Final 4-bit ALU Design for 4-bit CPU

Nahin Ul Sadad
Lecturer
CSE, RUET

ALU

ALU chip is shown below:

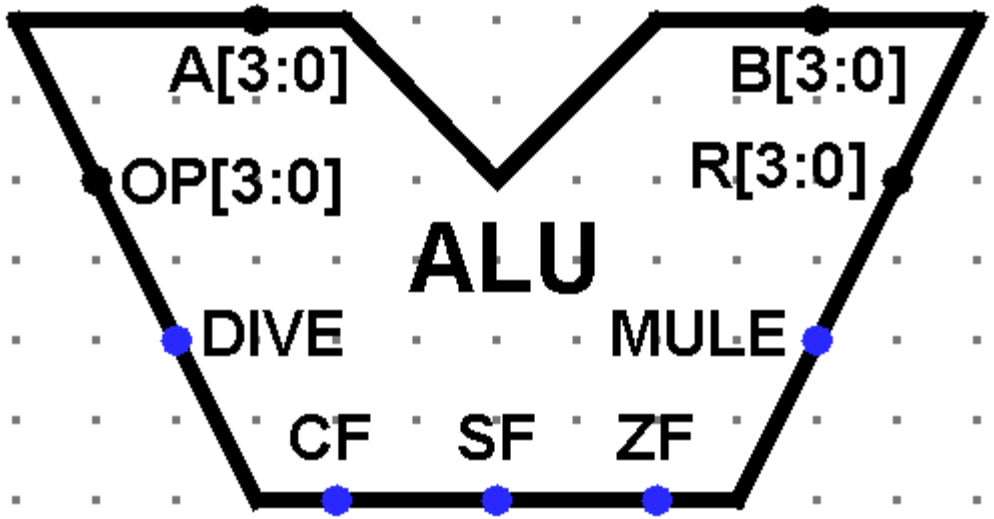


Figure: 4-bit ALU where it performs operations on A and B based on OP value.

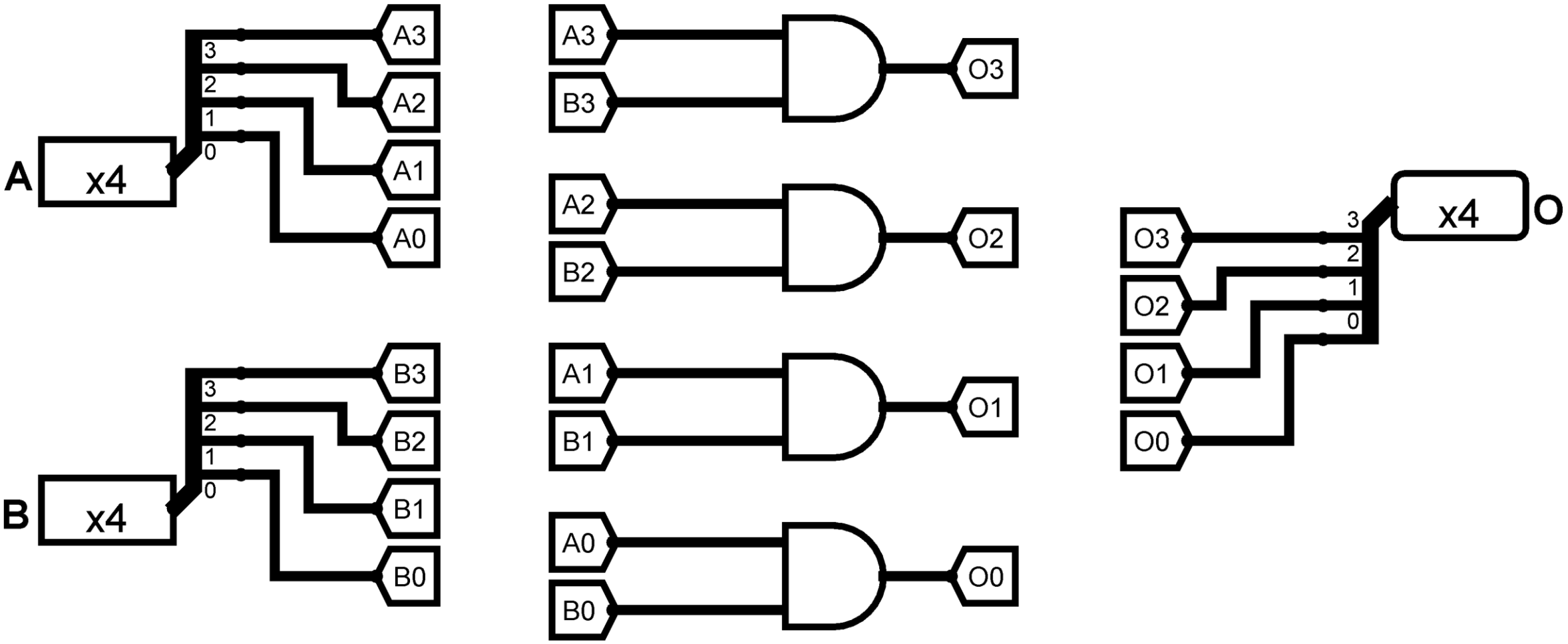
Here,
A = Data1 of ALU
B = Data2 of ALU
OP = Opcode of ALU
R = Result of ALU

CF = Carry Flag
SF = Sign Flag
ZF = Zero Flag

DIVE = DIV Exception
MULE = MUL Exception

Logic Circuits
AND Gate
OR Gate
XOR Gate
NOT Gate

4-bit AND gate



4-bit AND Gate Simulation

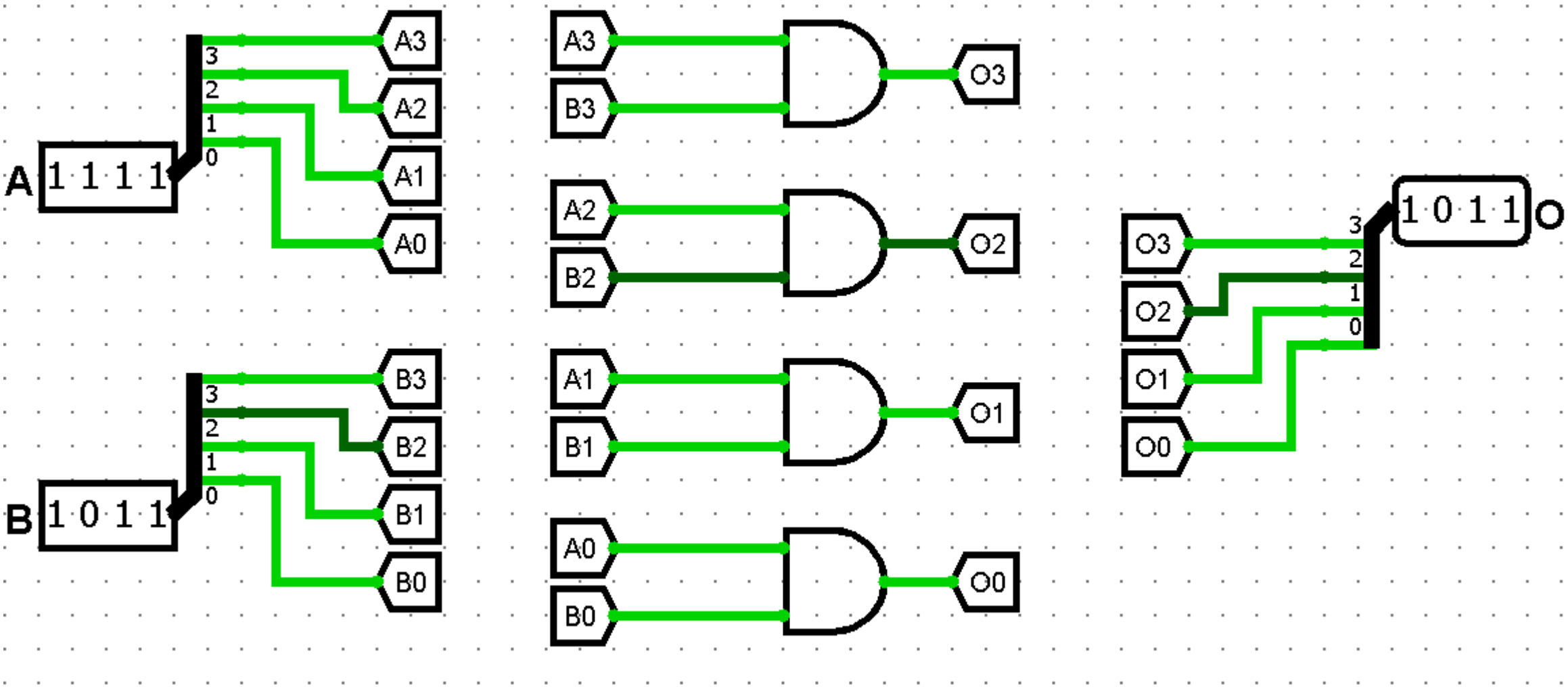
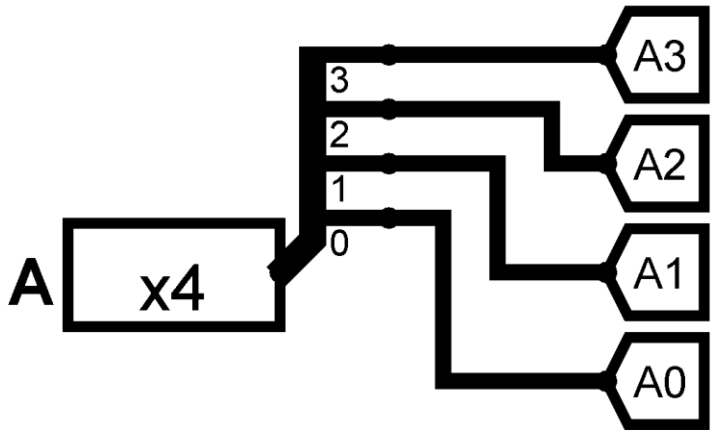
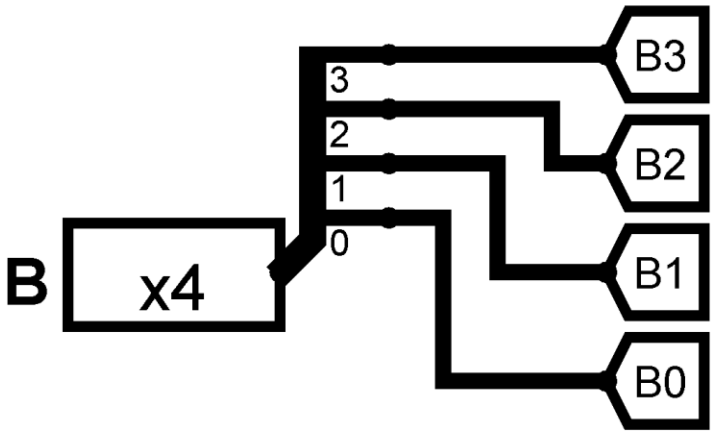
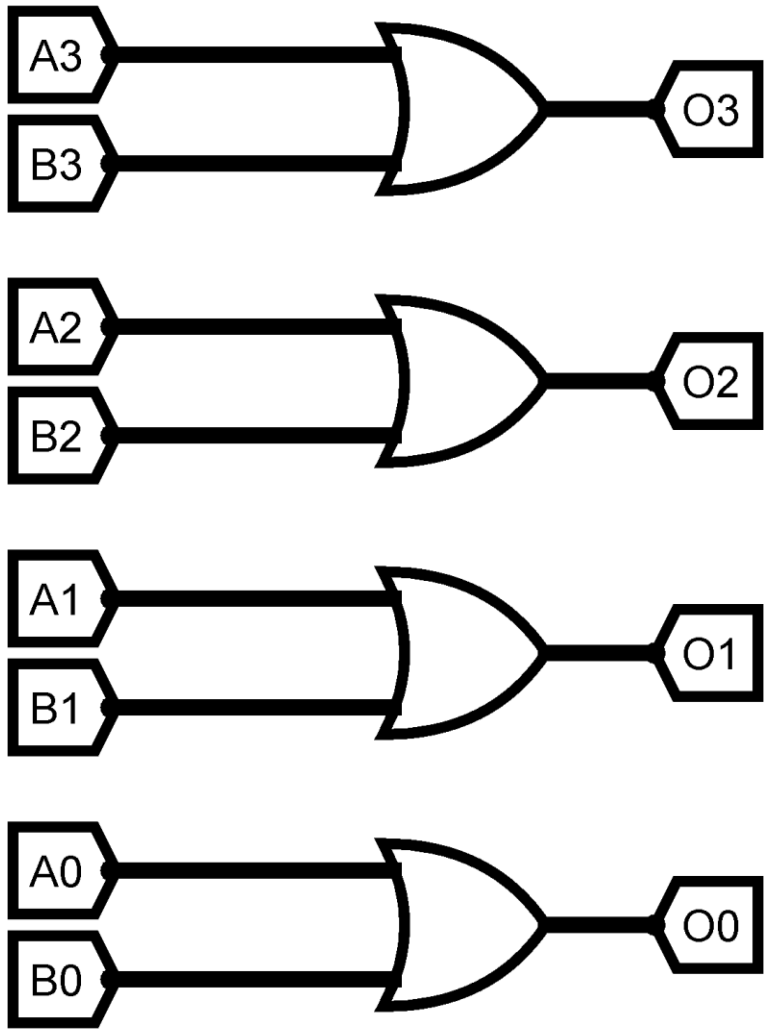


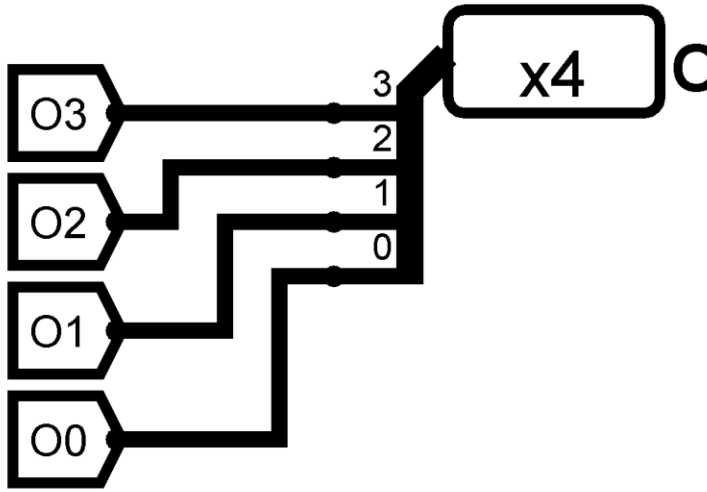
Figure: 4-bit AND Gate Simulation for input A = 1111 and B = 1011.
Output is O = 1011

