Faculty of Science, Engineering and Technology

# Computer Systems

Week 2

## Overview

This laboratory session will be focussed on building a full adder of multiple bits, and working with Flip-Flops. You should spend time reviewing the week's lecture, and watching the tutorial videos linked below.

**Student Name**: Lau Ngoc Quyen

**Student ID**: 104198996

**Purpose:** Building a 4 bit adder

**Task:**

**Time:** This lab is due by the start of your week 3 lab.

**Assessment:** This lab is worth 1% (up to a maximum of 5%) of your assessment for this unit, and only if demonstrated to your lab demonstrator in the week it is due.

**Resources:**

- Swin tutorials
  - Full-adder tutorial
  - Intro to Flip Flops

### *Submission Details*

You must submit the following files to Canvas:

- A document containing all required work as described below.

SWINBURNE UNIVERSITY OF TECHNOLOGY

# Instructions

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:**

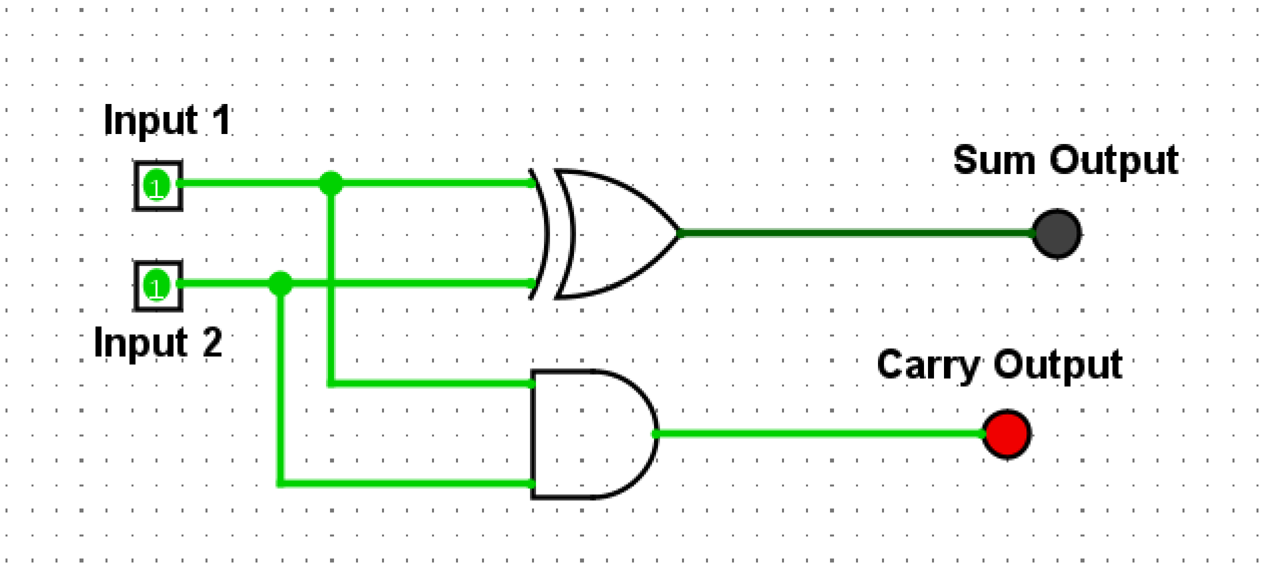


***Image 1: Half-adder circuit***

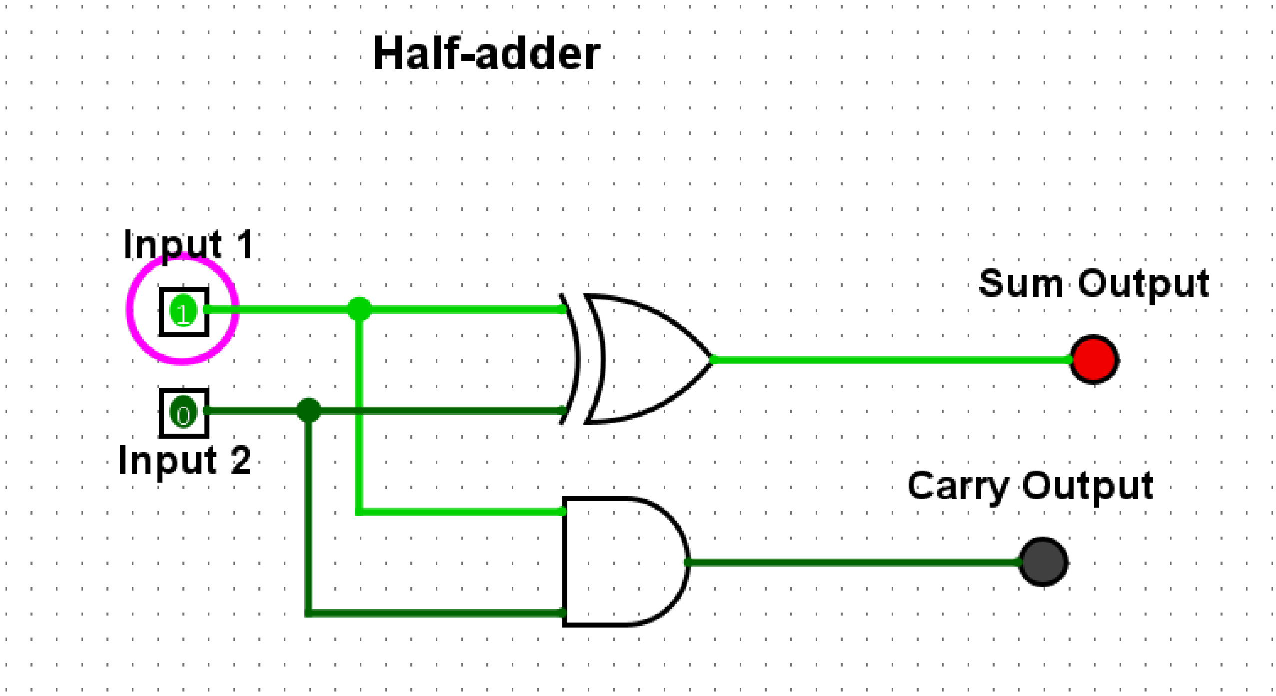


*Image 1.1: Input 1= 1; Input 2 = 0*

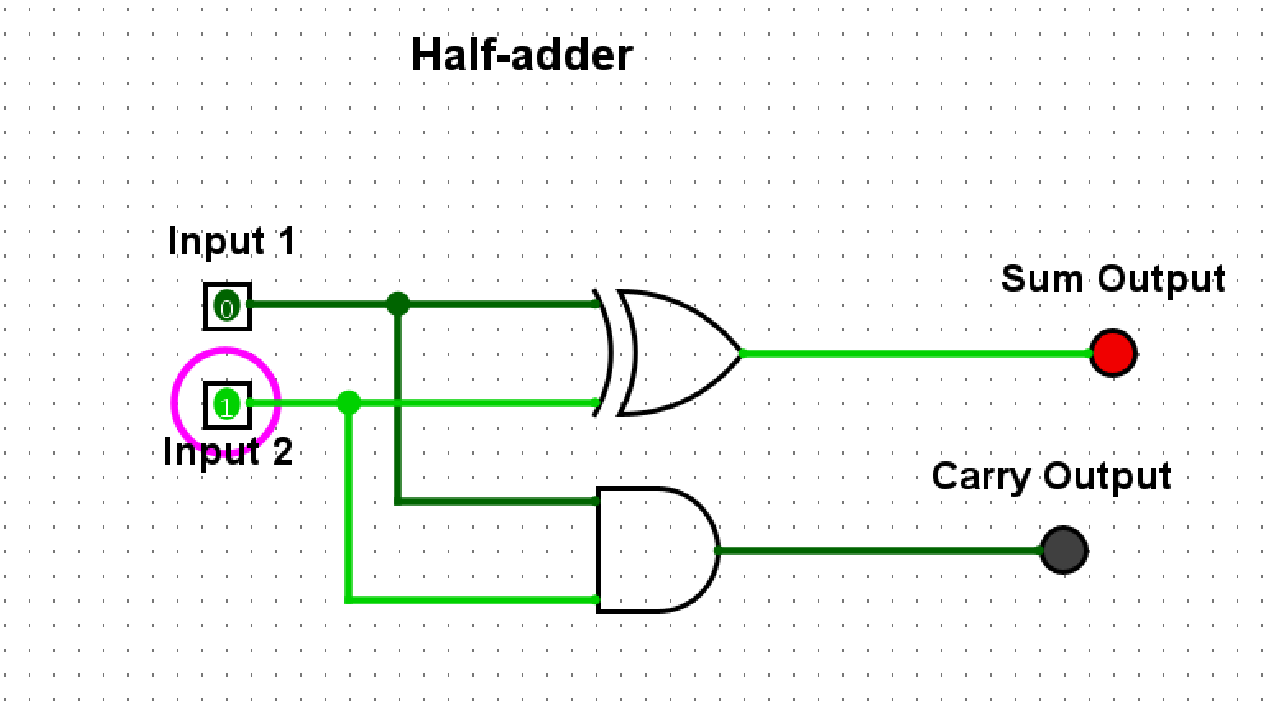


*Image 1.2: Input 1= 0; Input 2 = 1*

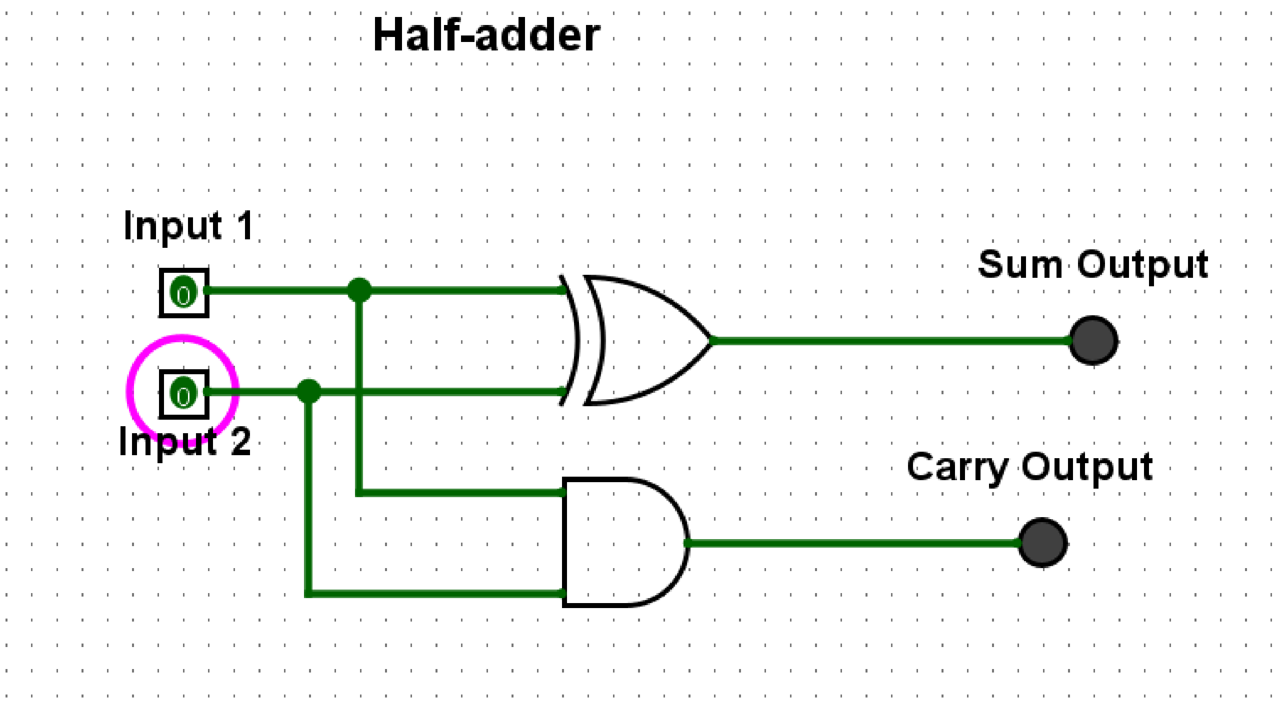


*Image 1.3: Input 1= 0; Input 2 = 0*

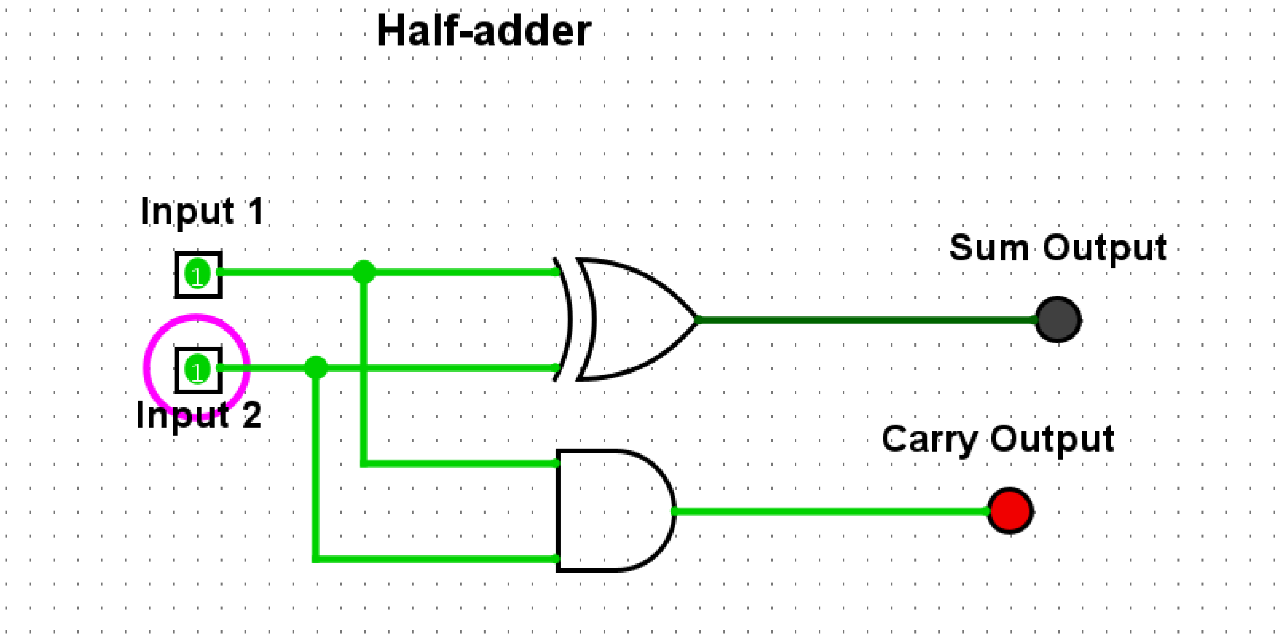


*Image 1.4: Input 1= 1; Input 2 = 1*

| Input 1 | Input 2 | Sum Output | Carry Output |
|---|---|---|---|
| 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 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.*

