CS M152A Lab 4 Report

Jonathan Chau

UID: 705-166-732

TA: Boyuan He

March 14, 2021

1. Introduction and Requirements

This lab utilizes the finite state machine (FSM) design within the Xilinx ISE Environment. Finite state machines can be used to model many real-world systems, such as behavioral modeling of sequential circuits. The two types of FSM designs include the Moore machine, which depends on the state only; and the Mealy machine, which depends on both the state and the input. For this project, we are tasked to build a parking meter by designing an FSM and display it using four 7-segment LED displays located in the Nexys3 board. An example of a seven-segment LED display is shown below.

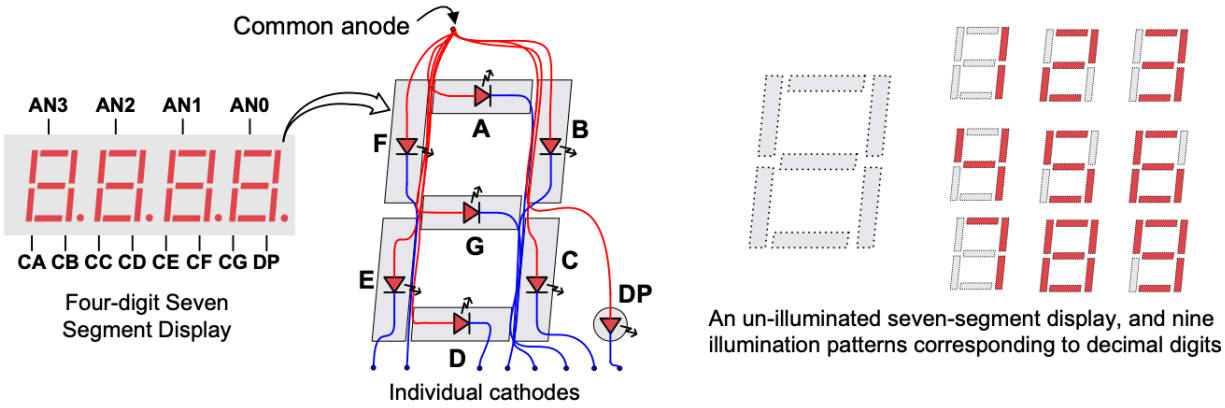

Figure 1. An example of how a 7-segment LED display works

According to the project specifications, we should design a parking meter so that:

- The clock frequency is 100 Hz.
- The 7-segment LED display should indicate the time in seconds left before expiring.
- When the timer is 0s, it should flash with a period of 1 second with a 50% duty cycle. When the timer is less than 180s, it should flash with a period of 2 seconds with a 50% duty cycle. If the timer is greater than 180s, it should not flash at all.
- If the time exceeds 9999s, the timer should start counting down from 9999s.

2. Design Description

The overall design of the module is based off of the parking meter described in the project manual. This parking meter takes eight inputs and results in nine outputs. The inputs are given as follows:

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

Figure 2. The inputs of the overall module

The outputs of the module are led_seg, a1, a2, a3, a4, val1, val2, val3, and val4. The way the anode outputs (a1, a2, a3, a4) work is interesting, as the selected anode is illuminated when it is set to LOW, not HIGH. The value outputs (val1, val2, val3, val4) indicate the number that should be displayed in BCD code. The led_seg indicates which part of the seven-segment LED display to illuminate by setting each of the seven-bits to HIGH.

2.1 Finite State Machine

Usually, a module has a cleaner design when it has less states. This statement is the reason why my finite state machine (FSM) only has three states. The FSM can still be labeled as a Moore Machine as each state determines how the outputs should be outputted. The three states can be described as follows:

- **INITIAL:** Timer is set to 0000. Display is on for 0.5s and off for 0.5s.
- **SHORT:** Timer is set to a number that is equal to 0180 or less. Display is on for every even number and off for every odd number
- **ON:** Timer is set to a number greater than 0180. Display is illuminated continuously.

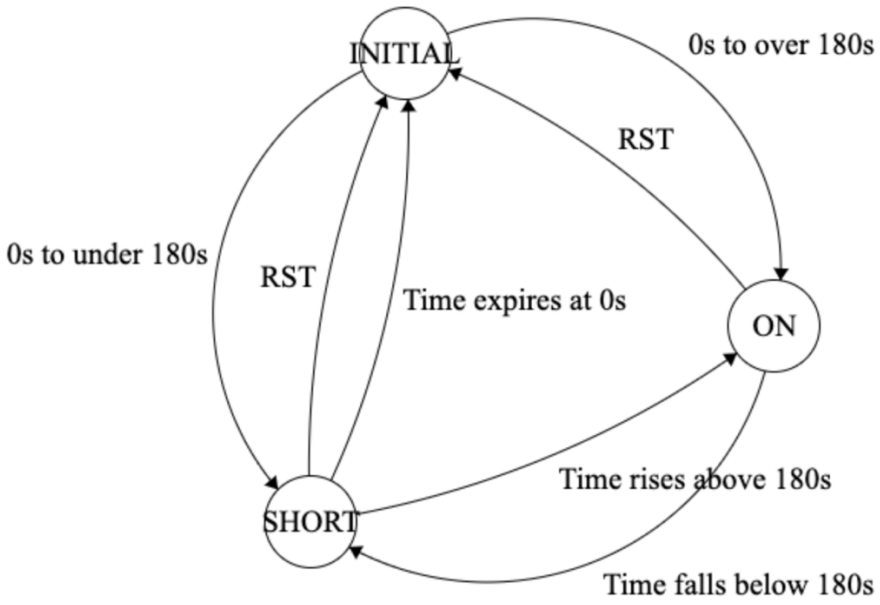

Figure 3. FSM Machine of the Parking Meter

To implement this module, I first broke it down into four submodules, before coalescing them into one overall module. All modules except for the final module contain combinational always blocks, whereas the final module contains both the combinational and sequential always blocks. The four submodules are described as follows.

- The first module takes each of the eight inputs described of the overall module and outputs the resulting timer using simple arithmetic. This submodule checks the case when the timer exceeds 9999s.

- The second submodule takes the input clock and the timer to determine the current state of the module. Hence, the output is the state.

- The third submodule takes the input clock and timer as inputs and outputs the four values (val1, val2, val3, val4) by taking each decimal digit of the timer and converting it into BCD code.

- The final submodule has a combinational always block to indicate whether the numbers should be illuminated and what digit should be illuminated based on the state of the module. The sequential always block processes the display mechanisms by using the outputs of the third submodule to output led_seg and the anode outputs.

2.2 RTL Schematic

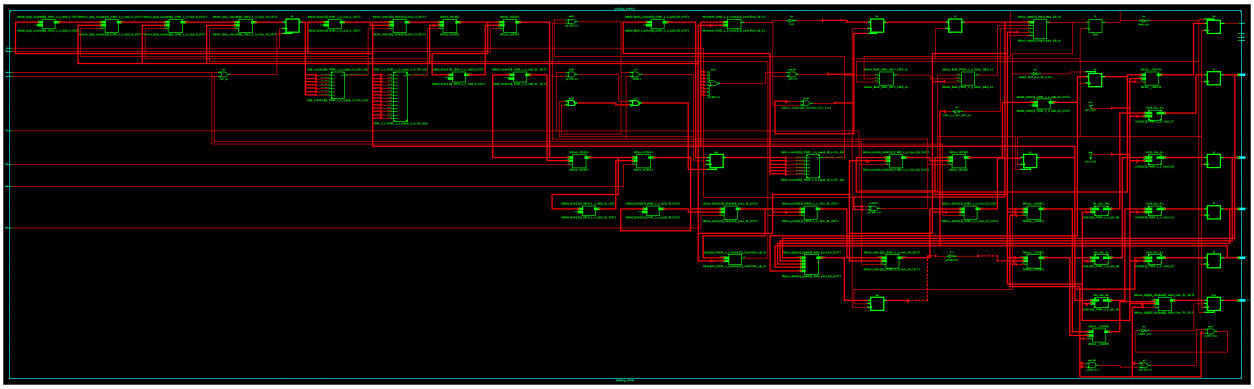

Figure 4. RTL Schematic of the Parking Meter

Since I combined all of the submodules into one giant module, the RTL schematic is filled with multiplexers and counters. The large number of multiplexers explains how my code uses switch statements to select which anode to illuminate and what specific combination the led_seg should be displayed. The large number of counters also explains how my code keeps track of many variables, such as the timer and duty cycle counters as they increment in a timely fashion.

3. Simulation Documentation

Testing was conducted using the Xilinx ISE to check for expected behavior for the parking meter. The project manual implies that each clock cycle lasts 10 milliseconds. The testing accounts for when only one of the input signals other than the clock input are pressed, as presses of two or more inputs at the same time are not accounted for. We should also note that each input lasts up to one clock cycle.

3.1 Design Test

For this section of testing, I want to see if each of the tasks produces expected behavior of the parking meter in general. Testing for accurate behavior of the LED displays will be explained in the next section.