4-bit OR gate


A
x4
3
2
1
0
A3
A2
A1
A0


B
x4
3
2
1
0
B3
B2
B1
B0


A3
B3
O3
A2
B2
O2
A1
B1
O1
A0
B0
O0


O3
O2
O1
O0
3
2
1
0
x4
O

4-bit OR Gate Simulation

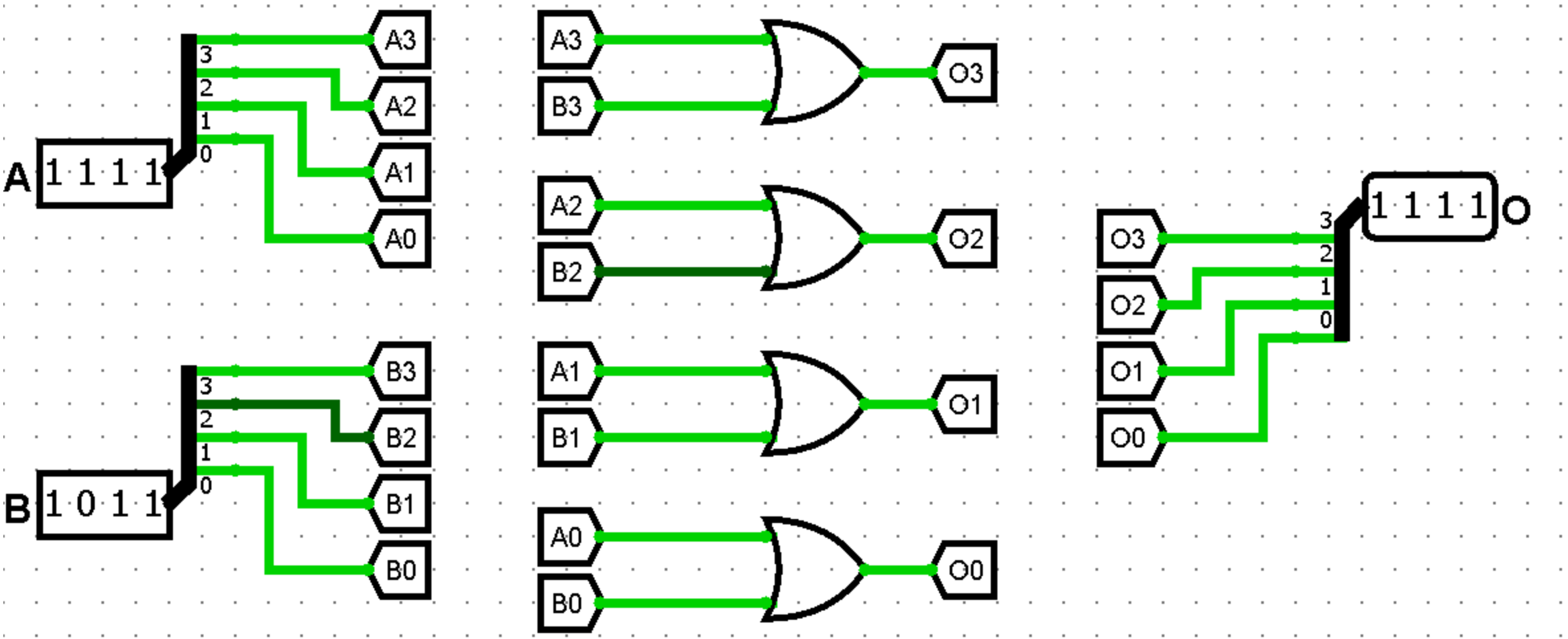
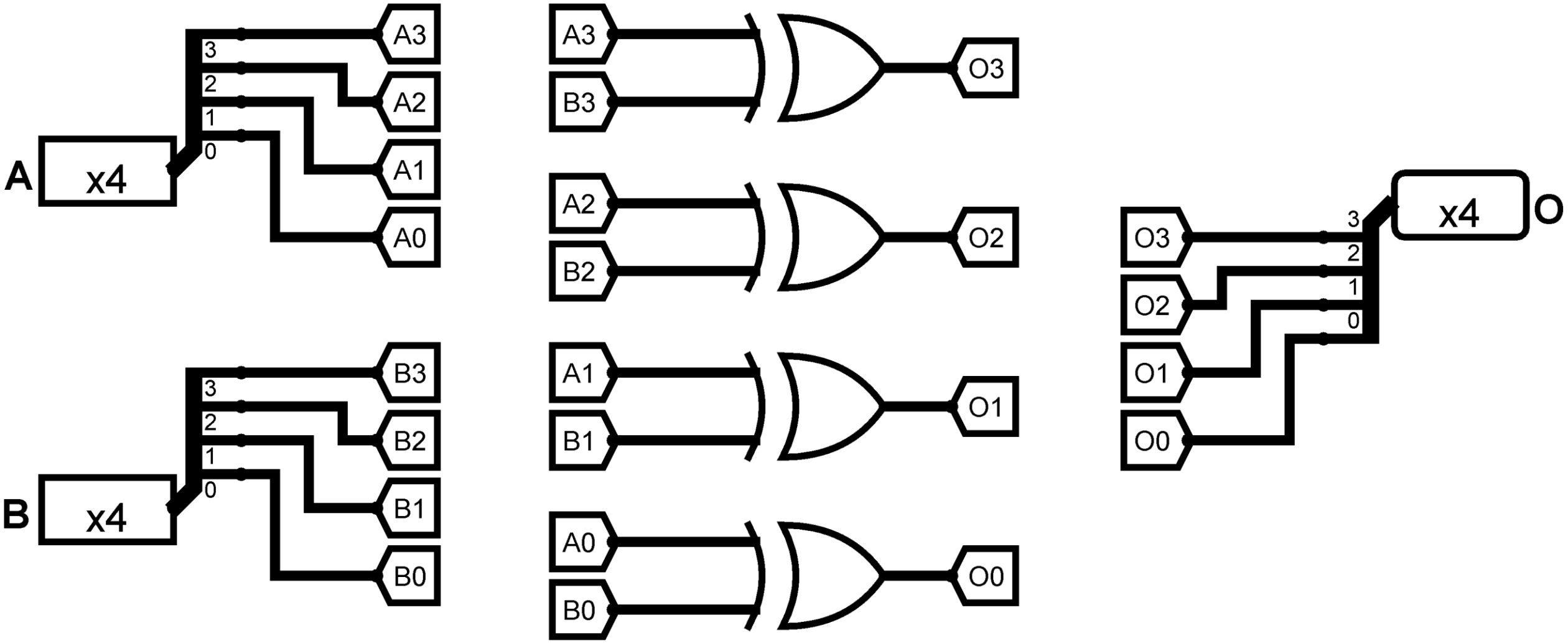


Figure: 4-bit OR Gate Simulation for input A = 1111 and B = 1011.
Output is O = 1111

4-bit XOR gate



4-bit XOR Gate Simulation

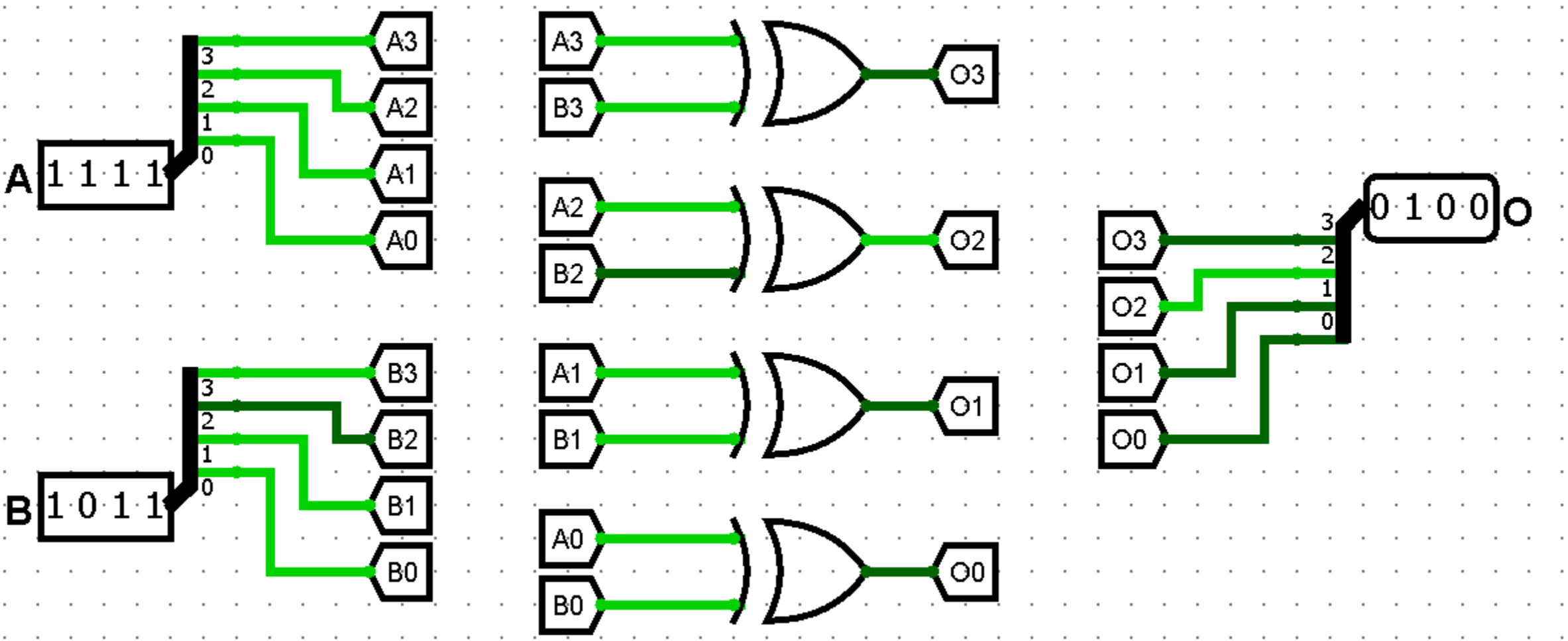
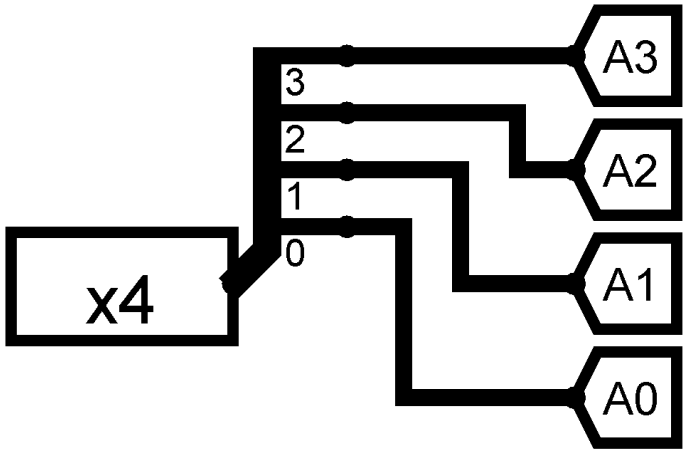
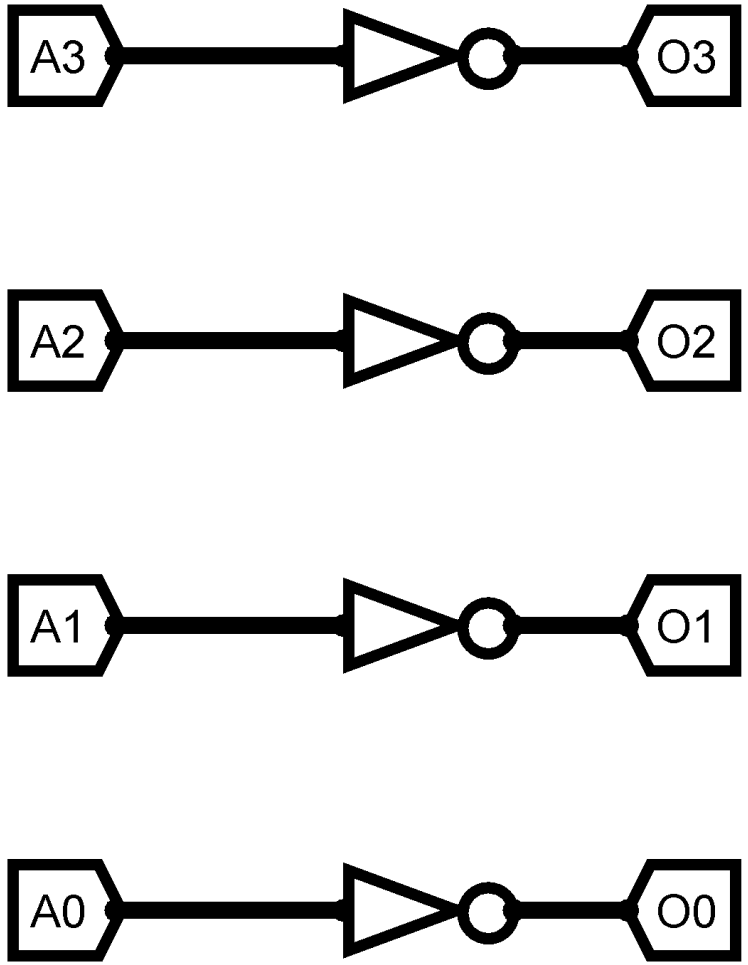
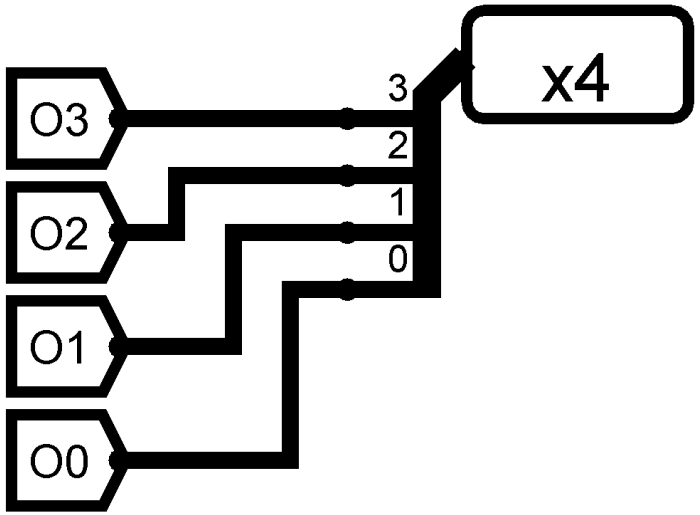


Figure: 4-bit XOR Gate Simulation for input A = 1111 and B = 1011.
Output is O = 0100

