CSSE4010 A4

David Gaul

s4671313

Lab Session: Wednesday 10-12

Submission Date: 19th-ish September

1 Introduction

Prac 4 required the creation of a finite state machine (FSM) that would generate a high output for two clock cycles when a consecutive sequence of ”11001” was input into it. This could be approached with two different methods

- Creating a singular FSM that would check that the correct sequence had been input, as well as keeping track of the number of clock cycles on the output
- Developing two inter-linked FSMs. One for checking for the correct sequence, the other to keep track of the number of clock cycles

I'm trying to stay positive about how interesting this report is going to be.

But I'm boring.

I run out of things to say after the first date. And now we're up to the fourth.

Prepare for a report that's as bland as my soul.

2 State and Block Diagrams

Since it was possible to approach this design problem with two separate methods, both should be addressed with relevant state diagrams. The system containing the single FSM is described in figure 1.

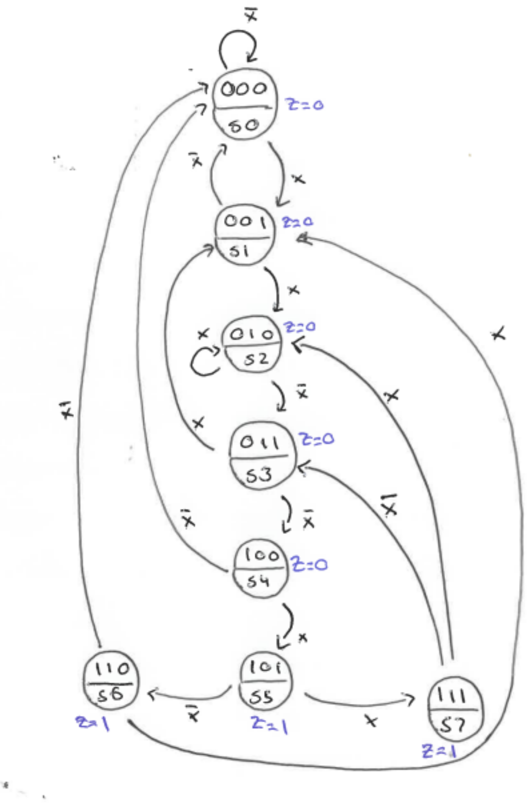


Figure 1: State diagram of the system employing a singular FSM

A fair bit of complexity can be observed within the design of a singular FSM. Due to the heavy inter-linking of states and reliance on previous states, this model becomes difficult to scale appropriately. Although it would perform the task, its flexibility is heavily limited. For this reason, a design employing two FSMs was proposed. The state diagram for this design can be found in figure 2

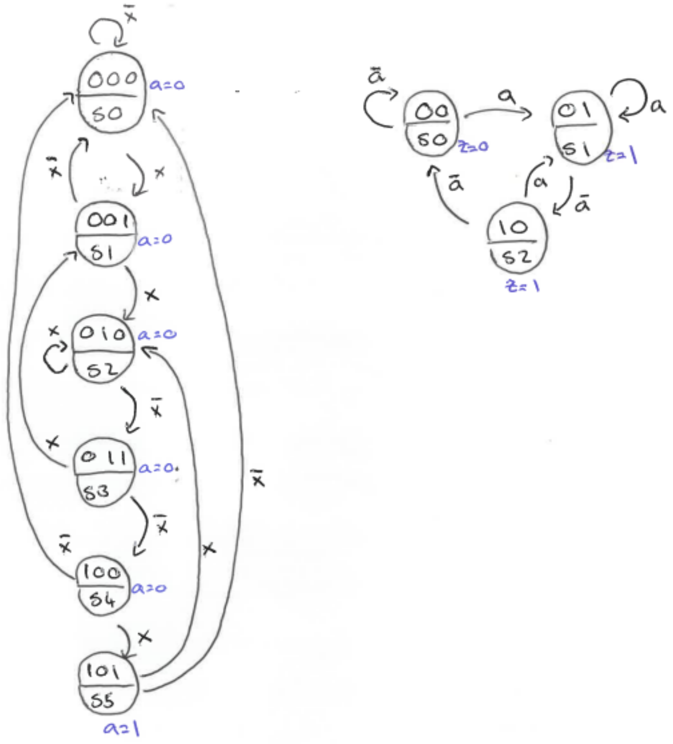


Figure 2: State diagram of the system employing two FSMs

The second design was ultimately implemented, and will be discussed for the remainder of the report. To implement the state diagram within Xilinx, a design architecture was adopted to accommodate for a sequence detector as well as the clock counter. The block diagram for this architecture can be found within figure 3

