CMPE 160 Laboratory Exercise 8

Analysis and Simulation of Sequential Circuits

Andrei Tumbar
Performed: March 26th
Submitted: April 2th

Lab Section: 4
Instructor: Mr. Byers
TA: Sam Myers
Kobe Balin
Georgi Thomas

Lecture Section: 1
Professor: Mr. Cliver

By submitting this report, you attest that you neither have given nor have received any assistance (including writing, collecting data, plotting figures, tables or graphs, or using previous student reports as a reference), and you further acknowledge that giving or receiving such assistance will result in a failing grade for this course.

Your Signature: _____

Abstract

In the laboratory exercise a 4-bit shift register was created using four D flip-flops. The circuit had the ability to load a serial input into the shift register or to perform a parallel load of the four bits. After every clock, the bits where shifted through the flip-flops. A circuit was implemented and simulated using AOI logic and later converted to tri-state buffers for simplicity.

Design Methodology

The shift register must be able to set the values of Q_A, Q_B, Q_C, and Q_D synchronously (on the clock-edge) and also to shift the value of the four-bit number synchronously. This functionality is decided by the value of the `SL` or select pin. When this pin is `HIGH`, the values in the four bit register are set to A, B, C, and D on the next rising clock edge. Otherwise, when the select pin is `LOW`, the shift register will shift the four bit number and Q_A will take the value of the serial pin (`SIN`). The final functionality of this circuit is to reset the value of every flip-flop asynchronously. This can be achieved using the active-low asynchronous resets on every D flip-flop.

Table 1: Function table for 4-bit shift register.

`CLK`	`RST`	`SL`	$Q_A{}^*$	$Q_B{}^*$	$Q_C{}^*$	$Q_D{}^*$	
X	0	X	0	0	0	0	reset
↑	1	0	`SIN`	Q_A	Q_B	Q_C	shift
↑	1	1	A	B	C	D	load
not rising	1	X	Q_A	Q_B	Q_C	Q_D	hold

Table 1 shows all of the possible actions that can be performed on this circuit. The first of which is the asyncronous reset which does not rely on any particular value of the clock and will not be affected by the value of the select pin. The shift action will make the next value of each flip-flop (denoted by the *), to be the value of the previous bit. The first flip-flop will take on the value of the serial input. This action will only be performed when the clock edge is rising. The load action will also only occur on the rising clock edge and will take on the values of the parallel input pins, A, B, C, and D. Finally, the hold action will occur when the reset is not `LOW` and the clock edge is not rising. This will merely keep the current value of each D flip-flop.

The circuit was then implemented in Quartus using the D flip-flop component.