4-bit NOT Gate







4-bit NOT Gate Simulation

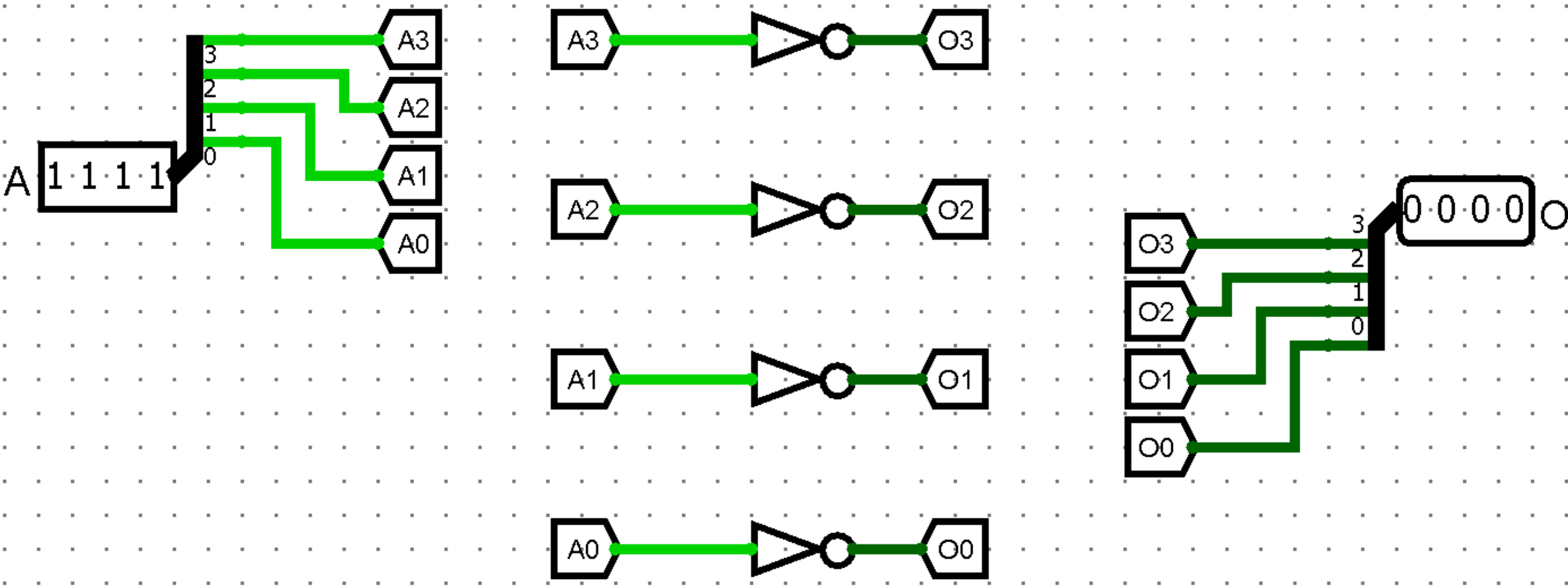
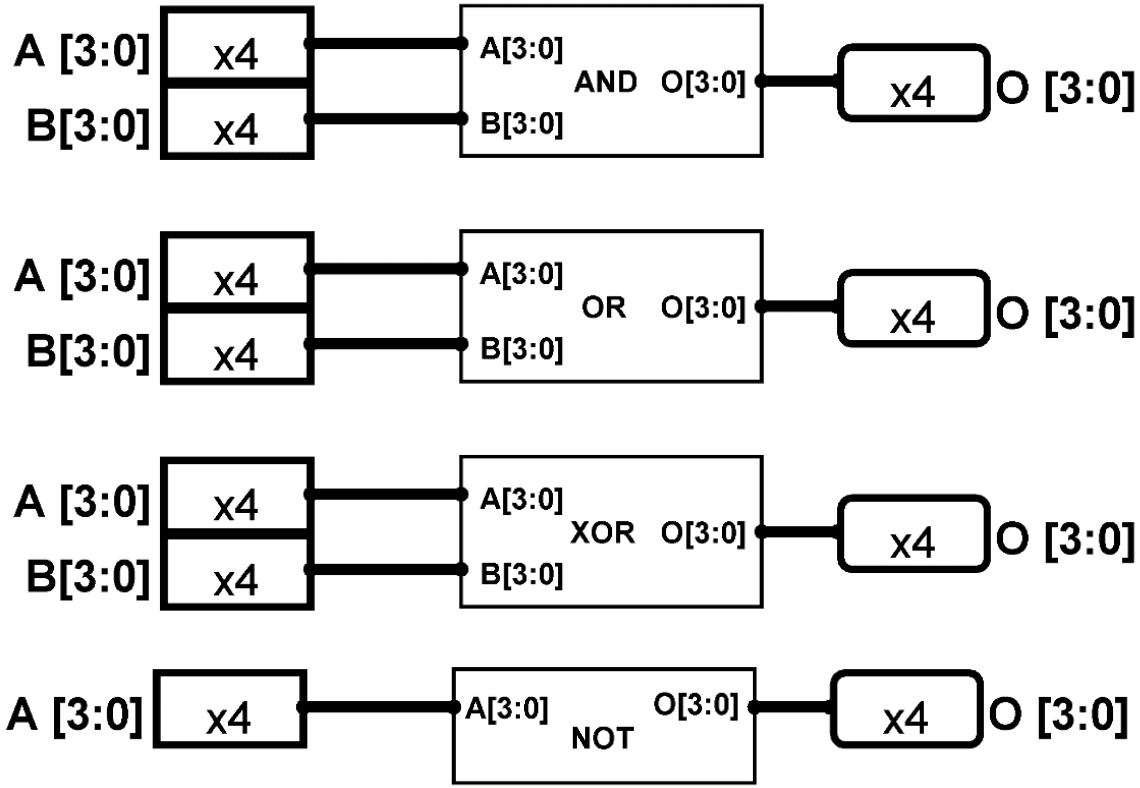


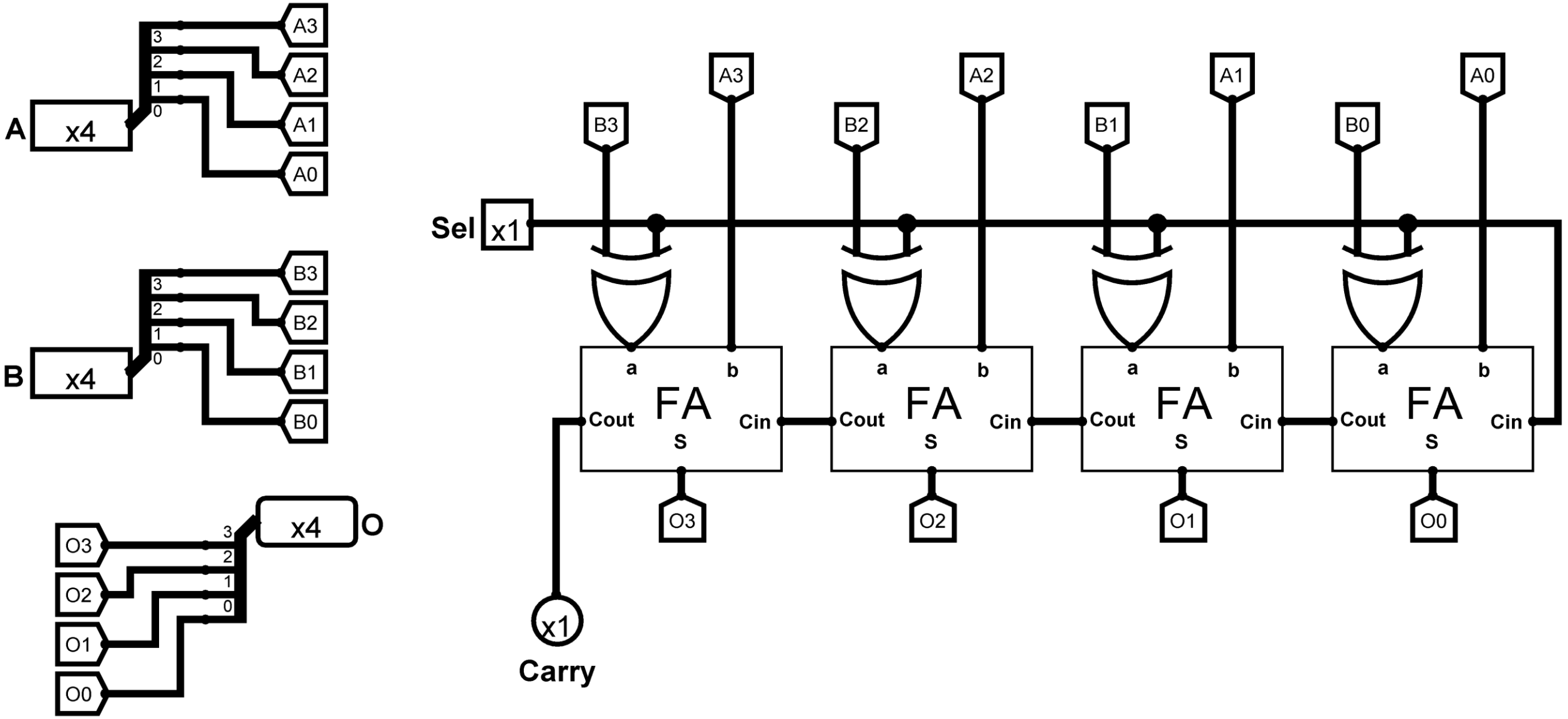
Figure: 4-bit NOT Gate Simulation for input A = 1111. Output is O = 0000

Logic Circuits

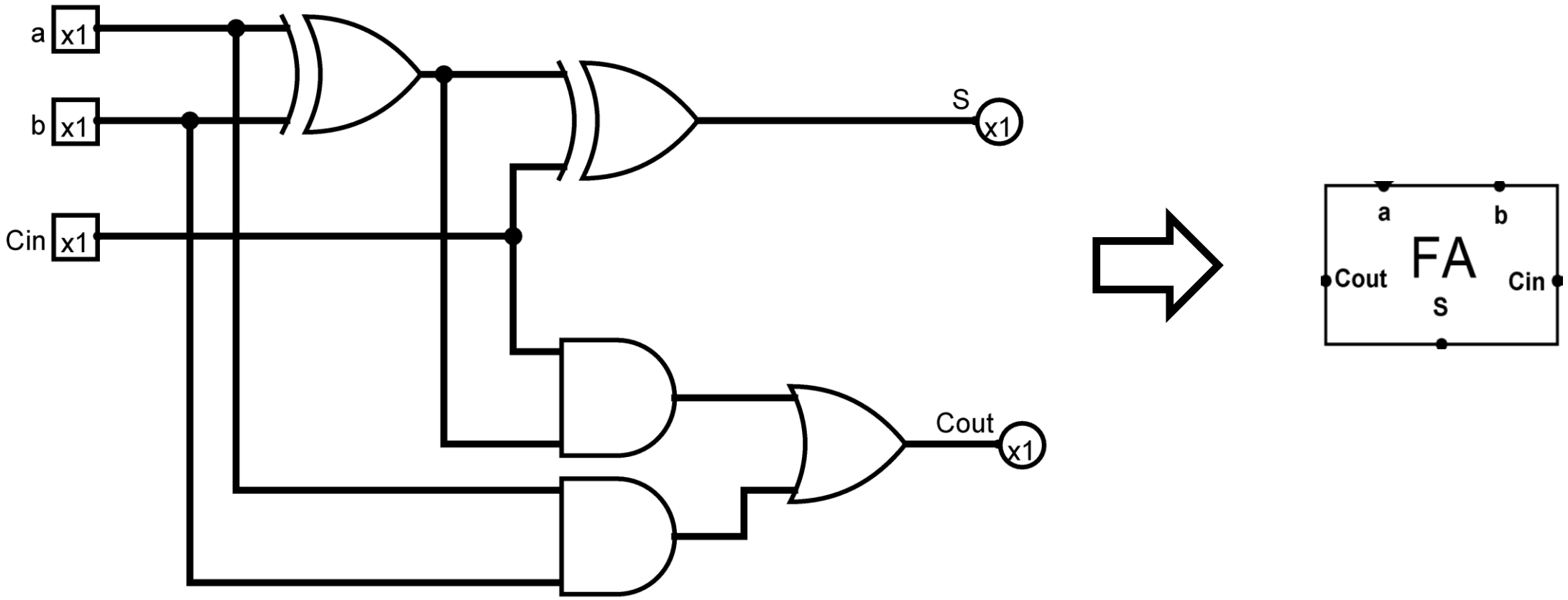


Arithmetic Circuits
Adder
Subtractor
Multiplier
Divider

4-bit Adder/Subtractor



4-bit Adder/Subtractor



4-bit Adder/Subtractor Simulation

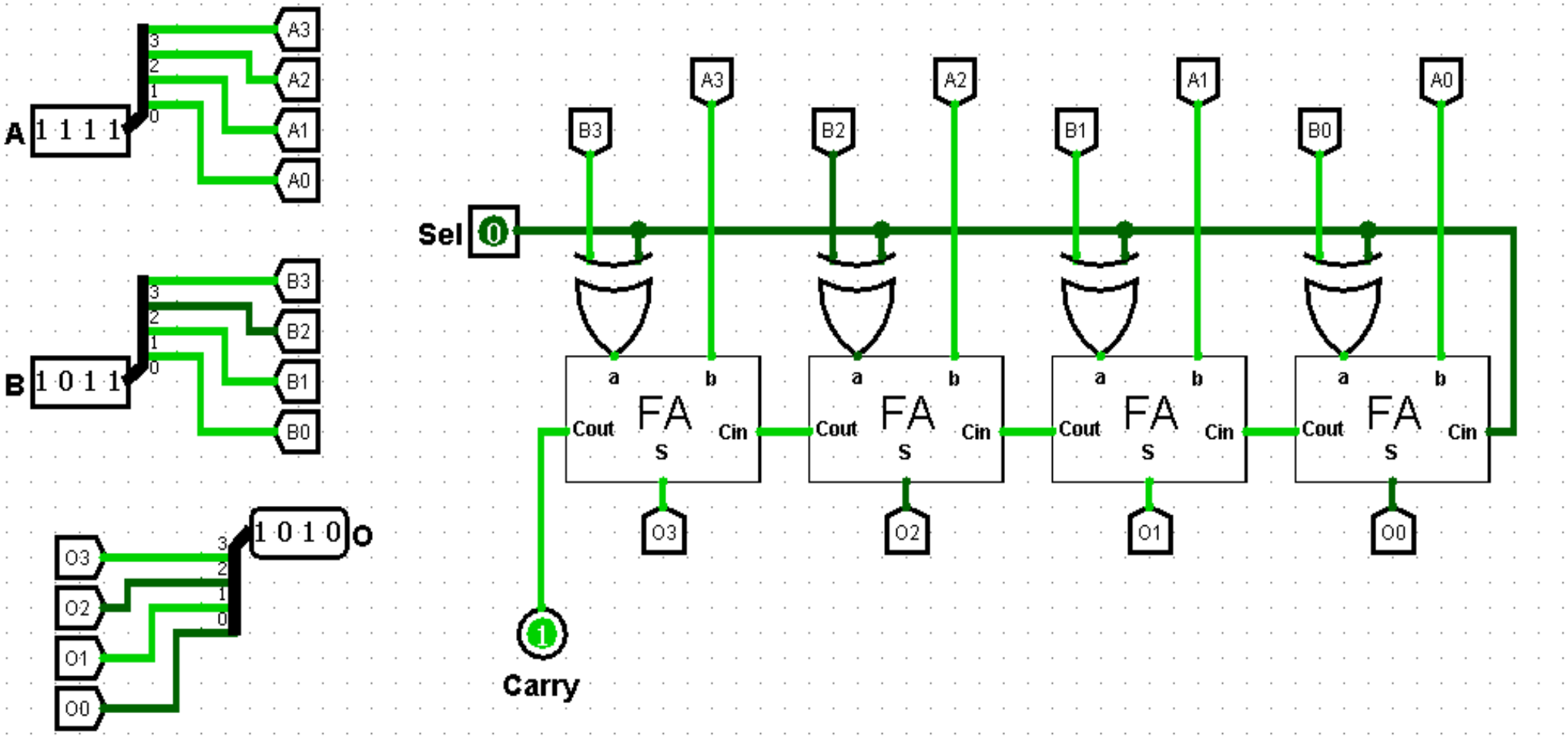


Figure: 4-bit Adder/Subtractor Simulation for input A = 1111, B = 1011, Sel = 0 (ADD). Output O = 1010 and Carry = 1

