Cal Poly Pomona

4-Bit General-Purpose Register

Experiment 11

Steven Keesler, Andrew Riad
11/14/2025

0.0 PRELAB

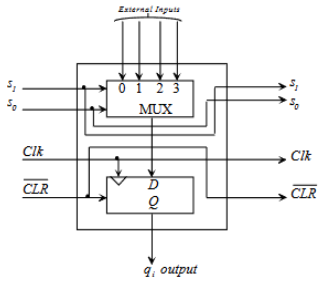
0.1 Design a 4-bit general-purpose register as follows in table 0.1:

Table 0.1: Given State Table

S1	S0	Function
0	0	Load external data
0	1	Rotate left; (A0 <- A3 , Ai <- Ai – 1 for i = 1,2,3)
1	0	Rotate right; (A3 <- A0 , Ai <- Ai + 1 for i = 0,1,2)
1	1	Increment

0.2 Use the following cell S as the building block, as given by figure 0.2:

Figure 0.2: Basic Cell S Diagram

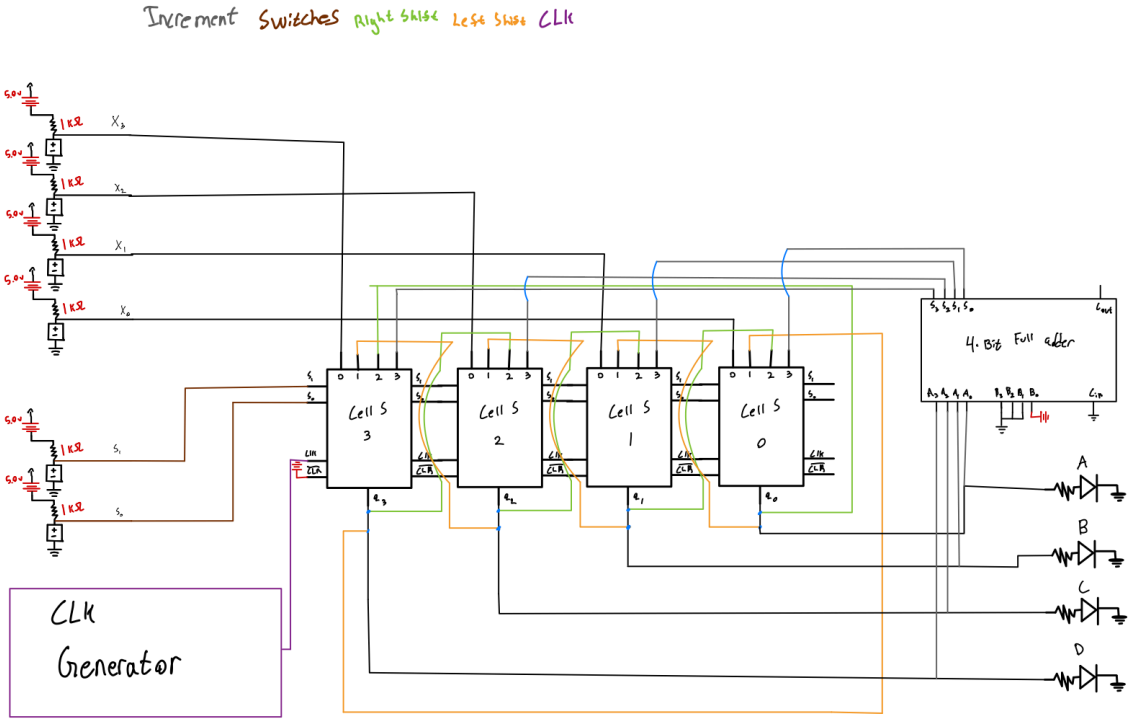


Internal Organization of the Basic Cell *S*

Table 0.2: Schematic State Table

Inputs		Current State				Next State			
S1	S0	T3	T2	T1	T0	Q3	Q2	Q1	Q0
0	0	W	X	Y	Z	W	X	Y	Z
0	1	A3	A2	A1	A0	A2	A1	A0	A3
1	0	A3	A2	A1	A0	A0	A3	A2	A1
1	1	A3	A2	A1	A0	A3+C2	A2+C1	A1+C0	A0+1

Figure 0.3 : 4-Bit General-Purpose Register Circuit Schematic



Logic Explanation of Table 0.2 & Figure 0.3:

Select 0: To load external data, the switches for WXYZ go straight into the Select 0 of each Basic Cell S0-3. (W to T3, X to T2, Y to T1, Z to T0)

Select 1: To rotate left, the output of a given cell, Q0-2, goes into the Select 1 input of the Basic Cell S to its left, T1-3. The output of the basic cell, Q3, goes to the input of T0.

