THIRD YEAR B. TECH FIFTH SEMESTER

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

HDL MINI PROJECT
(ECL 303)

ON

Washing Machine Control System on Modelsim using Verilog

By

Nishchay Valia (BT21ECE128)
Sahil Mandi (BT21ECE098)

Under the guidance of

Dr. Sushmita Dandeliya
(Course Coordinator)

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

NAGPUR

(An Institution of National Importance by Act of Parliament)

ABSTRACT

This paper aims to showcase the capabilities and versatility of Verilog Hardware Description Language (HDL) by developing the control system for an automated washing machine. The project successfully achieves this goal through the implementation of the washing machine's control system using the Finite State Machine model. This control system generates the necessary signals to oversee and manage the washing machine's overall operation. The digital design is rigorously tested and simulated using MODELSIM software, a product by Mentor Graphics.

TABLE OF CONTENTS

1.Abstract

2.Table of Contents

3. Summary

4. Specifications

5.Block Diagram

6. Source Code Justification

7.Verilog Code

8.Waveform

9.Transcript Window Result

Summary

The Automatic Washing Machine controller encompasses a range of essential functionalities. The washing machine operates within distinct states, including Check_Door, Fill_Water, Add_Detergent, Drain_Water, and Spin. The controller comprises two fundamental blocks: a Finite-State Machine (FSM) block and a Timer block. The FSM block receives inputs from the user, timer, and hardware components like the door sensor, while its outputs regulate the Timer block and other washing machine components. Table1 delineates the FSM's input and output signals and their respective functions.

The Timer block plays a pivotal role in generating precise time periods for each washing cycle, following a reset. It is composed of an up-counter and combinational logic to provide accurate time signals once specific count values are reached. The timer values are contingent on the clock frequency employed within the system, ensuring proper timing for the washing machine's operation.

Specifications:

Signal Name	**Direction**	**Purpose**
Clock	input	Clock, positive edge triggered.
Reset	input	Reset, synchronous and active high.
Start	Input	Should generate a pulse; 1 = start washing process; 0 = don't start. This start should be used to generate a pulse to trigger an internal flag register.

Door_close	Input	Used to see if the door is closed;
		Door_close = 0; Machine stops (in any state).
		Door_close = 1; Machine resumes/starts.