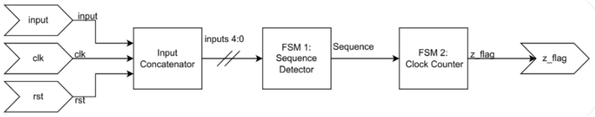


Figure 3: Block diagram representation of the design architecture

3 Simulation Results

Several test cases had to be evaluated during simulation. The full set of tests that were run, and will be explored are as follows:

- A base test case receiving 11001, with no overlap and no reset
- A test case receiving 11001 with no overlap, but receiving a reset

- A test case receiving an input of 1011001010 to test for detection in the middle of a sequence
- A test case receiving an overlapping input of 110011001 with no reset
- A test case receiving an overlapping input of 110011001 with a reset in the middle

The full range of test cases can be found in figures 4, 5, 6, 7 and 8.

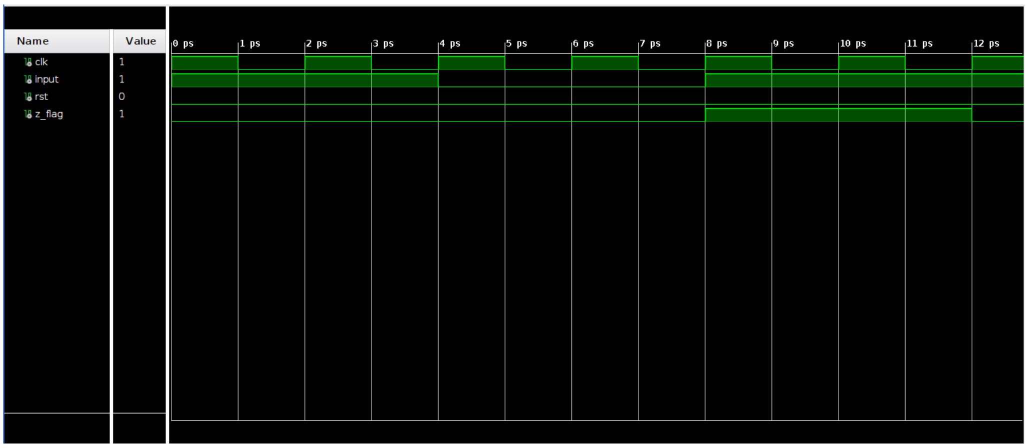


Figure 4: Simulation results of a base case receiving 11001 with no overlap and no reset

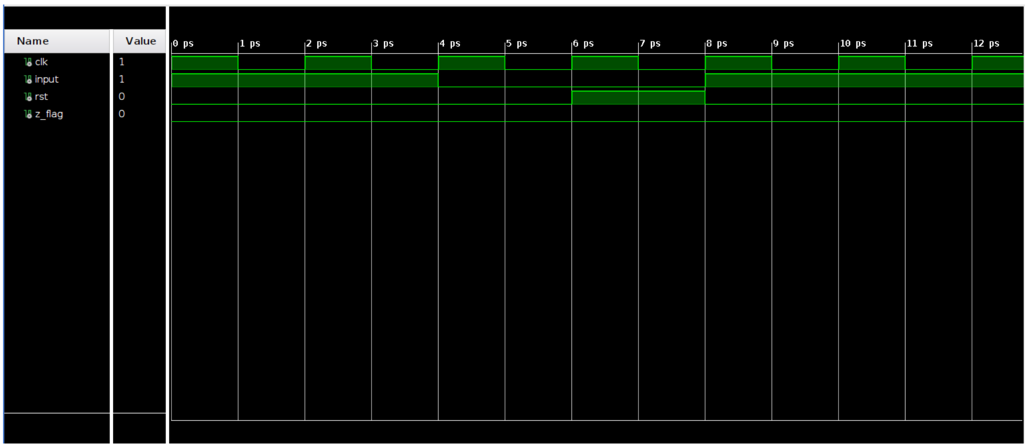


Figure 5: Simulation results of a base case receiving 11001 with no overlap, but receiving a reset