Select 2: To rotate right, the output of a given cell, Q1-3, goes into the Select 2 input of the Basic Cell S to its right, T0-2. The output of the basic cell, Q0, goes to the input of T3.

Select 3: To increment up we use a full adder with the first set of BCD permanently set to 0001, so it's always adding one, and run the outputs from it back into the Select 3 of the original cell it came from.

This is shown in table 0.2 through the use of Carry ins (+C0-2) added to Q1-3, and a simple +1 for Q0.

We use 2 dip switches to determine our selects, and use a waveform generator as needed for the clock.

Figure 0.1 below is the schematic built following table 0.1, and the above logic.

1.0 **INTRODUCTION**

Implement the above design and demonstrate to the instructor. Demonstrate the operations using switches and LEDs, etc. as needed. Schematics will be pulled from Pre-Lab when needed.

2.0 **OBJECTIVES**

The goal is the following:

1. Realize the 3 different states: rotate left, rotate right, and increment, using the minimal amount of required parts.
2. Encode these states to the loop when needed and drive 4 LEDs (displaying BCD).
3. Allow us to load a desired BCD into the register to rotate or increment when needed.
4. Verify operation across all 4 selects.

3.0 **REQUIREMENT**

Implement the state table 3.1 below, pulled from pre-lab table 0.1. Using switches and LEDs, etc. as needed, show Rotate Left, Rotate Right, and Increment of a loaded External Data. Schematics will be pulled from Pre-Lab when needed.

Table 3.1: Given State Table

S1	S0	Function
0	0	Load external data
0	1	Rotate left; (A0 <- A3 , Ai <- Ai – 1 for i = 1,2,3)
1	0	Rotate right; (A3 <- A0 , Ai <- Ai + 1 for i = 0,1,2)
1	1	Increment

