

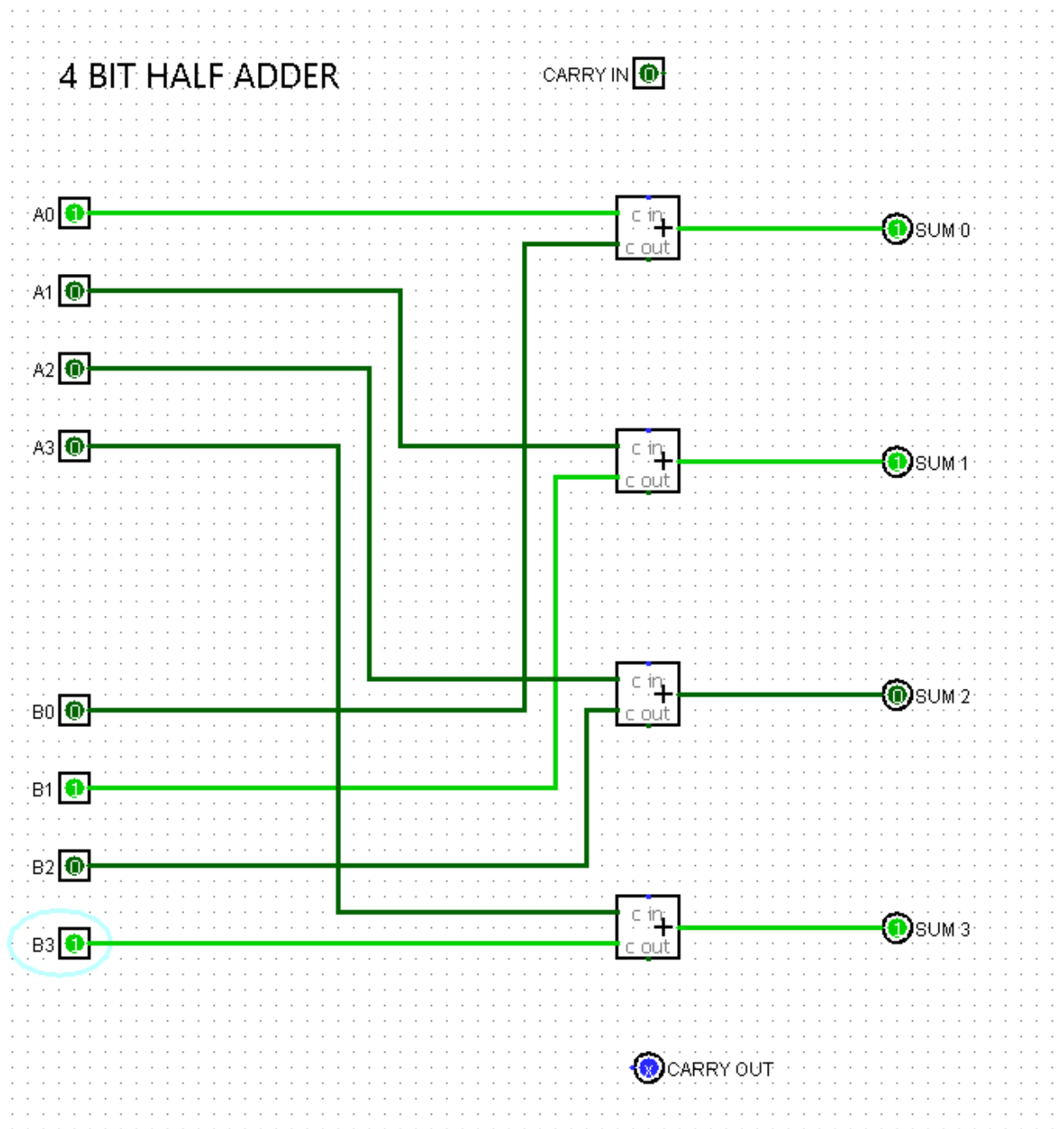
# ASSIGNMENT 4: LOGISIM

## U19CS012 [D-12]

Use Logisim software to create and store the followings circuits for further usage: (For Practice)

### 1. 4-bit half adder

Circuit Image:

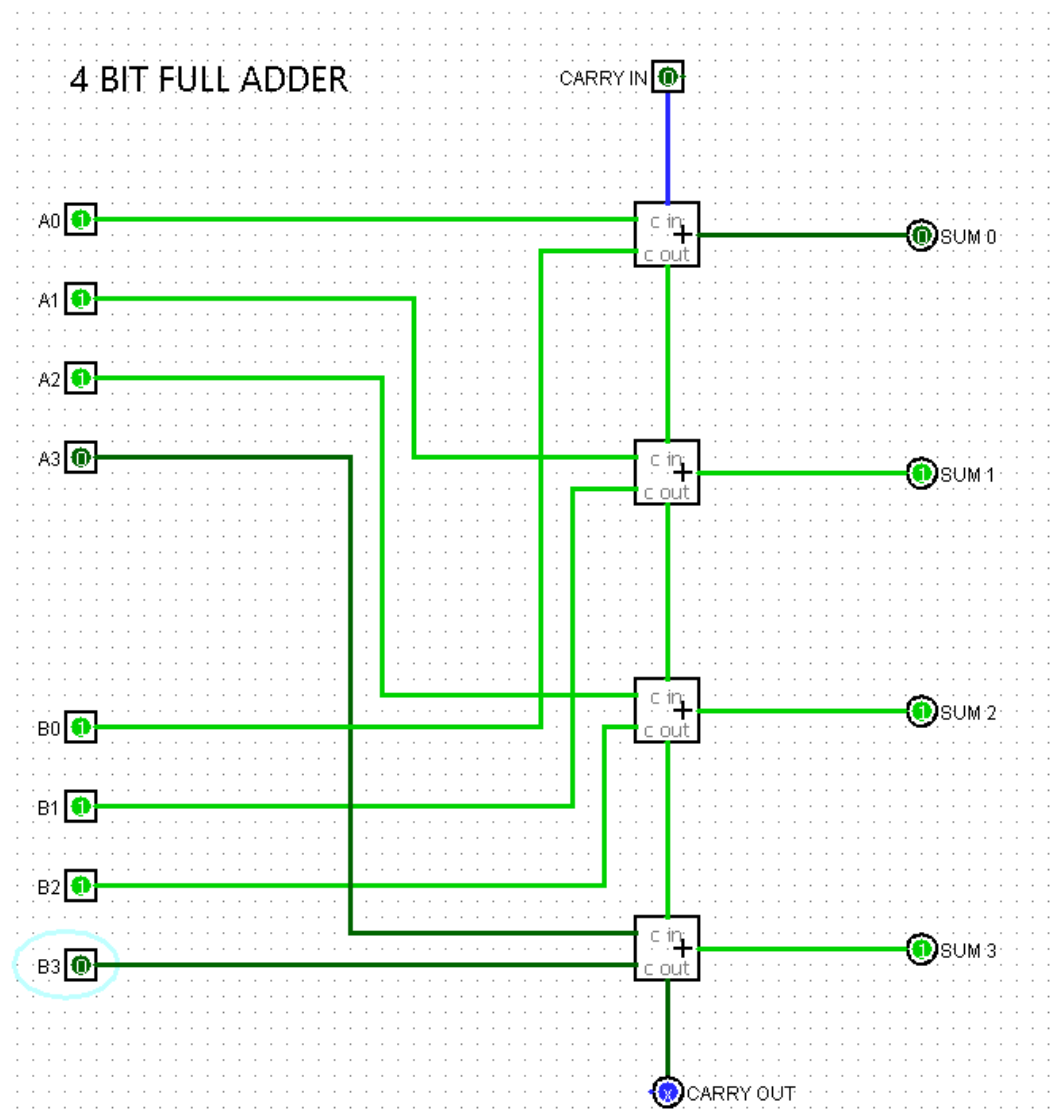


Truth Table:

CARRYIN	A0	A1	A2	A3	B0	B1	B2	B3	SUM0	SUM1	SUM2	SUM3	CARRYOUT
0	0	0	0	0	0	0	0	0	0	0	0	0	x
0	0	0	0	0	0	0	0	1	0	0	0	1	x
0	0	0	0	0	0	0	1	0	0	0	1	0	x
0	0	0	0	0	0	0	1	1	0	0	1	1	x
0	0	0	0	0	0	1	0	0	0	1	0	0	x
0	0	0	0	0	0	1	0	1	0	1	0	1	x
0	0	0	0	0	0	1	1	0	0	1	1	0	x
0	0	0	0	0	0	1	1	1	0	1	1	1	x
0	0	0	0	0	1	0	0	0	1	0	0	0	x
0	0	0	0	0	1	0	0	1	1	0	0	1	x
0	0	0	0	0	1	0	1	0	1	0	1	0	x
0	0	0	0	0	1	1	0	0	1	1	0	0	x
0	0	0	0	0	1	1	0	1	1	1	0	1	x
0	0	0	0	0	1	1	1	1	1	1	1	1	x
0	0	0	0	1	0	0	0	0	0	0	0	1	x
0	0	0	0	1	0	0	0	1	0	0	0	0	x
0	0	0	0	1	0	0	0	1	0	0	0	1	x
0	0	0	0	1	0	0	1	0	0	0	1	1	x
0	0	0	0	1	0	1	0	1	0	1	0	0	x
0	0	0	0	1	0	1	1	0	0	1	1	1	x
0	0	0	0	1	0	1	1	1	0	1	1	0	x

## 2. 4-bit full adder

Circuit Image:



## Truth Table:

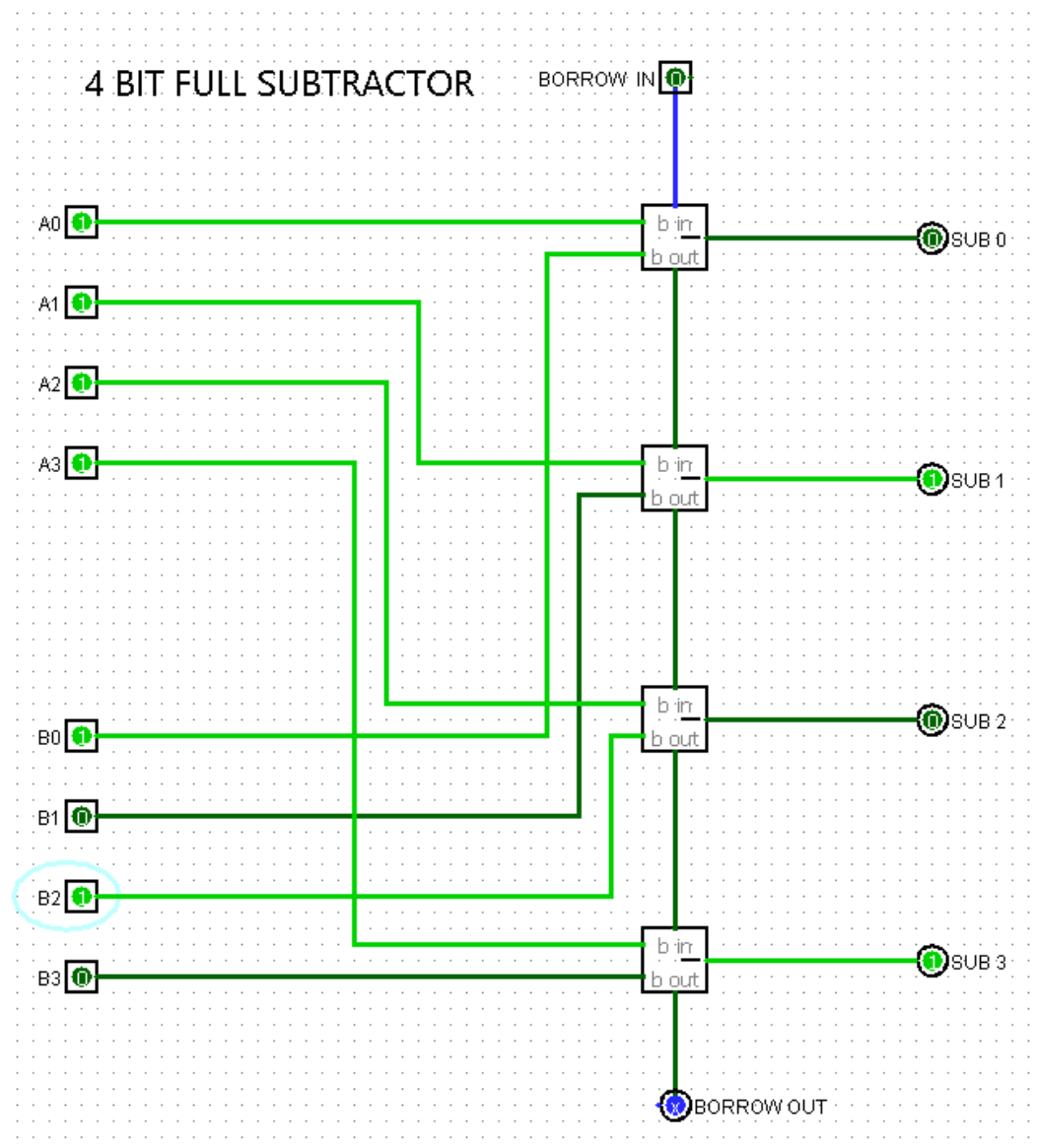
CARRYIN	A0	A1	A2	A3	B0	B1	B2	B3	SUM0	SUM1	SUM2	SUM3	CARRYOUT
0	0	0	0	0	0	0	0	0	0	0	0	0	x
0	0	0	0	0	0	0	0	1	0	0	0	1	x
0	0	0	0	0	0	0	1	0	0	0	1	0	x
0	0	0	0	0	0	0	1	1	0	0	1	1	x
0	0	0	0	0	0	1	0	0	0	1	0	0	x
0	0	0	0	0	0	1	0	1	0	1	0	1	x
0	0	0	0	0	0	1	1	0	0	1	1	0	x
0	0	0	0	0	0	1	1	1	0	1	1	1	x
0	0	0	0	0	1	0	0	0	1	0	0	0	x
0	0	0	0	0	1	0	0	1	1	0	0	1	x
0	0	0	0	0	1	0	1	0	1	0	1	0	x
0	0	0	0	0	1	0	1	1	1	0	1	1	x
0	0	0	0	0	1	1	0	0	1	1	0	0	x
0	0	0	0	0	1	1	0	1	1	1	0	1	x
0	0	0	0	0	1	1	1	0	1	1	1	0	x
0	0	0	0	0	1	1	1	1	1	1	1	1	x
0	0	0	0	1	0	0	0	0	0	0	0	1	x
0	0	0	0	1	0	0	0	1	0	0	1	0	x
0	0	0	0	1	0	0	1	0	0	0	1	1	x
0	0	0	0	1	0	1	0	0	0	1	0	0	x
0	0	0	0	1	0	1	1	0	0	1	0	1	x
0	0	0	0	1	0	1	1	1	0	1	0	1	x
0	0	0	1	0	0	0	0	0	0	0	0	0	x
0	0	0	1	0	0	0	1	0	0	0	1	0	x
0	0	0	1	0	0	1	0	0	0	1	1	0	x
0	0	0	1	0	0	1	1	0	0	1	1	1	x
0	0	0	1	0	1	0	0	0	0	0	0	0	x
0	0	0	1	0	1	0	1	0	0	0	1	0	x
0	0	0	1	0	1	1	0	0	0	1	1	1	x
0	0	0	1	0	1	1	1	0	0	1	1	1	x
0	0	1	0	0	0	0	0	0	0	0	0	0	x
0	0	1	0	0	0	0	1	0	0	0	1	0	x
0	0	1	0	0	0	1	0	0	0	1	1	1	x
0	0	1	0	0	1	0	0	0	0	0	0	0	x
0	0	1	0	0	1	0	1	0	0	0	1	0	x
0	0	1	0	0	1	1	0	0	0	0	1	1	x
0	0	1	0	0	1	1	1	0	0	0	1	1	x
0	0	1	1	0	0	0	0	0	0	0	0	0	x
0	0	1	1	0	0	0	1	0	0	0	1	0	x
0	0	1	1	0	0	1	0	0	0	0	1	1	x
0	0	1	1	0	0	1	1	0	0	0	1	1	x
0	0	1	1	1	0	0	0	0	0	0	0	0	x
0	0	1	1	1	0	0	1	0	0	0	1	0	x
0	0	1	1	1	0	1	0	0	0	0	1	1	x
0	0	1	1	1	0	1	1	0	0	0	1	1	x
0	0	1	1	1	1	0	0	0	0	0	0	0	x
0	0	1	1	1	1	0	1	0	0	0	1	0	x
0	0	1	1	1	1	1	0	0	0	0	1	1	x
0	0	1	1	1	1	1	1	0	0	0	1	1	x
0	1	0	0	0	0	0	0	0	0	0	0	0	x
0	1	0	0	0	0	0	0	1	0	0	0	1	x
0	1	0	0	0	0	0	1	1	0	0	0	1	x
0	1	0	0	0	0	1	1	1	0	0	0	1	x
0	1	0	0	1	0	0	0	0	0	0	0	0	x
0	1	0	0	1	0	0	0	1	0	0	0	1	x
0	1	0	0	1	0	0	1	1	0	0	0	1	x
0	1	0	1	0	0	0	0	0	0	0	0	0	x
0	1	0	1	0	0	0	1	0	0	0	0	1	x
0	1	0	1	0	0	1	0	0	0	0	0	1	x
0	1	0	1	0	0	1	1	0	0	0	0	1	x
0	1	0	1	1	0	0	0	0	0	0	0	0	x
0	1	0	1	1	0	0	1	0	0	0	0	1	x
0	1	0	1	1	0	1	0	0	0	0	0	1	x
0	1	0	1	1	0	1	1	0	0	0	0	1	x
0	1	1	0	0	0	0	0	0	0	0	0	0	x
0	1	1	0	0	0	0	0	1	0	0	0	1	x
0	1	1	0	0	0	1	0	0	0	0	0	1	x
0	1	1	0	0	0	1	1	0	0	0	0	1	x