3.1.1 Initial State

The first test checks to see if the display signals illuminate for 500ms and off for another 500ms. The illumination is shown when the anode outputs are set to LOW for one clock cycle and HIGH for the other three. In other words, each of the four anode outputs take turns setting to LOW so that its corresponding digit illuminates. We can see that this is the case at 0 to 500ms. When the digits turn off, all anode outputs are set to HIGH. This should last for 500ms. We can see that this is the case from 500 to 1000ms, since 1000 – 500 is 500. Finally, when left at initial state, the pattern should repeat itself every second. The waveform below verifies this criterion as the pattern explained above repeats from 1000 to 2000ms. To complete the test, the initial state should have all of its value outputs (val1, val2, val3, val4) set to 0, which is does according to the figure below.

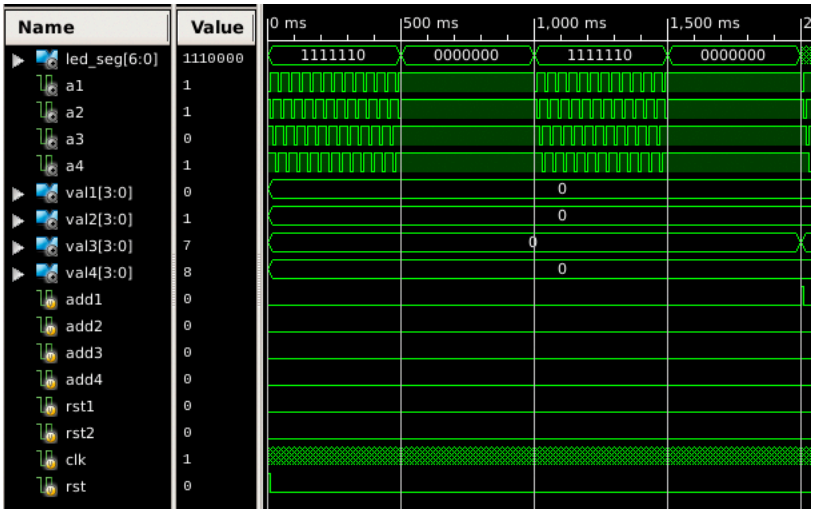

Figure 5. Simulation Waveform of the Initial State

3.1.2 Add1

We want to see if the add1 signal adds 60 seconds and turns off the display after one second. Before 2000ms, the value outputs read 0000. At 2000ms, add1 is applied and the timer should display 0060. As we can see, from 2000 to 3000ms, the value outputs display 0, 0, 6, 0, which corresponds to 0060. To explain, val1 describes the digit at the thousands place. The

output val2 displays the digit at the hundreds place. The same logic follows for val3 and val4. This means we have added 60 seconds to the timer. At 3000ms, the timer decrements to 0059. At this time, all of the display digits should be off, which we can see according to the figure below.

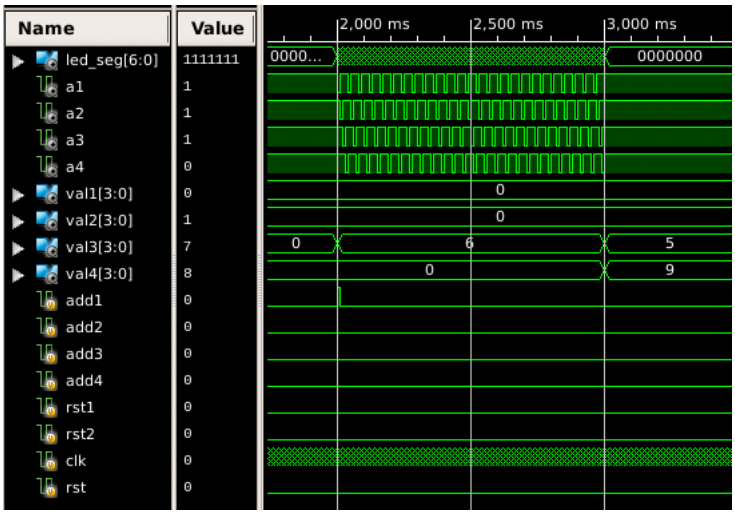

Figure 6. Simulation Waveform with the add1 Signal

3.1.3 Add2, at Odd Time

When we apply the add signal to an active timer, the timer should decrement as it would before. That means if the timer decrements at 3000ms, and we apply the add signal between 3000 and 4000ms, the timer should still decrement at 4000ms. The figure shown below shows that add2 signal applied sometime between 3 and 4 seconds. At 4000ms, we see the timer decrement from 179 to 178s, which means the module works as expected. In addition to that, I applied the add2 signal when the timer reads an odd number (0059). The new number shown is 179. However, since 179 is an odd number less than 180, all of the anodes should still be HIGH. The waveform below shows that this is the case. To complete this test, add2 should add 120 seconds to the timer. The old timer is 0059, and the new timer is 0179. 179 – 59 is 120, which verifies that the input add2 functions as expected.

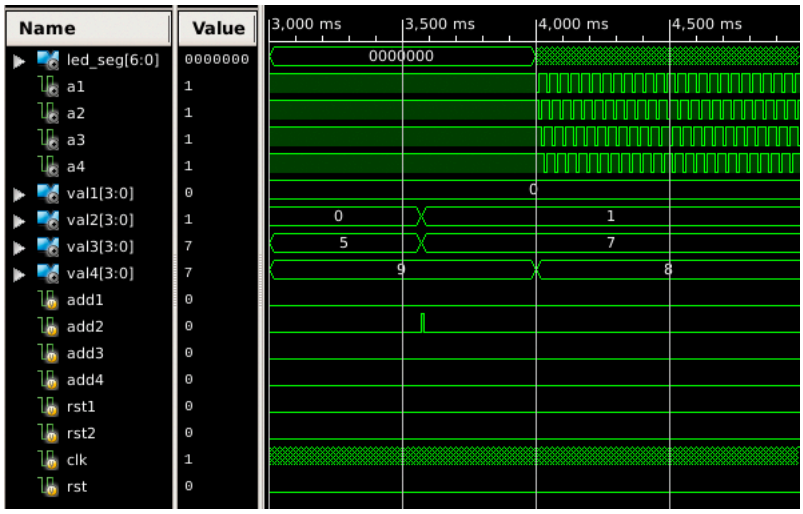

Figure 7. Simulation Waveform with the add2 Signal

3.1.4 Add3

The add3 signal was applied at 6000ms. Just before that, the timer decremented from 0177 to 0176, also at 6000ms. After applying the signal, the new timer reads 0356. The add3 signal should add 180 seconds to the timer. To verify that, we subtract 176 from 356. 356 – 176 is 180, which verifies that the add3 signal works as expected. Since the timer is over 180s, there should be no problem displaying the digits when the timer reads 0355, an odd number, at 7000ms.

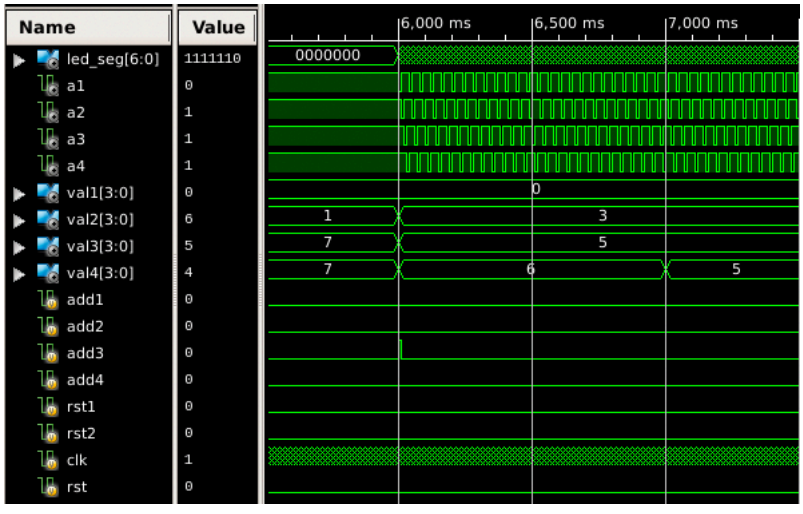

Figure 8. Simulation Waveform with the add3 Signal

3.1.5 Add4

The add4 signal was applied at 8000ms, which should add 300s to the timer. Just before, at 8000ms, the timer decremented from 0355 to 0354. After applying the signal, the new timer reads 0654. Subtracting 354 from 654 leaves 300, which verifies that the add4 signal adds 300 seconds to the timer.

