RVM Machine

RVM Machine has the following file with following hierarchy.

- RVM_wrapper
  - RVM
    - Clk_div
    - FSM
    - Bottle_counter
    - Can_counter
    - display

RVM _wrapper:

```
module RVM_wrapper
  (AF,
   BF,
   LED,
   SF,
   anodes,
   cathodes,
   clk,
   reset,
   sensor);
 output AF;
 output BF;
 output LED;
 output SF;
 output [3:0]anodes;
 output [6:0]cathodes;
 input clk;
 input reset;
 input [3:0]sensor;

 wire AF;
 wire BF;
 wire LED;
 wire SF;
 wire [3:0]anodes;
 wire [6:0]cathodes;
 wire clk;
 wire reset;
 wire [3:0]sensor;

 RVM RVM_i
    (.AF(AF),
     .BF(BF),
     .LED(LED),
```

```
      .SF(SF),
      .anodes(anodes),
      .cathodes(cathodes),
      .clk(clk),
      .reset(reset),
      .sensor(sensor));
endmodule
```

RVM:

```
module RVM
  (AF,
   BF,
   LED,
   SF,
   anodes,
   cathodes,
   clk,
   reset,
   sensor);
  output AF;
  output BF;
  output LED;
  output SF;
  output [3:0]anodes;
  output [6:0]cathodes;
  input clk;
  input reset;
   input [3:0]sensor;

  wire FSM_0_AF;
  wire FSM_0_BC;
  wire FSM_0_BF;
  wire FSM_0_CC;
  wire FSM_0_LED;
  wire FSM_0_SF;
  wire [2:0]FSM_0_data_out;
  wire [3:0]bottle_counter_0_bot_counter;
  wire [3:0]can_counter_0_can_counter;
  wire clk_1;
  wire clk_div_0_clk_div;
  wire [3:0]display_0_anodes;
  wire [6:0]display_0_cathodes;
  wire reset_1;
  wire [3:0]sensor_1;

  assign AF = FSM_0_AF;
```

```
  assign BF = FSM_0_BF;
  assign LED = FSM_0_LED;
  assign SF = FSM_0_SF;
  assign anodes[3:0] = display_0_anodes;
  assign cathodes[6:0] = display_0_cathodes;
  assign clk_1 = clk;
  assign reset_1 = reset;
  assign sensor_1 = sensor[3:0];
  FSM FSM_0
     (.AF(FSM_0_AF),
      .BC(FSM_0_BC),
      .BF(FSM_0_BF),
      .CC(FSM_0_CC),
      .LED(FSM_0_LED),
      .SF(FSM_0_SF),
      .clk(clk_div_0_clk_div),
      .data_out(FSM_0_data_out),
      .reset(reset_1),
      .sensor(sensor_1));
  bottle_counter bottle_counter_0
     (.BC(FSM_0_BC),
      .bot_counter(bottle_counter_0_bot_counter),
      .clk(clk_div_0_clk_div),
      .reset(reset_1));
  can_counter can_counter_0
     (.CC(FSM_0_CC),
      .can_counter(can_counter_0_can_counter),
      .clk(clk_div_0_clk_div),
      .reset(reset_1));
  clk_div clk_div_0
     (.clk(clk_1),
      .clk_div(clk_div_0_clk_div),
      .clk_reset(reset_1));
  display display_0
     (.anodes(display_0_anodes),
      .bottle_counter(bottle_counter_0_bot_counter),
      .can_counter(can_counter_0_can_counter),
      .cathodes(display_0_cathodes),
      .clk(clk_1),
      .clk_div(clk_div_0_clk_div),
      .data_out(FSM_0_data_out),
      .reset(reset_1));
endmodule
```

