GitHub Url: https://github.com/Almo-o/Assembly-Programming/tree/main

Part 1: Logic Gates and Truth Tables

-3-Input AND gate.

A	B	C	A and B and C
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

- AND gate feeds into OR gate

A	B	C	(A And B) Or C
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

2. Output would be 1

Part 2: Boolean Algebra Simplification

- (A + A . B)
  A (1 + B)
  A (1)
  = A

- A . (B + C) + B . C
  = A . B + A . C + B . C

A	B	C	A.(B+C)+B.C	A.B+A.C+B.C
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

Part 3 - Karnaugh Maps and Minimization

	C/D				
A/B		00	01	11	10
	00	0	1	0	1
	01	0	0	0	0
	11	0	0	1	0
	10	0	0	0	0

= inverse(A.B.D).C + Inverse(A.B.C).D + ABCD

P0 = A0 .B0

	B1/B0				
A1/A0		00	01	11	10
	00	0	0	0	0
	01	0	1	1	0
	11	0	1	1	0
	10	0	0	0	0

P1 = Inverse(A1).A0.B1 + A0.B1.Inverse(B0) + A1.Inverse(B1).B0 + A1.Inverse(B1).B0

	B1/B0				
A1/A0		00	01	11	10
	00	0	0	0	0
	01	0	0	1	1,1
	11	0	1	0	1
	10	0	1,1	1	0

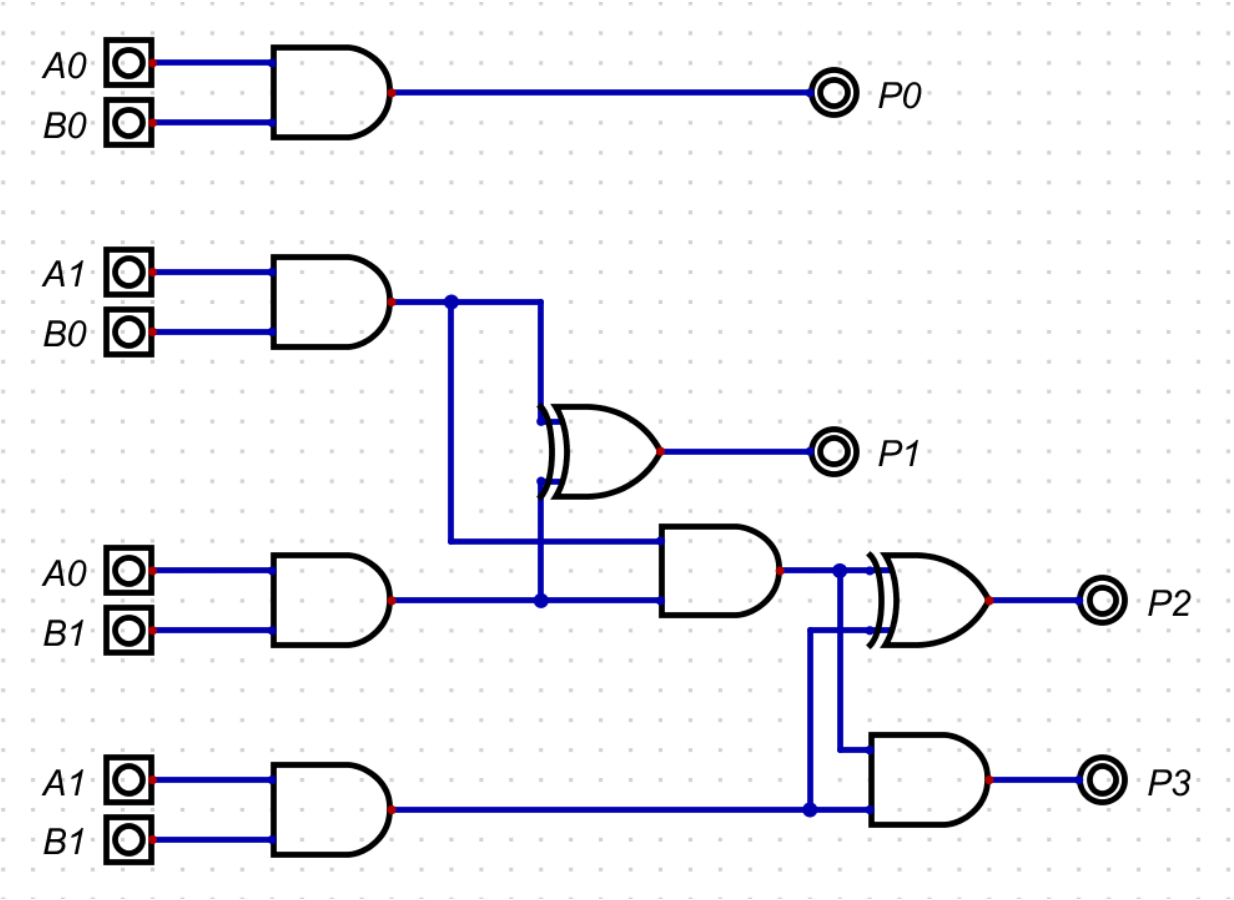
P2 = A1.B1.Inverse(B0) +A1.Inverse(A0).B1

