Digital Design Assignment for the Summer Internship Program

Contents

1 Motivation:

In this paper, documentation about the digital design assignment for the summer internship program, the assignment is a controller unit for a washing machine, home appliances are an essential tool in modern life, and with advances at digital design all those appliances required a kind of a IC to control and introduce automatic features aims at offering the customer a smart and easy interactions with those appliances.

2 Specifications:

2.1 States:

This washing machine has 5 states:

State	Time
Idle	Till coin is deposited
Filling Water	2
Washing	5
Rinsing	2
Spinning	1

2.2 Inputs and Outputs:

Port	Direction	Description
rst_n	Input	Active low asynchronous clock
clk	Input	System clock
clk_freq[1:0]	Input	Input Clock Frequency Configuration Code
coin_in	input	Input flag which is asserted when a coin is deposited
double_wash	input	Input flag which is asserted if the user requires a double wash option
timer_pause	input	Input flag when it is set to ‘1’ spinning phase is paused until this flag is de-asserted
wash_done	output	Active high output asserted when the spinning phase is done and deasserted when coin_in is set to ‘1’

- The machine starts when a coin is deposited. (Assume that the user will only deposit the exact amount of coins)
- There is a double wash input, when it is turned on, a second wash and rinse occur after completing the first rinse. Assume that the double wash button is pressed before depositing the coins (if needed) and stays pressed till the job completes
- If the timer_pause flag is asserted during the spin cycle, the machine stops spinning until the flag is de-asserted. Note that the machine is designed to stop when this flag is raised only during the spin cycle.

2.3 Clock Configurations:

clk_freq[1:0]	Clock Frequency (MHz)
2'b00	1
2'b01	2
2'b10	4
2'b11	8

3 System Overview:

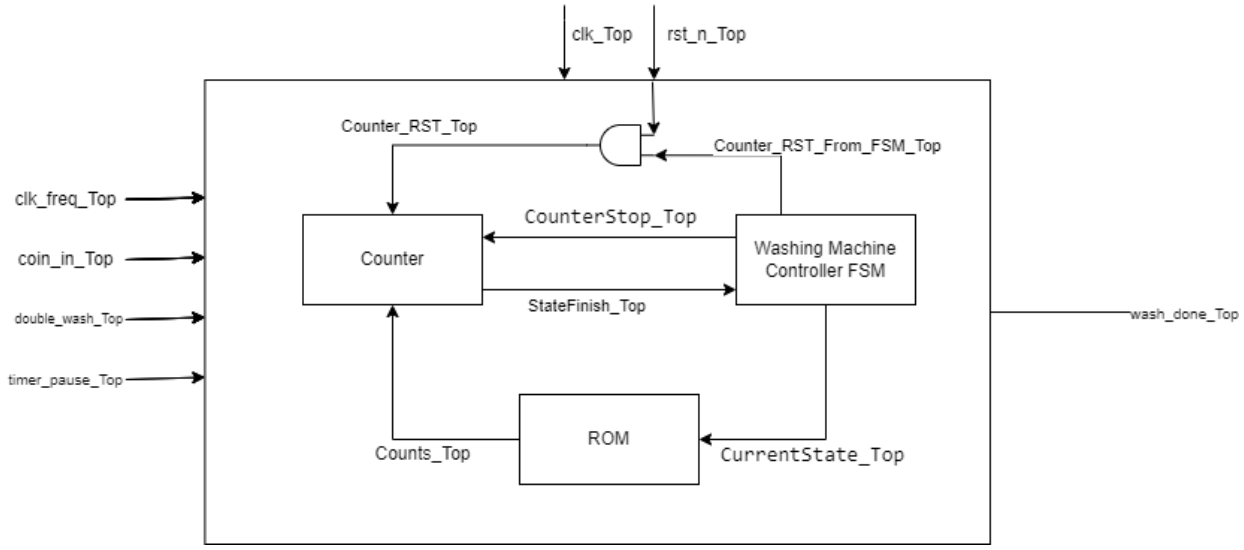

Figure 1 Washing Machine Controller.

The washing machine controller consists of 3 main units as shown in Figure 1, and each has its unique use the following section will discuss each one briefly

3.1 Washing Machine Controller FSM:

It is the controller's brain, sending the different control signals to the other units and handling the output's assertion and de-assertion depending on the inputs and the current state.

