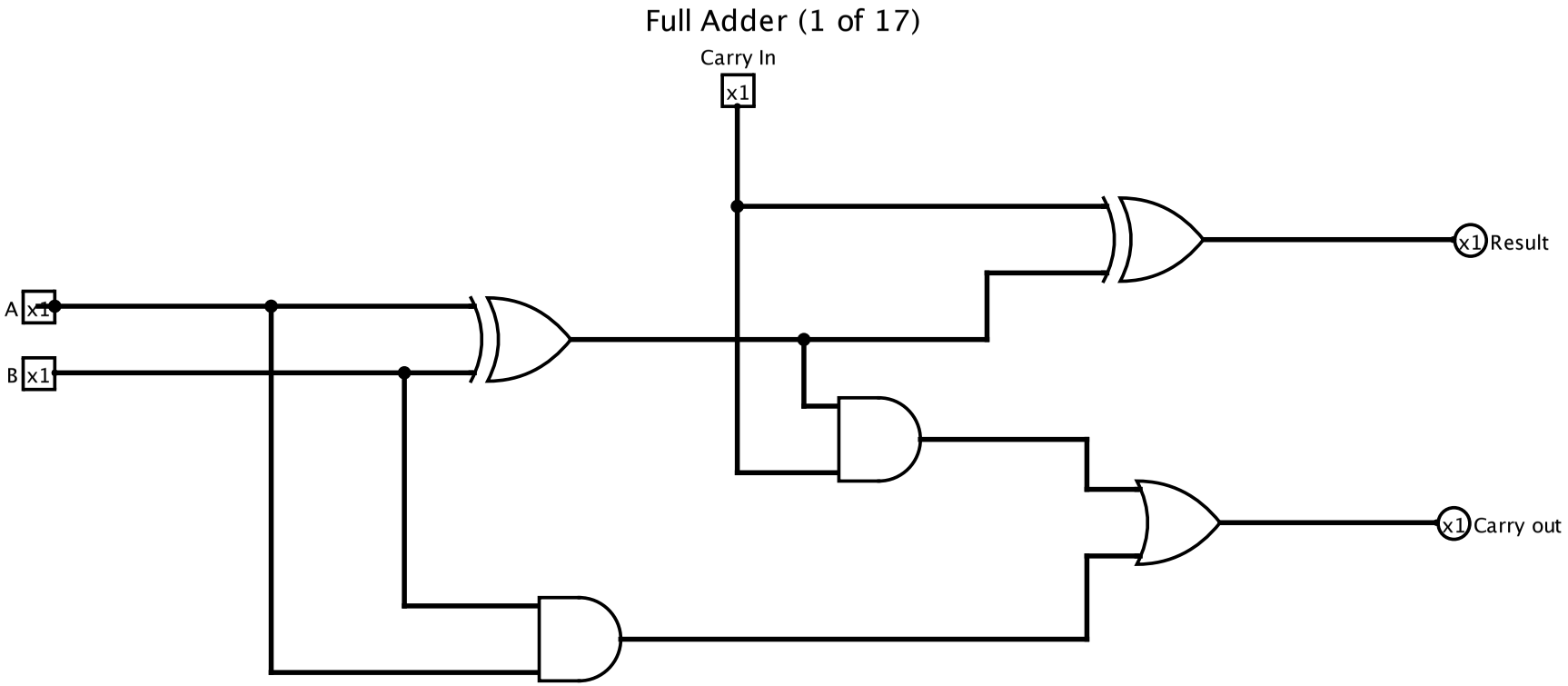
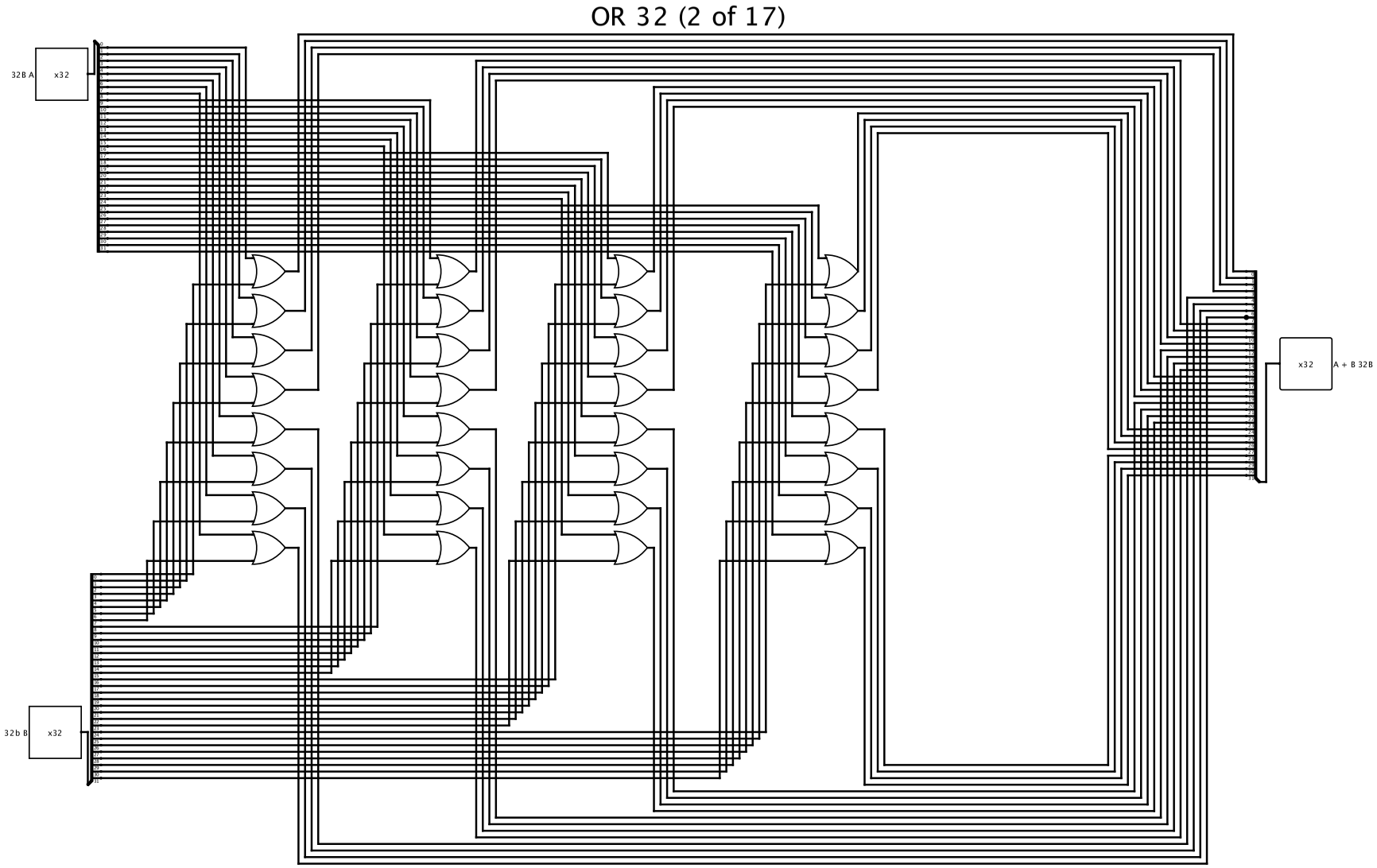