0	1	1	0	1	0	0	0	0	0	0	0	0	x
0	1	1	0	1	0	0	1	0	0	0	0	1	x
0	1	1	0	1	0	1	1	0	0	0	0	1	x
0	1	1	1	0	0	0	0	0	0	0	0	0	x
0	1	1	1	0	0	0	1	0	0	0	0	1	x
0	1	1	1	0	0	1	0	0	0	0	0	1	x
0	1	1	1	0	0	1	1	0	0	0	0	1	x
0	1	1	1	1	0	0	0	0	0	0	0	0	x
0	1	1	1	1	0	0	1	0	0	0	0	1	x
0	1	1	1	1	0	1	0	0	0	0	0	1	x
0	1	1	1	1	0	1	1	0	0	0	0	1	x
0	1	1	1	1	1	0	0	0	0	0	0	0	x
0	1	1	1	1	1	0	1	0	0	0	0	1	x
0	1	1	1	1	1	1	0	0	0	0	0	1	x
0	1	1	1	1	1	1	1	0	0	0	0	1	x
1	0	0	0	0	0	0	0	0	1	1	1	1	0
1	0	0	0	0	0	0	0	1	1	1	1	1	0
1	0	0	0	0	0	0	1	0	1	1	1	1	0
1	0	0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	0	0	1	0	0	1	1	1	1	0
1	0	0	0	0	0	1	0	1	1	1	1	1	0
1	0	0	0	0	0	1	1	0	1	1	1	1	0
1	0	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	0	1	0	0	0	1	1	1	1	0
1	0	0	0	0	1	0	0	1	1	1	1	1	0
1	0	0	0	0	1	0	1	0	1	1	1	1	0
1	0	0	0	0	1	0	1	1	1	1	1	1	0
1	0	0	0	0	1	1	0	0	1	1	1	1	0
1	0	0	0	0	1	1	0	1	1	1	1	1	0
1	0	0	0	0	1	1	1	0	1	1	1	1	0
1	0	0	0	0	1	1	1	1	1	1	1	1	0
1	0	0	0	1	0	0	0	0	1	1	1	1	0
1	0	0	0	1	0	0	0	1	1	1	1	1	0
1	0	0	0	1	0	0	1	0	1	1	1	1	0
1	0	0	0	1	0	0	1	1	1	1	1	1	0
1	0	0	0	1	1	0	0	0	1	1	1	1	0
1	0	0	0	1	1	0	0	1	1	1	1	1	0
1	0	0	0	1	1	0	1	0	1	1	1	1	0
1	0	0	0	1	1	0	1	1	1	1	1	1	0
1	0	0	0	1	1	1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	0	1	1	1	1	1	0
1	0	0	0	1	1	1	1	0	1	1	1	1	0
1	0	0	0	1	1	1	1	1	1	1	1	1	0
1	0	0	1	0	0	0	0	0	1	1	1	1	0
1	0	0	1	0	0	0	0	1	1	1	1	1	0
1	0	0	1	0	0	0	1	0	1	1	1	1	0
1	0	0	1	0	0	0	1	1	1	1	1	1	0
1	0	0	1	0	0	1	0	0	1	1	1	1	0
1	0	0	1	0	0	1	0	1	1	1	1	1	0
1	0	0	1	0	0	1	1	0	1	1	1	1	0
1	0	0	1	0	0	1	1	1	1	1	1	1	0
1	0	0	1	1	0	0	0	0	1	1	1	1	0
1	0	0	1	1	0	0	0	1	1	1	1	1	0
1	0	0	1	1	0	0	1	0	1	1	1	1	0
1	0	0	1	1	0	0	1	1	1	1	1	1	0
1	0	0	1	1	1	0	0	0	1	1	1	1	0
1	0	0	1	1	0	0	0	1	1	1	1	1	0
1	0	0	1	1	0	0	1	0	1	1	1	1	0
1	0	0	1	1	0	0	1	1	1	1	1	1	0
1	0	0	1	1	1	0	0	0	1	1	1	1	0
1	0	0	1	1	0	0	0	1	1	1	1	1	0
1	0	0	1	1	0	0	1	0	1	1	1	1	0
1	0	0	1	1	0	0	1	1	1	1	1	1	0
1	0	0	1	1	1	0	0	0	1	1	1	1	0
1	0	0	1	1	0	0	0	1	1	1	1	1	0
1	0	0	1	1	0	0	1	0	1	1	1	1	0
1	0	0	1	1	0	0	1	1	1	1	1	1	0
1	0	0	1	1	1	0	0	0	1	1	1	1	0
1	0	0	1	1	0	0	0	1	1	1	1	1	0
1	0	0	1	1	0	0	1	0	1	1	1	1	0
1	0	0											

