Tyler Swenson

EE457

Lab 7 Report

3/18/2025

EE457 Washing Machine Simulator

# Introduction

In this lab, we will be designing a washing machine that will have two different wash cycles: standard wash and soak’n spin. The machine will start when a switch is turned on and will continue to run the selected cycle provided by another switch. The machine will continue to run until completion or until an emergency stop button is pressed, in which case the machine will either drain any water that is currently in it or immediately stop. When the machine is done, it can then be turned off by turning the original switch off. There will be several displays to indicate various parts of the cycle, such as water level, a cycle animation, and done to indicate the machine has completed the cycle.

# Theory of Operation

## Requirements

1. The design will reset when the reset signal is low.
2. The device will only turn on when SW1 transitions to high when in the OFF state.
3. The device will only turn off when SW1 transitions to low when in the DONE state.
4. When the machine is on, and the emergency stop button is pressed (LOW), the machine will do the following:
   1. If the machine has water in it, the machine will drain the water then transfer to the DONE state.
   2. If the machine has no water in it, the machine will immediately be in the DONE state.
5. Sequences will having the following timings:
   1. Fill: 3 seconds
   2. Wash/Rinse: 6 seconds
   3. Spin: 6 seconds
   4. Drain: 3 seconds
6. All inputs and resets will be synchronized.
7. Sequences will be simulated using the 7 segment LED’s:
   1. HEX3-HEX0 will display d0nE when in the DONE state.
   2. HEX4 will show a cycle animation in the RINSE, SPIN, and WASH states.
   3. HEX5 will show the current water level.
   4. Animations will update every ¼ second.

## Description of the Design

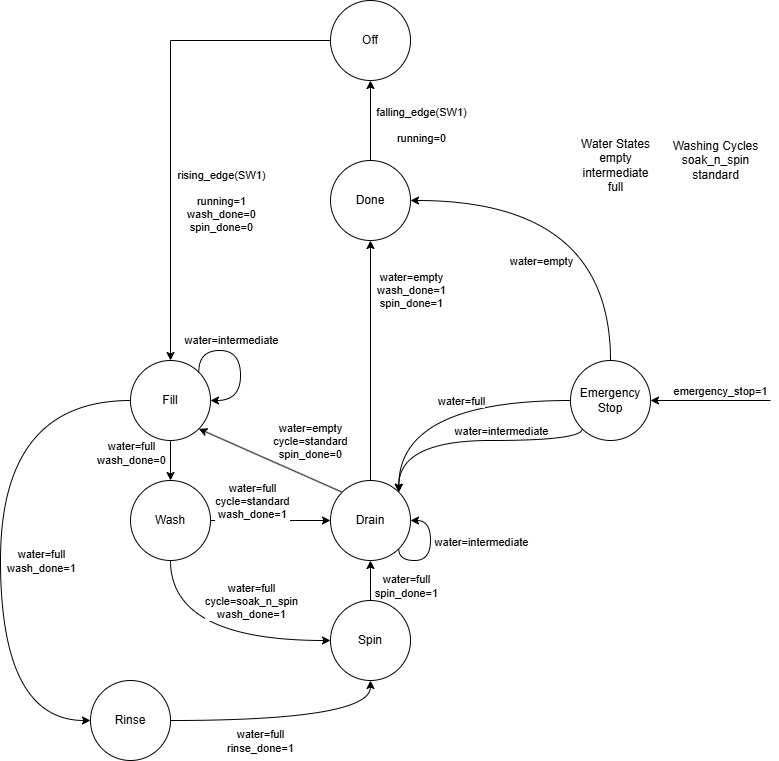


Figure 1: Washing Machine State Machine

The user will be able to provide inputs from KEY(0) for the reset signal, KEY(1) for the emergency stop, SW(0) for the desired cycle, and SW(1) for the on/off switch. Only when SW(1) transitions to ON will the machine start. The design will move and process throughout the states of each cycle until completion. The standard wash cycle will consist of: fill, wash, drain, fill, rinse, spin, and drain. The soak’n spin cycle will consist of fill, wash, spin, drain. Each cycle will take their respective timings and then proceed to the next sequence. The cycle will stop immediately if the emergency stop button (KEY(1)) is pressed. Once the emergency stop button is pressed, then the machine will drain the water in the machine then stop. If there’s no water in the machine, then the machine will stop immediately. Cycles, water fill, and done will be animated/displayed though the hex displays. The machine will only turn off when SW(1) transitions to LOW.

# Verification

## Test Plan

To test the design, a testbench will test the various cycles and cases for the washing machine, such as: standard wash, standard wash using emergency stop with both water present and not, soak’n spin, and finally the soak’n spin using emergency stop with both water present and not.

