HEADER

CDA 3201L #002

TIME: 2:00 PM – 3:50 PM (FRIDAYS)

LAB REPORT BY: PARAM CHOKSHI

PURPOSE AND OBJECTIVES

- Understand and implement sequential circuits using different types of flip-flops and other basic logical gates.
- Understand and implement 4-bit synchronous left-shift registers using D flip-flops.
- Understand and implement a 4-bit bi-directional shift register using D flip-flops.
- Utilize the Logisim software to implement the circuits.

ASSIGNMENT:

Part A:

Design a 4-bit synchronous left-shift register using D flip-flops. Your shift register should have an asynchronous parallel load, serial in, serial out, and parallel out bus. You may convert a flip-flop of another type into a D flip-flop, if needed. Note that your DFF has asynchronous clear (CLR) and preset (PR) for loading zero/one.

Part B:

Utilize a 4-bit bi-directional shift register with parallel and serial operating modes.

You have different modes according to the select bits (S0 and S1). According to these two control inputs, we can have "Parallel (broadside) load", "Shift right (in the direction QA toward QD)", Shift left (in the direction QD toward QA), "Inhibit clock (do nothing)".

Design a circuit using only one 4-bit bi-directional shift register and test different cases including:

1. Parallel load

2. Circular Left shift or rotate left

3. Circular Right shift or rotate right

COMPONENTS USED

- Logism (simulation software)
- D flip-flop, Multiplexers, clock, and connection wires (in logism)

DESIGN DESCRIPTION

Part A:

We will implement the 4-bit synchronous left-shift register using 4 D flips flops and 4 multiplexers. Parallel In is represented by inputs D4-D1 that serve as input lines to the D flip flops through the multiplexers. Parallel Out is represented by outputs Q4-Q1. Serial In input is passed to the multiplexer which then get passed to the D flip-flops. A bit enters on each Shift pulse. Serial Out from terminal Q of the last D flip-flop; a bit exits on each Shift pulse. We provide the synchronous parallel load, with Preset, through the multiplexers and into the D flip-flops. The truth table for the multiplexer is shown in Table 2. We use synchronous Clear to reset the states of the D flip-flops to 0. The implementation of the 4-bit synchronous left-shift register is shown in Figure 1. The truth table of the D flip-flop is shown under Table 1, and the truth table of the shift register is shown under Table 3.

Table 1: Truth Table for the D flip-flop.

CLK	D	Q	Q’
0	0	Q	Q’
0	1	Q	Q’
1	0	0	1
1	1	1	0

Table 2: Truth Table for the 2-to-1 MUX with inputs X1 and X2, and select S that allows parallel load.

X1	X2	S	Output
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
0	0	0	0
0	0	1	0
0	1	0	1

0	1	1	1

Table 3: Truth Table for the 4-bit synchronous left-shift register with SERIAL IN = 1

CLK	SERIAL IN	Q1	Q2	Q3	Q4	Serial Out
0	1	1	0	0	0	0
1	1	1	1	0	0	0
2	1	1	1	1	0	0
3	1	1	1	1	1	1

The entries on Table 3 will depend on the previous state of the input to the left. In addition, it will depend on the value if the “SERIAL IN”.

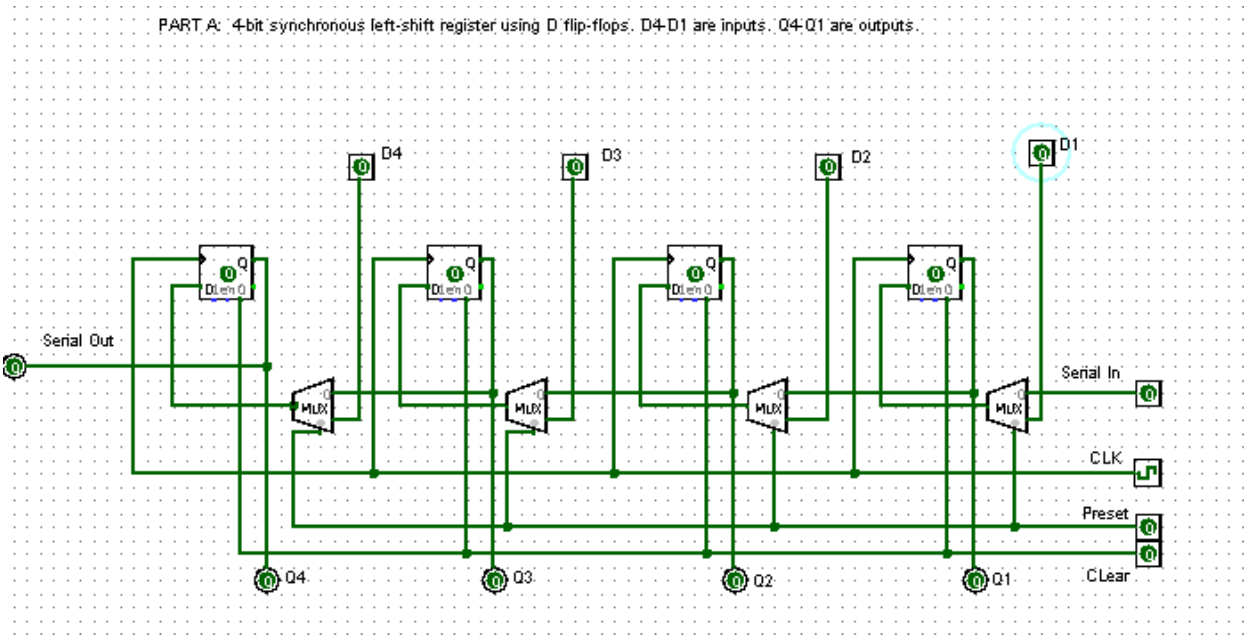

Figure 1: 4-bit synchronous left-shift register using D flip-flops implemented in Logisim.

