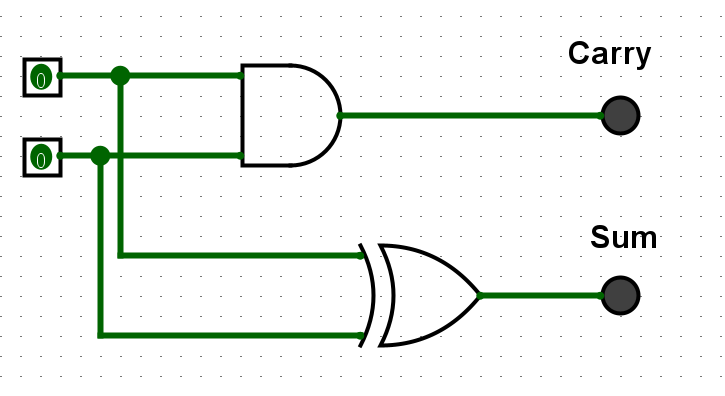
1. Start Logisim and open a new canvas

2. Using this week's lectures as a guide, construct a half-adder and test it.

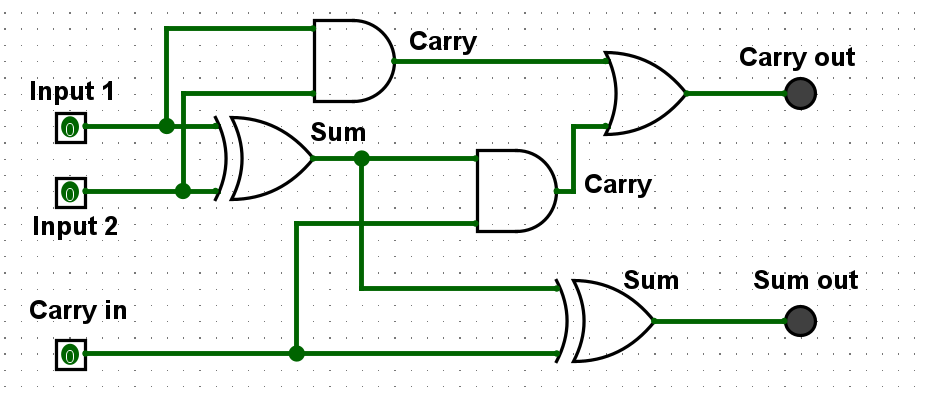


**Check its correctness by testing and filling out a truth table like the following. Add the circuit screen shot and the table to your submission document:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Input 1** | **Input 2** | **Sum output** | **Carry output** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

3. Save the current circuit.

4. Now extend your half-adder to a full-adder, which in addition to the two input pins, also handles a *carry-in bit.*



**Check its correctness by testing and filling out a truth table like the following. Add the**

**circuit screen shot and the table to your submission document:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input 1** | **Input 2** | **Carry in** | **Sum out** | **Carry out** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

5. Now you’re going to build a 4 bit adder. Before you start, plan it out. Think about the parts

you need. Review this week’s lecture slides on full-adders, and watch the video linked

under resources. Discuss it with your demonstrator if you need to.

6. A 4-bit adder sums together two 4-bit binary numbers. Each bit of each number is represented by a binary on/off pin. So, you will need two sets of four input pins.

6.1.

Layout the pins you need for each input bit.

* Input A: Represents the first 4-bit number, each bit of the number is equivalent to each pin
* Input B: Represent the last 4-bit number, each pin connects to one bit of the number

6.2. While doing this, workout the order of your bits (Big-endian and little-endian)*.* Use labels to indicate the significance of each bit (i.e., which column, from least significant (20 column) to most significant (23 column).

Input A (big-endian): A3 (MSB) - A2 - A1 - A0 (LSB)

- **A3** – Represents the 23 column. (Most Significant Bit, MSB)

- **A2** – Represents the 22 column.

- **A1** – Represents the 21 column.