Clk_Div:

```
module clk_div#
```

```
(
   parameter REQUIRED_CLK = 250000 //2 Hz, Supposing base clock as 50Mhz
)
(
   input clk,
   input clk_reset,
   output wire clk_div
//         output reg done_clk
   );
 reg [31:0]divider_clk_counter = 32'd0;
 reg [3:0] done_counter = 4'd0;
 reg clk_reset_det = 0;
 reg clk_inv = 0;

 always@(posedge clk)
   begin
      clk_reset_det <= clk_reset;
                  if (clk_reset)
                        begin
                              done_counter <= 0;
                              divider_clk_counter      <= 0;
                              clk_inv <= 0;
                        end
                  else
                        begin
                              if (divider_clk_counter < REQUIRED_CLK -1)
                                    begin
                                          divider_clk_counter <= divider_clk_counter + 1;
                                    end
                              else
                                    begin
                                          if(done_counter > 5)
                                                begin
//                                                      done_clk <= 1;
                                                      done_counter <= 0;
                                                end
                                          else
                                                begin
                                                      divider_clk_counter <= 0;
                                                      done_counter <= done_counter + 1;
                                                      clk_inv = ~clk_inv;
//                                                      done_clk <= 0;
                                                end
                                    end
                        end
   end
assign clk_div = clk_inv;
endmodule
```

FSM:

```
module FSM
(
   input clk,
   input reset,
   input [3:0]sensor,
   output reg AF,
   output reg SF,
   output reg BF,
   output reg [2:0]data_out,
   output wire BC,
   output wire CC,
   output wire LED
//         output reg clk_div_in
   );

reg [3:0]state = 4'd0;
reg LED_ON = 0;
reg [9:0]state_counter = 0;
reg [1:0] data_counter = 2'd0;
reg [2:0] can_counter  = 3'd0;
reg [2:0] bottle_counter  = 3'd0;
reg [2:0] Error_counter  = 3'd0;
reg [3:0] LED_Counter  = 3'd0;

reg BCC = 1'b0;
reg CCC = 1'b0;
reg LEDD = 1'b0;

always@(posedge clk)
   begin
      if (reset)
         begin
            state <= 4'd0;
         end
      else
         begin
            case(state)
               4'b0000: //IDLE
                  begin
                     BF <= 1'b1;
                     SF <= 1'b1;
                     AF <= 1'b0;
                     data_out <= 3'd0;
                     if (sensor == 4'd1)
                        begin
                           state <= 4'b0001;
```

```
                    end
                  else
                    begin
                      state <= 4'd0;
                    end
                end
              4'b0001: // DETECTED
                begin
                  SF <= 1'b1;
                  AF <= 1'b1;
                  BF <= 1'b1;
                  data_out <= 3'd1;
                                                    if (data_counter > 1)
                                                          begin
                                                                data_counter <= 0;
                                                                if (sensor == 4'b0011)
                                                                      begin
                                                                            state <=
4'b0011;
                                                                      end
                                                                else
                                                                      if (sensor == 4'b0101)
                                                                            begin
                                                                                  state
<= 4'b0101;
                                                                            end
                                                                      else
                                                                            begin
                                                                                  if
(sensor == 4'b1001)

          begin

                    state <= 4'b1001;

          end
                                                                                        else


          begin

                    state <= 4'b0001;

          end
                                                                            end
                                                          end
                                                    else
                                                          begin
```

```
                                                                    data_counter <=
data_counter + 1;
                                                                end
            end
          4'b0011: // CAN DETECTED
            begin
              SF <= 1'b0;
              AF <= 1'b1;
              BF <= 1'b1;
              data_out <= 3'd2;
                                                        if (can_counter > 3)
                                                                begin
                                                                    can_counter <= 0;
                                                                    CCC <= 1;
                                                                    state <= 4'b1010;
                                                                end
                                                        else
                                                                begin
                                                                    can_counter <= can_counter
+ 1;
                                                                end
            end
          4'b0101: // BOTTLE DETECTED
            begin
              SF <= 1'b0;
              AF <= 1'b1;
              BF <= 1'b0;
              data_out <= 3'd3;
              if (bottle_counter > 3)
                begin
                  bottle_counter <= 0;
                  state <= 4'b1010;
                                                                    BCC <= 1;
                end
              else
                begin
                  bottle_counter <= bottle_counter + 1;
                end
            end
          4'b1001: // Error Detection
            begin
              SF <= 1'b1;
              AF <= 1'b0;
              data_out <= 3'd4;
              if (Error_counter > 3)
                begin
                  Error_counter <= 0;
                  state <= 4'b1010;
```

```
                    end
                  else
                    begin
                      Error_counter <= Error_counter + 1;
                    end
                end
              4'b1010: // END
                begin
                  data_out <= 3'd5;
                  BCC <= 0;
                  CCC <= 0;
                                                          LED_ON <= 1;
//                                                        clk_div_in <= 1;
                                                        if (LED_Counter > 5)
                                                                begin
                                                                        state <= 4'b0000;
                                                                        LED_Counter <= 0;
                                                                        LEDD <= 0;
                                                                end
                                                        else
                                                                begin
                                                                        LED_Counter <=
LED_Counter + 1;
                                                                        LEDD = ~LEDD;
                                                                end
                end
              default:
                state <= 4'd0;
            endcase
          end
    end

    assign BC = BCC;
    assign CC = CCC;
    assign LED = LEDD;
endmodule
```