PART B:

We will implement the 4-bit bi-directional shift register using D flips flops and 4 multiplexers. Parallel In is represented by inputs D4-D1 that serve as input lines to the D flip flops through the multiplexers. Parallel Out is represented by outputs Q4-Q1. To implement shift right we have used Serial In Left as input passed to the multiplexer which then passes the output to the leftmost D flip-flop. To implement shift left we have used Serial In Right as input to the multiplexer which then passes the output to the rightmost D flip-flop. To utilize parallel and serial operating modes we will use the input sequence S1S0 passed to the multiplexer. According to this control inputs we can have parallel load, shift right, shift left and inhibit clock which does nothing. The circuit implementation of the 4-bit bi-directional shift register is shown in Figure 2. The truth table for the control outputs is shown in Table 4, and the truth tables for the 4-bit bidirectional shift register is shown under Tables 5 and 6. For S1S0 = 00. There is no change so the Q1-Q4 outputs remain in the same state they were. For S1S0 = 11. There is a parallel load and the state of the D flip-flops and Q is reset to 0.

Table 4: Truth Table for the Control Inputs S1S0.

S1	S0	MODE
0	0	HOLD/LOAD
0	1	SHIFT RIGHT
1	0	SHIFT LEFT
1	1	PARALLEL LOAD

Table 5: Truth Table for the 4-bit bidirectional shift register with “SERIAL IN LEFT” input.

S1S0	CLOCK	SERIAL IN LEFT	Q1	Q2	Q3	Q4
01	0	1	1	0	0	0
01	1	1	1	1	0	0
01	2	1	1	1	1	0
01	3	1	1	1	1	1

Table 6: Truth Table for the 4-bit bidirectional shift register with “SERIAL IN RIGHT” input.

S1S0	CLOCK	SERIAL IN RIGHT	Q1	Q2	Q3	Q4
10	0	1	0	0	0	1
10	1	1	0	0	1	1
10	2	1	0	1	1	1
10	3	1	1	1	1	1

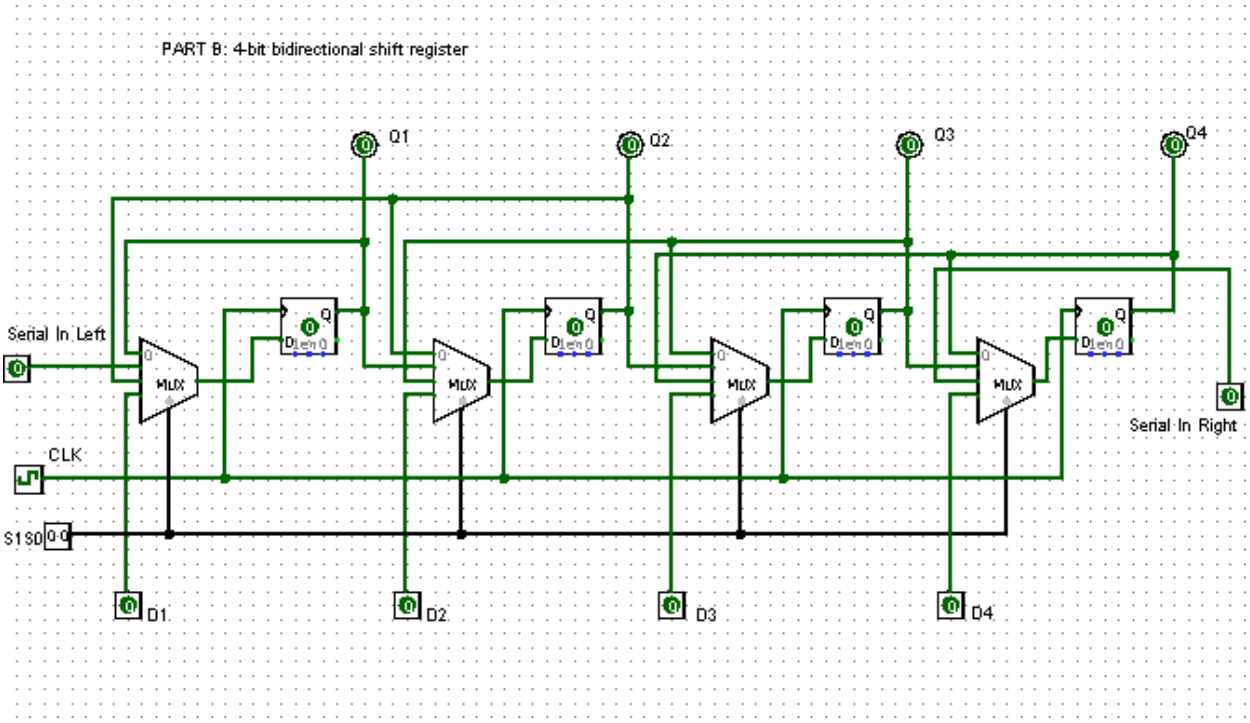

Figure 2: 4-bit bi-directional shift register implemented in Logisim.

OBSERVATION AND DATA ANALYSIS

- Using logical gates, sequential circuits like flip-flops can be used to implement different flip-flops.
- Logic gates helps in implementing simple and cost-effective sequential circuits.
- Shift registers can be implemented using D flip-flops with Multiplexers.

DISCUSSION AND CONCLUSION

- There was no observed difference between the actual and the expected results, due to the fact that logism (a simulation software) was used.