4-bit Adder/Subtractor Simulation

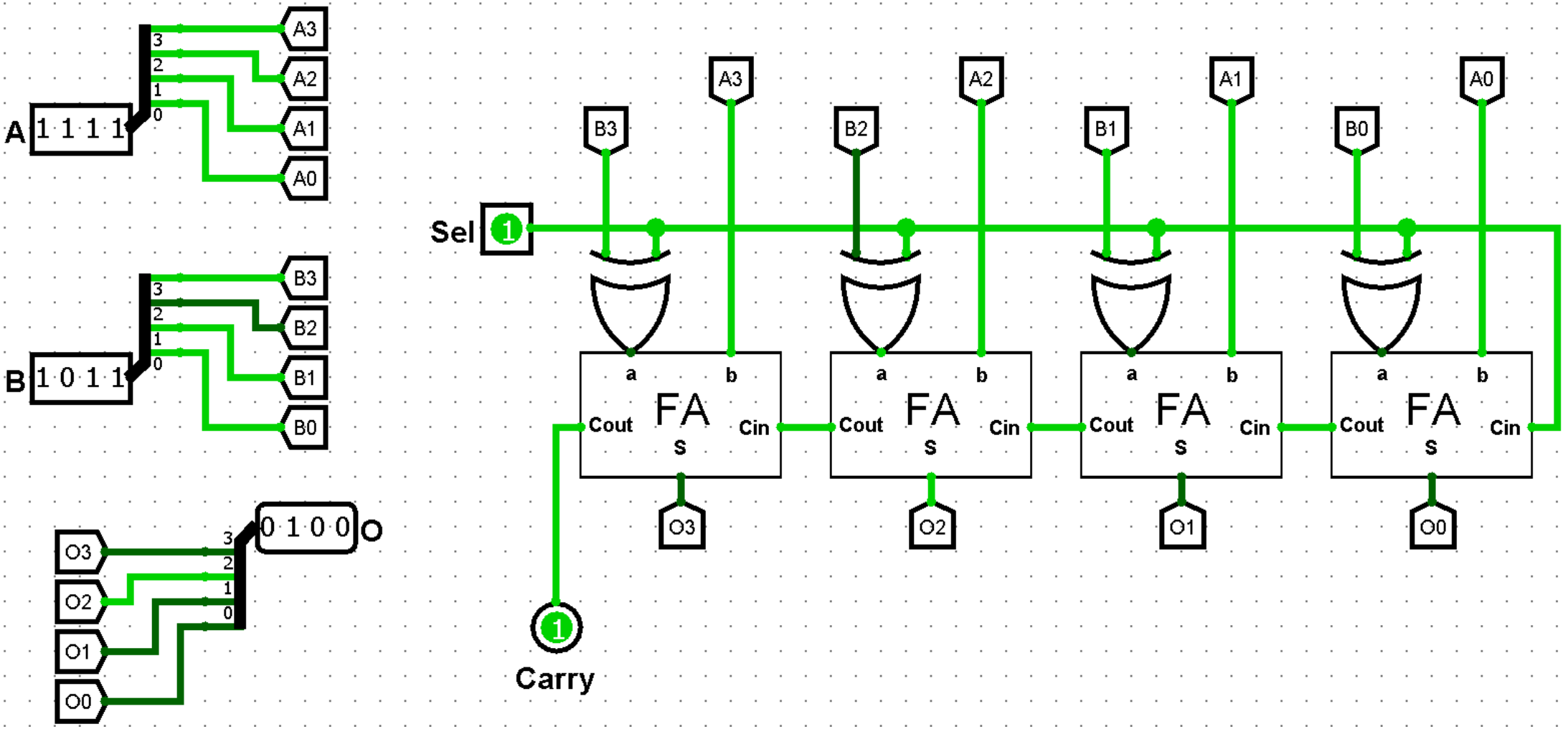
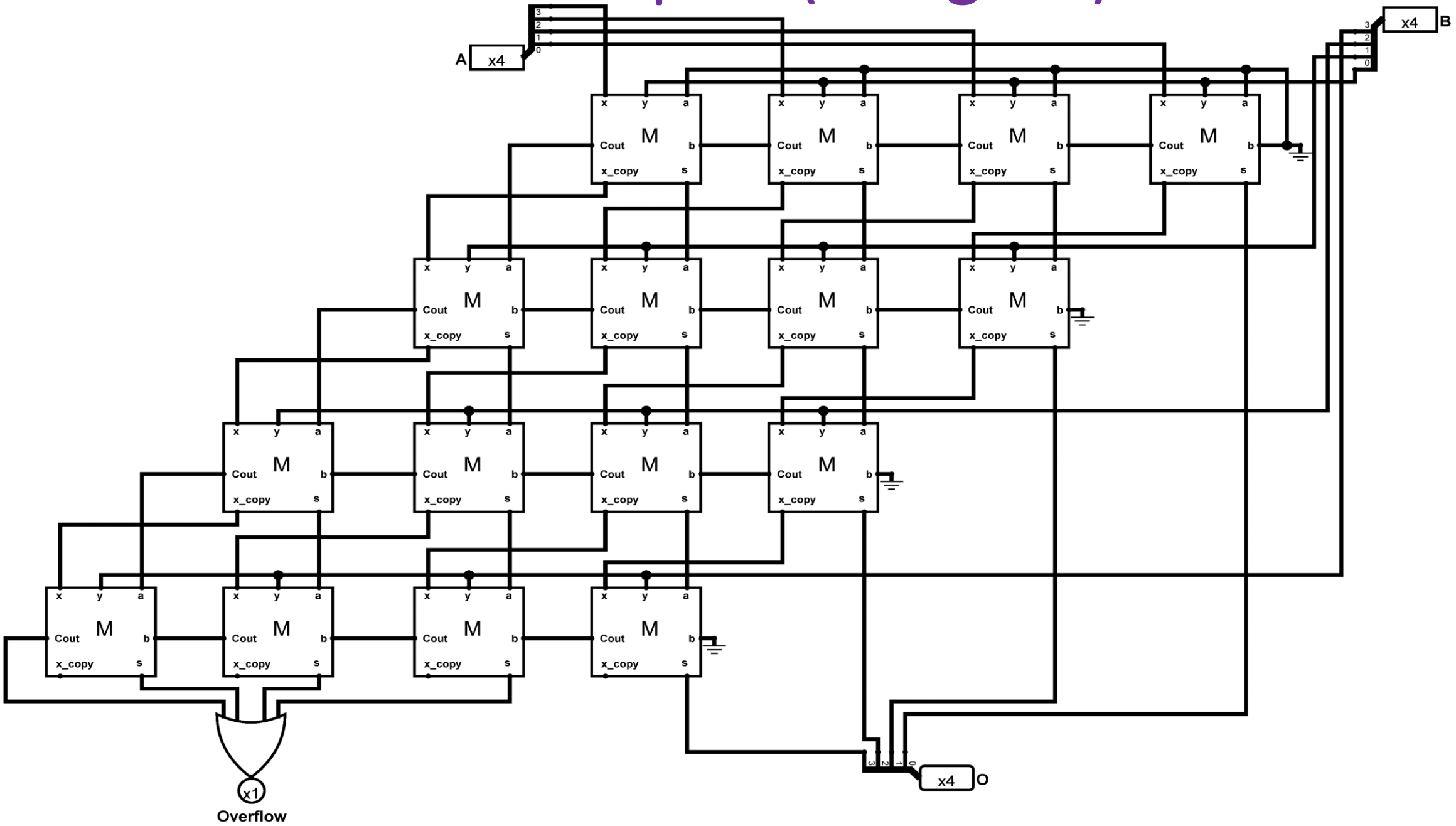


Figure: 4-bit Adder/Subtractor Simulation for input A = 1111, B = 1011, Sel = 0 (SUB). Output O = 0100 and Carry = 1

4-bit Multiplier (Unsigned)



4-bit Multiplier (Unsigned) Simulation

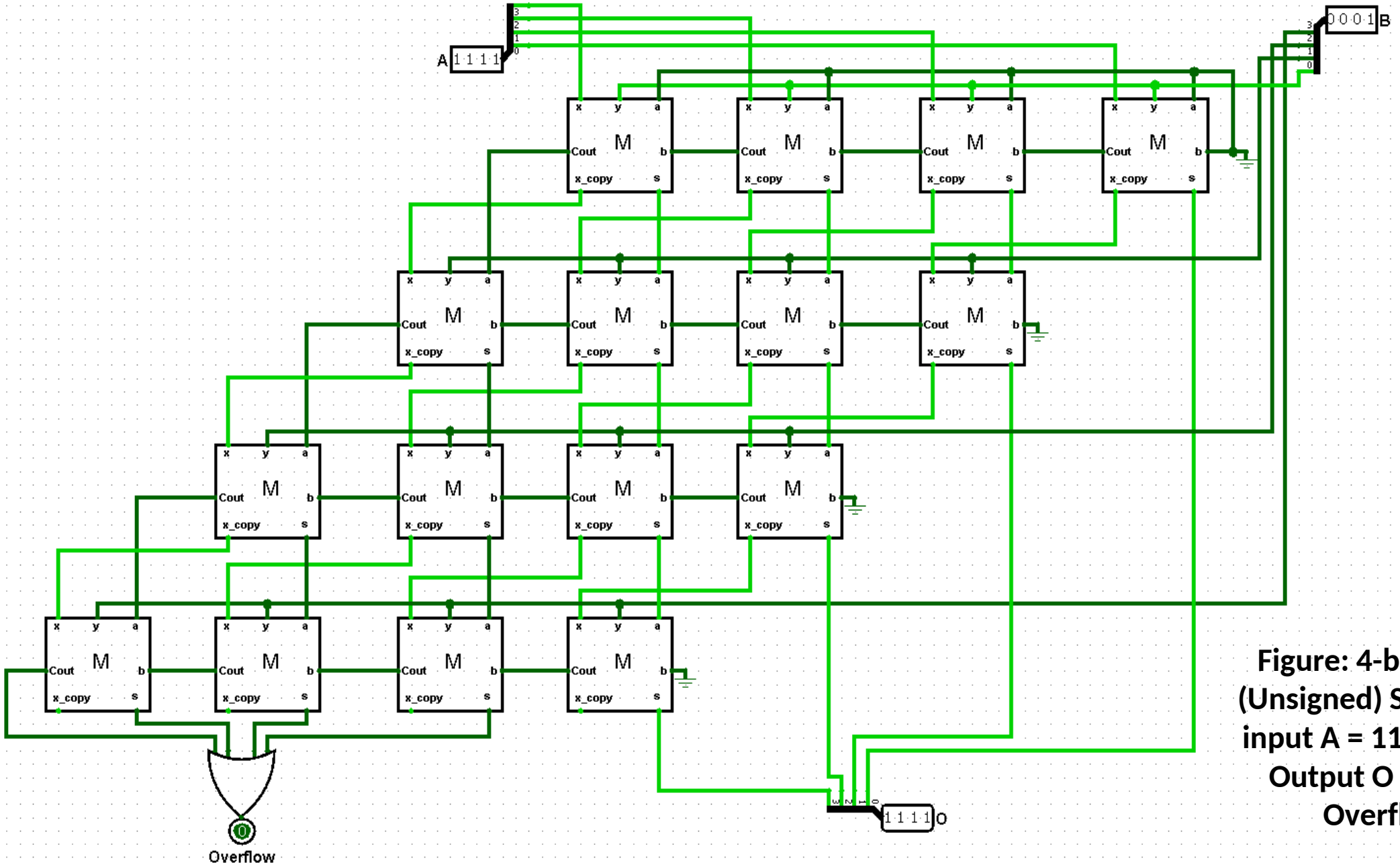
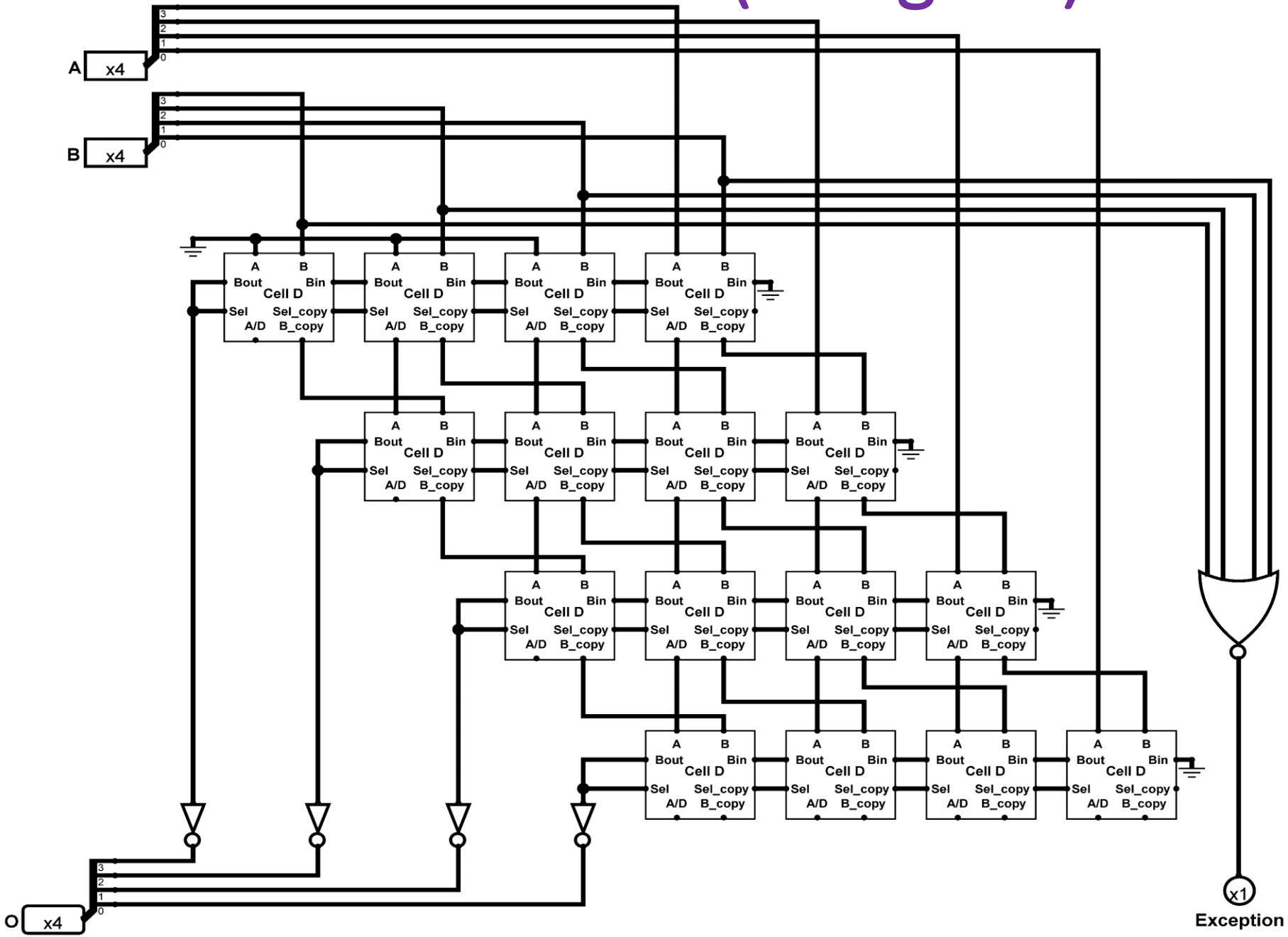


Figure: 4-bit Multiplier (Unsigned) Simulation for input A = 1111, B = 0001. Output O = 1111 and Overflow = 0

4-bit Divider (Unsigned)