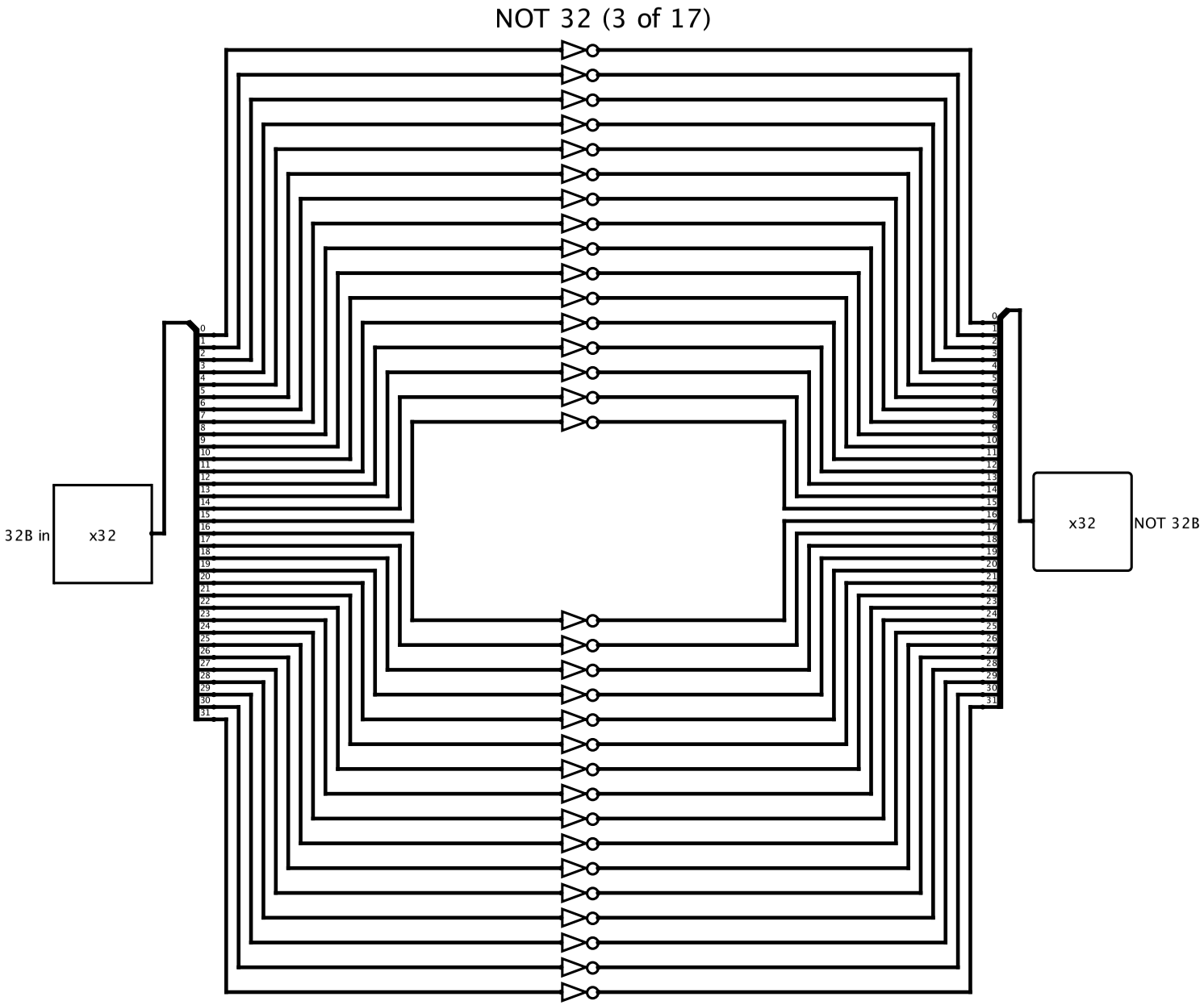
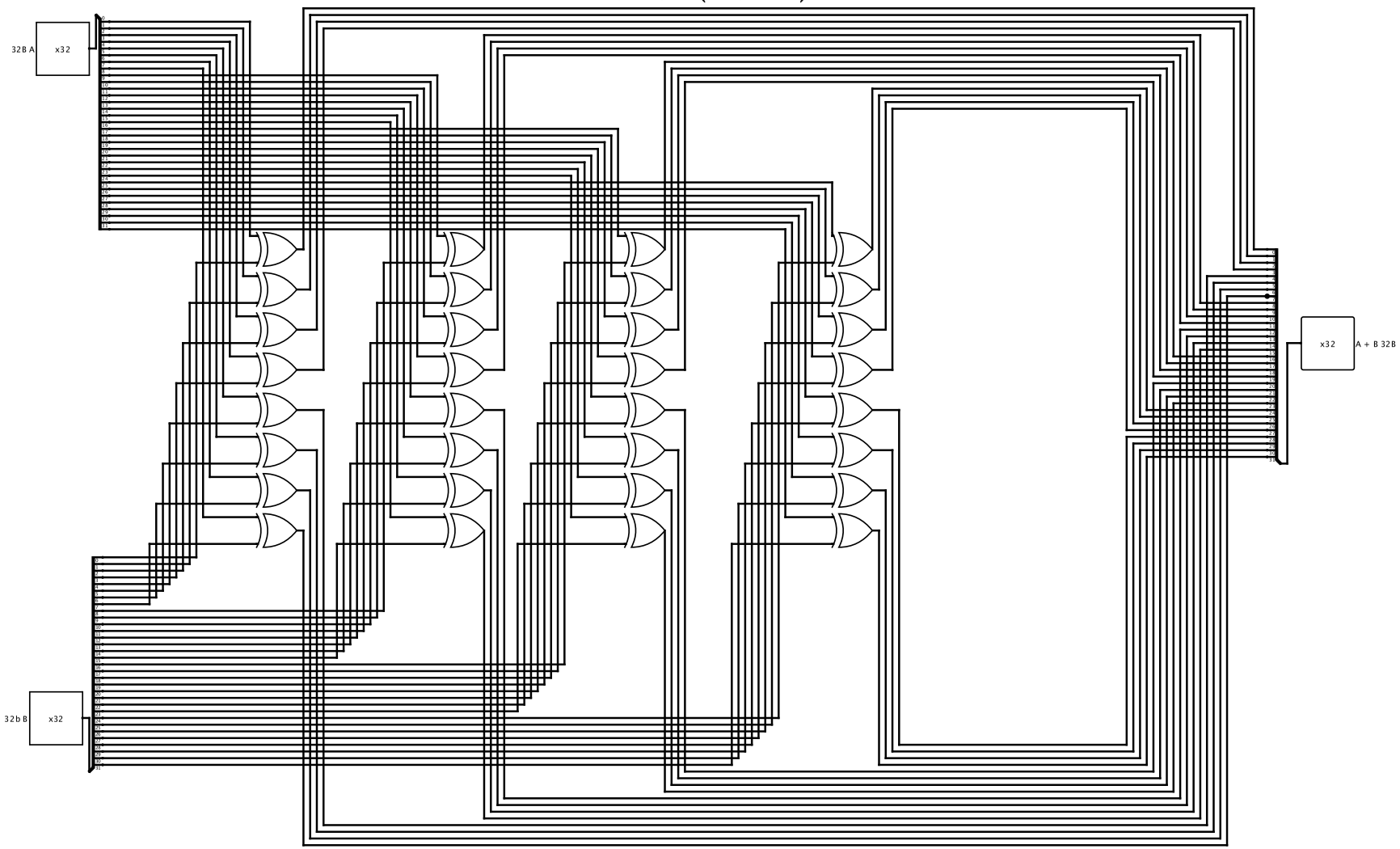