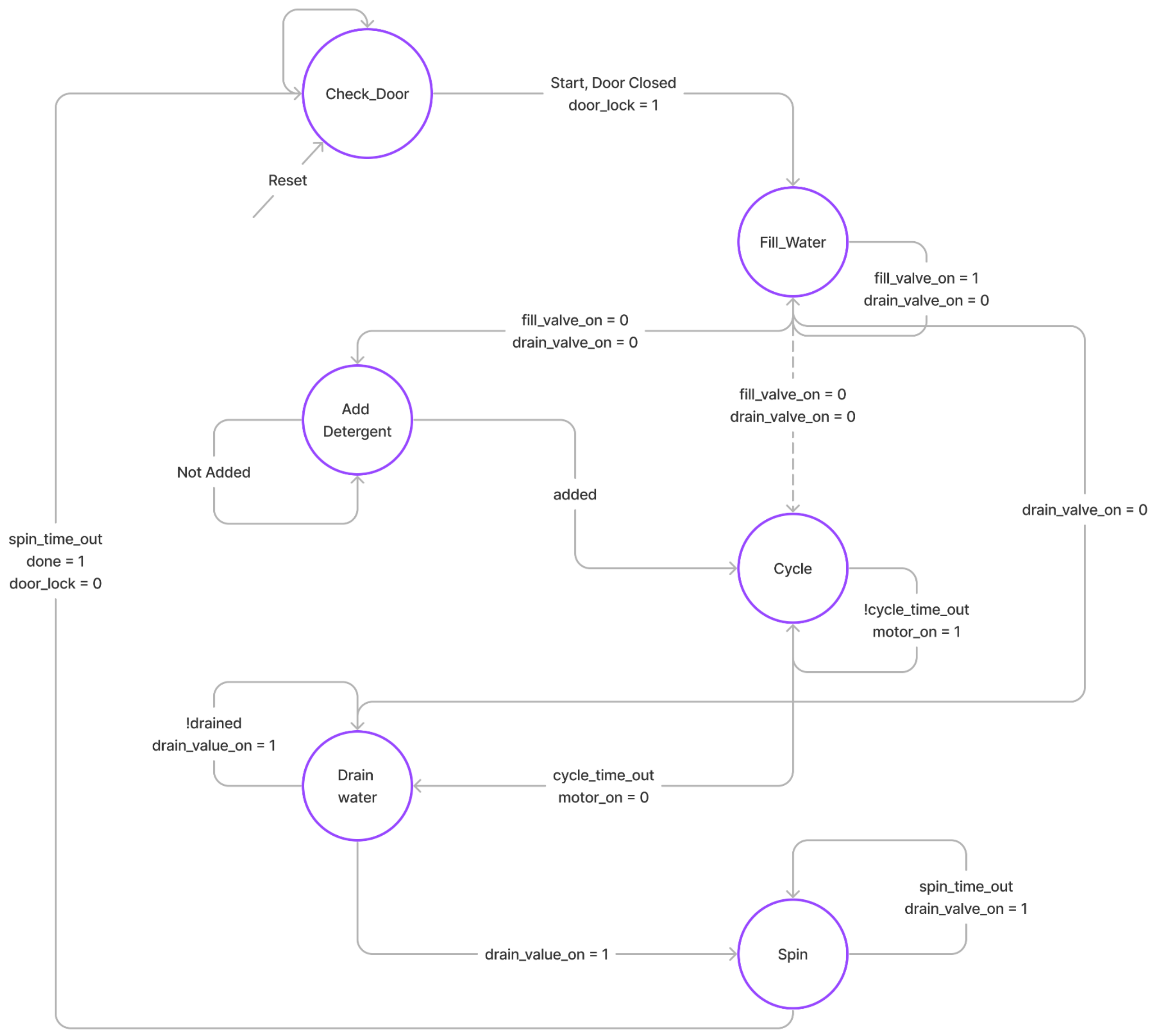
Filled	Input	From some sensor; Filled = 1 means water filled OK.
Drained	Input	From some sensor; Drained = 1 means drained OK.
Added	Input	From user; Added = 1 means detergent added OK.
Cycle_Timeout	Input	From timer; Cycle_Timeout = 1 means time finished go to next state
Spin_Timeout	Input	From timer; Spin_Timeout = 1 means time finished go to next state
Motor_on	output	Motor_on = 1; drive the motor. Motor_on = 0; motor off, not driven.

Fill_Valve_on	output	Fill_valve_on = 1; valve open& filling will proceed.
		Fill_valve_on = 0; valve closed & filling stops.

Drain_valve_on	output	Drain_valve_on = 1; drain valve open & draining in progress. Drain_valve_on = 0; drain valve closed and draining stopped.
Door_lock	output	To lock the door (some mechanism) Door_lock = 1 when the machine has started Door_lock = 0 when the machine has done washing The essential condition for the Door_lock is that there must be no Water inside the machine. That is, Door_lock can only be 0 when Draining has been finished.
Done	output	Done = 1 when draining has been finished and washing has been finished. This should also deassert Door_lock.

Table 1: Alphabetical listing of input and output signals for the FSM

FSM Diagram



The working of the washing machine control system

Explanation

In the design of the source code, it was chosen that a case statement be the preference for the design. A FSM-structure was chosen for the design. It was chosen that based on the amount of states, a 3 bit signal would be enough to represent the state. There were a total of 6 states in our design. There is 1 “off” state for when the power is off, 1 “Check_Door” state for when the machine is not running, or incrementing, and 6 states for the washing machine. We originally wanted to add two scenarios that we could handle, which are if the machine asynchronously loses power, or if the door asynchronously opens. We were successful in implementing both signals in simulation. Then After the Check_Door State there is Fill_Water state which checks that the water is filled in the drum .After that there is Add_Detergent state which checks if the detergent has been added .After it the Cycle stage is there which accomplishes the washing cycle ,After that Drain _Water stage has been encountered which checks if the water has been draind out or not and finally there is Spin stage and when the process has been completed the process is Finally Done and each stage has been changed after the encounter of the positive pulse of the clock only .