4.0 **PARTS LIST**

The Parts list for this experiment is shown below in Table 4.1

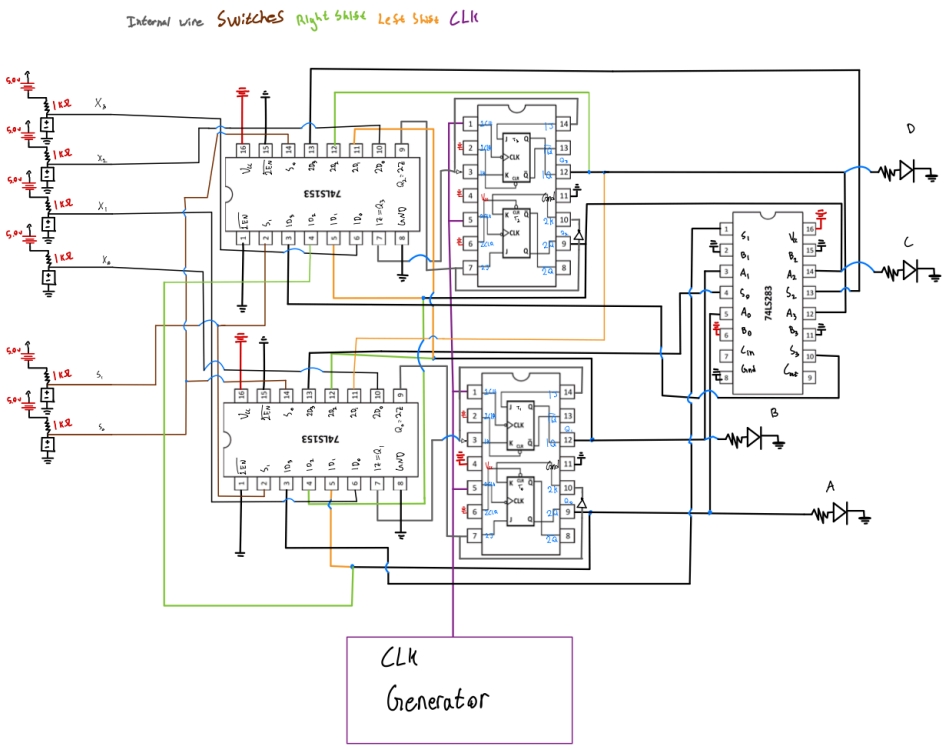
Table 4.1 - Parts List

Item No.	Part Number	Function	Quantity
1	74LS153	4-1 Multiplexer	2
2	74LS04	Inverter	1
3	74LS73	JK Flip Flop	2
4	74LS283	4-bit Full Adder	1
5	DIP Switch	Input	2
6	LED	Output	4

5.0 **DESIGN/IMPLEMENTATION**

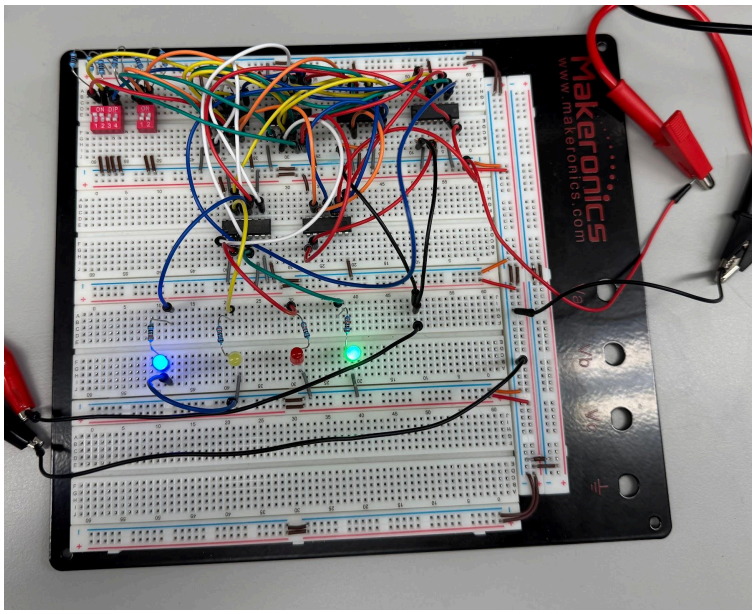
The design from figure 0.3 in the prelab will be tested, and it will be shown again below in figure 5.1. A dip switch will be used to provide the inputs for the parallel load, as well as for the selects, and the outputs for each given state will be shown using 4 LEDs.