Can_Counter:

```
module can_counter(
    input clk,
    input reset,
    input CC,
    output reg [3:0]can_counter = 8'd0
    );
```

```
 reg CC_det;

 always@(posedge clk)
   begin
     if(reset)
       begin
         can_counter <= 0;
       end
     else
       begin
         CC_det <= CC;
         if (CC)
           begin
             if (can_counter > 9)
               begin
                 can_counter <= 0;
               end
             else
               begin
                 can_counter <= can_counter + 1;
               end
           end
         else
           can_counter <= can_counter;
       end
   end
endmodule
```

Bottle_Counter:

```
module bottle_counter(
   input clk,
   input reset,
   input BC,
   output reg [3:0]bot_counter = 8'd0
   );

 reg BC_det;

 always@(posedge clk)
   begin
     if(reset)
       begin
         bot_counter <= 0;
       end
     else
       begin
```

```
            BC_det <= BC;
            if (BC)
               begin
                  if (bot_counter > 9)
                     begin
                        bot_counter <= 0;
                     end
                  else
                     begin
                        bot_counter <= bot_counter + 1;
                     end
               end
             else
               bot_counter <= bot_counter;
         end
   end
endmodule
```

Display:

```
module display(
            input clk,
            input clk_div,
            input reset,
            input [2:0]data_out,
            input [3:0]can_counter,
            input [3:0]bottle_counter,
            output reg [6:0] cathodes,
            output reg [3:0] anodes
   );

reg [1:0]switch = 2'd0;
reg [19:0]clock_counter = 20'd0;
reg [3:0]clk_counter_1 = 4'd0;
reg [3:0]clk_counter_2 = 4'd0;
reg [3:0]clk_counter_3 = 4'd0;
reg [1:0] switcheroni = 2'd0;

always@(posedge clk)
            begin
                        if(reset)
                                    begin
                                                anodes <= 4'b1111;
                                    end
                        else
                                    begin
                                                case(switch)
```

```
                                        2'd0:
                                                begin
                                                        anodes <= 4'b1110;
                                                end
                                        2'd1:
                                                begin
                                                        anodes <= 4'b1101;
                                                end
                                        2'd2:
                                                begin
                                                        anodes <= 4'b1011;
                                                end
                                        2'd3:
                                                begin
                                                        anodes <= 4'b1111;
                                                end
                                        default:
                                                begin
                                                        anodes <= 4'b1111;
                                                end
                                endcase
                        end
        end

always@(posedge clk)
        begin
                if(reset)
                        begin

                        end
                else
                        begin
                                if (clock_counter < 100000-1)
                                        begin
                                                clock_counter <= clock_counter + 1;
                                        end
                                else
                                        begin
                                                clock_counter <= 0;
                                                if (switch > 2)
                                                        begin
                                                                switch <= 0;
                                                        end
                                                else
                                                        begin
                                                                switch <= switch + 1;
                                                        end
                                        end
```

```
                    case (switch)
                        2'd0:
                            begin
                                case(switcheroni)
                                    2'd0:
                                        begin
                                            case (data_out)
                                                4'd0:
begin
    cathodes <= 7'b1110001;
end
                                                4'd1:
begin
    cathodes <= 7'b1110000;
end
                                                4'd2:
begin
    cathodes <= 7'b0001001;
end
                                                4'd3:
begin
    cathodes <= 7'b1110000;
end
                                                4'd4:
begin
    cathodes <= 7'b1111010;
end
                                                4'd5:
begin
    cathodes <= 7'b1000010;
```

```
end
default:
begin
        cathodes <= 7'b1111111;
end
                                                        endcase
                                                end
                                        2'd1:
                                                begin
                                                        case (can_counter)
                                                                4'd0:
begin
        cathodes <= 7'b0000001;
end
                                                                4'd1:
begin
        cathodes <= 7'b1001111;
end
                                                                4'd2:
begin
        cathodes <= 7'b0010010;
end
                                                                4'd3:
begin
        cathodes <= 7'b0000110;
end
                                                                4'd4:
begin
        cathodes <= 7'b1001100;
```

```
end
                                                                    4'd5:
begin
        cathodes <= 7'b0100100;
end
                                                                    4'd6:
begin
        cathodes <= 7'b0100000;
end
                                                                    4'd7:
begin
        cathodes <= 7'b0001111;
end
                                                                    4'd8:
begin
        cathodes <= 7'b0000000;
end
                                                                    4'd9:
begin
        cathodes <= 7'b0000100;
end
default:
begin
        cathodes <= 7'b1111111;
end
                                                                endcase
                                                            end
                                                        2'd2:
```

```
                                                                    begin
                                                                        case (bottle_counter)
                                                                            4'd0:
begin
        cathodes <= 7'b0000001;
end
                                                                            4'd1:
begin
        cathodes <= 7'b1001111;
end
                                                                            4'd2:
begin
        cathodes <= 7'b0010010;
end
                                                                            4'd3:
begin
        cathodes <= 7'b0000110;
end
                                                                            4'd4:
begin
        cathodes <= 7'b1001100;
end
                                                                            4'd5:
begin
        cathodes <= 7'b0100100;
end
                                                                            4'd6:
begin
```

```
                cathodes <= 7'b0100000;
        end
                                                                                4'd7:
        begin
                cathodes <= 7'b0001111;
        end
                                                                                4'd8:
        begin
                cathodes <= 7'b0000000;
        end
