Digital Design Lab
EEN 315 Section 3G

Project 1
Shift Register and Signature Analyzer

Group 5
Nathan Paternoster
(Partners: Michael Finale, Daniel Bennett)

Sayan Maity, TA

University of Miami

November 5th, 2013

Abstract

A *register* is a component that holds memory. The most basic type of register is a flip-flop. A *shift register* is a unit of memory that "shifts" data in a binary stream to its output at every clock cycle. A shift register is made out of a cascade of flip-flops sharing the same clock. An N-bit shift register will contain N amount of flip-flops and be able to store N amount of bits in its memory. Shift registers are commonly used for serial-to-parallel and parallel-to-serial data conversion, arithmetic circuits, and as counters.

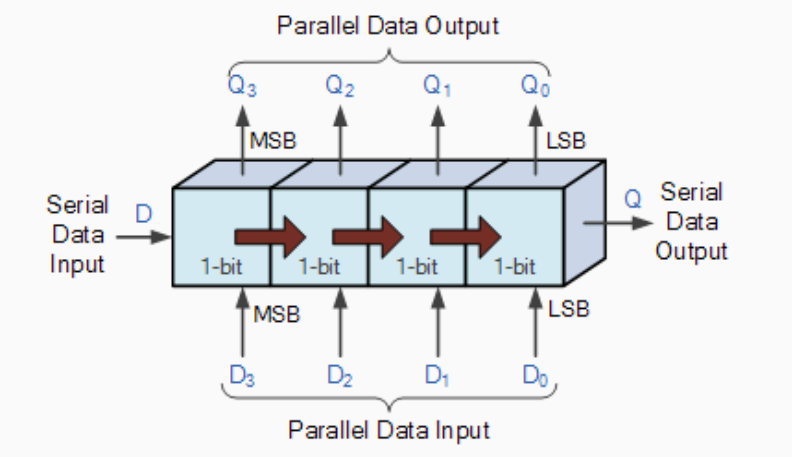
A *signature* is a concise representation of a large amount of data. In computing, signatures are used to verify that a correct sequence of bits is flowing through a certain point in a circuit. This is to ensure that the circuit is working correctly and there won't be false data at the output. An analogy is to think of receiving a monetary check from someone. That check contains the signature of the person who is paying you. It is important to verify that the money being paid to you is coming from the right bank account. That person's signature can be checked against a previous signing to complete the verification.

A signature can be generated using a *pseudo-random binary sequence (PRBS) generator*, which is itself constructed from a shift register. A *signature analyzer* will use the PRBS generator to compute the data's signature. In this project we will learn how to design a shift register in order to implement a signature analyzer. We will use Altera Quartus to simulate this circuit

Table of Contents

Overview

A shift register is constructed from a cascade of flip-flops. In this project we will use 4 JK flip-flops for a 4-bit register. Shift registers may input and output data either serially or in parallel. Data input serially is inputted one bit at a time from the right or left. Data input in parallel is data inputted in all the bits at once. In this project we will only implement serial/parallel input and parallel output.



The shift register will contain a **status line** to select between serial and parallel modes and a **direction line** to choose between left and right shifting (when in serial mode). The direction line will be implemented by adding a set of four 2-to-1 MUXs before the flip-flops. The status line will be implemented in the same way with a second set of four 2-to-1 MUXs. The single select line for each set will be the user's direction and status inputs respectively. There will be a **load input** that when active will enable parallel input loading. This will override the direction line.

To design the signature analyzer we will first build the shift register into a *pseudo-random binary sequence (PRBS) generator*. A shift register can be made into a *ring counter* by feeding back its serial output to its serial input. Once this ring counter is initialized it will generate a sequence of N states, N being the size of the shift register in bits. We can increase the length of this sequence generator to its maximum length of 2N-1 states by performing modulo-2 addition (XOR) on the two least significant bits and then feeding this back into the serial input.