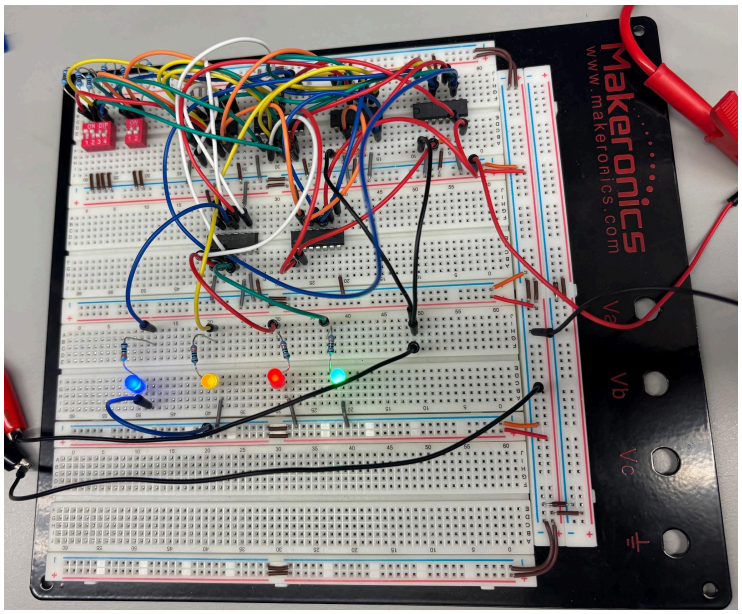
Figure 5.1

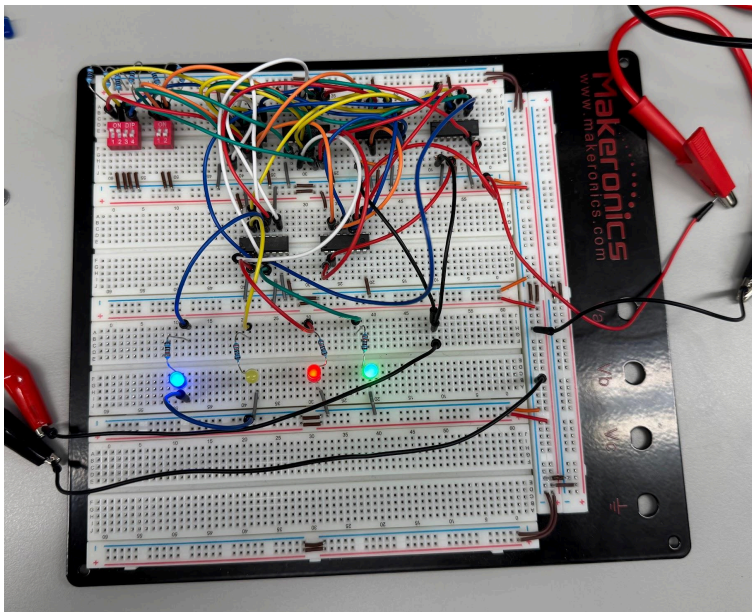


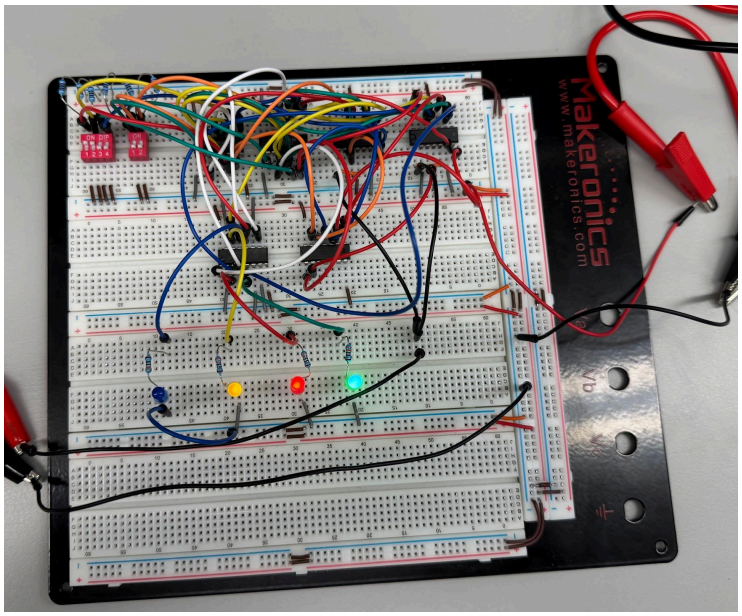
6.0 **TEST SETUP**

Below is an image of the breadboard, the 2 dip switches provide the parallel load for Select 0, and S1/S0 to choose the select. Two 74LS153s & two 74LS137s allow us to implement 4 of the Basic Cell S that was shown in the Pre-Lab given to us. We load in a given input into Select 0, and going through the different selects, the desired effects are shown on the 4 LEDs.









7.0 **TEST RESULTS/VERIFICATION MATRIX**

The test results are summarized below in Table 7.1:

Table 7.1 - State Table of Circuit

States		Variables			Verification
Select (S1 S0)	Function	Test Input	Expected Output	Observed Output	Works? (Y/N)
00	Parallel Load	1001	1001	1001	Y
01	Rotate Left	1001	0011	0011	Y
10	Rotate Right	1001	1100	1100	Y
11	Increment	1001	1010	1010	Y

8.0 Conclusion

The assembled 4-bit register made from two 74LS153s & two 74LS137s has operated correctly: loading inputs at Select 0, Rotating left at Select 1, Rotating Right at Select 2, and Incrementing at Select 3. All of this displayed on our 4 LEDs. The measurements across all 4 states matched the expected behavior from 3.1, as shown above in Table 7.1.

9.0 Post Lab

Design a 4-bit register with a reset input, a parallel input and a positive edge-triggered clock. The 4-bit register is cleared to 0 at the positive edge of the reset. On the other hand, if the load input is high, 4-bit data is transferred to the register at the positive edge of the clock.