                                                                                4'd9:
        begin
                cathodes <= 7'b0000100;
        end
        default:
        begin
                cathodes <= 7'b1111111;
        end
                                                                endcase
                                                        end
                                                default:
                                                        begin
                                                                cathodes <=
7'b1111111;
                                                        end
                                                endcase
                                        end
                                2'd1:
                                        begin
                                        case(switcheroni)
                                                2'd0:
                                                        begin
                                                                case (data_out)
```

```
                                                                4'd0:
        begin
        cathodes <= 7'b1000010;
                                                                    end
                                                                4'd1:
        begin
        cathodes <= 7'b0110000;
                                                                    end
                                                                4'd2:
        begin
        cathodes <= 7'b0001000;
                                                                    end
                                                                4'd3:
        begin
        cathodes <= 7'b0000001;
                                                                    end
                                                                4'd4:
        begin
        cathodes <= 7'b1111010;
                                                                    end
                                                                4'd5:
        begin
        cathodes <= 7'b1101010;
                                                                    end
                                                                default:
        begin
        cathodes <= 7'b1111111;
                                                                    end
                                                            endcase
                                                        end
                                                    2'd1:
                                                        begin
                                                            case (can_counter)
```

```
                                                                    4'd0:
        begin
                cathodes <= 7'b0000001;
        end
                                                                    4'd1:
        begin
                cathodes <= 7'b1001111;
        end
                                                                    4'd2:
        begin
                cathodes <= 7'b0010010;
        end
                                                                    4'd3:
        begin
                cathodes <= 7'b0000110;
        end
                                                                    4'd4:
        begin
                cathodes <= 7'b1001100;
        end
                                                                    4'd5:
        begin
                cathodes <= 7'b0100100;
        end
                                                                    4'd6:
        begin
                cathodes <= 7'b0100000;
```

```
end
                                                                    4'd7:
begin
        cathodes <= 7'b0001111;
end
                                                                    4'd8:
begin
        cathodes <= 7'b0000000;
end
                                                                    4'd9:
begin
        cathodes <= 7'b0000100;
end
default:
begin
        cathodes <= 7'b1111111;
end
                                                endcase
                                        end
                                2'd2:
                                        begin
                                                case (bottle_counter)
                                                                    4'd0:
begin
        cathodes <= 7'b0000001;
end
                                                                    4'd1:
begin
        cathodes <= 7'b1001111;
```

```
        end
                                                                4'd2:
        begin
                cathodes <= 7'b0010010;
        end
                                                                4'd3:
        begin
                cathodes <= 7'b0000110;
        end
                                                                4'd4:
        begin
                cathodes <= 7'b1001100;
        end
                                                                4'd5:
        begin
                cathodes <= 7'b0100100;
        end
                                                                4'd6:
        begin
                cathodes <= 7'b0100000;
        end
                                                                4'd7:
        begin
                cathodes <= 7'b0001111;
        end
                                                                4'd8:
        begin
```

```
                cathodes <= 7'b0000000;
        end
                                                                                4'd9:
        begin
                cathodes <= 7'b0000100;
        end
        default:
        begin
                cathodes <= 7'b1111111;
        end
                                                                endcase
                                                        end
                                                default:
                                                        begin
                                                                cathodes <=
7'b1111111;
                                                        end
                endcase
              end
                                        2'd2:
                                                begin
                                                        case(switcheroni)
                                                                2'd0:
                                                                        begin
                                                                                case
(data_out)
                                                                                4'd0:
        begin
                cathodes <= 7'b1001111;
        end
                                                                                4'd1:
        begin
                cathodes <= 7'b1000010;
```

```
        end
                                                                                4'd2:
        begin
                cathodes <= 7'b0110001;
        end
                                                                                4'd3:
        begin
                cathodes <= 7'b1100000;
        end
                                                                                4'd4:
        begin
                cathodes <= 7'b0110000;
        end
                                                                                4'd5:
        begin
                cathodes <= 7'b0110000;
        end
        default:
        begin
                cathodes <= 7'b1111111;
        end
                                                                                endcase
                                                                        end
                                                                2'd1:
                                                                        begin
                                                                                case
(can_counter)
        4'd0:
                begin
```

```
                cathodes <= 7'b0000001;
        end
4'd1:
        begin
                cathodes <= 7'b1001111;
        end
4'd2:
        begin
                cathodes <= 7'b0010010;
        end
4'd3:
        begin
                cathodes <= 7'b0000110;
        end
4'd4:
        begin
                cathodes <= 7'b1001100;
        end
4'd5:
        begin
                cathodes <= 7'b0100100;
        end
4'd6:
        begin
```