Full Adder (1 of 17)
Carry In
x1
A
x1
B
x1
x1
Result
x1
Carry out

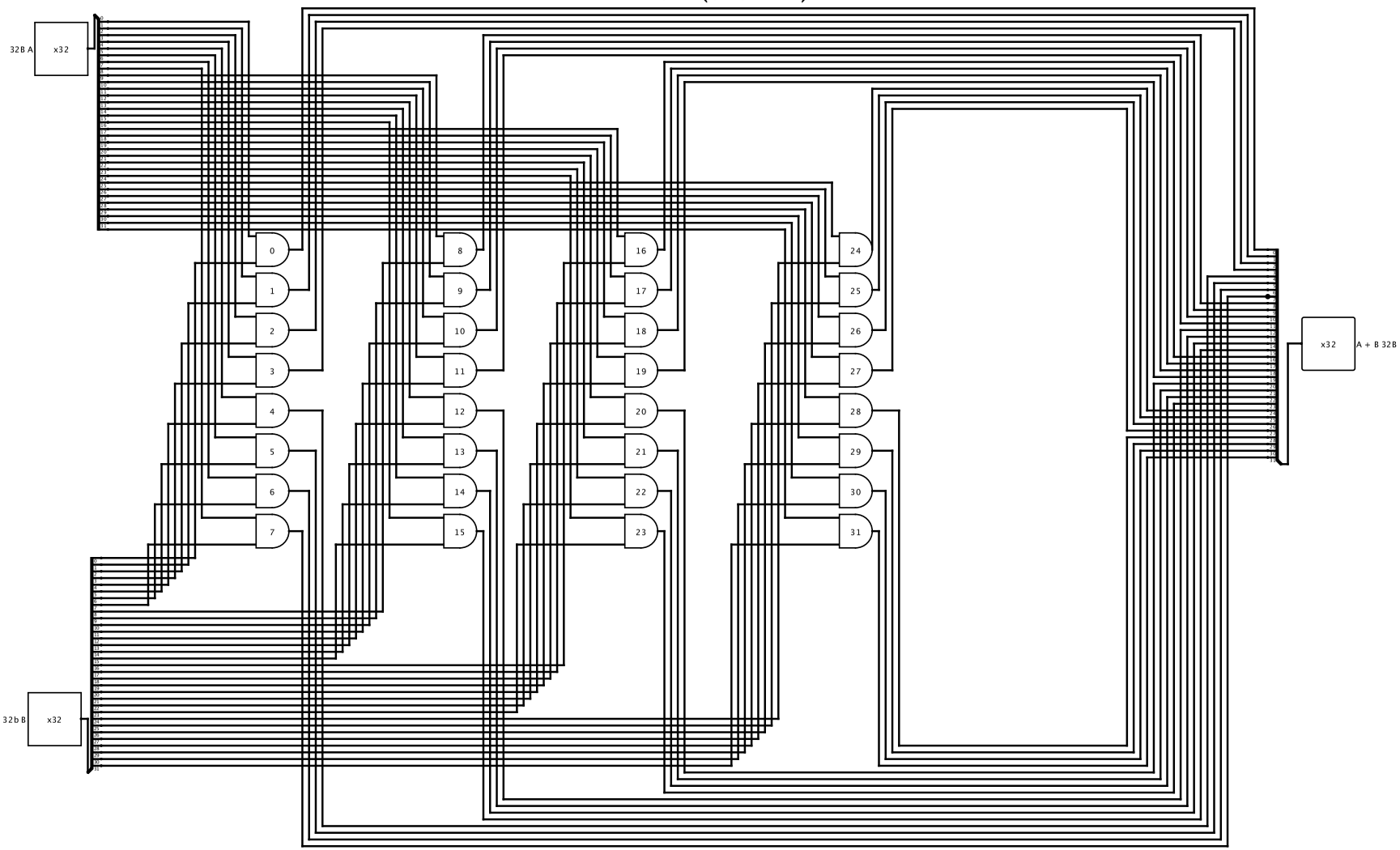

OR 32 (2 of 17)
32B A
x32
32b B
x32
x32
A + B 32B


NOT 32 (3 of 17)
32B in
x32
x32
NOT 32B

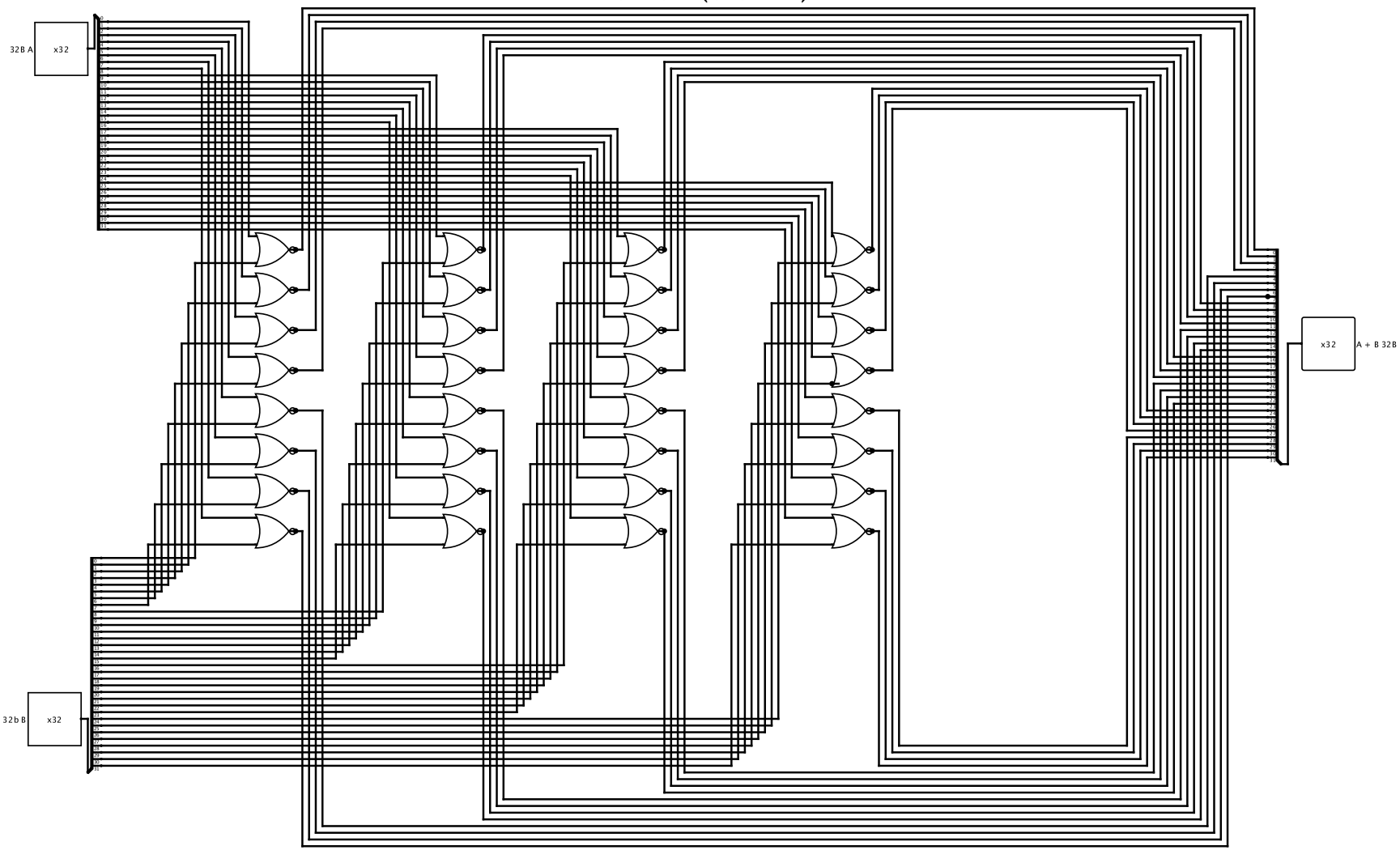
XOR 32 (4 of 17)



