Workshop 1
Exercise 1: Inverter
The input from the switch is connected directly to one LED, and via an inverter (a NOT gate) to the other LED.

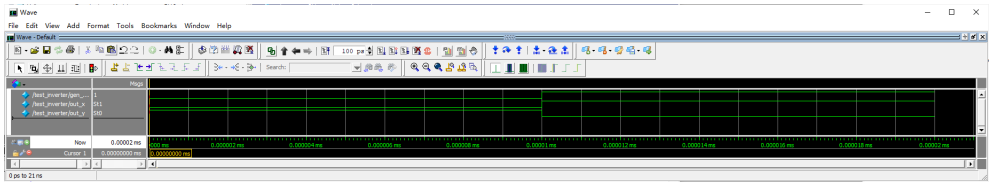

Figure 1: Inverter.

Exercise 1: Multiplexer
Exercise 2 demonstrates how module instances can be nested.

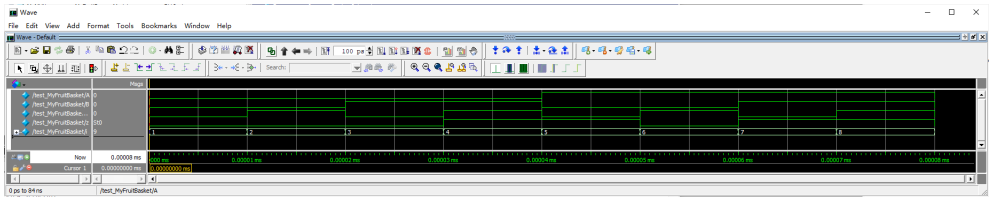

Figure 2: Multiplexer.

Part 2
The objective is for you to be able to do these fluently, without looking at the Workshop 1 slides. You will make a game for a (very) young child. It will use four switches and four LEDs. When played correctly, the game goes as follows.
1. Initially, all the switches should be in the OFF position, and there will be a single LED on, LED[2].
2. The player must switch SW[2] on (because this corresponds to LED[2] which is the only LED that is on).
3. LED[2] will remain on, and a new LED will come on, say, LED[3].
4. The player must keep SW[2] on and also switch SW[3] on.
5. A third LED will now come on, and the process repeats.
6. The game ends when all LEDs and all switches are ON.

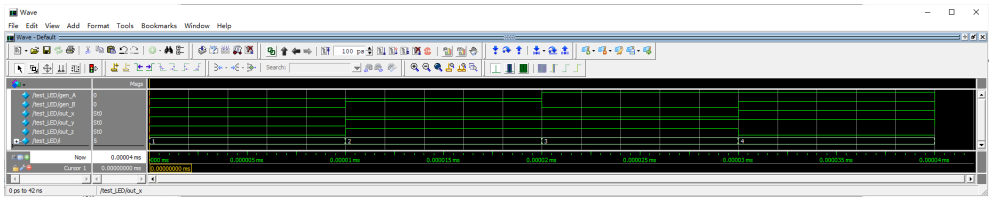

Figure 3: Game 1.

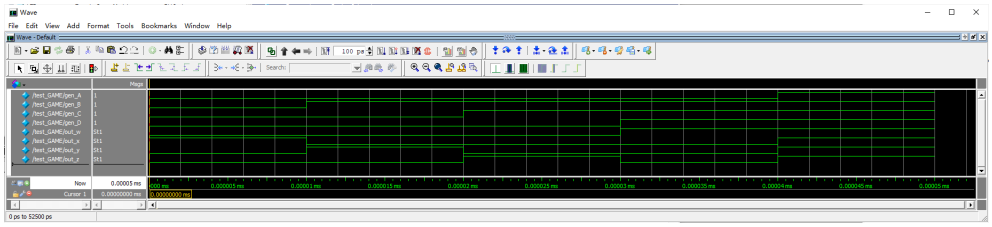

Figure 4: Game 2.

Workshop 2
Exercise 1: Multiplexers Revisited
In Exercise 2 of Workshop 1 we implemented a 2 to 1 multiplexer based on a Boolean expression that satisfied the truth table we needed. For practice, we broke the design into three smaller parts.

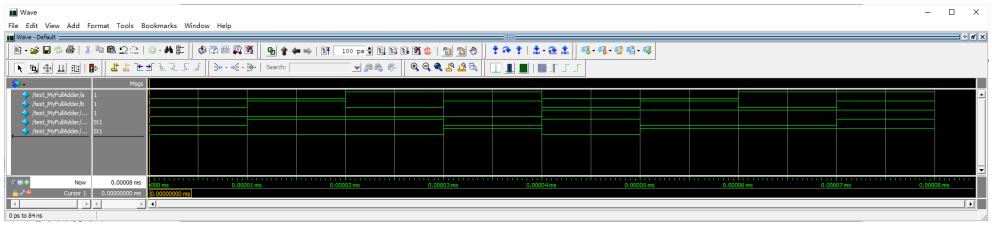

Figure 5: W2E1.

Exercise 2: Adding Numbers
Numbers and Arithmetic in Digital Electronics

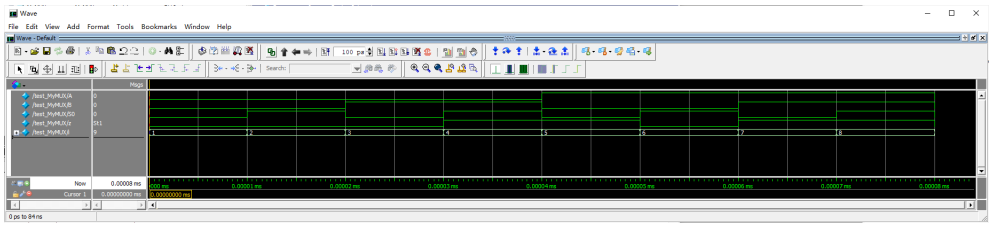

Figure 6: W2E1.

Workshop 3
Exercise 1: Timing and Glitches

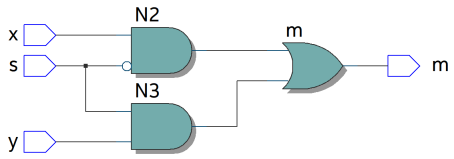

Figure 1: MyMUX RTL viewer.

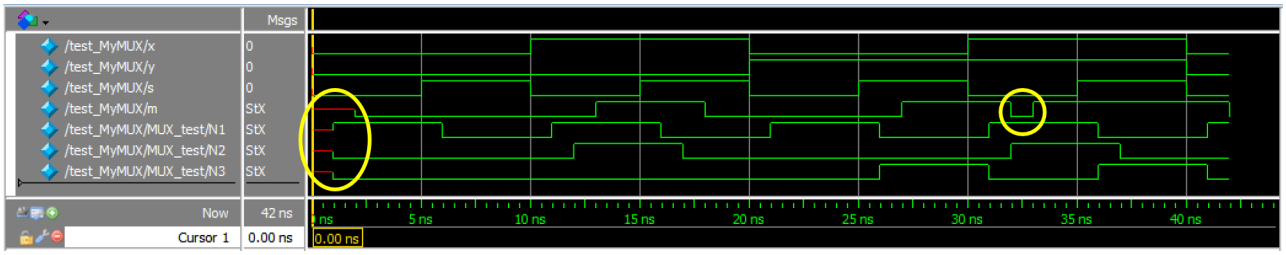

Figure 2: Glitch revealed in simulation.

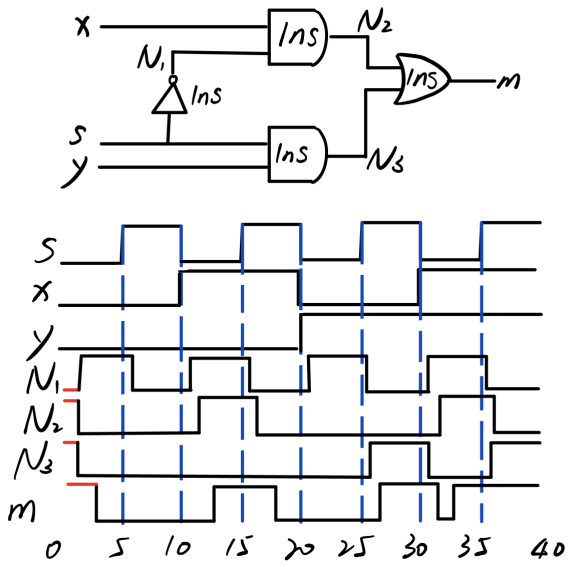

Figure 3: MyMUX Timing diagram.

