button_press)
50                 leds <= candidate2_vote;
51             else if(candidate3_button_press)
52                 leds <= candidate3_vote;
53             else if(candidate4_button_press)
54                 leds <= candidate4_vote;
55         end
56     end
57 end
58 endmodule
```

TOP MODULE VOTING MACHINE

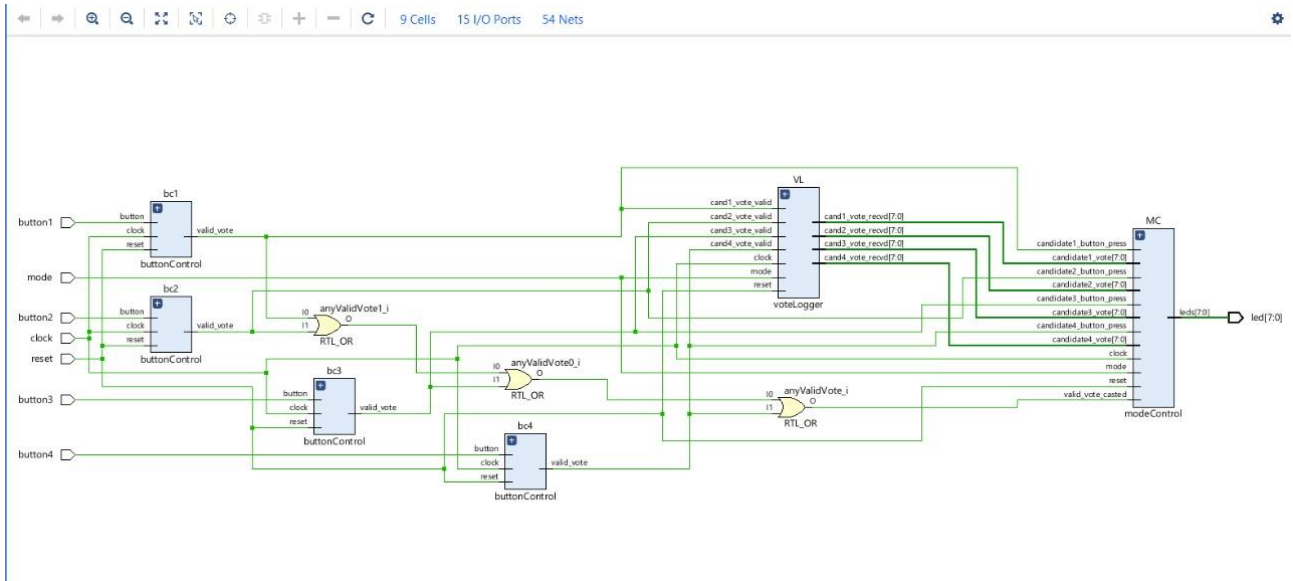
```
1  `timescale 1ns / 1ps
2
3  module votingMachine(
4  input clock,
5  input reset,
6  input mode,
7  input button1,
8  input button2,
9  input button3,
10 input button4,
11 output [7:0] led
12 );
13
14 wire valid_vote_1;
15 wire valid_vote_2;
16 wire valid_vote_3;
17 wire valid_vote_4;
18 wire [7:0] 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
24 assign anyValidVote = valid_vote_1|valid_vote_2|valid_vote_3|valid_vote_4;
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
```

```
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),
60  .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),
64  .cand1_vote_recvd(cand1_vote_recvd),
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),
73  .reset(reset),
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),
82  .candidate3_button_press(valid_vote_3),
83  .candidate4_button_press(valid_vote_4),
84  .leds(led)
85  );
86
87  endmodule
--
```

- RTL SCHEMATIC



- TESTBENCH