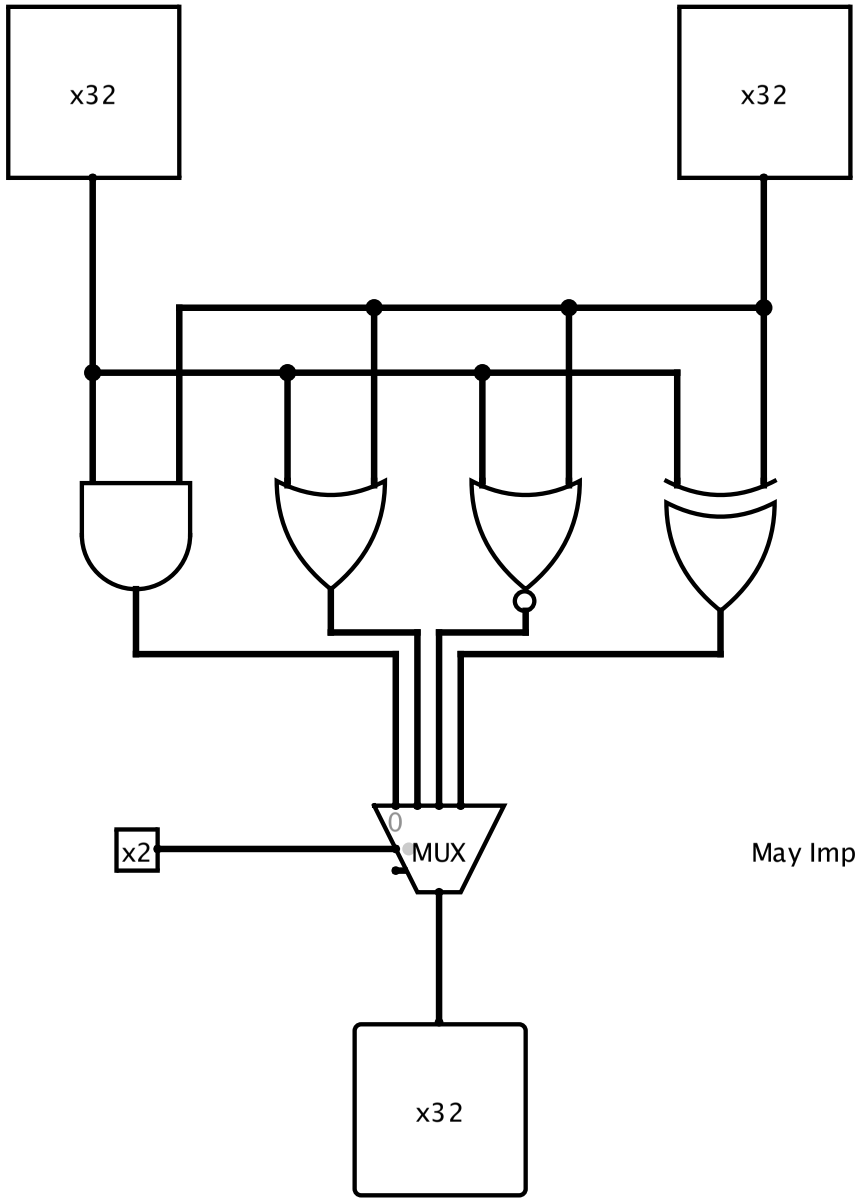
AND 32 (5 of 17)


32B A
x32
32b B
x32
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
x32
A + B 32B

NOR 32 (6 of 17)



Logical32 (7 of 17)