Truth Table:

BORROWIN	A0	A1	A2	A3	B0	B1	B2	B3	SUB0	SUB1	SUB2	SUB3	BORROWOUT
0	0	0	0	0	0	0	0	0	0	0	0	0	x
0	0	0	0	0	0	0	0	1	0	0	0	1	x
0	0	0	0	0	0	0	1	0	0	0	1	0	x
0	0	0	0	0	0	0	1	1	0	0	1	1	x
0	0	0	0	0	0	1	0	0	0	1	0	0	x
0	0	0	0	0	0	1	0	1	0	1	0	1	x
0	0	0	0	0	0	1	1	0	0	1	1	0	x
0	0	0	0	0	0	1	1	1	0	1	1	1	x
0	0	0	0	0	1	0	0	0	1	0	0	0	x
0	0	0	0	0	1	0	0	1	1	0	0	1	x
0	0	0	0	0	1	0	1	0	1	0	1	0	x
0	0	0	0	0	1	1	0	0	1	1	0	1	x
0	0	0	0	0	1	1	1	0	1	1	1	0	x
0	0	0	0	0	1	1	1	1	1	1	1	1	x
0	0	0	0	1	0	0	0	0	0	0	0	1	x
0	0	0	0	1	0	0	0	1	0	0	0	0	x
0	0	0	0	1	0	0	1	0	0	0	1	1	x
0	0	0	0	1	0	0	1	1	0	0	1	0	x
0	0	0	0	1	0	1	0	0	0	1	0	1	x
0	0	0	0	1	0	1	1	0	0	1	1	1	x
0	0	0	0	1	0	1	1	1	0	1	1	0	x

#### 4. 4-bit full subtractor

Circuit Image:

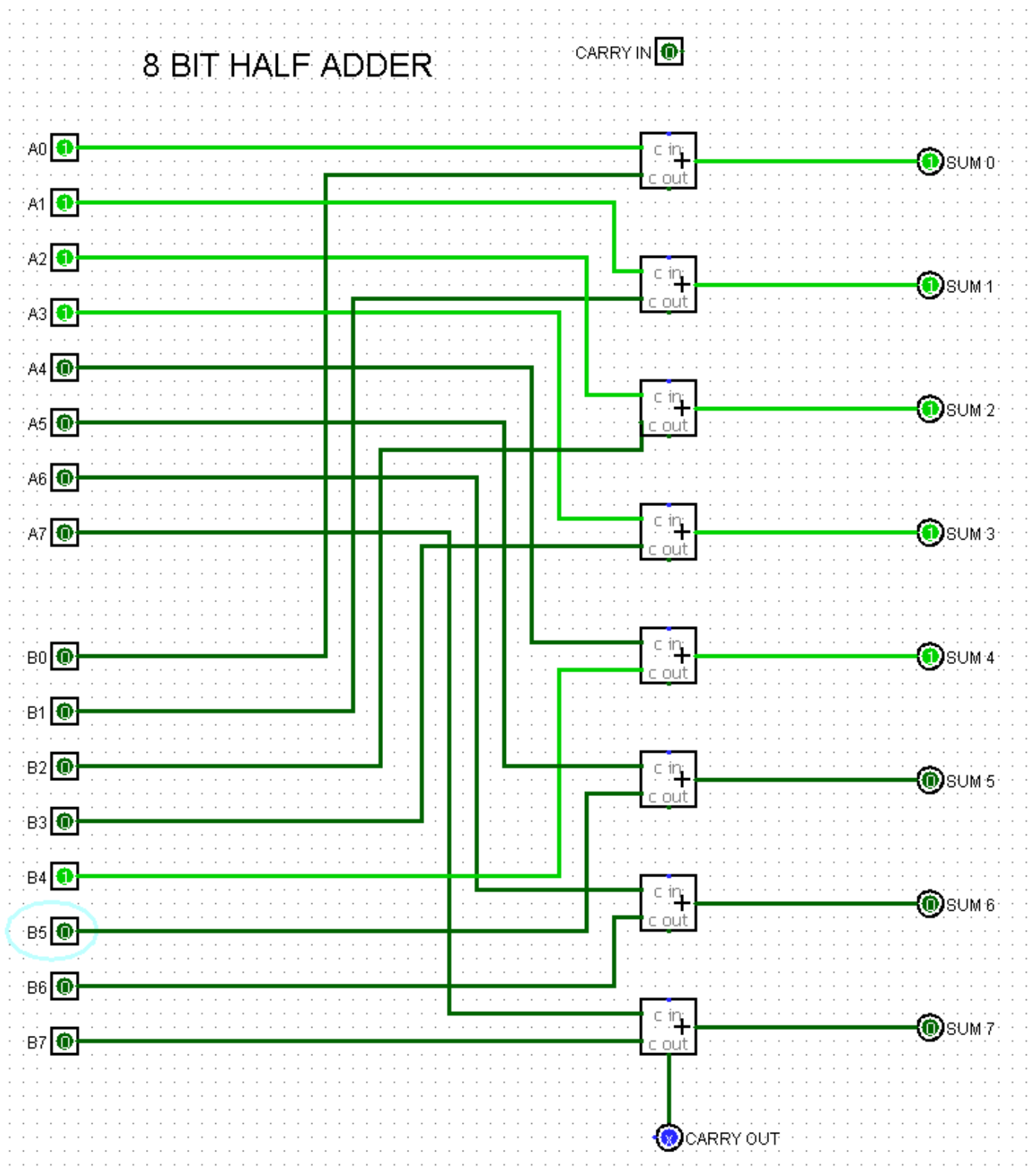


Truth Table:

BORROWIN	A0	A1	A2	A3	B0	B1	B2	B3	SUB0	SUB1	SUB2	SUB3	BORROWOUT
0	0	0	0	0	0	0	0	0	0	0	0	0	x
0	0	0	0	0	0	0	0	1	0	0	0	1	x
0	0	0	0	0	0	0	1	0	0	0	1	1	x
0	0	0	0	0	0	0	1	1	0	0	1	0	x
0	0	0	0	0	0	1	0	0	0	1	1	1	x
0	0	0	0	0	0	1	0	1	0	1	1	0	x
0	0	0	0	0	0	1	1	0	0	1	0	1	x
0	0	0	0	0	0	1	1	1	0	1	0	0	x
0	0	0	0	0	1	0	0	0	1	1	1	1	x
0	0	0	0	0	1	0	0	1	1	1	1	0	x
0	0	0	0	0	1	0	1	0	1	1	0	1	x
0	0	0	0	0	1	0	1	1	1	1	0	0	x
0	0	0	0	0	1	1	0	0	1	0	1	1	x
0	0	0	0	0	1	1	0	1	1	0	1	0	x
0	0	0	0	0	1	1	1	0	1	0	0	1	x
0	0	0	0	0	1	1	1	1	1	0	0	0	x
0	0	0	0	1	0	0	0	0	0	0	0	1	x
0	0	0	0	1	0	0	0	1	0	0	0	0	x
0	0	0	0	1	0	0	1	0	0	0	1	0	x
0	0	0	0	1	0	0	1	1	0	0	1	1	x
0	0	0	0	1	0	1	0	0	0	1	1	0	x
0	0	0	0	1	0	1	0	1	0	1	1	1	x
0	0	0	0	1	1	0	0	0	1	1	1	0	x
0	0	0	0	1	1	0	0	1	1	1	1	1	x
0	0	0	0	1	1	1	0	0	1	1	1	0	x
0	0	0	0	1	1	1	0	1	1	1	1	1	x

5. Using the previously built 4-bit half adder, build 8-bit half adder

Circuit Image:

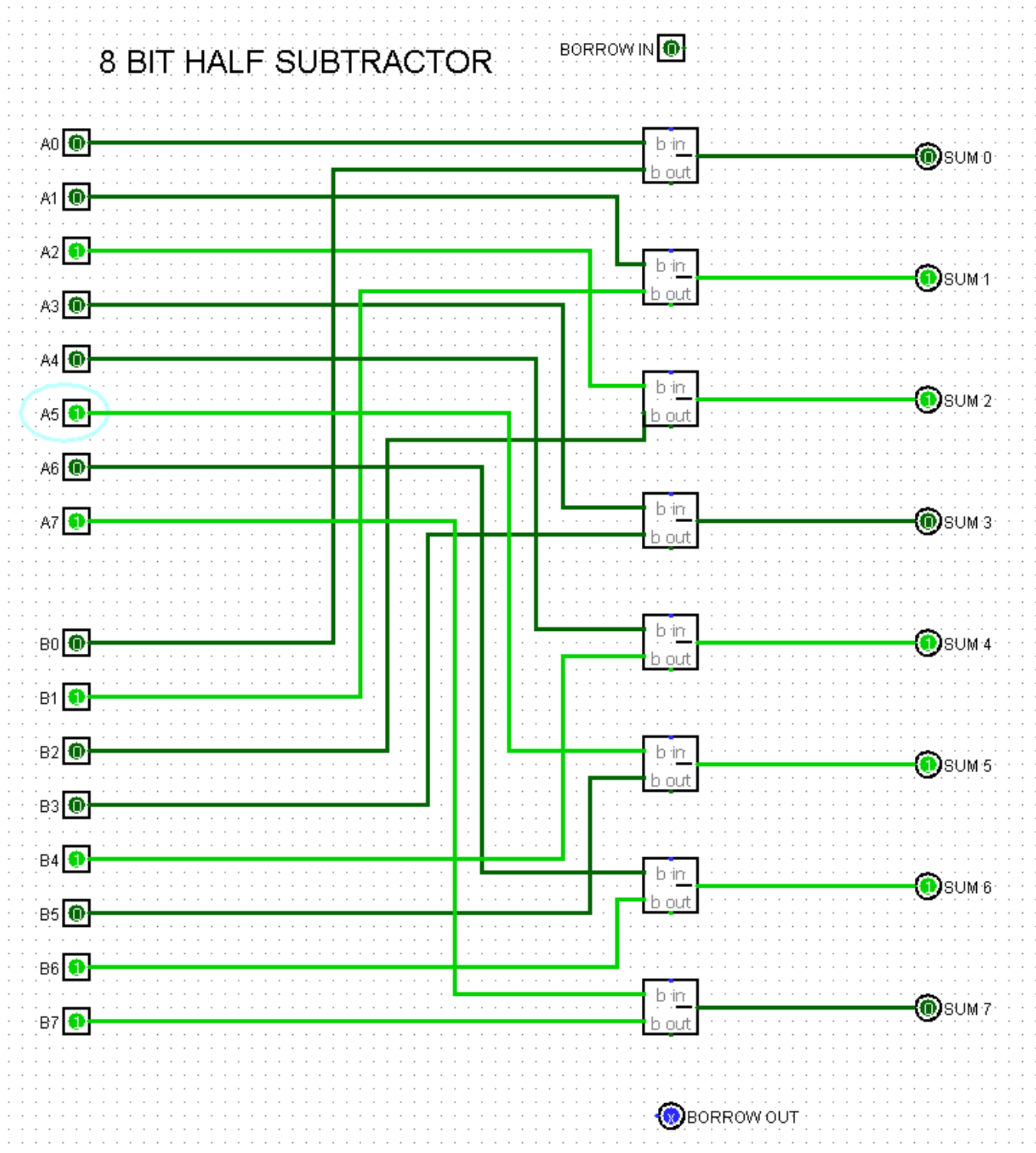


Truth Table: [Not Possible for More Than 12 Inputs: Logisim Limitation]

00001111 + 00010000 = 00011111 [Bit By Bit Adder]

6. Using the previously built 4-bit half subtractor, build 8-bit half subtractor

Circuit Image:

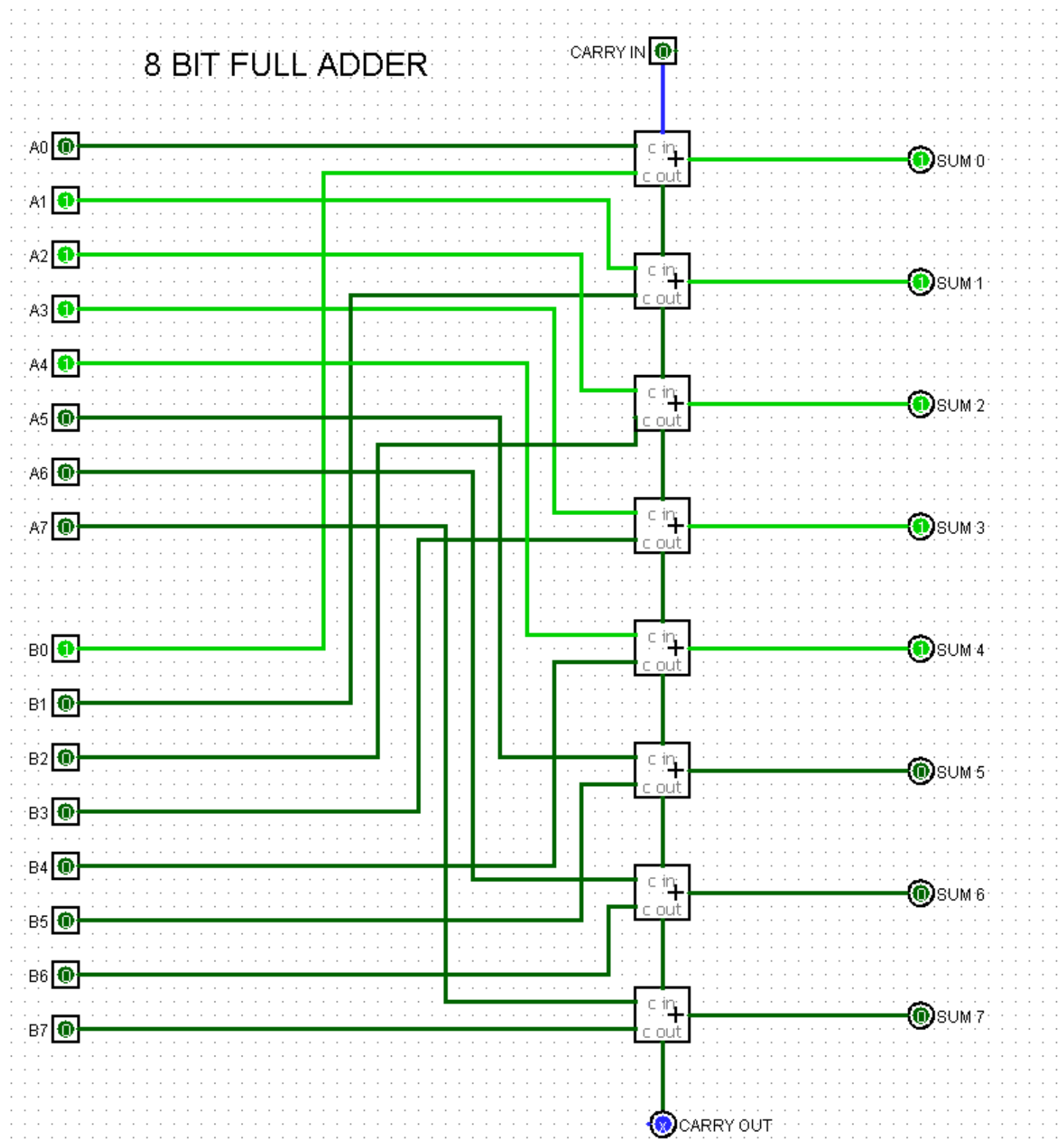


Truth Table: [Not Possible for More Than 12 Inputs: Logisim Limitation]

$10100100 - 11010010 = 01110110$  [Bit By Bit Subtractor]

## 7. Using the previously built 4-bit full adder, build 8-bit full adder

Circuit Image:



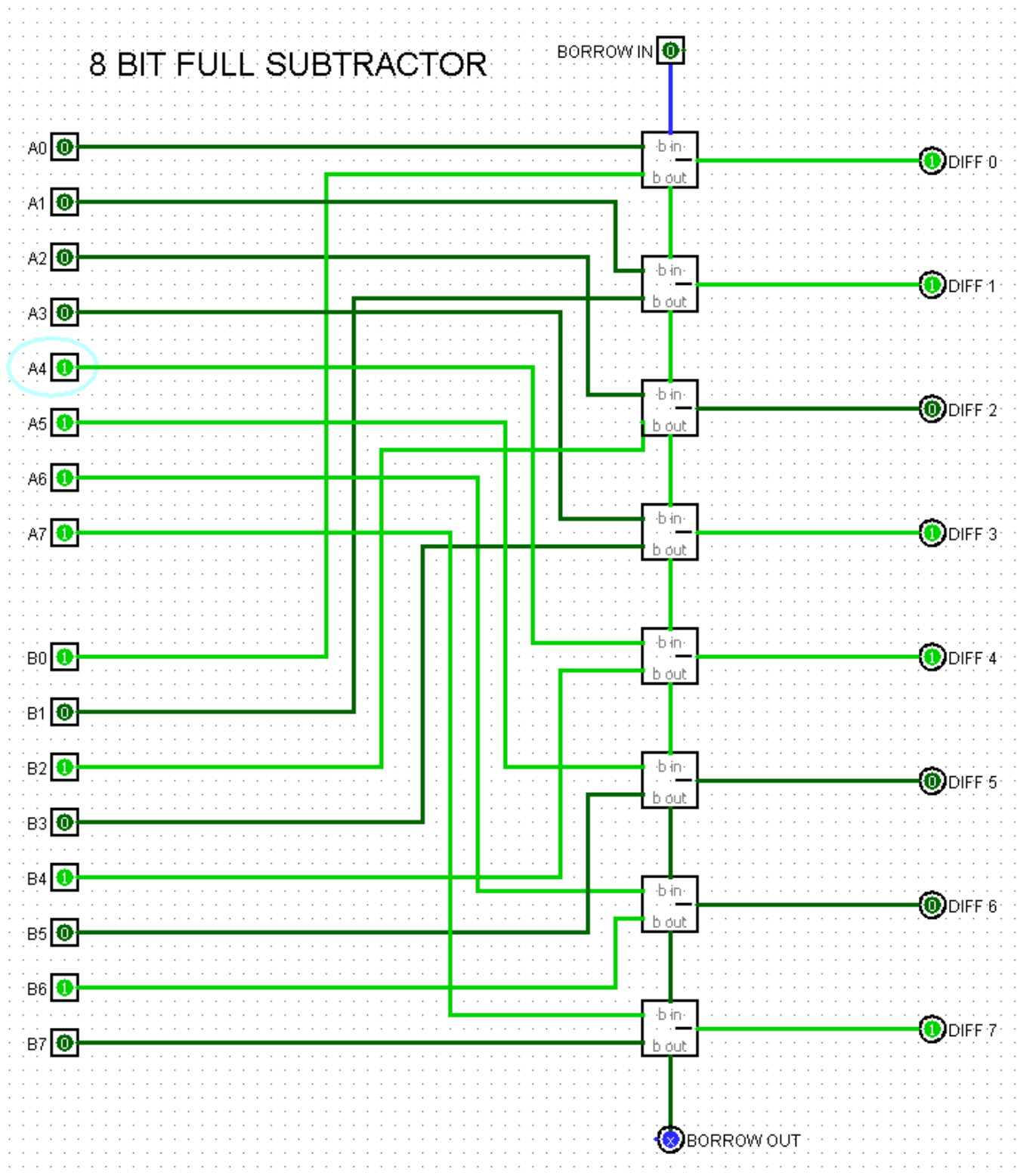
Truth Table: [Not Possible for More Than 12 Inputs: Logisim Limitation]

00011110[30] + 00000001[1] = 00011111[31] (Shown Above)



8. Using the previously built 4-bit full subtractor, build 8-bit full subtractor

Circuit Image:



Truth Table: [Not Possible for More Than 12 Inputs: Logisim Limitation]

11110000[240] - 01010101[85] = 10011011[155] (Shown Above)