4-bit Divider (Unsigned) Simulation

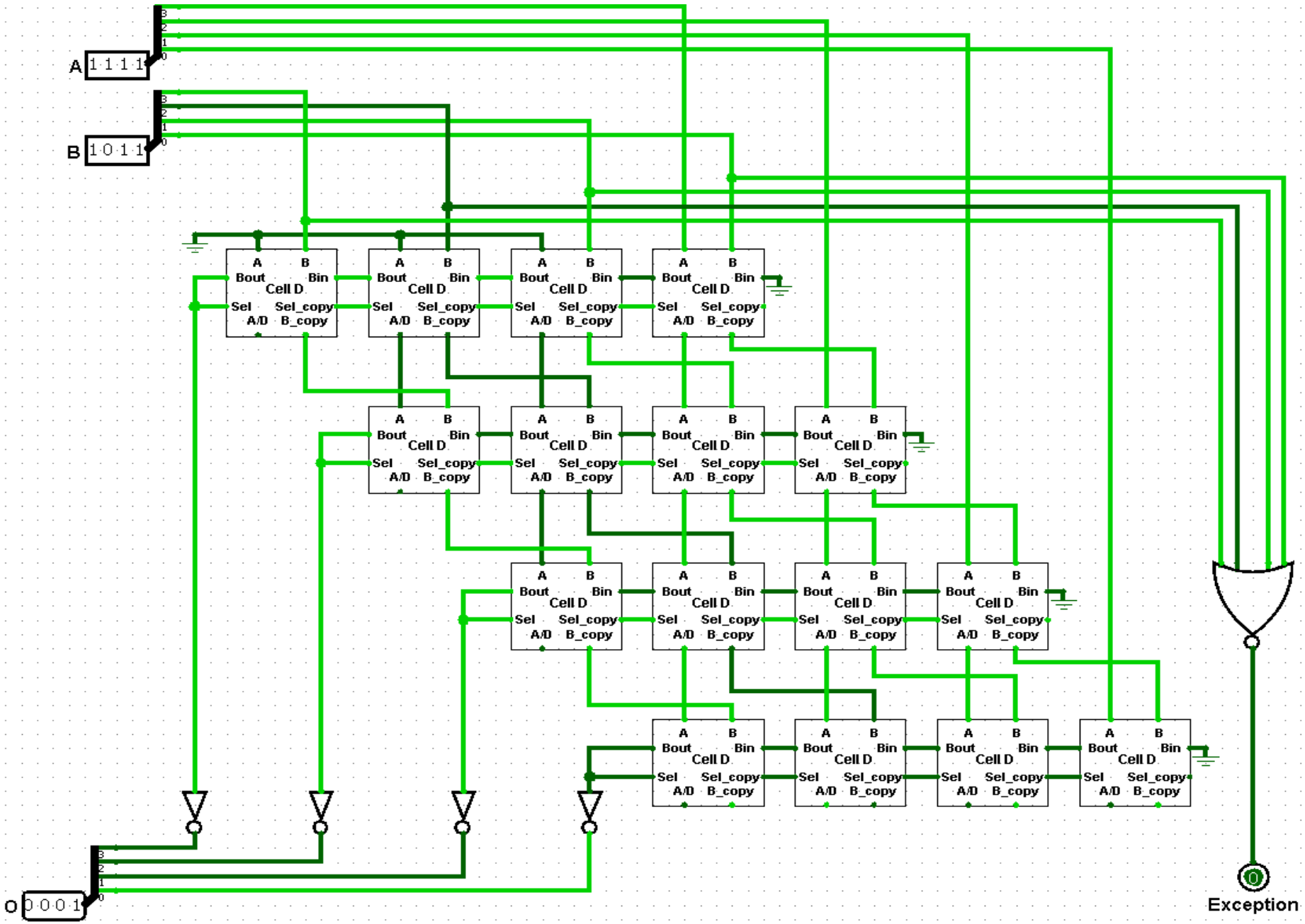
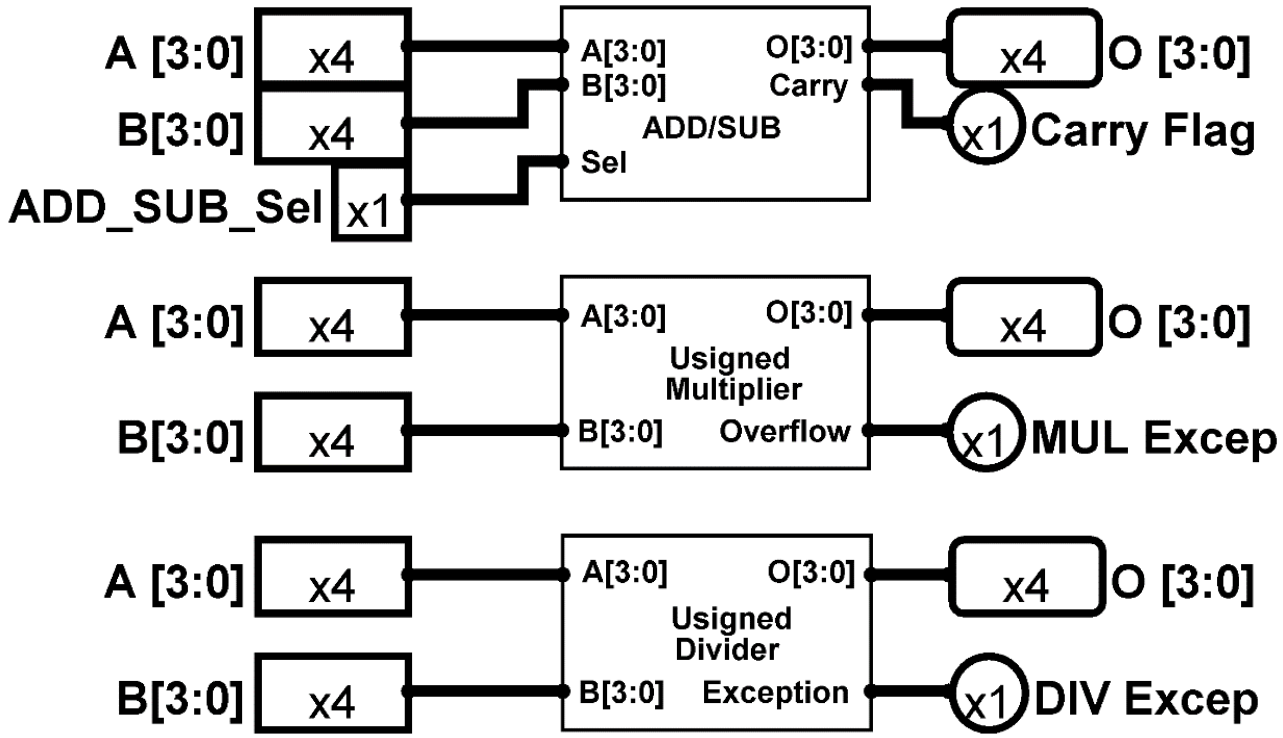


Figure: 4-bit Divider (Unsigned) Simulation for input A = 1111, B = 1011. Output O = 0001 and Exception = 0

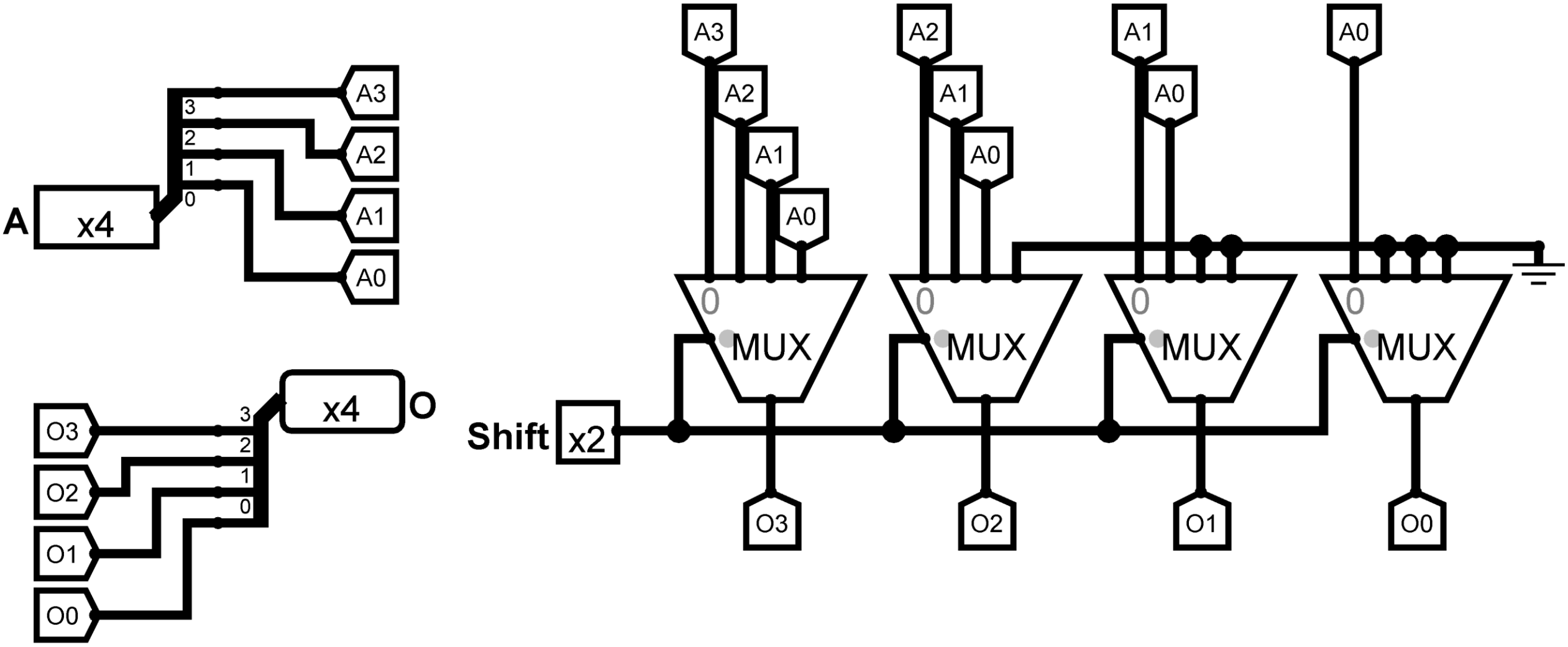
Arithmetic Circuits Block

If $\mathbf{ADD\_SUB\_Sel = 0,\ O = ADD}$
If $\mathbf{ADD\_SUB\_Sel = 1,\ O = SUB}$

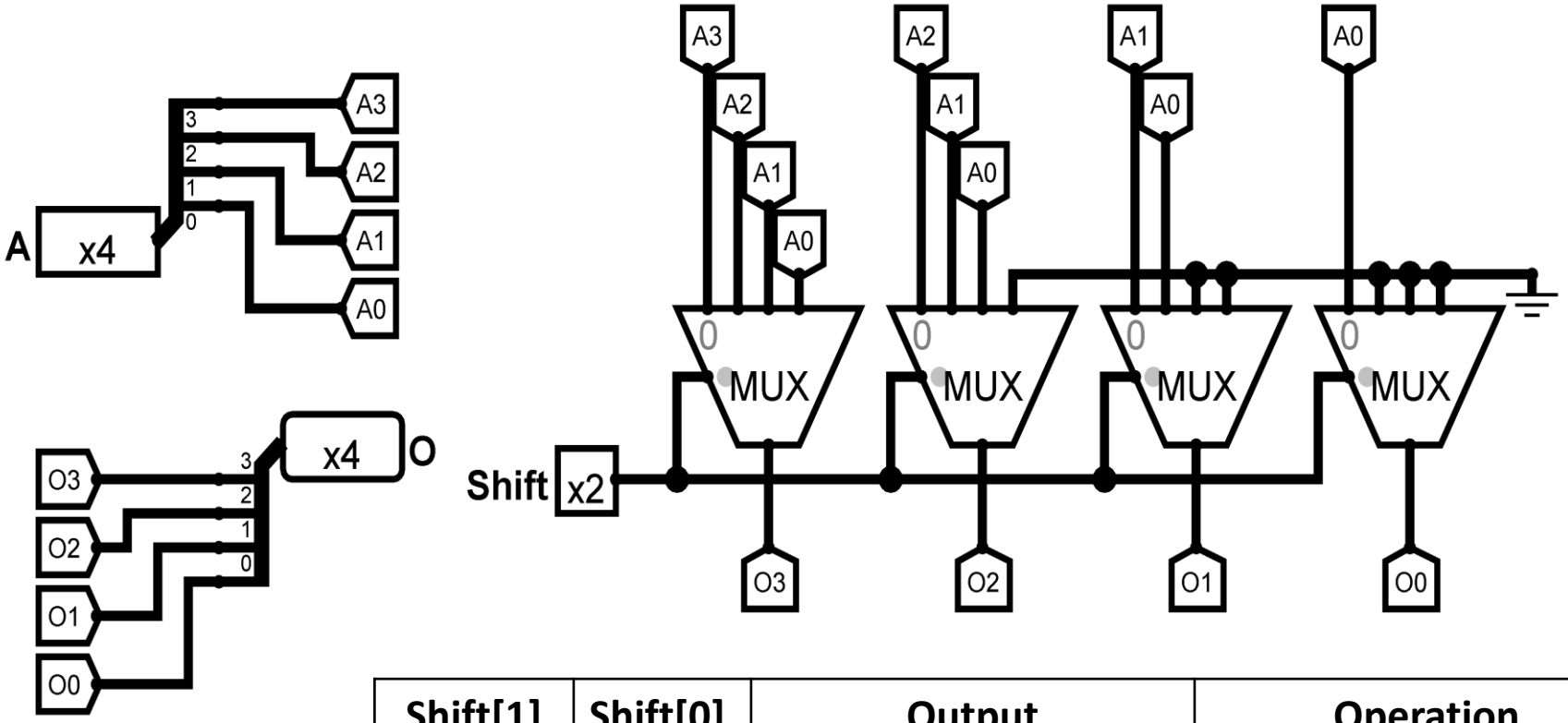


Shifter and Rotate Circuits
Shifter
Rotate (Barrel Shifter)

4-bit Left Shifter



4-bit Left Shifter



Shift[1]	Shift[0]	Output	Operation
0	0	**A[3]A[2]A[1]A[0]**	No Shift
0	1	**A[2]A[1]A[0]0**	1 bit Left Shift
1	0	**A[1]A[0]00**	2 bit Left Shift
1	1	**A[0]000**	3 bit Left Shift