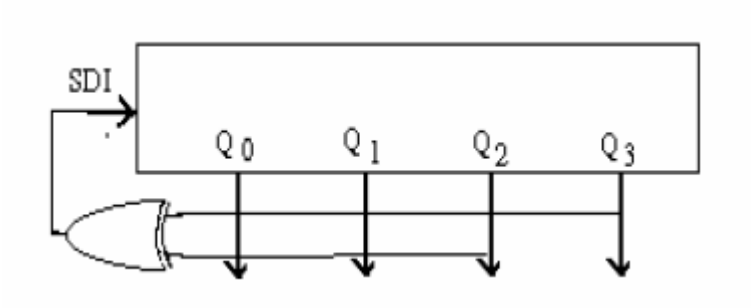


Figure 2: A maximum length ring counter

Finally, this maximum length ring counter can be made into a PRBS generator by introducing a **PROBE input** into the modulo-2 addition. The PROBE input will allow us to start up the shift register into a non-zero value. It will essentially replace the serial input.

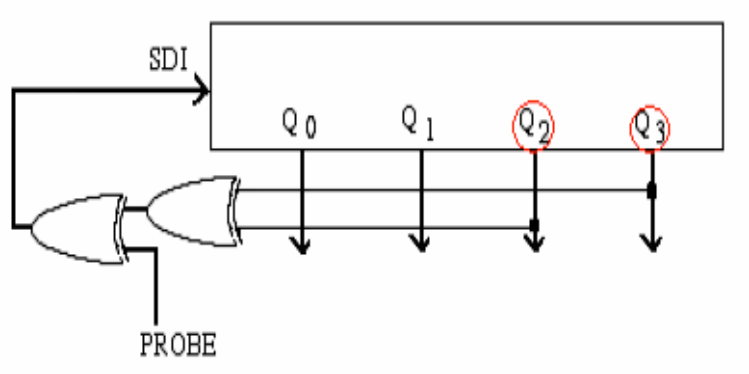


Figure 3: A PRBS generator.

We will use the probe to input a binary stream. The parallel output of the signature analyzer after the entire binary stream data has been inputted is the data's unique signature. The PRBS generator will be accurate in nearly every case. There is only a 2N-1 possibility of an error passing undetected.

Objectives

To understand the behavior of shift registers. To reinforce knowledge of sequential design and JK flip-flop design. To introduce the concepts of testing, signature generation, and pseudo-random binary sequence (PRBS) generation.

Equipment

Description	Chip Number	Quantity
Quad 2-Input NAND Gates	7400	2
Quad 2-line to 1-line Multiplexers	74157	2
Dual positive-edge triggered J-K flip-flops	7476	2

- *Note: Since we are simulating this circuit, these components are for design purposes only.*

Description

The focus of this lab is to correctly design each stage of the circuit so that all the specifications are met. The first step of the shift register design is to perform flip-flop conversion from JK to D flip-flops since a shift register will use D flip-flops. This will be accomplished by comparing the excitation tables of both the JK and D flip-flops.

The second step is to provide for bi-directional shifting capability. This can be achieved by adding a set of four 2-to-1 multiplexers before the four D flip-flops. The single select line that all four will share will be the user's DIR control. When DIR is low the register will perform a right shift. The flip-flops will accept the

output of the flip-flop representing the bit located to its left as its input. Since D_0 represents the LSB and D_3 the MSB, D_0 will accept Q_1 as its input, D_1 will accept Q_2 as its input, D_2 will accept Q_3 as its input, and D_3 will accept the user's serial (external) input as its input. When DIR is high the register will perform a left shift. The flip-flops will accept the output of the flip-flop representing the bit located to its right as its input. In this case D_0 will accept the user's serial (external) input as its input, D_1 will accept Q_0 as its input, D_2 will accept Q_1 as its input, and D_3 will accept Q_2 as its input.