Verilog Code

```verilog
module Automatic_Washing_Machine (
    input Clock,
    input Reset,
    input Start,
    input Door_Close,
    input Filled,
    input Drained,
    input Detergent_Added,
    input Cycle_Timeout,
    input Spin_Timeout,
    output reg Motor_on,
    output reg Fill_valve_on,
    output reg Drained_valve_on,
    output reg Door_Lock,
    output reg Done
);

parameter Check_Door = 3'b000;
parameter Fill_Water = 3'b001;
parameter Add_Detergent = 3'b010;
parameter Cycle = 3'b011;
parameter Drain_Water = 3'b100;
parameter Spin = 3'b101;

reg [2:0] Current_State, Next_State;

always @(posedge Clock, negedge Reset) begin
    if (Reset == 0) begin
        Current_State <= Check_Door;
    end else begin
        Current_State <= Next_State;
    end
end

always @(*) begin
    case (Current_State)
        Check_Door: begin
            if (Start == 1 && Door_Close == 1) begin
                Next_State = Fill_Water;
                Motor_on <= 0;
                Fill_valve_on <= 0;
                Drained_valve_on <= 0;
                Door_Lock <= 1;
            Done <= 0;
        end else begin
            Next_State = Current_State;
            Motor_on <= 0;
            Fill_valve_on <= 0;
            Drained_valve_on <= 0;
            Door_Lock <= 0;
            Done <= 0;
        end
    end
    Fill_Water: begin
        if (Filled == 1) begin
            Next_State = Add_Detergent;
            Motor_on <= 0;
            Fill_valve_on <= 0;
            Drained_valve_on <= 0;
            Door_Lock <= 1;
            Done <= 0;
        end else begin
            Next_State = Current_State;
            Motor_on <= 0;
            Fill_valve_on <= 1;
            Drained_valve_on <= 0;
            Door_Lock <= 1;
            Done <= 0;
        end
    end
    Add_Detergent: begin
        if (Detergent_Added == 1) begin
            Next_State = Cycle;
            Motor_on <= 0;
            Fill_valve_on <= 0;
            Drained_valve_on <= 0;
            Door_Lock <= 1;
            Done <= 0;
        end else begin
            Next_State = Current_State;
            Motor_on <= 0;
            Fill_valve_on <= 0;
            Drained_valve_on <= 0;
            Door_Lock <= 1;
            Done <= 0;
        end
    end
    Cycle: begin
        if (Cycle_Timeout == 1) begin
```

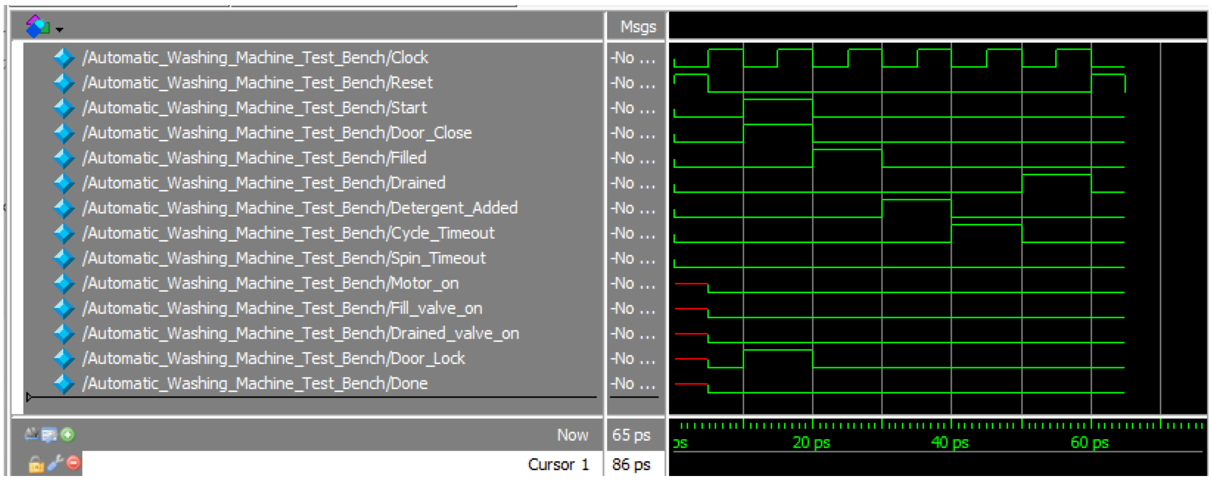
```
                Next_State = Drain_Water;
                Motor_on <= 0;
                Fill_valve_on <= 0;
                Drained_valve_on <= 0;
                Door_Lock <= 1;
                Done <= 0;
            end else begin
                Next_State = Current_State;
                Motor_on <= 1;
                Fill_valve_on <= 0;
                Drained_valve_on <= 0;
                Door_Lock <= 1;
                Done <= 0;
            end
        end
        Drain_Water: begin
            if (Drained == 1) begin
                Next_State = Spin;
                Motor_on <= 0;
                Fill_valve_on <= 0;
                Drained_valve_on <= 1;
                Door_Lock <= 1;
                Done <= 0;
            end else begin
                Next_State = Current_State;
                Motor_on <= 0;
                Fill_valve_on <= 0;
                Drained_valve_on <= 1;
                Door_Lock <= 1;
                Done <= 0;
            end
        end
        Spin: begin
            if (Spin_Timeout == 1) begin
                Next_State = Check_Door;
                Motor_on <= 0;
                Fill_valve_on <= 0;
                Drained_valve_on <= 0;
                Door_Lock <= 0;
                Done <= 1;
            end else begin
                Next_State = Current_State;
                Motor_on <= 0;
                Fill_valve_on <= 0;
                Drained_valve_on <= 1;
                Door_Lock <= 1;
                Done <= 0;
            end
        end
        default: begin
            Next_State = Check_Door;
        end
    endcase
end

endmodule
```