4-bit Left Shifter Simulation

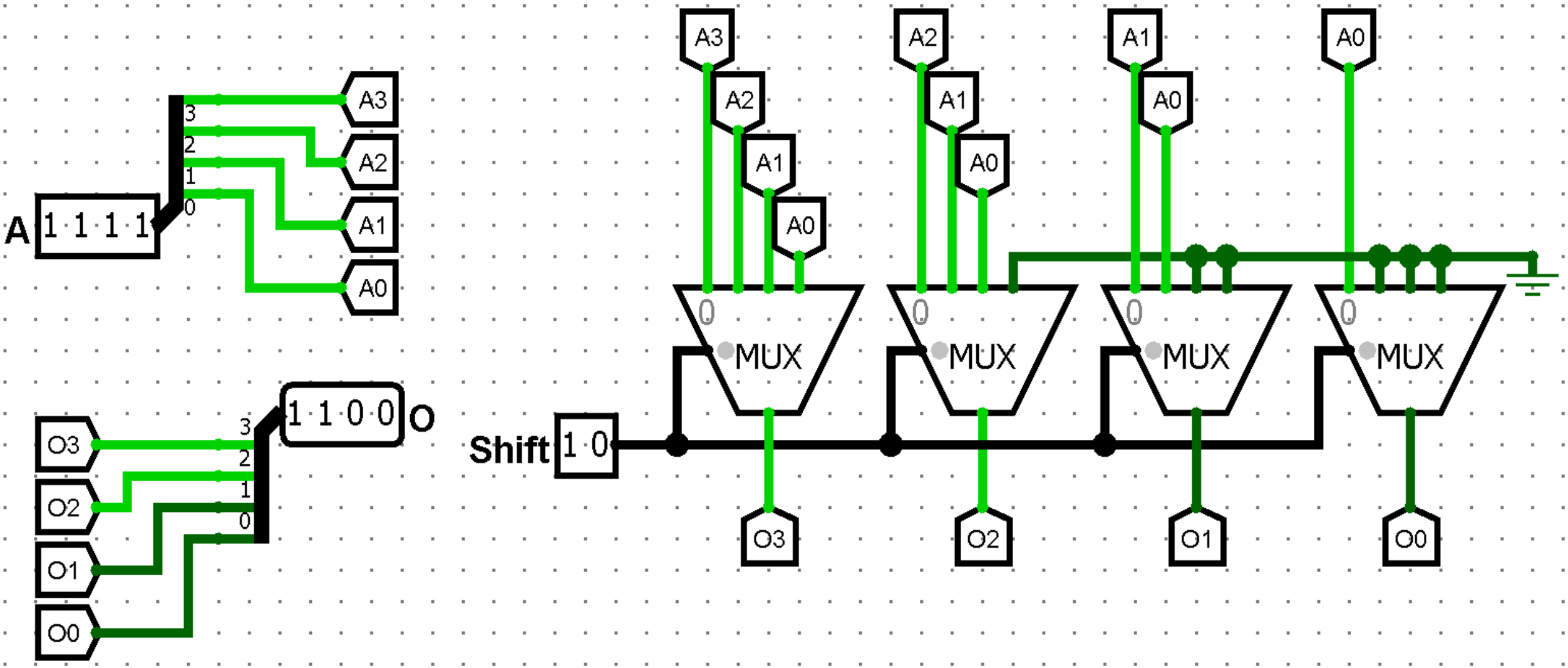
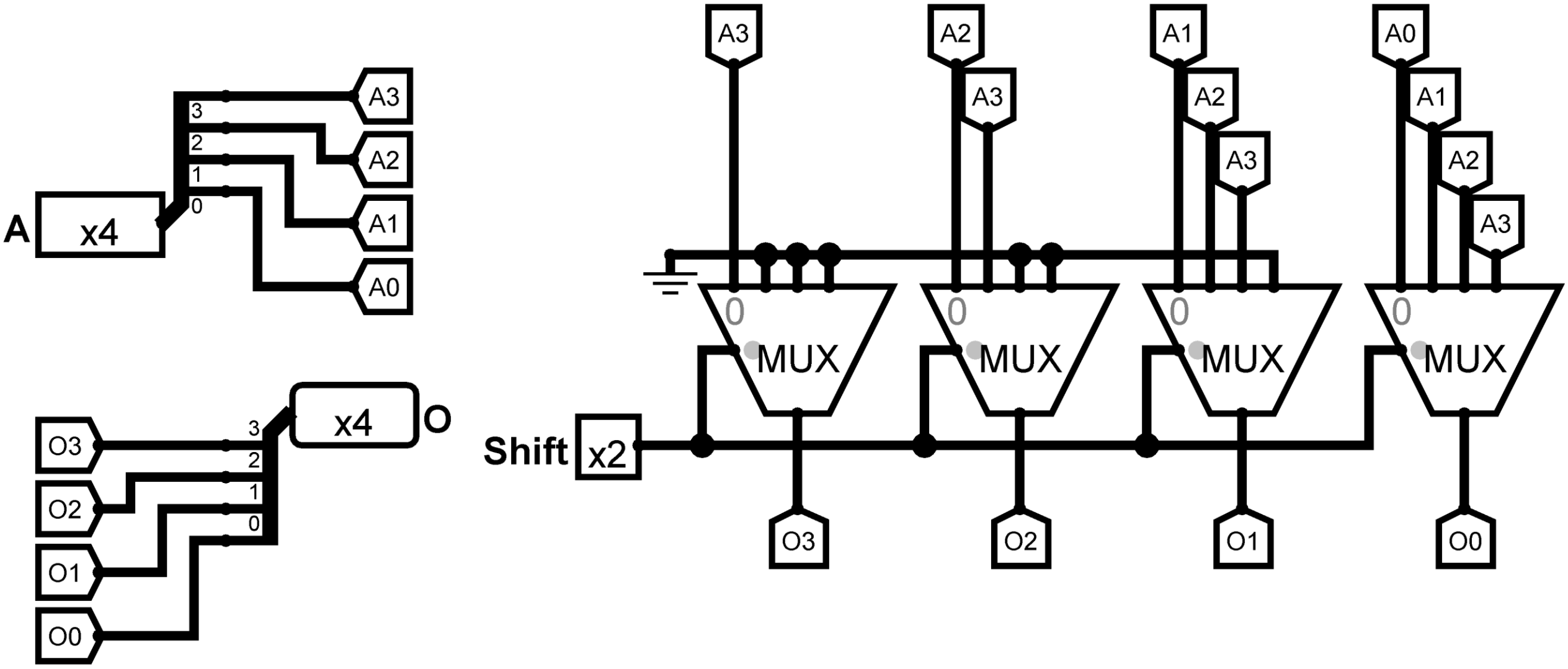
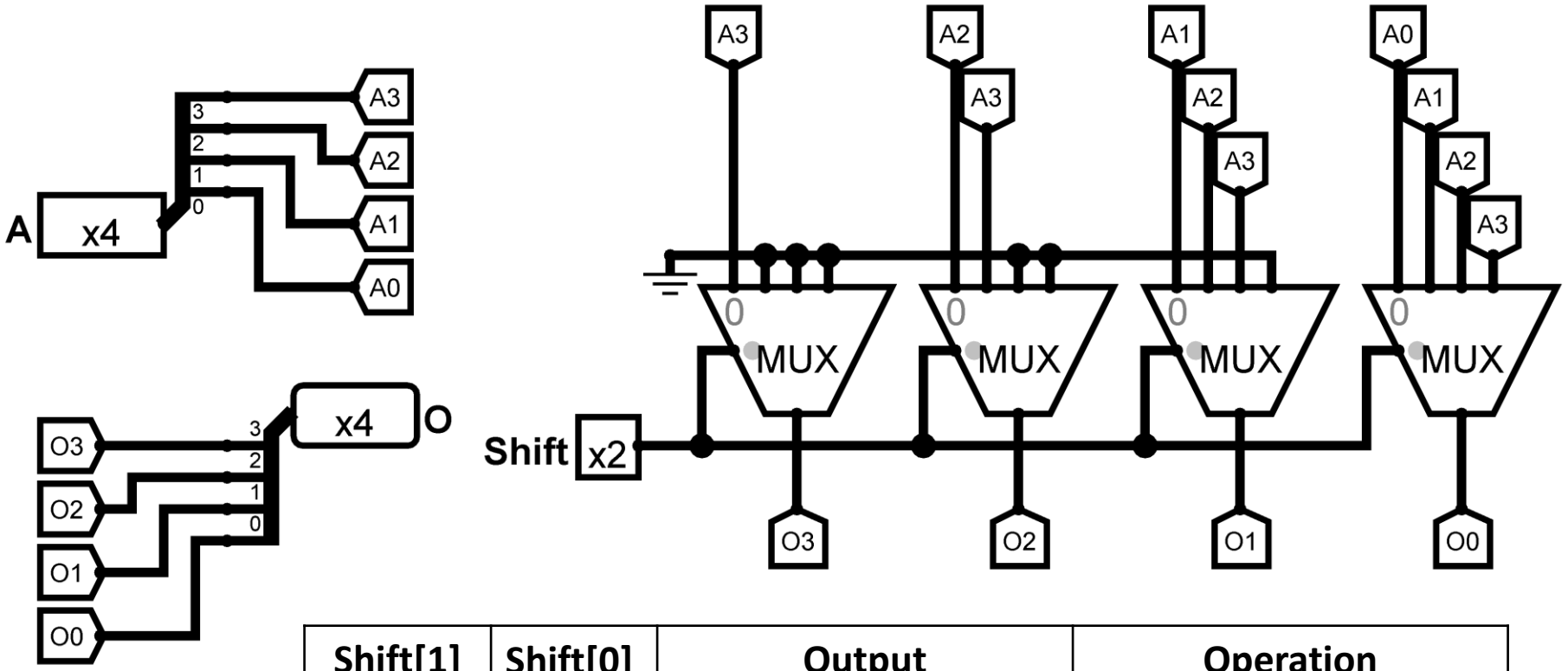


Figure: 4-bit Left Shifter Simulation for input A = 1111, Shift = 10 (2-bit Left Shift).
Output O = 1100

4-bit Right Shifter



4-bit Right Shifter



Shift[1]	Shift[0]	Output	Operation
0	0	**A[3]A[2]A[1]A[0]**	No Shift
0	1	**0A[3]A[2]A[1]**	1 bit Right Shift
1	0	**00A[3]A[2]**	2 bit Right Shift
1	1	**000A[3]**	3 bit Right Shift

4-bit Right Shifter Simulation

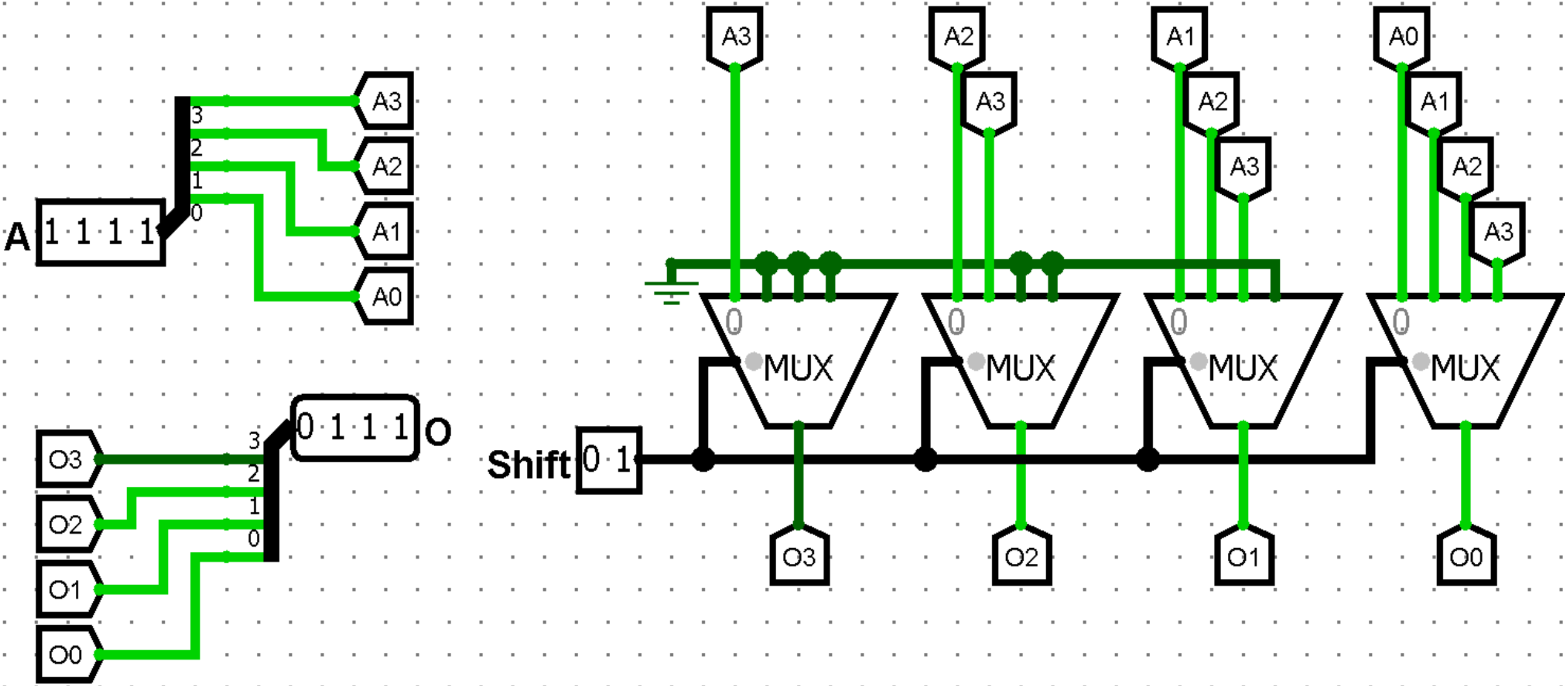
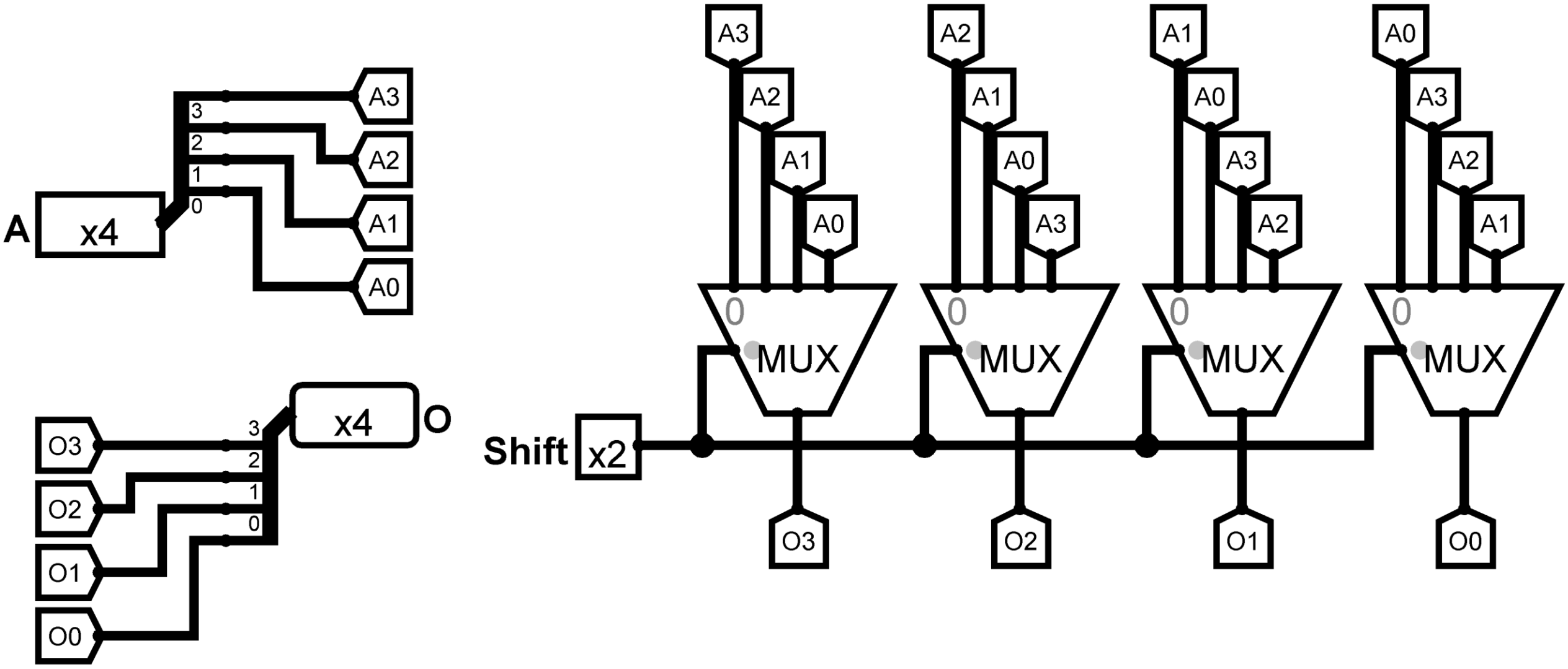
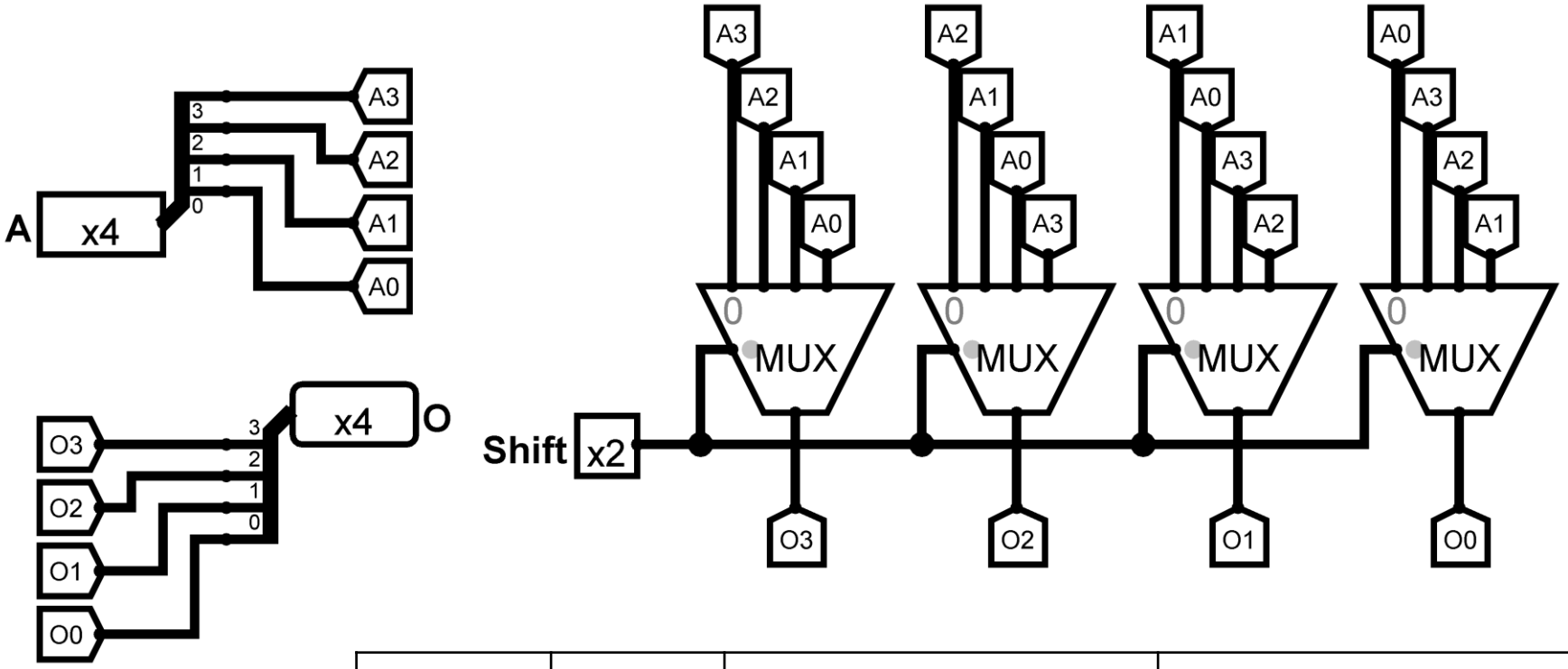


