DIGITAL SYSTEM DESIGN PROJECT REPORT 2023

COURSE CODE-EC204

COURSE INSTRUCTOR-RATHNAMALA RAO

ON AUTOMATIC WASHING MACHINE

BY:

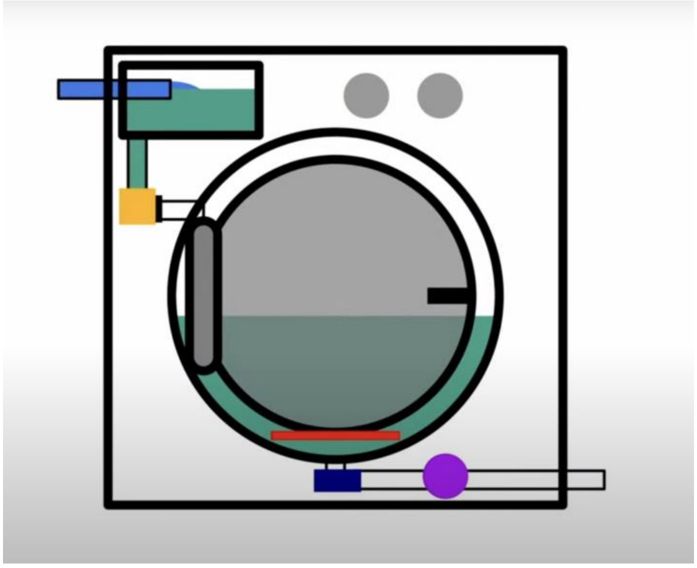
K VICTOR VISHAL (221EC223)

KUSHAL GOWDA C R (221EC224)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION

NATIONAL INSTITUTE OF TECHNOLOGY, KARNATAKA, SURATHKAL

ABSTRACT



OUR AUTOMATIC WASHING PROJECT IS A BASIC VERSION IMPLEMENTATION OF THE CONTROL SYSTEM IN A REGULAR WASHING MACHINE.THE MAIN INNOVATION IN OUR WASHING MACHINE IS THE INCLUSION OF DIFFERENT MODES ON THE BASIS OF WEIGHT OF THE INPUT LOAD TO PROVIDE EFFIECENT CLEANING TO THE CLOTHES

THE WASHING MACHINE HAS THREE MODES AND FOUR STAGES OF PROCESSING TO CLEAN THE CLOTHES

- SOAK
- WASH
- RINSE
- SPIN

THE LAST STAGE FURUTHER HELPS TO DRY THE CLOTH INORDER SAVE EXTRA TIME ON EXTERNAL DRYING

THE WASHING MACHINE ALSO HAS A DEDICATED WATER REGULATION SYSTEM WHICH MONITORS THE WATER INFLOW AND OUTFLOW FROM THE MAIN TUB DURING VARIOUS STAGES INVOLVED IN CLEANSING

WE ALSO HAVE IMPLEMENTED SAFETY FEATURES LIKE TERMINATING THE WHOLE PROCESS WHEN THE LID IS OPEN AND ALSO HAVE

PROVIDED ASYNCHRONOUS CANCEL BUTTON TO TERMINATE THE WHOLE PROCESS INCASE OF EMERGENCY.

PROBLEM STATEMENT

THE WASHING MACHINE NORMALLY TREATS ALL LOADS EQUALLY AND PROVIDES CYCLES OF CONSTANT TIME TO EACH STAGES INVOLVED IN THE WHOLE CLEANING PROCESS IRRESPECTIVE OF THE WEIGHT OF THE LOAD THESE LEADS TO INSUFFICIENT DETERGENT EXPOSURE AND AGITATION LEADING TO DIRT AND STAINS STILL STICKING HE CLOTHES, THEREFORE LEADING TO CUSTOMER DISSATISFACTION

THEREFORE, WE PROPOSE TO IMPROVE CUSTOMER EXPERIENCE BY ADDING DIFFERENT MODES WHICH CUSTOMER CAN CHOOSE BASED ON THE WEIGHT OF THEIR LOADS.NOW THE TIME DURATION OF EACH OF THE DIFFERENT STAGES INVOLVED IS GOING TO VARY BASED ON THE WEIGHT OF THE LOAD

ALSO, ANOTHER MAJOR CONCERN OF SAFETY DUE TO INCREASING NUMEBR OF ACCIDENTS INVOLVING WASHING MACHINE IS ADDRESSED BY PROVIDING A SAFETY FEATURE TO STOP THE WHOLE PROCESS INCASE OF LID BEING OPEN AND PROVISION FOR AND EMERGENCY SAFETY BUTTON

PRE-ASSUMPTIONS

- THE START BUTTON IS LIKE A PUSH BUTTON
- THE INTERNAL MOTOR MECHANICS OF WASHING MACHINE IS ASSUMED TO COMPLY WITH THE DIFFERENT STATES
- WATERINLET IS ASSUMED TO OCCUR INDEPENDENTLY JUST LIKE THE TUB MOTOR MECHANICS
- THE STATE OF THE LID IS ASSUMED TO GIVEN BY A SENSOR

