Part III

COMBINATIONAL CIRCUITS

Combinational Logic is the bedrock for all digital logic circuits. A combinational circuit's output is determined only by the status of the various inputs and an external clock signal is not necessary as in sequential circuits. All of the circuits completed so far in this manual have been combinational and the two labs in this part of the manual are designed to further develop the concepts of combinational digital logic with two relatively complex examples.

4

ARITHMETIC LOGIC UNIT (ALU)

4.1 PURPOSE

In this lab you will build an Arithmetic Logic Unit (ALU). An ALU is an important digital logic device used to perform all sorts of arithmetic and logic functions in a circuit. The commercial 74181 ALU has two four-bit data inputs along with a one-bit mode (M) and a four-bit select input. Depending on those settings, the device will complete one of the functions listed in Table 4.1.

Select	Logic (M=1)	Arithmetic (M=0)
0000	A'	A
0001	(A + B)'	A + B
0010	A'B	A + B'
0011	Logical 0	minus 1 (2's Comp)
0100	(AB)'	A + AB'
0101	B'	(A + B) plus AB'
0110	A XOR B	A minus B minus 1
0111	AB'	AB' minus 1
1000	A' + B	A plus AB
1001	(A XOR B)'	A plus B
1010	B	(A + B') plus AB
1011	AB	AB minus 1
1100	Logical 1	A plus A
1101	A + B'	(A + B) plus A
1110	A + B	(A + B') plus A
1111	A	A minus 1

Table 4.1: Function Table for 74181 ALU

Notes: in the "Arithmetic" column, the + sign indicates logic *OR* while the words *plus* and *minus* indicate arithmetic add and subtract operations. The value of *A plus A* is the same as shifting the bits left to the next most significant position.

The ALU built in this lab is not as complex as a 74181 Integrated Circuit (IC), however it demonstrates the basic functions of an ALU.

4.2 PROCEDURE

This is a rather complex circuit so several completed subcircuits are provided.

Load the ALU starter circuit in *Logisim-evolution*. That starter circuit already has the `main`, `ALU`, and `Arithmetic` subcircuits completed.

4.2.1 *main*

The `main` circuit does nothing more than provide a human-friendly interface for the rest of the ALU. That interface include two four-bit inputs (labeled *InA* and *InB*), a three-bit select, a one-bit mode, a carry-in and carry-out bit (so the ALU could be chained to another to create an eight-bit device), a *compare* output (TRUE if the two inputs are equal), and a four-bit output (labeled *ALUOut*). In operation, numbers are entered on *InA* and *InB*, the mode and select are set, and then the result is read on *ALUOut*.

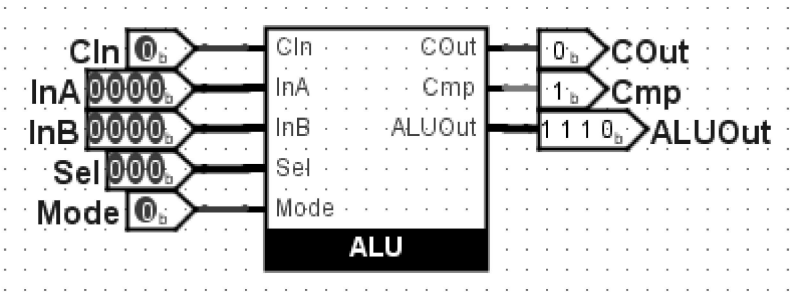

Figure 4.1: ALU main

4.2.2 *ALU*

The `ALU` subcircuit contains the logic that routes *InA*, *InB*, and *Sel* to two other subcircuits, `Arithmetic` or `Logic`. It then uses a multiplexer to route the output of one of those subcircuits to an output port depending on the setting of the *Mode* bit. Note that the inputs are sent to both subcircuits but only the output specified by the *Mode* is returned to the user. This type of logic is also used in the `Arithmetic` circuit.

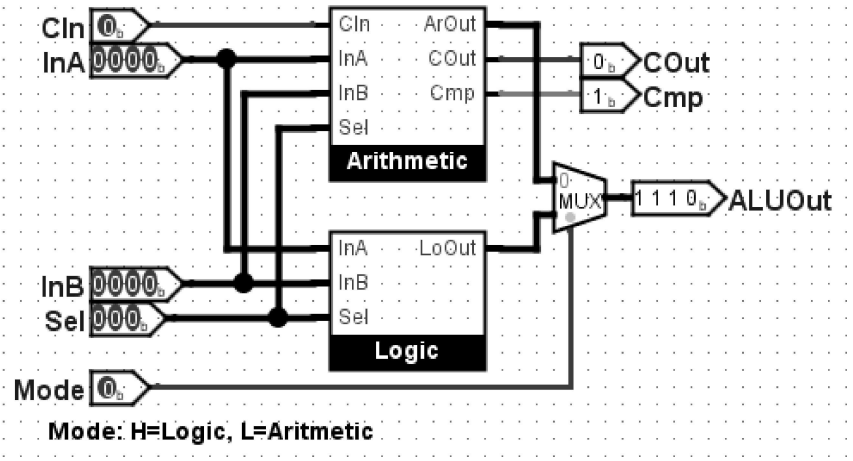

Figure 4.2: ALU Subcircuit

4.2.3 *Arithmetic*

This subcircuit contains numerous devices from the *Arithmetic* library and they are all wired appropriately for whatever operation is selected. The concept for this subcircuit is rather simple but routing the wiring to all of the devices is challenging.