Test Bench

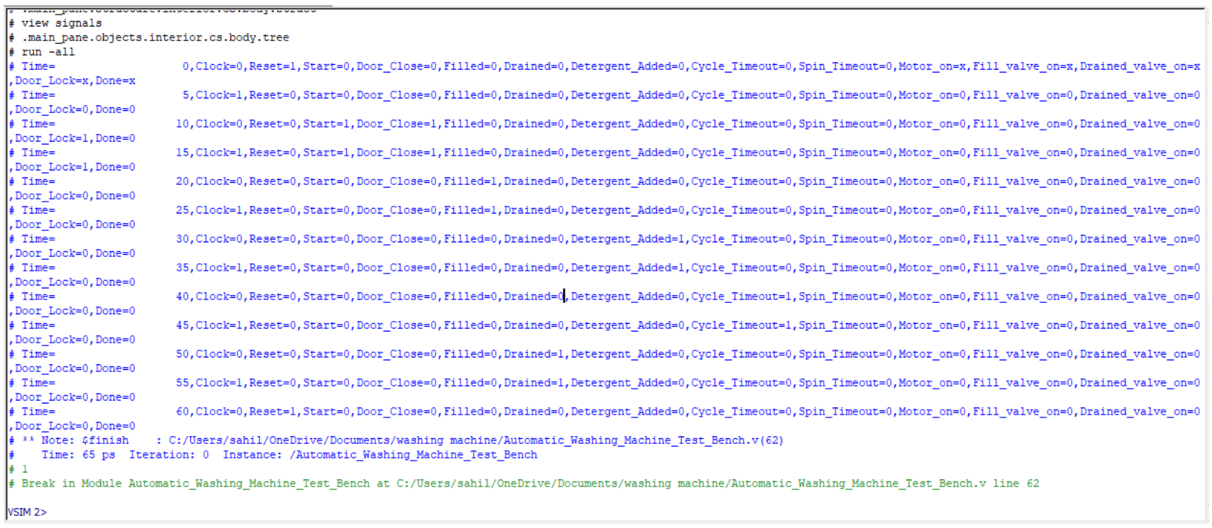
```
module Automatic_Washing_Machine_Test_Bench;

  reg Clock, Reset, Start, Door_Close, Filled, Drained, Detergent_Added,
Cycle_Timeout, Spin_Timeout;
  wire Motor_on, Fill_valve_on, Drained_valve_on, Door_Lock, Done;

  Automatic_Washing_Machine Customer1(
    .Clock(Clock),
    .Reset(Reset),
    .Start(Start),
    .Door_Close(Door_Close),
    .Filled(Filled),
    .Drained(Drained),
    .Detergent_Added(Detergent_Added),
    .Cycle_Timeout(Cycle_Timeout),
    .Spin_Timeout(Spin_Timeout),
    .Motor_on(Motor_on),
    .Fill_valve_on(Fill_valve_on),
    .Drained_valve_on(Drained_valve_on),
    .Door_Lock(Door_Lock),
    .Done(Done)
  );

  initial begin

$monitor("Time=%d,Clock=%b,Reset=%b,Start=%b,Door_Close=%b,Filled=%b,Dra
ined=%b,Detergent_Added=%b,Cycle_Timeout=%b,Spin_Timeout=%b,Motor_on=%b,
Fill_valve_on=%b,Drained_valve_on=%b,Door_Lock=%b,Done=%b",
      $time, Clock, Reset, Start, Door_Close, Filled, Drained,
Detergent_Added, Cycle_Timeout, Spin_Timeout,
```

```verilog
      Motor_on, Fill_valve_on, Drained_valve_on, Door_Lock, Done);
  end

  initial begin
    Clock = 0;
    Reset = 1;
    Start = 0;
    Door_Close = 0;
    Filled = 0;
    Drained = 0;
    Detergent_Added = 0;
    Cycle_Timeout = 0;
    Spin_Timeout = 0;
    #5 Reset = 0;
    #5 Start = 1;
    Door_Close = 1;
    #10 Filled = 1;
    Start = 0;
    Door_Close = 0;
    #10 Detergent_Added = 1;
    Filled = 0;
    #10 Cycle_Timeout = 1;
    Detergent_Added = 0;
    #10 Drained = 1;
    Cycle_Timeout = 0;
    #10 Spin_Timeout = 1;
    Drained = 0;
    Spin_Timeout = 0;

    // Reset the state machine for the next test case
    Reset = 1;
    #5 Reset = 0;

    // Add additional test cases if needed

    // End the simulation
    $finish;
  end

  always begin
    #5 Clock = ~Clock;
  end

endmodule
```

