8

ARITHMETIC LOGIC UNIT (ALU)

8.1 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 8.1.

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 8.1: 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 an 74181 IC, however it demonstrates the basic functions of an ALU.

8.2 PROCEDURE

Load the ALU starter circuit in *Logisim-evolution*.

8.2.1 *main*

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

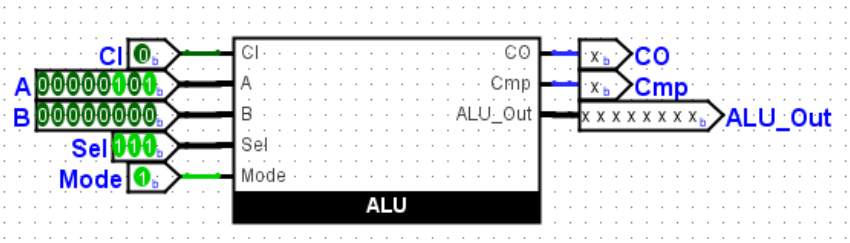

Figure 8.1: ALU main

8.2.2 *ALU*

The `ALU` subcircuit is designed to contain the logic that routes *A*, *B*, and *Sel* to two other subcircuits, `arith\_main` or `logic\_main`. 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 Labs 3 and 4.

The Arithmetic and Logic subcircuits that were built in Labs 3 and 4 will be reused for this lab. This is the way that circuit designers can reuse their work to build more complex circuits without having to reinvent the proverbial wheel for every project. To reload those old labs, click PROJECT -> LOAD LIBRARY -> LOGISIM-EVOLUTION LIBRARY[1]. Find Lab 3, Arithmetic, and click *Open*. That lab is now available as a library in the library list on the left side of the *Logisim-Evolution* workspace. Follow the same procedure to also load Lab 4, Logic, as a library.

Open the Arithmetic library and drop the `arithmetic` subcircuit in the `ALU` subcircuit. This process is exactly like dropping a device from the Wiring library and should not be difficult to figure out. In the

1 In order to load labs for reuse it is important to store the project files in the same folder.

same way, drop the `logic` subcircuit from the Logic library onto the `ALU` subcircuit.

At this point, the ALU subcircuit should resemble Figure 8.2.

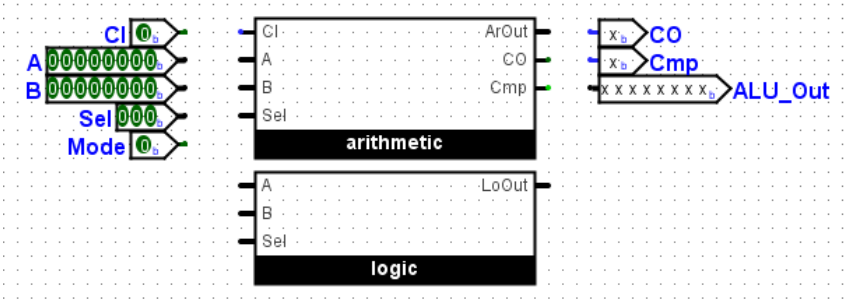

Figure 8.2: ALU Subcircuit

8.2.3 *Challenge*

Continue to wire the `ALU` subcircuit. Note: the inputs and outputs that were provided will need to be repositioned in order to complete this build.

- Wire the *CI* input to the *CI* port on the *arithmetic* device
- Wire inputs A and B to ports *A* and *B* on both devices.
- Wire the *Sel* input to the *Sel* ports on both devices.
- The *arithmetic* *CO* and *Cmp* ports should be wired to the *CO* and *Cmp* outputs
- Add an 8-bit, 2-input multiplexer. The inputs should be wired to the output ports on both devices. The multiplexer select bit should be wired to the *Mode* input. Finally, the multiplexer output should be wired to the *ALU_Out* output.

8.2.4 *Testing the Circuit*

The ALU can be tested from the `main` circuit. Several values can be entered on *A* and *B* and then various arithmetic and logic operations selected. The outputs for each check should be accurate. A test vector file has been provided for this lab so all of the ALU's functions can be exercised.

8.3 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 08: ALU
February 18, 2018
```

Save the file with this name: *Lab08_ALU* and submit that file for grading.