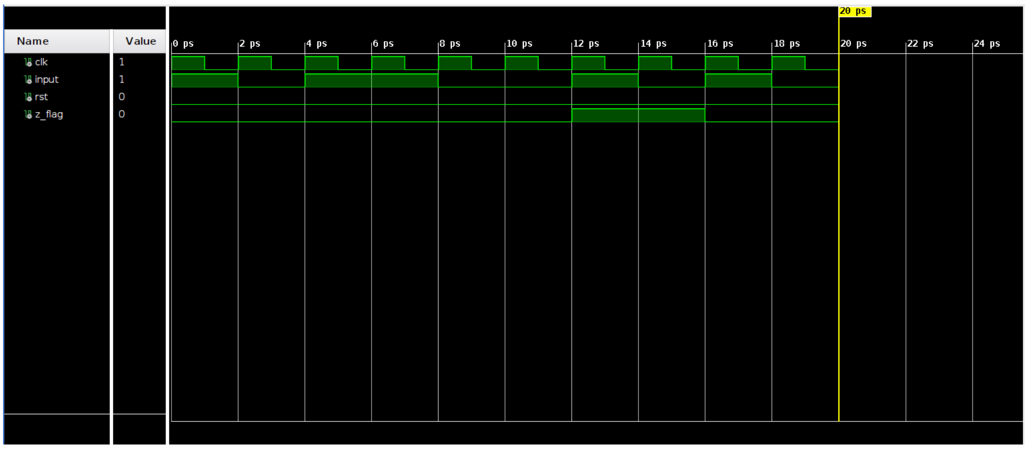


Figure 6: Simulation results of receiving an input of 1011001010 to show detection in the middle of a sequence