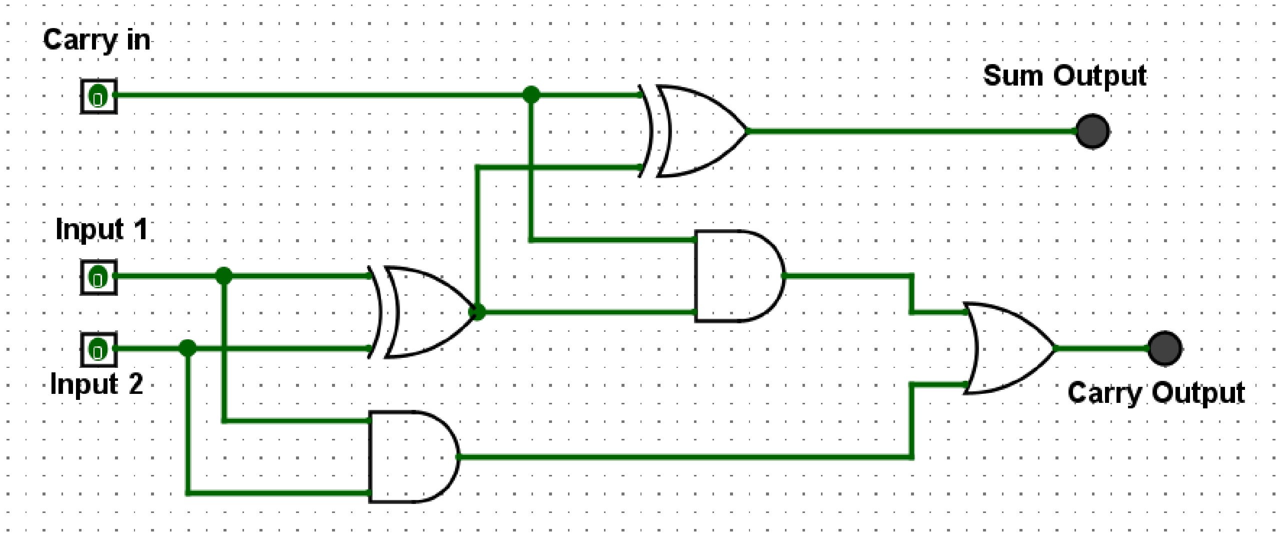


***Image 2**: Input 1 = 0; Input 2 = 0; Cin = 0*

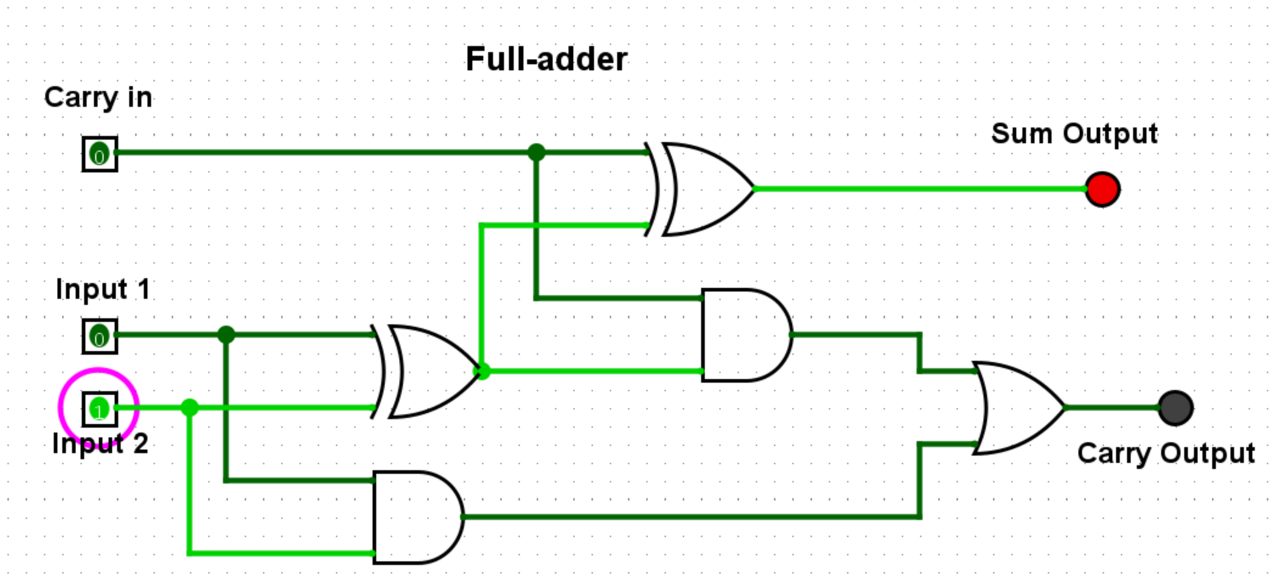


***Image 2.1*: Input 1 = 0, Input 2 = 1, Cin = 0**

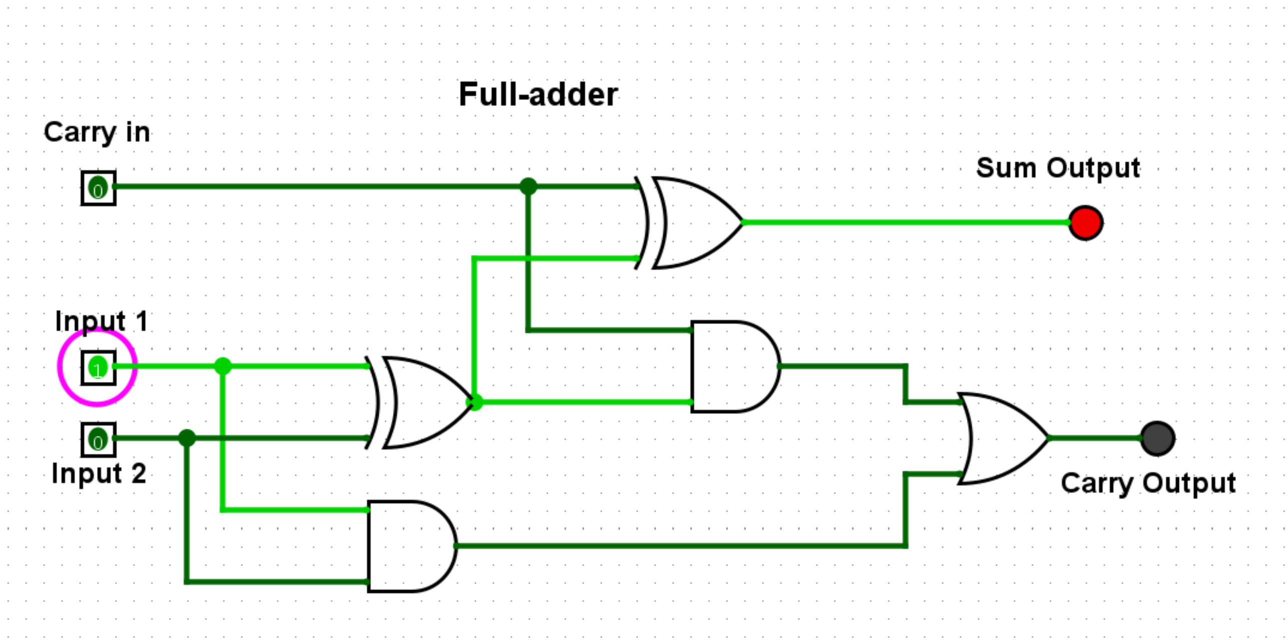


***Image 2.2*: Input 1 = 1, Input 2 = 0, Cin = 0**

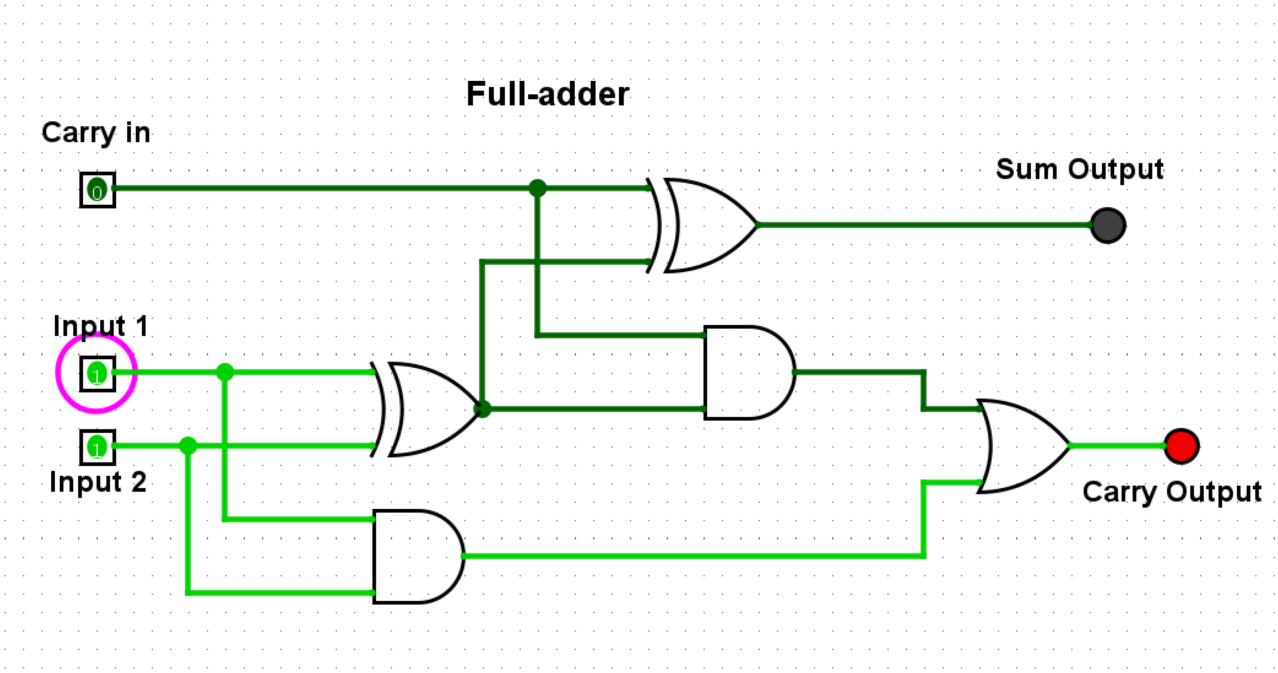


***Image 2.3**: Input 1 = 1, Input 2 = 1, Cin = 0*

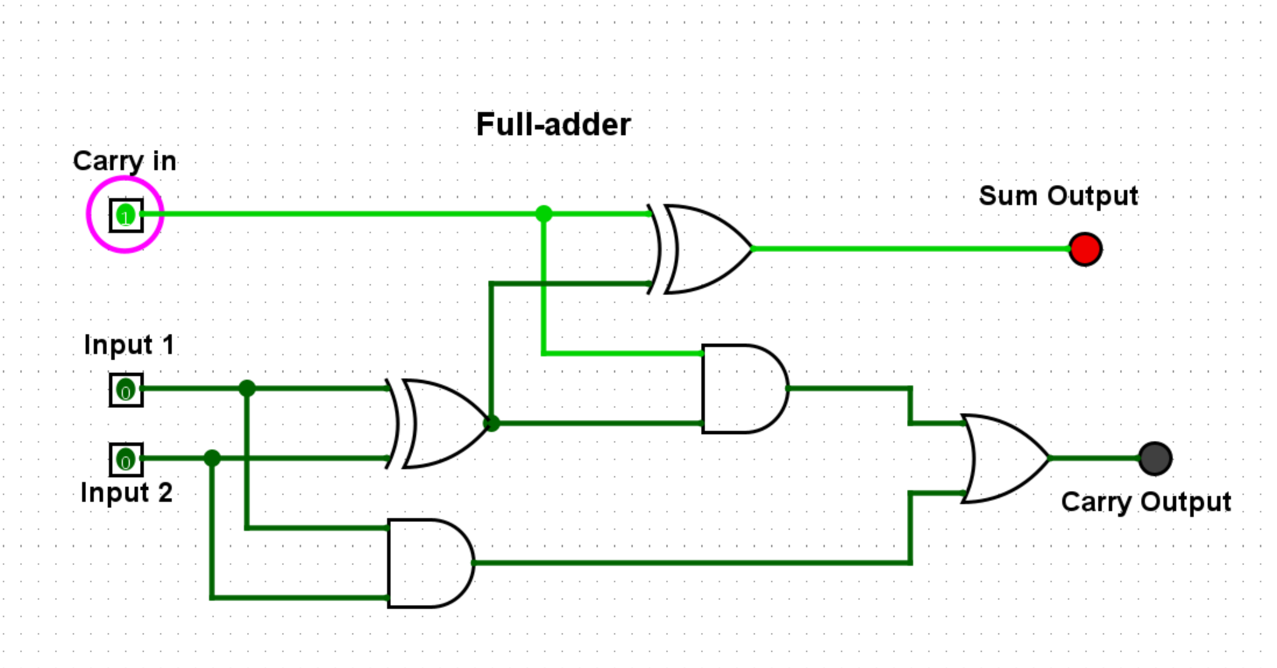


***Image 2.4**: Input 1 = 0, Input 2 = 0, Cin = 1*

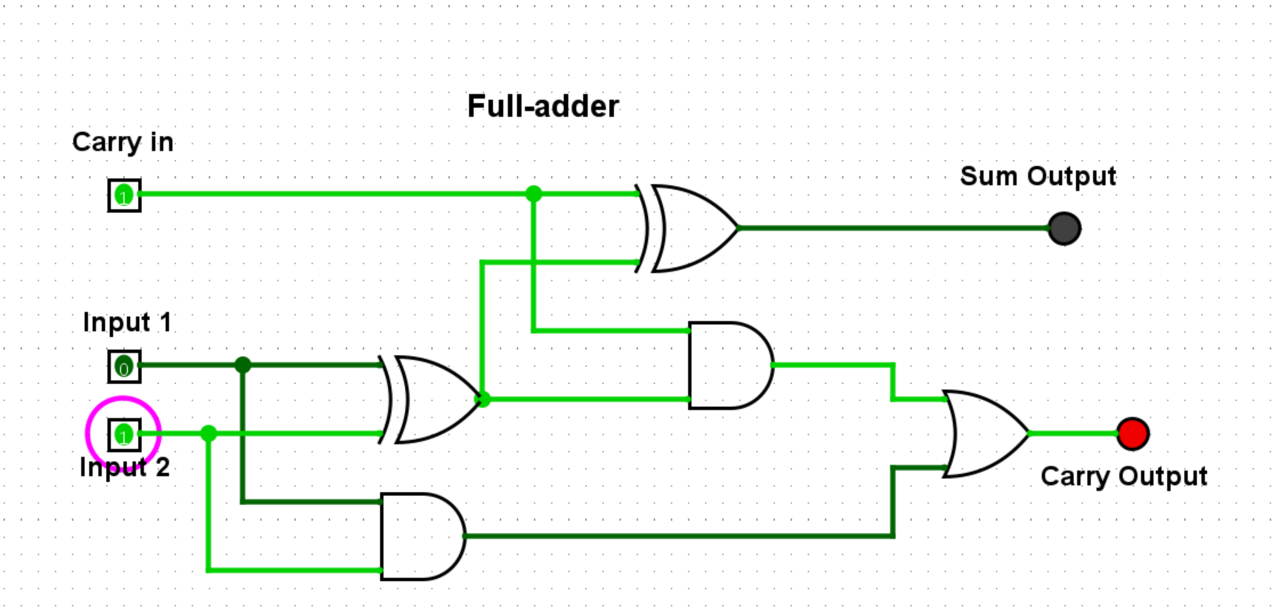


*Image 2.5: Input 1 = 0, Input 2 = 1, Cin = 1*

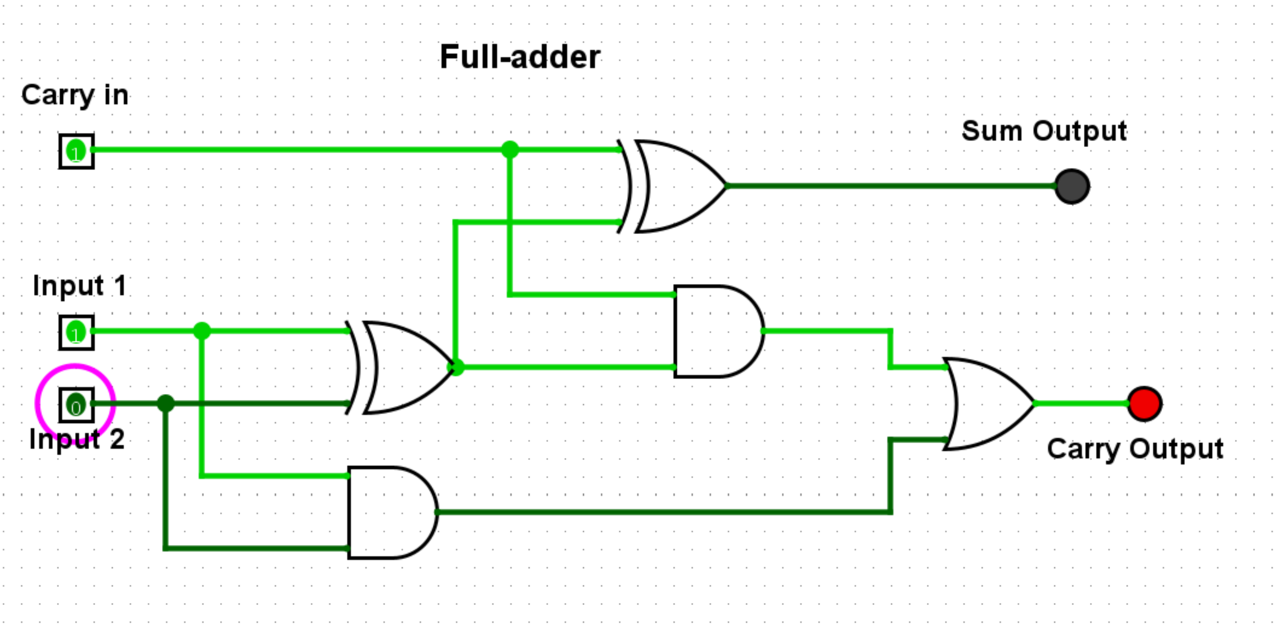


*Image 2.6: Input 1 = 1, Input 2 = 0, Cin = 1*

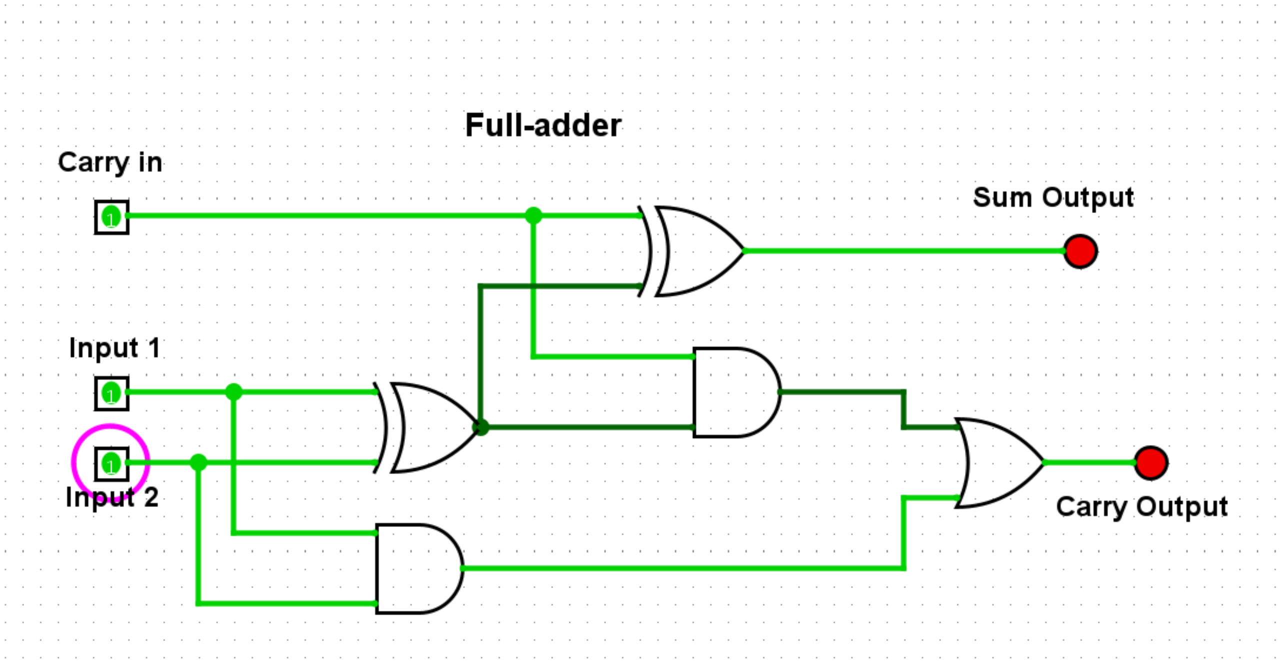


***Image 2.7**: Input = 1, Input 2 = 1, Cin = 1*

**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 | SumOutput | CarryOutput |
|---|---|---|---|---|
| 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.

**HINT:** for this task you might want to re-use your half-adder and full-adder from earlier!

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.
   6.2. While doing this, workout the order of your bits (*Big-endian or Little-endian?).* Use labels to indicate the significance of each bit (i.e., which column, from least significant ($2_0$ column) to most significant ($2_3$ column).
7. 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!
8. 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.