The third step is to allow the shift register to accept parallel inputs in addition to the serial input. This will be done with another set of four 2-to-1 multiplexers placed before the D flip-flops but after the directional multiplexers. This parallel/serial control must override the directional control, which is why it is placed after the directional components. These multiplexers will be controlled by a single select line, the $\overline{LOAD}$ control. When $\overline{LOAD}$ is high the flip-flops will receive the serial inputs as their inputs. However, when it is low, the flip-flops will receive four external parallel inputs called P_0, P_1, P_2, and P_3.

The signature analyzer will require a quick modification to the shift register. Modulo-2 addition will be performed on the two least significant output bits Q_2 and Q_3 and that signal will be fed back into the serial input in order to create a PRBS generator. In addition, a PROBE input will be added in this feedback path. The modulo-2 addition and PROBE addition will both be accomplished with XOR gates. Q_2 and Q_3 will be put into a XOR gate. The output of that gate along with the PROBE input will be put into a second XOR gate. The output of that gate will finally be attached to the serial input.

Specifications

Must design a four-bit bi-directional serial/parallel-in and parallel out shift register with a direction control that when *low* will indicate a right shift and when *high* will indicate a left shift and a load control that when *low* will enable parallel loading. Must use JK flip-flops.

Must design a signature analyzer that utilizes the shift register already built. Must perform modulo-2 addition on the two least significant bits. Must test the signature analyzer with the following four sequences: 1010101010, 1100110011, 1111000011, and 1111111100.

Design Synthesis

Step 1: JK to D flip-flop conversion

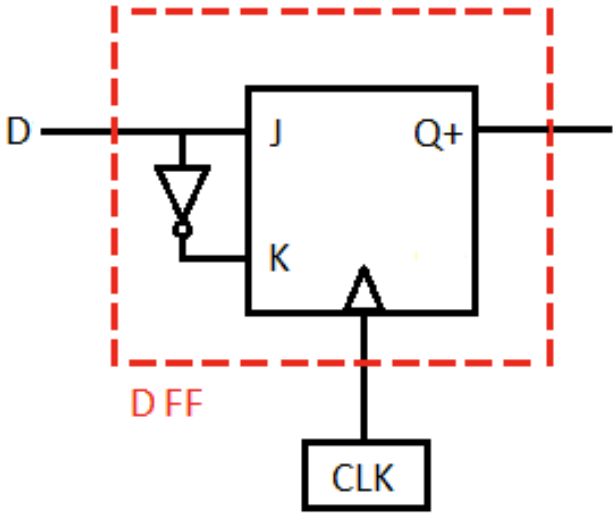
D	Q	Q+	J	K
0	0	0	0	x
0	1	0	x	1
1	0	1	1	x
1	1	1	x	0