00 = AND

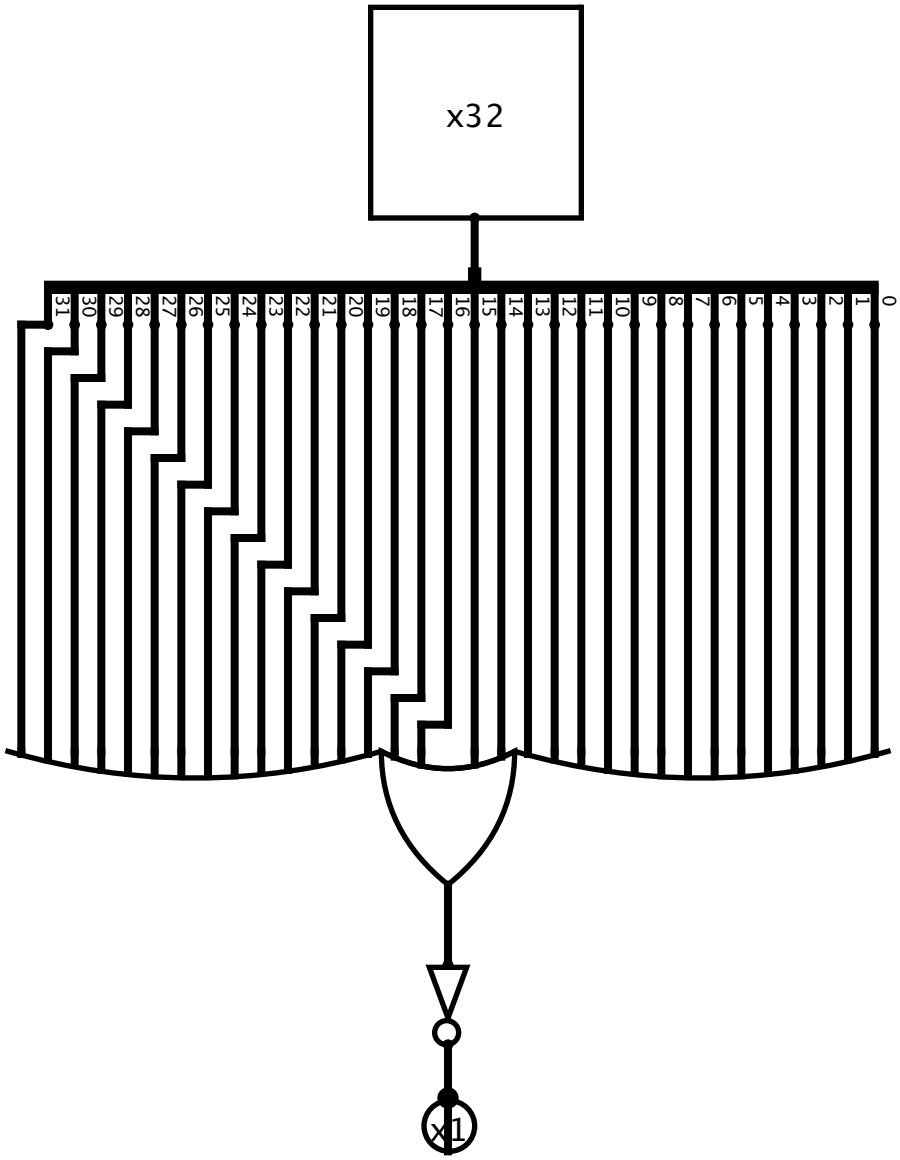
01 == OR

11 XOR

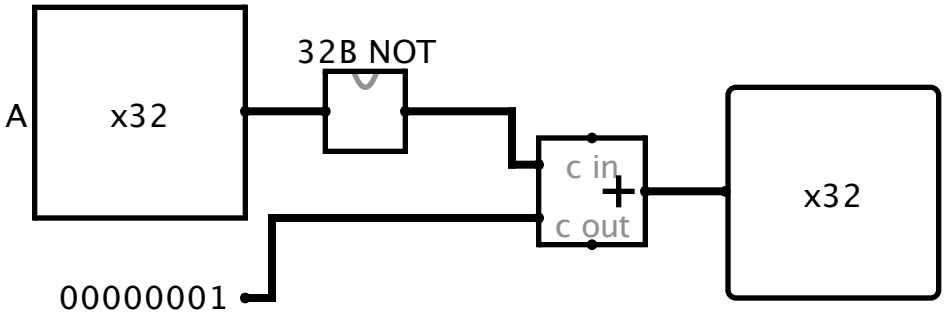
10 = NOR

May Implement Later for smaller ALU

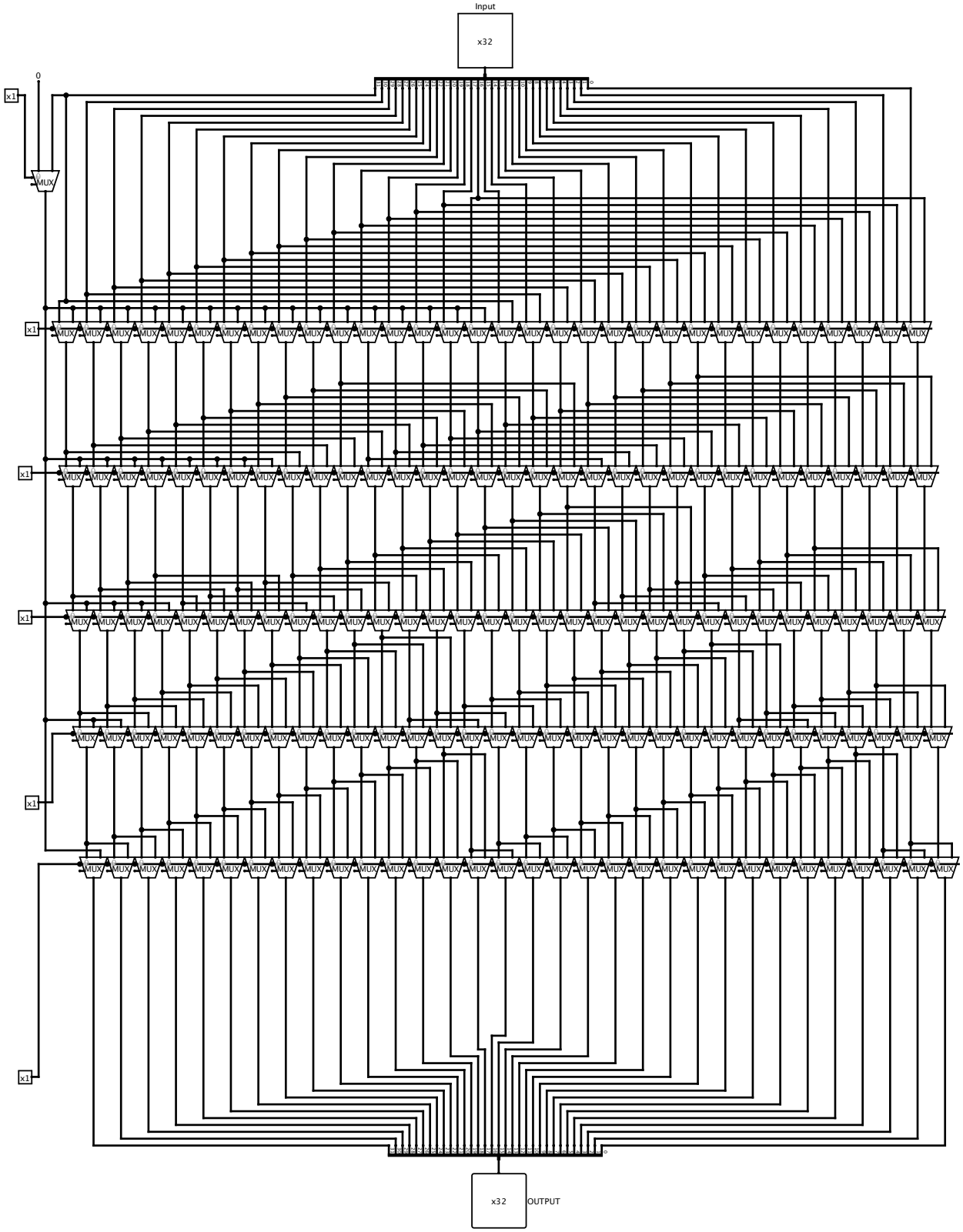
R_Zero (8 of 17)