**Once complete, check its correctness by testing and filling out a truth table like the following (over page). Add the circuit screen shot and the table to your submission document:**

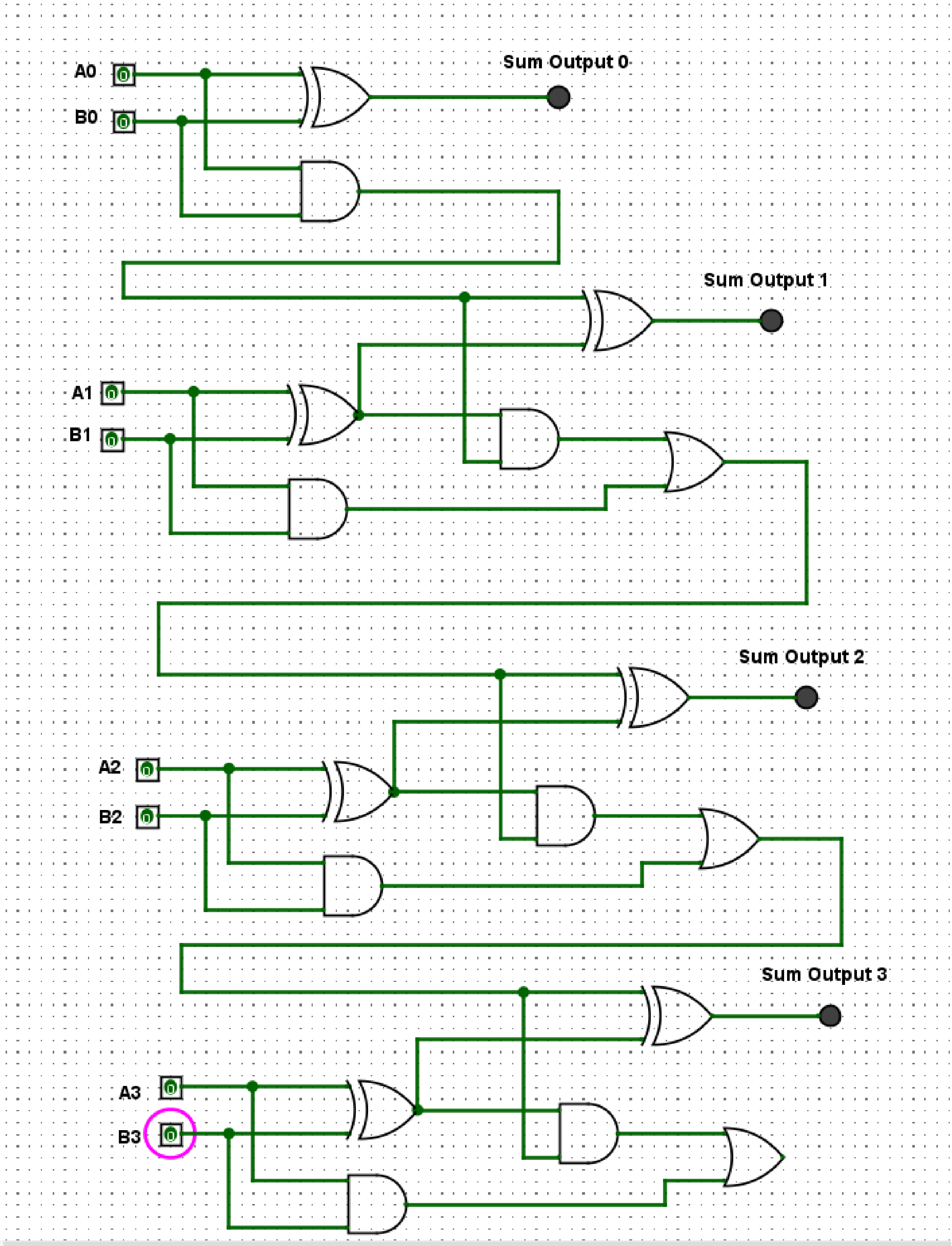


***Image 3: 4bits adder circut***

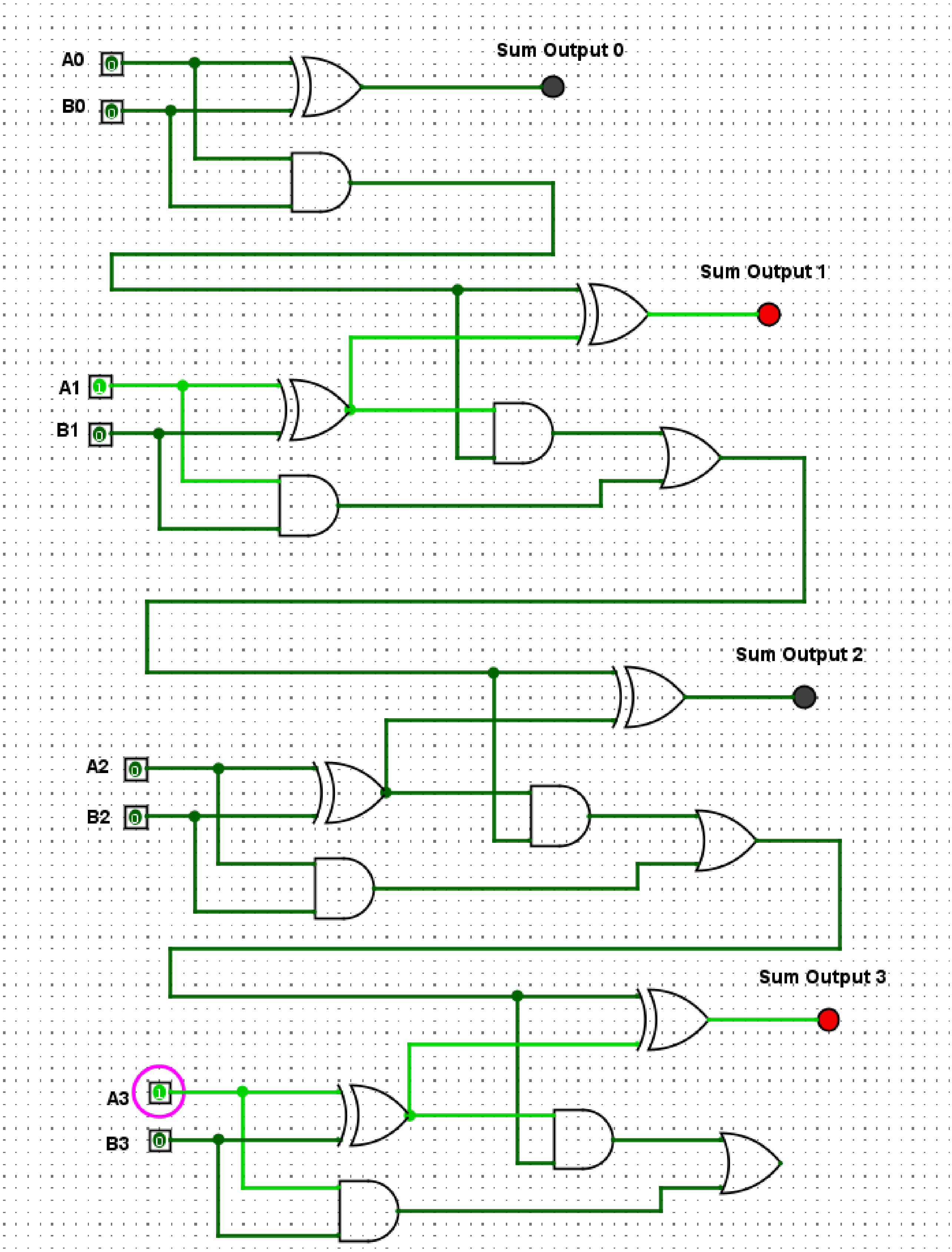


***Image 4**: Input A: 0101, Input B: 0000*

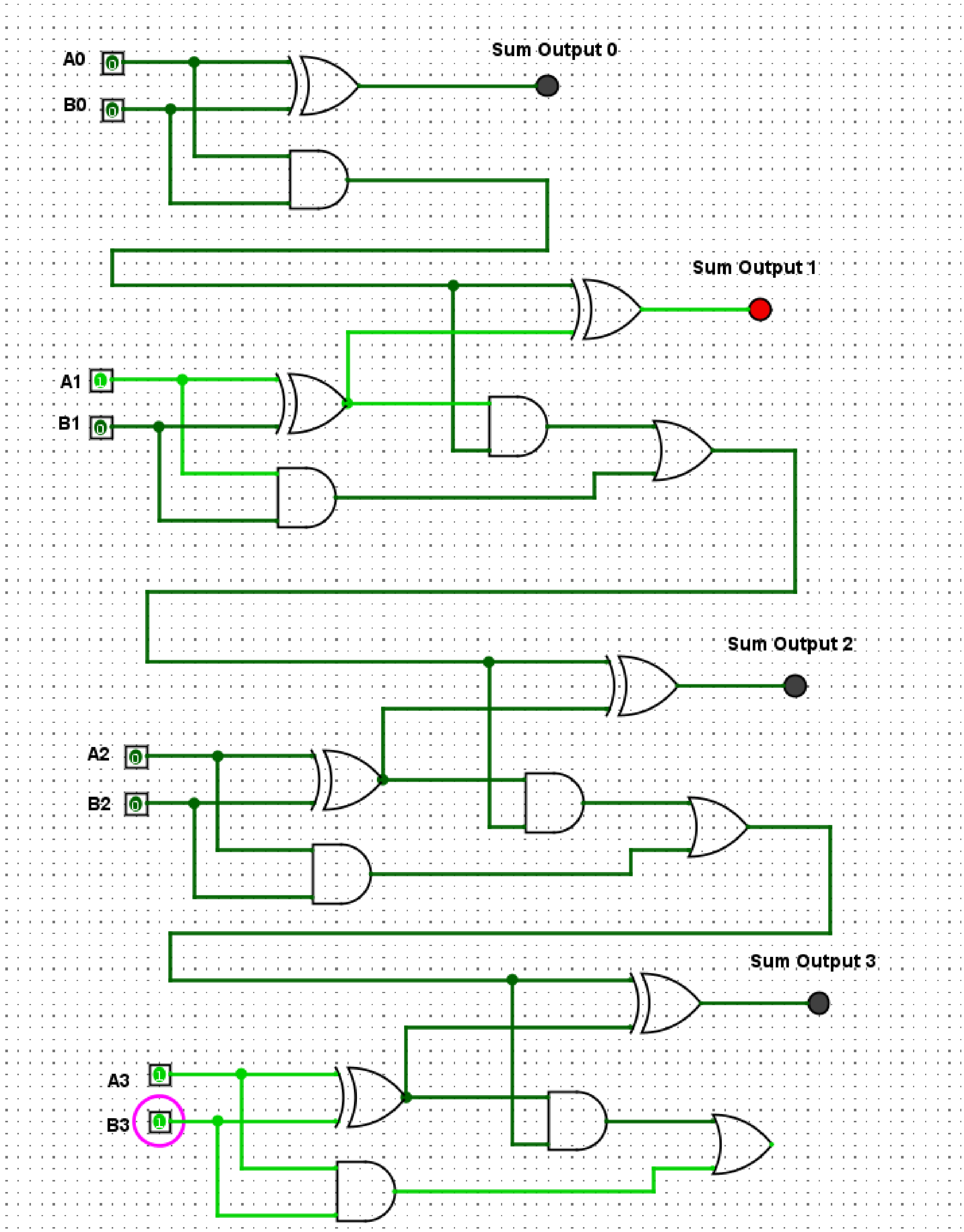