D \ Q	0	1
0	0	x
1	1	x

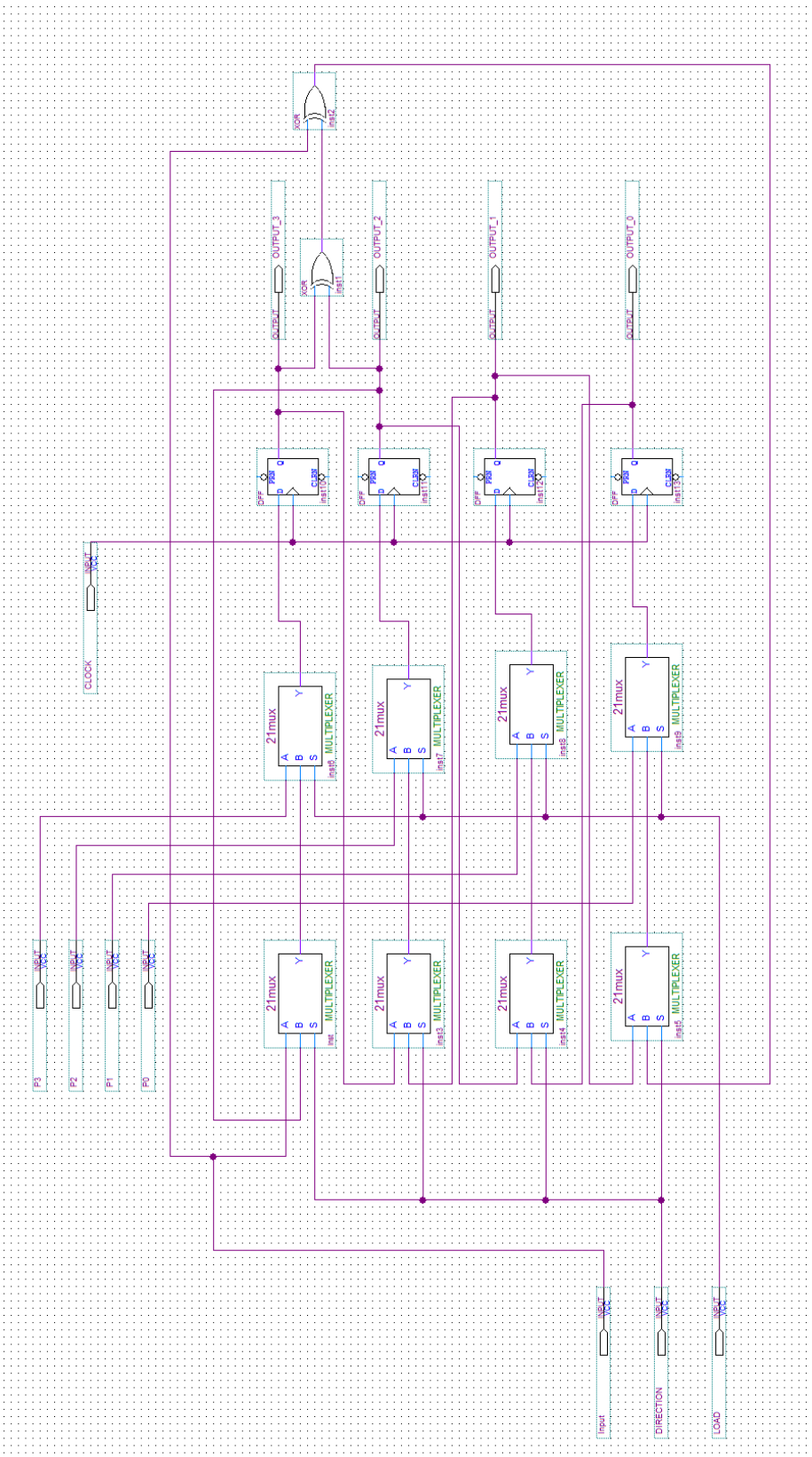
J = D

D \ Q	0	1
0	x	1
1	x	0

K = $\overline{D}$



Complete Logic Diagram



Results and Simulations

Shift Register

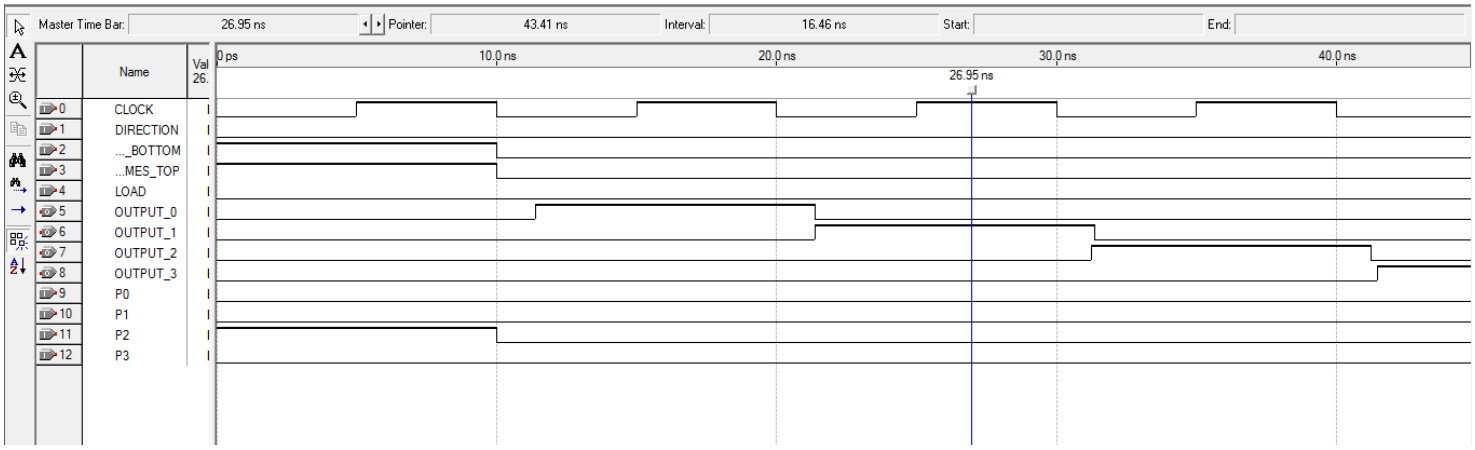


Figure 1: Demonstrating a right shift. DIR is low and the