x32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
x1

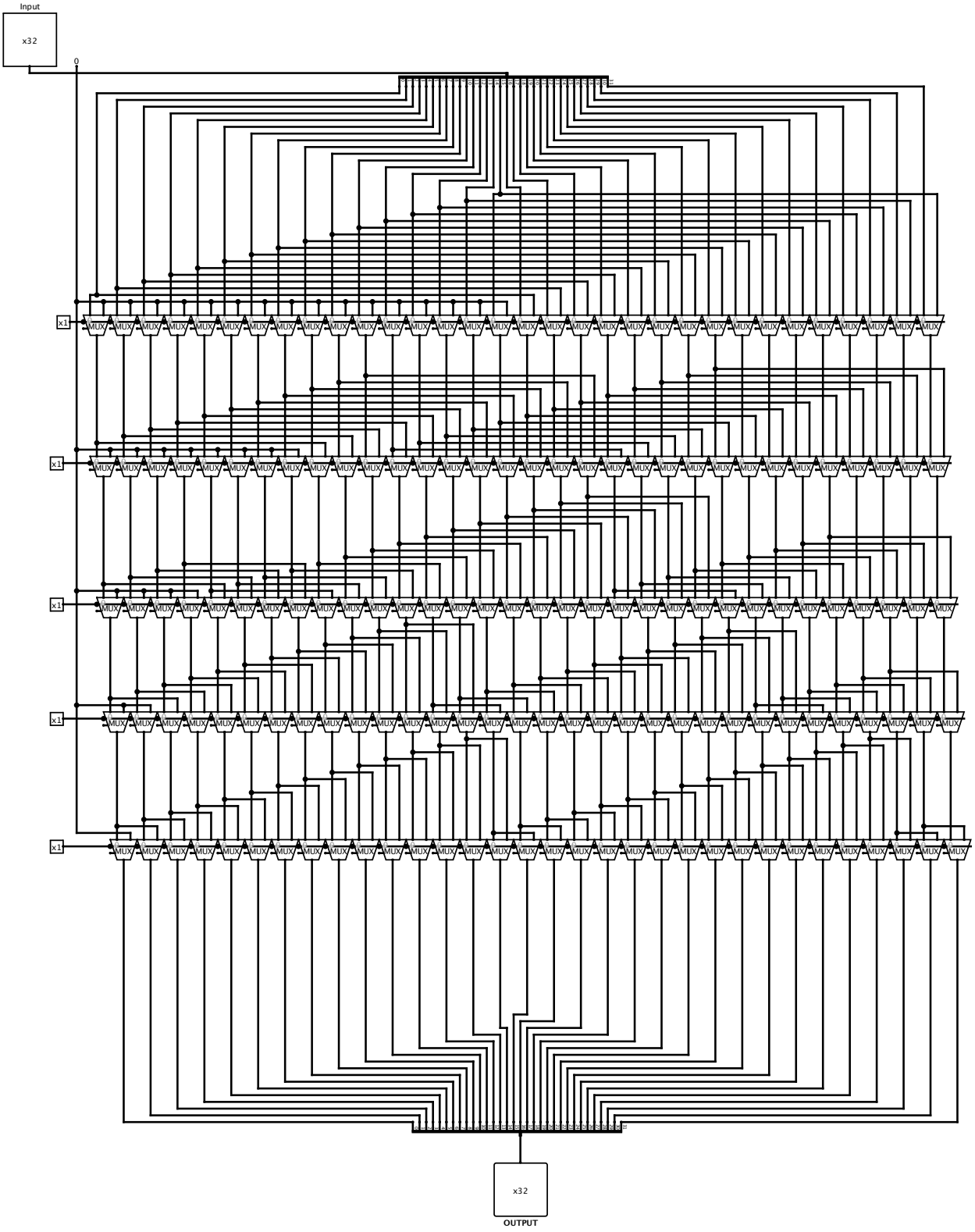
2's Complement (9 of 17)



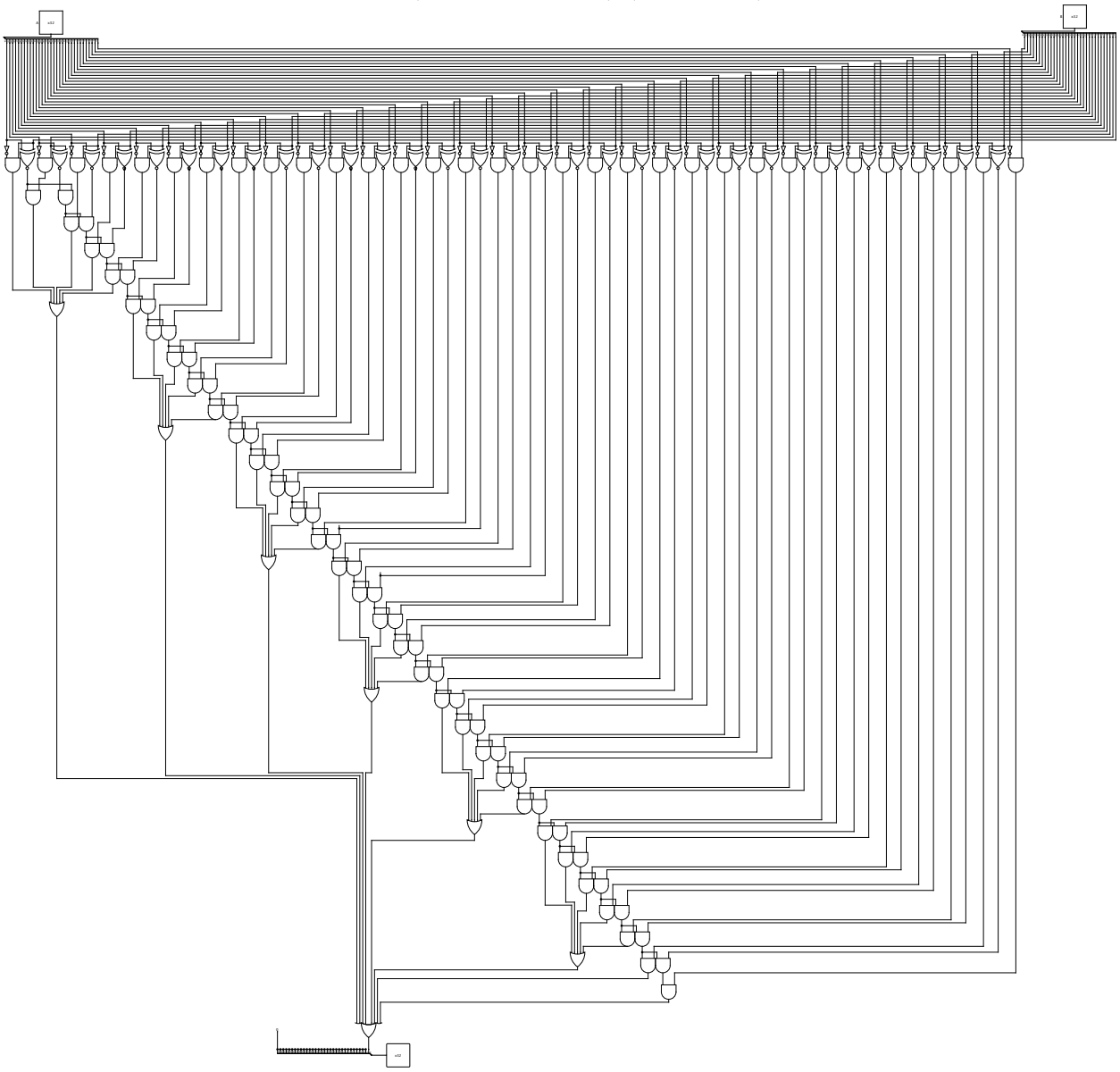
Shift Right (10 of 17)