Exercise 2: A Synchronised MUX

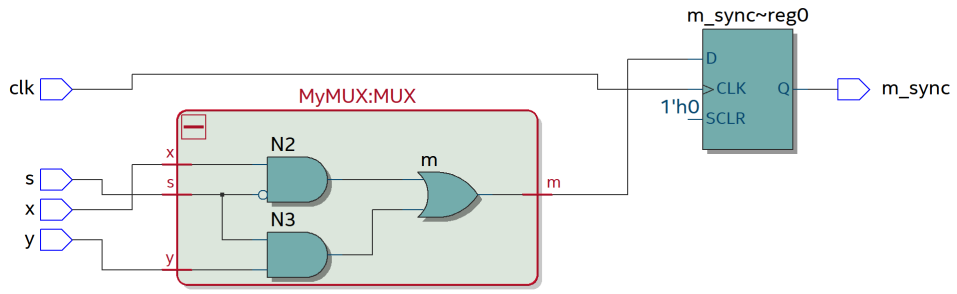

Figure 4: MySyncMUX RTL viewer.

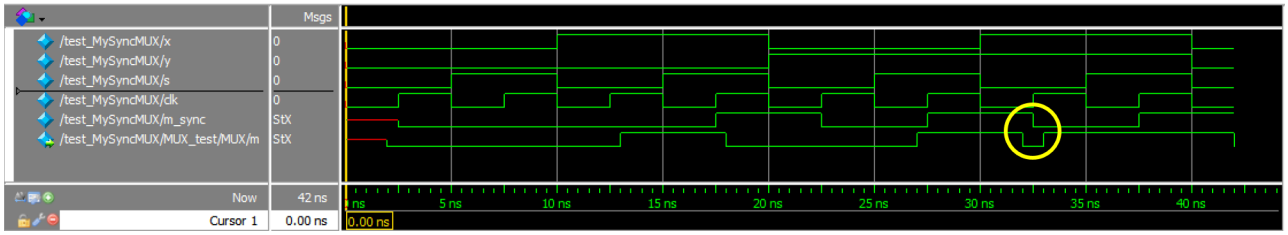

Figure 5: Glitch blocked by flip-flop.

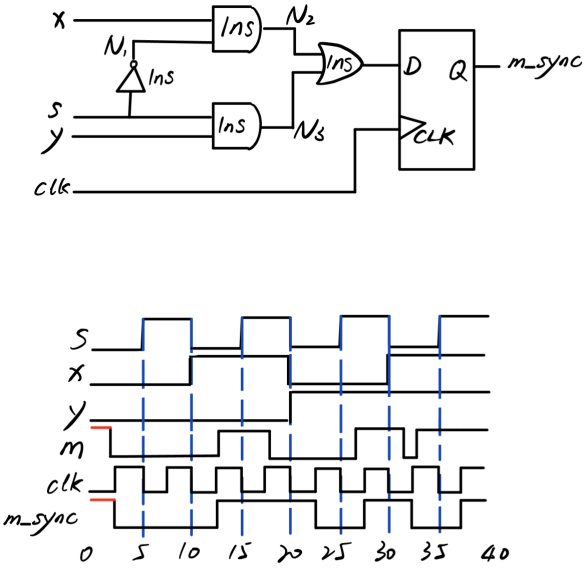

Figure 6: MySyncMUX Timing diagram.

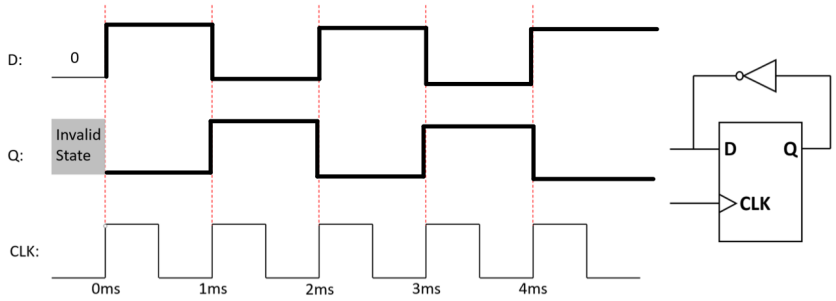

Figure 7: Ideal Flip-flop Timing diagram.

Exercise 3: Let's Count

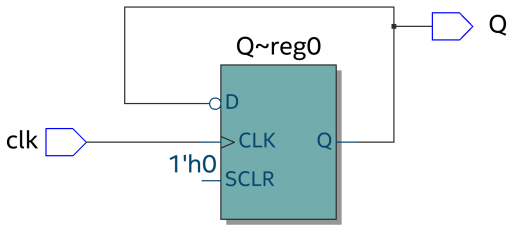

Figure 8: MyClkDivider RTL viewer.

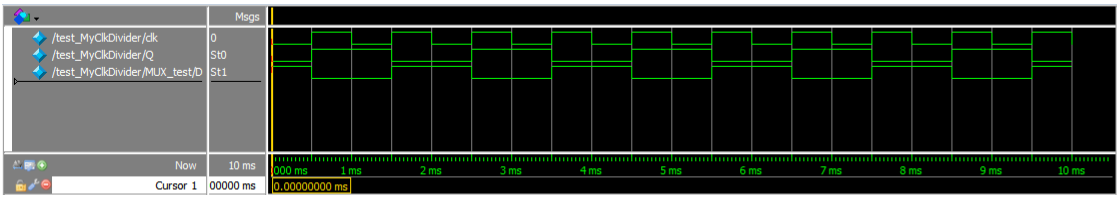

Figure 9: MyClkDivider RTL Simulation.

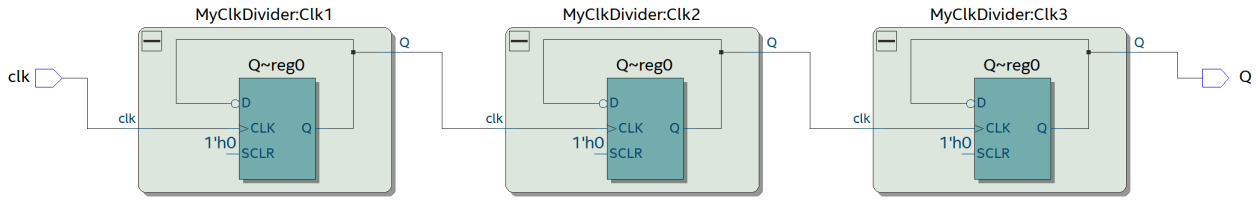

Figure 10: ThreeDividers RTL Simulation.

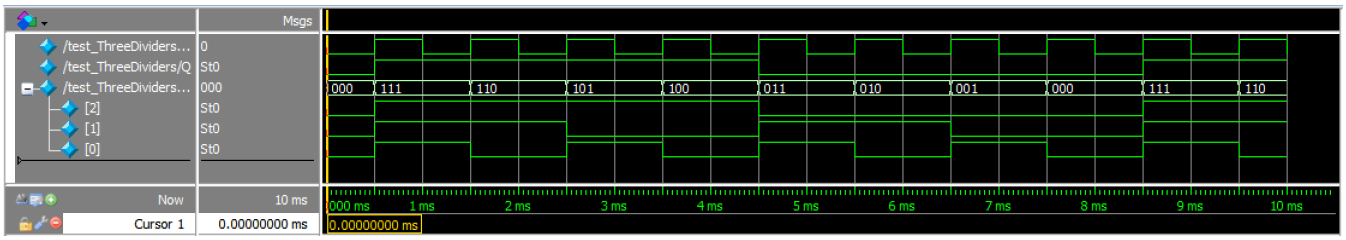

Figure 11: ThreeDividers waveforms.

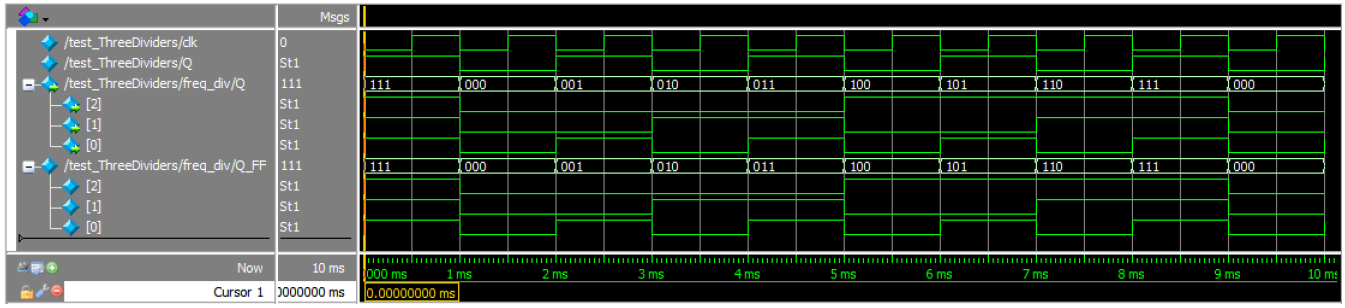

Figure 12: ThreeDividers waveforms.