- **A0** – Represents the 20 column. (Least Significant Bit, LSB)

Input B (Little-endian) B0 (MSB) - B1 - B2 - B3 (LSB)

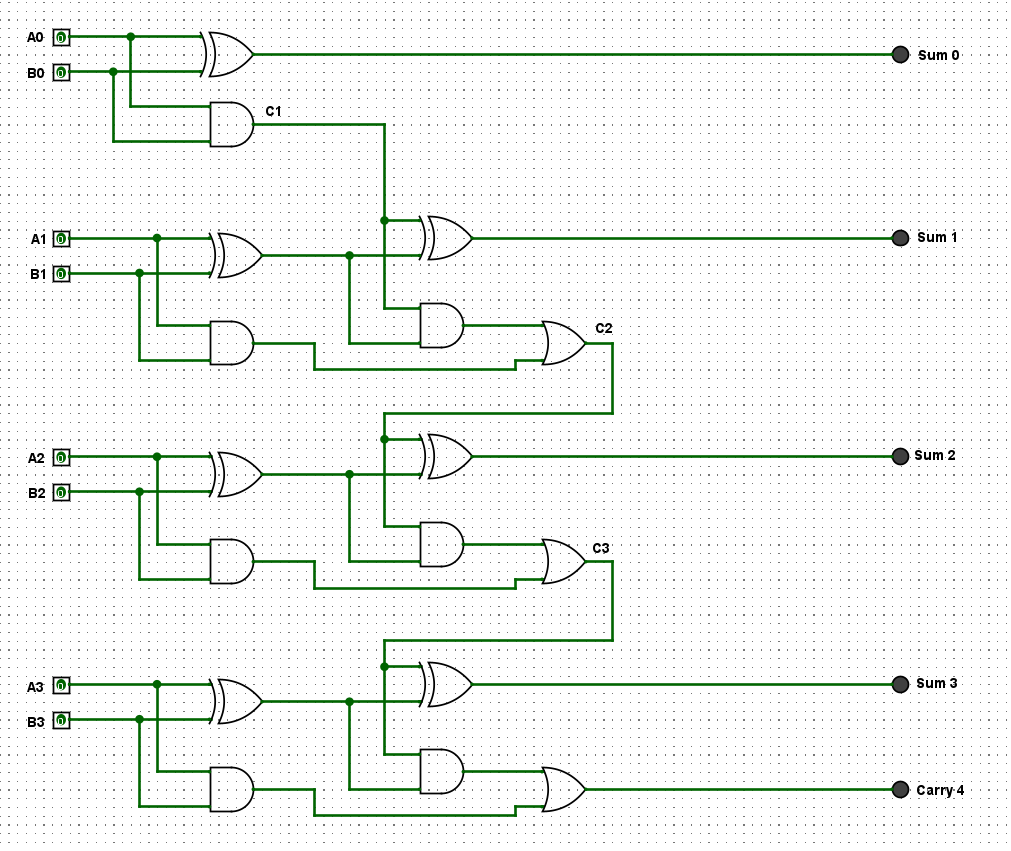
**- B3** – Represents the 23 column. (Most Significant Bit, MSB)

**- B2** – Represents the 22 column.

**- B1** – Represents the 21 column.

**- B0** – Represents the 20 column. (Least Significant Bit, LSB)

1. Now start implementing your adder. Use a half-adder to add the first bits from both binary numbers, and wire up an LED to represent the output bit. We did this last week so it should now be straight forward!
2. Now wire-up a full-adders to add all remaining bits. Remember that a full-adder also adds the carry-in bit from the previous adder. Use an LED to show the sum output for each column.



|  |  |  |
| --- | --- | --- |
| **Input A** | **Input B** | **Output** |
| 0101 | 0000 | 0101 |
| 0101 | 0001 | 0100 |
| 0101 | 0010 | 0111 |
| 0101 | 0011 | 0110 |
| 0101 | 0100 | 0011 |
| 0101 | 0101 | 0010 |
| 0101 | 0110 | 0000 |
| 0101 | 0111 | 0001 |
| 0101 | 1000 | 1101 |
| 0101 | 1001 | 1100 |
| 0101 | 1010 | 1111 |
| 0101 | 1011 | 1110 |
| 0101 | 1100 | 1011 |
| 0101 | 1101 | 1010 |
| 0101 | 1110 | 1000 |
| 0101 | 1111 | 1001 |