Waveform



Output Transcript



```
# view signals
# .main_pane.objects.interior.cs.body.tree
# run -all
# Time=                    0,Clock=0,Reset=1,Start=0,Door_Close=0,Filled=0,Drained=0,Detergent_Added=0,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=x,Fill_valve_on=x,Drained_valve_on=x
,Door_Lock=x,Done=x
# Time=                    5,Clock=1,Reset=0,Start=0,Door_Close=0,Filled=0,Drained=0,Detergent_Added=0,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_valve_on=0
,Door_Lock=0,Done=0
# Time=                   10,Clock=0,Reset=0,Start=1,Door_Close=1,Filled=0,Drained=0,Detergent_Added=0,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_valve_on=0
,Door_Lock=1,Done=0
# Time=                   15,Clock=1,Reset=0,Start=1,Door_Close=1,Filled=0,Drained=0,Detergent_Added=0,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_valve_on=0
,Door_Lock=1,Done=0
# Time=                   20,Clock=0,Reset=0,Start=0,Door_Close=0,Filled=1,Drained=0,Detergent_Added=0,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_valve_on=0
,Door_Lock=0,Done=0
# Time=                   25,Clock=1,Reset=0,Start=0,Door_Close=0,Filled=1,Drained=0,Detergent_Added=0,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_valve_on=0
,Door_Lock=0,Done=0
# Time=                   30,Clock=0,Reset=0,Start=0,Door_Close=0,Filled=0,Drained=0,Detergent_Added=1,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_valve_on=0
,Door_Lock=0,Done=0
# Time=                   35,Clock=1,Reset=0,Start=0,Door_Close=0,Filled=0,Drained=0,Detergent_Added=1,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_valve_on=0
,Door_Lock=0,Done=0
# Time=                   40,Clock=0,Reset=0,Start=0,Door_Close=0,Filled=0,Drained=0,Detergent_Added=0,Cycle_Timeout=1,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_valve_on=0
,Door_Lock=0,Done=0
# Time=                   45,Clock=1,Reset=0,Start=0,Door_Close=0,Filled=0,Drained=0,Detergent_Added=0,Cycle_Timeout=1,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_valve_on=0
,Door_Lock=0,Done=0
# Time=                   50,Clock=0,Reset=0,Start=0,Door_Close=0,Filled=0,Drained=1,Detergent_Added=0,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_valve_on=0
,Door_Lock=0,Done=0
# Time=                   55,Clock=1,Reset=0,Start=0,Door_Close=0,Filled=0,Drained=1,Detergent_Added=0,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_valve_on=0
,Door_Lock=0,Done=0
# Time=                   60,Clock=0,Reset=1,Start=0,Door_Close=0,Filled=0,Drained=0,Detergent_Added=0,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_valve_on=0
,Door_Lock=0,Done=0
# ** Note: $finish    : C:/Users/sahil/OneDrive/Documents/washing machine/Automatic_Washing_Machine_Test_Bench.v(62)
#    Time: 65 ps  Iteration: 0  Instance: /Automatic_Washing_Machine_Test_Bench
# 1
# Break in Module Automatic_Washing_Machine_Test_Bench at C:/Users/sahil/OneDrive/Documents/washing machine/Automatic_Washing_Machine_Test_Bench.v line 62

VSIM 2>