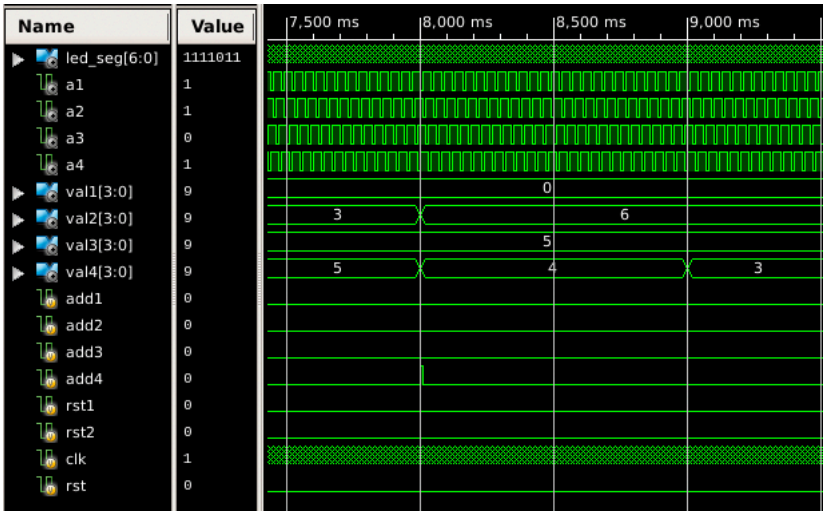

Figure 9. Simulation Waveform with the add4 Signal

3.1.6 9999s

From 10000 to 10390ms, I continuously applied the add4 signal, exceeding the maximum time. When the timer exceeds 9999, the timer should reset to 9999 and decrement from there. When this happens, the timer should not decrement at 11000ms, but rather 11390ms, since add4 was last applied at 10390ms. We can see in the figure below that around 10300 to 10400ms, the timer continuously reads 9999 while add4 is continuously applied. This action shows the meter works as expected. We can also see that the timer decrements from 9999 to 9998 at 11390ms, which means that the timer is reset to 9999, rather than continuing the behavior when time is added.

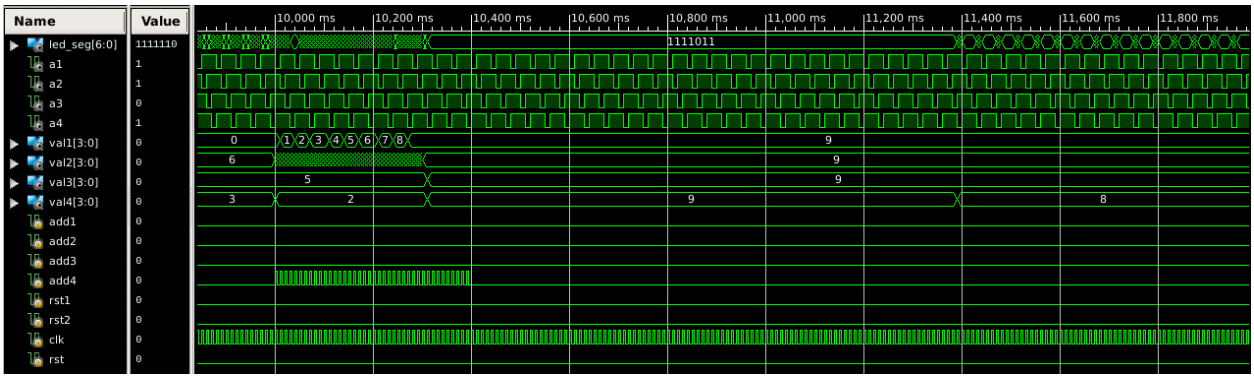

Figure 10. Simulation Waveform when Timer exceeds 9999s

3.1.7 rst

The rst signal was applied at 12000ms, which should reset the timer to 0000 and put the parking meter in initial state. To show that this is the case we have to look at the time before and after and what happens at 12500ms and 13000ms. The time before is 9998, and the time after is 0000. At 12500ms, the anodes are all set to 1, which means the display is off. At 13000ms, the anodes begin switching on and off, keeping the display constantly illuminated. This verifies that the rst signal works as expected.

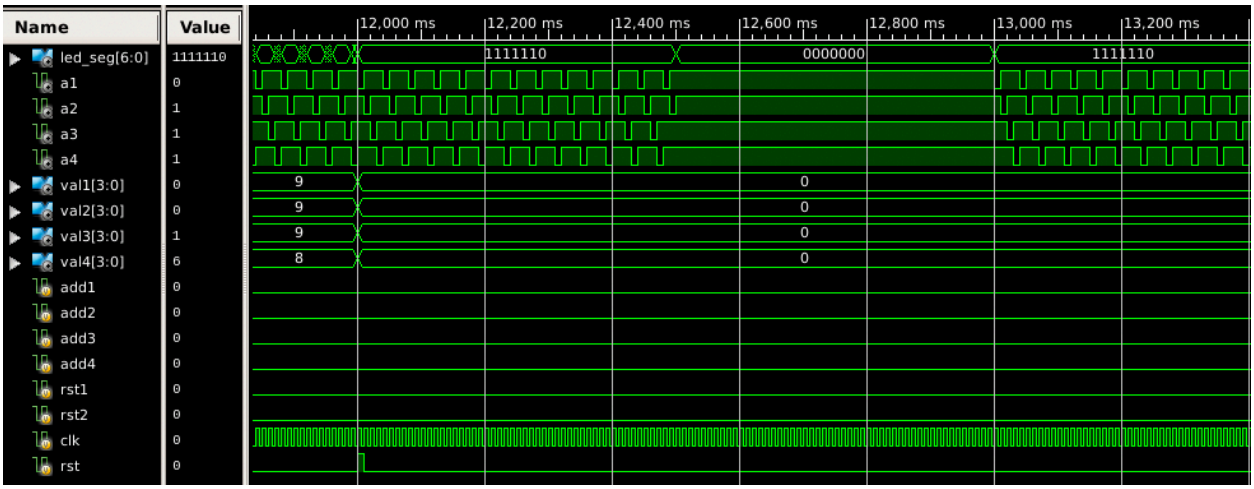

Figure 11. Simulation Waveform with the rst Signal

3.1.8 rst1

At 13500ms, the rst1 signal was applied, which sets the timer to 16s. We can see that after 13500ms, the timer is set to 0016, and turns the display off one second later at 14500ms. This ensures that the rst1 signal works as expected.

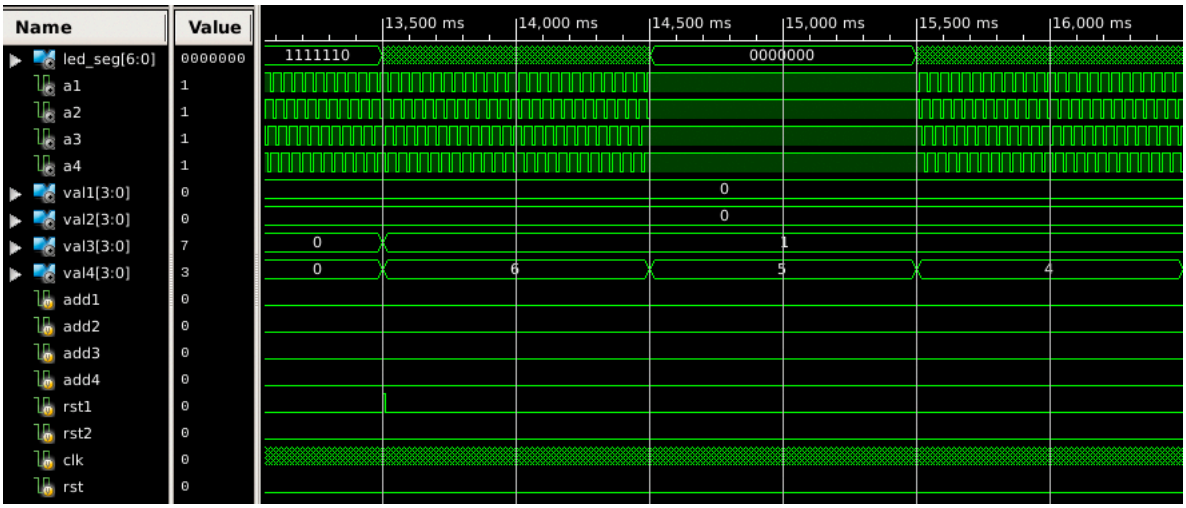

Figure 12. Simulation Waveform with the rst1 Signal

3.1.9 Add to Over 180s at Odd Time

When the timer goes above 180s at odd time, the display should illuminate immediately. The add2 signal was applied at 27000ms, right in the middle of when the timer read 0063, a number when the display turns off. So we should see the anodes rotate LOW outputs starting at 27000ms, rather than 27500ms, when the timer decrements. The waveform below verifies that this is the case.