```

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

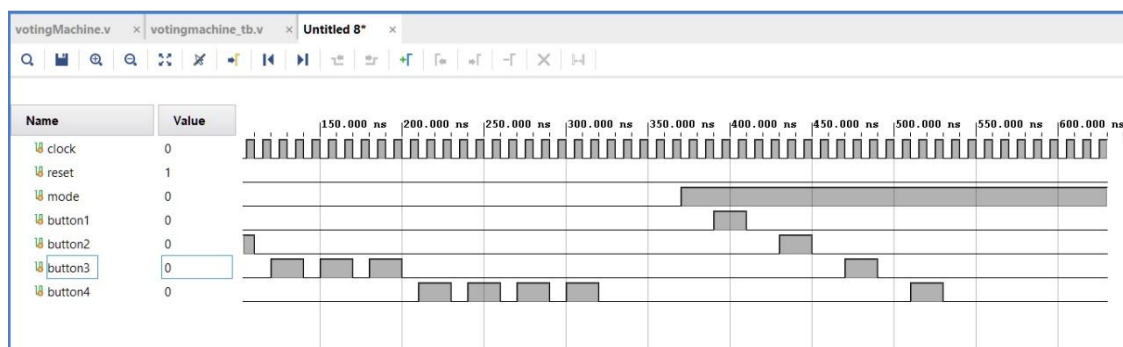
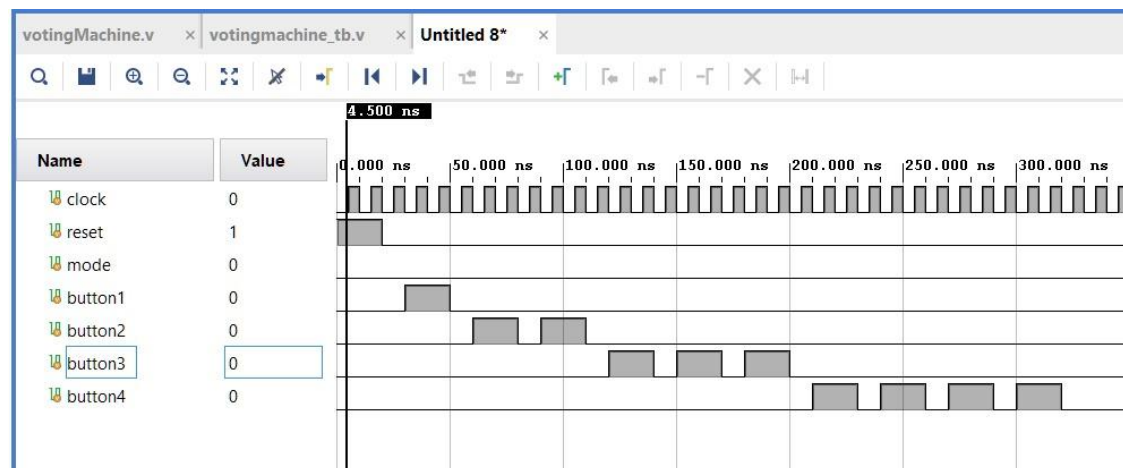
```

```

26         .led(led)
27     );
28
29     // Clock generation
30     initial begin
31         clock = 0;
32         forever #5 clock = ~clock; // 10ns clock period
33     end
34
35     // Stimulus generation
36     initial begin
37         // Initialization
38         reset = 1;
39         mode = 0;
40         button1 = 0;
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 button2 = 0;
53         #10 button2 = 1;
54         #20 button2 = 0;
55
56         #10 button3 = 1; // Candidate C gets 3 votes
57         #20 button3 = 0;
58         #10 button3 = 1;
59         #20 button3 = 0;
60         #10 button3 = 1;
61         #20 button3 = 0;
62
63         #10 button4 = 1; // Candidate D gets 4 votes
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 mode = 1;
74
75         // Show votes for each candidate
76         #20 button1 = 1; // Display votes for Candidate A
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
88         // End simulation
89         #100 $finish;
90     end
91
92 endmodule

```

- SIMULATION RESULTS**



- HARDWARE RESULTS**

CANDIDATE	NO. OF VOTES (INPUT)
A	1
B	2
C	3
D	4