Exercise 4: Synchronous Counter

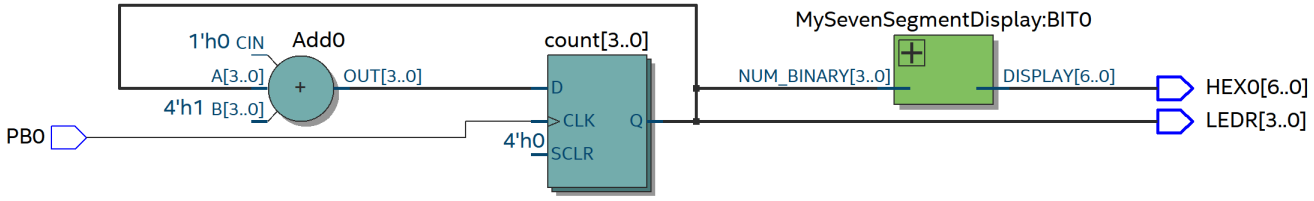

Figure 13: Synchronous Counter RTL view.

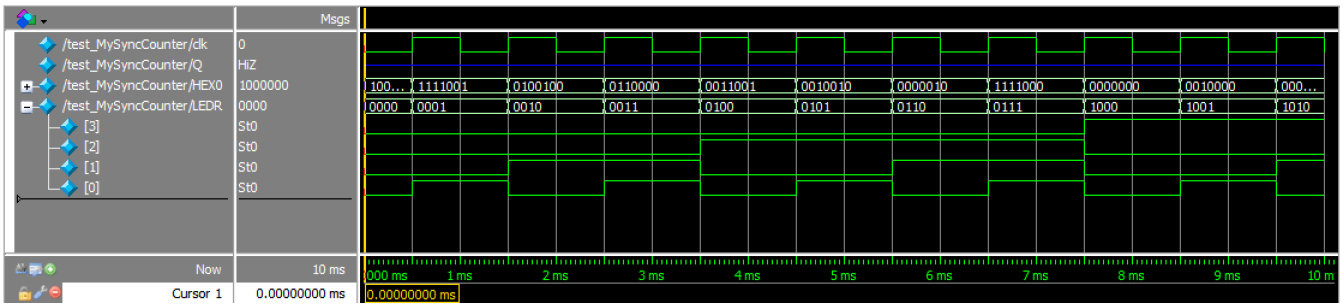

Figure 14: Synchronous Counter waveform.

Workshop 4
Exercise 1: Pipelined Arithmetic (Simulation)

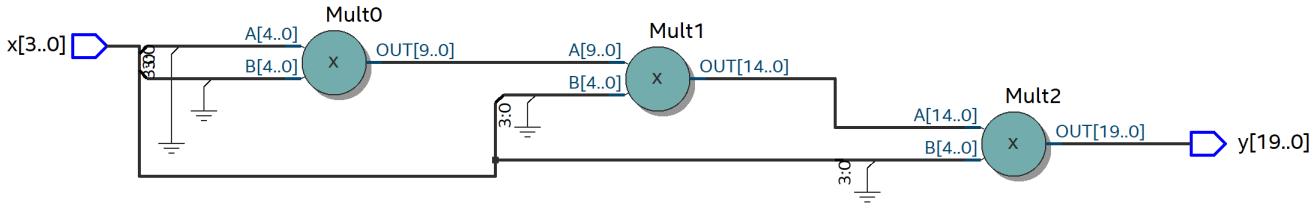

Figure 1: MyMult1 RTL viewer.

The x [3..0] to A [4..0] and B [4..0] in the first part of the RTL viewer is the first multiplied by 2.

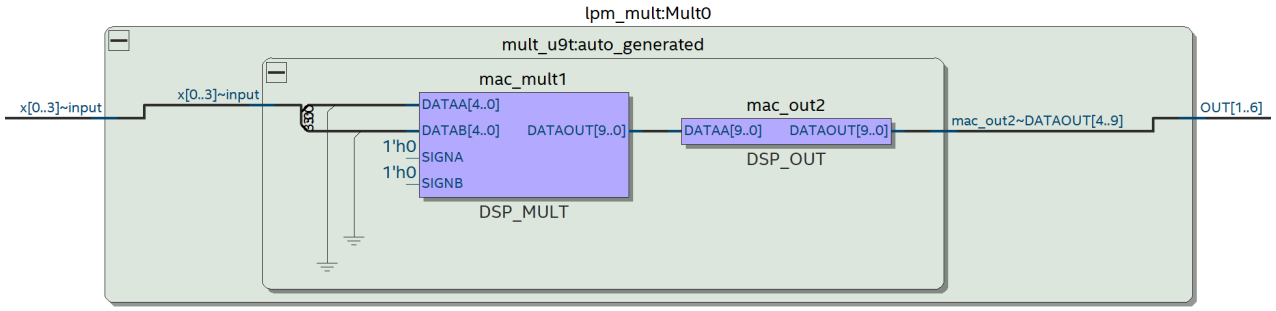

Figure 2: MyMult1 TMV viewer.

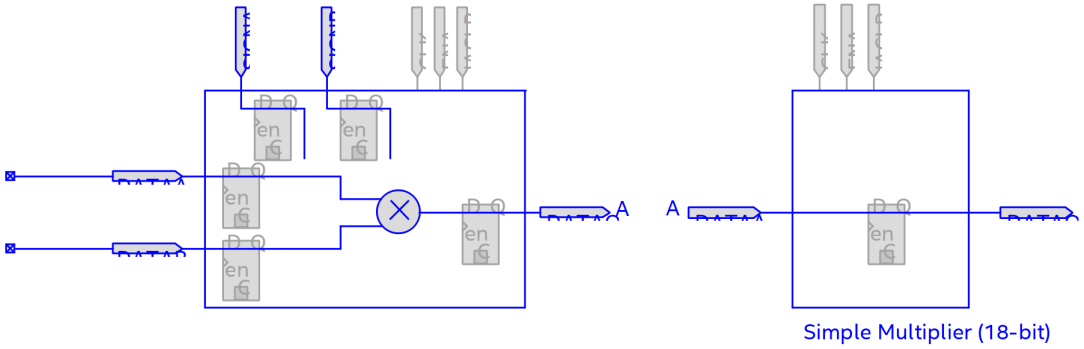

Figure 3: MyMult1 internal structure.

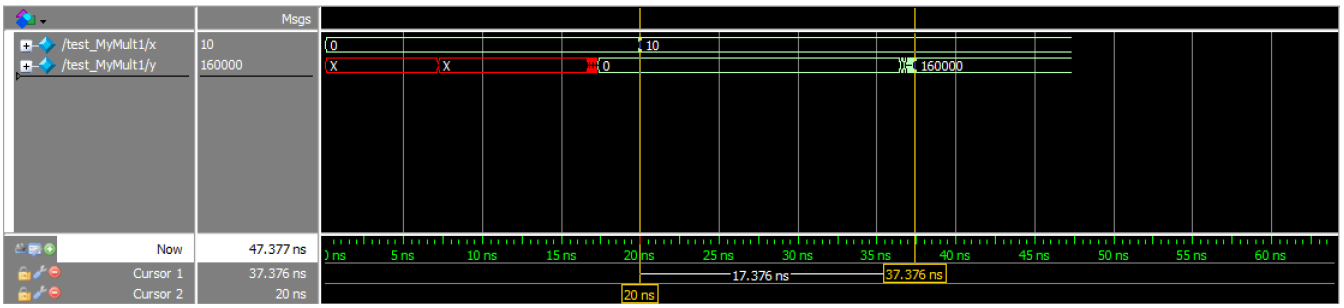

Figure 4: MyMult1 Gate Level Simulation.

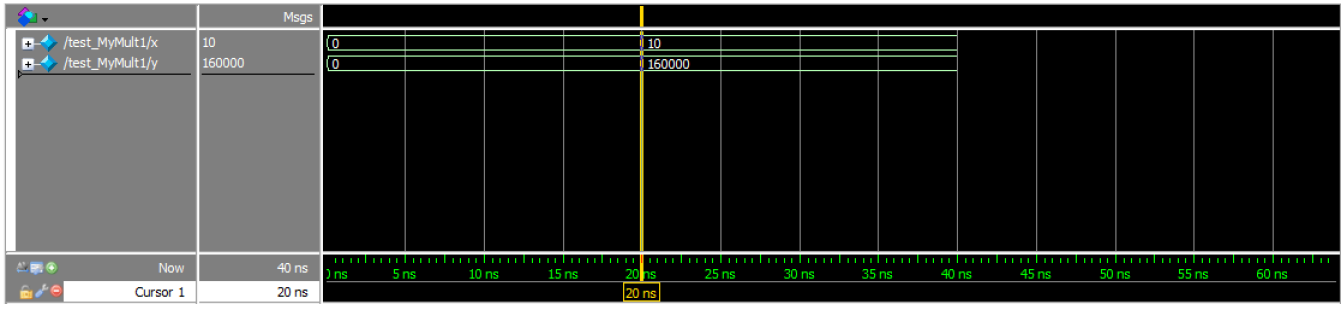

Figure 5: MyMult1 RTL Simulation.