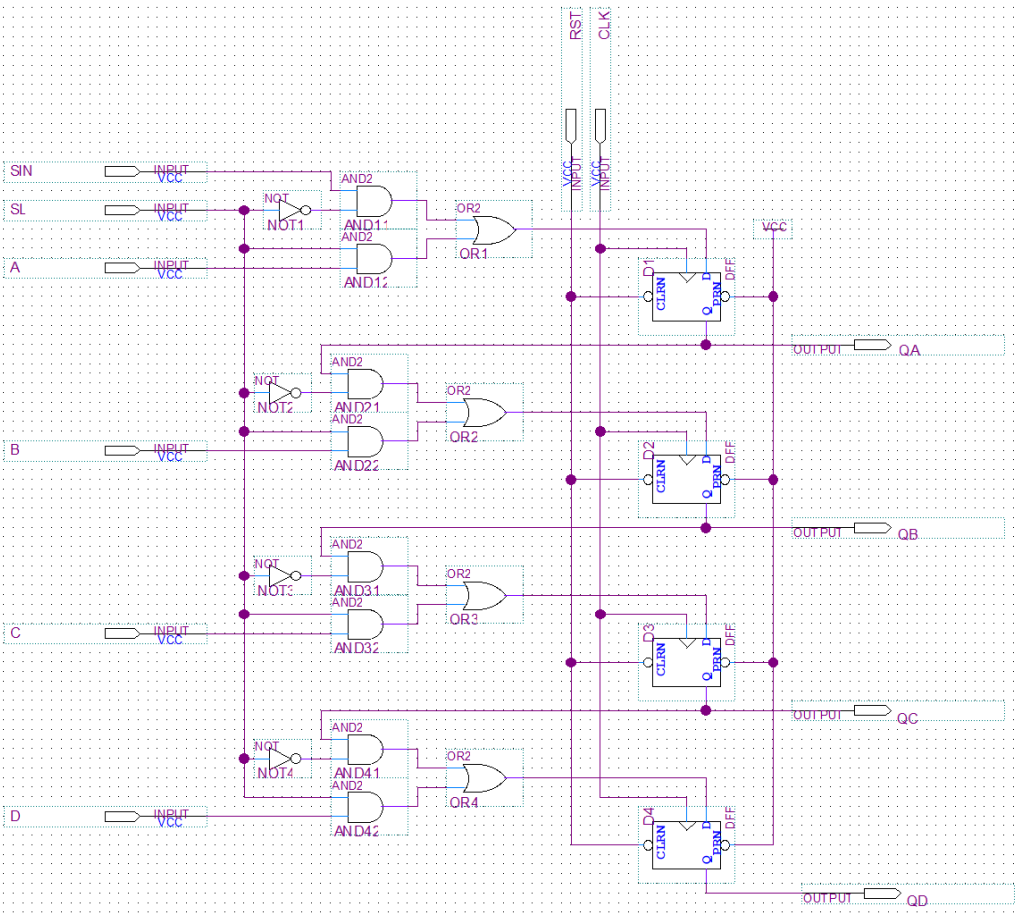

Figure 1: 4-bit shift register circuit schematic

Figure 1 is a circuit schematic of the 4-bit shift register described in Table 1. The AOI logic that feeds into the `D` value of each flip-flop will determine whether to shift the register or to load it.

The AOI logic can be further simplified using a new component call a tri-state buffer. The tri-state buffer is an useful tool in circuit design which allows a component of a circuit to disconnected electrically from the rest of a circuit. This use will become clear later.

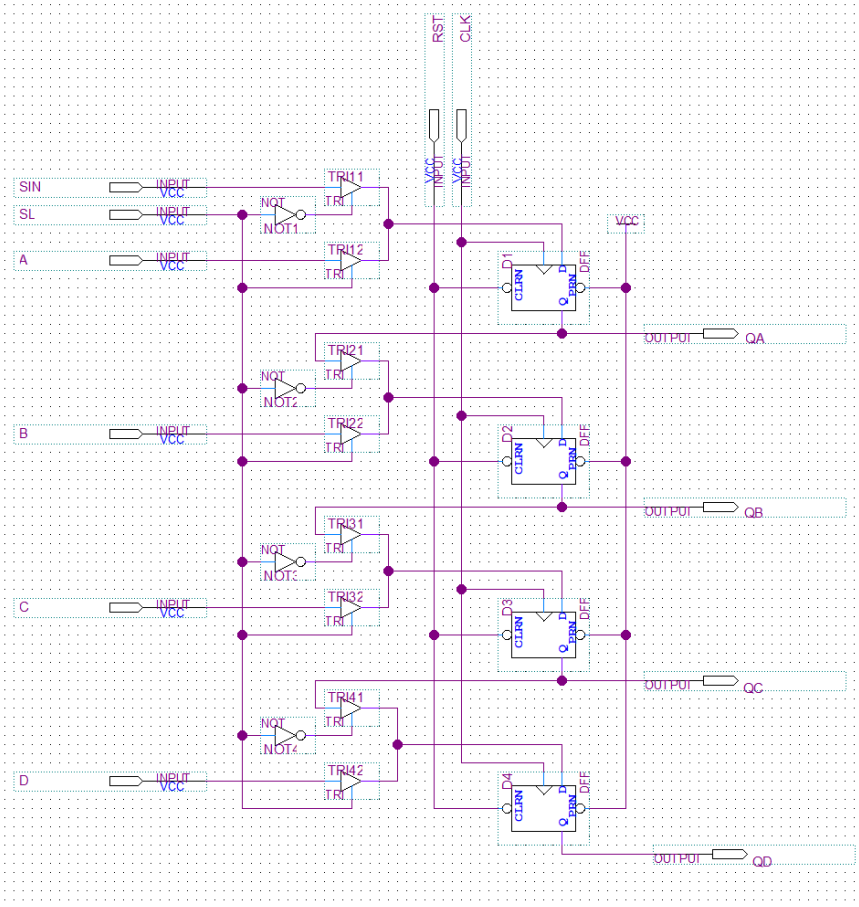

Figure 2: 4-bit shift register with tri-state buffers

Figure 2 shows the circuit schematic with the AOI logic feeding into the `D` inputs for each flip-flop changed to tri-state buffers. The tri-state buffer will take two inputs, a data input `A` and a control input which we will call the enable (`EN`). When the enable is `HIGH`, the output of the buffer will be logically equivalent to the input value, `A`. When the enable is `LOW`, the tri-state will yield a special output commonly refered to as ***Hi-Z***. In this state, the output is neither high or low but rather disconnected from the rest of the circuit. For example, when looking at the tri buffers going to the first D flip-flop in Figure 2, when the select is `LOW`, only the top buffer is enabled meaning that `D` is only wired to the `SIN` pin and the `A` pin is effectively disconnected. This is another way to implement a 2-1 mux with the select pin as an input and `SIN` and `A` being select lines.

