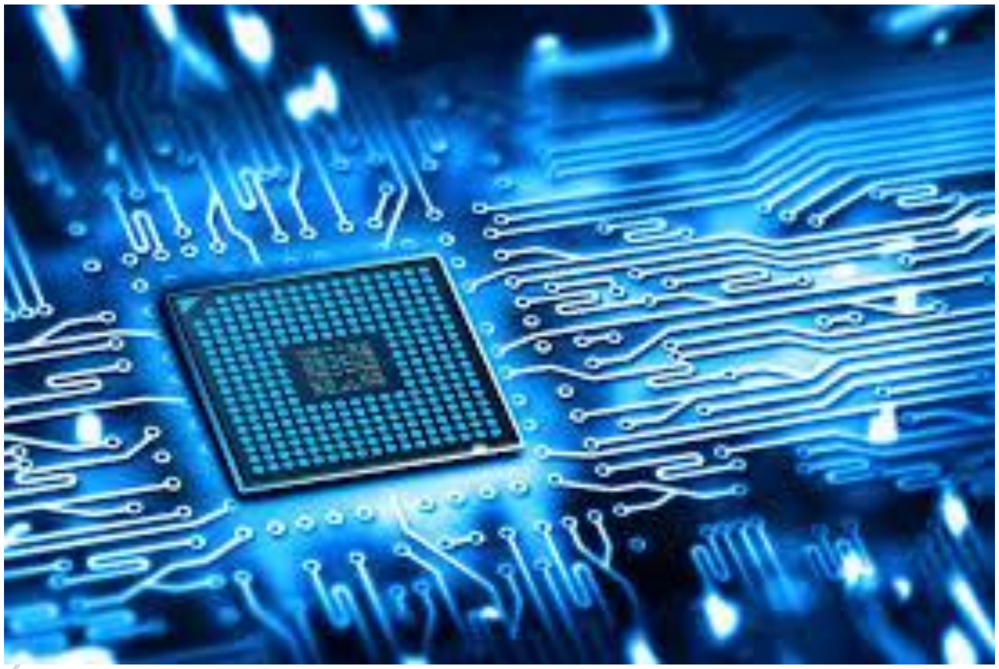
Fall 2024

Washing Machine Controller

EDA project 1



Team 20

CSE312 ELECTRONIC DESIGN AUTOMATION

EDA PROJECT 1: WASHING MACHINE

PRESSENTED BY:

Youssef Moustafa Elsayed	**22P0047**
Mohamed Yasser Khallaf	**22P0245**
Nour El-deen Ahmed Ali Rehab	**22P0040**
Ibrahim Shaker Hammad	**22P0050**
Mostafa Fouad Ahmed	**22P0077**
Abdelrahman Mostafa	**22P0053**
Ahmed Abbady Mohamed	**22P0308**

ABSTRACT

In this project, the aim is to implement a washing machine controller, assume realistic design specifications for a washing machine containing the different options and buttons as the power button, alarms, and other more features. We use a Verilog compiler and QuestaSim that compiles the code and simulates its testbench code.

TABLE OF CONTENTS

LIST OF FIGURES

I. PROPOSED DESIGN SPECIFECATIONS

Objective

The objective of this design is to implement a finite state machine (FSM) that controls the operation of a fully automated washing machine. The design handles tasks such as checking door status, filling water, adding detergent, washing, rinsing, draining, and spinning. It supports user-defined settings such as water temperature and spin speed, ensures safe operations (e.g., door locking), and reacts appropriately to various input signals.

The washing machine controller is implemented as a Moore finite state machine, where the outputs are determined only by the current state. This ensures predictable and stable operations.

Functional Requirements

We want in our design to make a software for the washing machine controller which aims to change between the washing machine states in order to reach the point where the clothes are washed, dried or both according to the inputs (the modes which the user choose). We have 7 states and other options like pausing or stopping:

Door Check State:

- The machine verifies that the door is closed before proceeding.
- If the door is open, the alarm signal is raised, and the machine waits for the door to close.

Water Filling:

- The fill_value_on output is activated to allow water to enter the drum and timer exceeds a certain threshold.
- The water temperature is adjusted based on the waterTemp input, with heater_on controlling the heater.

Detergent Dispensing:

- The soap_value_on output is activated to dispense detergent.
- If detergent is not added (detergentfilled == 0), the machine returns to the check_door state and raises an alarm.

Washing (Cycling):

- The motor starts (motorStart = 1), and its speed is set based on the spinSpeed input.
- The washing cycle continues until timer exceeds a certain threshold which means it finished its cycling.

Rinsing:

- The machine performs a rinse cycle with motor operations.
- If washOnly is set, the machine skips the spin and drying phases after rinsing.

Draining Water:

- The drain_value_on output is activated to empty water from the drum.

Spinning:

- The motor speed is adjusted to the highest level to spin the drum and extract water from clothes. Also it finishes when timer exceeds a certain threshold.

Completion:

- The done signal is asserted, and the door is unlocked after all operations are complete.

Pause Handling:

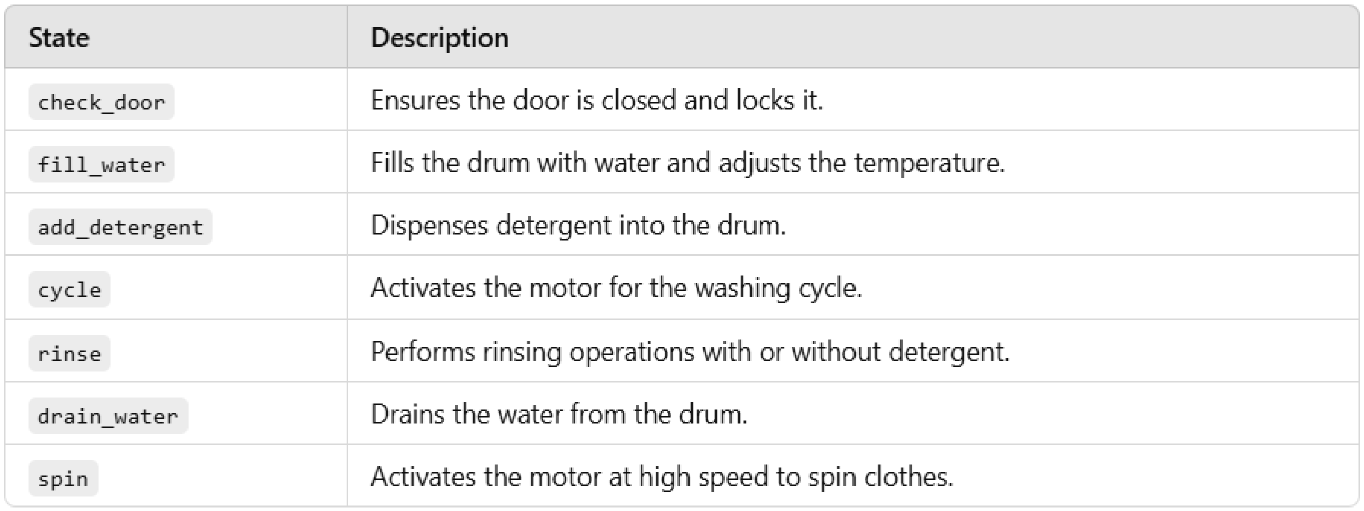
- The machine maintains its current state if the pause signal is activated.

Stop Handling:

- The machine resets to the check_door state when the stop signal is asserted

Timer:

- It is significant to control the time of the state which needs motor or time.



State	Description
`check_door`	Ensures the door is closed and locks it.
`fill_water`	Fills the drum with water and adjusts the temperature.
`add_detergent`	Dispenses detergent into the drum.
`cycle`	Activates the motor for the washing cycle.
`rinse`	Performs rinsing operations with or without detergent.
`drain_water`	Drains the water from the drum.
`spin`	Activates the motor at high speed to spin clothes.

Figure 1: States description

Non-Functional Requirements

It is the functions that are not related to the states of the machine and the constraints we have in it.

Safety:

- The door remains locked (doorLock = 1) during all operations except the check_door state.
- The machine halts operations immediately if the stop signal is triggered.

Robustness:

- The controller handles errors gracefully, such as missing detergent or open door conditions, and signals issues via the alarm output.

Energy Efficiency:

- The heater and motor operate only when required to conserve energy.

Modularity:

- Each state corresponds to a specific function, making the design modular and easy to debug.

Constraints:

- The design should support up to 4 water temperature settings and 4 spin speed levels.
- Inputs must be debounced to handle real-world signal noise.
- Transitions between states should occur synchronously with the clk signal.

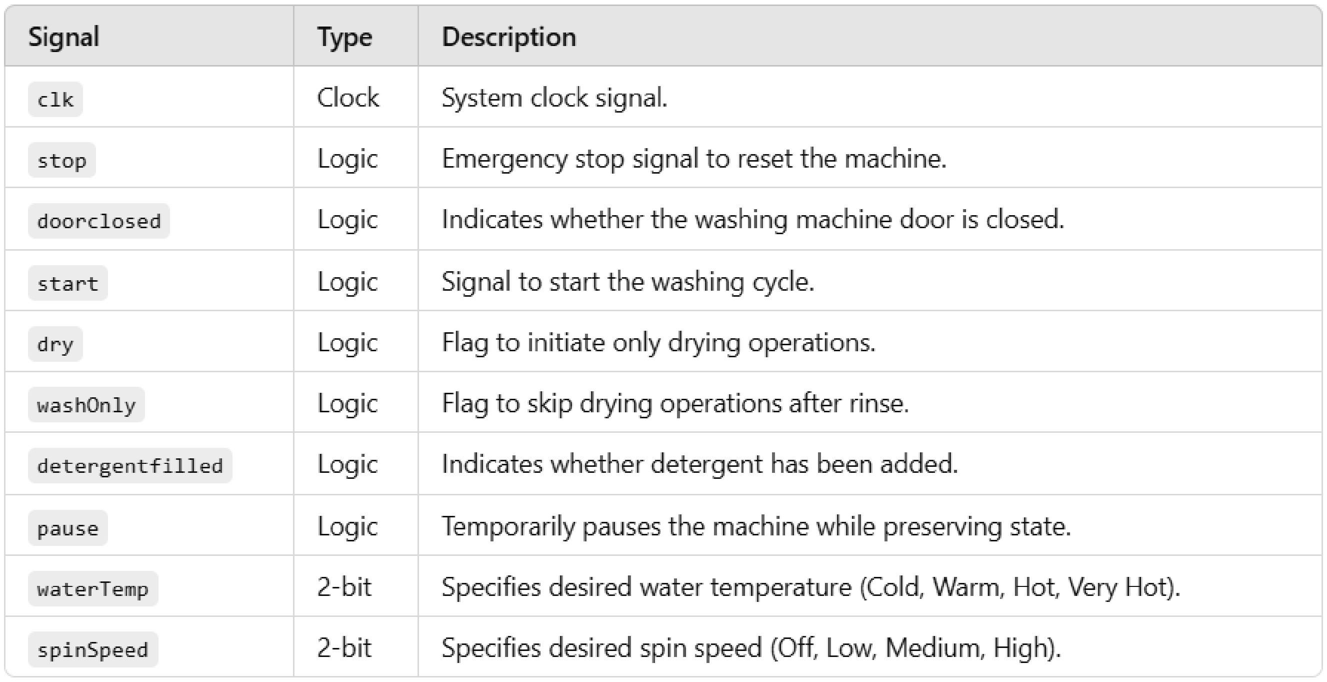
Operation Steps & Expected Behavior

In order to operate the washing machine you follow certain steps. We have a list of inputs which represent either buttons to select the modes in our design or actions you as a user need to make. The inputs are clk, stop, doorclosed, start, pause, dry, washOnly, detergentfilled, waterTemp, spinSpeed.

Clk respresnts clk or plugging it in electricity. Start, stop, pause, dry, washOnly, waterTemp, spinSpeed represent buttons. Start, stop and pause are clear they power on or off or pause it for a while. waterTemp and spinSpeed are used to choose the temperature of water and speed of motor that spins the clothes respectively. The default value of waterTemp is warm while the spinSpeed have default values according to the state the machine is currently in. Finally dry and washOnly are used to choose the modes of the washing machine. Making dry and washOnly equal zero means the washing machine operates normally where it washes then dries the clothes. Making dry 1 makes the washing machine dry the clothes only without washing them, it considers you already have your clothes washed and want to make an extra drying step. Making washOnly equal 1 and dry zero makes the machine wash the clothes and stop at the rinsing step.

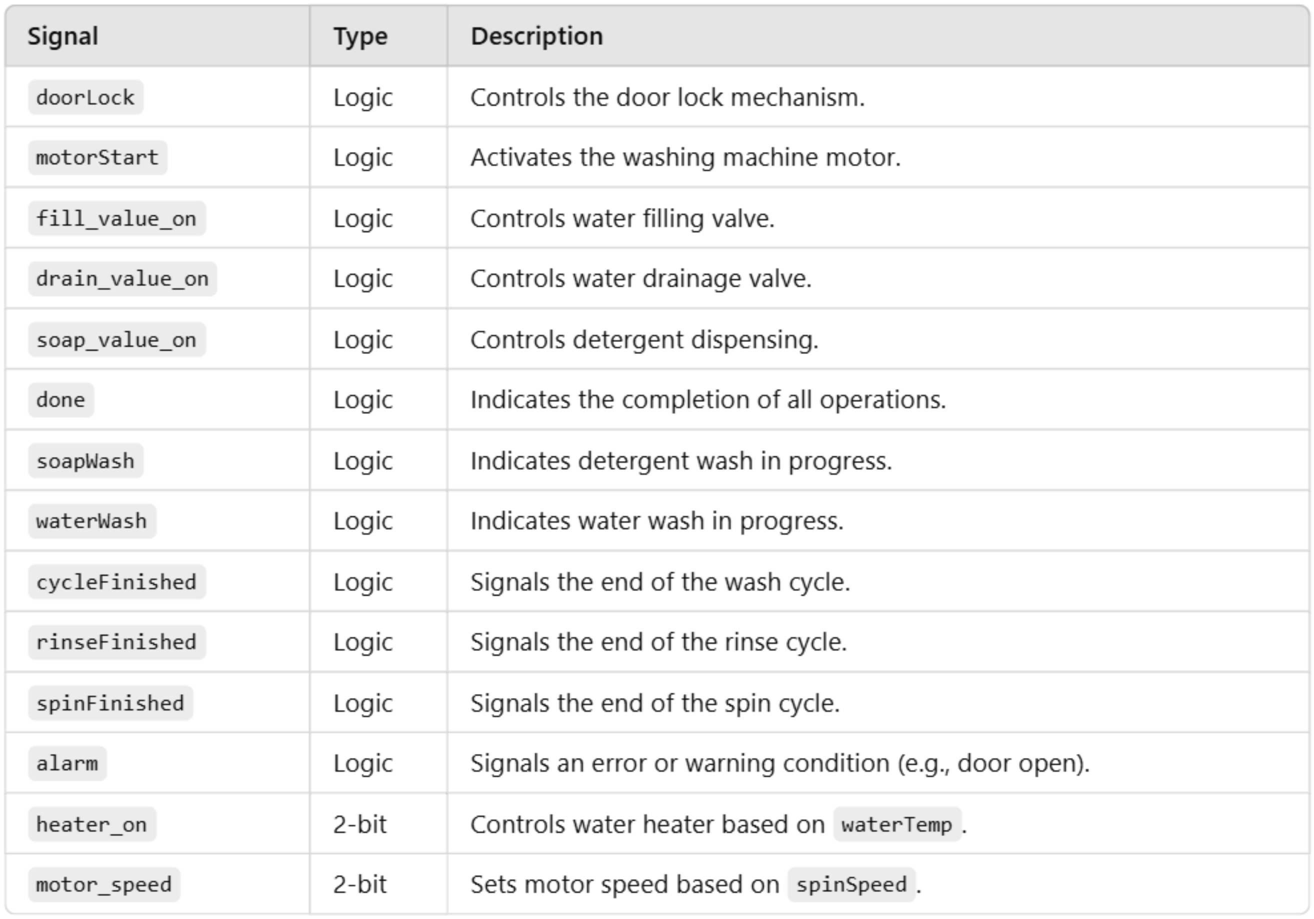
Doorclosed and detergentfilled are logic actions the user must make. Doorclosed means you have to close door before starting to ensure it works normally. Detergentfilled means you have to put soap before starting.