	B1/B0				
A1/A0		00	01	11	10
	00	0	0	0	0
	01	0	0	0	0
	11	0	0	0	1
	10	0	0	1	1,1

P3 = A1.A0.B1.B0

	B1/B0				
A1/A0		00	01	11	10
	00	0	0	0	0
	01	0	0	0	0
	11	0	0	1	0
	10	0	0	0	0

2 - BIT MULTIPLIER



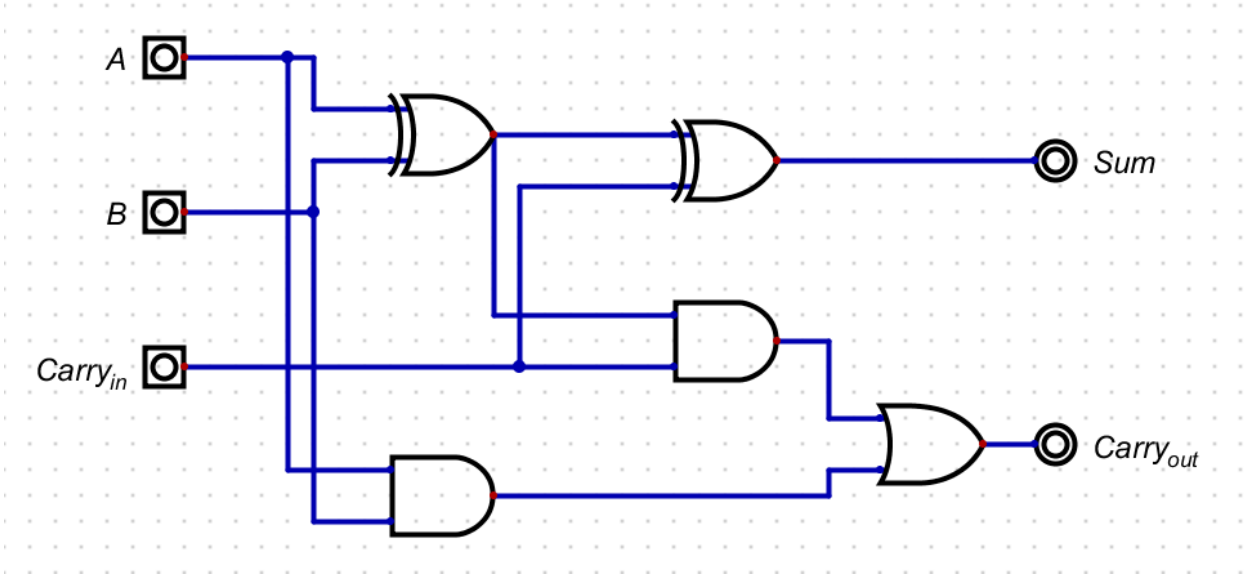
4-Bit Full Adder Implementation

Truth table for a full adder.