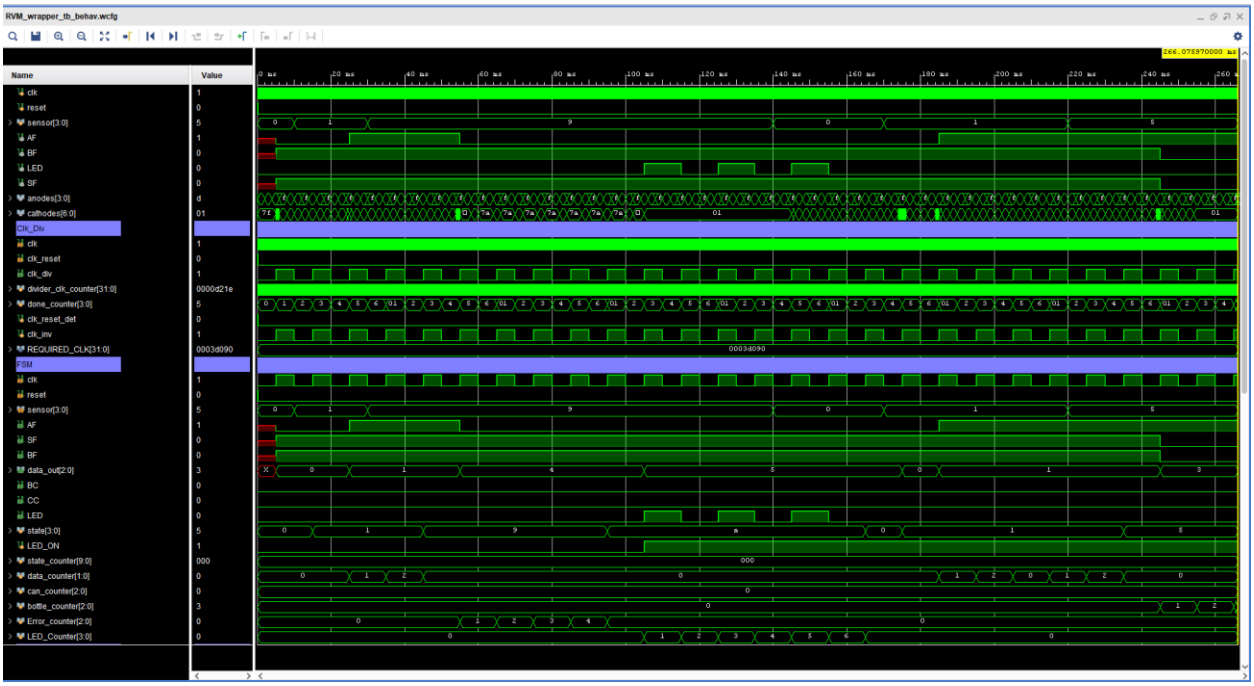
```
                cathodes <= 7'b0100000;
        end
4'd7:
        begin
                cathodes <= 7'b0001111;
        end
4'd8:
        begin
                cathodes <= 7'b0000000;
        end
4'd9:
        begin
                cathodes <= 7'b0000100;
        end
default:
        begin
                cathodes <= 7'b1111111;
        end
                                                                endcase
                                                        end
                                                2'd2:
                                                        begin
                                                                case (bottle_counter)
                                                                        4'd0:
begin
        cathodes <= 7'b0000001;
end
```

```
                                                                    4'd1:
        begin
                cathodes <= 7'b1001111;
        end
                                                                    4'd2:
        begin
                cathodes <= 7'b0010010;
        end
                                                                    4'd3:
        begin
                cathodes <= 7'b0000110;
        end
                                                                    4'd4:
        begin
                cathodes <= 7'b1001100;
        end
                                                                    4'd5:
        begin
                cathodes <= 7'b0100100;
        end
                                                                    4'd6:
        begin
                cathodes <= 7'b0100000;
        end
                                                                    4'd7:
        begin
                cathodes <= 7'b0001111;
```

```
        end
                                                                                4'd8:

        begin

                cathodes <= 7'b0000000;

        end
                                                                                4'd9:

        begin

                cathodes <= 7'b0000100;

        end

        default:

        begin

                cathodes <= 7'b1111111;

        end
                                                                                endcase
                                                                        end
                                                                default:
                                                                        begin