The machine starts when the start signal is activated, provided the door is closed. The machine progresses through the predefined states, ensuring smooth transitions and meeting user-specified settings. Any abnormal condition (e.g., door open or missing detergent) triggers an alarm and pauses operations until resolved.



Signal	Type	Description
`clk`	Clock	System clock signal.
`stop`	Logic	Emergency stop signal to reset the machine.
`doorclosed`	Logic	Indicates whether the washing machine door is closed.
`start`	Logic	Signal to start the washing cycle.
`dry`	Logic	Flag to initiate only drying operations.
`washOnly`	Logic	Flag to skip drying operations after rinse.
`detergentfilled`	Logic	Indicates whether detergent has been added.
`pause`	Logic	Temporarily pauses the machine while preserving state.
`waterTemp`	2-bit	Specifies desired water temperature (Cold, Warm, Hot, Very Hot).
`spinSpeed`	2-bit	Specifies desired spin speed (Off, Low, Medium, High).

Figure 2: Inputs description



Signal	Type	Description
`doorLock`	Logic	Controls the door lock mechanism.
`motorStart`	Logic	Activates the washing machine motor.
`fill_value_on`	Logic	Controls water filling valve.
`drain_value_on`	Logic	Controls water drainage valve.
`soap_value_on`	Logic	Controls detergent dispensing.
`done`	Logic	Indicates the completion of all operations.
`soapWash`	Logic	Indicates detergent wash in progress.
`waterWash`	Logic	Indicates water wash in progress.
`cycleFinished`	Logic	Signals the end of the wash cycle.
`rinseFinished`	Logic	Signals the end of the rinse cycle.
`spinFinished`	Logic	Signals the end of the spin cycle.
`alarm`	Logic	Signals an error or warning condition (e.g., door open).
`heater_on`	2-bit	Controls water heater based on `waterTemp` .
`motor_speed`	2-bit	Sets motor speed based on `spinSpeed` .

Figure 3: Outputs description

II. FSM STATE DIAGRAM

The finite state machine diagram represents the states and transitions as follows:

1. **Initial State (check_door)**: Start of the process, decision-making based on doorclosed and start.
2. Transition arrows between states:
    - From **check_door**:
        - To **fill_water** (if start = 1 and doorclosed = 1 and detergentfilled = 1 and stop !=1 and not dry).
        - To **drain_water** (if dry = 1).
        - Loopback if conditions unmet.
    - From **fill_water**:
        - To **add_detergent** (when fill_value_on = 1 and timer > threshold).
        - Loopback if water is still being filled.
    - From **add_detergent**:
        - To **cycle** (if detergentfilled = 1 and soap_value_on = 1 and timer >threshold)
        - Loopback if soap is still being filled.
    - From **cycle**:
        - To **rinse** (when cycleFinished = 1 and timer > threshold).
        - Loopback if cycles still haven't finished.
    - From **rinse**:
        - To **drain_water** (if rinse completes and timer > threshold).
        - Loopback if the clothes still haven't been rinsed properly.
        - To **check_door** (if washOnly = 1).
    - From **drain_water**:
        - To **spin** (if water is drained and timer > threshold).
        - Loopback if the clothes still haven't been dried.
    - From **spin**:

- To **check_door** (when spinning finishes and timer > threshold).
- Loopback if the spin to continue draining is not finished.

3. **Final State**: Achieved after returning to check_door with done = 1.

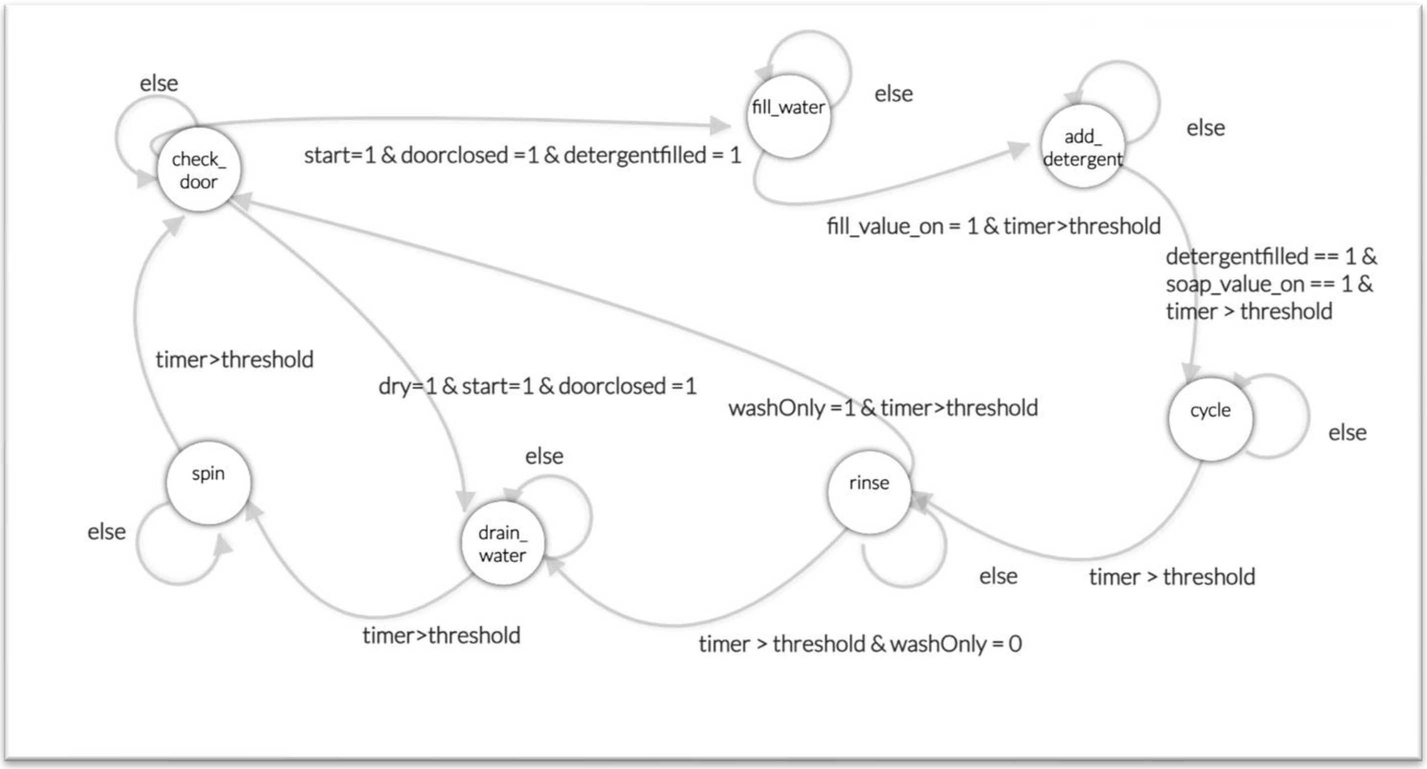

else
check_door
start=1 & doorclosed =1 & detergentfilled = 1
fill_water
else
add_detergent
else
fill_value_on = 1 & timer>threshold
detergentfilled == 1 & soap_value_on == 1 & timer > threshold
cycle
else
timer>threshold
dry=1 & start=1 & doorclosed =1
washOnly =1 & timer>threshold
spin
else
else
drain_water
rinse
else
timer > threshold
timer>threshold
timer > threshold & washOnly = 0

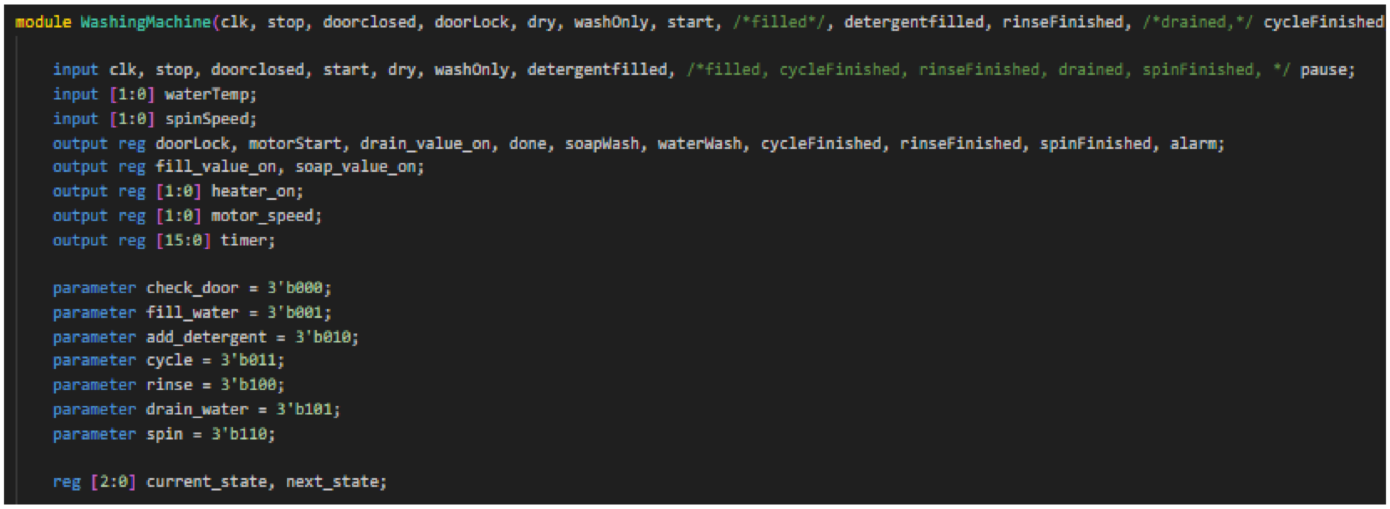
Figure 4: FSM diagram

III. DESIGN RTL CODE

We will move on the most important parts in the code and try to explain it more, for the full code you can preview it from the zip file.

1. FSM Encoding

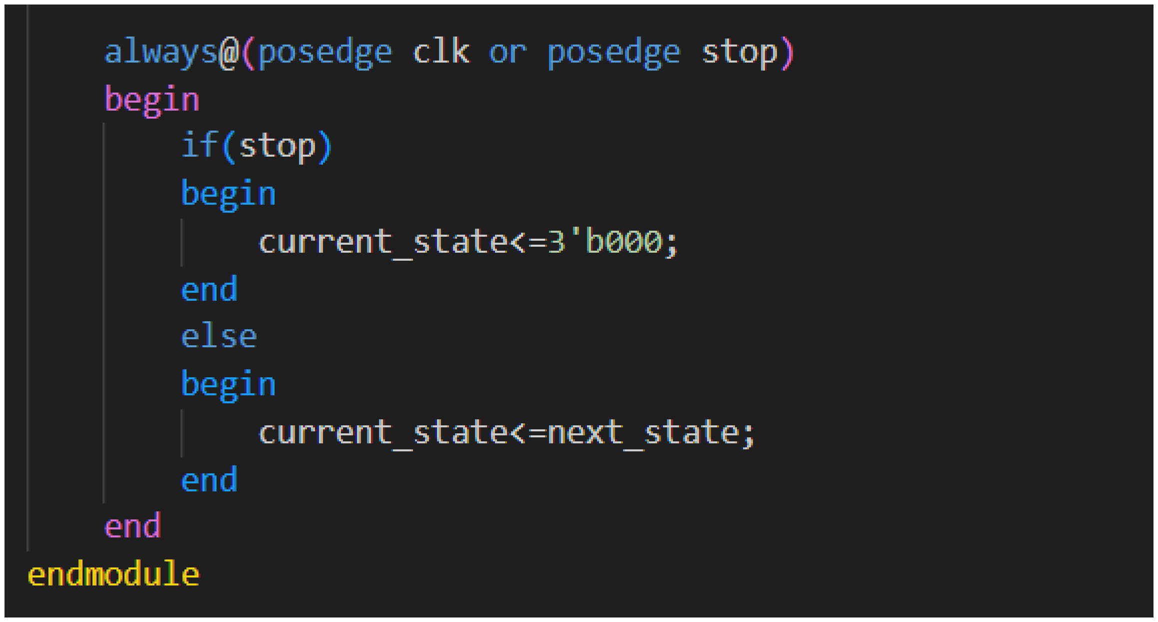
Here we put the inputs, outputs & encode the parameters



```
module WashingMachine(clk, stop, doorclosed, doorLock, dry, washOnly, start, /*filled*/, detergentfilled, rinseFinished, /*drained,*/ cycleFinished

    input clk, stop, doorclosed, start, dry, washOnly, detergentfilled, /*filled, cycleFinished, rinseFinished, drained, spinFinished, */ pause;
    input [1:0] waterTemp;
    input [1:0] spinSpeed;
    output reg doorLock, motorStart, drain_value_on, done, soapWash, waterWash, cycleFinished, rinseFinished, spinFinished, alarm;
    output reg fill_value_on, soap_value_on;
    output reg [1:0] heater_on;
    output reg [1:0] motor_speed;
    output reg [15:0] timer;

    parameter check_door = 3'b000;
    parameter fill_water = 3'b001;
    parameter add_detergent = 3'b010;
    parameter cycle = 3'b011;
    parameter rinse = 3'b100;
    parameter drain_water = 3'b101;
    parameter spin = 3'b110;

    reg [2:0] current_state, next_state;
```

2. State transition

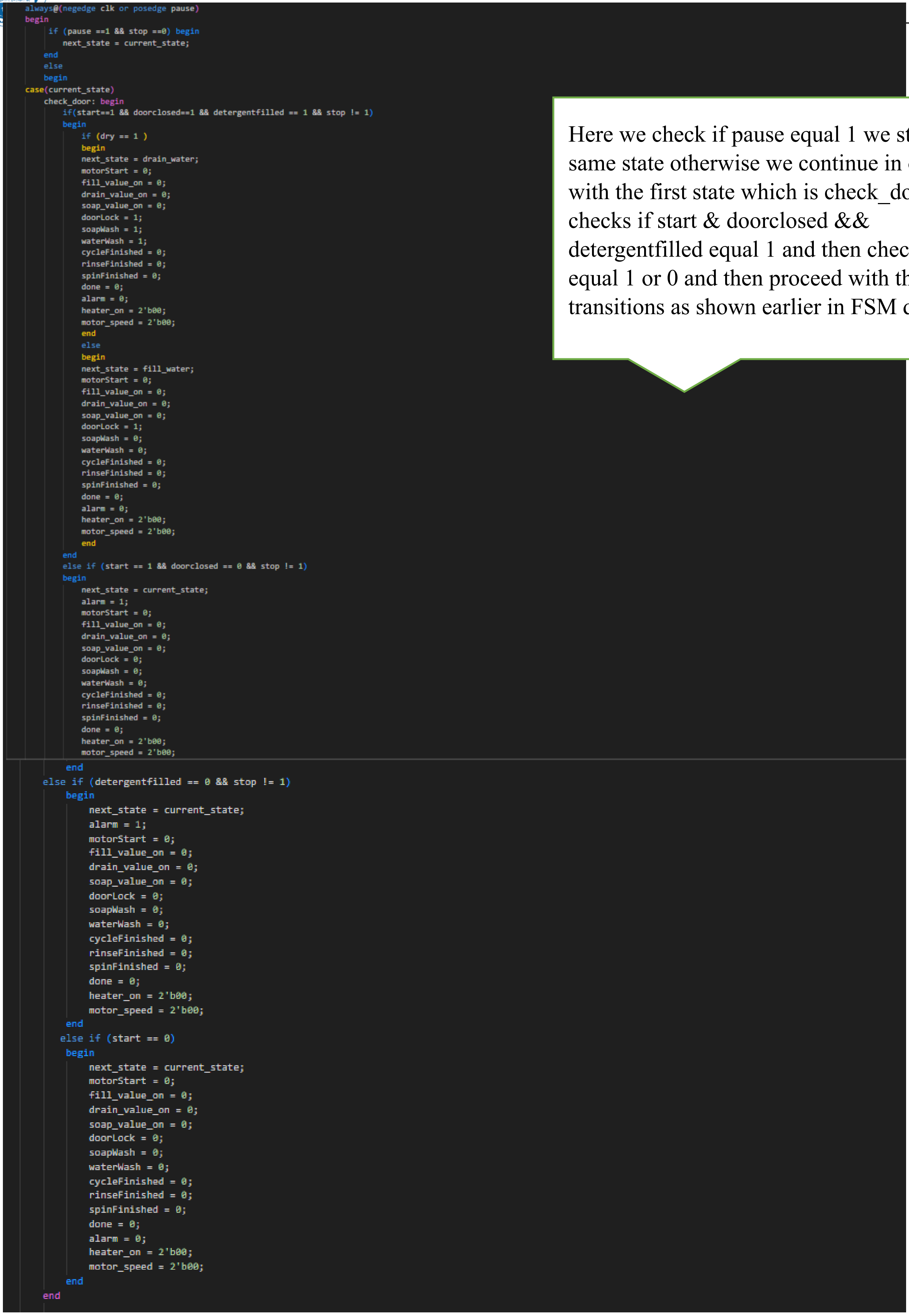
Here we use the clock to update the current_state to next_state, this is the sequential part. If stop equal 1 we return to the initial state which is check_door.



```
    always@(posedge clk or posedge stop)
    begin
        if(stop)
        begin
            current_state<=3'b000;
        end
        else
        begin
            current_state<=next_state;
        end
    end
endmodule
```

3. Next state & Output logic

Here we see the transitions from one state to another as explained in the FSM diagram and how next_state is computed based on the current state and inputs (this is usually inside a combinational block). Also, we define the output based on the current state.