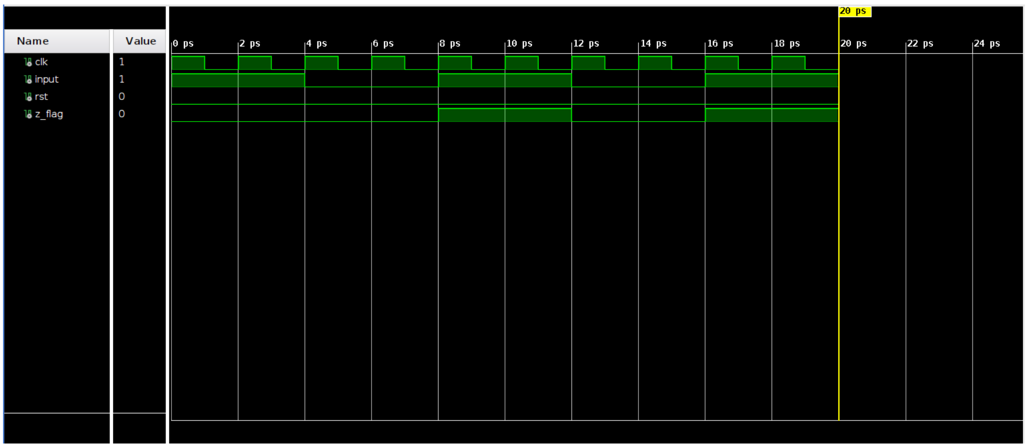


Figure 7: Simulation results of receiving 110011001 with overlap, but no reset

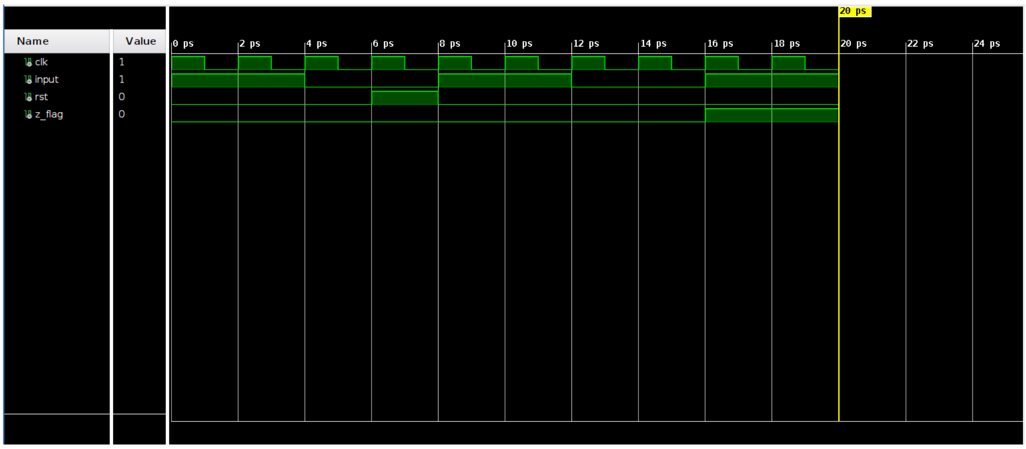


Figure 8: Simulation results of receiving 110011001 with overlap with a reset

4 Register Transfer Level Schematic

Following the successful implementation of the design, it's important to analyse the generated RTL schematics. Figure 9 shows the top level design of the entire schematic.

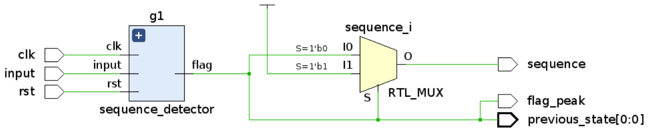


Figure 9: RTL schematic of the top level design

As can be seen in figure 9, the entire design can be broken into two main parts, each representing their respective FSM. The second FSM, receives an input from the first FSM, the input detector. It also receives an input clock line in order to main synchronicity, which becomes a serious issue for designing multi-FSM designs in Vivado. Within this figure, flag_peak and previous_state were signals used to peak at the states of both FSMs during testing. They can be ignored for simulation purposes.

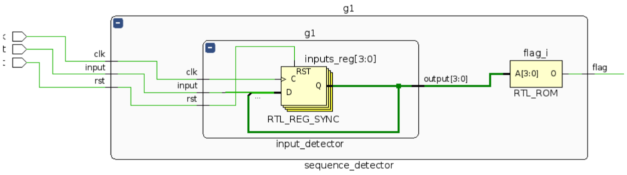


Figure 10: Zoomed in RTL schematic for FSM 1

A zoomed in view of the RTL schematic for FSM 1 can be found in figure 10. This comprises of a flip-flop feeding into a register which stores the input sequence, as described within the design architecture of this report.

5 Synthesis Results

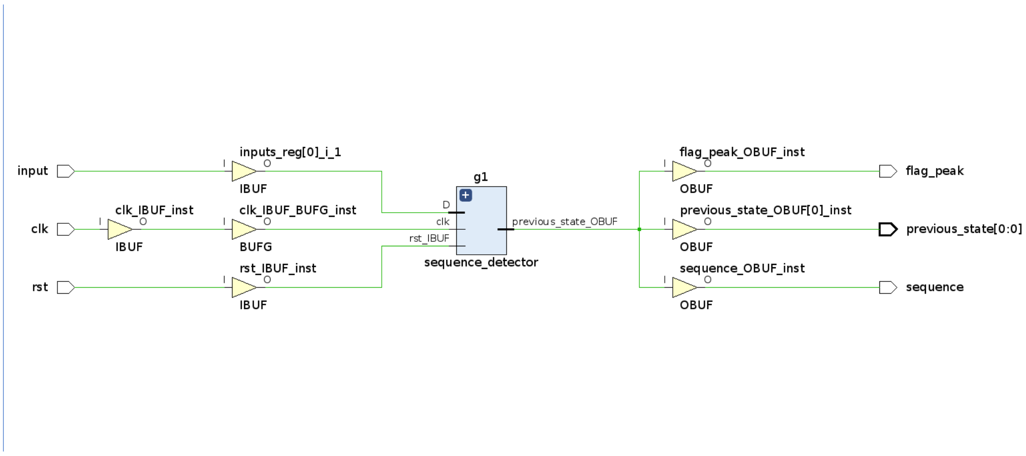
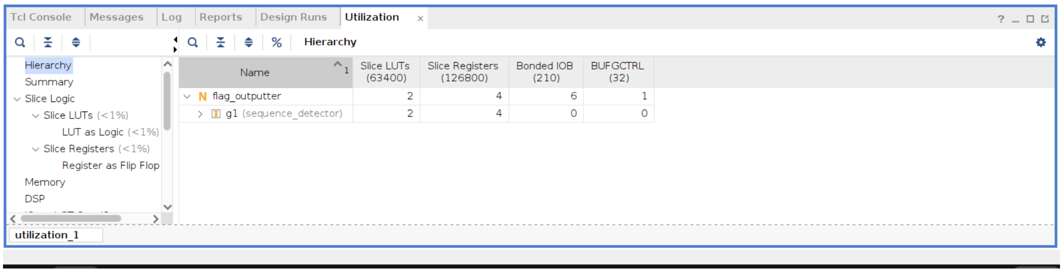