                                                                                cathodes <=
7'b1111111;
                                                                        end
                                                                endcase
                                                        end

                                                default:
                                                        begin
                                                                switch <= 2'd0;
                                                        end
                                        endcase
                                end
        end

always@(posedge clk_div)
        begin
                if (clk_counter_1 > 9)
                        begin
                                if(clk_counter_2 > 1)
                                        begin
```

```
                                        if(clk_counter_3 > 1)
                                                begin
                                                        clk_counter_1 <= 0;
                                                        clk_counter_2 <= 0;
                                                        clk_counter_3 <= 0;
                                                        switcheroni <= 2'd0;
                                                end
                                        else
                                                begin
                                                        switcheroni <= 2'd2;
                                                        clk_counter_3 <= clk_counter_3 + 1;
                                                end
                                end
                        else
                                begin
                                        switcheroni <= 2'd1;
                                        clk_counter_2 <= clk_counter_2 + 1;
                                end
                end
        else
                begin
                        clk_counter_1  <= clk_counter_1 + 1;
                end
    end

endmodule
```

Simulation Results:


RVM_wrapper_tb_behav.wcfg
Name
Value
clk
reset
sensor[3:0]
AF
BF
LED
SF
anodes[3:0]
cathodes[6:0]
Clk_Div
clk
clk_reset
clk_div
divider_clk_counter[31:0]
done_counter[3:0]
clk_reset_det
clk_inv
REQUIRED_CLK[31:0]
FSM
clk
reset
sensor[3:0]
AF
SF
BF
data_out[2:0]
BC
CC
LED
state[3:0]
LED_ON
state_counter[9:0]
data_counter[1:0]
can_counter[2:0]
bottle_counter[2:0]
Error_counter[2:0]
LED_Counter[3:0]