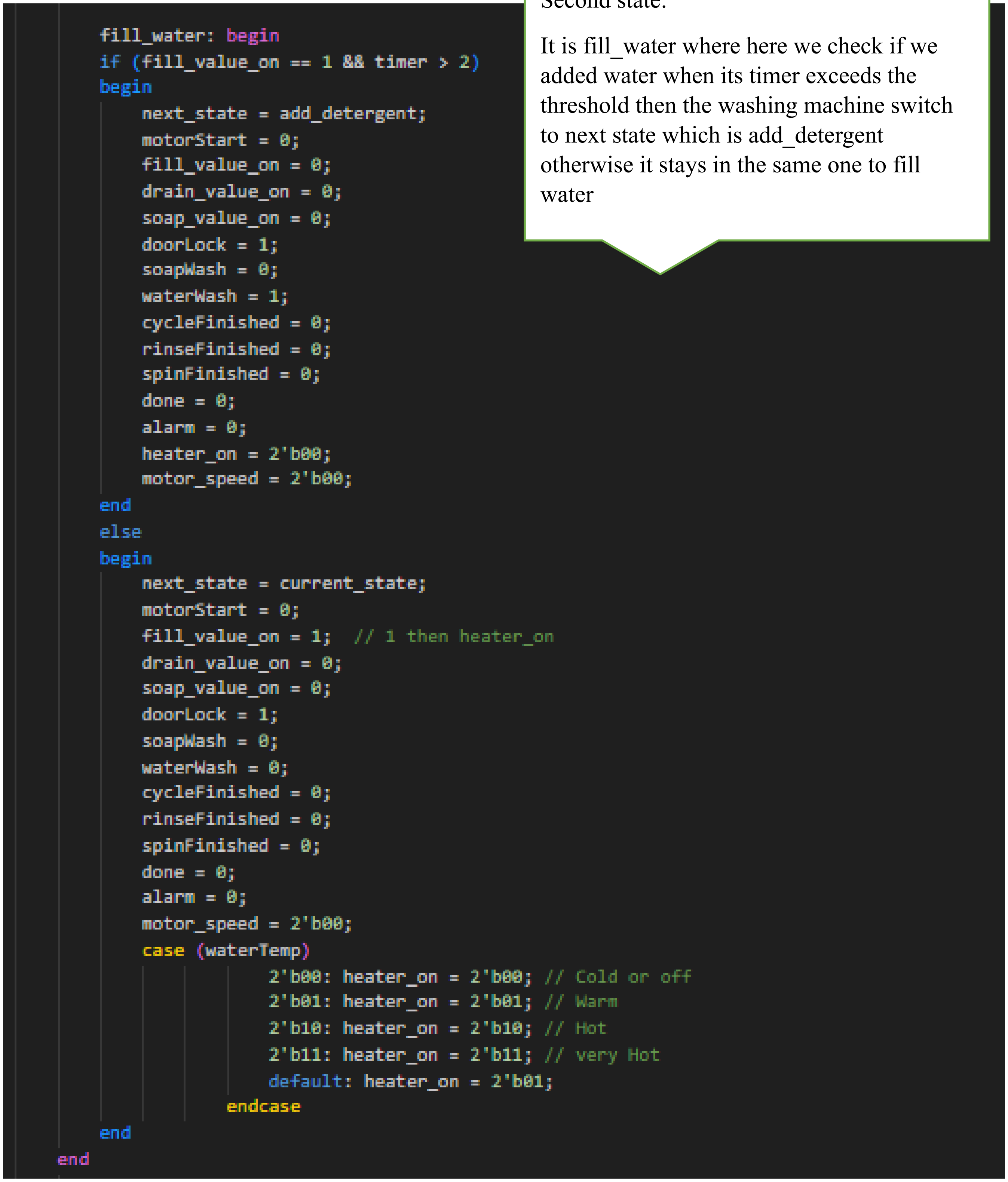


```
always@(negedge clk or posedge pause)
begin
     if (pause ==1 && stop ==0) begin
        next_state = current_state;
    end
    else
    begin
case(current_state)
    check_door: begin
        if(start==1 && doorclosed==1 && detergentfilled == 1 && stop != 1)
        begin
            if (dry == 1 )
            begin
            next_state = drain_water;
            motorStart = 0;
            fill_value_on = 0;
            drain_value_on = 0;
            soap_value_on = 0;
            doorLock = 1;
            soapWash = 1;
            waterWash = 1;
            cycleFinished = 0;
            rinseFinished = 0;
            spinFinished = 0;
            done = 0;
            alarm = 0;
            heater_on = 2'b00;
            motor_speed = 2'b00;
            end
            else
            begin
            next_state = fill_water;
            motorStart = 0;
            fill_value_on = 0;
            drain_value_on = 0;
            soap_value_on = 0;
            doorLock = 1;
            soapWash = 0;
            waterWash = 0;
            cycleFinished = 0;
            rinseFinished = 0;
            spinFinished = 0;
            done = 0;
            alarm = 0;
            heater_on = 2'b00;
            motor_speed = 2'b00;
            end
        end
        else if (start == 1 && doorclosed == 0 && stop != 1)
        begin
            next_state = current_state;
            alarm = 1;
            motorStart = 0;
            fill_value_on = 0;
            drain_value_on = 0;
            soap_value_on = 0;
            doorLock = 0;
            soapWash = 0;
            waterWash = 0;
            cycleFinished = 0;
            rinseFinished = 0;
            spinFinished = 0;
            done = 0;
            heater_on = 2'b00;
            motor_speed = 2'b00;
        end
    else if (detergentfilled == 0 && stop != 1)
        begin
            next_state = current_state;
            alarm = 1;
            motorStart = 0;
            fill_value_on = 0;
            drain_value_on = 0;
            soap_value_on = 0;
            doorLock = 0;
            soapWash = 0;
            waterWash = 0;
            cycleFinished = 0;
            rinseFinished = 0;
            spinFinished = 0;
            done = 0;
            heater_on = 2'b00;
            motor_speed = 2'b00;
        end
       else if (start == 0)
        begin
            next_state = current_state;
            motorStart = 0;
            fill_value_on = 0;
            drain_value_on = 0;
            soap_value_on = 0;
            doorLock = 0;
            soapWash = 0;
            waterWash = 0;
            cycleFinished = 0;
            rinseFinished = 0;
            spinFinished = 0;
            done = 0;
            alarm = 0;
            heater_on = 2'b00;
            motor_speed = 2'b00;
        end
    end
```

Here we check if pause equal 1 we stay in same state otherwise we continue in our fsm with the first state which is check_door that checks if start & doorclosed && detergentfilled equal 1 and then checks if dry equal 1 or 0 and then proceed with the transitions as shown earlier in FSM diagram

Second state:

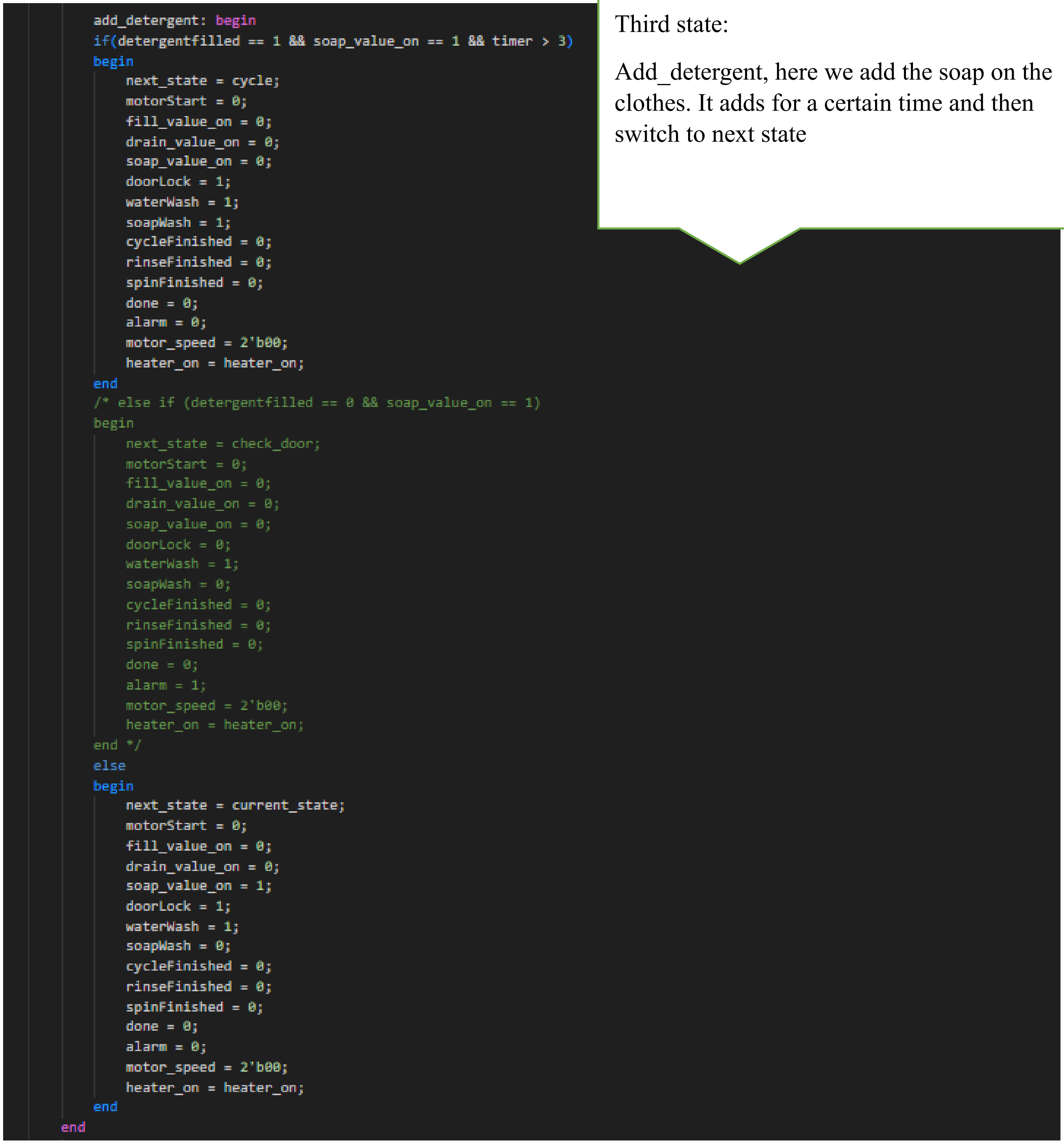
It is fill_water where here we check if we added water when its timer exceeds the threshold then the washing machine switch to next state which is add_detergent otherwise it stays in the same one to fill water



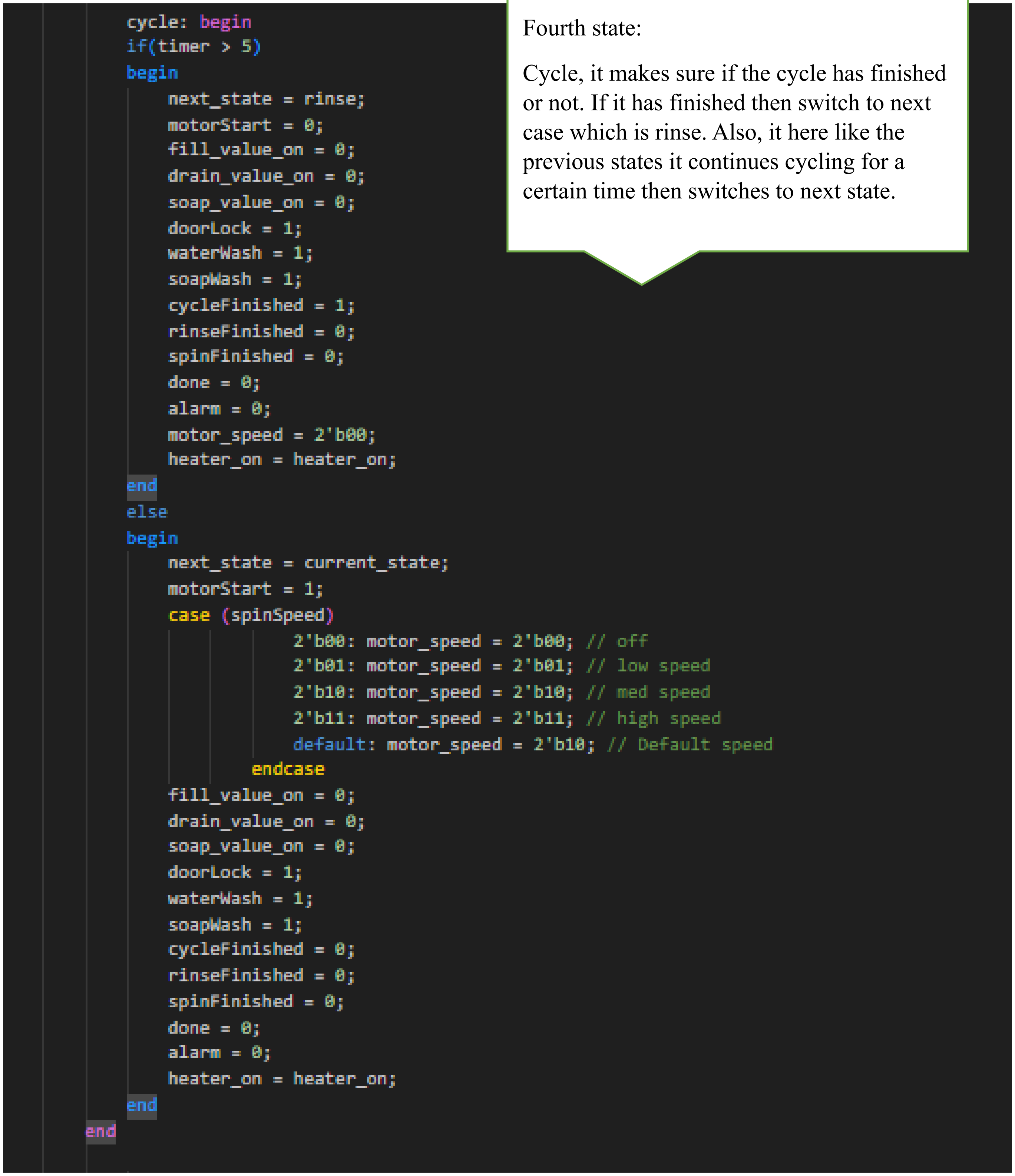
```
fill_water: begin
if (fill_value_on == 1 && timer > 2)
begin
    next_state = add_detergent;
    motorStart = 0;
    fill_value_on = 0;
    drain_value_on = 0;
    soap_value_on = 0;
    doorLock = 1;
    soapWash = 0;
    waterWash = 1;
    cycleFinished = 0;
    rinseFinished = 0;
    spinFinished = 0;
    done = 0;
    alarm = 0;
    heater_on = 2'b00;
    motor_speed = 2'b00;
end
else
begin
    next_state = current_state;
    motorStart = 0;
    fill_value_on = 1;  // 1 then heater_on
    drain_value_on = 0;
    soap_value_on = 0;
    doorLock = 1;
    soapWash = 0;
    waterWash = 0;
    cycleFinished = 0;
    rinseFinished = 0;
    spinFinished = 0;
    done = 0;
    alarm = 0;
    motor_speed = 2'b00;
    case (waterTemp)
            2'b00: heater_on = 2'b00; // Cold or off
            2'b01: heater_on = 2'b01; // Warm
            2'b10: heater_on = 2'b10; // Hot
            2'b11: heater_on = 2'b11; // very Hot
            default: heater_on = 2'b01;
        endcase
end
end
```