initial input is shifted from the MSB (OUTPUT_0) rightward to the LSB (OUTPUT_3) at every clock pulse.

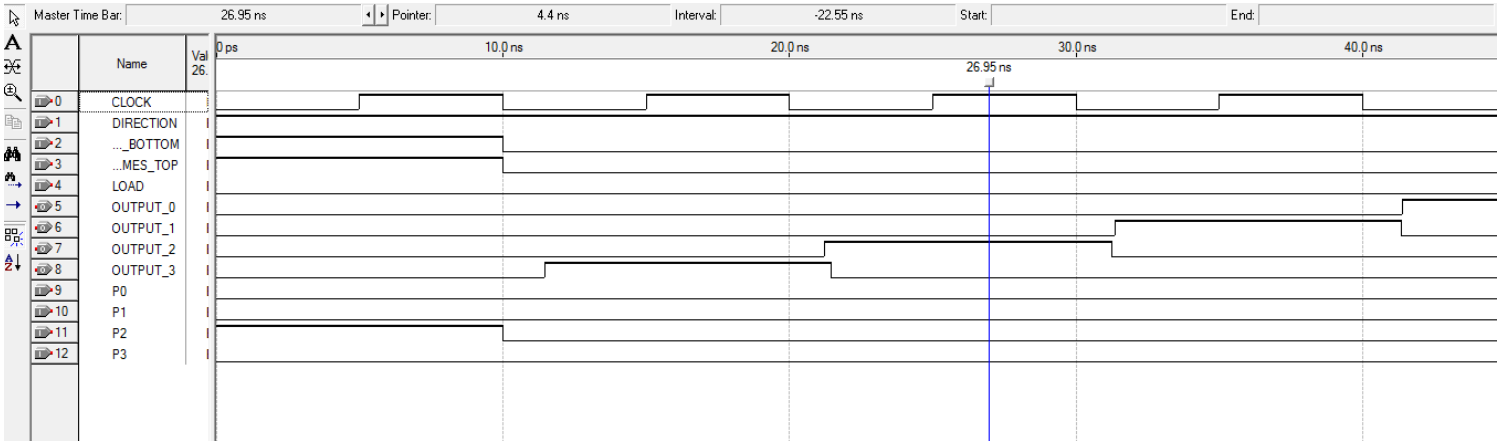


Figure 2: Demonstrating a left shift. DIR is high and the initial input is shifted from the LSB (OUTPUT_3) leftward to the MSB (OUTPUT_0).