Figure 5 is a zero-latency environment, and events usually occur on active clock edges. GLS can also be zero delay, but is more often used in unit delay or full timing mode. The event may be triggered by a clock, but it will propagate according to the delay on each element. A glitch appears at about 37ns. If we change the input, the glitch and propagation delay will not change. We can change the design as follows to reduce the propagation delay. Check Figure 6, Figure 7 and Gate Level Simulation in Figure 8 In RTL and TMV.

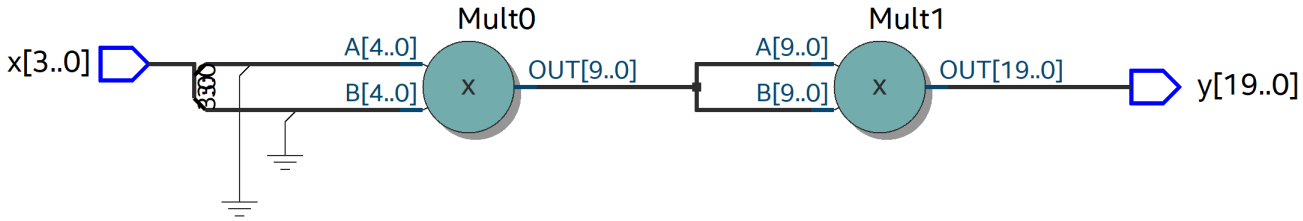

Figure 6: MyMult2 RTL viewer.

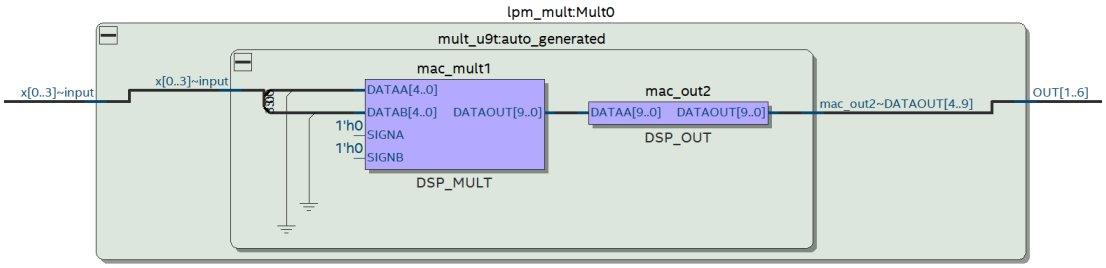

Figure 7: MyMult2 TMV viewer.

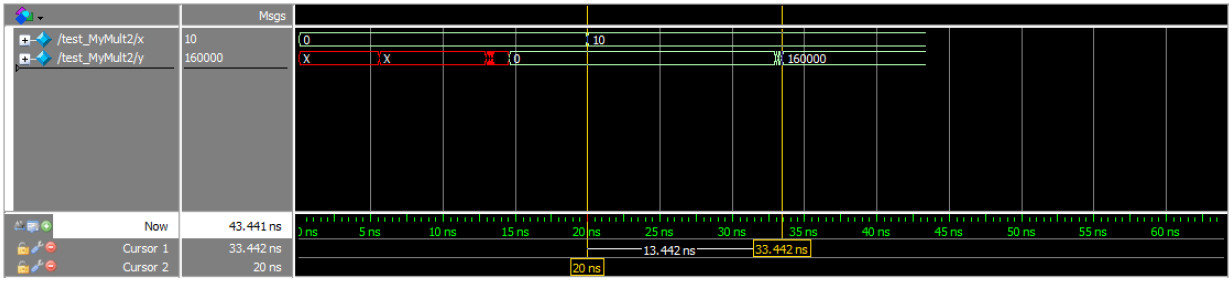

Figure 8: MyMult2 Gate Level Simulation.

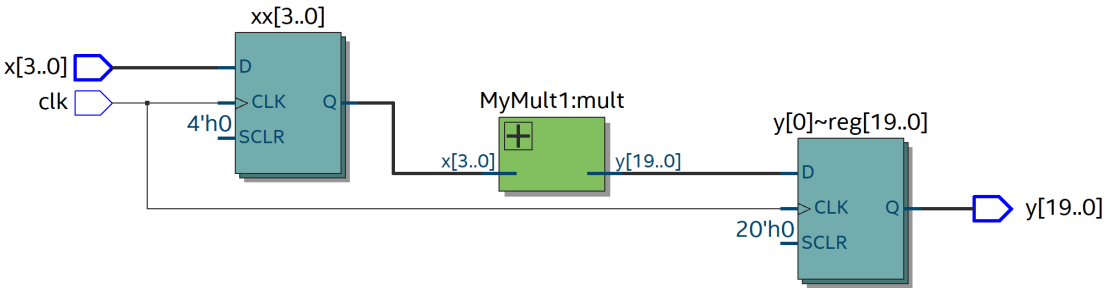

Figure 9: MyPipeline RTL viewer.

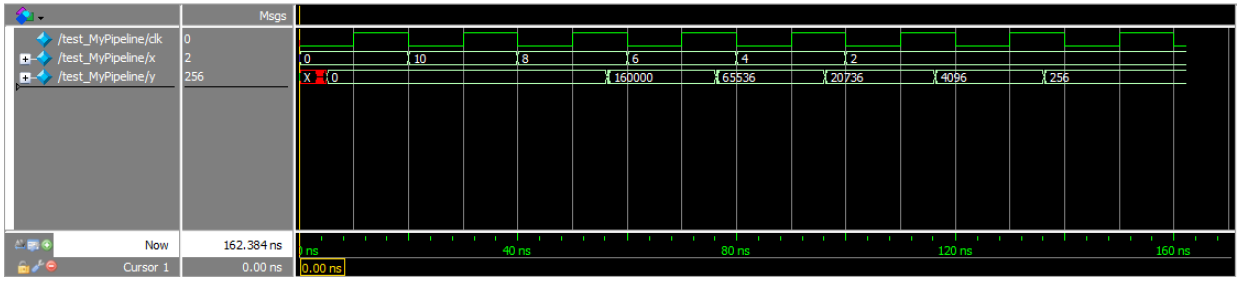

Figure 10: MyPipeline Gate Level simulation.

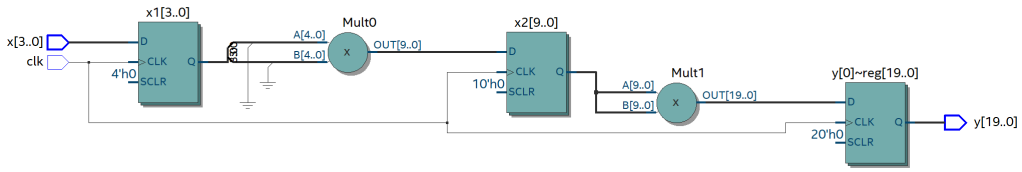

Figure 11: MyPipeline2 RTL viewer.

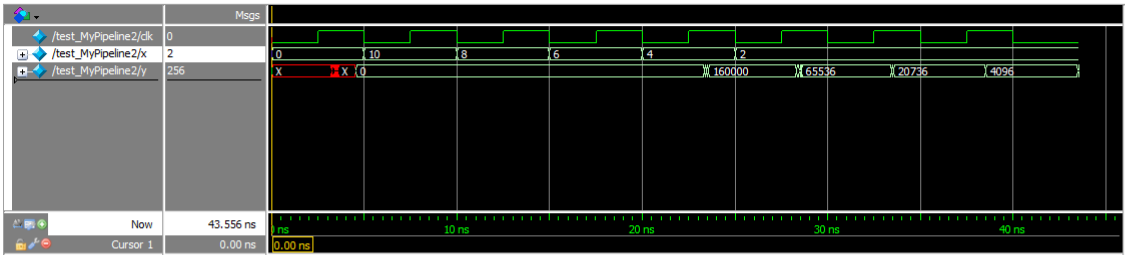

Figure 12: MyPipeline2 Gate Level simulation.