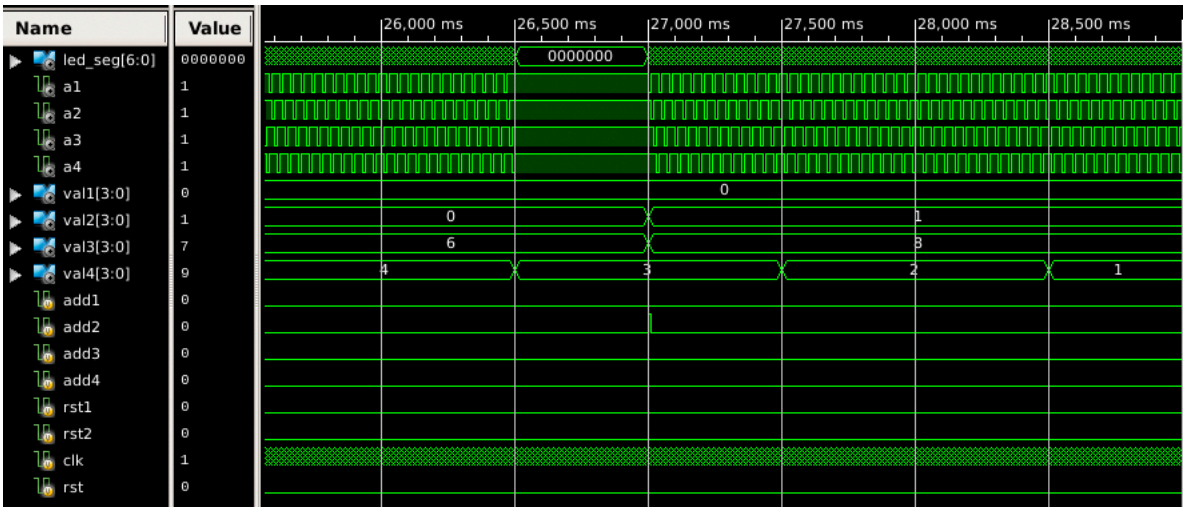

Figure 13. Simulation Waveform demonstrating the effects of going over 180s at Odd Time

3.1.10 Running Below 180s

When the timer runs below 180s, the display should start flashing on or off for a period of two seconds. That means the displays are off for every odd number less than 0180. In other words, we should see the anodes rotate LOW outputs when the timer reads 0181 and not 0179. The simulation waveform below verifies that this is the case.

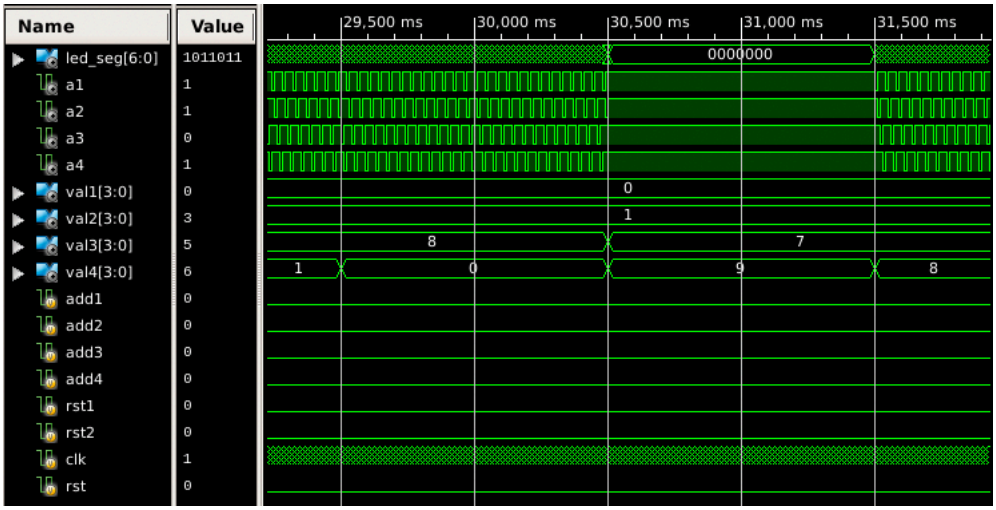

Figure 14. Simulation Waveform where Timer falls below 180s

3.1.11 Add to Over 180s at Even Time

When the timer adds to above 180s at even time, the display should be kept continuously illuminated. Since the display illuminates when the timer reads every number, adding time should keep the digits illuminated. Furthermore, when the timer decrements, the digits should continue to light up. In the waveform below, we can see that the add3 signal is applied when the timer reads 0178, which becomes 0358. At both times, we see the waveforms of the anode outputs are not constantly HIGH. When the timer decrements to 0357, the waveforms are still not constantly HIGH, indicating that the digits are still illuminated.

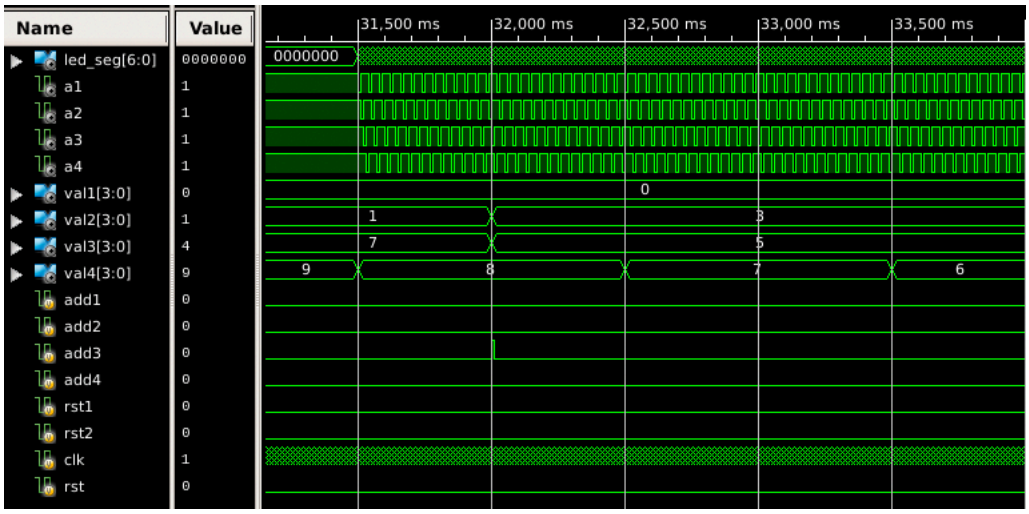

Figure 15. Simulation Waveform demonstrating the effects of going over 180s at Even Time

3.1.12 rst2

While the timer is actively counting down, the rst2 signal is applied at 35000ms, 500ms after the last timer decrement. Immediately, the timer should be set to 0150. The figure below shows that the timer resets to 0150. When the rst signal is applied, the next time the timer decrements should be 1000ms later, not 500ms. That means we should see the timer decrement from 0150 to 0149 at 36000ms. The waveform below shows that this is the case.

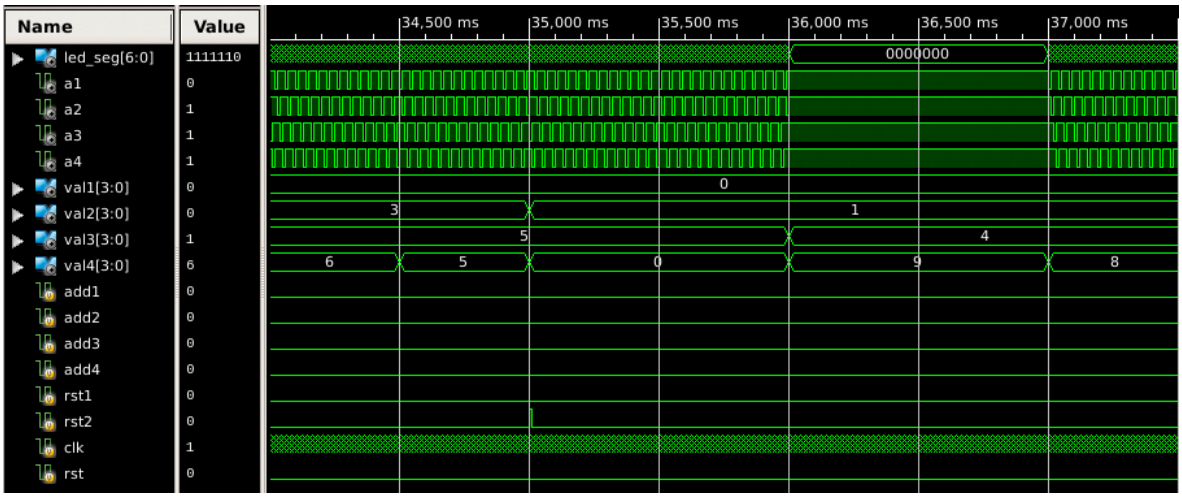

Figure 16. Simulation Waveform with the rst2 Signal

3.1.13 rst at odd time

When the rst1 signal is applied for an odd number less than 180, the digits should immediately flash on with 0016, rather than continuing to be off. In the waveform below, the rst1 signal is applied at 38500ms, 500ms after the last decrement, and when the timer read 0147. One crucial observation is that the digits illuminate at 38500ms, rather than 39000ms, the next time the timer would decrement. This verifies that the signal produced expected behavior.