*Image 5: Input A: 0101, Input B: 0001*

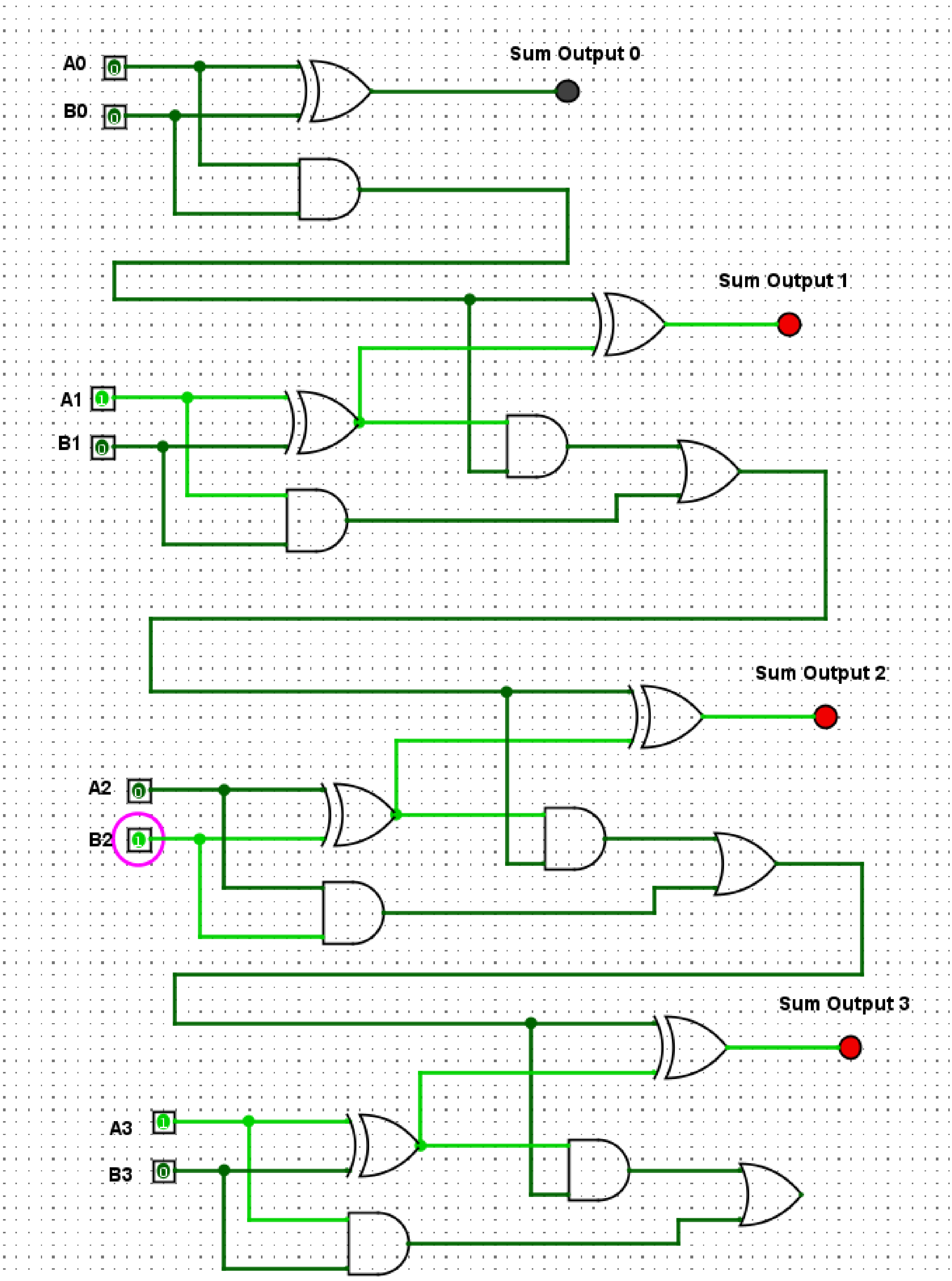


***Image 6*: Input A: 0101, Input B: 0010**

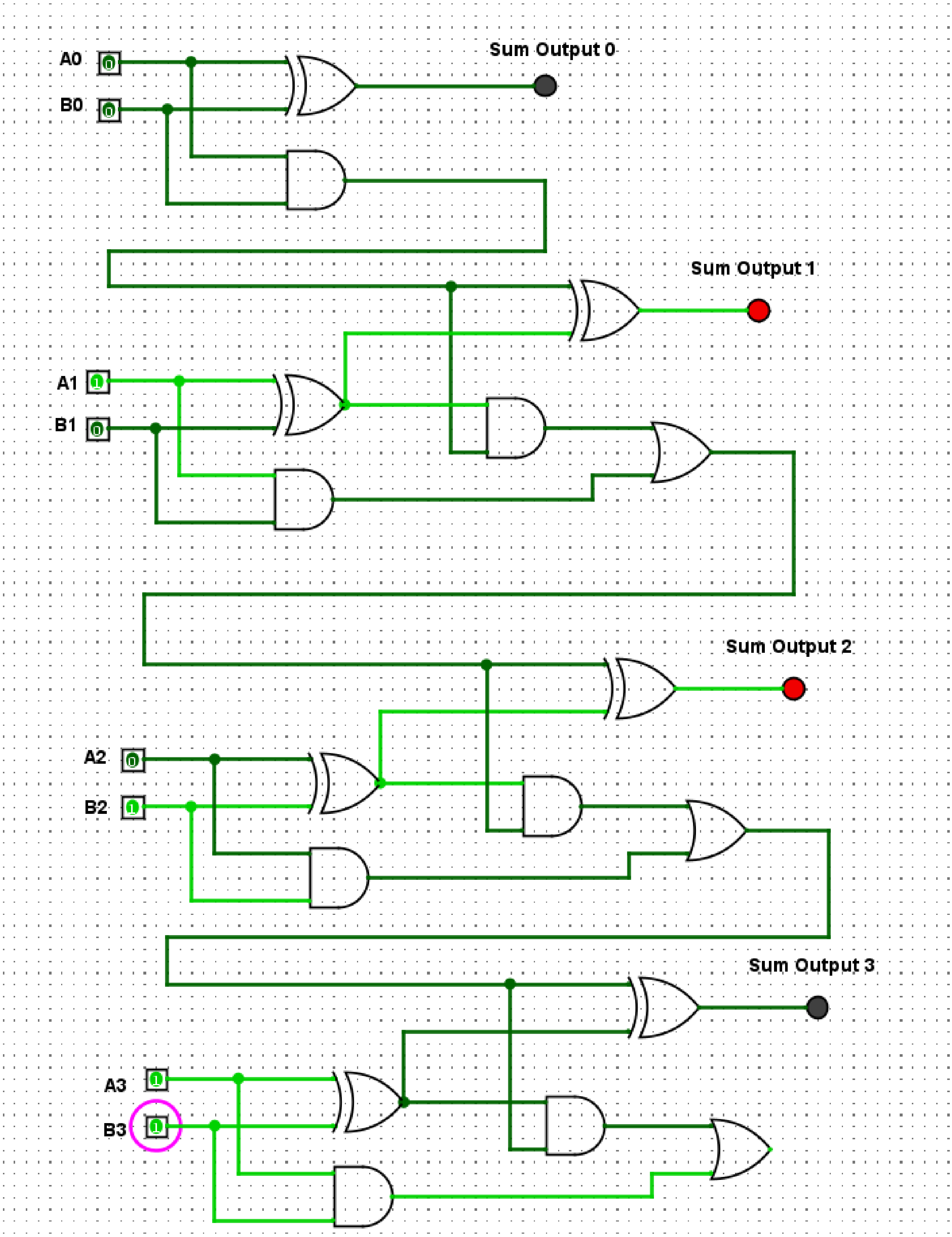


***Image 7**: Input A: 0101, Input B: 0011*

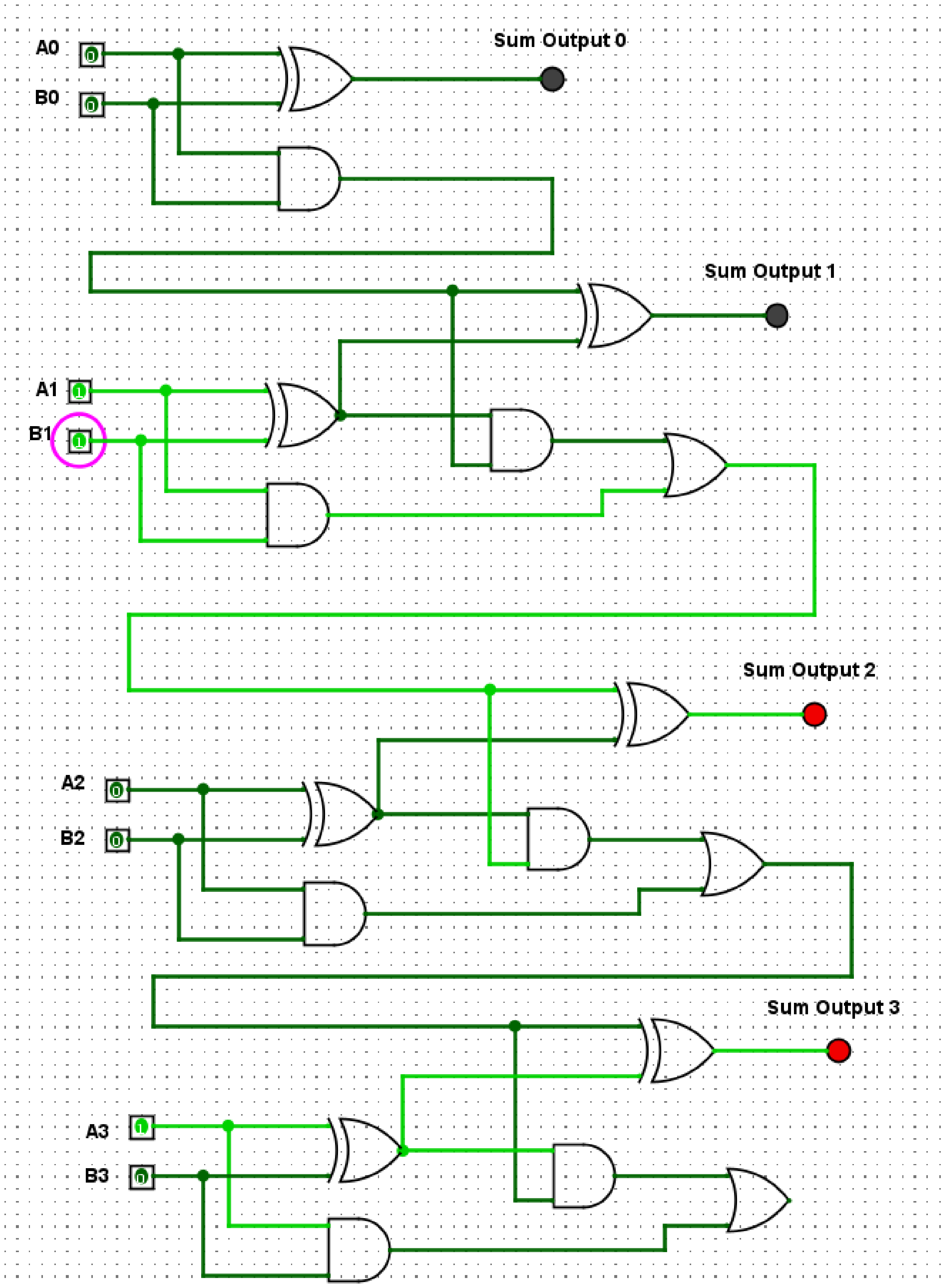


***Image 8: Input A: 0101, Input B: 0100***

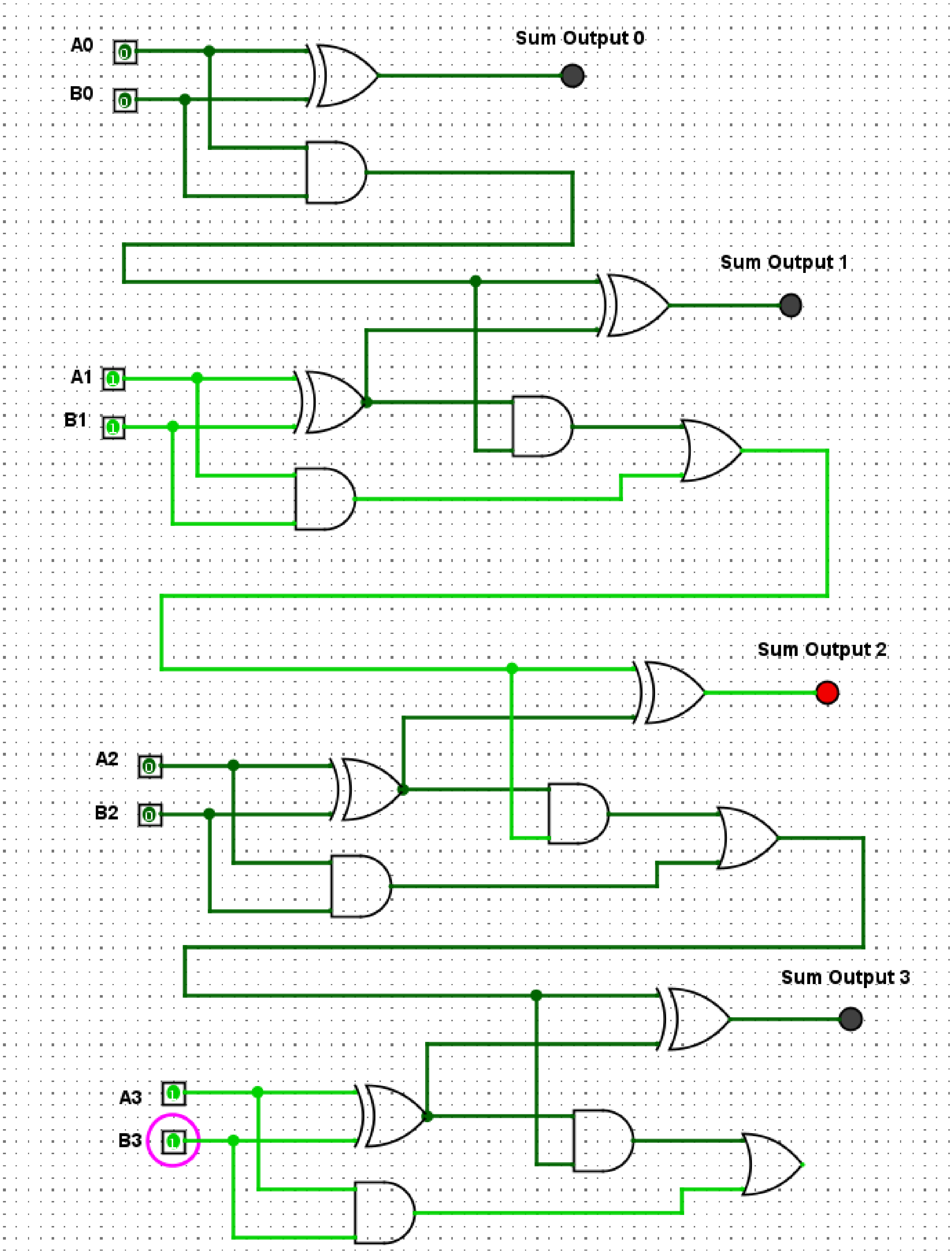


***Image 9**: Input A: 0101, Input B: 0101*

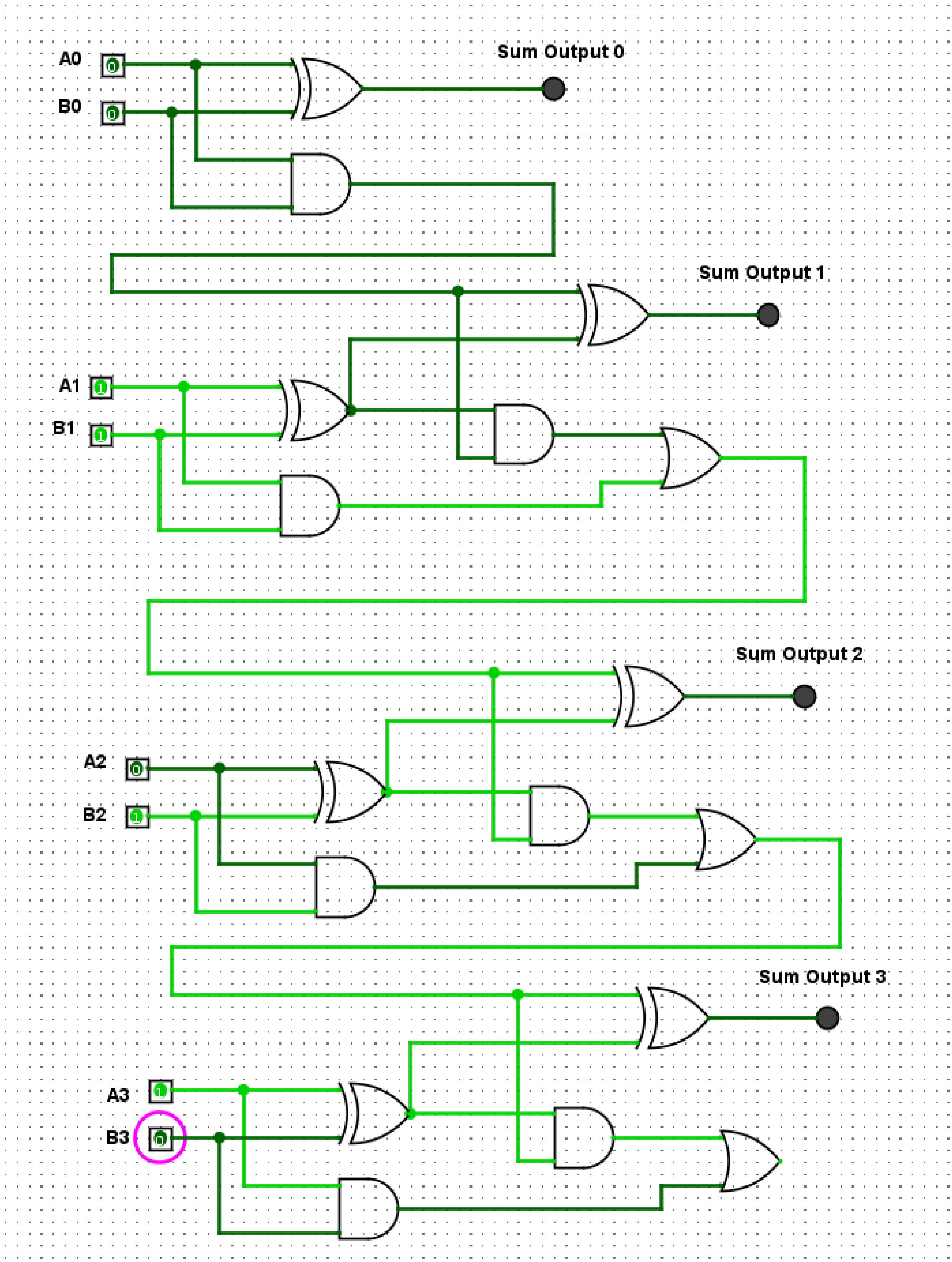