Exercise 2: Pipelined Arithmetic (Implementation)

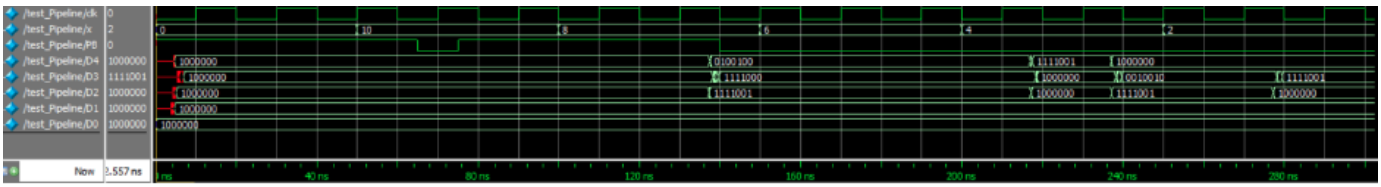

Figure 13: Pipeline Gate Level Simulation with out edge detector.

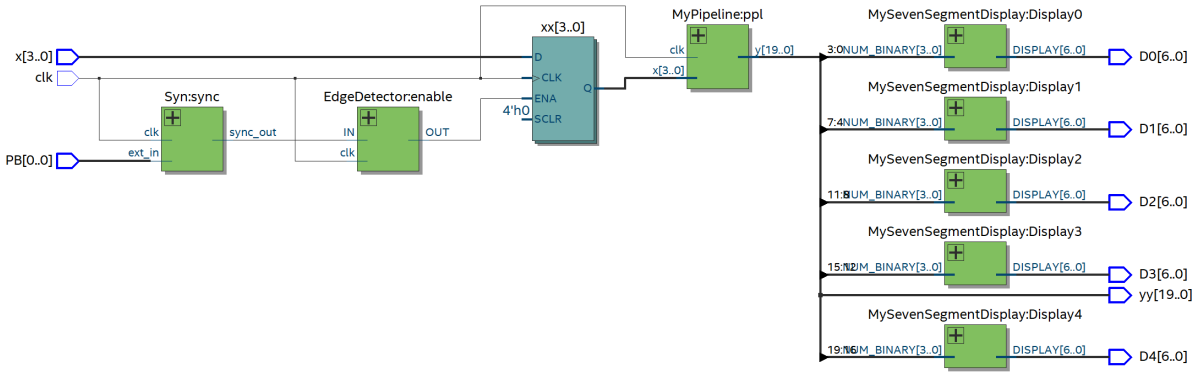

Figure 14: Pipeline RTL viewer.

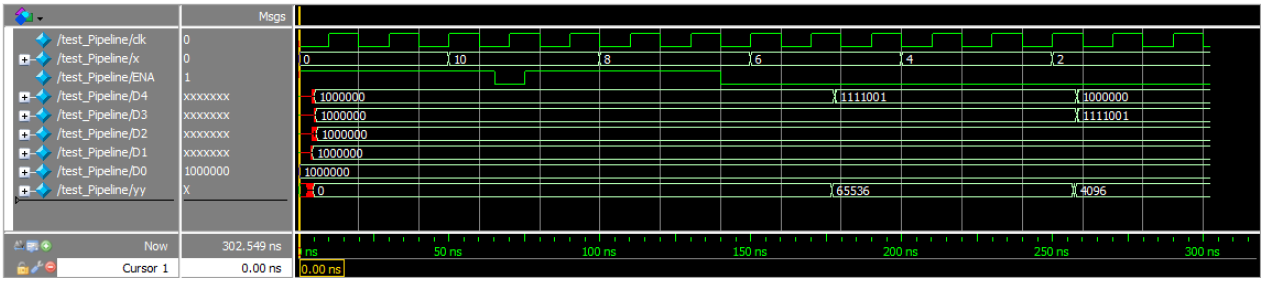

Figure 15: Pipeline Gate Level Simulation.

Workshop 5

Exercise 1: A Sequence Detector

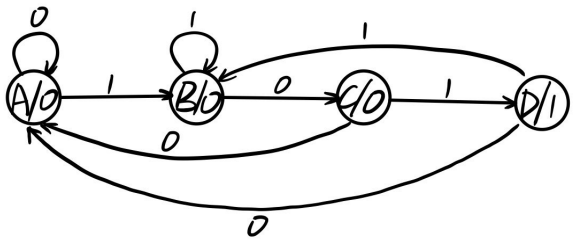

Figure 1: Moore FSM state diagram.

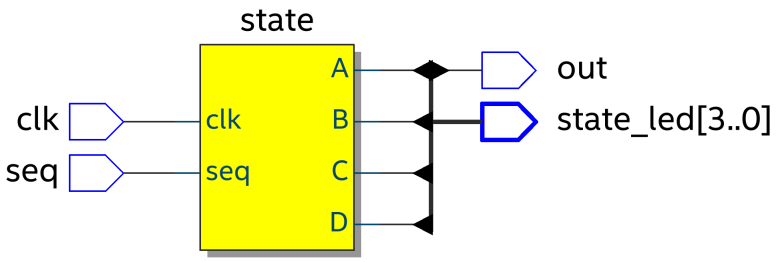

Figure 2: Moore FSM RTL viewer.

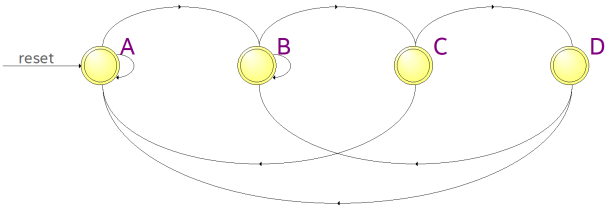

Figure 3: Moore FSM state machine viewer.

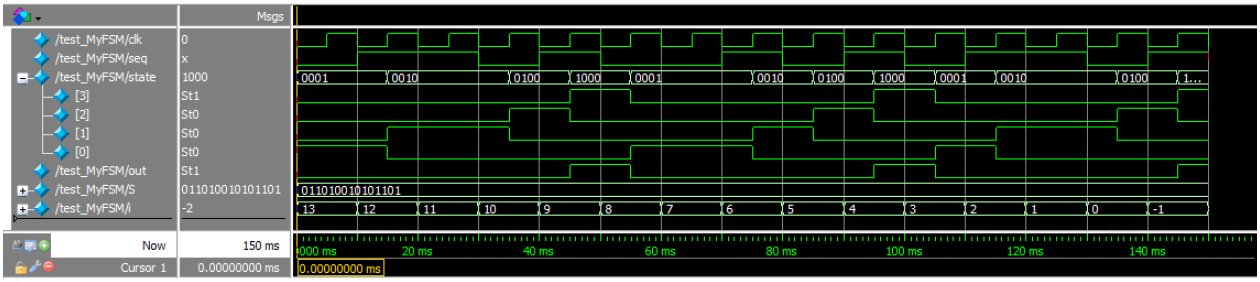

Figure 4: Moore FSM RTL simulation.

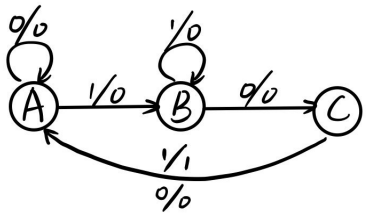

Figure 5: Mealy FSM state diagram.

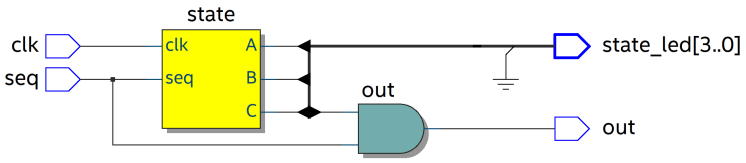

Figure 6: Mealy FSM RTL viewer.

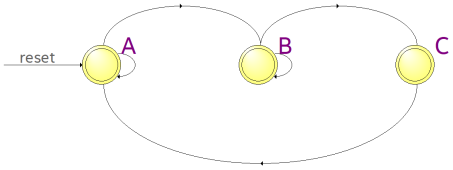

Figure 7: Mealy FSM state machine viewer.

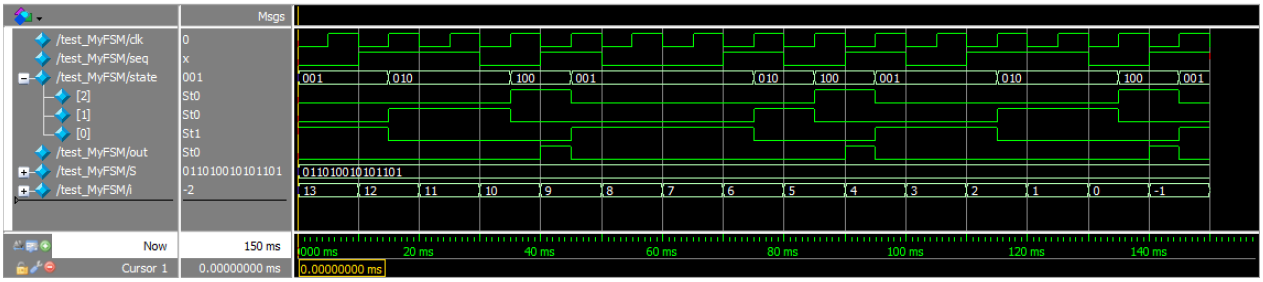

Figure 8: Mealy FSM RTL simulation.

