3

ARITHMETIC OPERATIONS

3.1 PURPOSE

This lab develops an arithmetic unit that includes eight different arithmetic functions using *Logisim-Evolution* library components. This device will eventually be used as part of the Arithmetic Logic Unit (ALU) in Lab 8. This device will have two inputs, labeled *A* and *B*, and will output the following calculations.

1. -1
2. $A - 1$
3. $A + B$
4. $A - B$
5. $AB - 1$
6. $AB' - 1$
7. $A + A$
8. $A + 1$

3.2 PROCEDURE

Start a new *Logisim-Evolution* project and create a subcircuit named `arithmetic` that will eventually contain the entire arithmetic unit. Begin the build by adding eight devices in the subcircuit as in Figure 3.1. *Notes: exact device placement is not important at this point since they can be repositioned as necessary; however, they should be in the correct order on the subcircuit. Also, all of the devices, inputs, and outputs need to be set for eight data bits in the properties panel.*

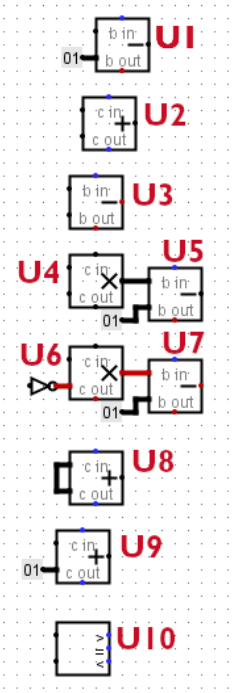

Figure 3.1: Placing the Arithmetic Components

Here are the devices in Figure 3.1. *Note: the device numbers were added to the illustration as an aid for the following discussion; they will not be present in the Logisim-Evolution subcircuit.*

- U1-Subtractor (*Arithmetic* library). This device subtracts the bottom input from the top input on its east side and sends the result to the output on its west side.
- Constant (*Wiring* library). Subtractor *U1* is intended to supply the *A-1* output and the "01" on its bottom input is a constant used in this calculation. It should be placed near the bottom input for subtractor *U1* and its properties set for Facing: East, Data Bits: 8, Value: 0x1 (this means hexadecimal 1).
- U2-Adder (*Arithmetic* library). This device adds the top and bottom inputs on its east side and sends the result to the output on its west side.
- U3-Subtractor (*Arithmetic* library). This device subtracts the bottom input from the top input on its east side and sends the result to the output on its west side.
- U4-Multiplier (*Arithmetic* library). This device multiplies the top and bottom inputs on its east side and sends the result to the output on its west side.
- U5-Subtractor (*Arithmetic* library). This device is connected to multiplier U4 output and is intended to subtract one from its product.

- Constant (*Wiring* library). Because subtractor U5 is designed to subtract one from the product of U4, the "01" on its bottom input is a constant. It should be placed near the bottom input and its properties set for Facing: East, Data Bits: 8, Value: 0x1 (this means hexadecimal 1).
- U6-Multiplier (*Arithmetic* library). This device multiplies the top and bottom inputs on its east side and sends the result to the output on its west side.
- Not Gate (*Gates* library). This is placed near the bottom input of multiplier U6 in order to negate that input.
- U7-Subtractor (*Arithmetic* library). This device is connected to multiplier U6 output and is intended to subtract one from its product.
- Constant (*Wiring* library). Because subtractor U7 is designed to subtract one from the product of U6, the "01" on its bottom input is a constant. It should be placed near the bottom input and its properties set for Facing: East, Data Bits: 8, Value: 0x1 (this means hexadecimal 1).
- U8-Adder (*Arithmetic* library). This device is designed to output the value of $A + A$ so the two inputs on its east side are tied together. The result is sent to the output on its west side.
- U9-Adder (*Arithmetic* library). This device adds the top and bottom inputs on its east side and sends the result to the output on its west side.
- Constant (*Wiring* library). Adder *U9* is intended to supply the *A+1* output and the "01" on its bottom input is a constant. It should be placed near the bottom input for adder *U9* and its properties set for Facing: East, Data Bits: 8, Value: 0x1 (this means hexadecimal 1).
- U10-Comparator (*Arithmetic* library). A comparator compares the two inputs on its east side. If the top input is greater than the bottom input then output ">" will go high. If the top and bottom inputs are equal then output "=" will go high. If the top input is less than the bottom input then output "<" will go high.