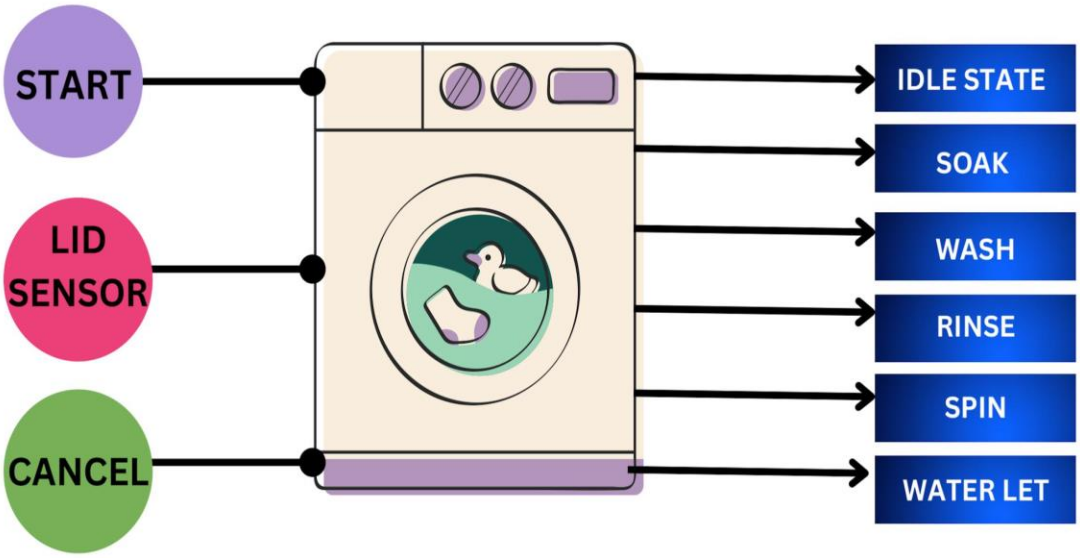
DESIGN

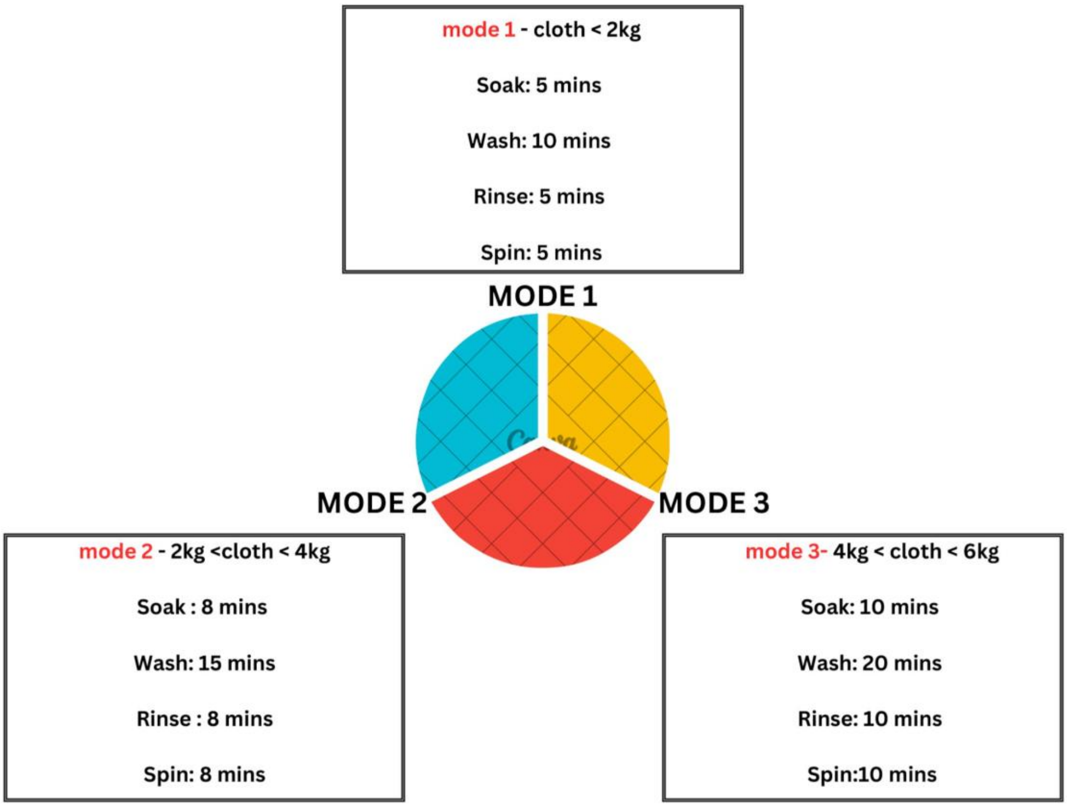
INPUTS: CLOCK, START, LID, CANCEL, MODE_1, MODE_2,MODE_3.