Exercise 2: Basic Combination Lock

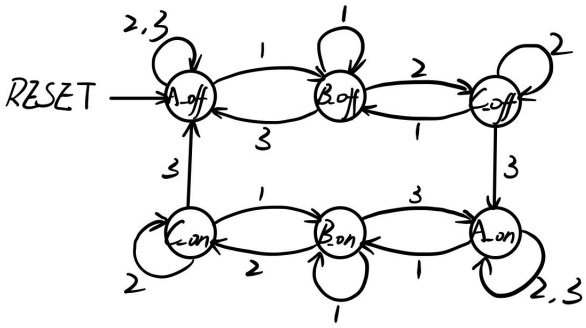

Figure 9: Combination lock state diagram.

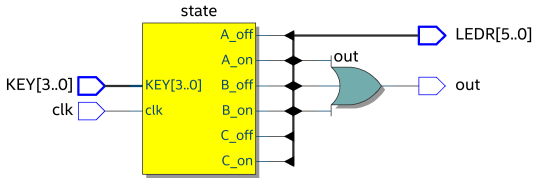

Figure 10: Combination lock FSM RTL viewer.

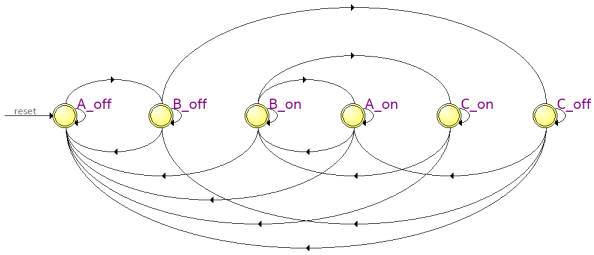

Figure 11: Combination lock FSM state machine viewer.

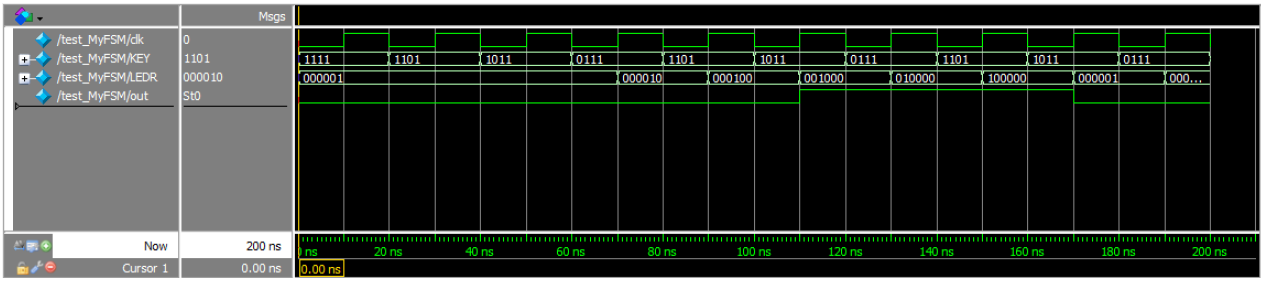

Figure 12: Combination lock FSM RTL simulation.

Exercise 3: A Press is a Press

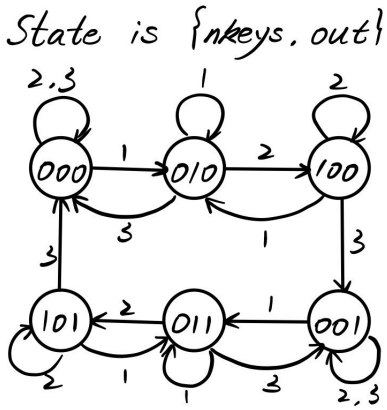

Figure 13: Combination lock revised state diagram.

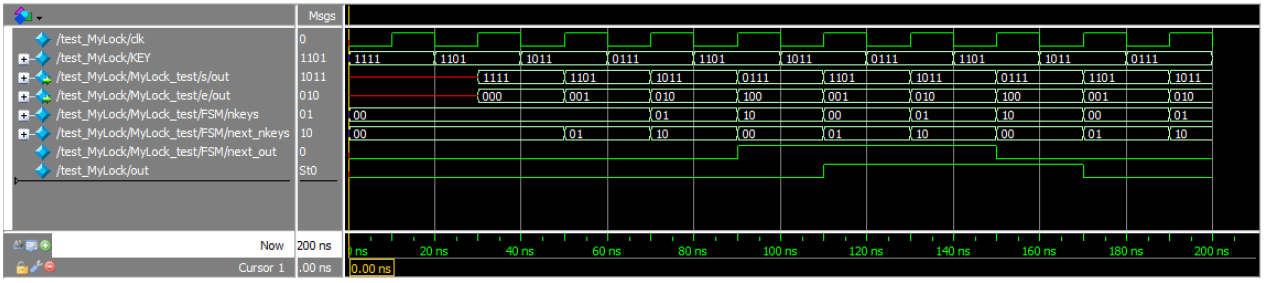

Figure 14: Combination lock revised RTL simulation.

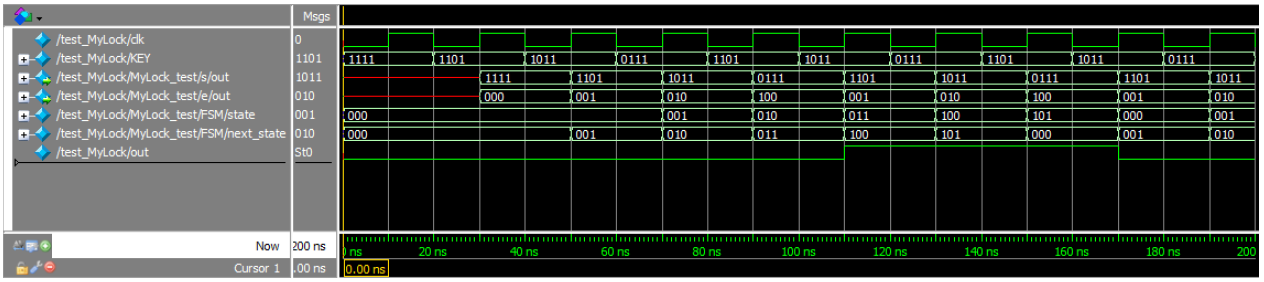

Figure 15: Combination lock original RTL simulation.

Workshop 6

Exercise 1: Debounce the Bounce

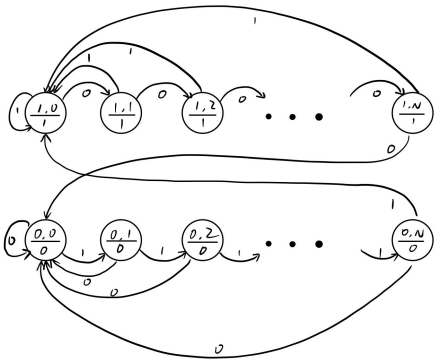

Figure 1: Debouncer state diagram.

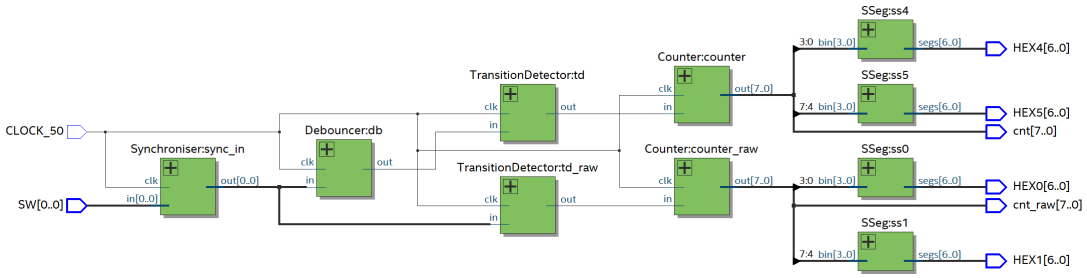

Figure 2: Debouncer RTL viewer.