The next step is to place all of the inputs and outputs. Figure 3.2 shows where those items should go and the label for each item. Note: exact placement is not important since they can be repositioned. The *Radix* property for each of the inputs and outputs should be set to "Hexadecimal." The *Data Bits* property should be set as follows.

- CI: 1
- A: 8
- B: 8
- ArOut: 8
- CO: 1
- Cmp: 1

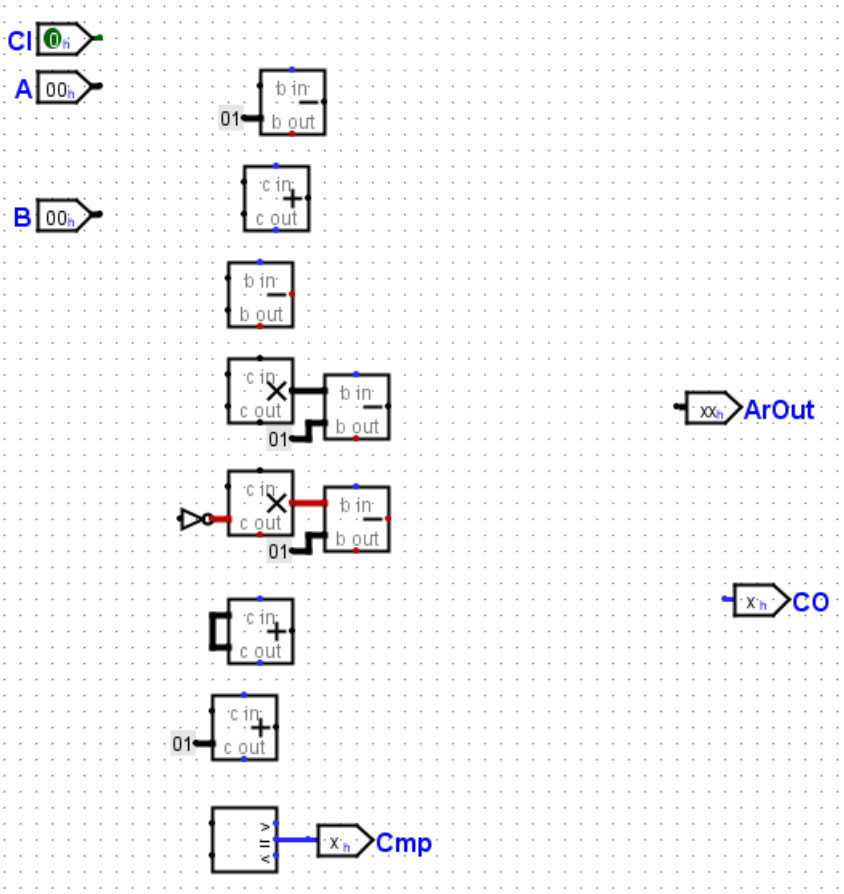

Figure 3.2: Arithmetic Inputs and Outputs

This circuit uses a Multiplexer (*Plexers* library) to select which device's output to connect to *ArOut*. A multiplexer is a digital logic workhorse that is found in many circuits, including Central Processing Units (CPUs). It is designed to switch a selected input to the output while ignoring all other input ports. In Figure 3.3, two multiplexers have been placed in the circuit. The top multiplexer will select one of eight inputs coming from the eight arithmetic devices to connect to *ArOut*. The bottom multiplexer will connect the carry out signal from the selected arithmetic device to the *CO* output. Also notice that the bottom multiplexer has a constant zero wired to input port zero.