OUTPUTS: IDLE,READY,SOAK,WASH,RINSE, SPIN, WATER_LET,DONE

SPECIFICATION OF STAGES INVOLVED:

```
Mode_1 - cloth < 2kg
         Soak: 5 mins (5*60*250 = 75,000)
         Wash: 10 mins (10*60*250 = 1,50,000)
         Rinse: 5 mins (5*60*250 = 75,000)
         Spin: 5 mins (5*60*250 = 75,000)

    mode2 - 2kg < cloth < 4kg
         Soak  : 8 mins (8*60*250 = 1,20,000)
         Wash  : 15 mins (15*60*250 = 2,25,000)
         Rinse : 8 mins (8*60*250 = 1,20,000)
         Spin  : 8 mins (8*60*250 = 1,20,000)

 mode3 - 4kg < cloth < 6kg
         Soak  : 10 mins (10*60*250 = 1,50,000)
         Wash  : 20 mins (20*60*250 = 3,00,000)
         Rinse : 10 mins (10*60*250 = 1,50,000)
         Spin  : 10 mins (10*60*250 = 1,50,000)
```





mode 1 - cloth < 2kg

Soak: 5 mins

Wash: 10 mins

Rinse: 5 mins

Spin: 5 mins

mode 2 - 2kg <cloth < 4kg

Soak : 8 mins

Wash: 15 mins

Rinse : 8 mins

Spin: 8 mins

mode 3- 4kg < cloth < 6kg

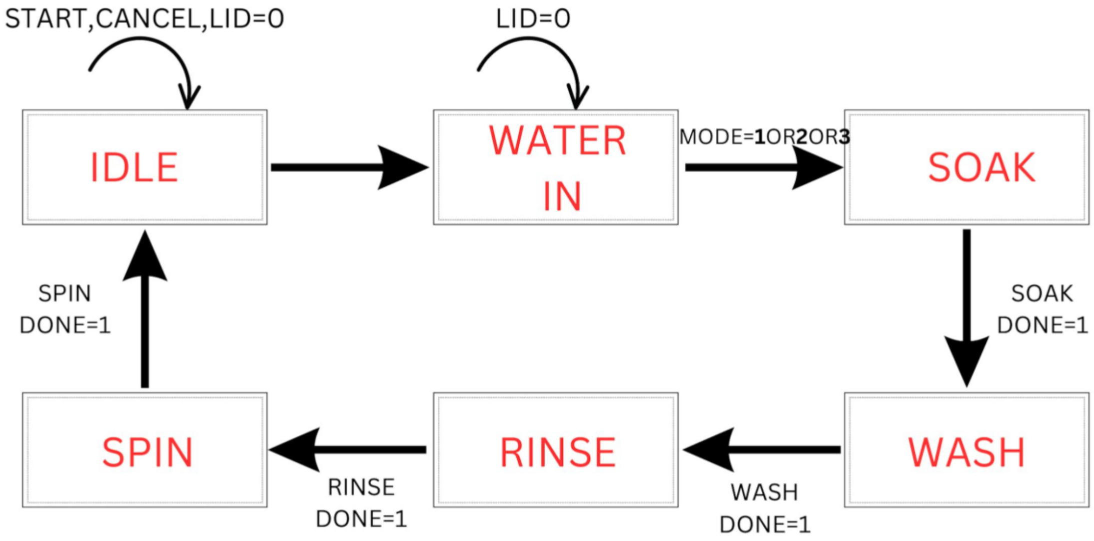
Soak: 10 mins

Wash: 20 mins

Rinse: 10 mins

Spin:10 mins

STATE DIAGRAM



VERILOG CODE

```
 `timescale 1ms / 1ms


module washingmachine(clk,lid,start,cancel,mode_1,mode_2,mode_3,

o_idle,o_ready,o_soak,o_wash,o_rinse,o_spin,o_waterinlet,o_done);


input clk,start,cancel,lid,mode_1,mode_2,mode_3;

output o_idle,o_ready,o_soak,o_wash,o_rinse,o_spin,o_waterinlet,o_done;


parameter IDLE  = 6'b000001,

          READY = 6'b000010,

  SOAK  = 6'b000100,

  WASH  = 6'b001000,

  RINSE = 6'b010000,

  SPIN  = 6'b100000;



   reg [5:0] PS,NS;


  reg soak_done,wash_done,rinse_done,spin_done;


        wire soak_up,wash_up,rinse_up,spin_up;

wire soak_pause,wash_pause,rinse_pause,spin_pause;

reg [18:0] soakcounter,washcounter,rinsecounter,spincounter; //19 bits can count
upto 5,24,288


//timer pause logic when lid is open

assign soak_pause  = (PS == SOAK)  && (lid);

assign wash_pause  = ( PS == WASH )  && (lid) ;

assign rinse_pause = ( PS == RINSE ) && (lid) ;

assign spin_pause  = ( PS == SPIN )  && (lid) ;
assign soak_up  = (PS == SOAK) && (mode_1 || mode_2 || mode_3);
assign wash_up  = (PS == WASH);
assign rinse_up = (PS == RINSE);
assign spin_up  = (PS == SPIN);

//SOAK DONE
        always@(mode_1,mode_2,mode_3,soakcounter)
    begin
    if(mode_1)
  soak_done = (soakcounter == 75000) ? 1'b1 : 1'b0;
else if(mode_2)
  soak_done = (soakcounter == 120000) ? 1'b1 : 1'b0;
                else if(mode_3)
                  soak_done = (soakcounter == 150000) ? 1'b1 : 1'b0;
             end

//WASH DONE
            always@(mode_1,mode_2,mode_3,washcounter)
    begin
    if(mode_1)
  wash_done = (washcounter == 150000) ? 1'b1 : 1'b0;
else if(mode_2)
  wash_done = (washcounter == 225000) ? 1'b1 : 1'b0;
                else if(mode_3)
  wash_done = (washcounter == 300000) ? 1'b1 : 1'b0;

end
 //RINSE DONE
              always@(mode_1,mode_2,mode_3,rinsecounter)
    begin
    if(mode_1)
  rinse_done = (rinsecounter == 75000) ? 1'b1 : 1'b0;
else if(mode_2)
```

```
  rinse_done = (rinsecounter == 120000) ? 1'b1 : 1'b0;
              else if(mode_3)
  rinse_done = (rinsecounter == 150000) ? 1'b1 : 1'b0;

end

//SPIN DONE
              always@(mode_1,mode_2,mode_3,spincounter)
    begin
    if(mode_1)