Third state:

Add_detergent, here we add the soap on the clothes. It adds for a certain time and then switch to next state



```
        add_detergent: begin
        if(detergentfilled == 1 && soap_value_on == 1 && timer > 3)
        begin
            next_state = cycle;
            motorStart = 0;
            fill_value_on = 0;
            drain_value_on = 0;
            soap_value_on = 0;
            doorLock = 1;
            waterWash = 1;
            soapWash = 1;
            cycleFinished = 0;
            rinseFinished = 0;
            spinFinished = 0;
            done = 0;
            alarm = 0;
            motor_speed = 2'b00;
            heater_on = heater_on;
        end
        /* else if (detergentfilled == 0 && soap_value_on == 1)
        begin
            next_state = check_door;
            motorStart = 0;
            fill_value_on = 0;
            drain_value_on = 0;
            soap_value_on = 0;
            doorLock = 0;
            waterWash = 1;
            soapWash = 0;
            cycleFinished = 0;
            rinseFinished = 0;
            spinFinished = 0;
            done = 0;
            alarm = 1;
            motor_speed = 2'b00;
            heater_on = heater_on;
        end */
        else
        begin
            next_state = current_state;
            motorStart = 0;
            fill_value_on = 0;
            drain_value_on = 0;
            soap_value_on = 1;
            doorLock = 1;
            waterWash = 1;
            soapWash = 0;
            cycleFinished = 0;
            rinseFinished = 0;
            spinFinished = 0;
            done = 0;
            alarm = 0;
            motor_speed = 2'b00;
            heater_on = heater_on;
        end
    end
```



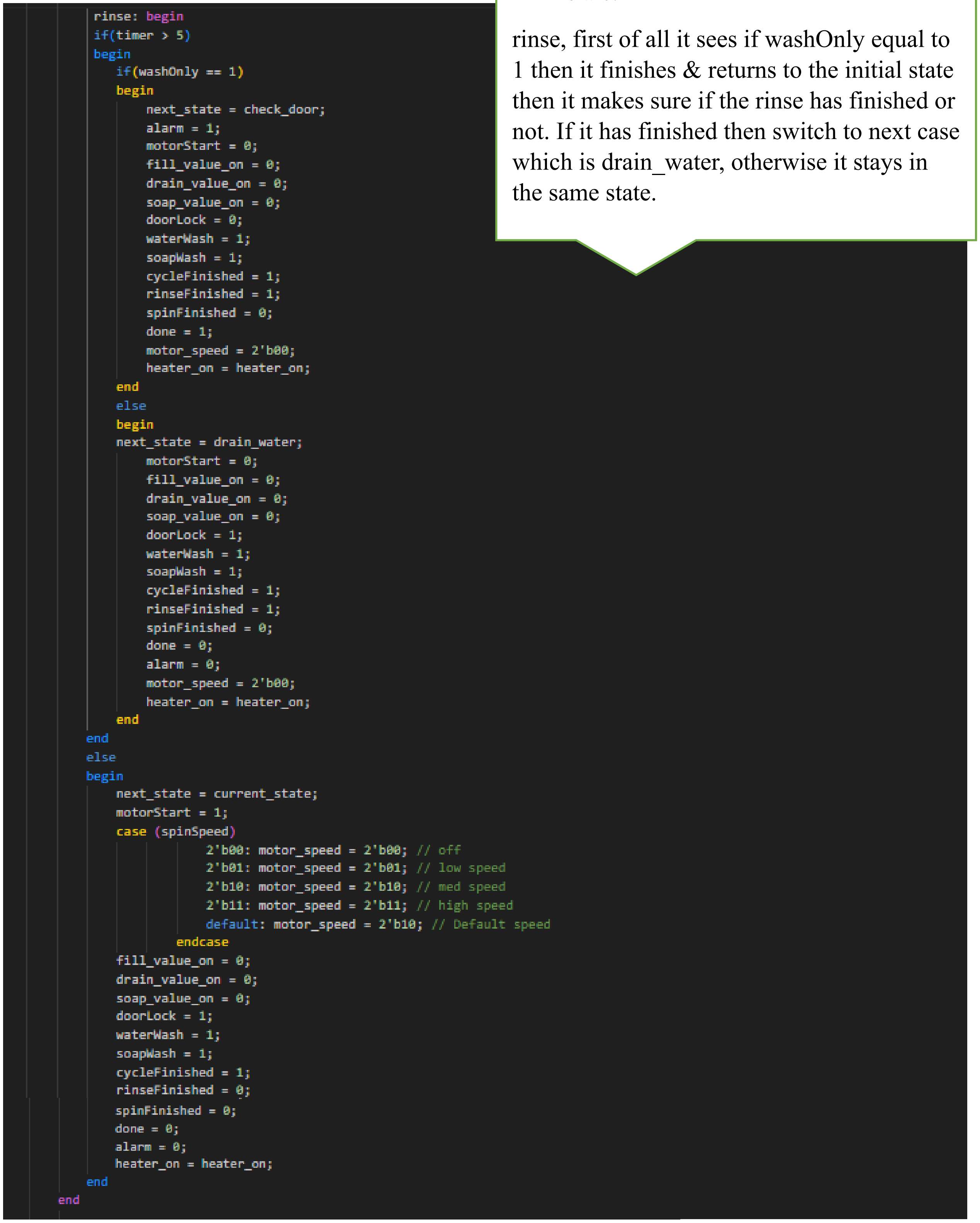
```
cycle: begin
if(timer > 5)
begin
    next_state = rinse;
    motorStart = 0;
    fill_value_on = 0;
    drain_value_on = 0;
    soap_value_on = 0;
    doorLock = 1;
    waterWash = 1;
    soapWash = 1;
    cycleFinished = 1;
    rinseFinished = 0;
    spinFinished = 0;
    done = 0;
    alarm = 0;
    motor_speed = 2'b00;
    heater_on = heater_on;
end
else
begin
    next_state = current_state;
    motorStart = 1;
    case (spinSpeed)
            2'b00: motor_speed = 2'b00; // off
            2'b01: motor_speed = 2'b01; // low speed
            2'b10: motor_speed = 2'b10; // med speed
            2'b11: motor_speed = 2'b11; // high speed
            default: motor_speed = 2'b10; // Default speed
        endcase
    fill_value_on = 0;
    drain_value_on = 0;
    soap_value_on = 0;
    doorLock = 1;
    waterWash = 1;
    soapWash = 1;
    cycleFinished = 0;
    rinseFinished = 0;
    spinFinished = 0;
    done = 0;
    alarm = 0;
    heater_on = heater_on;
end
end
```

Fourth state:

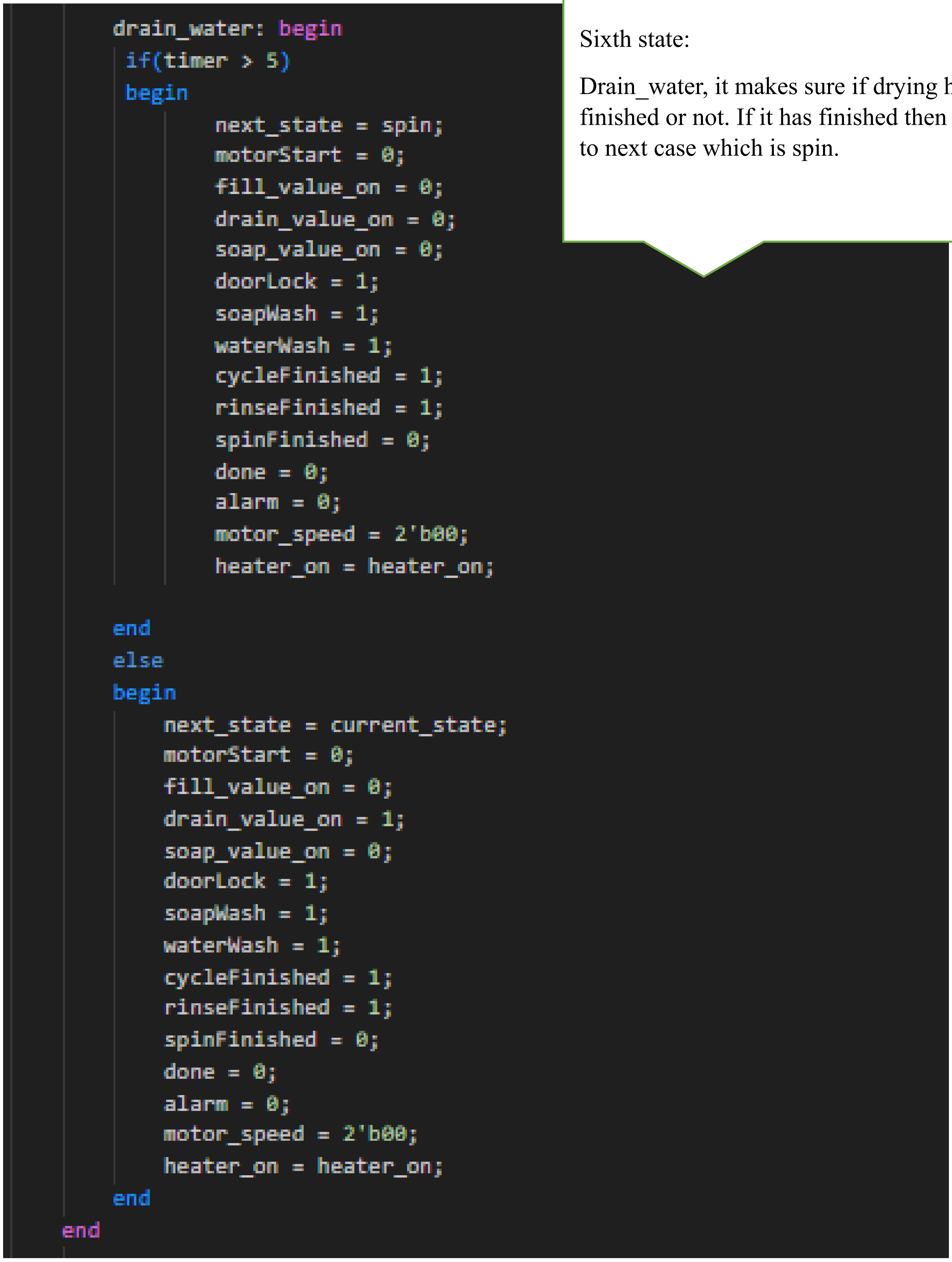
Cycle, it makes sure if the cycle has finished or not. If it has finished then switch to next case which is rinse. Also, it here like the previous states it continues cycling for a certain time then switches to next state.

Fifth state:

rinse, first of all it sees if washOnly equal to 1 then it finishes & returns to the initial state then it makes sure if the rinse has finished or not. If it has finished then switch to next case which is drain_water, otherwise it stays in the same state.



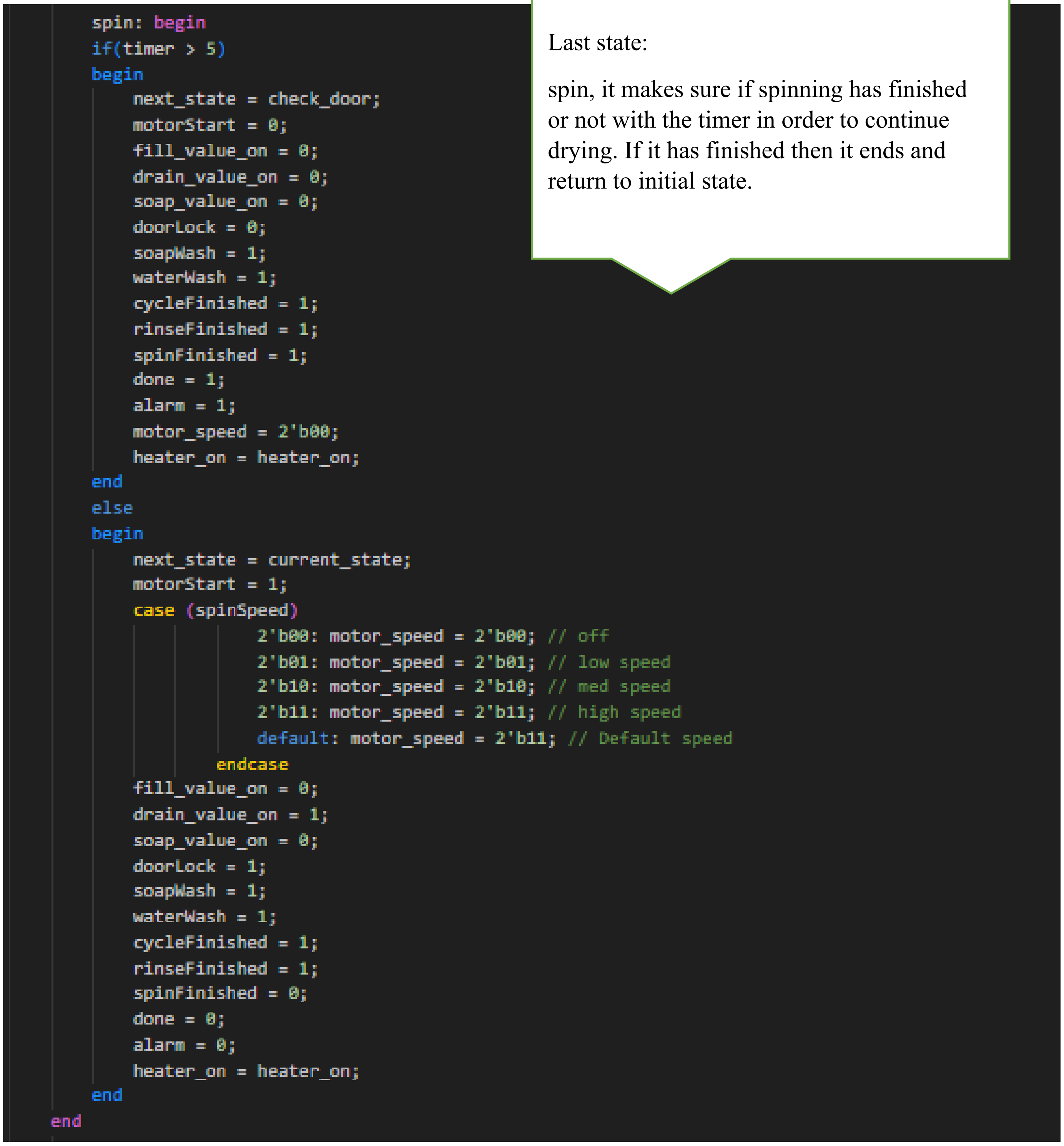
```
    rinse: begin
    if(timer > 5)
    begin
        if(washOnly == 1)
        begin
            next_state = check_door;
            alarm = 1;
            motorStart = 0;
            fill_value_on = 0;
            drain_value_on = 0;
            soap_value_on = 0;
            doorLock = 0;
            waterWash = 1;
            soapWash = 1;
            cycleFinished = 1;
            rinseFinished = 1;
            spinFinished = 0;
            done = 1;
            motor_speed = 2'b00;
            heater_on = heater_on;
        end
        else
        begin
        next_state = drain_water;
            motorStart = 0;
            fill_value_on = 0;
            drain_value_on = 0;
            soap_value_on = 0;
            doorLock = 1;
            waterWash = 1;
            soapWash = 1;
            cycleFinished = 1;
            rinseFinished = 1;
            spinFinished = 0;
            done = 0;
            alarm = 0;
            motor_speed = 2'b00;
            heater_on = heater_on;
        end
    end
    else
    begin
        next_state = current_state;
        motorStart = 1;
        case (spinSpeed)
                    2'b00: motor_speed = 2'b00; // off
                    2'b01: motor_speed = 2'b01; // low speed
                    2'b10: motor_speed = 2'b10; // med speed
                    2'b11: motor_speed = 2'b11; // high speed
                    default: motor_speed = 2'b10; // Default speed
                endcase
        fill_value_on = 0;
        drain_value_on = 0;
        soap_value_on = 0;
        doorLock = 1;
        waterWash = 1;
        soapWash = 1;
        cycleFinished = 1;
        rinseFinished = 0;
        spinFinished = 0;
        done = 0;
        alarm = 0;
        heater_on = heater_on;
    end
end
```



```
drain_water: begin
 if(timer > 5)
 begin
        next_state = spin;
        motorStart = 0;
        fill_value_on = 0;
        drain_value_on = 0;
        soap_value_on = 0;
        doorLock = 1;
        soapWash = 1;
        waterWash = 1;
        cycleFinished = 1;
        rinseFinished = 1;
        spinFinished = 0;
        done = 0;
        alarm = 0;
        motor_speed = 2'b00;
        heater_on = heater_on;

end
else
begin
    next_state = current_state;
    motorStart = 0;
    fill_value_on = 0;
    drain_value_on = 1;
    soap_value_on = 0;
    doorLock = 1;
    soapWash = 1;
    waterWash = 1;
    cycleFinished = 1;
    rinseFinished = 1;
    spinFinished = 0;
    done = 0;
    alarm = 0;
    motor_speed = 2'b00;
    heater_on = heater_on;
end
end
```