Results and Analysis

The two Quartus models were simulated in Modelsim to produce a waveform with all of the relative input and output signals. The input signals were defined using the following criteria:

- Reset before the first clock cycle.
- Clock Cycle 1: parallel load of the number B_{16} (hex) to the register
- Clock Cycles 2-3: shift right two times while serial input is equal to 0.
- Clock Cycles 4-5: shift right two times while serial input is equal to 1.

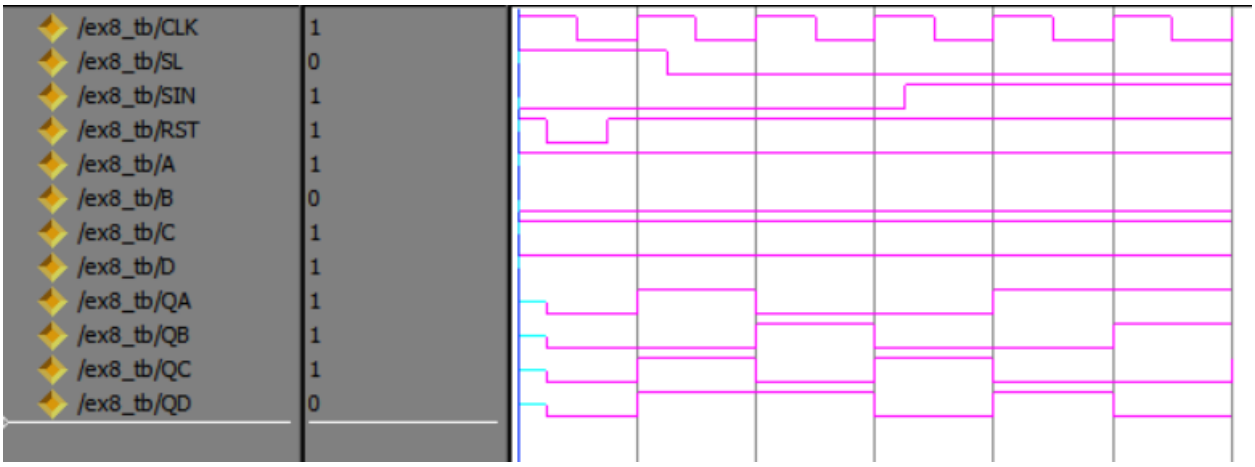

Figure 3: Shift register simulation wave capture.

Figure 3 shows the results of the a modelsim simulation run with the 4-bit shift register circuit. Both the diagrams from Figures 1 and 2 yielded the same waveforms because they are just different implementations of the same circuit.

When looking at the timing of a circuit that includes a clock, one must examine the minimum clock period or the maximum clock frequency. This is an important classification to ensure the integrity of the outputs. There are times before and after the clock edge where a change in the D inputs should not change. These periods are called setup and clock to output times denoted by t_s and t_{co} respectively. The final time period that causes a delay is the propagation delay from the circuit logic denoted by t_pd. The following equation could be extracted from quartus after designing the circuit.

$$T_{min} = t_{co} + t_{pd} + t_s = 5.809\,ns + 0\,ns + 2.201\,ns = 8.1\,ns \tag{1}$$

$$f_{max} = \frac{1}{T_{min}} = \frac{1}{8.1 \cdot 10^{-6}} = 123.4\,Mhz \tag{2}$$

The minimum clock period and the maximum frequency were calculated to be $8.1\,ns$ and $123.4\,Mhz$ respectively. If these limits are exceeded the output of the circuit is undefined

because a insufficient amount of time is allowed before and after the clock edge to provide a meaningful output.

Conclusion

In this laboratory exercise, a 4-bit shift register with parallel load and asynchronous reset was implemented in quartus and simulated in Modelsim. The lab was looking to provide more insight on the functionality of a serial loader and to also expose the idea of the tri-state buffer. This exercise was successful in doing so as it provided a comprehensive shift register example.

Questions

1. A right bit shift operation with a zero introduced for the most significant bit is the same as performing an integer division by two.

2. A left bit shift with a zero introduced in the least significant bit is the same as multiplying by two.

3. If the input signal of a flip-flop is changed on the rising clock edge, the output is undefined because there is no way of knowing if the flip-flop looked at the value before or after the change.