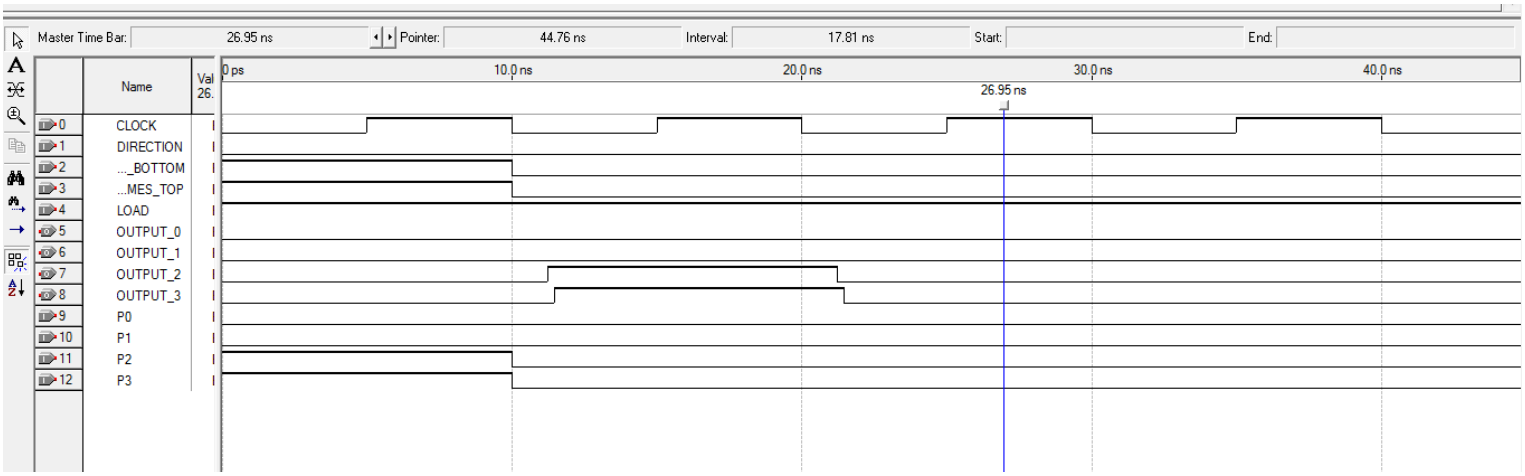


Figure 3: Demonstrating a parallel input shift. LOAD is high so the DIR control is overwritten. The parallel inputs 0011 (P0, P1, P2, P3) are shifted to the output (OUTPUT_0 through OUTPUT_3).

Signature Analyzer

Sequence 1: 1010101010

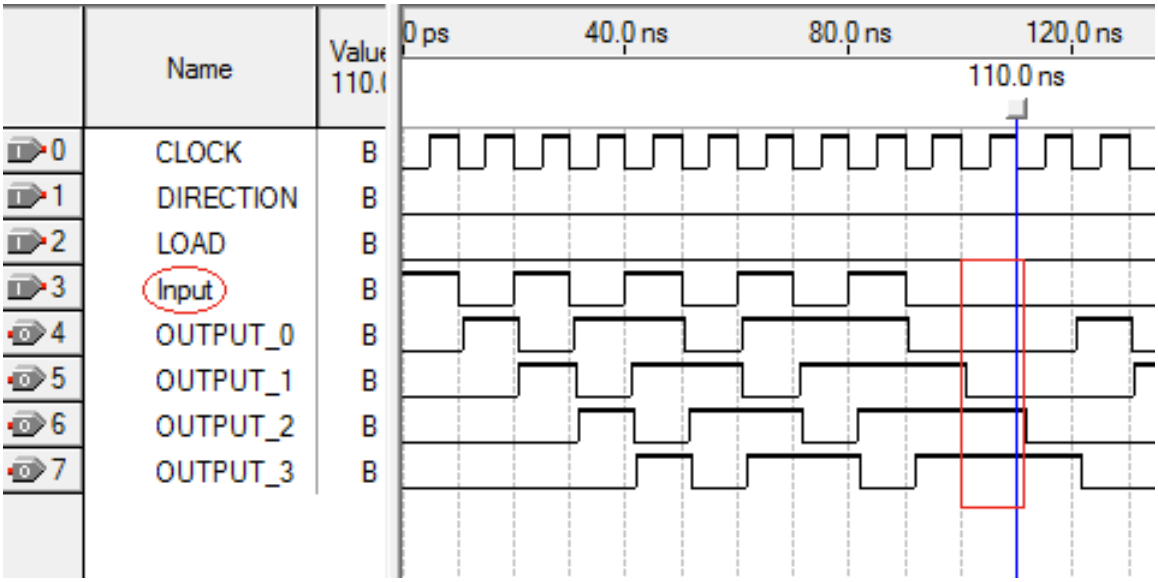


Figure 4: The input sequence is entered in, one bit at each clock pulse. The resulting signature is the output after the last bit of the sequence has been put in. The signature for this sequence is 0011 as represented by the output boxed in red.