Sixth state:

Drain_water, it makes sure if drying has finished or not. If it has finished then switch to next case which is spin.



```
spin: begin
if(timer > 5)
begin
    next_state = check_door;
    motorStart = 0;
    fill_value_on = 0;
    drain_value_on = 0;
    soap_value_on = 0;
    doorLock = 0;
    soapWash = 1;
    waterWash = 1;
    cycleFinished = 1;
    rinseFinished = 1;
    spinFinished = 1;
    done = 1;
    alarm = 1;
    motor_speed = 2'b00;
    heater_on = heater_on;
end
else
begin
    next_state = current_state;
    motorStart = 1;
    case (spinSpeed)
                2'b00: motor_speed = 2'b00; // off
                2'b01: motor_speed = 2'b01; // low speed
                2'b10: motor_speed = 2'b10; // med speed
                2'b11: motor_speed = 2'b11; // high speed
                default: motor_speed = 2'b11; // Default speed
            endcase
    fill_value_on = 0;
    drain_value_on = 1;
    soap_value_on = 0;
    doorLock = 1;
    soapWash = 1;
    waterWash = 1;
    cycleFinished = 1;
    rinseFinished = 1;
    spinFinished = 0;
    done = 0;
    alarm = 0;
    heater_on = heater_on;
end
end
```

Last state:

spin, it makes sure if spinning has finished or not with the timer in order to continue drying. If it has finished then it ends and return to initial state.

IV. Verification PLAN

The purpose of this verification plan is to validate the functional and non-functional requirements of the washing machine controller FSM. The plan ensures the design transitions between states correctly, handles inputs appropriately, and generates the expected outputs.

In this verification plan, we used direct testing, random testing, and assertions as follows:

Direct testing:

In direct testing, we manually design test cases which target specific scenarios, states, transitions. At the beginning, we created a task to initialize all the inputs when needed, and then we started changing the inputs according to the case to be tested

In test 1, we simulate a normal start scenario where the user initiates the washing machine with detergent added and default settings for water temperature and spin speed

In test 2, we pause the washing machine mid-operation and then resumes it, which ensures that the FSM remains in its current state when paused and then resumes normal operations when not paused

In test 3, we test stopping the washing machine operation and then restarting it.

In test 4, we attempt to start the washing machine without detergent being filled.

In test 5, we test activating the dry mode to skip washing and rinsing, proceeding directly to drain water and spin cycles.

In test 6, we test activating the washOnly mode to stop after rinsing without proceeding to the spin cycle.

Test 7, we intentionally make doorclosed equal to 0 to see the reaction.

Random testing:

In random testing, we generate random sequences of inputs to explore the FSM's behavior under various conditions. This approach helps uncover hidden bugs that direct testing may not cover.

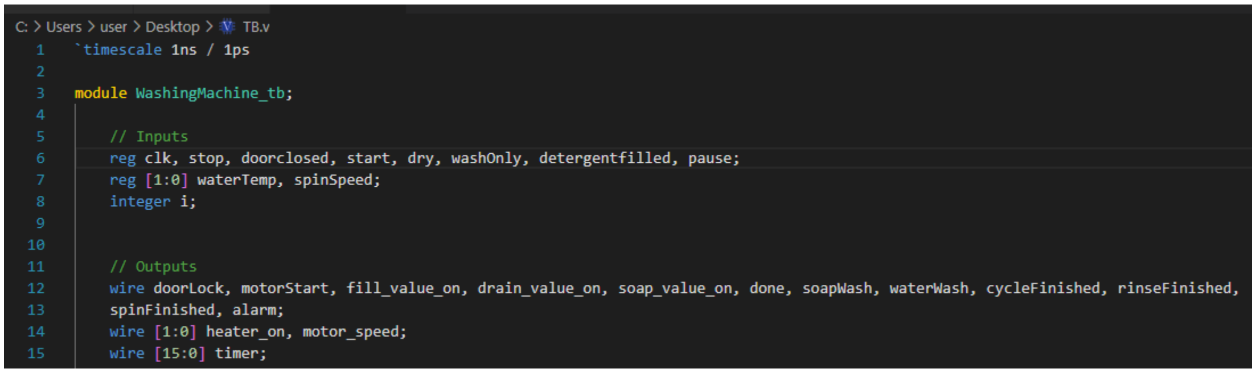
Assertions:

Assertions ensure that the FSM behaves as expected. They can be monitor properties, validate state transitions, and catch unexpected behavior during simulation. We used PSL for assertions since it allows for formal descriptions of sequences and state transitions, which would be lengthy with explicit procedural logic. At first, we assign the default clock as the rising edge of the clock, which is required for timing analysis . We then made the following assertions:

Assertion number	**Description**
1	Ensures that if the pause signal is active and stop is not asserted, the system's next_state remains the same as current_state.
2	Ensures that when the stop signal is active, the current_state is reset to 3'b000
3	Ensures that if the start, doorclosed, and detergentfilled signals are active and system is still working, the doorLock signal is active.
4	Ensures the system transitions from check_door to drain_water when conditions such as doorclosed, dry, and others are met.
5	Ensures the system transitions from check_door to fill_water when dry is 0 and other conditions are satisfied.
6	Ensures a transition from fill_water to add_detergent occurs after a timer exceeds 2 and stop is not asserted.
7	Ensures the transition from add_detergent to cycle happens if the timer exceeds 3, detergent is filled, and stop is not asserted.
8	Ensures the transition from cycle to rinse happens when the timer exceeds 5 and stop is not active.
9	Ensures a transition from rinse to check_door when washOnly is active, the timer exceeds 5, and stop is not asserted.
10	Ensures a transition from rinse to drain_water when washOnly is inactive, the timer exceeds 5, and stop is not asserted.
11	Ensures a transition from drain_water to spin happens when the timer exceeds 5 and stop is not asserted.
12	Ensures the transition from spin to check_door occurs when the timer exceeds 5 and pause is not asserted.
13	Ensures the alarm is activated when the done signal is set, as long as stop is not asserted.
14	Ensures that when transitioning from rinse or spin to check_door, the done signal is asserted.
15	Ensures the alarm is activated if detergentfilled is not active and stop is not asserted.
16	Ensures the alarm is activated if doorclosed is not active and stop is not asserted.

V. TESTBENCH CODE

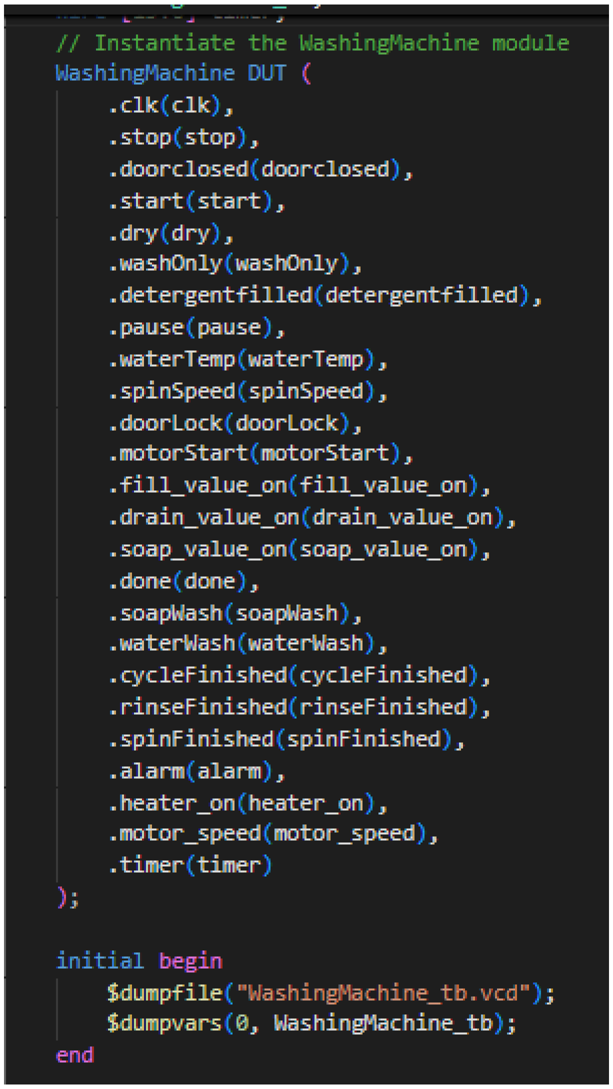
Inputs and outputs:



C: > Users > user > Desktop > TB.v

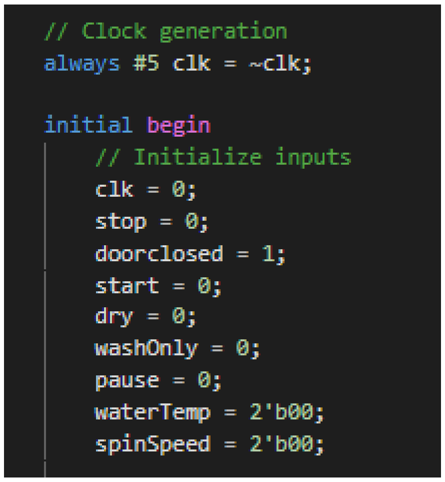
```
`timescale 1ns / 1ps

module WashingMachine_tb;

    // Inputs
    reg clk, stop, doorclosed, start, dry, washOnly, detergentfilled, pause;
    reg [1:0] waterTemp, spinSpeed;
    integer i;


    // Outputs
    wire doorLock, motorStart, fill_value_on, drain_value_on, soap_value_on, done, soapWash, waterWash, cycleFinished, rinseFinished,
    spinFinished, alarm;
    wire [1:0] heater_on, motor_speed;
    wire [15:0] timer;
```

Instantiation:



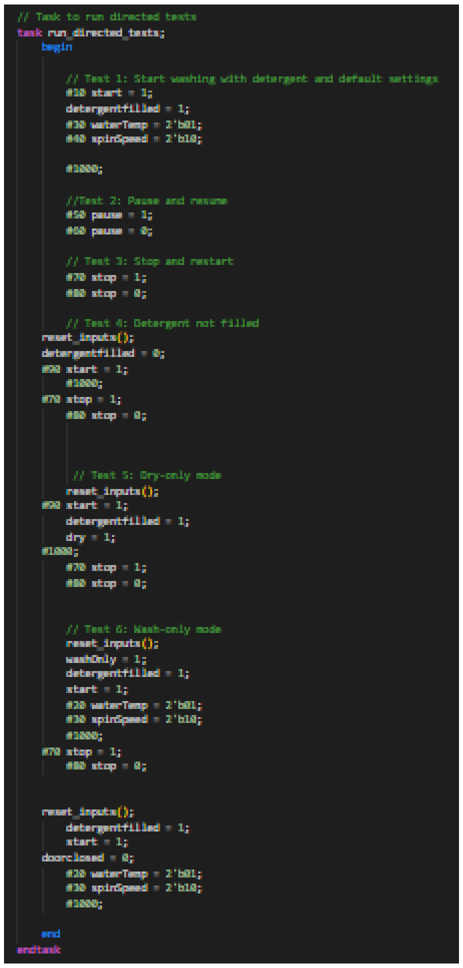
```
// Instantiate the WashingMachine module
WashingMachine DUT (
    .clk(clk),
    .stop(stop),
    .doorclosed(doorclosed),
    .start(start),
    .dry(dry),
    .washOnly(washOnly),
    .detergentfilled(detergentfilled),
    .pause(pause),
    .waterTemp(waterTemp),
    .spinSpeed(spinSpeed),
    .doorLock(doorLock),
    .motorStart(motorStart),
    .fill_value_on(fill_value_on),
    .drain_value_on(drain_value_on),
    .soap_value_on(soap_value_on),
    .done(done),
    .soapWash(soapWash),
    .waterWash(waterWash),
    .cycleFinished(cycleFinished),
    .rinseFinished(rinseFinished),
    .spinFinished(spinFinished),
    .alarm(alarm),
    .heater_on(heater_on),
    .motor_speed(motor_speed),
    .timer(timer)
);

initial begin
    $dumpfile("WashingMachine_tb.vcd");
    $dumpvars(0, WashingMachine_tb);
end
```

Initialization and clock generation:



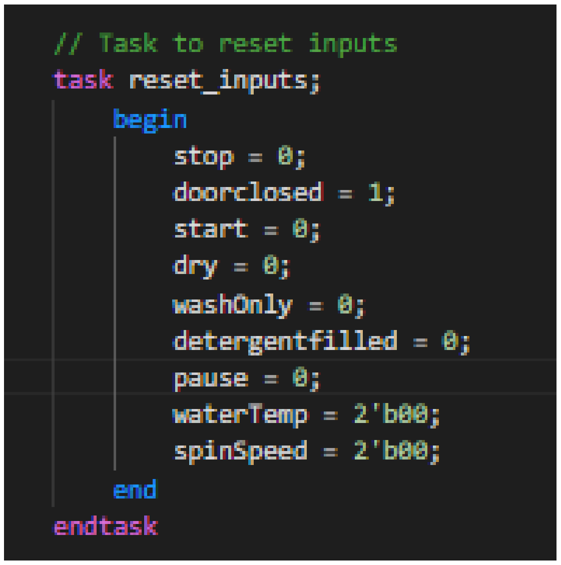
```
// Clock generation
always #5 clk = ~clk;

initial begin
    // Initialize inputs
    clk = 0;
    stop = 0;
    doorclosed = 1;
    start = 0;
    dry = 0;
    washOnly = 0;
    pause = 0;
    waterTemp = 2'b00;
    spinSpeed = 2'b00;
```

Direct testing:



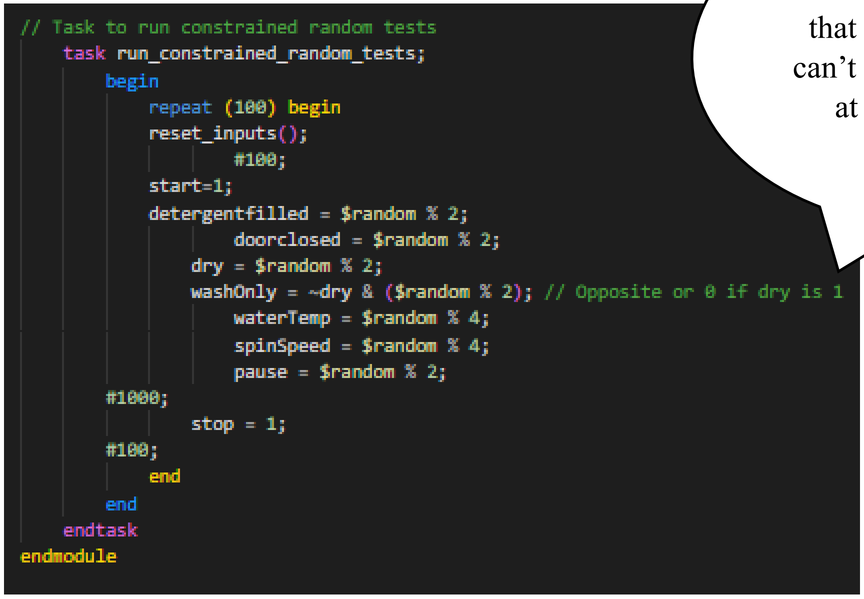
```
// Task to run directed tests
task run_directed_tests;
    begin

        // Test 1: Start washing with detergent and default settings
        #10 start = 1;
        detergentfilled = 1;
        #30 waterTemp = 2'b01;
        #40 spinSpeed = 2'b10;

        #1000;

        //Test 2: Pause and resume
        #50 pause = 1;
        #60 pause = 0;

        // Test 3: Stop and restart
        #70 stop = 1;
        #80 stop = 0;

        // Test 4: Detergent not filled
    reset_inputs();
    detergentfilled = 0;
    #90 start = 1;
        #1000;
    #70 stop = 1;
        #80 stop = 0;



         // Test 5: Dry-only mode
        reset_inputs();
    #90 start = 1;
        detergentfilled = 1;
        dry = 1;
    #1000;
        #70 stop = 1;
        #80 stop = 0;


        // Test 6: Wash-only mode
        reset_inputs();
        washOnly = 1;
        detergentfilled = 1;
        start = 1;
        #20 waterTemp = 2'b01;
        #30 spinSpeed = 2'b10;
        #1000;
    #70 stop = 1;
        #80 stop = 0;


    reset_inputs();
        detergentfilled = 1;
        start = 1;
    doorclosed = 0;
        #20 waterTemp = 2'b01;
        #30 spinSpeed = 2'b10;
        #1000;

    end
endtask
```

Task to reset inputs:



```
// Task to reset inputs
task reset_inputs;
    begin
        stop = 0;
        doorclosed = 1;
        start = 0;
        dry = 0;
        washOnly = 0;
        detergentfilled = 0;
        pause = 0;
        waterTemp = 2'b00;
        spinSpeed = 2'b00;
    end
endtask
```

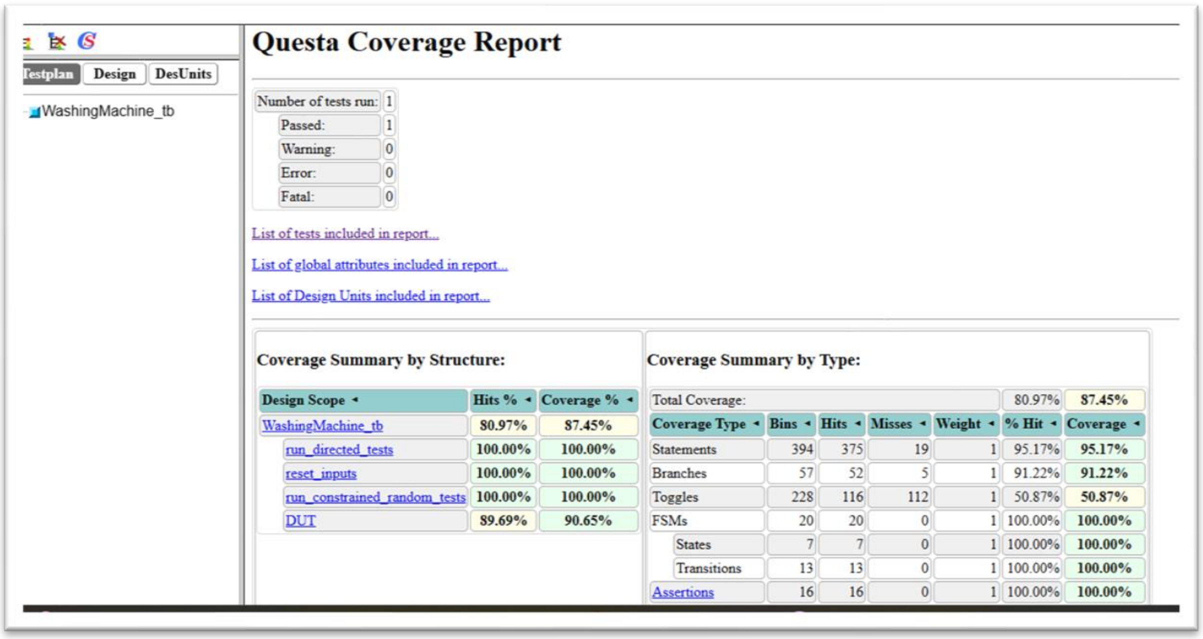
Task for constrained random testing:



```
// Task to run constrained random tests
    task run_constrained_random_tests;
        begin
            repeat (100) begin
            reset_inputs();
                    #100;
            start=1;
            detergentfilled = $random % 2;
                    doorclosed = $random % 2;
                dry = $random % 2;
                washOnly = ~dry & ($random % 2); // Opposite or 0 if dry is 1
                    waterTemp = $random % 4;
                    spinSpeed = $random % 4;
                    pause = $random % 2;
        #1000;
                stop = 1;
        #100;
            end
        end
    endtask
endmodule
```

This condition to ensure that dry & washOnly can't be both equal to 1 at the same time

VI. SIMULATION WAVEFORM SNIPPETS & COVERAGE



Questa Coverage Report

Number of tests run:	1
Passed:	1
Warning:	0
Error:	0
Fatal:	0

List of tests included in report...

List of global attributes included in report...

List of Design Units included in report...

Coverage Summary by Structure:

Design Scope	Hits %	Coverage %
WashingMachine_tb	80.97%	87.45%
run_directed_tests	100.00%	100.00%
reset_inputs	100.00%	100.00%
run_constrained_random_tests	100.00%	100.00%
DUT	89.69%	90.65%

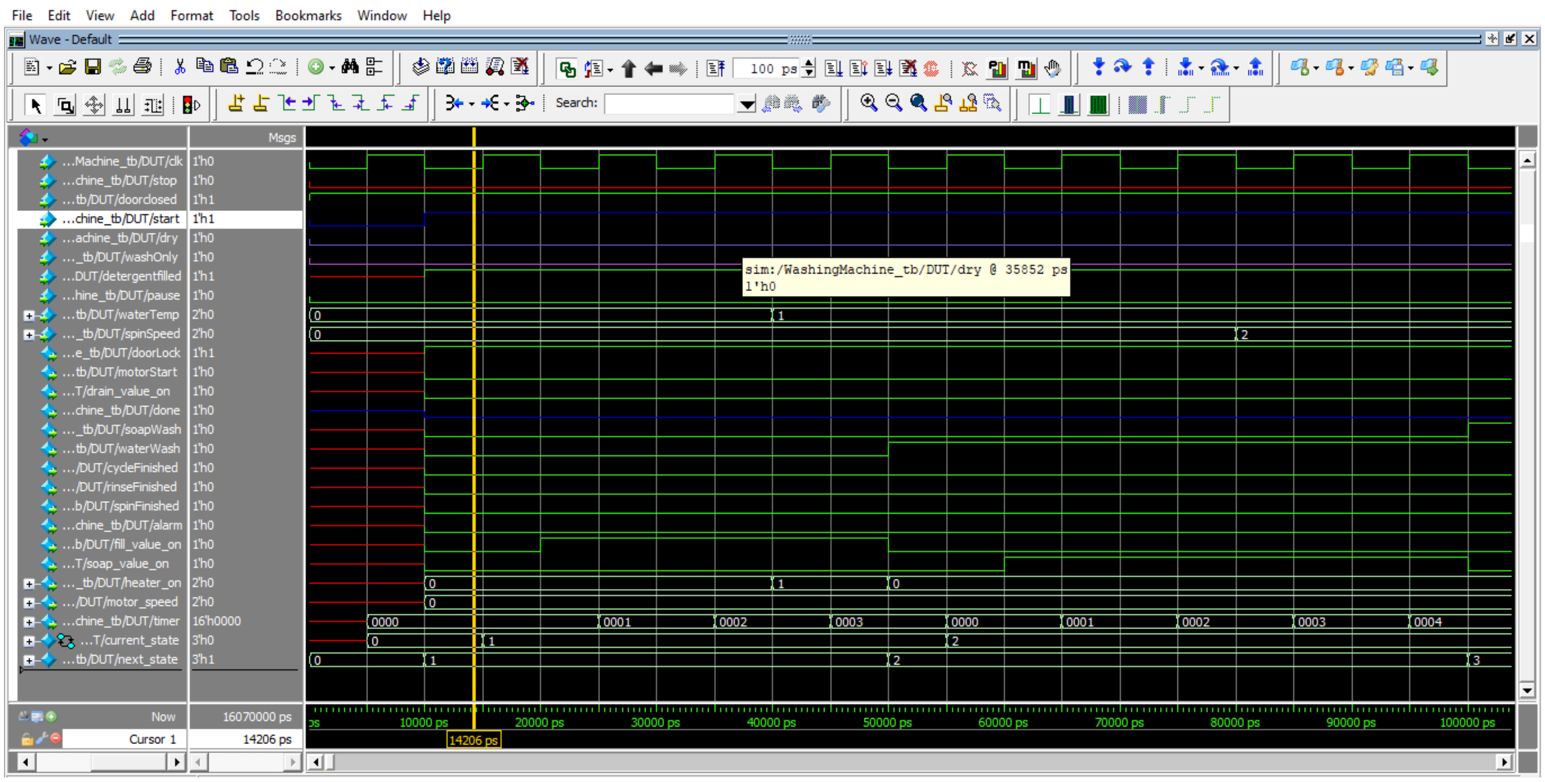
Coverage Summary by Type:

Coverage Type	Bins	Hits	Misses	Weight	% Hit	Coverage
Total Coverage:					80.97%	87.45%
Statements	394	375	19	1	95.17%	95.17%
Branches	57	52	5	1	91.22%	91.22%
Toggles	228	116	112	1	50.87%	50.87%
FSMs	20	20	0	1	100.00%	100.00%
States	7	7	0	1	100.00%	100.00%
Transitions	13	13	0	1	100.00%	100.00%
Assertions	16	16	0	1	100.00%	100.00%

Here coverage reaches almost 90%.

Waveform snippets

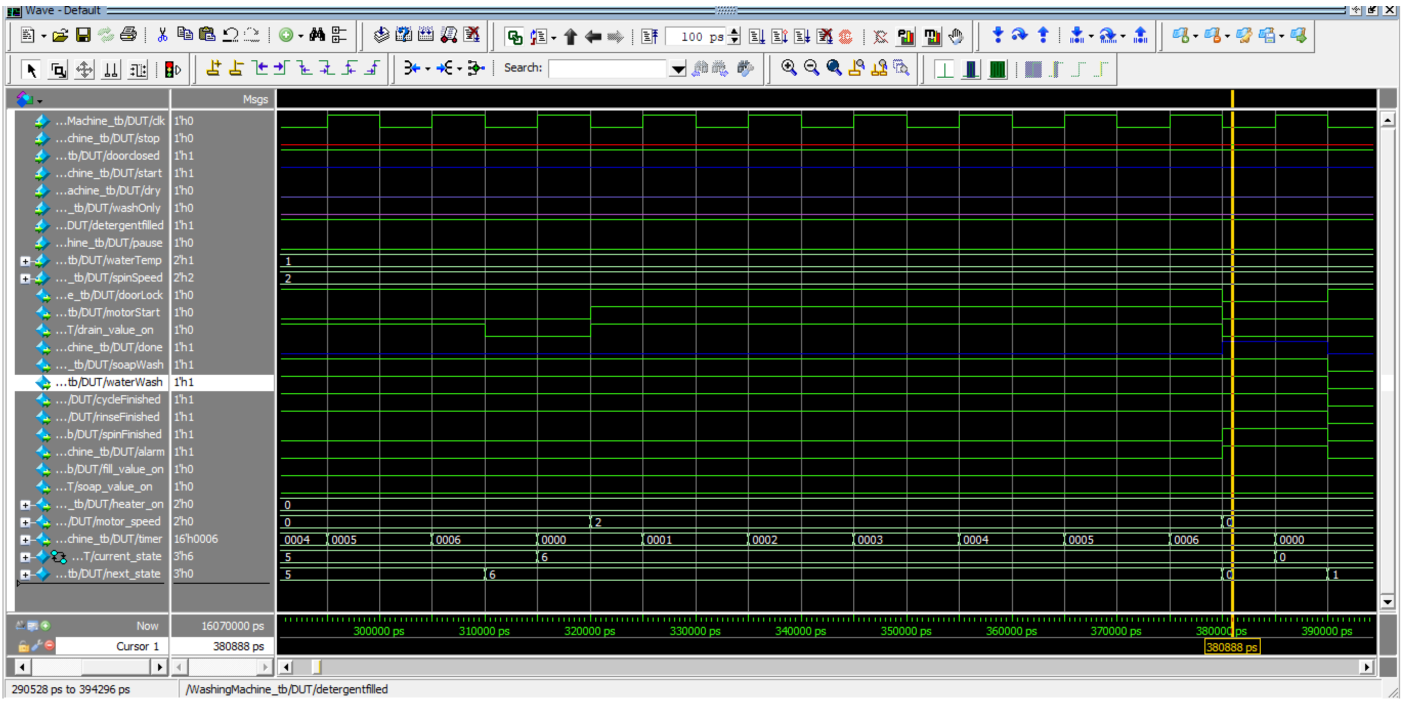
Testcase 1: Here we make a normal washing that pases through all states with dry (dry=0 & washOnly=0).