Figure 11: Synthesis schematic design

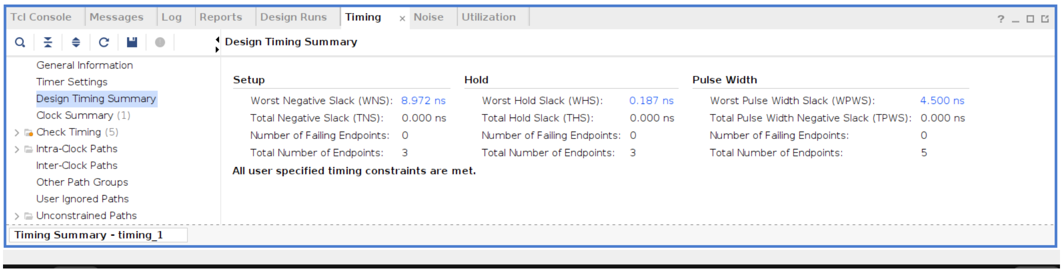
Similar to the RTL schematics shown within their relevant section, figure 11 shows the synthesized schematic. It also provides information as to the register configurations, and where the outputs and inputs are stored within the total design.



Name	Slice LUTs (63400)	Slice Registers (126800)	Bonded IOB (210)	BUFGCTRL (32)
flag_outputter	2	4	6	1
g1 (sequence_detector)	2	4	0	0

Figure 12: Synthesis results displaying the memory utilisation

Figure 12 shows the total memory utilisation of this design. In perhaps the lowest memory consumption of this entire course, the prac 4 system requires 2 look-up tables used within FSM 1. Totalling in at 4 registers, less than 1% of the entire memory was utilised for full functionality.



Design Timing Summary

```
Setup                                   Hold                                  Pulse Width
  Worst Negative Slack (WNS):  8.972 ns   Worst Hold Slack (WHS):       0.187 ns   Worst Pulse Width Slack (WPWS):            4.500 ns
  Total Negative Slack (TNS):  0.000 ns   Total Hold Slack (THS):       0.000 ns   Total Pulse Width Negative Slack (TPWS):   0.000 ns
  Number of Failing Endpoints: 0          Number of Failing Endpoints:  0          Number of Failing Endpoints:               0
  Total Number of Endpoints:   3          Total Number of Endpoints:    3          Total Number of Endpoints:                 5
All user specified timing constraints are met.
```

Figure 13: Simulation results showing the timing slack. Note that there are no timing delays that exist within the design

Another important metric to evaluate is the timing slack that appears. In order to ascertain the synchronicity of the design, no timing slack can be present. Figure 13 shows that no slack is present within the design, indicating that no timing issues are present and synchronicity is maintained.

6 Conclusion

In conclusion, the system worked as expected. An FSM was generated that was able to output a high signal for two clock counts when the input into the FSM was "11001". This was achieved by using a double FSM system, where the first FSM detected the sequence and the second counted clock cycles. Everything worked as per the plan.

See you next time for the final exciting epsiode of CSSE4011 prac reports!