Sequence 2: 1100110011

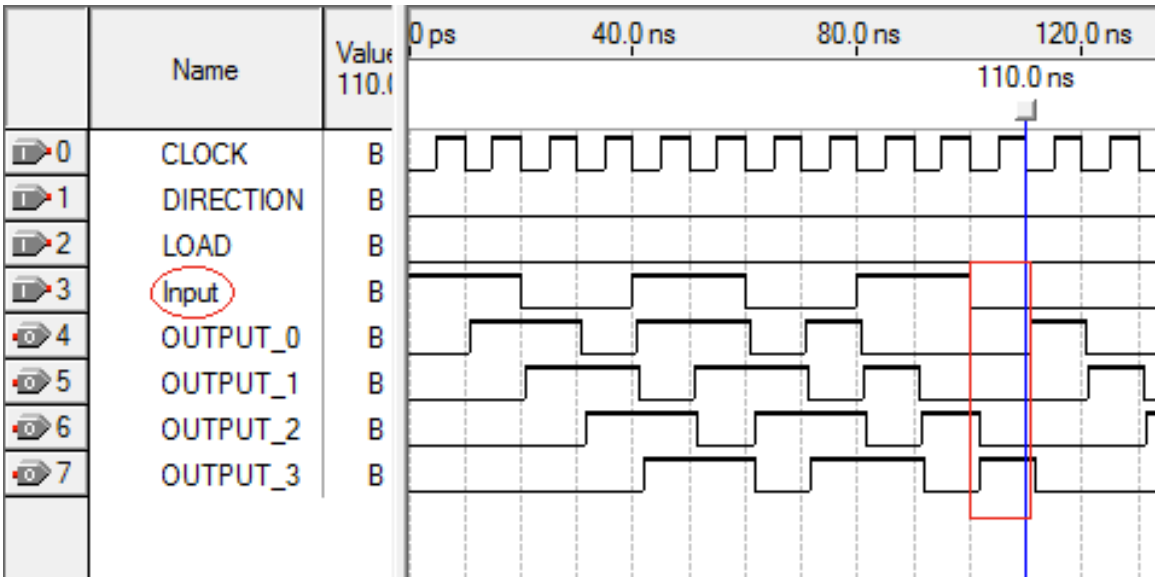


Figure 5: When this sequence is inputted the resulting signature is 0001 as represented by the output boxed in red.