  spin_done = (spincounter == 75000) ? 1'b1 : 1'b0;
else if(mode_2)
  spin_done = (spincounter == 120000) ? 1'b1 : 1'b0;
              else if(mode_3)
  spin_done = (spincounter == 150000) ? 1'b1 : 1'b0;

end
   //SOAK COUNTER
        always@(posedge clk)
            begin
              if(start)
        soakcounter <= 0;
   if(soak_done)
                  soakcounter <= 0;
   else if(soak_pause)
                  soakcounter <= soakcounter;
             else if(soak_up)
                  soakcounter <= soakcounter + 1'b1;
            end

        //WASH COUNTER
        always@(posedge clk)
            begin
              if(start)
        washcounter <= 0;
else if(wash_done)
                    washcounter <= 0;
else if(wash_pause)
                    washcounter <= washcounter;
                else if(wash_up)
                    washcounter <= washcounter + 1'b1;
            end

//RINSE COUNTER
always@(posedge clk)
            begin
              if(start)
        rinsecounter <= 0;
   else if(rinse_done)
                    rinsecounter <= 0;
   else if(rinse_pause)
                    rinsecounter <= rinsecounter;
               else if(rinse_up)
                    rinsecounter <= rinsecounter + 1'b1;
            end
        //SPIN COUNTER
        always@(posedge clk)
            begin
              if(start)
        spincounter <= 0;
   else if(spin_done)
                    spincounter <= 0;
   else if(spin_pause)
                    spincounter <= spincounter;
               else if(spin_up)
                    spincounter <= spincounter + 1'b1;
            end
```

```
always@(posedge clk)
       begin
       if(start)
      PS <= IDLE;
   else if(cancel)
                  PS <= IDLE;
               else
                  PS <= NS;
   end

always@(*)
             begin
case(PS)
  IDLE : begin
          if(!lid && !cancel)
     NS <= READY;
  else
                             NS <= PS;
 end

  READY : begin
                    if(!lid && !cancel && (mode_1 || mode_2 || mode_3))
  NS <= SOAK;
else
                              NS <= PS;
          end

                  SOAK : begin
                    if(!lid && !cancel && soak_done)
  NS <= WASH;
else
                              NS <= PS;
         end
```

```
                  WASH : begin
                         if(!lid && !cancel && wash_done)
  NS <= RINSE;
else
                              NS <= PS;
         end

                  RINSE : begin
                    if(!lid && !cancel && rinse_done)
  NS <= SPIN;
else
                              NS <= PS;
          end

                  SPIN : begin
                    if(!lid && !cancel && spin_done)
  NS <= IDLE;
else
                              NS <= PS;
         end
                  default :  NS <= IDLE;
                endcase
    end
//OUTPUT LOGIC
 assign o_idle = (PS == IDLE);
        assign o_ready = (PS == READY);
        assign o_soak = (PS == SOAK);
        assign o_wash = (PS == WASH);
        assign o_rinse = (PS == RINSE);
        assign o_spin = (PS == SPIN);
        assign o_waterinlet =  (PS == WASH) || (PS == RINSE) ;
        assign o_done = (PS == SPIN) && (spin_done);


     Endmodule
```