```

```
Time=
0,Clock=0,Reset=1,Start=0,Door_Close=0,Filled=0,Drained=0,Detergent_Adde
d=0,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=x,Fill_valve_on=x,Drained_va
lve_on=x,Door_Lock=x,Done=x
# Time=
5,Clock=1,Reset=0,Start=0,Door_Close=0,Filled=0,Drained=0,Detergent_Adde
d=0,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_va
lve_on=0,Door_Lock=0,Done=0
# Time=
10,Clock=0,Reset=0,Start=1,Door_Close=1,Filled=0,Drained=0,Detergent_Add
```

```
ed=0,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_v
alve_on=0,Door_Lock=1,Done=0
# Time=
15,Clock=1,Reset=0,Start=1,Door_Close=1,Filled=0,Drained=0,Detergent_Add
ed=0,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_v
alve_on=0,Door_Lock=1,Done=0
# Time=
20,Clock=0,Reset=0,Start=0,Door_Close=0,Filled=1,Drained=0,Detergent_Add
ed=0,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_v
alve_on=0,Door_Lock=0,Done=0
# Time=
25,Clock=1,Reset=0,Start=0,Door_Close=0,Filled=1,Drained=0,Detergent_Add
ed=0,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_v
alve_on=0,Door_Lock=0,Done=0
# Time=
30,Clock=0,Reset=0,Start=0,Door_Close=0,Filled=0,Drained=0,Detergent_Add
ed=1,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_v
alve_on=0,Door_Lock=0,Done=0
# Time=
35,Clock=1,Reset=0,Start=0,Door_Close=0,Filled=0,Drained=0,Detergent_Add
ed=1,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_v
alve_on=0,Door_Lock=0,Done=0
# Time=
40,Clock=0,Reset=0,Start=0,Door_Close=0,Filled=0,Drained=0,Detergent_Add
ed=0,Cycle_Timeout=1,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_v
alve_on=0,Door_Lock=0,Done=0
# Time=
45,Clock=1,Reset=0,Start=0,Door_Close=0,Filled=0,Drained=0,Detergent_Add
ed=0,Cycle_Timeout=1,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_v
alve_on=0,Door_Lock=0,Done=0
# Time=
50,Clock=0,Reset=0,Start=0,Door_Close=0,Filled=0,Drained=1,Detergent_Add
ed=0,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_v
alve_on=0,Door_Lock=0,Done=0
# Time=
55,Clock=1,Reset=0,Start=0,Door_Close=0,Filled=0,Drained=1,Detergent_Add
ed=0,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_v
alve_on=0,Door_Lock=0,Done=0
# Time=
60,Clock=0,Reset=1,Start=0,Door_Close=0,Filled=0,Drained=0,Detergent_Add
ed=0,Cycle_Timeout=0,Spin_Timeout=0,Motor_on=0,Fill_valve_on=0,Drained_v
alve_on=0,Door_Lock=0,Done=0
# ** Note: $finish    : C:/Users/sahil/OneDrive/Documents/washing
machine/Automatic_Washing_Machine_Test_Bench.v(62)
#    Time: 65 ps  Iteration: 0  Instance:
/Automatic_Washing_Machine_Test_Bench
```

Conclusion

In conclusion, this project successfully demonstrates the practical application of Verilog Hardware Description Language (HDL) in the development of an Automatic Washing Machine controller. By implementing the controller using a Finite-State Machine (FSM) model, we have showcased the versatility of Verilog HDL in designing complex control systems.