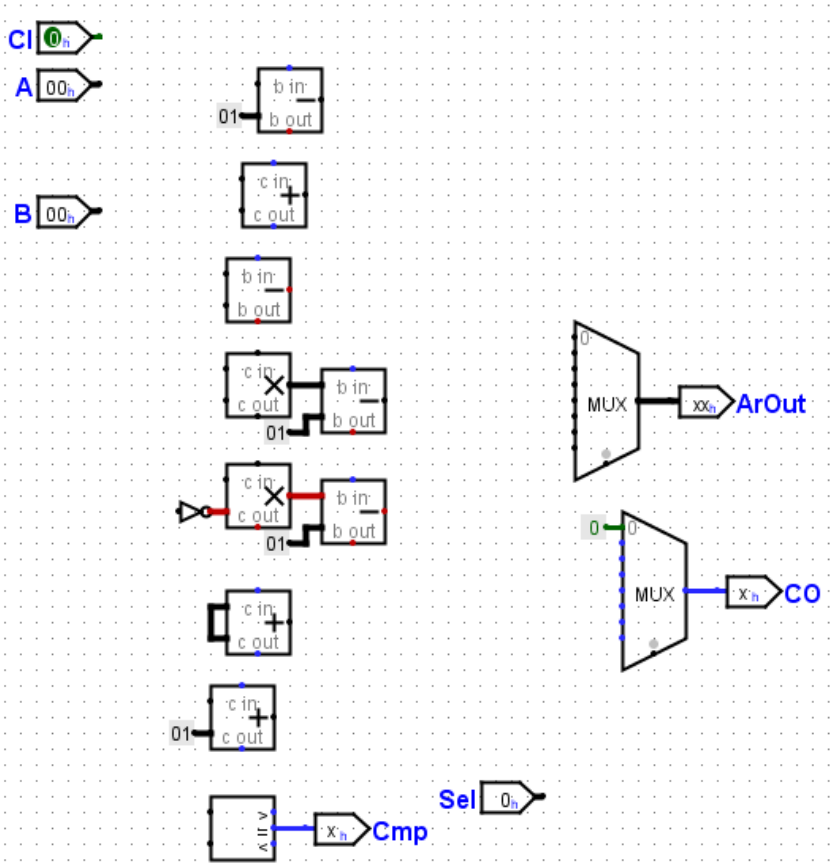

Figure 3.3: Placing the Multiplexers

The next step is to wire inputs *A* and *B* to each of the devices, as shown in Figure 3.4. Then the outputs of each device is wired to an appropriate port in the top multiplexer. This is not particularly challenging, but be careful to avoid crossed wires.

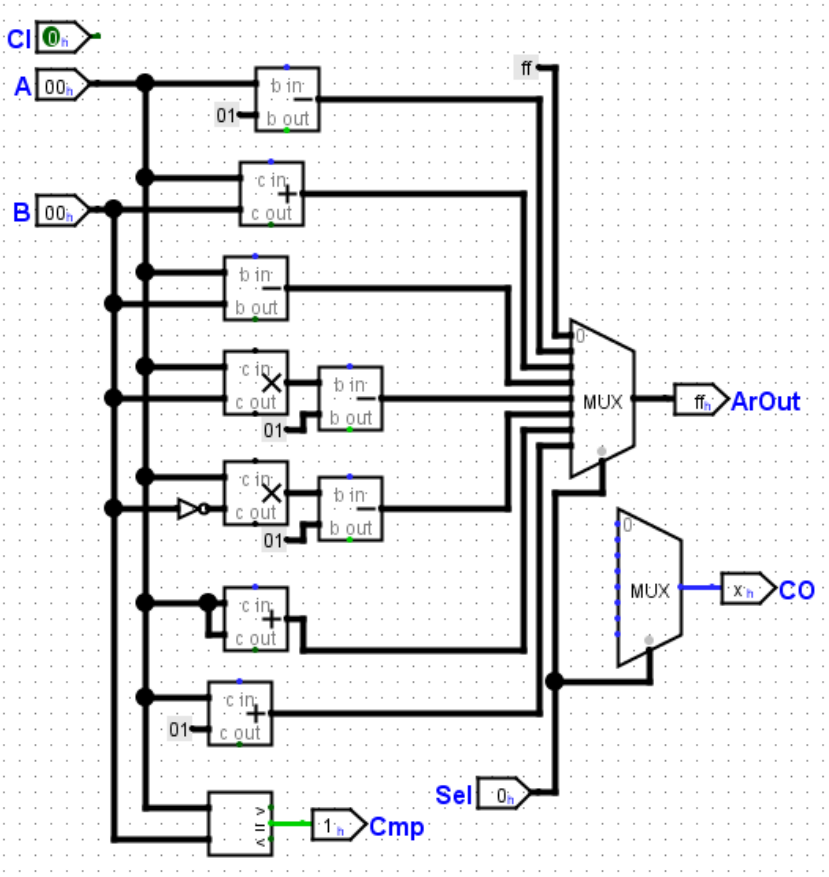

Figure 3.4: Wiring the Data Inputs and Outputs

Figure 3.4 shows the completed subcircuit with wires connecting *CI* to each device and then the devices wired to the appropriate port on the bottom multiplexer.

Notice that input zero on the top multiplexer is wired to a constant *ff*. That is the two's complement of negative one so that port will always transmit negative one, as in the specification sheet.