TESTBENCH CODE

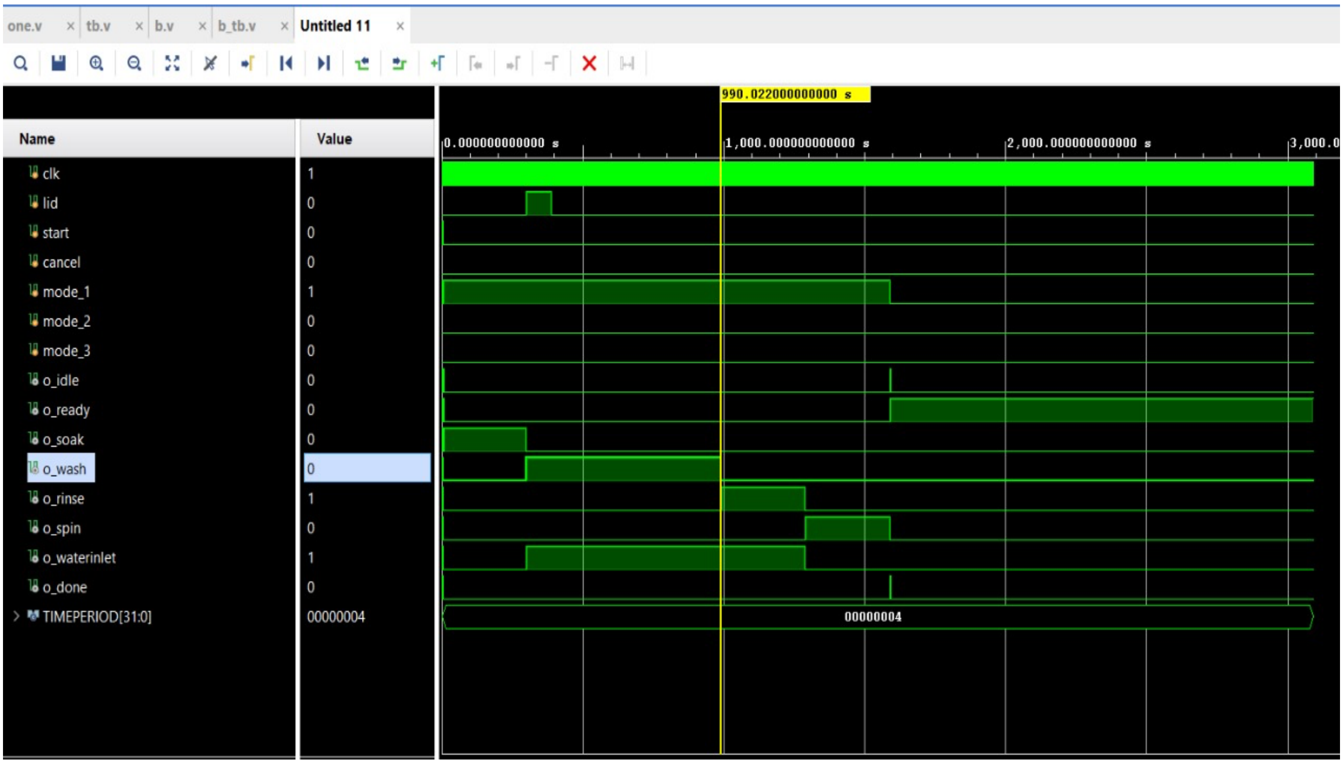
```
`timescale 1ms / 1ms


module tb_washingmachine();
  reg clk,lid,start,cancel,mode_1,mode_2,mode_3;
  wire o_idle,o_ready,o_soak,o_wash,o_rinse,o_spin,o_waterinlet,o_done;


washingmachine DUT(clk,lid,start,cancel,mode_1,mode_2,mode_3,

o_idle,o_ready,o_soak,o_wash,o_rinse,o_spin,o_waterinlet,o_done);


  parameter TIMEPERIOD = 4;
  initial
     begin
       clk = 1'b0;
       forever #(TIMEPERIOD/2) clk = ~ clk; //250hz
     end
  task scenario_1();
  begin


    @(posedge clk) start = 1'b0;
@(posedge clk) mode_1 = 1'b1;
#300040 @(posedge clk);
 lid = 1'b1;
 #90000
@(posedge clk) lid = 1'b0;


wait(DUT.o_done)
  @(posedge clk);mode_1 = 1'b0;
  end
endtask
```

```
 task scenario_2();
  begin
    @(posedge clk)
     start = 1'b0;
@(posedge clk)
 mode_1 = 1'b1;



wait(DUT.o_done)
  @(posedge clk);
  mode_1 = 1'b0;
  end
endtask
 task scenario_3();
  begin
    @(posedge clk)
     start = 1'b0;
@(posedge clk)
 mode_2 = 1'b1;
wait(DUT.o_done)
  @(posedge clk);
  mode_2 = 1'b0;
  end
endtask




task scenario_4();
  begin
    @(posedge clk)
     start = 1'b0
             @(posedge clk)mode_3 = 1'b1;
wait(DUT.o_done)
  @(posedge clk);
  mode_3 = 1'b0;
  end
endtask

task scenario_5();
  begin
    @(posedge clk) start = 1'b0;
@(posedge clk) mode_1 = 1'b1;

#500000
 @(posedge clk);
cancel = 1'b1;
#100000
@(posedge clk) cancel = 1'b0;
wait(DUT.o_done)
  @(posedge clk);
  mode_1 = 1'b0;
  end
endtask

initial
     begin
       {lid,start,cancel,mode_1,mode_2,mode_3} = 0;
       @(posedge clk) start = 1'b1;

 scenario_1;
   //scenario_2;
   //scenario_3;
       // scenario_4;
//   scenario_5;


       #1500000;
       $finish;
     end
```

```
endmodule
```