	Msgs
...Machine_tb/DUT/clk	1'h0
...chine_tb/DUT/stop	1'h0
...tb/DUT/doorclosed	1'h1
...chine_tb/DUT/start	1'h1
...achine_tb/DUT/dry	1'h0
..._tb/DUT/washOnly	1'h0
...DUT/detergentfilled	1'h1
...hine_tb/DUT/pause	1'h0
...tb/DUT/waterTemp	2'h0
..._tb/DUT/spinSpeed	2'h0
...e_tb/DUT/doorLock	1'h1
...tb/DUT/motorStart	1'h0
...T/drain_value_on	1'h0
...chine_tb/DUT/done	1'h0
..._tb/DUT/soapWash	1'h0
...tb/DUT/waterWash	1'h0
.../DUT/cycleFinished	1'h0
.../DUT/rinseFinished	1'h0
...b/DUT/spinFinished	1'h0
...chine_tb/DUT/alarm	1'h0
...b/DUT/fill_value_on	1'h0
...T/soap_value_on	1'h0
..._tb/DUT/heater_on	2'h0
.../DUT/motor_speed	2'h0
...chine_tb/DUT/timer	16'h0000
...T/current_state	3'h0
...tb/DUT/next_state	3'h1
Now	16070000 ps
Cursor 1	14206 ps

```
sim:/WashingMachine_tb/DUT/dry @ 35852 ps
1'h0
```

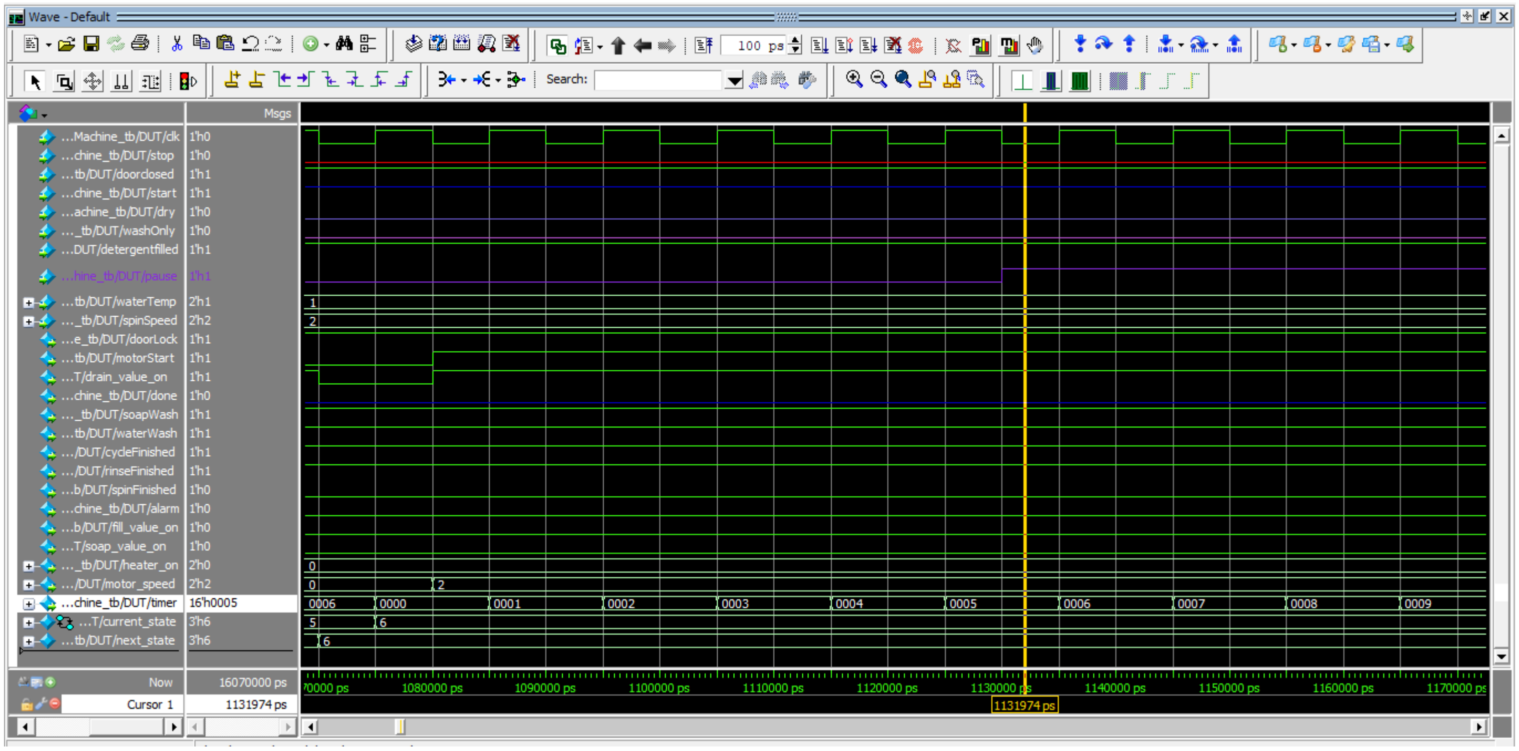
Here we can see after start=1 & doorclosed=1 & detergent is filled the fsm transmits through all states starting from check_door and timer counts in every states that need motor or action for a certain time.



```
Wave - Default
                                Msgs
...Machine_tb/DUT/clk           1'h0
...chine_tb/DUT/stop            1'h0
...tb/DUT/doorclosed            1'h1
...chine_tb/DUT/start           1'h1
...achine_tb/DUT/dry            1'h0
..._tb/DUT/washOnly             1'h0
...DUT/detergentfilled          1'h1
...hine_tb/DUT/pause            1'h0
...tb/DUT/waterTemp             2'h1
..._tb/DUT/spinSpeed            2'h2
...e_tb/DUT/doorLock            1'h0
...tb/DUT/motorStart            1'h0
...T/drain_value_on             1'h0
...chine_tb/DUT/done            1'h1
..._tb/DUT/soapWash             1'h1
...tb/DUT/waterWash             1'h1
.../DUT/cycleFinished           1'h1
.../DUT/rinseFinished           1'h1
...b/DUT/spinFinished           1'h1
...chine_tb/DUT/alarm           1'h1
...b/DUT/fill_value_on          1'h0
...T/soap_value_on              1'h0
..._tb/DUT/heater_on            2'h0
.../DUT/motor_speed             2'h0
...chine_tb/DUT/timer           16'h0006
...T/current_state              3'h6
...tb/DUT/next_state            3'h0

Now        16070000 ps
Cursor 1   380888 ps

290528 ps to 394296 ps    /WashingMachine_tb/DUT/detergentfilled
```

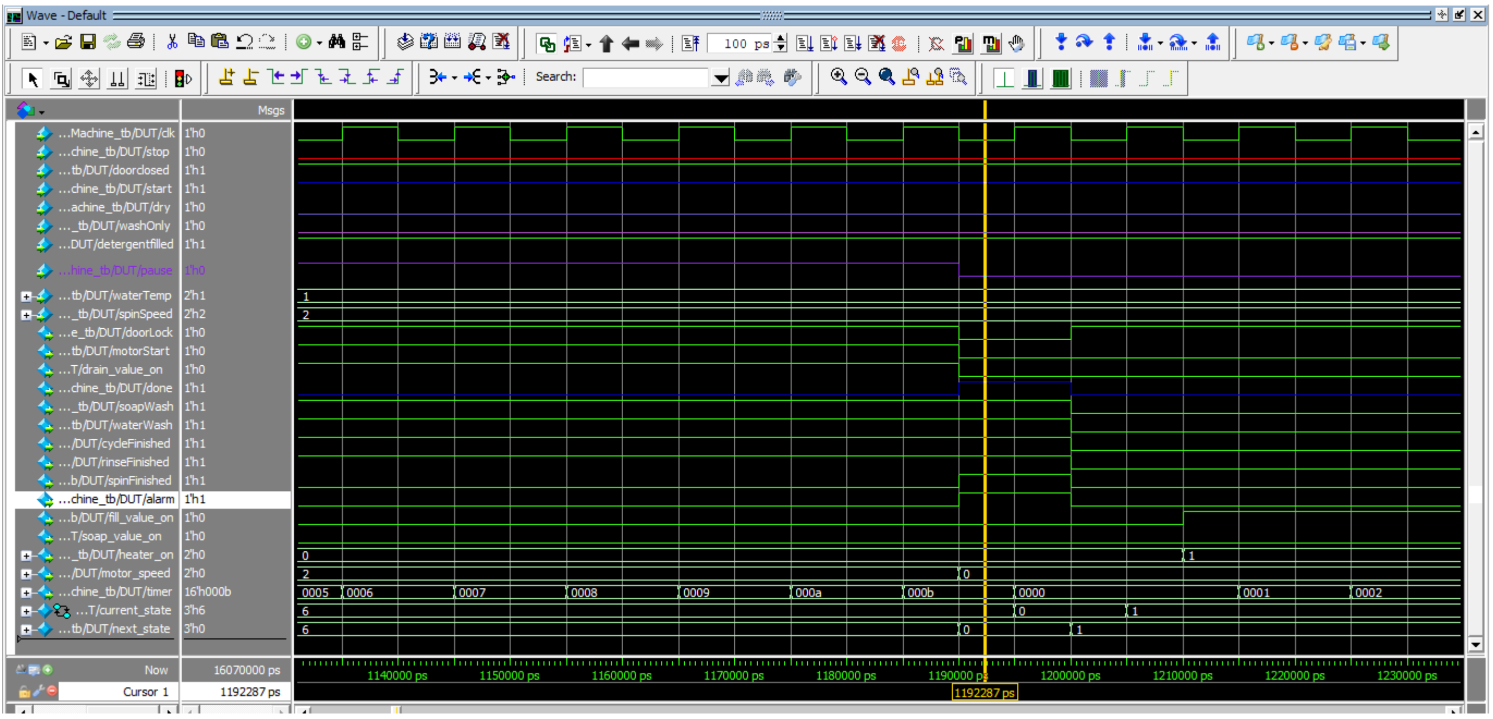
Here at the last state spin it loops 6 clock cycles until it finishes then return to first state check_door where done & alarn now equal means it has finished and doorlock became 0 to be able to open the washing machine.

<u>Testcase 2</u>: Here we changes pause to see its reaction is as we intended or not.


Wave - Default

```
...Machine_tb/DUT/clk      1'h0
...chine_tb/DUT/stop       1'h0
...tb/DUT/doorclosed       1'h1
...chine_tb/DUT/start      1'h1
...achine_tb/DUT/dry       1'h0
..._tb/DUT/washOnly        1'h0
...DUT/detergentfilled     1'h1
...hine_tb/DUT/pause       1'h1
...tb/DUT/waterTemp        2'h1
..._tb/DUT/spinSpeed       2'h2
...e_tb/DUT/doorLock       1'h1
...tb/DUT/motorStart       1'h1
...T/drain_value_on        1'h1
...chine_tb/DUT/done       1'h0
..._tb/DUT/soapWash        1'h1
...tb/DUT/waterWash        1'h1
.../DUT/cycleFinished      1'h1
.../DUT/rinseFinished      1'h1
...b/DUT/spinFinished      1'h0
...chine_tb/DUT/alarm      1'h0
...b/DUT/fill_value_on     1'h0
...T/soap_value_on         1'h0
..._tb/DUT/heater_on       2'h0
.../DUT/motor_speed        2'h2
...chine_tb/DUT/timer      16'h0005
...T/current_state         3'h6
...tb/DUT/next_state       3'h6

Now        16070000 ps
Cursor 1   1131974 ps
```

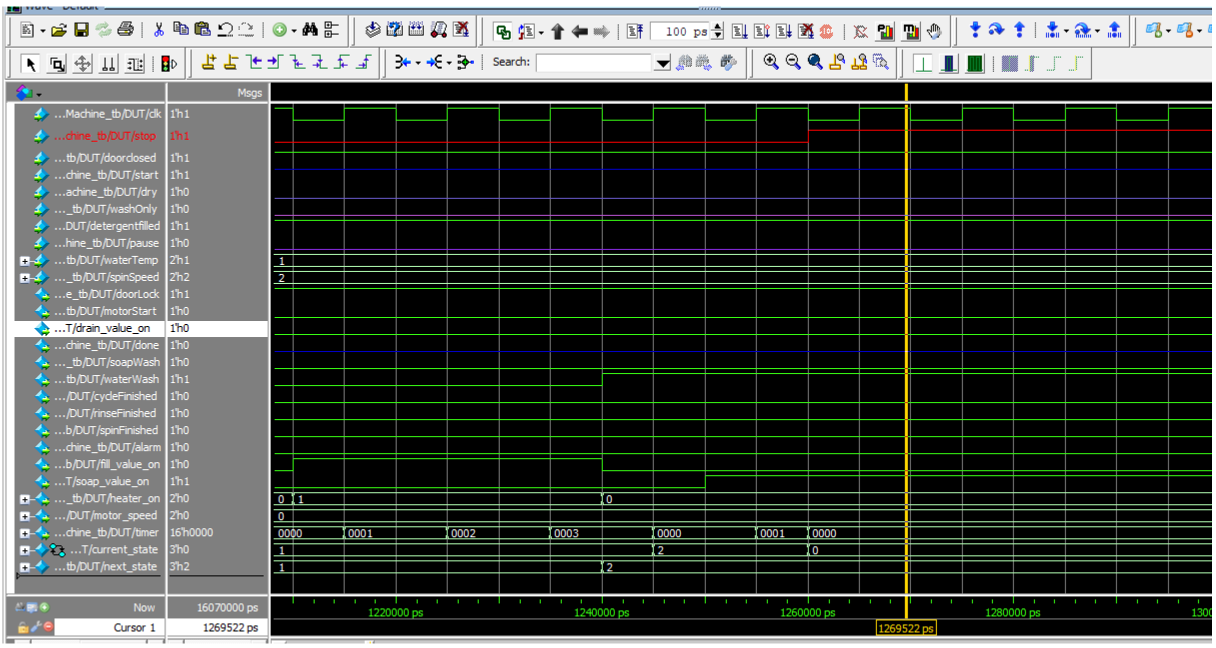
We see when pause became 1. TheFSM paused and the next state stays still which means it is equal to current state.



Wave - Default	Msgs
…Machine_tb/DUT/clk	1'h0
…chine_tb/DUT/stop	1'h0
…tb/DUT/doorclosed	1'h1
…chine_tb/DUT/start	1'h1
…achine_tb/DUT/dry	1'h0
…_tb/DUT/washOnly	1'h0
…DUT/detergentfilled	1'h1
…hine_tb/DUT/pause	1'h0
…tb/DUT/waterTemp	2'h1
…_tb/DUT/spinSpeed	2'h2
…e_tb/DUT/doorLock	1'h0
…tb/DUT/motorStart	1'h0
…T/drain_value_on	1'h0
…chine_tb/DUT/done	1'h1
…_tb/DUT/soapWash	1'h1
…tb/DUT/waterWash	1'h1
…/DUT/cycleFinished	1'h1
…/DUT/rinseFinished	1'h1
…b/DUT/spinFinished	1'h1
…chine_tb/DUT/alarm	1'h1
…b/DUT/fill_value_on	1'h0
…T/soap_value_on	1'h0
…_tb/DUT/heater_on	2'h0
…/DUT/motor_speed	2'h0
…chine_tb/DUT/timer	16'h000b
…T/current_state	3'h6
…tb/DUT/next_state	3'h0
Now	16070000 ps
Cursor 1	1192287 ps

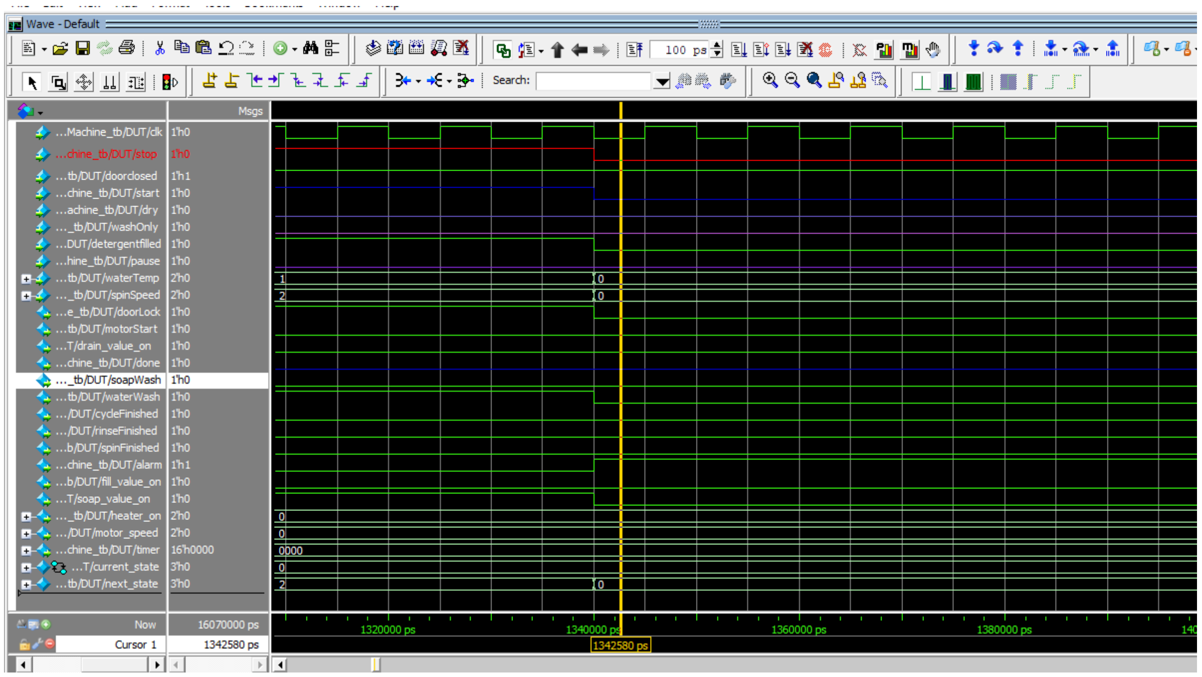
After it returned 0 the next state switched to the next one (from state 6 to state 0) which is the normal flow of states before the pause.