***Image 10**: Input A: 0101, Input B: 0110*

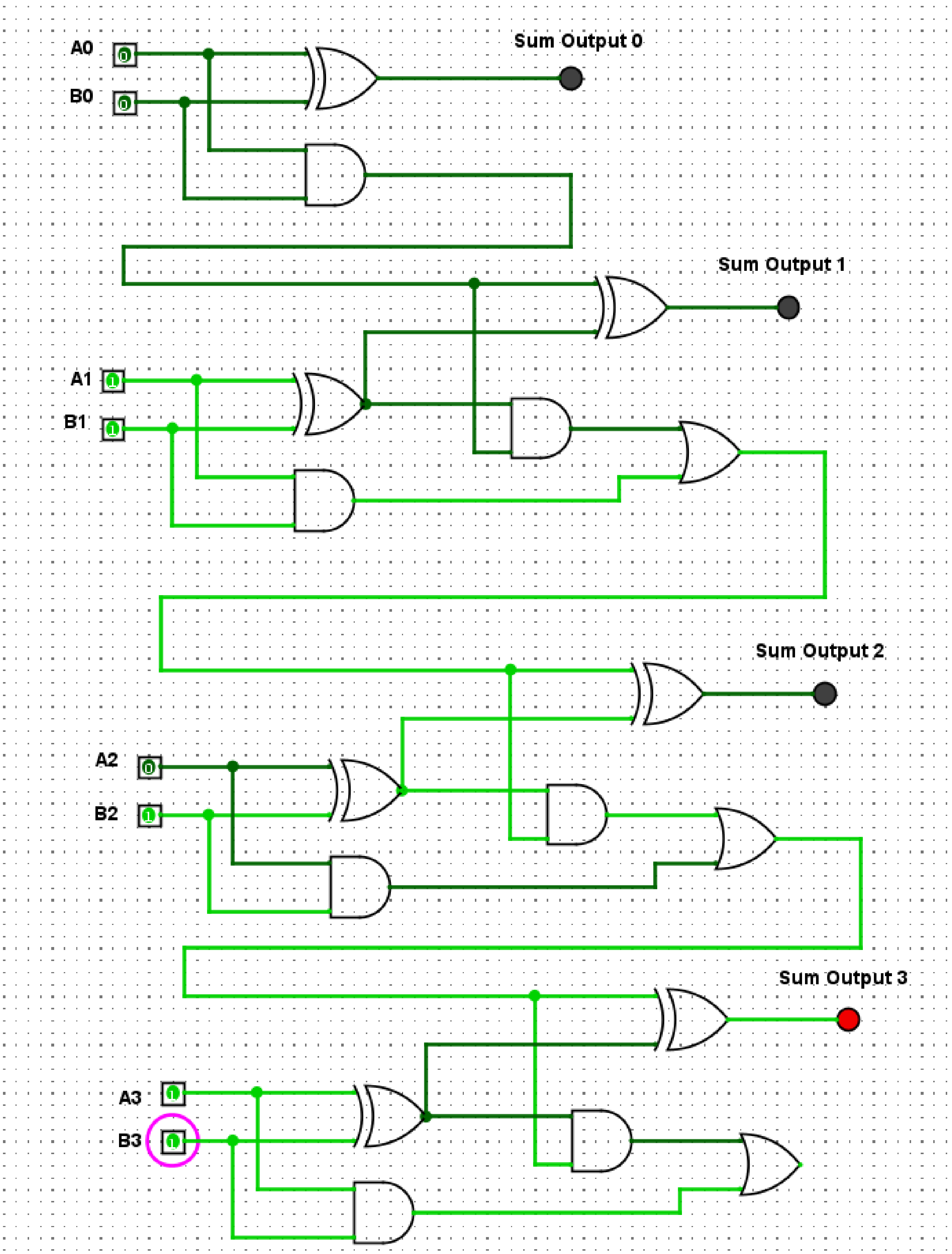


*Image 11: Input A: 0101, Input B: 0111*

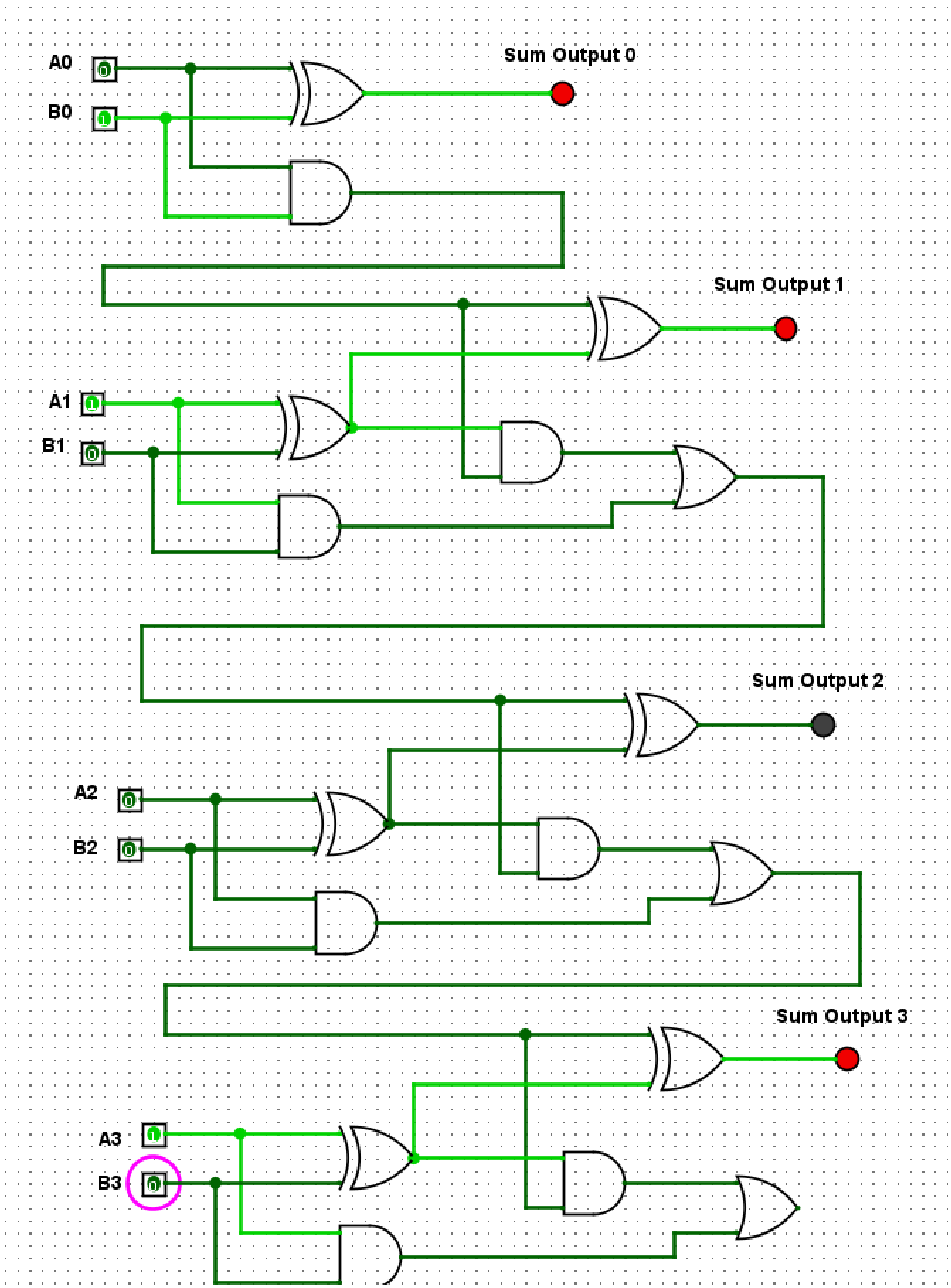


***Image 12**: Input A: 0101, Input B: 1000*

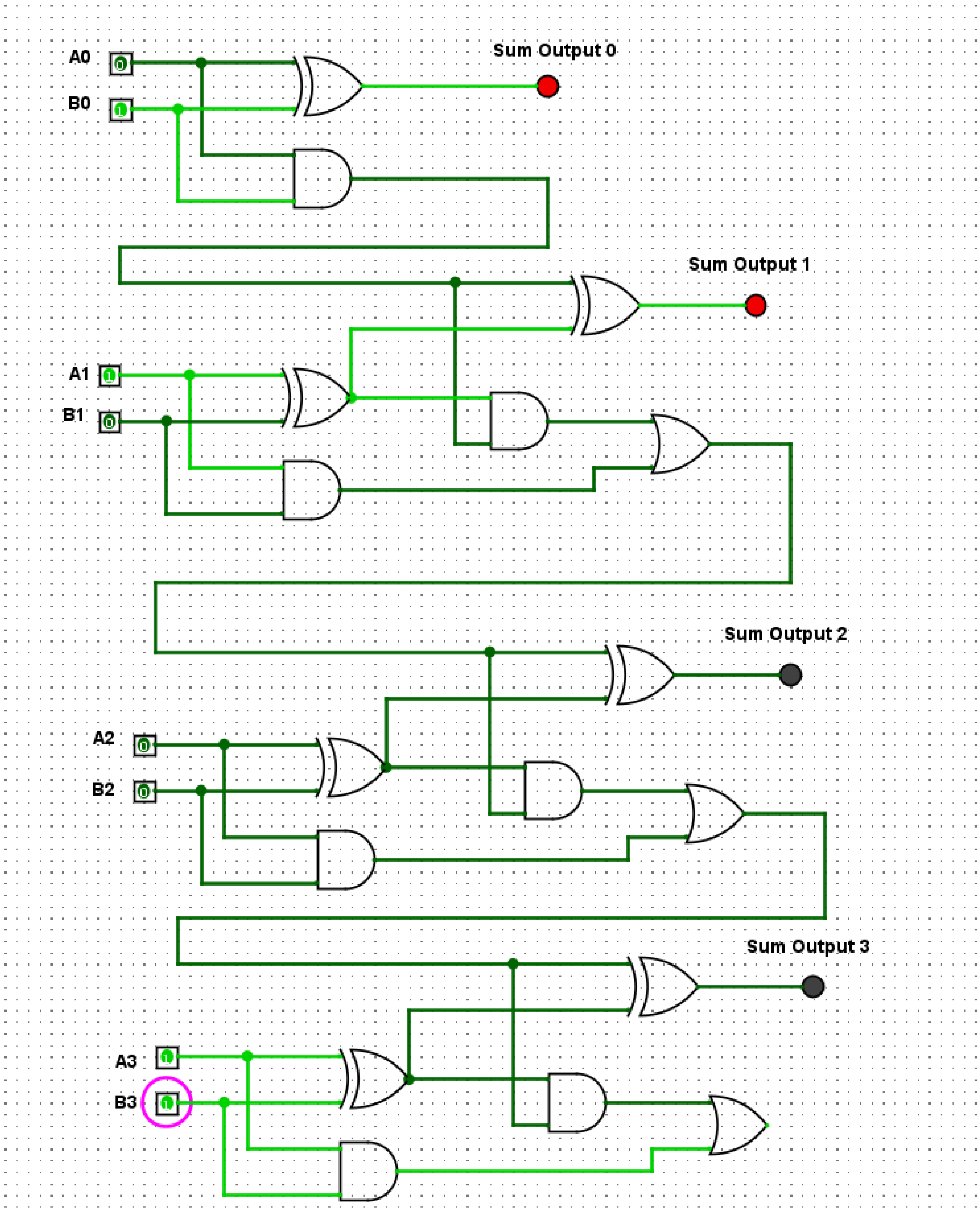


***Image 13**: Input A: 0101, Input B: 1001*

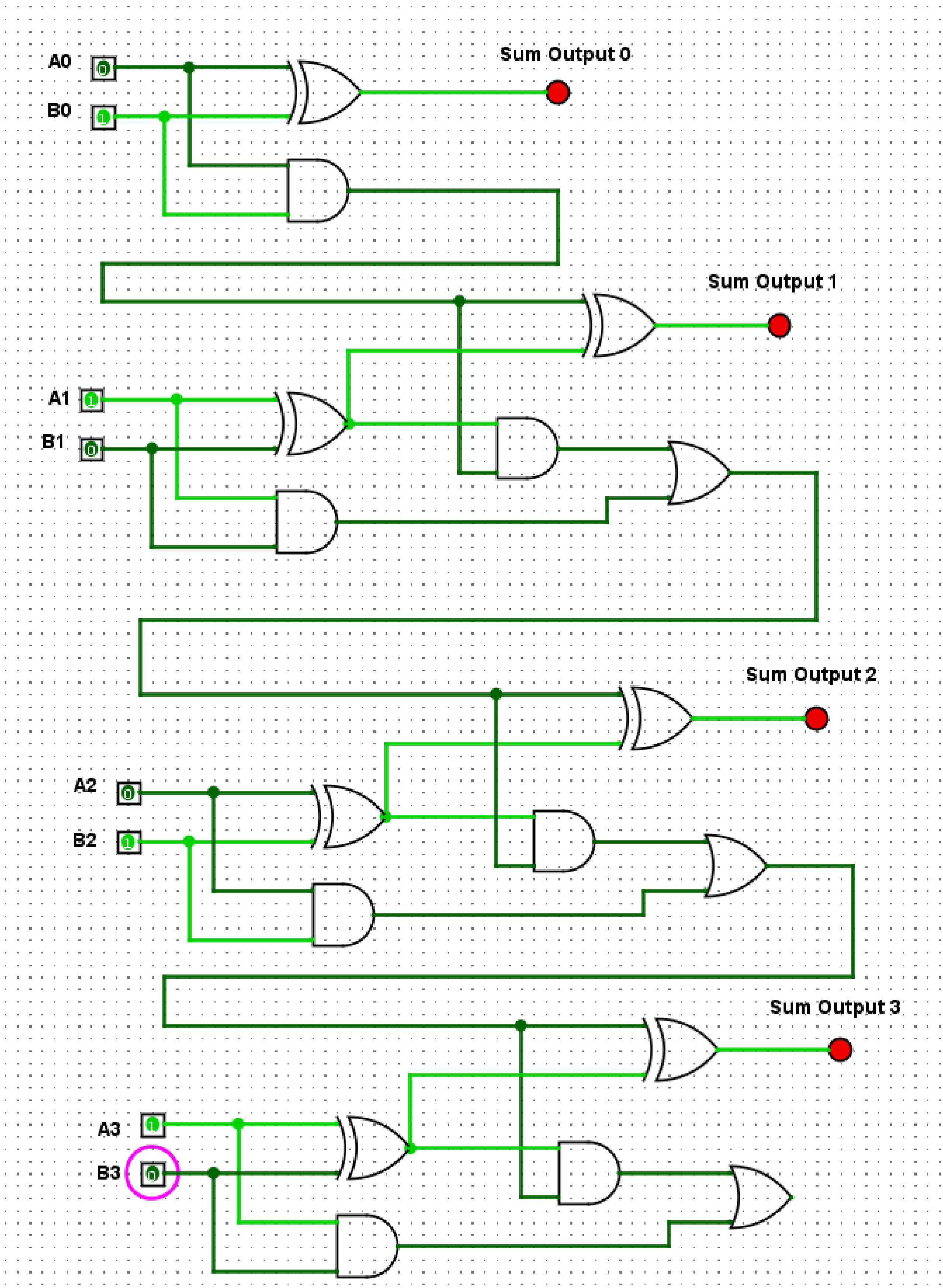