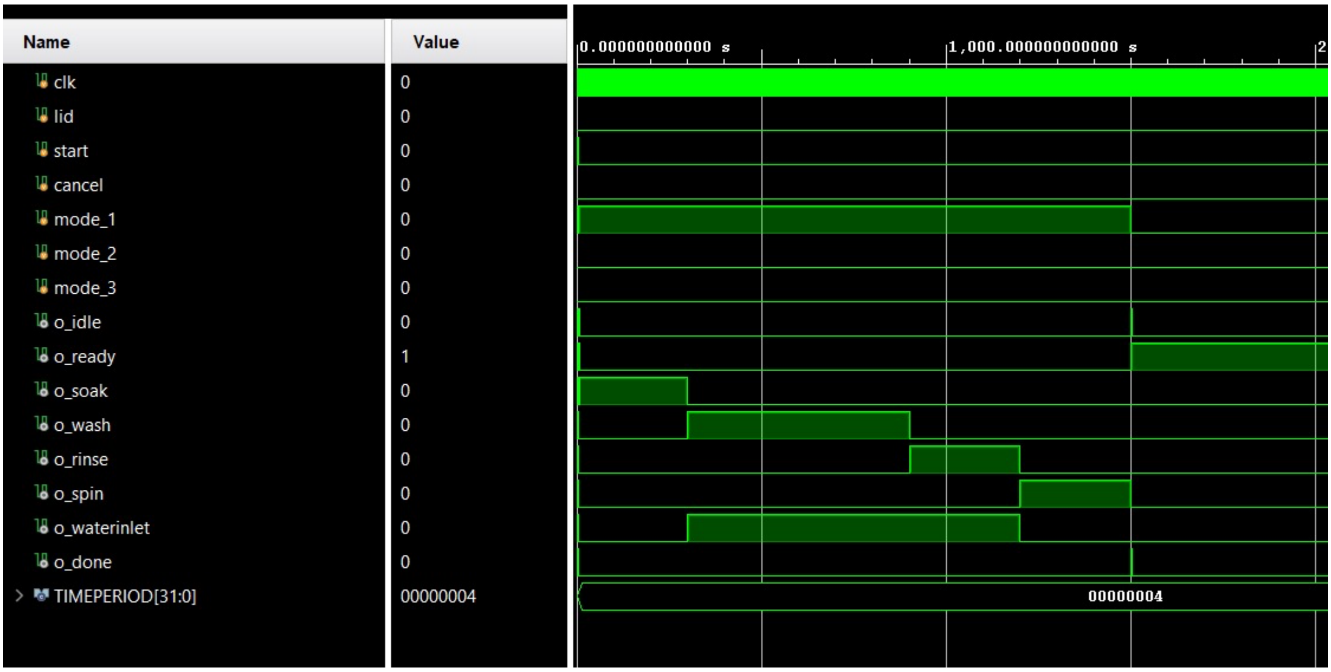
SIMULATIONS

SCENARIO 1: THE LID IS OPEN FOR 90S DURING WASH STAGE



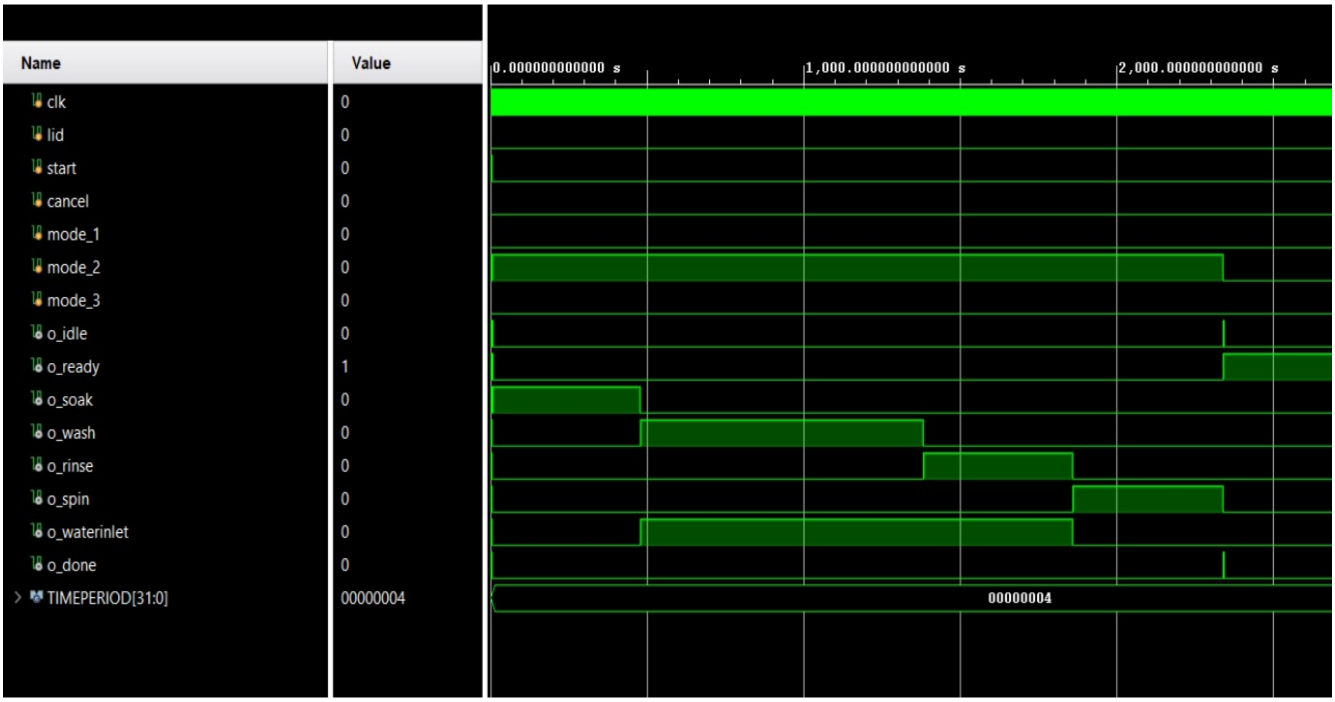
Name	Value
clk	1
lid	0
start	0
cancel	0
mode_1	1
mode_2	0
mode_3	0
o_idle	0
o_ready	0
o_soak	0
o_wash	0
o_rinse	1
o_spin	0
o_waterinlet	1
o_done	0
TIMEPERIOD[31:0]	00000004