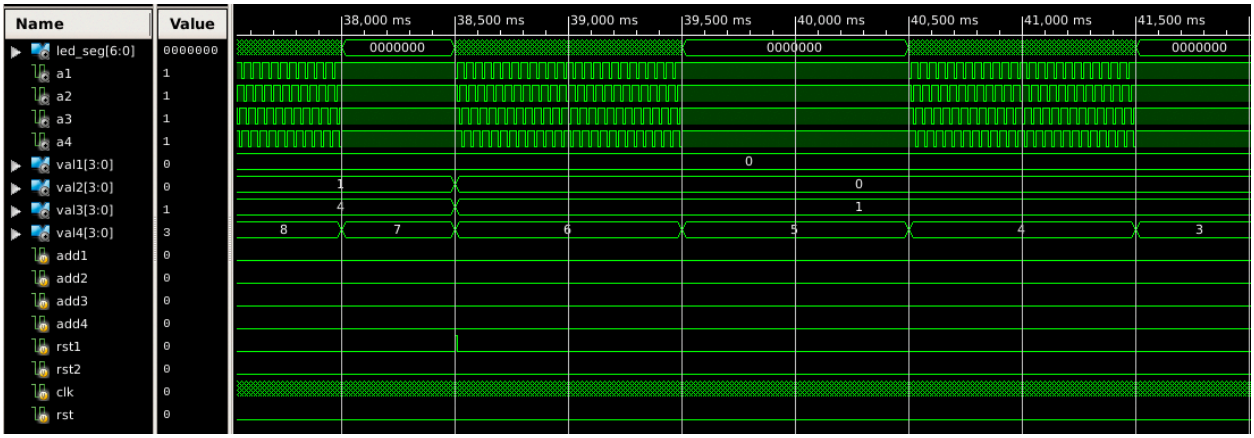

Figure 17. Simulation Waveform demonstrating the effects of when rst Signal is applied at Odd Time

3.1.14 Letting Time Expire

For the final design test, I let the time expire to 0000 at 54500ms. There, it should return to initial state where it flashes on and off every 500ms. We should also see that when the timer is 0001, the display should be off. The figure below checks both of these cases out.

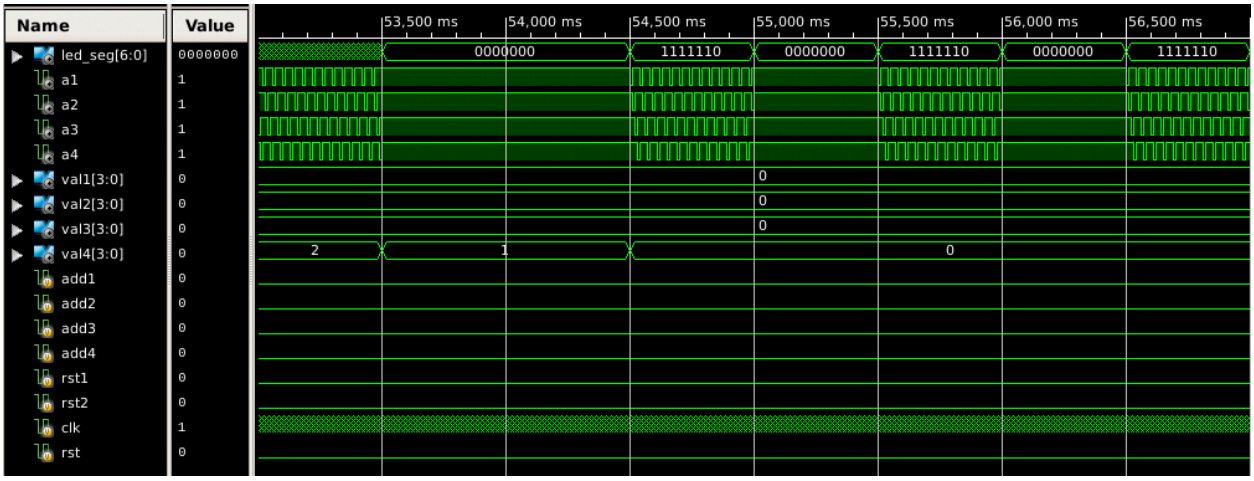

Figure 18. Simulation Waveform when Timer reaches 0000

3.2 Display Test

This section of testing checks for accurate behavior of the LED signal. For instance, when a1 is LOW, the display should correspond to the number shown in val1. This logic applies for a2, a3, and a4. The led_seg output consists of seven bits, one for each segment. Each segment will illuminate if its corresponding bit is 1, and off when it is 0. Refer to Figure 1 for more information about how the seven-segment display works.

3.2.1 Display is 0000

The first test checks for expected led_seg when the value is 0, which should be 1111110. All four values at this figure below output 0. That means no matter which anode is selected, the led_seg should continuously output 1111110. The figure below verifies that this is true.

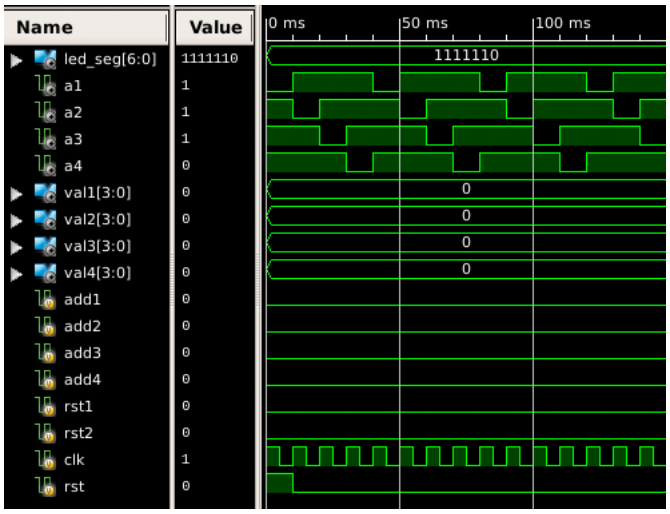

Figure 19. Simulation Waveform of led_seg at 0000

3.2.2 Nothing flashes

When the display digits are off, all anode outputs should remain HIGH and the led_seg output equal to 0000000. We can see in the waveform below that the parking meter produces the expected behavior.

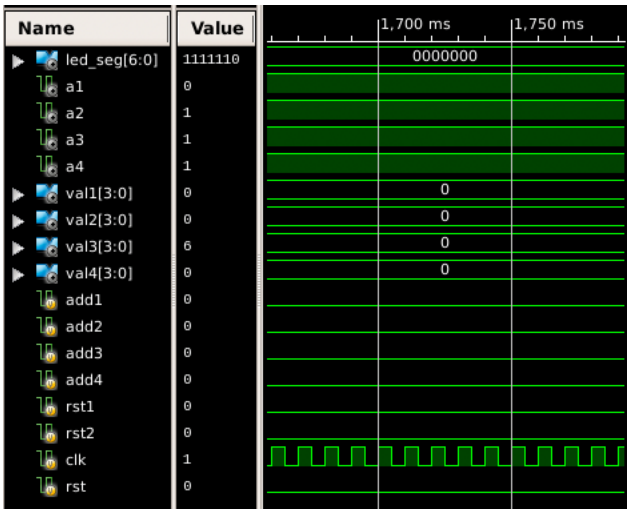

Figure 20. Simulation Waveform of led_seg when the digits are Off

3.2.3 Display is 0654

This test checks if the module selects the correct anode to output the correct segment display. At 8800ms, a1 is selected, which should display the value val1 has indicated, which is 0. Here, led_seg should output 1111110. 10ms later, a2 is selected, which should display 6, the value of val2. There, led_seg should output 1011111. At 8820s, a3 is selected, which should display 5, the value of a3. There, led_seg should output 1011011. Finally, a4 is selected at 8830ms, and it should display 4, the value of val4. The output led_seg should output 0110011. The figure below reads 1111110, 1011111, 1011011, and 0110011, which verifies all cases.

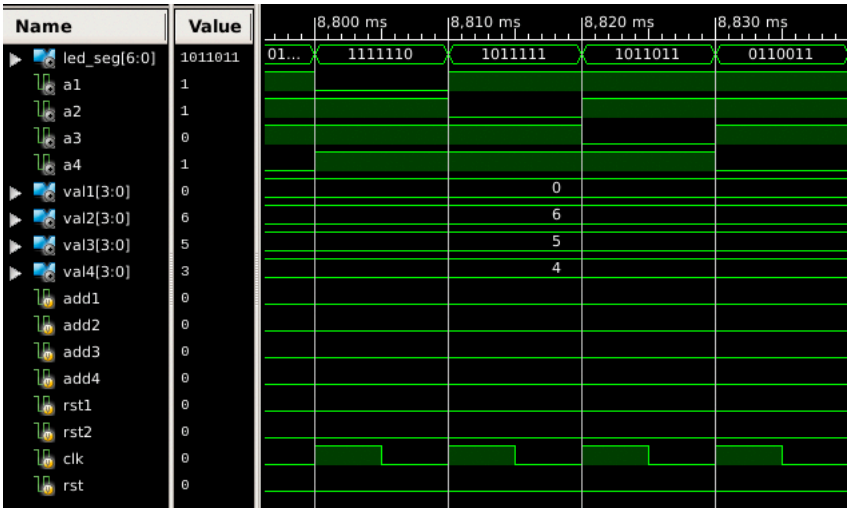

Figure 21. Simulation Waveform of led_seg at 0654

3.2.4 Display is 9999

This test checks for expected led_seg when the value is 9, which should be 1111011. All four values at this figure below output 9. That means no matter which anode is selected, the led_seg should continuously output 1111011. The figure below checks this case out.