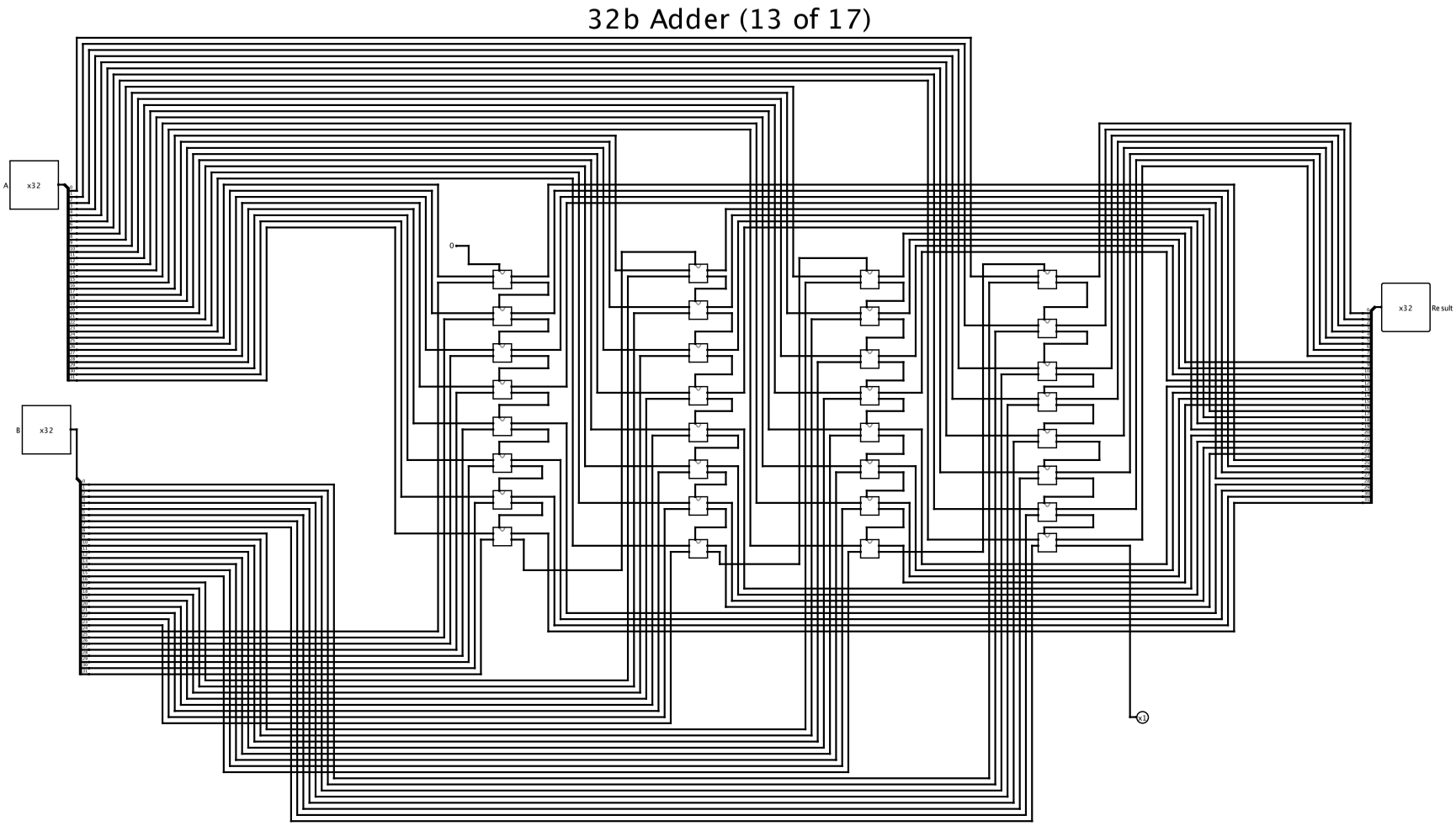


Shift Left (11 of 17)

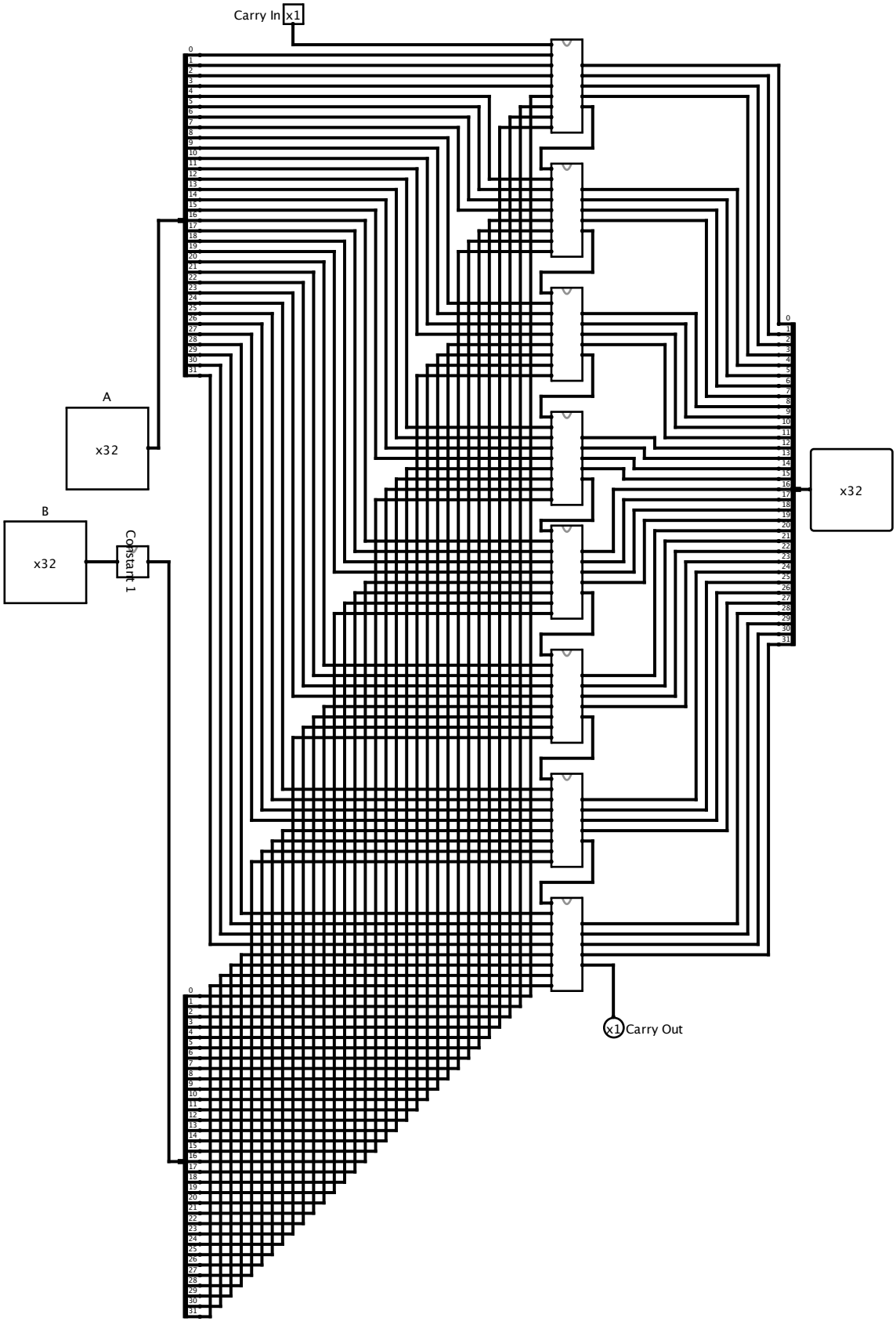


SLT (Set Less Than) (12 of 17)