A	B	C in	Sum	Carry out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

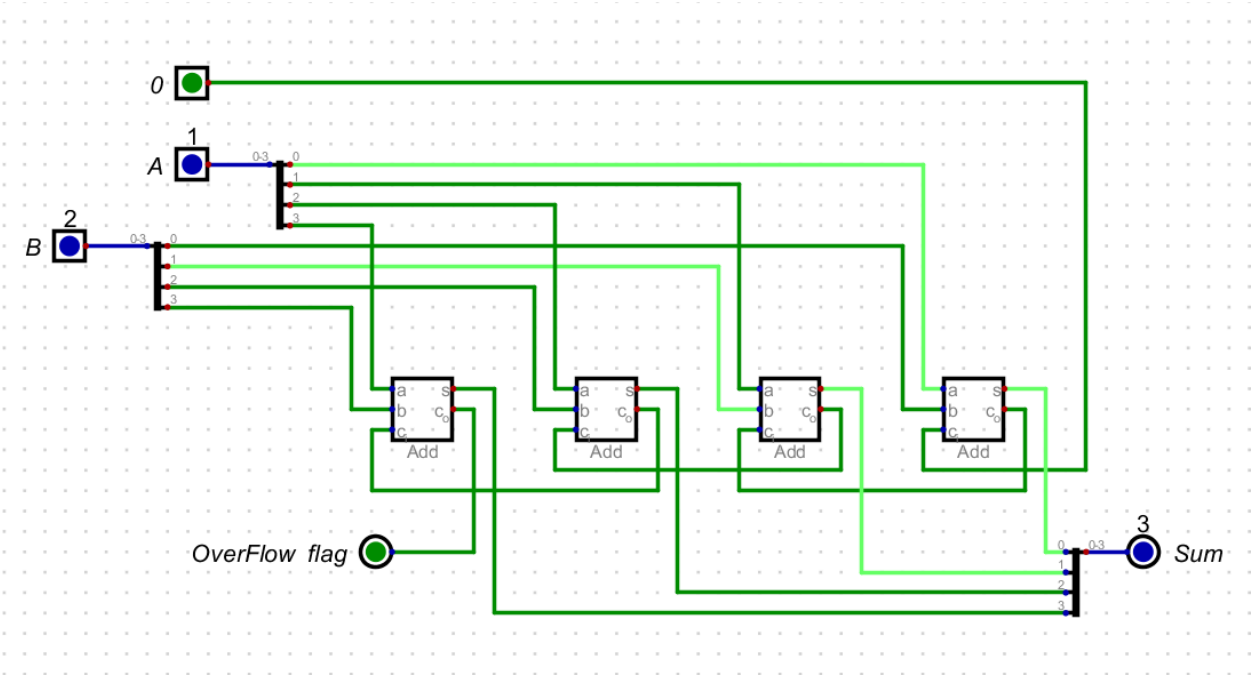
Boolean Expressions => Sum = (A xor B) xor C in
Carry = A.B + C in . (A xor B)

Single Bit Adder

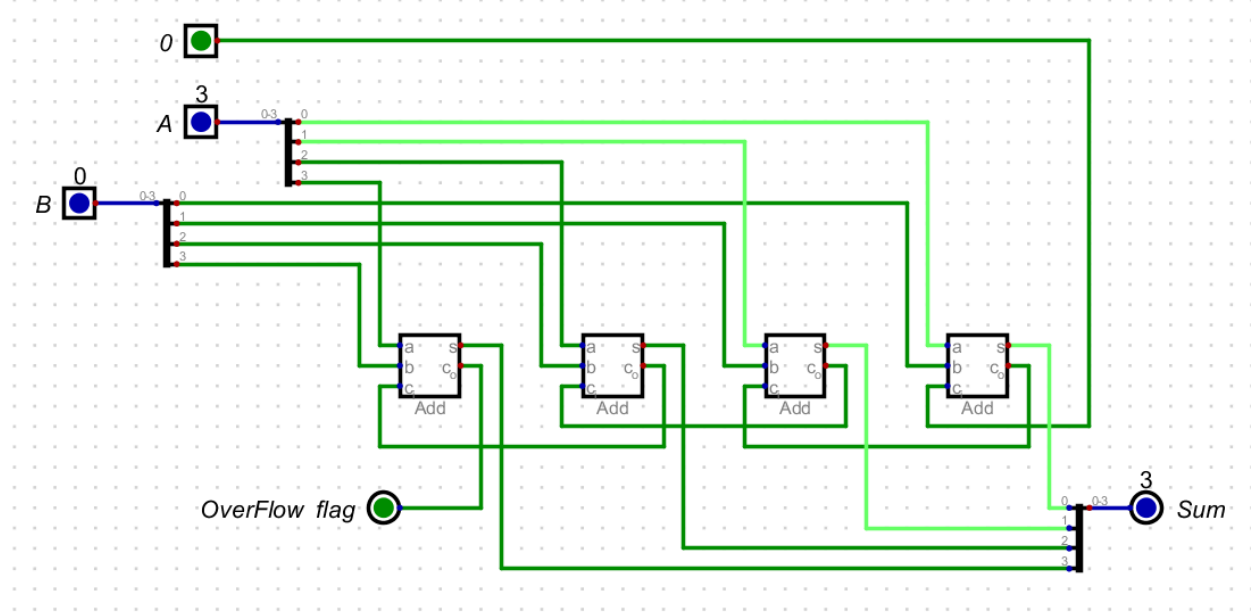


4-Bit Adder

Example 1) 1 + 2 = 3



Example 2) 3 + 0 = 3



Example 3) 4 + 4 = 8