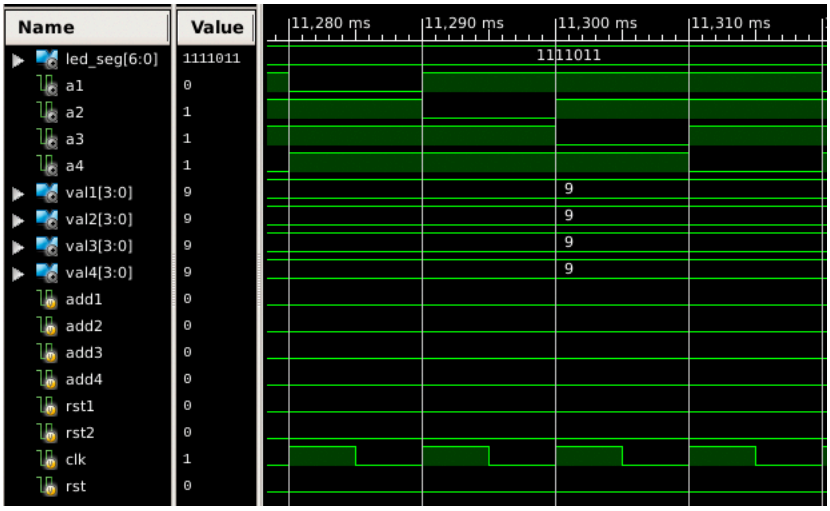

Figure 22. Simulation Waveform of led_seg at 9999

3.2.5 Display is 0072

This test checks if led_seg correctly displays the numbers 7 and 2. The first two anodes should make led_seg output 1111110, which corresponds to 0. At 17840s, a3 is selected, which should display 7, the value of a3. There, led_seg should output 1110000. Finally, a4 is selected at 17850ms, and it should display 2, the value of val4. The output led_seg should output 1101101. The figure below reads 1110000 at 17840ms, and 1101101 at 17850ms. These results shows that the led_seg is functioning as expected.

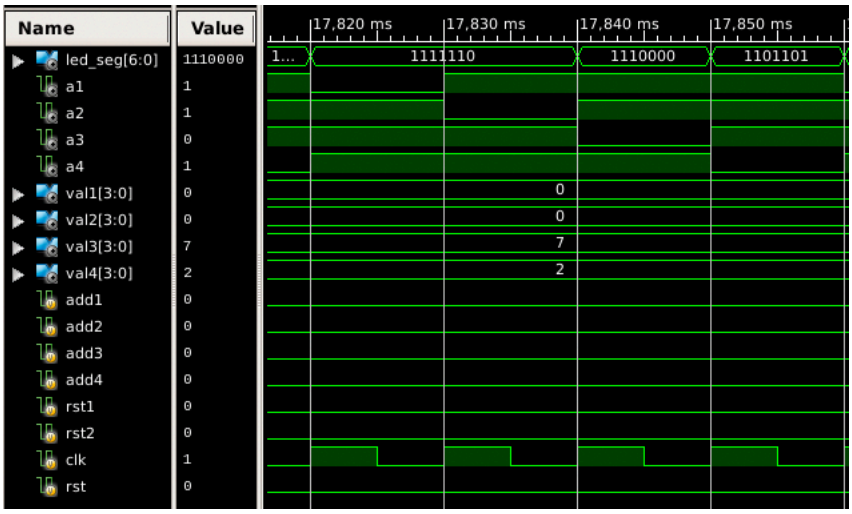

Figure 23. Simulation Waveform of led_seg at 0072

3.2.6 Display is 0183

This final test checks if led_seg correctly displays the numbers 1, 8 and 3. The first anode should make led_seg output 1111110, which corresponds to 0. At 27050ms, a2 is selected and should display 1, the value of val2. Therefore, led_seg should output 0110000. At 27060s, a3 is selected and should display 8, the value of a3. There, led_seg should output 1111111. Finally, a4 is selected at 27070ms, and it should display 3, the value of val4. The output led_seg should output 1111001. The figure below reads 0110000 at 27050ms, 1111111 at 27060ms, and 1111001 at 27070ms, which means led_seg works as intended.

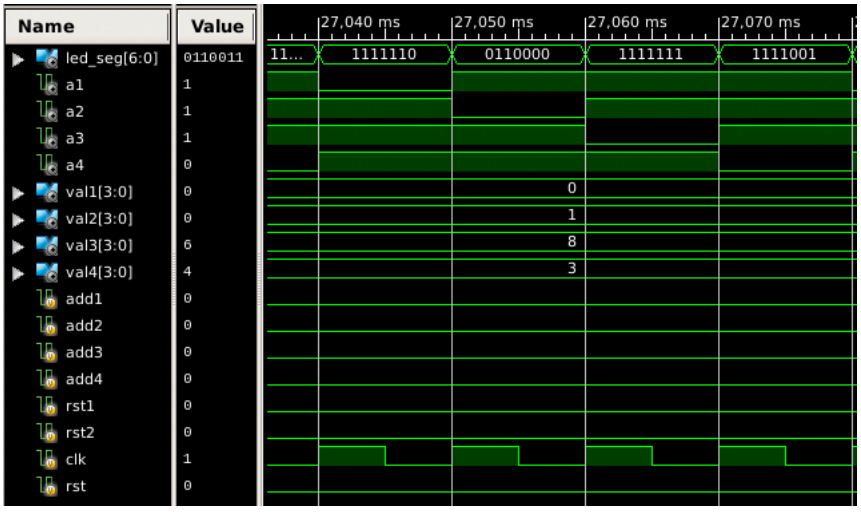

Figure 24. Simulation Waveform of led_seg at 0183

4. Synthesis and Implementation Map Report

Attached to the end of this paper are the synthesis and implementation map report. I made sure that my Verilog code is fully synthesizable by having no errors nor warnings in both reports. In both of these reports, the module uses a lot of registers and flip-flop pairs to implement a parking meter. We can see in the map report that some blocks were optimized to make the module run more efficiently.

5. Conclusion

In this lab, I learned how to design a finite state machine that outputs a display of four 7-segment LED displays by breaking the overall process down into submodules, and combine them into one working module afterwards. I had a few difficulties regarding the consistency of my code to the project manual, such as the behavior of the maximum time and how the LEDs should be displayed. However, I resolved these issues quickly after talking to the TA and implemented a parking meter that functions exactly as the project description specifies.