SCENARIO 2: MODE_1



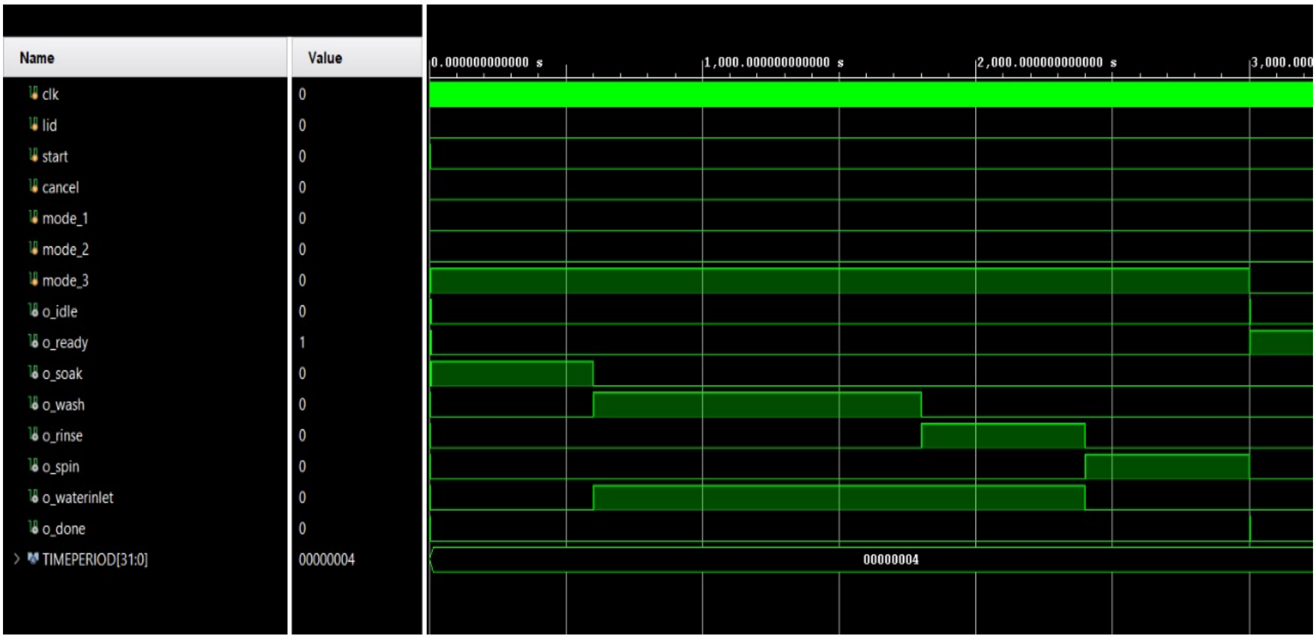
Name	Value
clk	0
lid	0
start	0
cancel	0
mode_1	0
mode_2	0
mode_3	0
o_idle	0
o_ready	1
o_soak	0
o_wash	0
o_rinse	0
o_spin	0
o_waterinlet	0
o_done	0
TIMEPERIOD[31:0]	00000004