To test the forward direction, the testbench will simply set the on switch to HIGH and wait for the cycle to finish:

sw(0) <= '1'; -- cycle switch (STANDARD = '1')

sw(1) <= '1'; -- on\_off\_switch

wait for wait\_for\_state(FILL);

wait for wait\_for\_state(WASH);

wait for wait\_for\_state(DRAIN);

wait for wait\_for\_state(FILL);

wait for wait\_for\_state(RINSE);

wait for wait\_for\_state(SPIN);

wait for wait\_for\_state(DRAIN);

sw(1) <= '0';

Figure 2: Code snippet for standard wash test bench

To test the emergency stop button in the standard wash, both the conditions of water present in the machine and not will be tested with the button being pressed:

Figure 3: Code snippet for standard wash emergency button press with water

sw(0) <= '1'; -- cycle switch (STANDARD = '1')

sw(1) <= '1'; -- on\_off\_switch

wait for wait\_for\_state(FILL);

wait for wait\_for\_state(WASH);

wait for wait\_for\_state(DRAIN);

-- Emergency stop

key(1) <= '0';

wait for 50 ns;

key(1) <= '1';

wait for 50 ns;

sw(1) <= '0';

sw(0) <= '1'; -- cycle switch (STANDARD = '1')

sw(1) <= '1'; -- on\_off\_switch

wait for wait\_for\_state(FILL);

-- Emergency stop

key(1) <= '0';

wait for 50 ns;

key(1) <= '1';

wait for wait\_for\_state(DRAIN);

sw(1) <= '0';

Figure 4: Code snippet for standard wash emergency button press without water

Similarly, the Soak’n spin will be tested in the same way:

sw(0) <= '0'; -- cycle switch (SOAK = '0')

sw(1) <= '1'; -- on\_off\_switch

wait for wait\_for\_state(FILL);

wait for wait\_for\_state(WASH);

wait for wait\_for\_state(SPIN);

wait for wait\_for\_state(DRAIN);

sw(1) <= '0';

Figure 5: Code snippet for soak’n spin test bench

sw(0) <= '0'; -- cycle switch (SOAK = '0')

sw(1) <= '1'; -- on\_off\_switch

wait for wait\_for\_state(FILL);

-- Emergency stop

key(1) <= '0';

wait for 50 ns;

key(1) <= '1';

wait for wait\_for\_state(DRAIN);

sw(1) <= '0';

Figure 6: Code snippet for soak’n spin emergency button press with water

An empty soak’n spin with a press from the emergency button will be tested, however, it will just simply be press right away since there would be no water in the machine at the start or end of the cycle.

## Test Bench

### Standard Wash

#### Full cycle

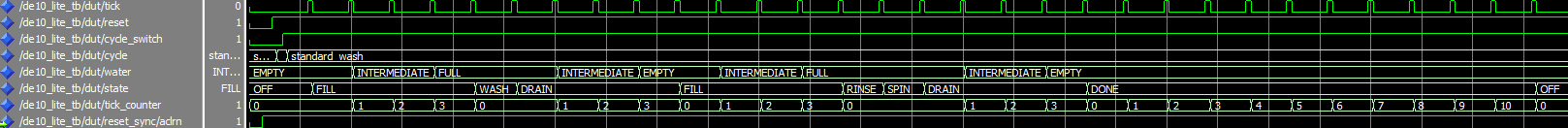


Figure 7: Standard wash full cycle

#### Emergency button press with water

A screenshot of a computer

AI-generated content may be incorrect.

Figure 8: Standard wash with emergency button press with water

#### Emergency button press without water

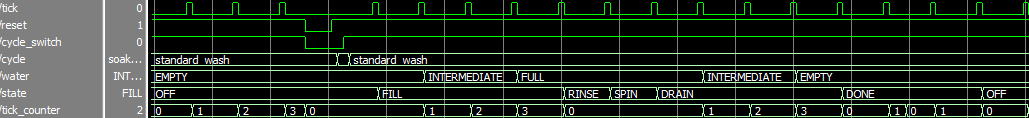


Figure 9: Standard wash with emergency button press with no water

### Soak’n Spin

#### Full Cycle

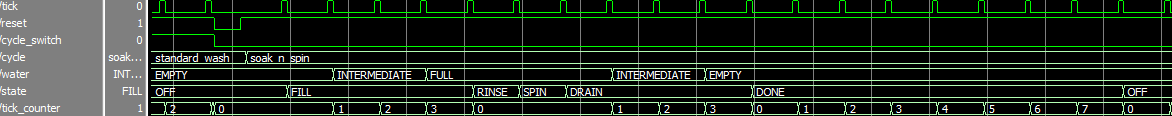


Figure 10: Soak’n Spin full cycle

#### Emergency button press with water

A screen shot of a computer

AI-generated content may be incorrect.



Figure 11: Soak’n spin with emergency button press with water

### Animations

A screen shot of a black background

AI-generated content may be incorrect.

Figure 12: Rinse animation

The rinse animation just simply transitions between left and right for the duration of the rinse.

A screen shot of a graph

AI-generated content may be incorrect.

Figure 13: Spin animation

The spin animation transitions from, bottom, right, mid, to left.



Figure 14: Fill water level animation

# Conclusion

In this lab, I learned to design a more complex state machine and reuse common states that were shared between the two cycles. This was able to combine any common state or animation code since they were similar. This expanded on what was done previously in labs with state machines and allowed for a more complex process to be made, as well as animations for each of the states. In addition to just a more complex state, the emergency button was introduced to add another complex component to what was added to previous labs. In this lab, trying to figure out the state machine was definitely more complex than previously. Though mapping it out before starting helped see what flags I may have needed to accomplish combining states so that the state machine wouldn’t just loop continuously. This lab I also had the first instance of quartus inferring latches because of those flags I had used. So overcoming that was a first.