6. Synthesis Report

```
=========================================================================
*                            Design Summary                             *
=========================================================================

Top Level Output File Name         : parking_meter.ngc

Primitive and Black Box Usage:
------------------------------
# BELS                             : 1559
#      GND                         : 1
#      INV                         : 22
#      LUT1                        : 2
#      LUT2                        : 42
#      LUT3                        : 75
#      LUT4                        : 83
#      LUT5                        : 221
#      LUT6                        : 608
#      MUXCY                       : 225
#      MUXF7                       : 32
#      VCC                         : 1
#      XORCY                       : 247
# FlipFlops/Latches                : 88
#      FD                          : 23
#      FDE                         : 4
#      FDR                         : 46
#      FDRE                        : 13
#      FDS                         : 2
# Clock Buffers                    : 1
#      BUFGP                       : 1
# IO Buffers                       : 34
#      IBUF                        : 7
#      OBUF                        : 27

Device utilization summary:
---------------------------

Selected Device : 6slx16csg324-3


Slice Logic Utilization:
 Number of Slice Registers:              88  out of  18224     0%
 Number of Slice LUTs:                 1053  out of   9112    11%
    Number used as Logic:              1053  out of   9112    11%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:   1071
   Number with an unused Flip Flop:     983  out of   1071    91%
   Number with an unused LUT:            18  out of   1071     1%
   Number of fully used LUT-FF pairs:    70  out of   1071     6%
   Number of unique control sets:         9

IO Utilization:
 Number of IOs:                          35
 Number of bonded IOBs:                  35  out of    232    15%

Specific Feature Utilization:
 Number of BUFG/BUFGCTRLs:                1  out of     16     6%
```

```
--------------------------
Partition Resource Summary:
--------------------------

  No Partitions were found in this design.

--------------------------


=====================================================================
Timing Report

NOTE: THESE TIMING NUMBERS ARE ONLY A SYNTHESIS ESTIMATE.
      FOR ACCURATE TIMING INFORMATION PLEASE REFER TO THE TRACE REPORT
      GENERATED AFTER PLACE-and-ROUTE.

Clock Information:
------------------
-----------------------------------+------------------------+-------+
Clock Signal                       | Clock buffer(FF name)  | Load  |
-----------------------------------+------------------------+-------+
clk                                | BUFGP                  | 88    |
-----------------------------------+------------------------+-------+

Asynchronous Control Signals Information:
----------------------------------------
No asynchronous control signals found in this design

Timing Summary:
---------------
Speed Grade: -3

   Minimum period: 17.271ns (Maximum Frequency: 57.901MHz)
   Minimum input arrival time before clock: 10.715ns
   Maximum output required time after clock: 3.701ns
   Maximum combinational path delay: No path found

Timing Details:
---------------
All values displayed in nanoseconds (ns)

=====================================================================
Timing constraint: Default period analysis for Clock 'clk'
  Clock period: 17.271ns (frequency: 57.901MHz)
  Total number of paths / destination ports: 3150971455 / 121
---------------------------------------------------------------------
Delay:               17.271ns (Levels of Logic = 27)
  Source:            timer_9_1 (FF)
  Destination:       digit3_3 (FF)
  Source Clock:      clk rising
  Destination Clock: clk rising

  Data Path: timer_9_1 to digit3_3
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDR:C->Q             14   0.447   1.322  timer_9_1 (timer_9_1)
     LUT6:I0->O            1   0.203   0.000  timer[13]_PWR_1_o_div_40_OUT<6>11_G (N1217)
```

```
     MUXF7:I1->O           12   0.140   0.908  timer[13]_PWR_1_o_div_40_OUT<6>11
(timer[13]_PWR_1_o_div_40_OUT<6>)
     MUXCY:DI->O            1   0.145   0.000
timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_17_OUT_Madd_cy<6>
(timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_17_OUT_Madd_cy<6>)
     MUXCY:CI->O            1   0.019   0.000
timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_17_OUT_Madd_cy<7>
(timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_17_OUT_Madd_cy<7>)
     XORCY:CI->O           11   0.180   0.883
timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_17_OUT_Madd_xor<8>
(timer[13]_PWR_1_o_mod_41/a[13]_GND_3_o_add_17_OUT<8>)
     LUT3:I2->O             4   0.205   0.683
timer[13]_PWR_1_o_mod_41/Mmux_a[8]_a[13]_MUX_625_o11
(timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_19_OUT_Madd_lut<8>)
     MUXCY:DI->O            1   0.145   0.000
timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_19_OUT_Madd_cy<8>
(timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_19_OUT_Madd_cy<8>)
     MUXCY:CI->O            1   0.019   0.000
timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_19_OUT_Madd_cy<9>
(timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_19_OUT_Madd_cy<9>)
     MUXCY:CI->O            1   0.019   0.000
timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_19_OUT_Madd_cy<10>
(timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_19_OUT_Madd_cy<10>)
     MUXCY:CI->O            1   0.019   0.000
timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_19_OUT_Madd_cy<11>
(timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_19_OUT_Madd_cy<11>)
     MUXCY:CI->O            0   0.019   0.000
timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_19_OUT_Madd_cy<12>
(timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_19_OUT_Madd_cy<12>)
     XORCY:CI->O            7   0.180   0.774
timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_19_OUT_Madd_xor<13>
(timer[13]_PWR_1_o_mod_41/a[13]_GND_3_o_add_19_OUT<13>)
     LUT6:I5->O            17   0.205   1.028
timer[13]_PWR_1_o_mod_41/Mmux_a[13]_a[13]_MUX_634_o11
(timer[13]_PWR_1_o_mod_41/a[13]_a[13]_MUX_634_o)
     LUT6:I5->O             2   0.205   0.721
timer[13]_PWR_1_o_mod_41/BUS_0011_INV_457_o2_SW2_SW0_SW0 (N1029)
     LUT6:I4->O            10   0.203   0.856
timer[13]_PWR_1_o_mod_41/Mmux_a[5]_a[13]_MUX_656_o11
(timer[13]_PWR_1_o_mod_41/a[5]_a[13]_MUX_656_o)
     MUXCY:DI->O            1   0.145   0.000
timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_23_OUT_Madd_cy<5>
(timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_23_OUT_Madd_cy<5>)
     MUXCY:CI->O            1   0.019   0.000
timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_23_OUT_Madd_cy<6>
(timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_23_OUT_Madd_cy<6>)
     XORCY:CI->O            4   0.180   0.788
timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_23_OUT_Madd_xor<7>
(timer[13]_PWR_1_o_mod_41/a[13]_GND_3_o_add_23_OUT<7>)
     LUT6:I4->O             5   0.203   1.059
timer[13]_PWR_1_o_mod_41/Mmux_a[7]_a[13]_MUX_668_o11
(timer[13]_PWR_1_o_mod_41/a[7]_a[13]_MUX_668_o)
     LUT6:I1->O            14   0.203   1.062  timer[13]_PWR_1_o_mod_41/BUS_0013_INV_487_o2_SW0
(N81)
     LUT6:I4->O             5   0.203   0.714
timer[13]_PWR_1_o_mod_41/Mmux_a[4]_a[13]_MUX_685_o11
(timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_27_OUT_Madd_lut<4>)
     MUXCY:DI->O            1   0.145   0.000
timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_27_OUT_Madd_cy<4>
(timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_27_OUT_Madd_cy<4>)
```

```
     MUXCY:CI->O           1   0.019   0.000
timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_27_OUT_Madd_cy<5>
(timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_27_OUT_Madd_cy<5>)
     XORCY:CI->O           2   0.180   0.845
timer[13]_PWR_1_o_mod_41/Madd_a[13]_GND_3_o_add_27_OUT_Madd_xor<6>
(timer[13]_PWR_1_o_mod_41/a[13]_GND_3_o_add_27_OUT<6>)
     LUT5:I2->O            1   0.205   0.580
timer[13]_PWR_1_o_mod_41/BUS_0015_INV_517_o26_SW0_SW0 (N401)
     LUT6:I5->O            1   0.205   0.684  timer[13]_PWR_1_o_mod_41/BUS_0015_INV_517_o26_SW0
(N183)
     LUT6:I4->O            1   0.203   0.000  timer[13]_PWR_1_o_mod_41/Mmux_o41
(timer[13]_PWR_1_o_mod_41_OUT<3>)
     FD:D                      0.102          digit3_3
    ----------------------------------------
    Total                     17.271ns (4.365ns logic, 12.906ns route)
                                       (25.3% logic, 74.7% route)

==========================================================================
Timing constraint: Default OFFSET IN BEFORE for Clock 'clk'
  Total number of paths / destination ports: 77543 / 107
--------------------------------------------------------------------------
Offset:              10.715ns (Levels of Logic = 11)
  Source:            add1 (PAD)
  Destination:       duty_counter_4 (FF)
  Destination Clock: clk rising

  Data Path: add1 to duty_counter_4
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O            13   1.222   1.277  add1_IBUF (add1_IBUF)
     LUT6:I1->O           10   0.203   1.085  Mmux_n015010 (n0150<5>)
     LUT5:I2->O           13   0.205   0.933  Madd_timer[13]_GND_1_o_add_11_OUT_cy<7>11
(Madd_timer[13]_GND_1_o_add_11_OUT_cy<7>)
     LUT4:I3->O            7   0.205   0.774  Madd_timer[13]_GND_1_o_add_11_OUT_cy<10>11
(Madd_timer[13]_GND_1_o_add_11_OUT_cy<10>)
     LUT5:I4->O            1   0.205   0.580  PWR_1_o_timer[13]_LessThan_18_o110111
(PWR_1_o_timer[13]_LessThan_18_o11011)
     LUT6:I5->O            1   0.205   0.580  PWR_1_o_timer[13]_LessThan_18_o12
(PWR_1_o_timer[13]_LessThan_18_o11)
     LUT6:I5->O            1   0.205   0.000  PWR_1_o_timer[13]_LessThan_18_o19_F (N1202)
     MUXF7:I0->O           1   0.131   0.580  PWR_1_o_timer[13]_LessThan_18_o19
(PWR_1_o_timer[13]_LessThan_18_o18)
     LUT6:I5->O           17   0.205   1.028  PWR_1_o_timer[13]_LessThan_18_o112
(PWR_1_o_timer[13]_LessThan_18_o)
     LUT5:I4->O            1   0.205   0.580  Mmux_duty_counter[6]_GND_1_o_mux_19_OUT5_SW1
(N1206)
     LUT6:I5->O            1   0.205   0.000  Mmux_duty_counter[6]_GND_1_o_mux_19_OUT5
(duty_counter[6]_GND_1_o_mux_19_OUT<4>)
     FDR:D                     0.102          duty_counter_4
    ----------------------------------------
    Total                     10.715ns (3.298ns logic, 7.417ns route)
                                       (30.8% logic, 69.2% route)

==========================================================================
Timing constraint: Default OFFSET OUT AFTER for Clock 'clk'
  Total number of paths / destination ports: 27 / 27
--------------------------------------------------------------------------
Offset:              3.701ns (Levels of Logic = 1)
  Source:            digit4_0 (FF)
```

```
  Destination:       val4<0> (PAD)
  Source Clock:      clk rising

  Data Path: digit4_0 to val4<0>
                               Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FD:C->Q               4   0.447   0.683  digit4_0 (digit4_0)
     OBUF:I->O                 2.571          val4_0_OBUF (val4<0>)
    ----------------------------------------
    Total                      3.701ns (3.018ns logic, 0.683ns route)
                                       (81.6% logic, 18.4% route)

==========================================================================

Cross Clock Domains Report:
--------------------------

Clock to Setup on destination clock clk
---------------+---------+---------+---------+---------+
               | Src:Rise| Src:Fall| Src:Rise| Src:Fall|
Source Clock   |Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
---------------+---------+---------+---------+---------+
clk            |   17.271|         |         |         |
---------------+---------+---------+---------+---------+

==========================================================================


Total REAL time to Xst completion: 20.00 secs
Total CPU time to Xst completion: 18.02 secs

-->


Total memory usage is 491584 kilobytes

Number of errors   :    0 (   0 filtered)
Number of warnings :    0 (   0 filtered)
Number of infos    :    1 (   0 filtered)
```