***Image 14**: Input A: 0101, Input B: 1010*

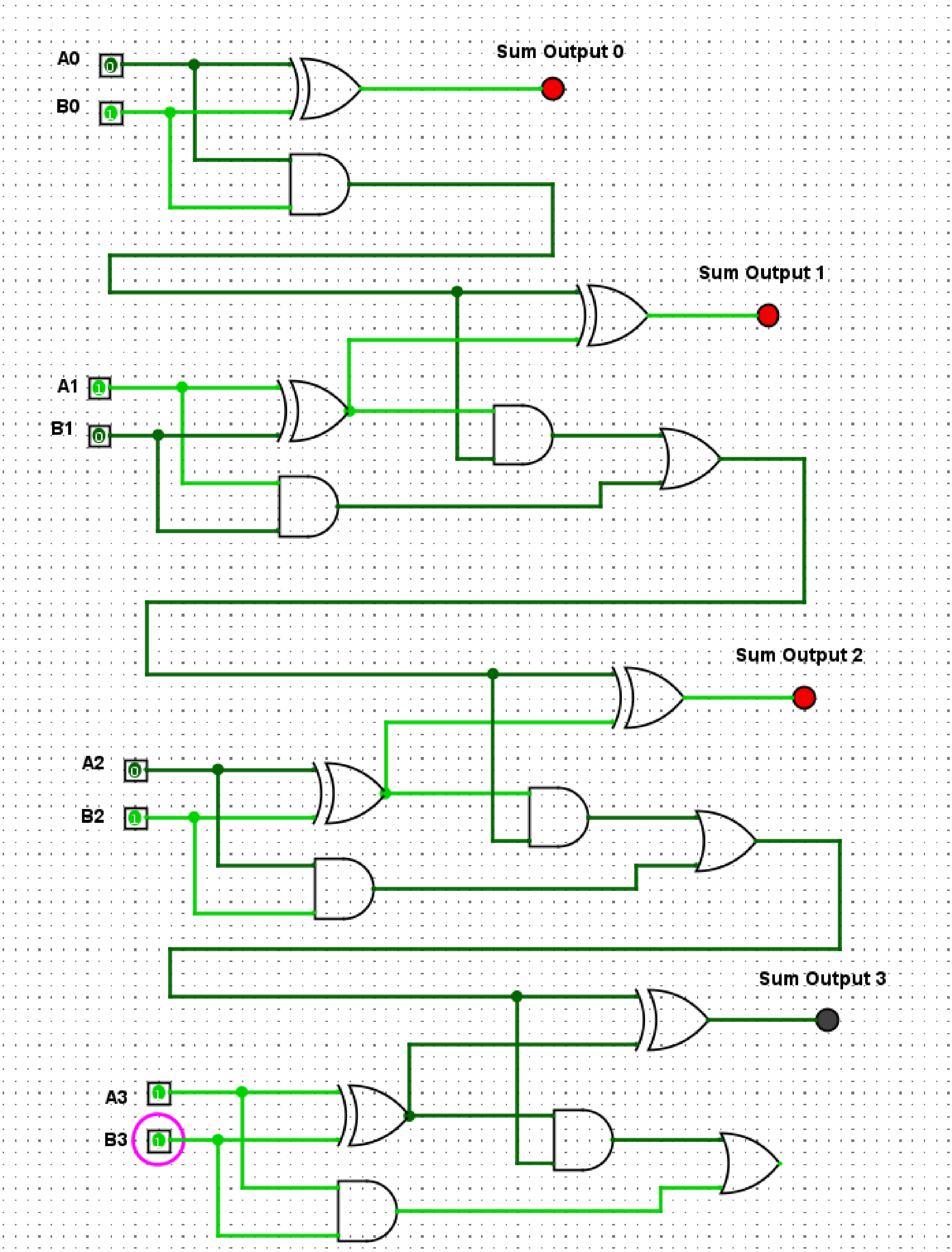


***Image 15*: Input A: 0101, Input B: 1011**

*z*

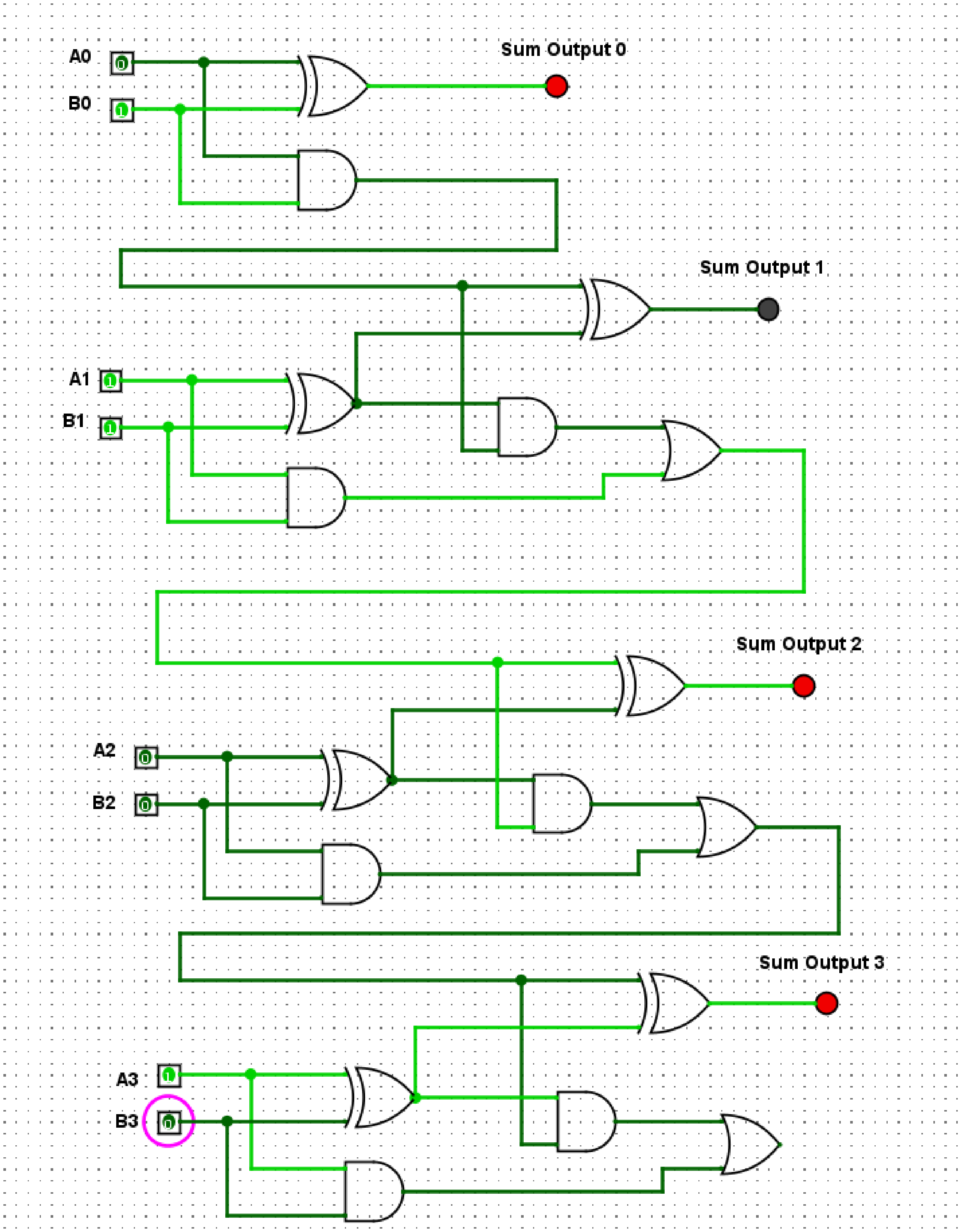


***Image 16**: Input A: 0101, Input B: 1100*

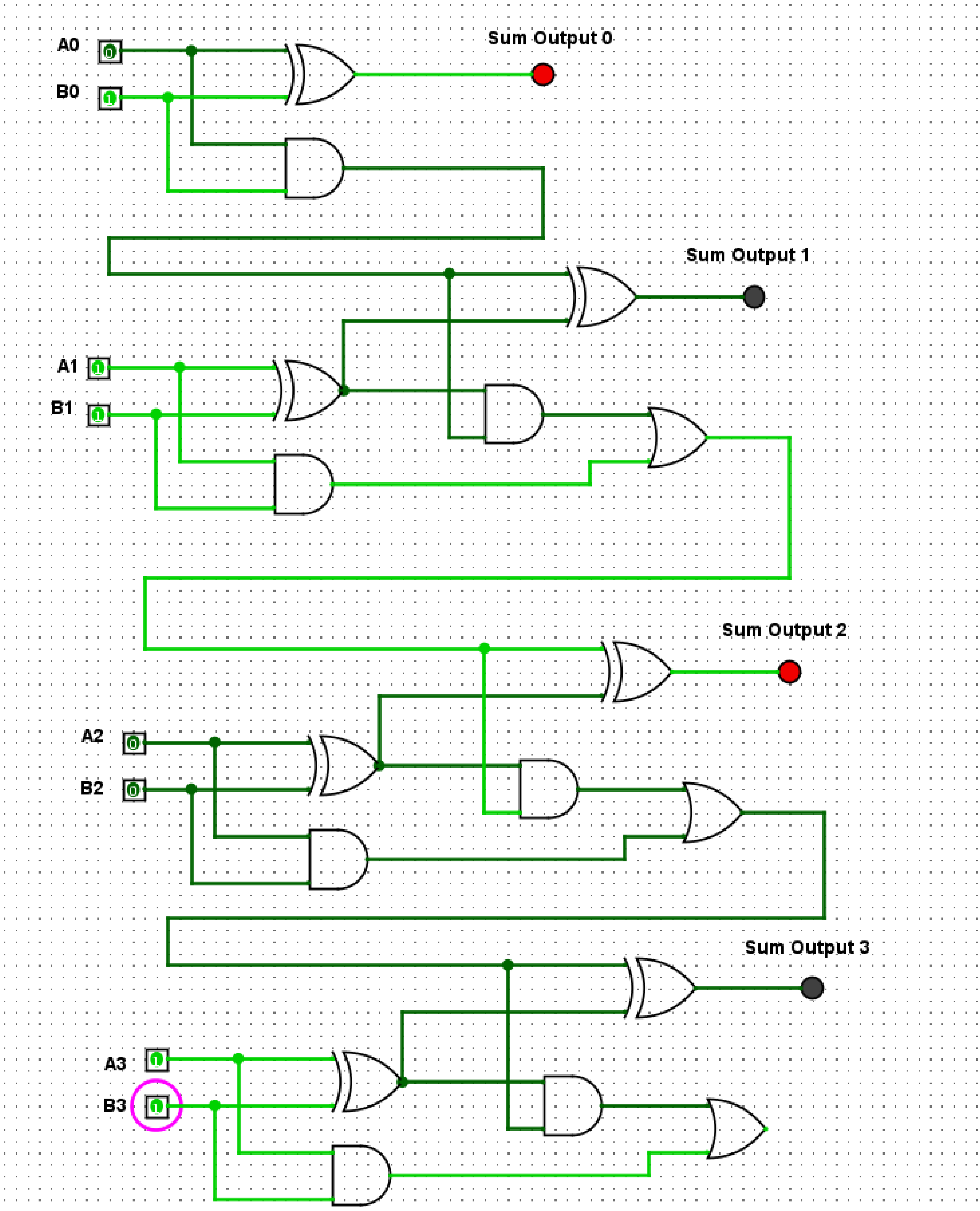


*Image 17: Input A: 0101, Input B: 1101*

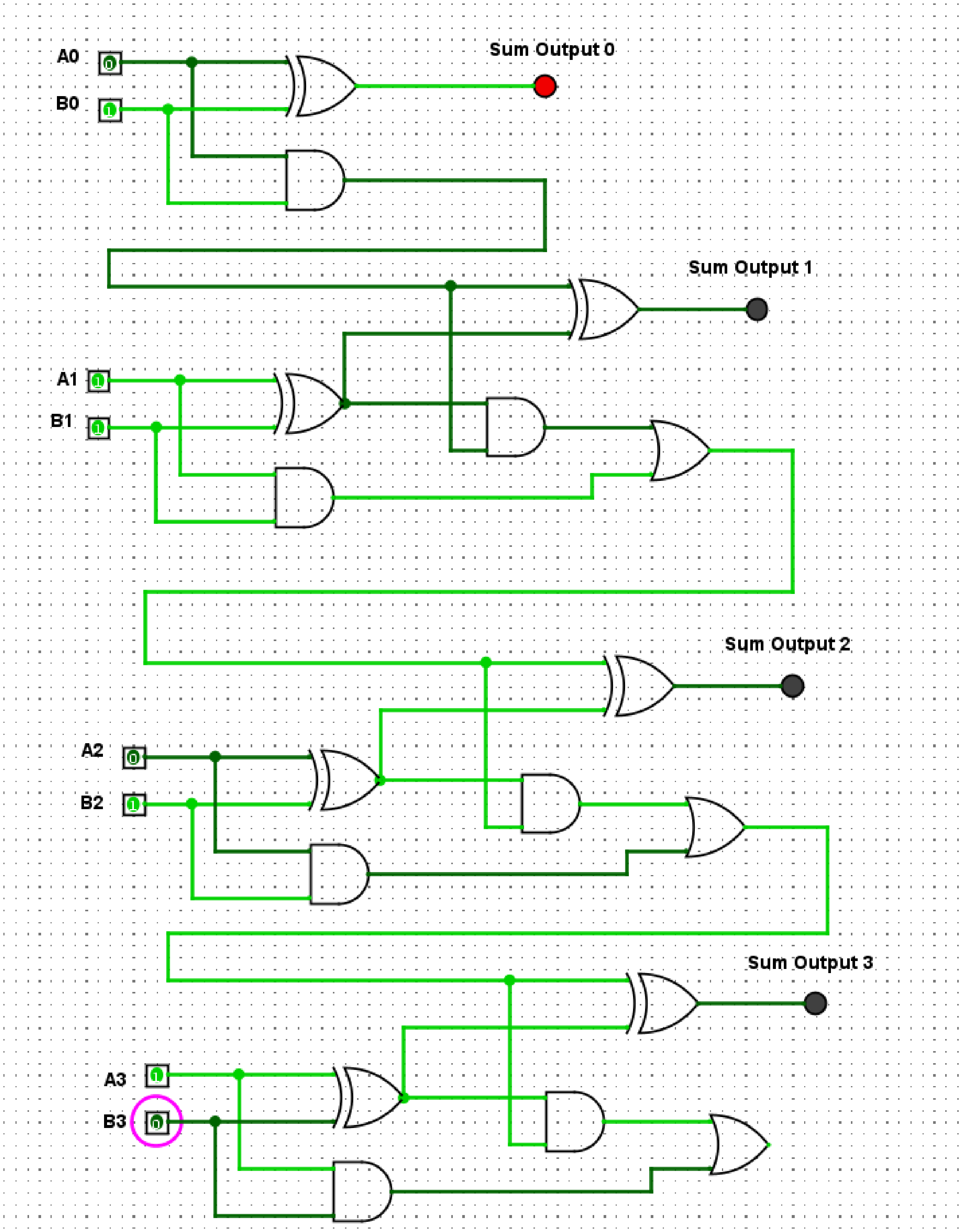


*Image 17: Input A: 0101, Input B: 1110*