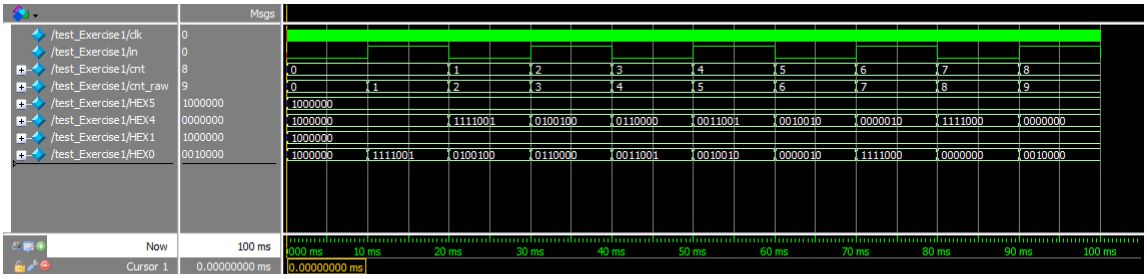

Figure 3: Debouncer RTL simulation.

Exercise 2: Rolling Fast and Slow

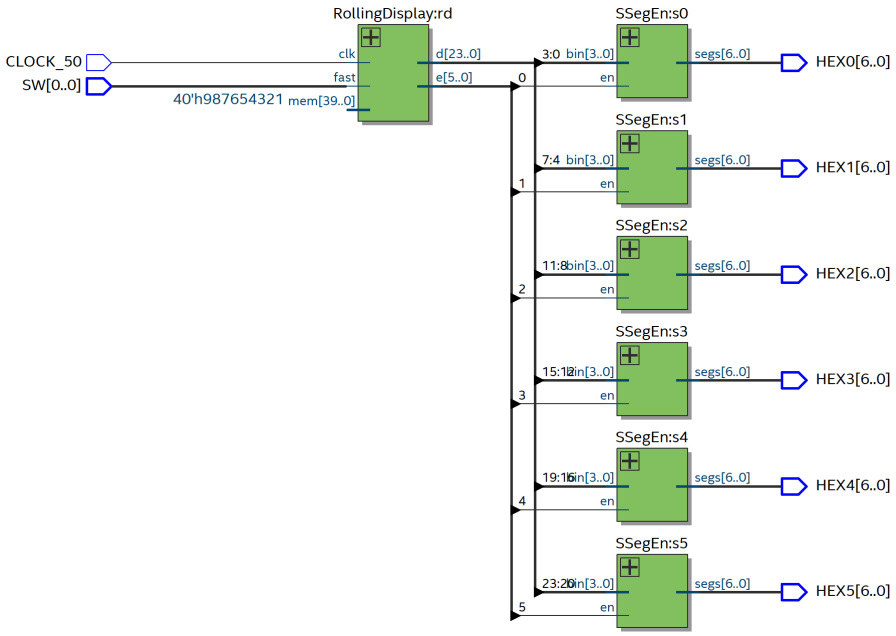

Figure 4: Rolling display RTL viewer.

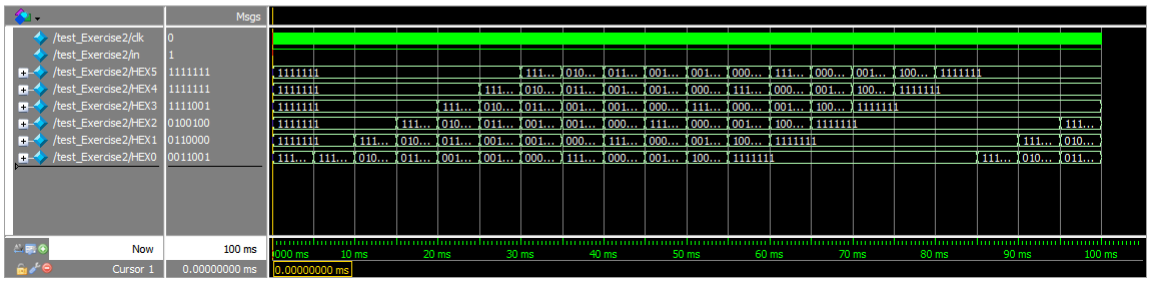

Figure 5: Rolling display fast mode RTL simulation.

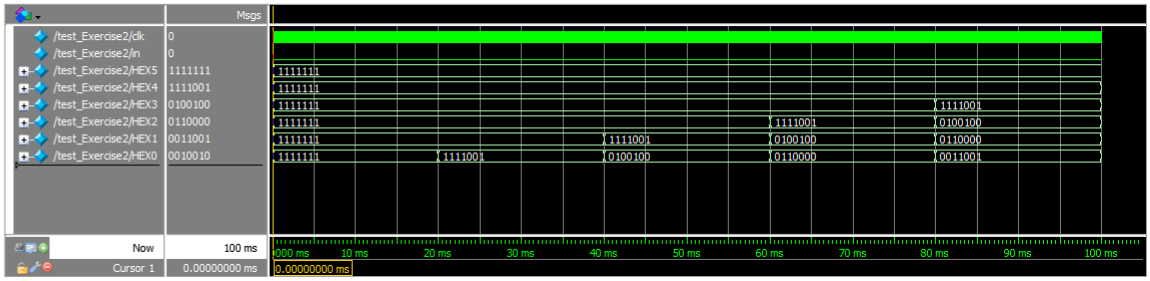

Figure 6: Rolling display slow mode RTL simulation.

Exercise 3: Editing a 10-digit Number

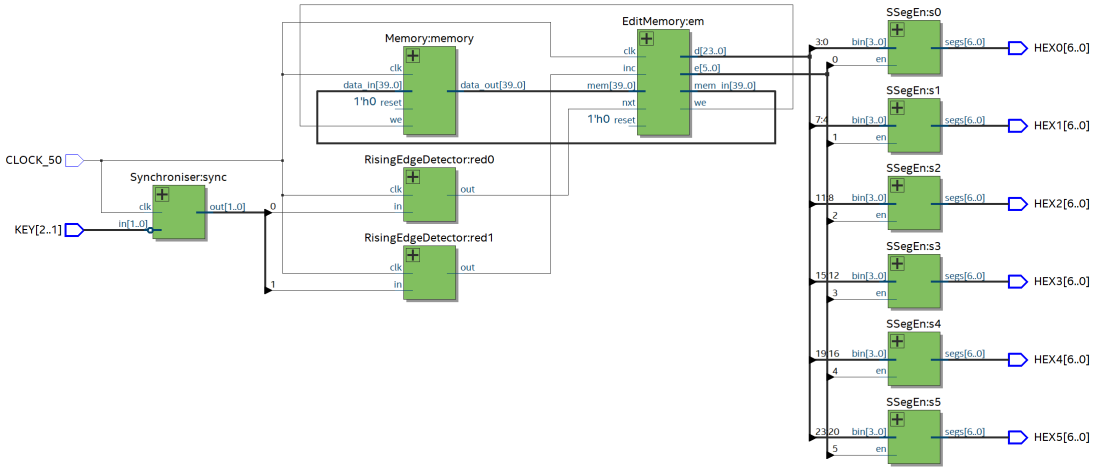

Figure 7: 10-digit number editing RTL viewer.

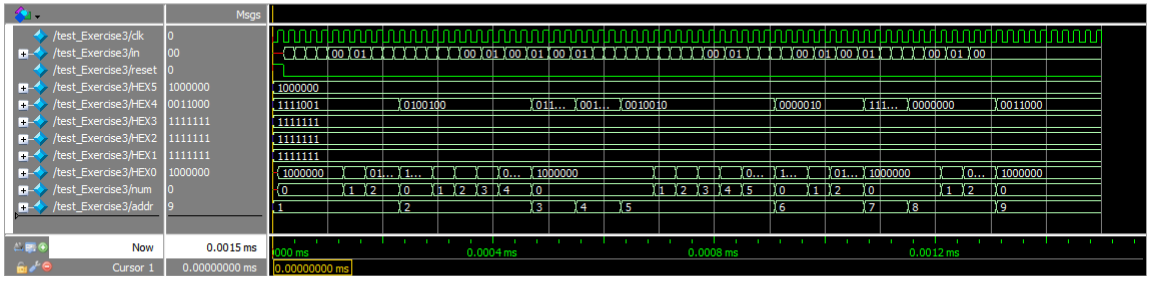

Figure 8: 10-digit number editing RTL simulation.