7. Implementation Report

```
Release 14.7 Map P.20131013 (lin64)
Xilinx Mapping Report File for Design 'parking_meter'

Design Information
------------------
Command Line   : map -intstyle ise -p xc6slx16-csg324-3 -w -logic_opt off -ol
high -t 1 -xt 0 -register_duplication off -r 4 -global_opt off -mt off -ir off
-pr off -lc off -power off -o parking_meter_map.ncd parking_meter.ngd
parking_meter.pcf
Target Device  : xc6slx16
Target Package : csg324
Target Speed   : -3
Mapper Version : spartan6 -- $Revision: 1.55 $
Mapped Date    : Mon Mar  8 19:56:53 2021

Design Summary
--------------
Number of errors:      0
Number of warnings:    0
Slice Logic Utilization:
  Number of Slice Registers:                    88 out of  18,224    1%
    Number used as Flip Flops:                  88
    Number used as Latches:                      0
    Number used as Latch-thrus:                  0
    Number used as AND/OR logics:                0
  Number of Slice LUTs:                     1,035 out of   9,112   11%
    Number used as logic:                   1,034 out of   9,112   11%
      Number using O6 output only:            913
      Number using O5 output only:              3
      Number using O5 and O6:                 118
      Number used as ROM:                       0
    Number used as Memory:                      0 out of   2,176    0%
    Number used exclusively as route-thrus:     1
      Number with same-slice register load:     0
      Number with same-slice carry load:        1
      Number with other load:                   0

Slice Logic Distribution:
  Number of occupied Slices:                  340 out of   2,278   14%
  Number of MUXCYs used:                      276 out of   4,556    6%
  Number of LUT Flip Flop pairs used:       1,050
    Number with an unused Flip Flop:          970 out of   1,050   92%
    Number with an unused LUT:                 15 out of   1,050    1%
    Number of fully used LUT-FF pairs:         65 out of   1,050    6%
    Number of unique control sets:              9
    Number of slice register sites lost
      to control set restrictions:             32 out of  18,224    1%

  A LUT Flip Flop pair for this architecture represents one LUT paired with
  one Flip Flop within a slice.  A control set is a unique combination of
  clock, reset, set, and enable signals for a registered element.
  The Slice Logic Distribution report is not meaningful if the design is
  over-mapped for a non-slice resource or if Placement fails.

IO Utilization:
  Number of bonded IOBs:                       35 out of     232   15%
```

```
Specific Feature Utilization:
  Number of RAMB16BWERs:                        0 out of      32    0%
  Number of RAMB8BWERs:                         0 out of      64    0%
  Number of BUFIO2/BUFIO2_2CLKs:                0 out of      32    0%
  Number of BUFIO2FB/BUFIO2FB_2CLKs:            0 out of      32    0%
  Number of BUFG/BUFGMUXs:                      1 out of      16    6%
    Number used as BUFGs:                       1
    Number used as BUFGMUX:                     0
  Number of DCM/DCM_CLKGENs:                    0 out of       4    0%
  Number of ILOGIC2/ISERDES2s:                  0 out of     248    0%
  Number of IODELAY2/IODRP2/IODRP2_MCBs:        0 out of     248    0%
  Number of OLOGIC2/OSERDES2s:                  0 out of     248    0%
  Number of BSCANs:                             0 out of       4    0%
  Number of BUFHs:                              0 out of     128    0%
  Number of BUFPLLs:                            0 out of       8    0%
  Number of BUFPLL_MCBs:                        0 out of       4    0%
  Number of DSP48A1s:                           0 out of      32    0%
  Number of ICAPs:                              0 out of       1    0%
  Number of MCBs:                               0 out of       2    0%
  Number of PCILOGICSEs:                        0 out of       2    0%
  Number of PLL_ADVs:                           0 out of       2    0%
  Number of PMVs:                               0 out of       1    0%
  Number of STARTUPs:                           0 out of       1    0%
  Number of SUSPEND_SYNCs:                      0 out of       1    0%

Average Fanout of Non-Clock Nets:               4.87

Peak Memory Usage:  779 MB
Total REAL time to MAP completion:  16 secs
Total CPU time to MAP completion:   15 secs

Table of Contents
-----------------
Section 1 - Errors
Section 2 - Warnings
Section 3 - Informational
Section 4 - Removed Logic Summary
Section 5 - Removed Logic
Section 6 - IOB Properties
Section 7 - RPMs
Section 8 - Guide Report
Section 9 - Area Group and Partition Summary
Section 10 - Timing Report
Section 11 - Configuration String Information
Section 12 - Control Set Information
Section 13 - Utilization by Hierarchy

Section 1 - Errors
------------------

Section 2 - Warnings
--------------------

Section 3 - Informational
-------------------------
INFO:MapLib:562 - No environment variables are currently set.
INFO:LIT:244 - All of the single ended outputs in this design are using slew
   rate limited output drivers. The delay on speed critical single ended outputs
   can be dramatically reduced by designating them as fast outputs.
INFO:Pack:1716 - Initializing temperature to 85.000 Celsius. (default - Range:
```

```
   0.000 to 85.000 Celsius)
INFO:Pack:1720 - Initializing voltage to 1.140 Volts. (default - Range: 1.140 to
   1.260 Volts)
INFO:Map:215 - The Interim Design Summary has been generated in the MAP Report
   (.mrp).
INFO:Pack:1650 - Map created a placed design.

Section 4 - Removed Logic Summary
---------------------------------
   2 block(s) optimized away

Section 5 - Removed Logic
-------------------------

Optimized Block(s):
TYPE           BLOCK
GND            XST_GND
VCC            XST_VCC

To enable printing of redundant blocks removed and signals merged, set the
detailed map report option and rerun map.

Section 6 - IOB Properties
--------------------------

+----------------------------------------------------------------------------------------------------
-----------------------------------------------------------+
| IOB Name                           | Type             | Direction | IO Standard          | Diff
| Drive    | Slew | Reg (s)      | Resistor | IOB      |
|                                    |                  |           |                      | Term
| Strength | Rate |              |          | Delay    |
+----------------------------------------------------------------------------------------------------
-----------------------------------------------------------+
| a1                                 | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |          |
| a2                                 | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |          |
| a3                                 | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |          |
| a4                                 | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |          |
| add1                               | IOB              | INPUT     | LVCMOS25             |
|          |      |              |          |          |
| add2                               | IOB              | INPUT     | LVCMOS25             |
|          |      |              |          |          |
| add3                               | IOB              | INPUT     | LVCMOS25             |
|          |      |              |          |          |
| add4                               | IOB              | INPUT     | LVCMOS25             |
|          |      |              |          |          |
| clk                                | IOB              | INPUT     | LVCMOS25             |
|          |      |              |          |          |
| led_seg<0>                         | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |          |
| led_seg<1>                         | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |          |
| led_seg<2>                         | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |          |
| led_seg<3>                         | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |          |
```

```
| led_seg<4>                          | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |           |
| led_seg<5>                          | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |           |
| led_seg<6>                          | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |           |
| rst                                 | IOB              | INPUT     | LVCMOS25             |
|          |      |              |          |           |
| rst1                                | IOB              | INPUT     | LVCMOS25             |
|          |      |              |          |           |
| rst2                                | IOB              | INPUT     | LVCMOS25             |
|          |      |              |          |           |
| val1<0>                             | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |           |
| val1<1>                             | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |           |
| val1<2>                             | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |           |
| val1<3>                             | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |           |
| val2<0>                             | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |           |
| val2<1>                             | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |           |
| val2<2>                             | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |           |
| val2<3>                             | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |           |
| val3<0>                             | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |           |
| val3<1>                             | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |           |
| val3<2>                             | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |           |
| val3<3>                             | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |           |
| val4<0>                             | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |           |
| val4<1>                             | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |           |
| val4<2>                             | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |           |
| val4<3>                             | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |           |
+-------------------------------------------------------------------------------------------------------
-------------------------------------------------------+

Section 7 - RPMs
----------------

Section 8 - Guide Report
------------------------
Guide not run on this design.

Section 9 - Area Group and Partition Summary
--------------------------------------------

Partition Implementation Status
-------------------------------
```

```
  No Partitions were found in this design.

-------------------------------

Area Group Information
----------------------

  No area groups were found in this design.

----------------------

Section 10 - Timing Report
--------------------------
A logic-level (pre-route) timing report can be generated by using Xilinx static
timing analysis tools, Timing Analyzer (GUI) or TRCE (command line), with the
mapped NCD and PCF files. Please note that this timing report will be generated
using estimated delay information. For accurate numbers, please generate a
timing report with the post Place and Route NCD file.

For more information about the Timing Analyzer, consult the Xilinx Timing
Analyzer Reference Manual; for more information about TRCE, consult the Xilinx
Command Line Tools User Guide "TRACE" chapter.

Section 11 - Configuration String Details
-----------------------------------------
Use the "-detail" map option to print out Configuration Strings

Section 12 - Control Set Information
------------------------------------
Use the "-detail" map option to print out Control Set Information.

Section 13 - Utilization by Hierarchy
-------------------------------------
Use the "-detail" map option to print out the Utilization by Hierarchy section.
```