It is a Moore finite state machine where the output only depends on the current state, where the output signal "wash_done" is only asserted at the idle state till a coin is deposited and a new cycle begins

Any Moore FSM has these three main blocks as shown in:

- Transition Logic (Next State Logic). (Combinational Logic)
- State Memory. (Sequential Logic)
- Output Logic. (Combinational Logic)

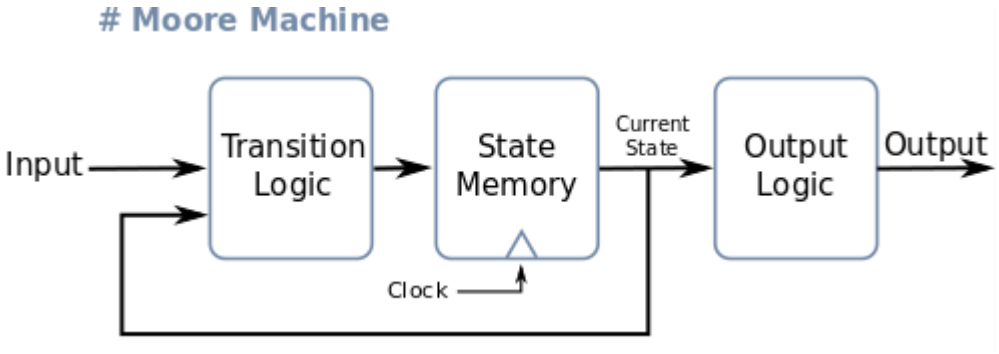

Figure 2 Moore Machine Main Building Blocks.

In our implementation, just for the ease of it both the Output and Transition Logic are combined in the same combinational block.

3.1.1 FSM Signals:

Table 1 FSM ports.

Signal	Direction	Description
rst_n	Global input	Global Asynchronous Reset
clk	Global input	Global clock signal
clk_freq [1:0]	Global input	It specifies which clock frequency is being used
coin_in	Global input	If asserted it specifies that the machine would begin its cycle
double_wash	Global input	If asserted it specifies that the machine would do a double washing and rinsing
timer_pause	Global input	If asserted at the spinning stage it specifies that the machine would stop until it's de asserted again
StateFinish	Input from the Counter unit	I asserted at the spinning stage it specifies that the current state is finished and the FSM would transition to the next state in the following clock edge
CurrentState	Output to the ROM unit	It sends the current state to ROM, to get the required number of counts needed to reach the required time for this state
Counter_RST_From_FSM	Output to the Counter unit	It is asserted at the beginning of every state to rest the counter and begin a new count
CounterStop	Output to the Counter unit	If timer_pause is asserted at the spinning state the FSM sends this to the counter to stop counting and hold the current value for now
wash_done	Global output	It is asserted at the idle state specifies that the machine is not busy and ready for a new cycle once a coin is deposited.

3.2 Counter Unit:

The counter unit is a simple unit whose main purpose is to count until reaches a specific number of counts sent from the ROM, and when it reached this number of counts it asserts a flag to the FSM declaring that the specific duration needed for this state is finished and it should move to the next state.

3.2.1 Counter Unit Signals:

Table 2 Counter Unit Ports.

Signal	Direction	Description
Counter_RST	Input	Asynchronous Reset a result from ANDing the global signal rst_n and Counter_RST_From_FSM
clk	Global input	Global clock signal
Counts [31:0]	Input from the ROM	It is the required number of counts corresponding to the current state
CounterStop	input form the FSM	If timer_pause is asserted at the spinning state the FSM sends this to the counter to stop counting and hold the current value for now
StateFinish	Output to the FSM	It is asserted when the counter reaches the number of counts specified, telling the FSM that the duration for the current state is finished.

3.3 ROM Unit:

It is a read-only memory that holds the corresponding number of counts needed for each state at different clock frequencies to reach the required duration for each state.

It uses the CurrentState signal from the FSM and clk_freq as an entrance address and outputs the corresponding number of counts to the counter unit as shown in Figure 1

3.3.1 ROM Unit Signals:

Table 3 ROM Unit Ports.

Signal	Direction	Description
clk_freq [1:0]	Global input	It specifies which clock frequency is being used
StateFromController	Input from the FSM	The current state that the FSM now is at.
CountsNum [31:0]	Output to the counter unit	It is the required number of counts corresponding to the current state

4 Test Bench:

The test bench's main purpose is to test and initially verify the functionality of our system, by applying different scenarios and observing the output.

our testing scenarios are as follows:

- The hierarchy of the rest signal.
- No coin is deposited
- Coin is deposited observing the corresponding time taken by every state.
- Double wash option is asserted
- Timer pause is asserted in different states

Note, all testing scenarios are done at a scaled time duration of the original duration required to make it practical for testing as for the original time duration the testing would take hours to complete, and this scale is shown in Table 4

Table 4 Scaled Time used in test bench

Original Duration	Scaled Duration
$1\ min = 60\ sec$	$60\mu \text{ sec} = 60000\ nsec$
$2\ min = 120\ sec$	$120\mu \text{ sec} = 120000\ nsec$
$5\ min = 300\ sec$	$300\mu \text{ sec} = 300000\ nsec$