CS M152A Lab 3

Ryan Riahi

Introduction

The purpose of this laboratory was to design a finite state machine (FSM) that simulates a parking meter. A FSM is a mathematical model of computation of an abstract machine that can be in one of a certain number of finite states at a time. FSMs are a very useful and powerful model of computation that are used in many real world scenarios. Their design allows for easy implementation into real world circuit and logic design. In this specific case, I designed an FSM to simulate the behavior of a parking meter (counting down time, adding time, resetting the time) and then programmed that behavior using the Xilinx ISE software.

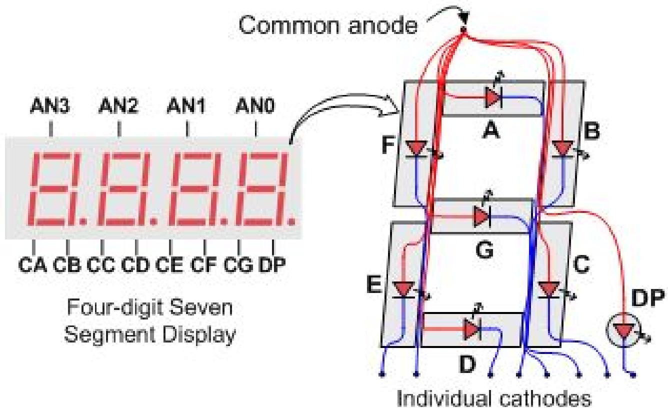
Design Description

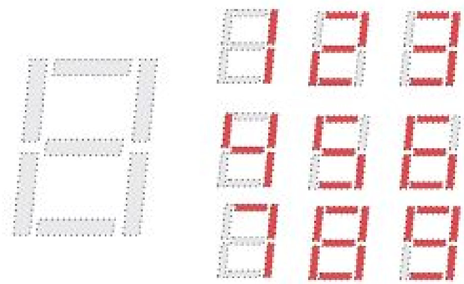
The specifications for the parking meter module are described using a set of inputs that are meant to model different coin denominations.

Inputs	Function
add1	add 60 seconds
add2	add 120 seconds
add3	add 180 seconds
add4	add 300 seconds
rst1	reset time to 16 seconds
rst2	reset time to 150 seconds
clk	frequency of 100 Hz
rst	resets to the initial state

Any one of these inputs can be pressed at any time, although two cannot be pressed at the same time.

Additionally, we must design and implement a seven segment LED display to show the reamining time left in the parking meter:

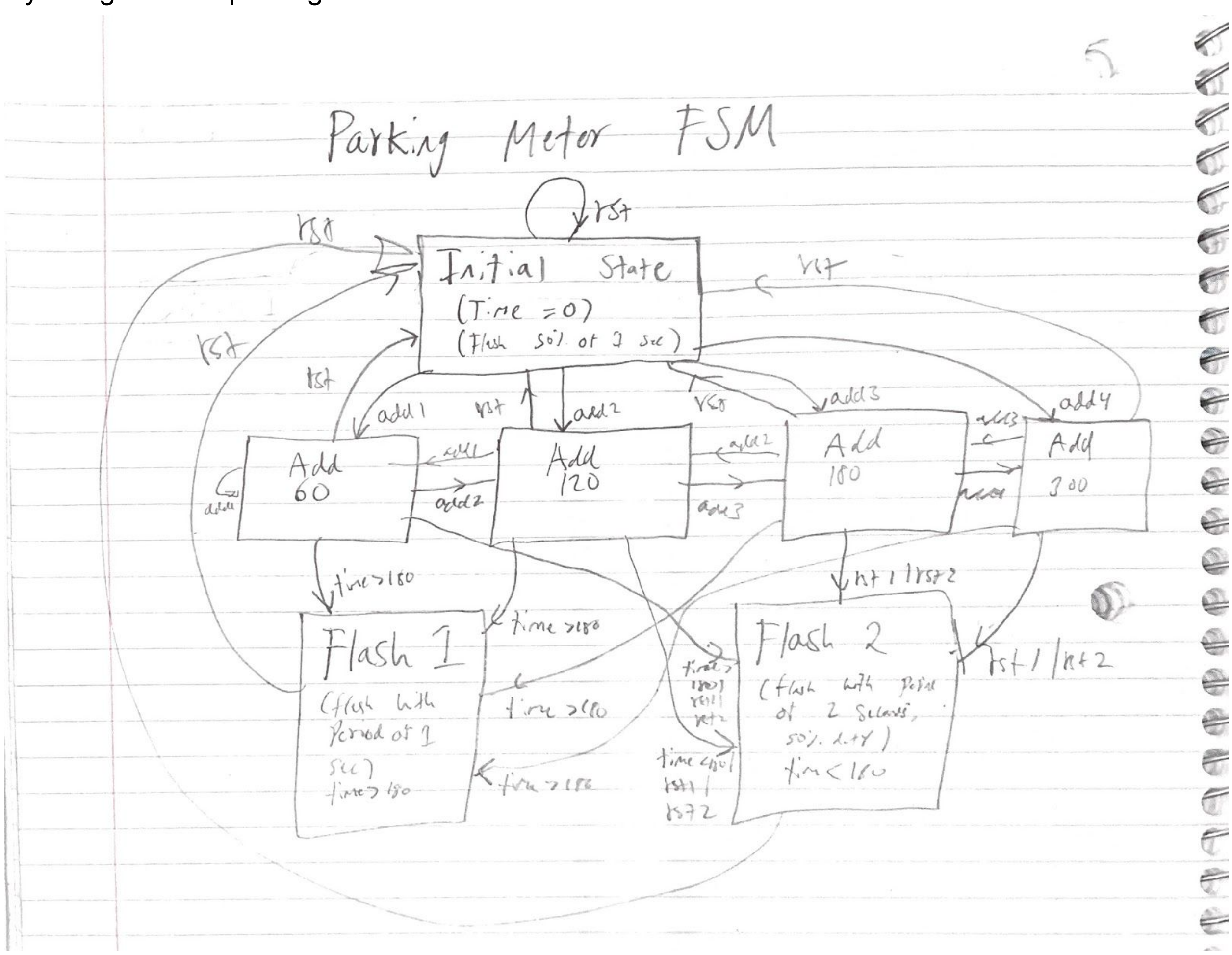




An un-illuminated seven-segment display, and nine illumination patterns corresponding to decimal digits

There are four 7 segment led displays. Each one is controlled by an anode value and a 7 bit input string. The 7 bit input string corresponds to values A - G (A being the MSB and G being the LSB). All the 1 values in the 7 bit input string correspond to a lit up segment. Additionally, the anode value for that led display must be set to 1 for any segments to light up at all. Thus, by controlling which bits are set to 1, we can display all decimal digits 0 - 9.

My design for the parking meter FSM is shown below:

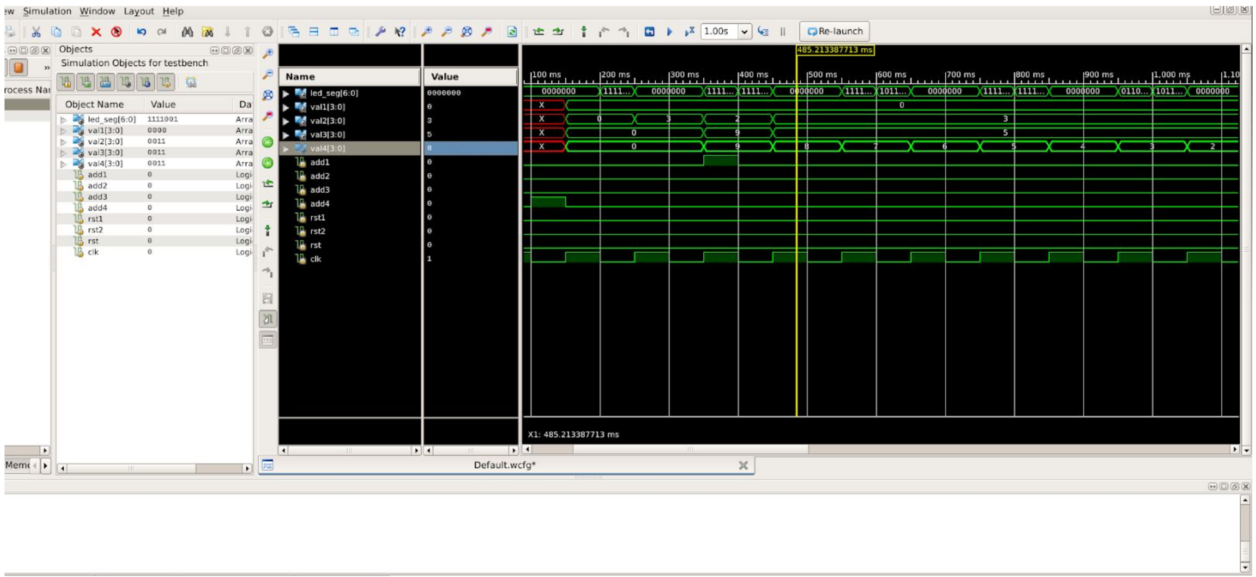


I designed my FSM with 7 states. There is an initial state where the clock has 0 seconds remaining and thus simply flashes the value 0 every 0.5 seconds. When any coins are added, the FSM will transition to the next "layer" of states which are the add states. In these states, the parking meter will add the specfied amount of time to the clock. If this time is less than 180 seconds, then the FSM will transition to the Flash 2 state which flashes with a period of 2 seconds and a 50% duty cycle. Otherwise, the FSM will transition to Flash 1 with a period of 1 second and a 50% duty cycle.

Simulation Documentation

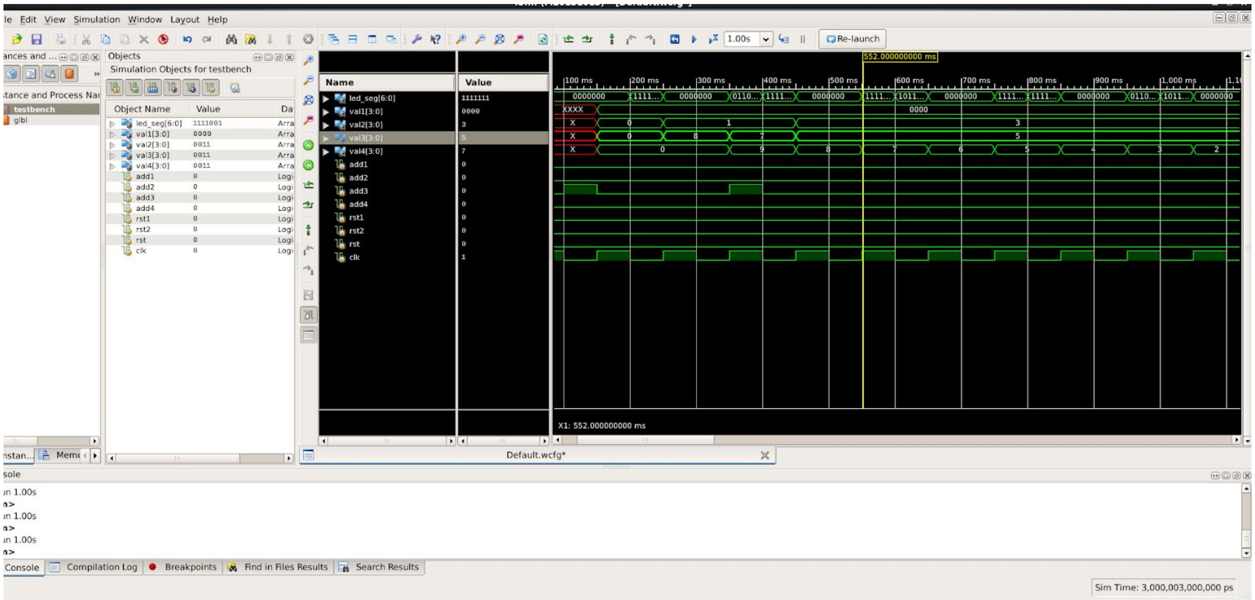
In order to test my design, I ran a simulation using VISM. The simulation was run on a test bench that resets the parking meter and then adds certain coins/ inputs into the meter over time.

Test with adding 300 seconds and then adding 60 seconds:

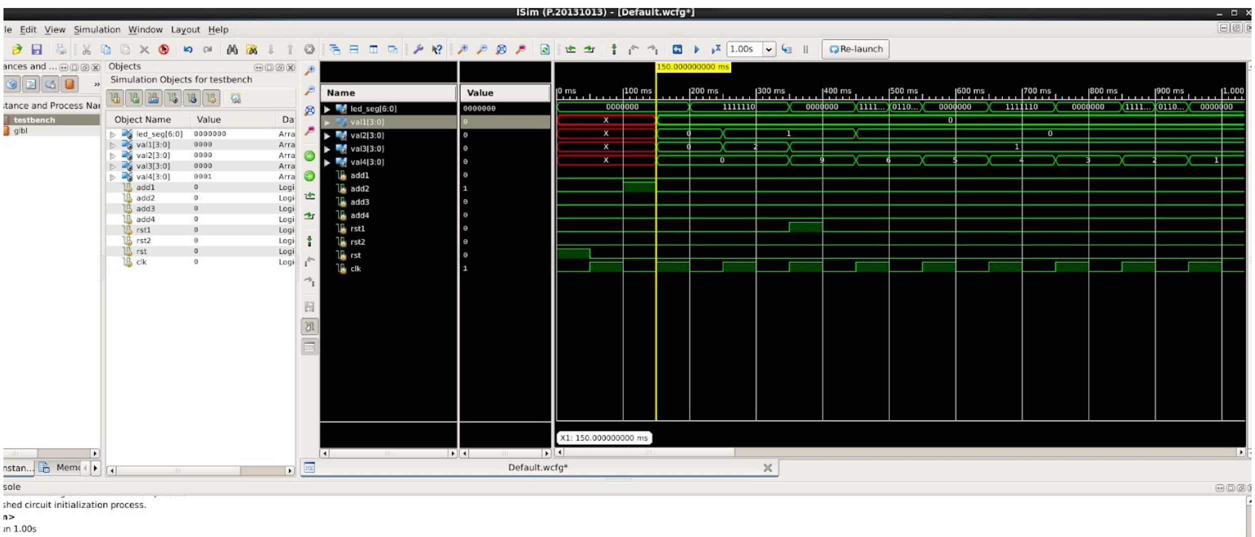


As can be seen, the counter is added 300 seconds, begins counting down, and when it reaches 298 seconds, 60 seconds are added and then it jumps up to 358. We can also see the led values alternating to display the time.

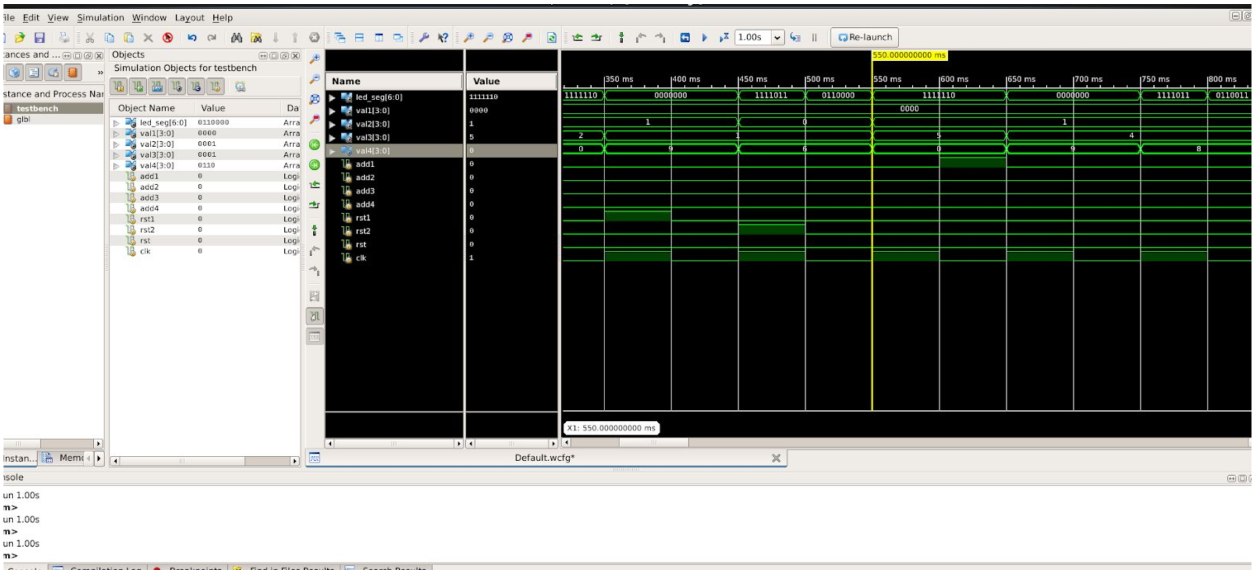
In the following test, I add 180 seconds, wait a couple seconds, then add another 180 seconds:



The next test was done by adding 120 seconds, and then resetting to 16 seconds:



I then test resseting to 150 seconds after the above test:



Conclusion

In conclusion, this lab was very informative and interesting. I feel like I have gained a mucher deeper understanding of finite state machines and how to implement them as well as their usefulness. This lab took me some time to grasp at first but once I had the idea of the FSM in my head it became easier to actually program.