Testcase 3: stopping to see the reaction of the machine



Signal	Msgs
...Machine_tb/DUT/clk	1'h1
...chine_tb/DUT/stop	1'h1
...tb/DUT/doorclosed	1'h1
...chine_tb/DUT/start	1'h1
...achine_tb/DUT/dry	1'h0
..._tb/DUT/washOnly	1'h0
...DUT/detergentfilled	1'h1
...hine_tb/DUT/pause	1'h0
...tb/DUT/waterTemp	2'h1
..._tb/DUT/spinSpeed	2'h2
...e_tb/DUT/doorLock	1'h1
...tb/DUT/motorStart	1'h0
...T/drain_value_on	1'h0
...chine_tb/DUT/done	1'h0
..._tb/DUT/soapWash	1'h0
...tb/DUT/waterWash	1'h1
.../DUT/cycleFinished	1'h0
.../DUT/rinseFinished	1'h0
...b/DUT/spinFinished	1'h0
...chine_tb/DUT/alarm	1'h0
...b/DUT/fill_value_on	1'h0
...T/soap_value_on	1'h1
..._tb/DUT/heater_on	2'h0
.../DUT/motor_speed	2'h0
...chine_tb/DUT/timer	16'h0000
...T/current_state	3'h0
...tb/DUT/next_state	3'h2
Now	16070000 ps
Cursor 1	1269522 ps

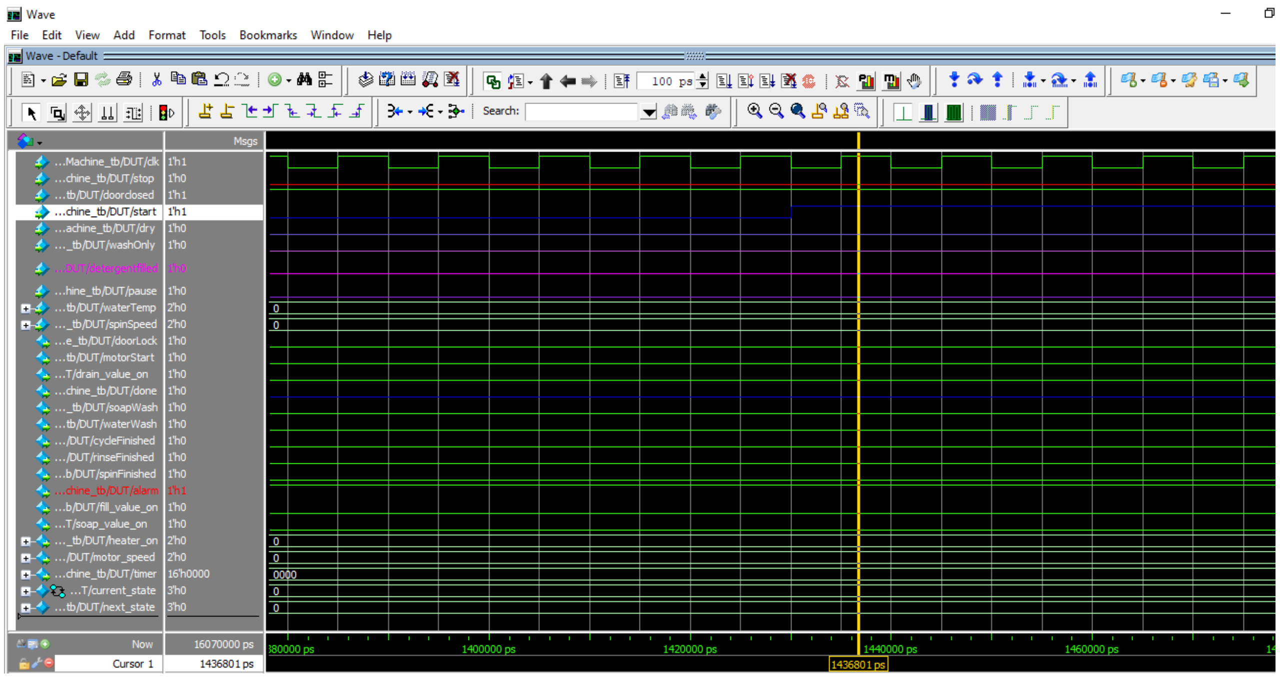
We see after stop became equal 1 the current state changed from 2(add_detergent) to 0(check_door) which is the idle 1 until stops return again equal to 0.



Signal	Msgs
...Machine_tb/DUT/clk	1'h0
...chine_tb/DUT/stop	1'h0
...tb/DUT/doorclosed	1'h1
...chine_tb/DUT/start	1'h0
...achine_tb/DUT/dry	1'h0
..._tb/DUT/washOnly	1'h0
...DUT/detergentfilled	1'h0
...hine_tb/DUT/pause	1'h0
...tb/DUT/waterTemp	2'h0
..._tb/DUT/spinSpeed	2'h0
...e_tb/DUT/doorLock	1'h0
...tb/DUT/motorStart	1'h0
...T/drain_value_on	1'h0
...chine_tb/DUT/done	1'h0
..._tb/DUT/soapWash	1'h0
...tb/DUT/waterWash	1'h0
.../DUT/cycleFinished	1'h0
.../DUT/rinseFinished	1'h0
...b/DUT/spinFinished	1'h0
...chine_tb/DUT/alarm	1'h1
...b/DUT/fill_value_on	1'h0
...T/soap_value_on	1'h0
..._tb/DUT/heater_on	2'h0
.../DUT/motor_speed	2'h0
...chine_tb/DUT/timer	16'h0000
...T/current_state	3'h0
...tb/DUT/next_state	3'h0
Now	16070000 ps
Cursor 1	1342580 ps

After stop is reset to 0 the machine resets all outputs to 0 and starts from the first state again and repeats the steps one more time not like what happened with the pause case.

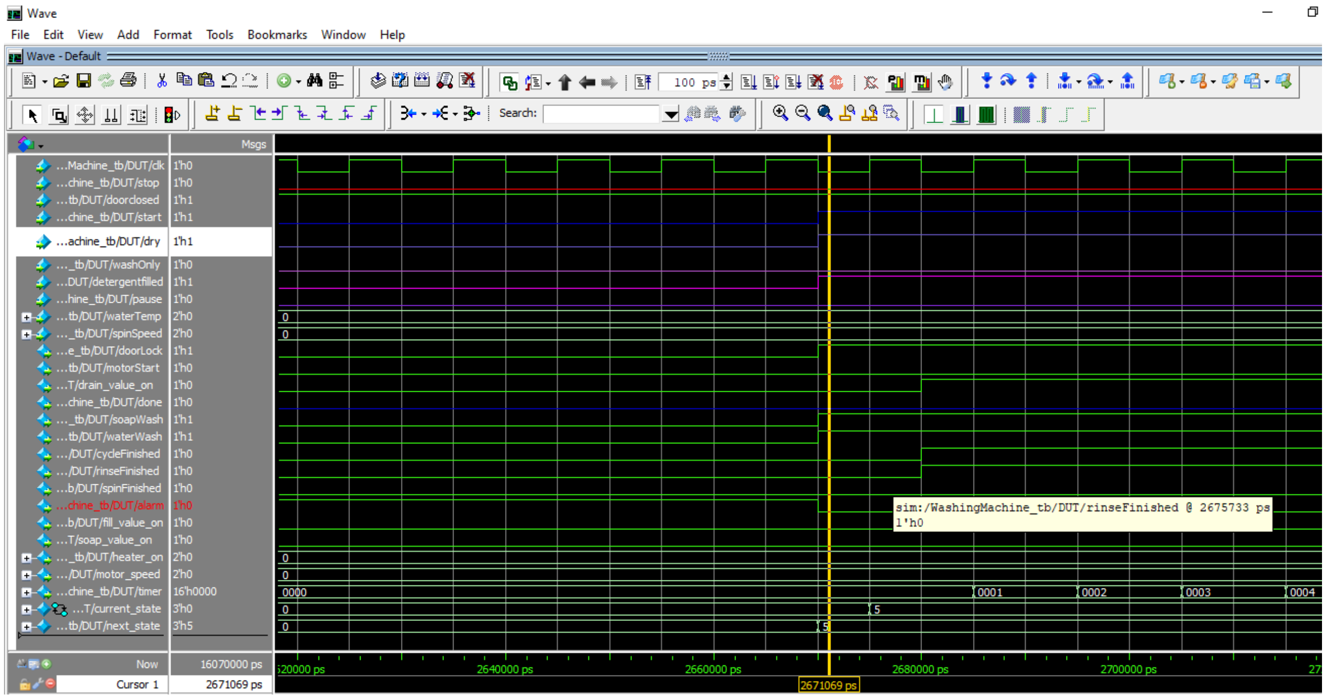
<u>Testcase 4</u>: Forgetting to add detergent



	Msgs
...Machine_tb/DUT/clk	1'h1
...chine_tb/DUT/stop	1'h0
...tb/DUT/doorclosed	1'h1
...chine_tb/DUT/start	1'h1
...achine_tb/DUT/dry	1'h0
..._tb/DUT/washOnly	1'h0
...DUT/detergentfilled	1'h0
...hine_tb/DUT/pause	1'h0
...tb/DUT/waterTemp	2'h0
..._tb/DUT/spinSpeed	2'h0
...e_tb/DUT/doorLock	1'h0
...tb/DUT/motorStart	1'h0
...T/drain_value_on	1'h0
...chine_tb/DUT/done	1'h0
..._tb/DUT/soapWash	1'h0
...tb/DUT/waterWash	1'h0
.../DUT/cycleFinished	1'h0
.../DUT/rinseFinished	1'h0
...b/DUT/spinFinished	1'h0
...chine_tb/DUT/alarm	1'h1
...b/DUT/fill_value_on	1'h0
...T/soap_value_on	1'h0
..._tb/DUT/heater_on	2'h0
.../DUT/motor_speed	2'h0
...chine_tb/DUT/timer	16'h0000
...T/current_state	3'h0
...tb/DUT/next_state	3'h0
Now	16070000 ps
Cursor 1	1436801 ps

We see after we start and detergentfilled equal to 0, then the machine stays in the same state and alarm equal to 1.

<u>Testcase 5</u>: Dry only

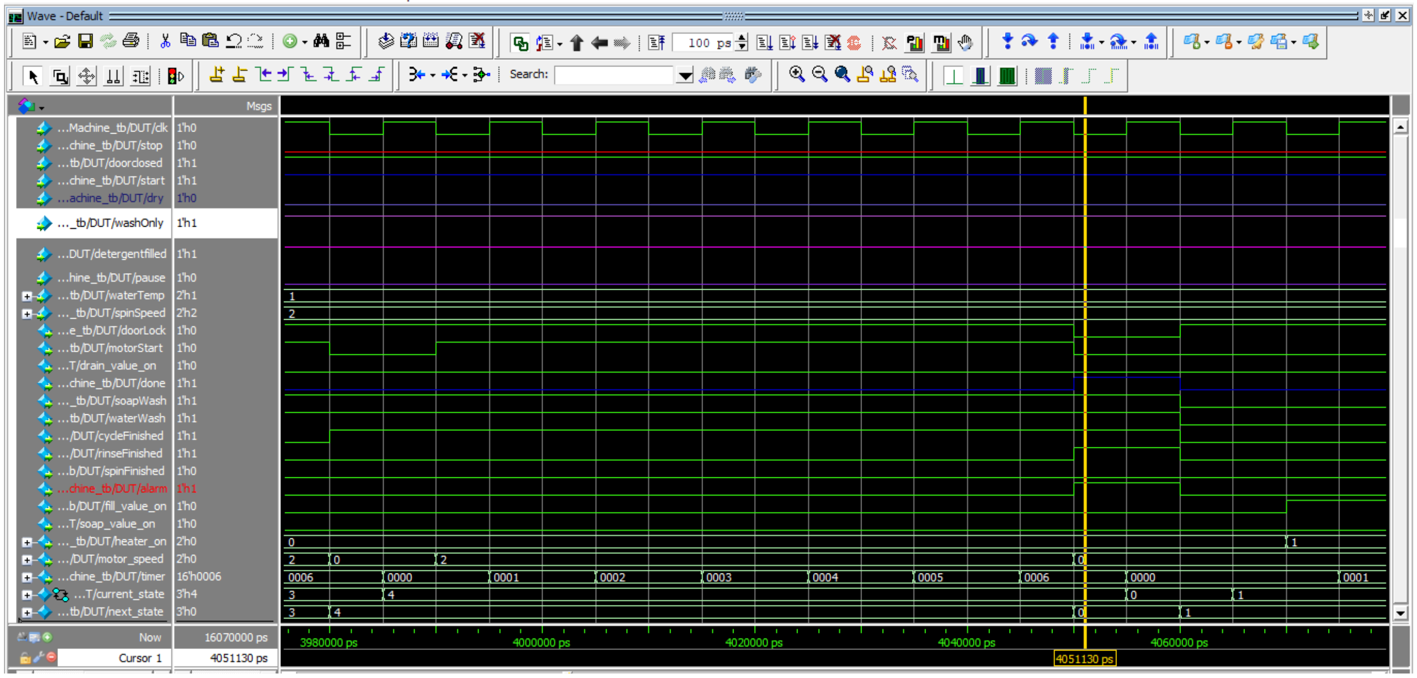


	Msgs
...Machine_tb/DUT/clk	1'h0
...chine_tb/DUT/stop	1'h0
...tb/DUT/doorclosed	1'h1
...chine_tb/DUT/start	1'h1
...achine_tb/DUT/dry	1'h1
..._tb/DUT/washOnly	1'h0
...DUT/detergentfilled	1'h1
...hine_tb/DUT/pause	1'h0
...tb/DUT/waterTemp	2'h0
..._tb/DUT/spinSpeed	2'h0
...e_tb/DUT/doorLock	1'h1
...tb/DUT/motorStart	1'h0
...T/drain_value_on	1'h0
...chine_tb/DUT/done	1'h0
..._tb/DUT/soapWash	1'h1
...tb/DUT/waterWash	1'h1
.../DUT/cycleFinished	1'h0
.../DUT/rinseFinished	1'h0
...b/DUT/spinFinished	1'h0
...chine_tb/DUT/alarm	1'h0
...b/DUT/fill_value_on	1'h0
...T/soap_value_on	1'h0
..._tb/DUT/heater_on	2'h0
.../DUT/motor_speed	2'h0
...chine_tb/DUT/timer	16'h0000
...T/current_state	3'h0
...tb/DUT/next_state	3'h5
Now	16070000 ps
Cursor 1	2671069 ps

```
sim:/WashingMachine_tb/DUT/rinseFinished @ 2675733 ps
1'h0
```

We start and dry equal 1 so the machine switches from first state to state 5 directly which is drain_water..

<u>Testcase6</u>: WashOnly is equal to 1


```
Wave - Default
                          Msgs
...Machine_tb/DUT/clk     1'h0
...chine_tb/DUT/stop      1'h0
...tb/DUT/doorclosed      1'h1
...chine_tb/DUT/start     1'h1
...achine_tb/DUT/dry      1'h0
..._tb/DUT/washOnly       1'h1
...DUT/detergentfilled    1'h1
...hine_tb/DUT/pause      1'h0
...tb/DUT/waterTemp       2'h1
..._tb/DUT/spinSpeed      2'h2
...e_tb/DUT/doorLock      1'h0
...tb/DUT/motorStart      1'h0
...T/drain_value_on       1'h0
...chine_tb/DUT/done      1'h1
..._tb/DUT/soapWash       1'h1
...tb/DUT/waterWash       1'h1
.../DUT/cycleFinished     1'h1
.../DUT/rinseFinished     1'h1
...b/DUT/spinFinished     1'h0
...chine_tb/DUT/alarm     1'h1
...b/DUT/fill_value_on    1'h0
...T/soap_value_on        1'h0
..._tb/DUT/heater_on      2'h0
.../DUT/motor_speed       2'h0
...chine_tb/DUT/timer     16'h0006
...T/current_state        3'h4
...tb/DUT/next_state      3'h0
Now                       16070000 ps
Cursor 1                  4051130 ps
```

When washonly is equal to 1, the machine starts normally until it reaches state 4 (rinse) it loops until finish rinsing then return to check_door the idle state and done is equal to 1.