Sequence 3: 1111000011

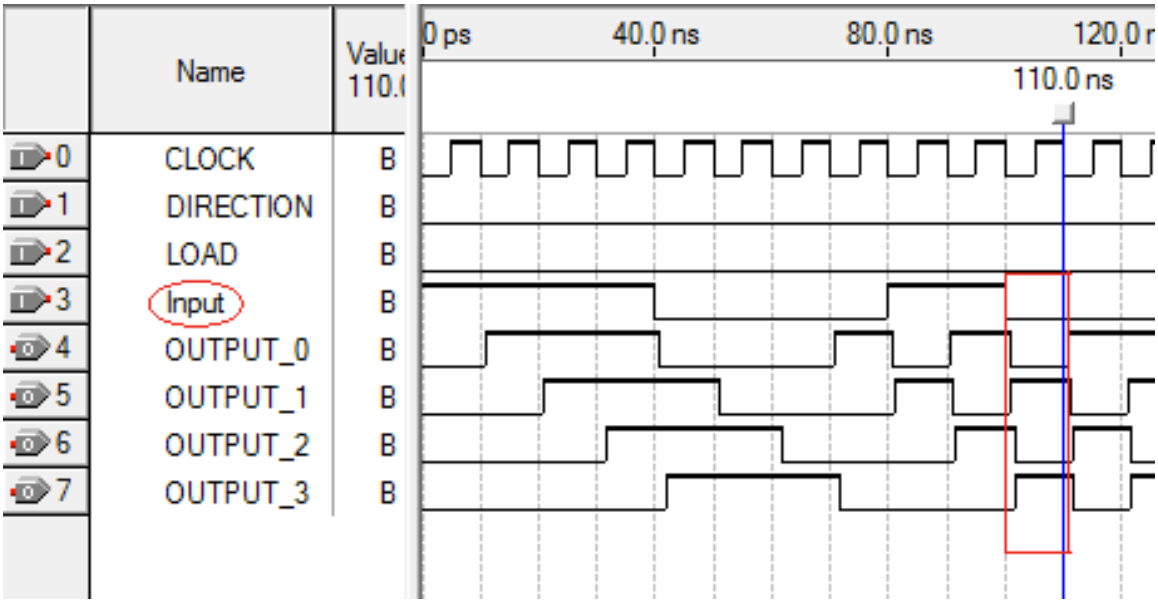


Figure 6: When this sequence is inputted the resulting signature is 0101 as represented by the output boxed in red.

Sequence 4: 1111111100

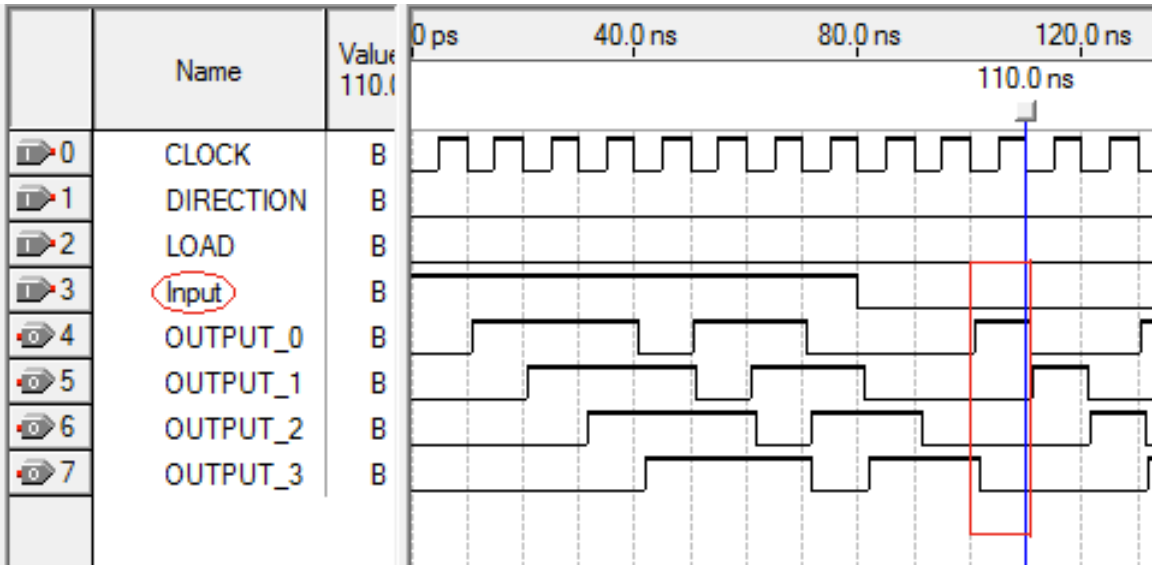


Figure 7: When this sequence is inputted the resulting signature is 1000 as represented by the output boxed in red.

Answers to the questions in the lab handout

[No questions]

Conclusion

This lab familiarized us with working with digital circuit simulations. We learned about the shift register, PRBS generator, and signature analyzer. We designed and implemented these memory components and got experience testing them.

Works Cited

[None]

Signed OFF

(Included)