SCENARIO 3:MODE_2



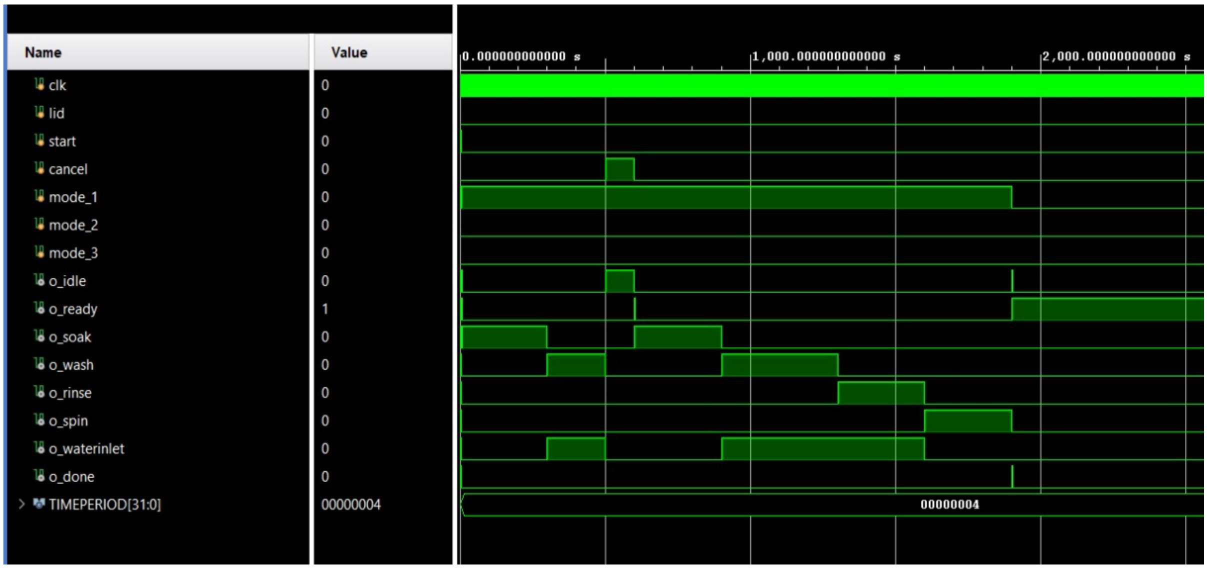
Name	Value
clk	0
lid	0
start	0
cancel	0
mode_1	0
mode_2	0
mode_3	0
o_idle	0
o_ready	1
o_soak	0
o_wash	0
o_rinse	0
o_spin	0
o_waterinlet	0
o_done	0
TIMEPERIOD[31:0]	00000004

SCENARIO 4:MODE_3



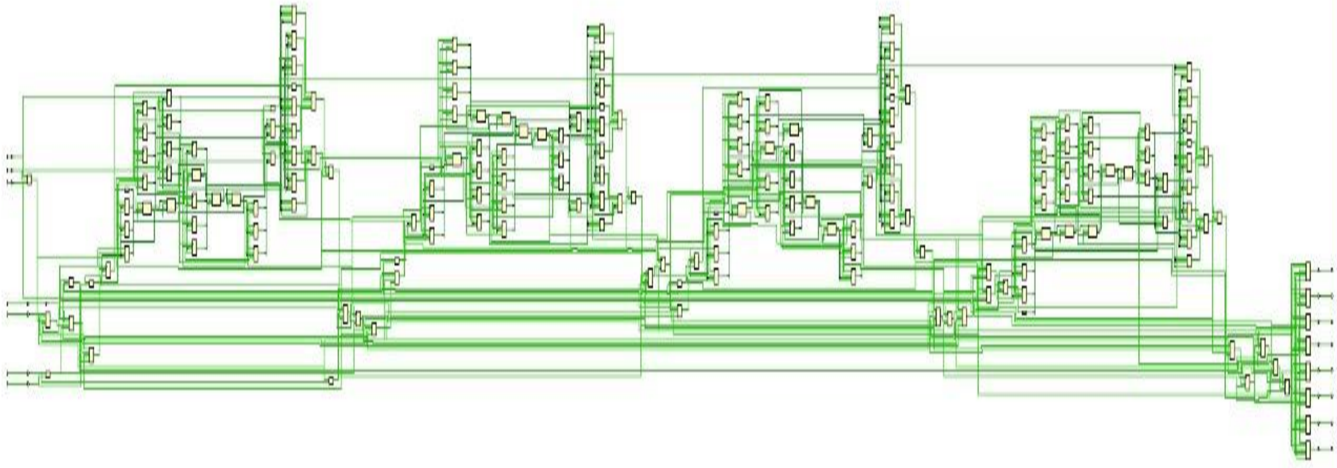
Name	Value
clk	0
lid	0
start	0
cancel	0
mode_1	0
mode_2	0
mode_3	0
o_idle	0
o_ready	1
o_soak	0
o_wash	0
o_rinse	0
o_spin	0
o_waterinlet	0
o_done	0
TIMEPERIOD[31:0]	00000004

SCENARION 5: CANCEL IS ON FOR 100S DURING WASH STAGE



Name	Value
clk	0
lid	0
start	0
cancel	0
mode_1	0
mode_2	0
mode_3	0
o_idle	0
o_ready	1
o_soak	0
o_wash	0
o_rinse	0
o_spin	0
o_waterinlet	0
o_done	0
TIMEPERIOD[31:0]	00000004

SYNTHESIS



CONCLUSION

In conclusion, the control system designed for the automatic washing machine with three distinct modes catering to different load weights, an emergency cancel button, and a safety feature pausing the process when the lid is open has demonstrated effective and user-friendly operation. The inclusion of tailored modes for varying load sizes ensures versatility and efficiency in the washing process. The emergency cancel button provides a quick and responsive way for users to halt the operation if needed. The safety feature addressing the lid status enhances user protection and aligns with modern safety standards. Through the integration of thoughtful functionalities, this control system not only offers practicality and convenience but also prioritizes user safety, marking a successful endeavour in the realm of household appliances.

REFERENCES

- Verilog https://www.asic-world.com/verilog/veritut.html
- Mohamed Anas MN repository
- https://www.explainthatstuff.com/washingmachine.html
- https://hmhub.in/stages-wash-cycle/
- Youtube https://www.youtube.com/watch?v=iAoi9jTzxcI&list=PLUn6cqainH8jZxS3ppSGPi3rNScz9cFZf&index=7