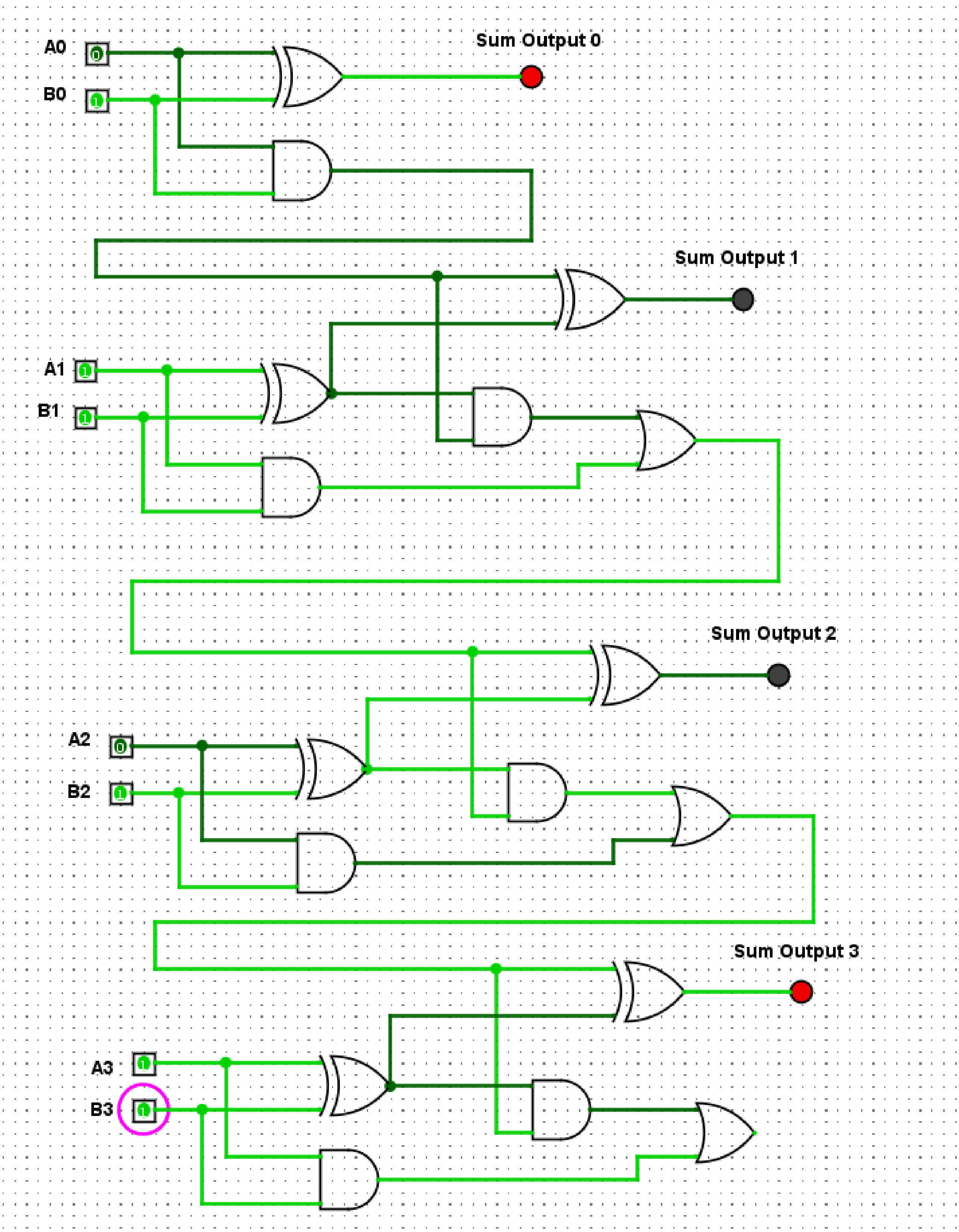


*Image 18: Input A: 0101, Input B: 1111*

| 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 |

**When complete:**

■ Submit your answers (screen shots, etc) in a single document using **Canvas**

■ Show your lab demonstrator your working circuits in class (you must do this to get the 1%).

Your lab demonstrator may request you to resubmit if issues exist.