Figure: 4-bit Right Shifter Simulation for input A = 1111, Shift = 01 (1-bit Right Shift).
Output O = 0111

4-bit Left Rotate



4-bit Left Rotate



Shift[1]	Shift[0]	Output	Operation
0	0	**A[3]A[2]A[1]A[0]**	No Rotate
0	1	**A[2]A[1]A[0]A[3]**	1 bit Left Rotate
1	0	**A[1]A[0]A[3]A[2]**	2 bit Left Rotate
1	1	**A[0]A[3]A[2]A[1]**	3 bit Left Rotate

4-bit Left Rotate Simulation

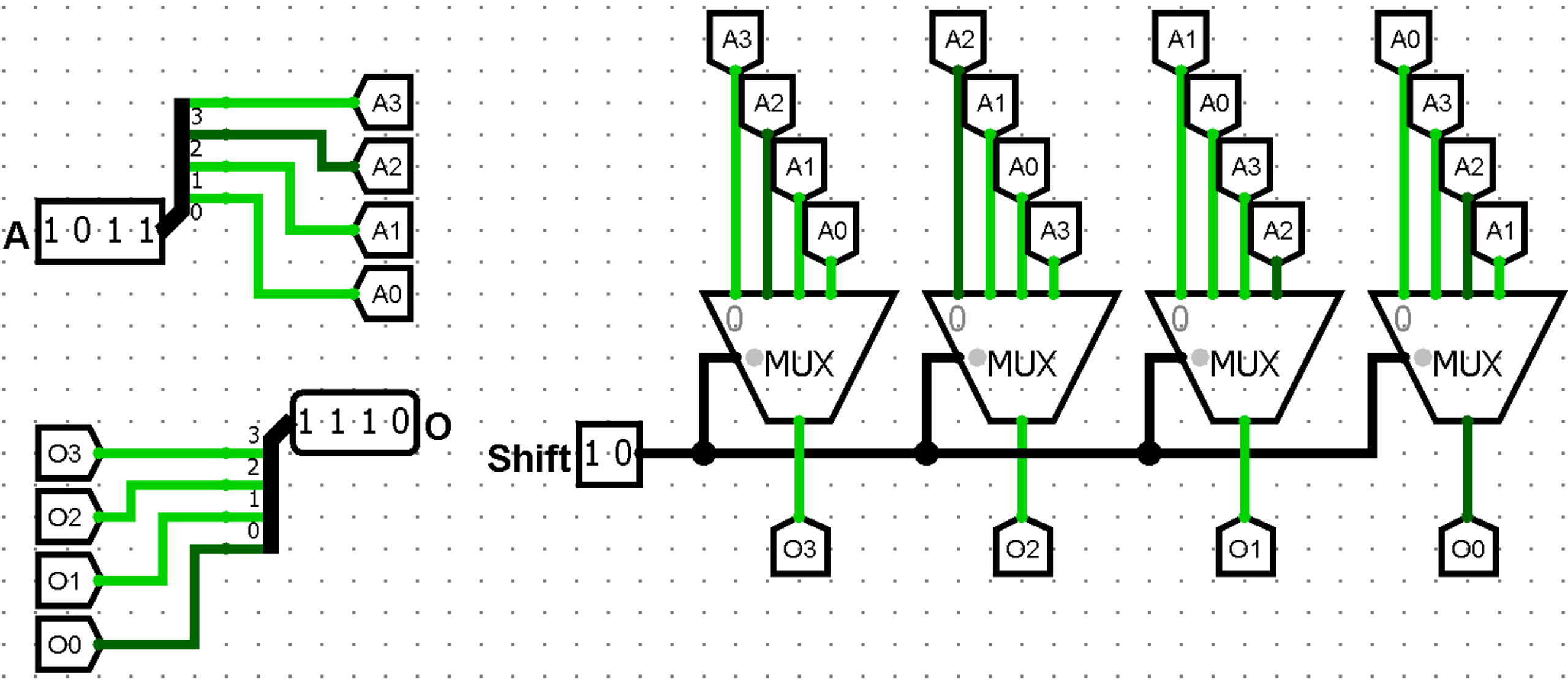
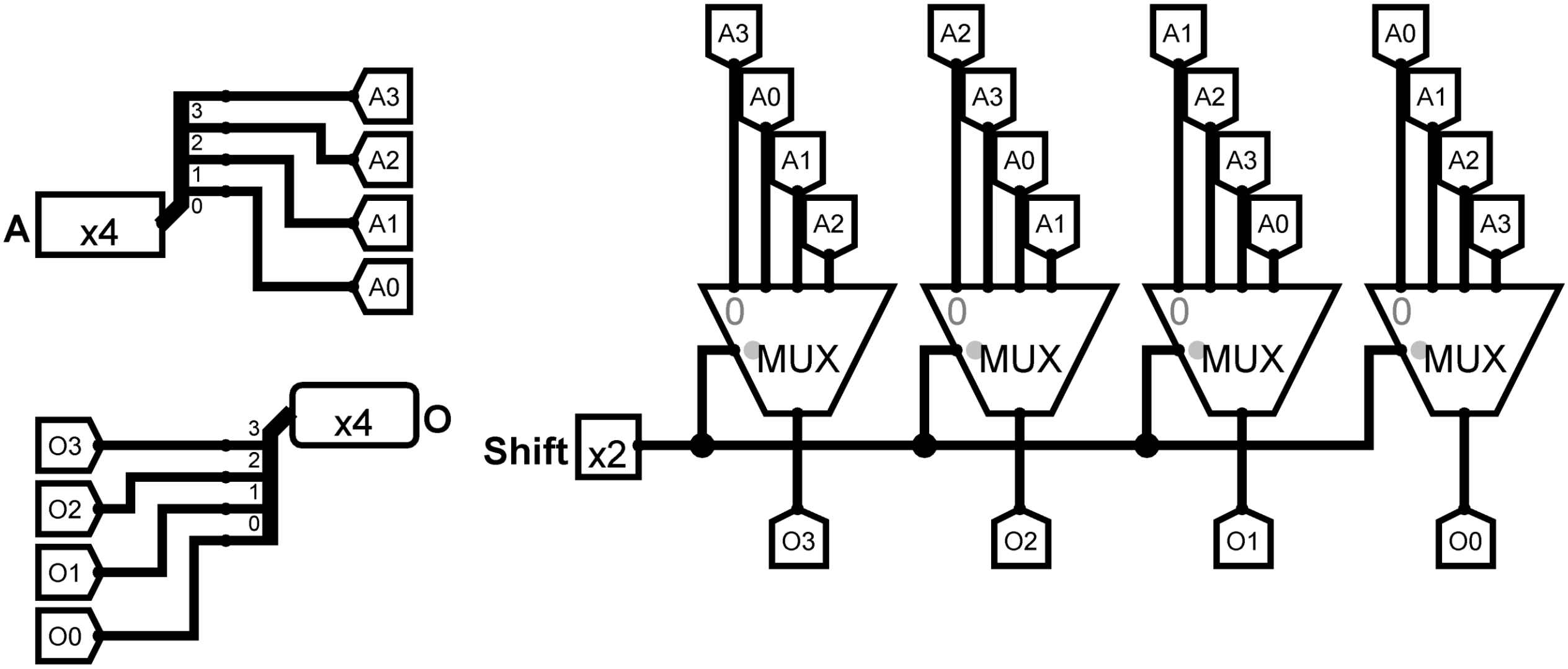
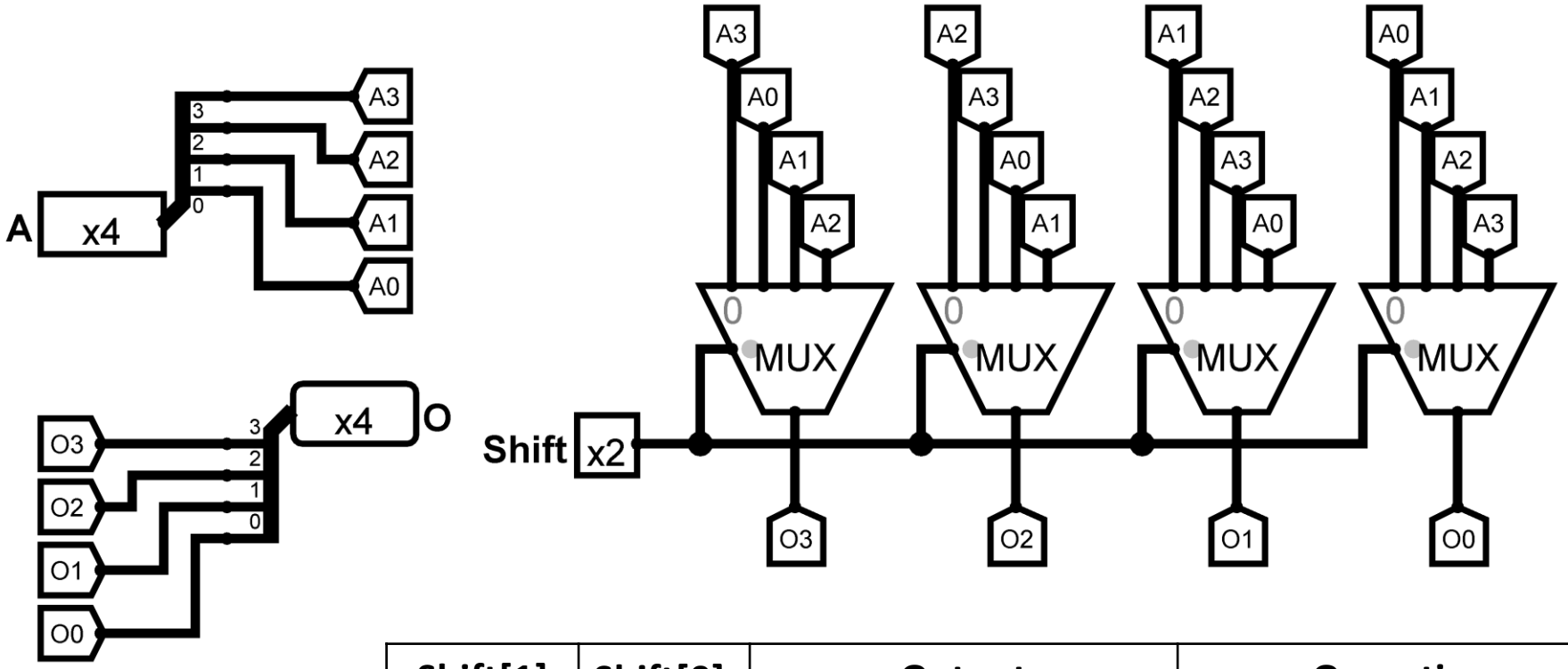


Figure: 4-bit Left Rotate Simulation for input A = 1111, Shift = 10 (1-bit Left Rotate). Output O = 1110

4-bit Right Rotate



4-bit Right Rotate



Shift[1]	Shift[0]	Output	Operation
0	0	**A[3]A[2]A[1]A[0]**	No Rotate
0	1	**A[0]A[3]A[2]A[1]**	1 bit Right Rotate
1	0	**A[1]A[0]A[3]A[2]**	2 bit Right Rotate
1	1	**A[2]A[1]A[0]A[3]**	3 bit Right Rotate