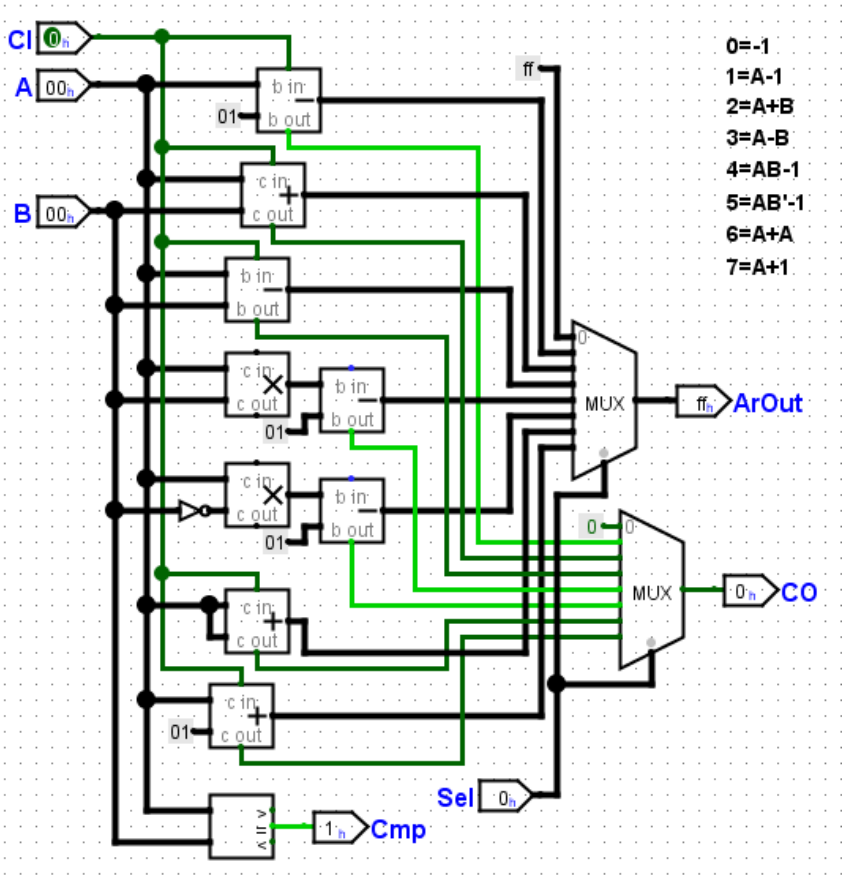

Figure 3.5: Arithmetic Final Circuit

Finally, the `main` circuit is completed by dropping the `arithmetic` subcircuit on the canvas and wiring an appropriate input or output port to each of the ports on the subcircuit. Figure 3.6 illustrates the main circuit.

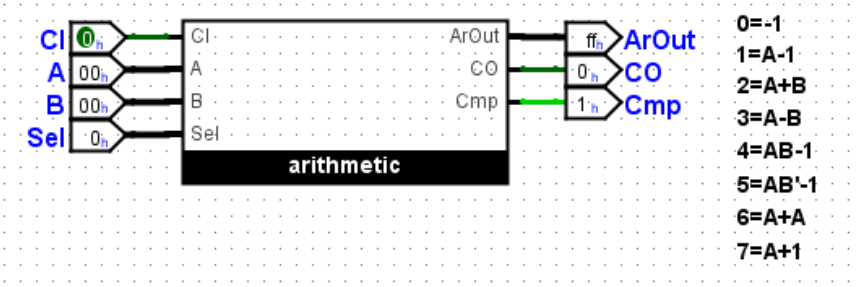

Figure 3.6: Arithmetic Main Circuit

This arithmetic device can be tested by entering numbers for the various inputs and checking to see if the output is correct. For example, if input *A* was set to 3 and input *B* were set to 4, then if *Sel* is set to 2, *ArOut* should be seven $(3+4)$. A test vector file has been provided for this lab so all of the arithmetic functions can be exercised.

As a last step, the `main` circuit must be renamed since this circuit will be reused in Lab 8. Click one time on the `main` circuit to activate it and in the properties panel change its label to `arith_main`.

3.3 DELIVERABLE

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

```
George Self
Lab 03: Arithmetic Operations
September 17, 2019
```

Save the file with this name: *`Lab03_Arithmetic`* and submit that file for grading.