Notice that two multiplexers are necessary since the circuit provides two different outputs. The top multiplexer routes the four-bit solution and the bottom multiplexer routes the carry-out bit. The *compare* output is always active since it is comparing the input signals and does not rely on the function that is selected.

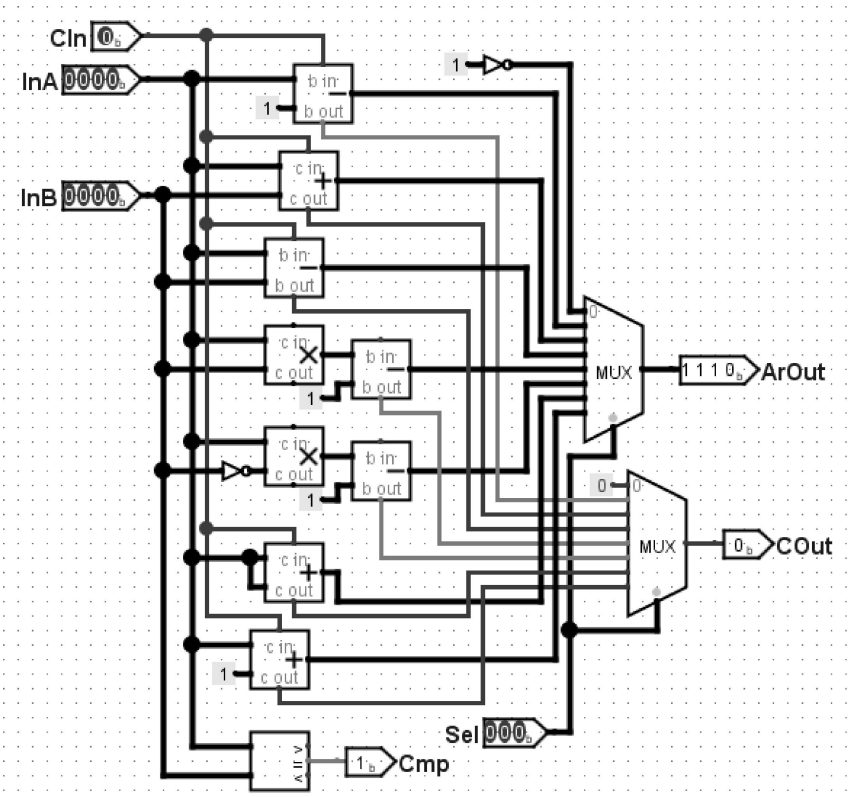

Figure 4.3: Arithmetic Subcircuit

4.2.4 *Challenge*

In the starter circuit, the `Logic` subcircuit is only a shell with three inputs and one output.

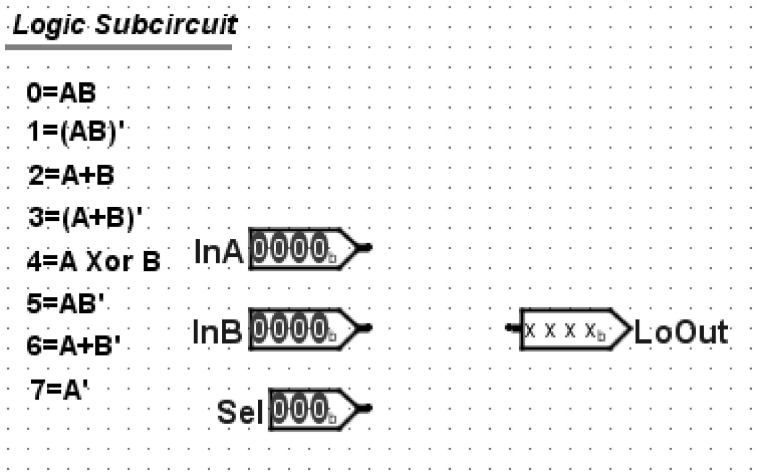

Figure 4.4: Logic Subcircuit

Complete that subcircuit by adding the necessary logic gates and wiring, similar to the `Arithmetic` subcircuit. This subcircuit is much

simpler than the `Arithmetic` subcircuit since there are no carry-in, carry-out, or compare bits. When completed, the subcircuit only needs eight logic gates and a multiplexer added to the starter.

4.2.5 *Testing the Circuit*

The `ALU` should be tested by entering several values on *InA* and *InB* and then select all possible arithmetic and logic operations. The outputs for each check should be accurate.

4.3 DELIVERABLE

To receive a grade for this lab, complete the Challenge. Be sure the standard identifying information is at the top left of the *main* circuit, similar to this:

```
George Self
Lab 04: ALU
February 18, 2018
```

Save the file with this name: *`Lab04_ALU`* and submit that file for grading.