4-bit Right Rotate Simulation

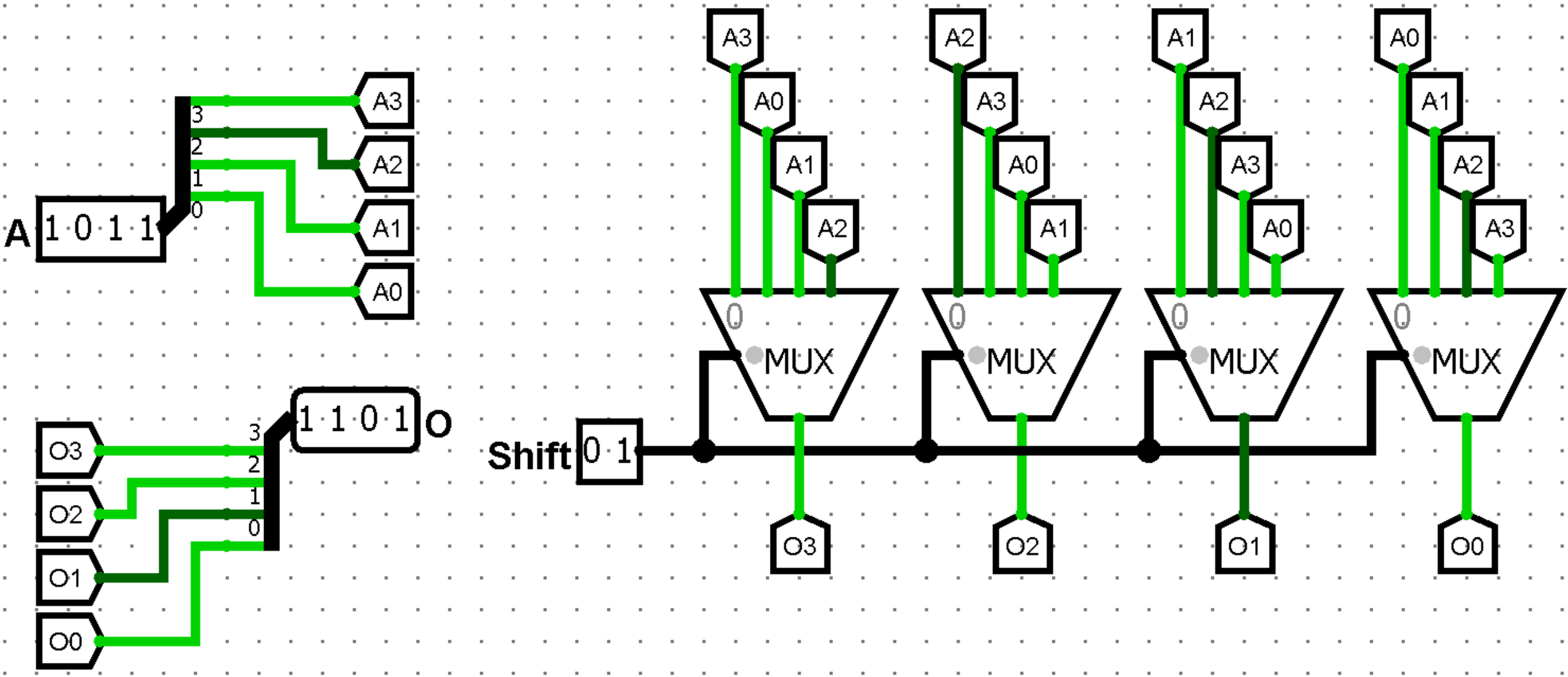
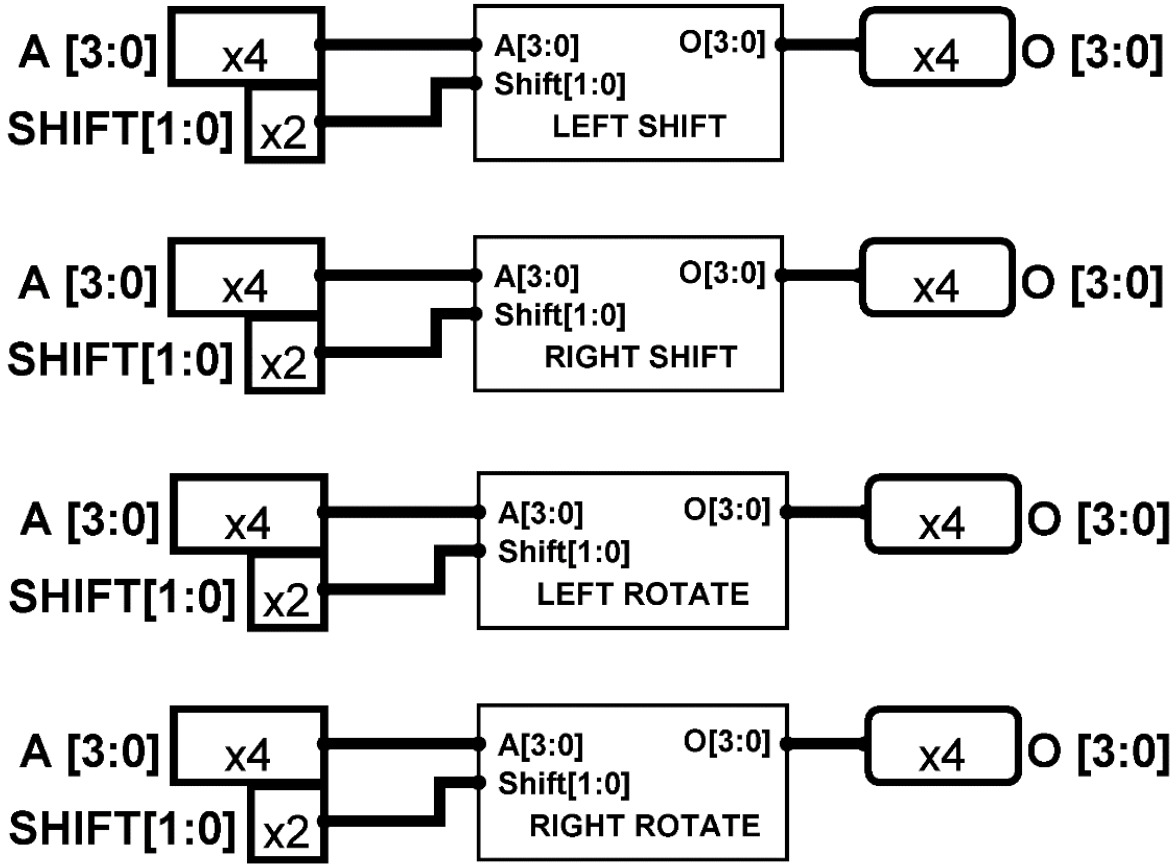
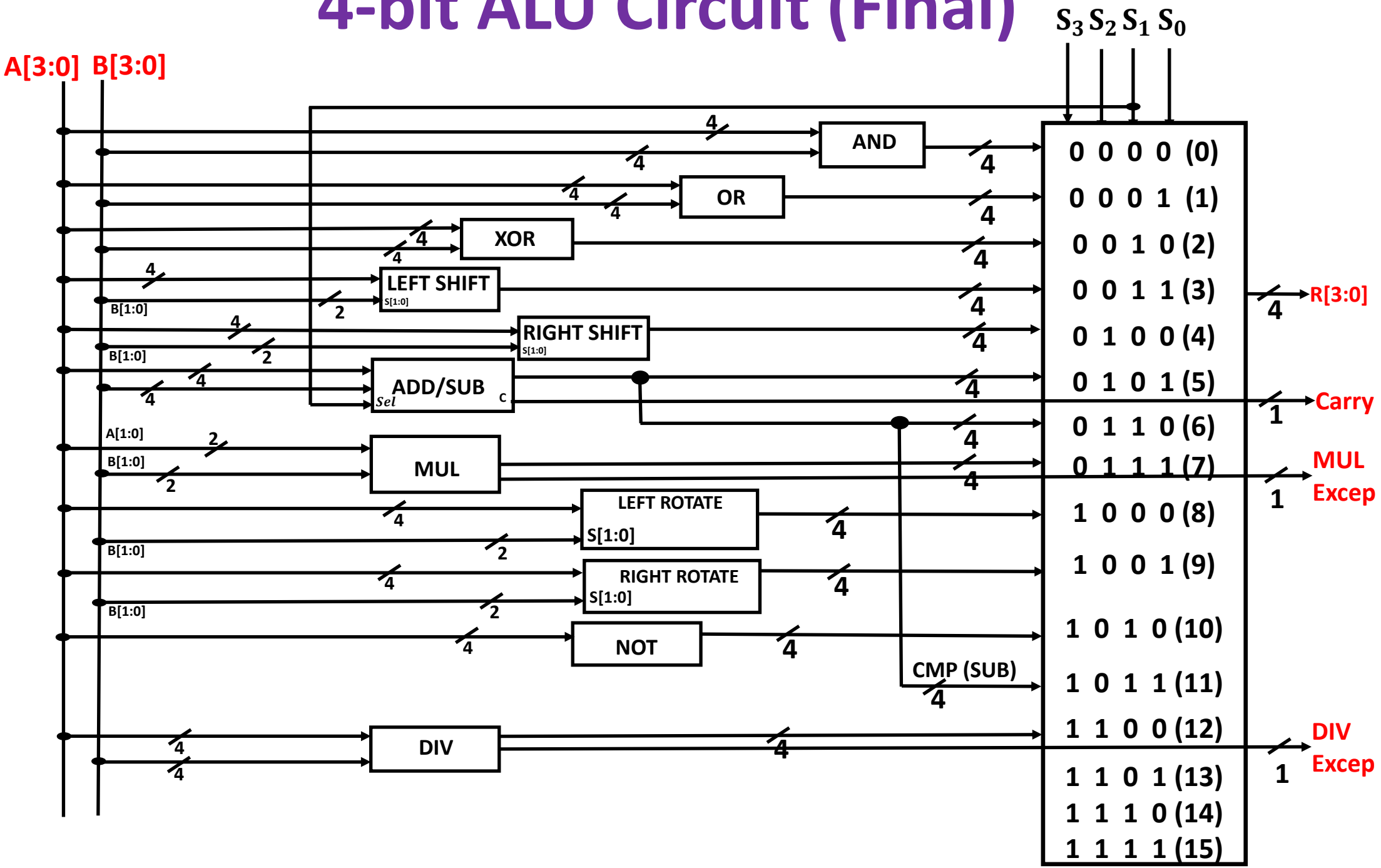


Figure: 4-bit Right Rotate Simulation for input A = 1111, Shift = 10 (1-bit Right Rotate). Output O = 1110

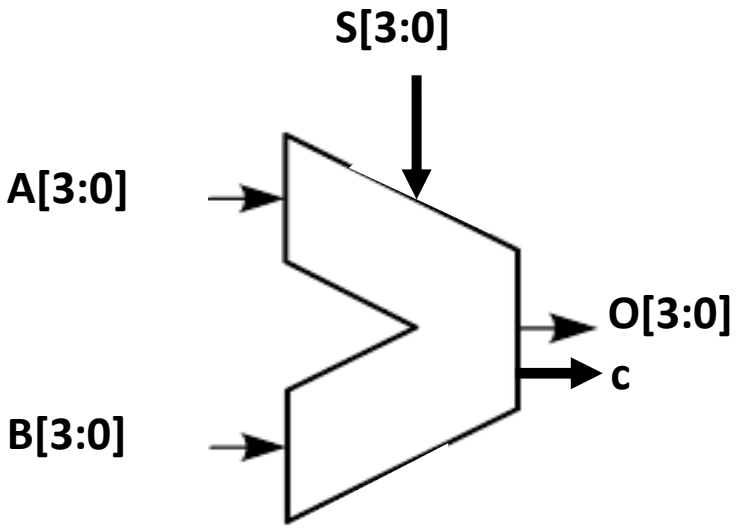
Shift and Rotate Circuits Block



4-bit ALU Circuit (Final)



4-bit ALU Circuit

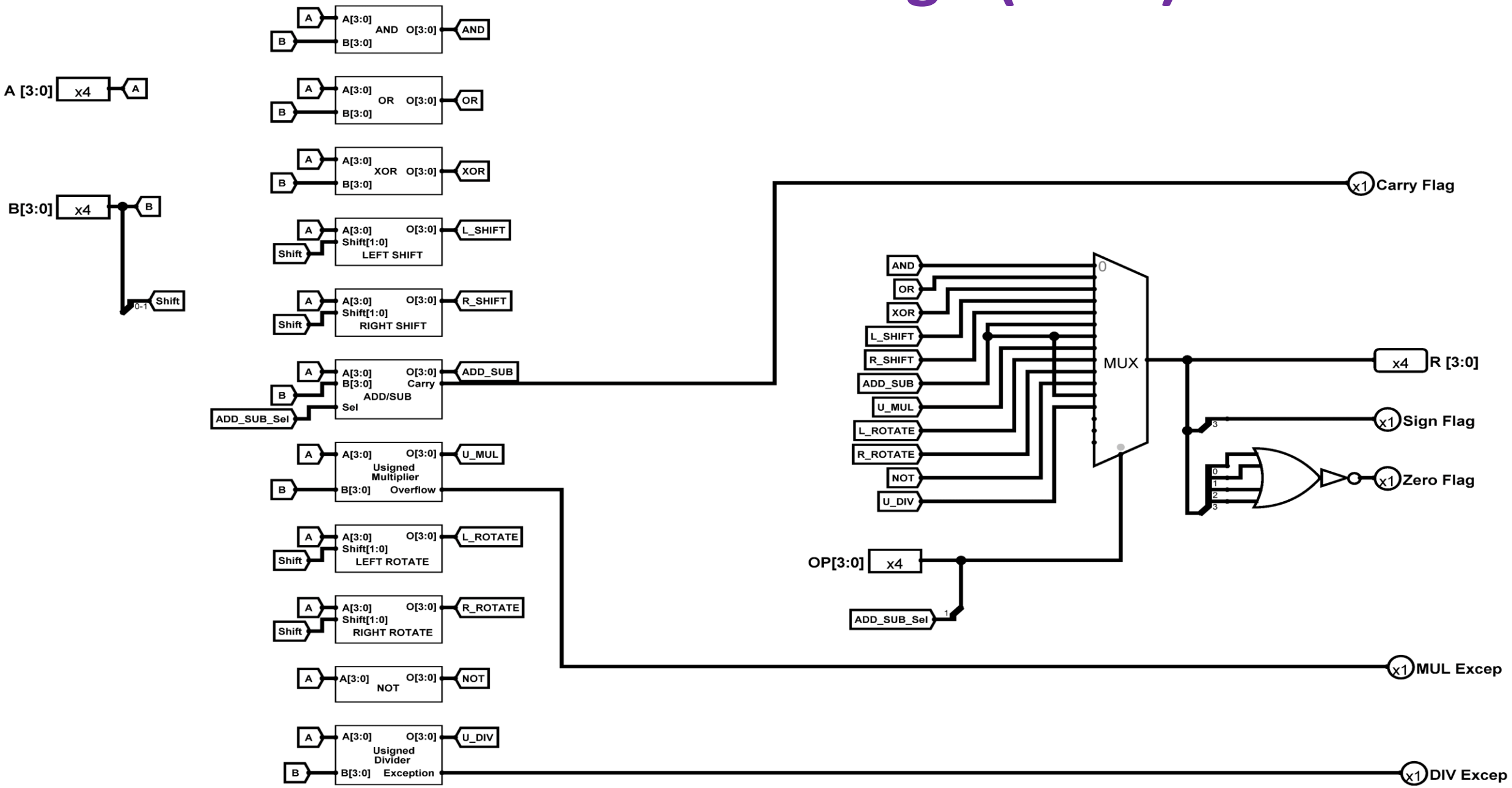


Operation	Selection lines			
	S_3	S_2	S_1	S_0
AND	0	0	0	0
OR	0	0	0	1
XOR	0	0	1	0
NOT	1	0	1	0
ADD	0	1	0	1
SUB	0	1	1	0
MUL	0	1	1	1
DIV	1	1	0	0
SHL	0	0	1	1
SHR	0	1	0	0
ROL	1	0	0	0
ROR	1	0	0	1
CMP	1	0	1	1

Here,
A[3:0] is data to be shifted or rotated.
And B[1:0] is number of shift/rotate (Max 3).

Here, CMP is same as SUB but it doesn't update register value.

4-bit ALU Design (Final)



4-bit ALU Simulation

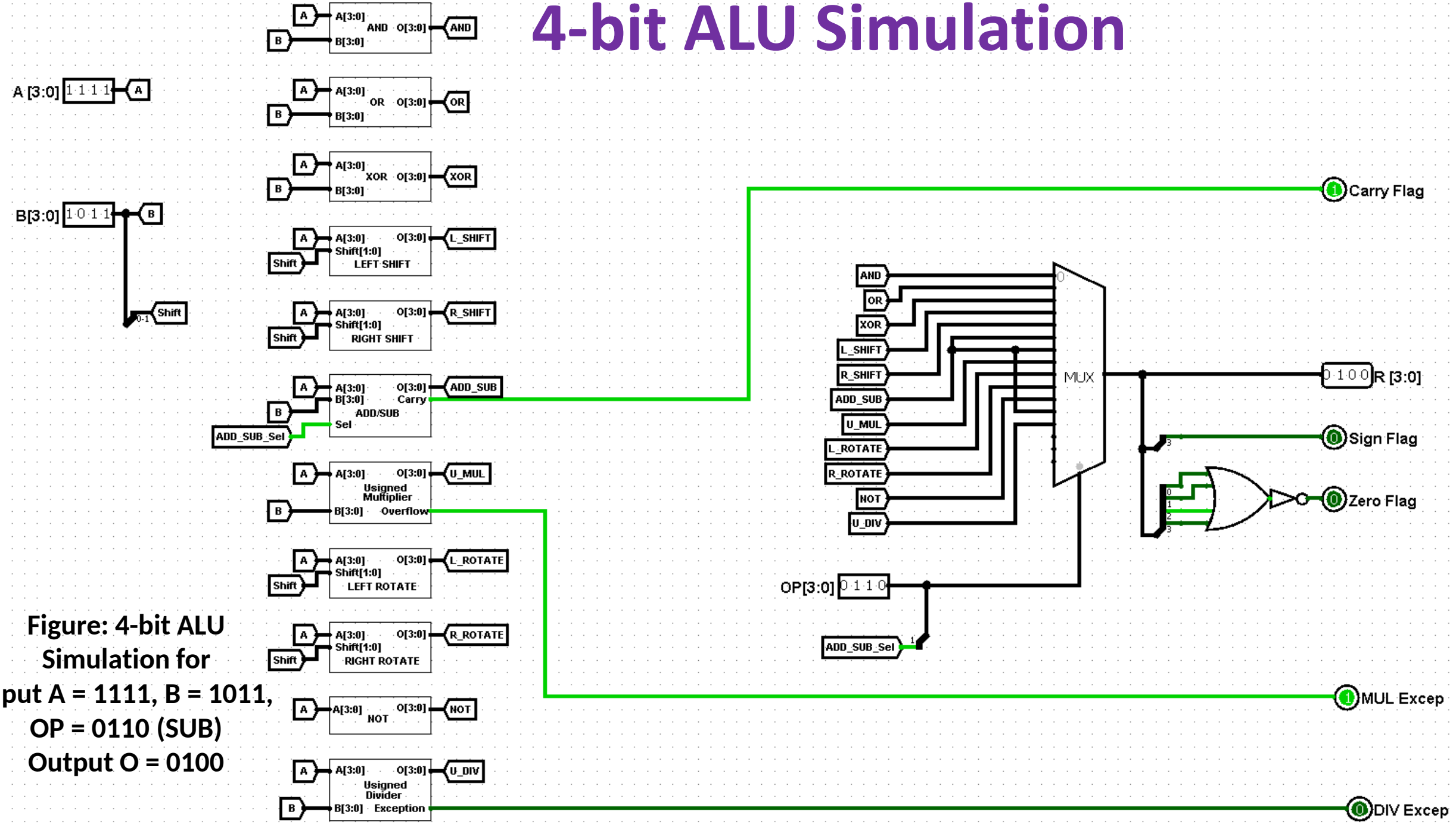


Figure: 4-bit ALU Simulation for input A = 1111, B = 1011, OP = 0110 (SUB) Output O = 0100

Thank You ☺