Exercise 4: Mode Selection

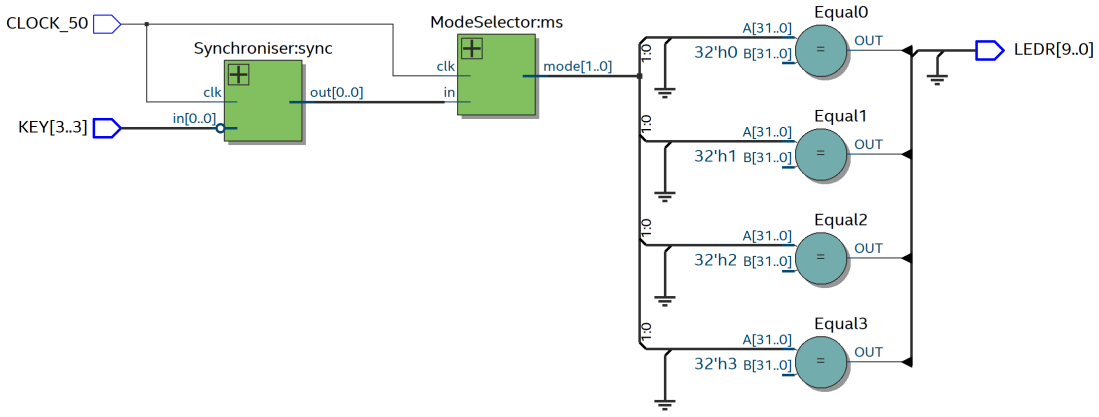

Figure 9: Mode Selection RTL viewer.

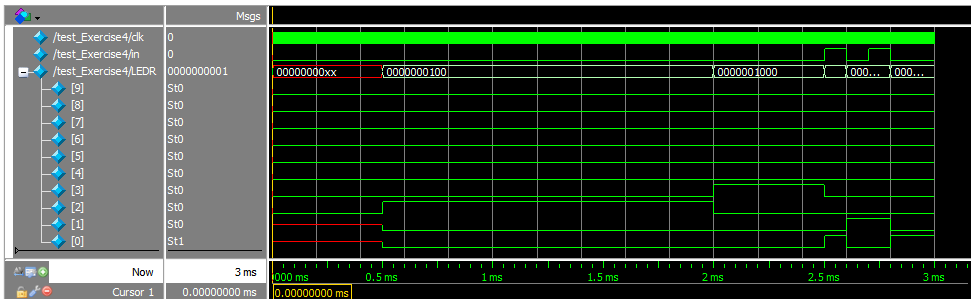

Figure 10: Mode Selection RTL simulation.

Exercise 5: Adding Reset to Exercise 3

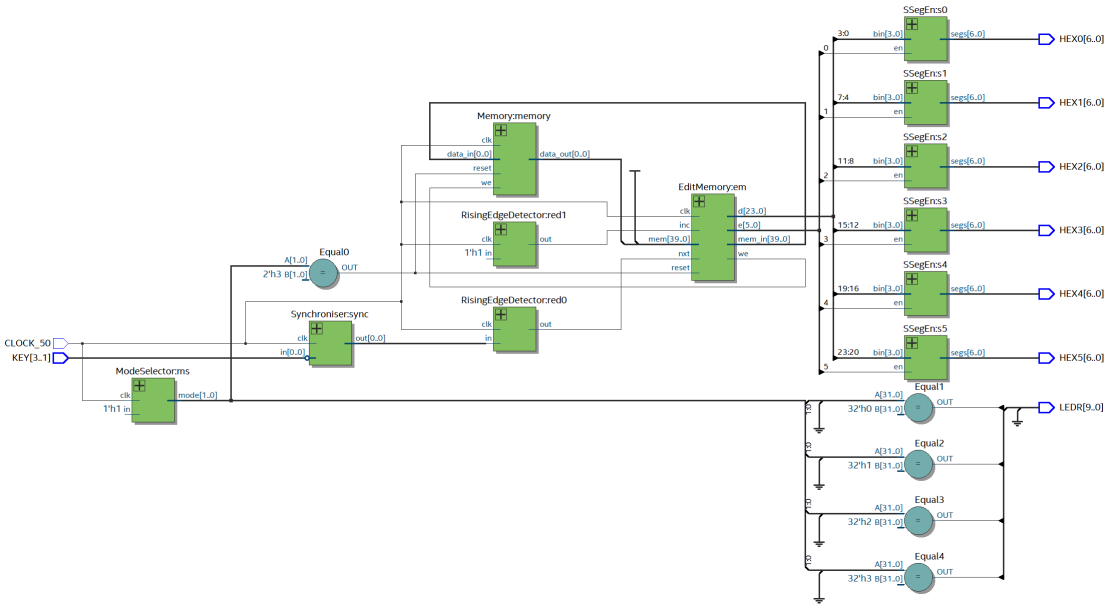

Figure 11: Exercise 5 RTL viewer.

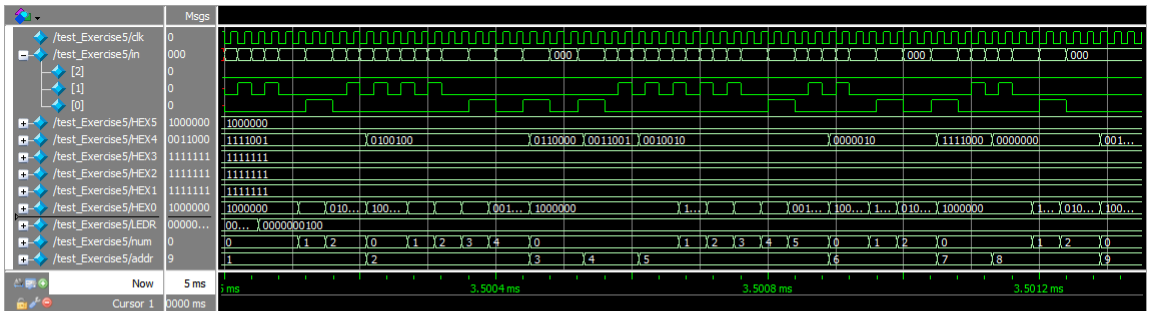

Figure 12: Exercise 5 RTL simulation.

Exercise 6: The Finished Product

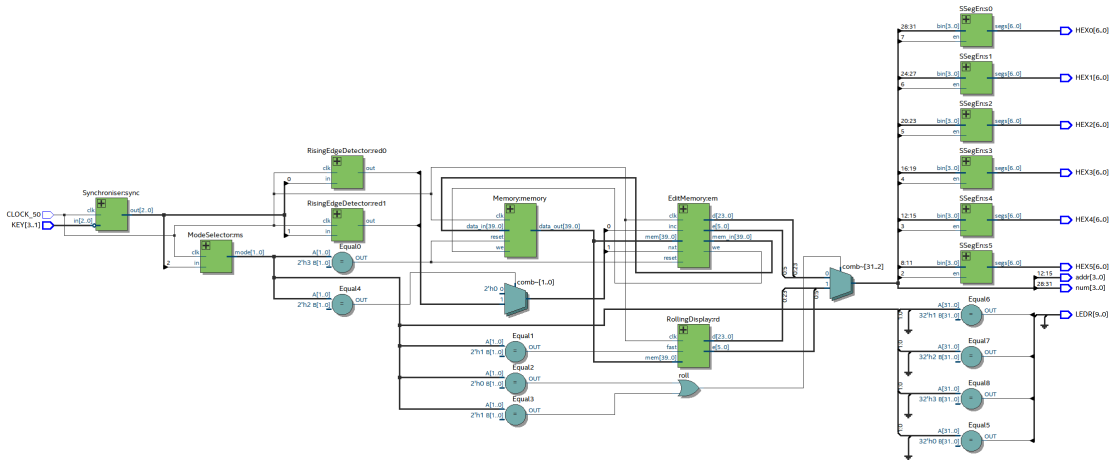

Figure 13: The Finished Product RTL viewer.

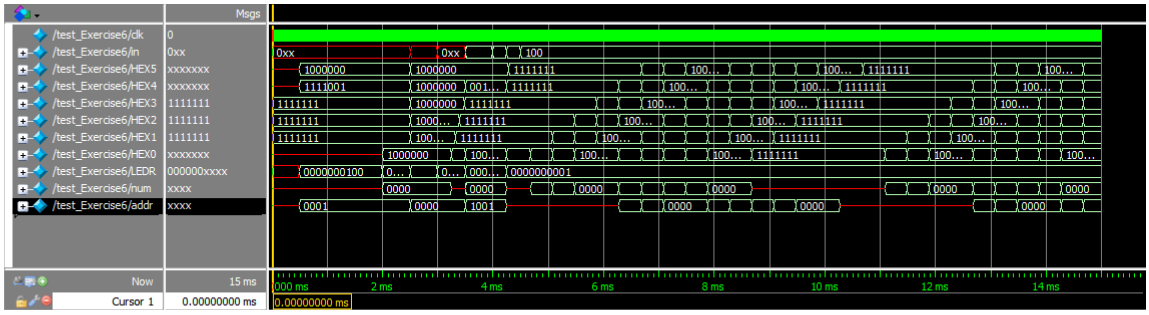

Figure 14: The Finished Product RTL simulation.