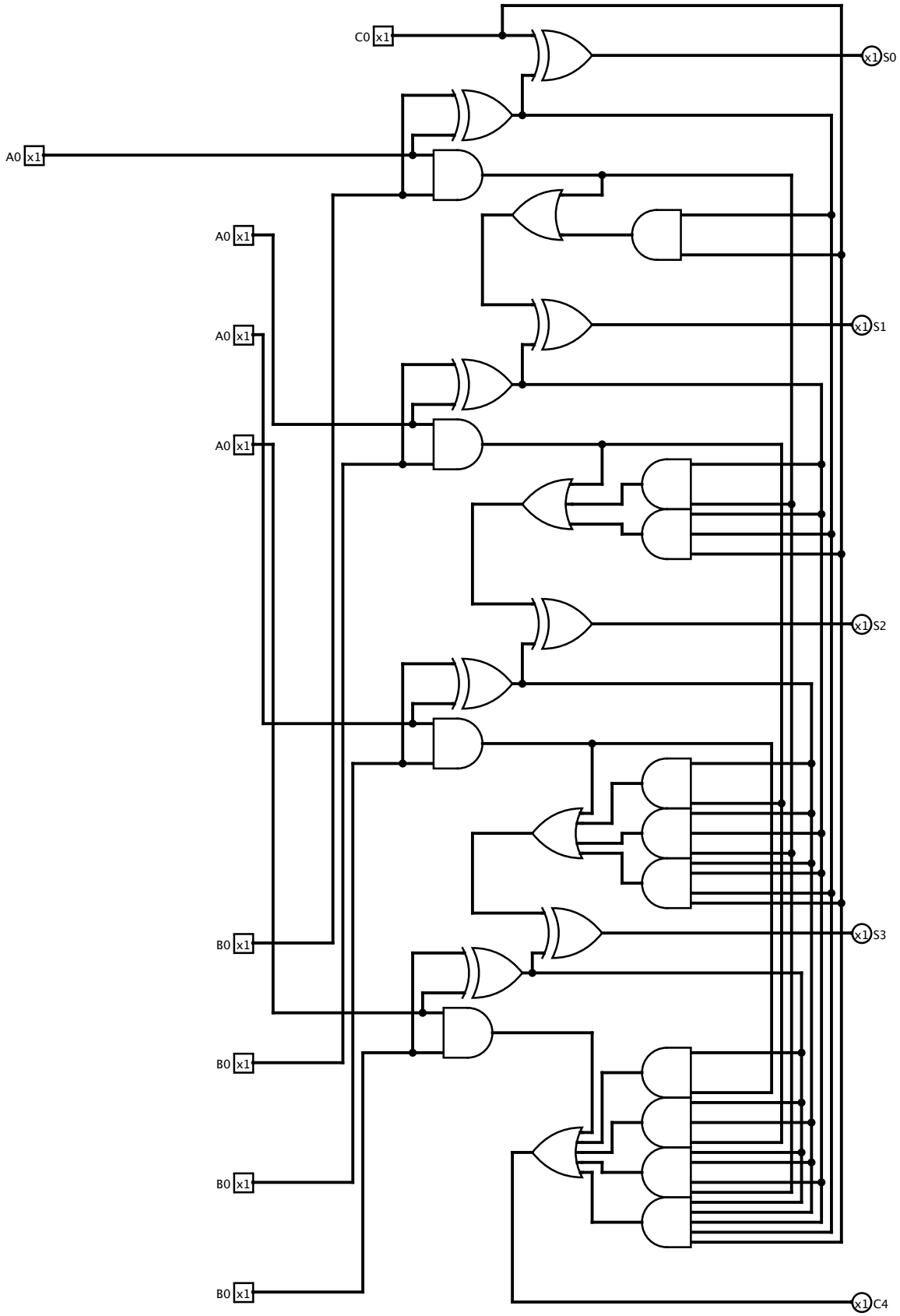



32b Adder (13 of 17)
A
x32
B
x32
0
x32
Result

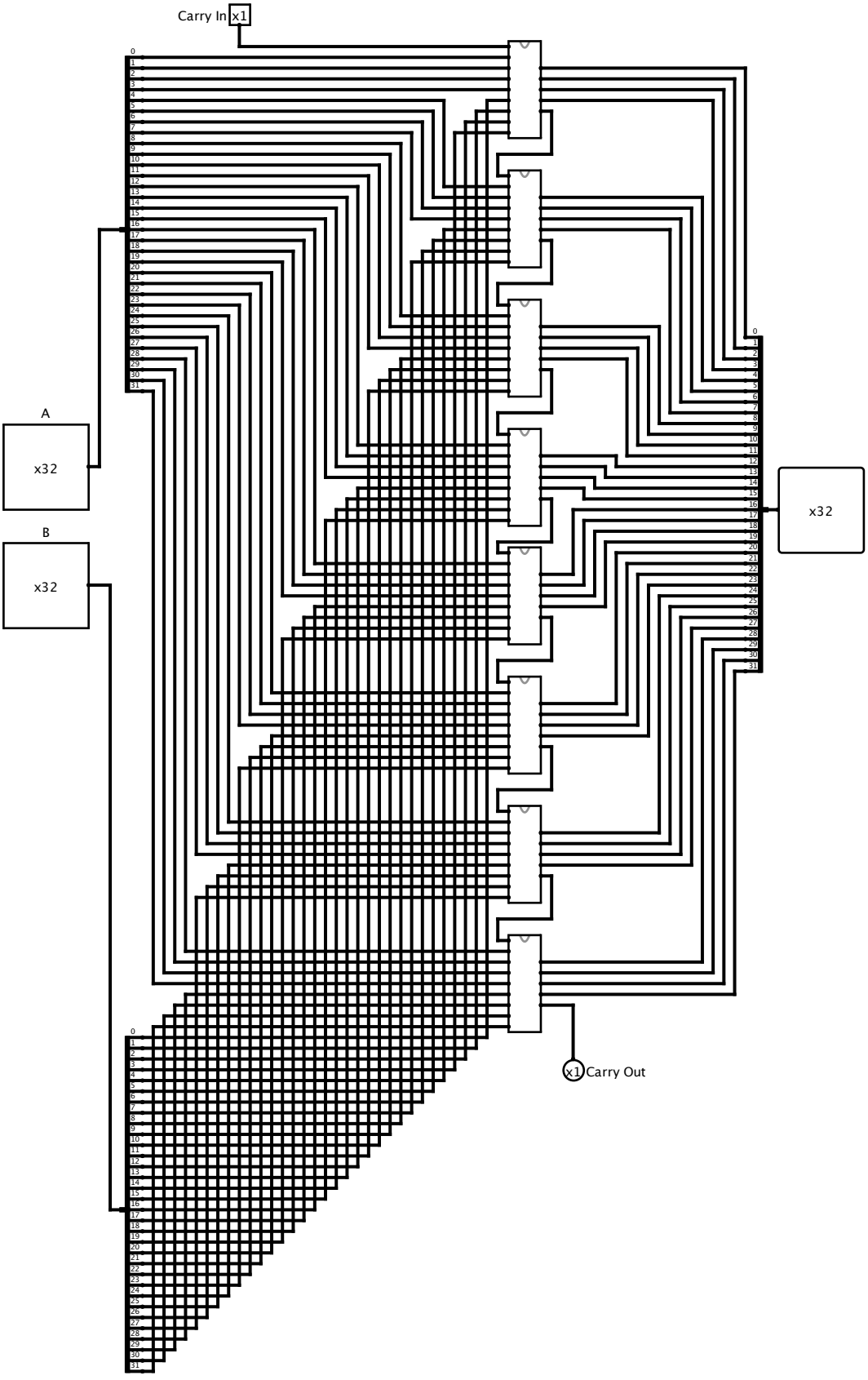
32b Subtract (14 of 17)



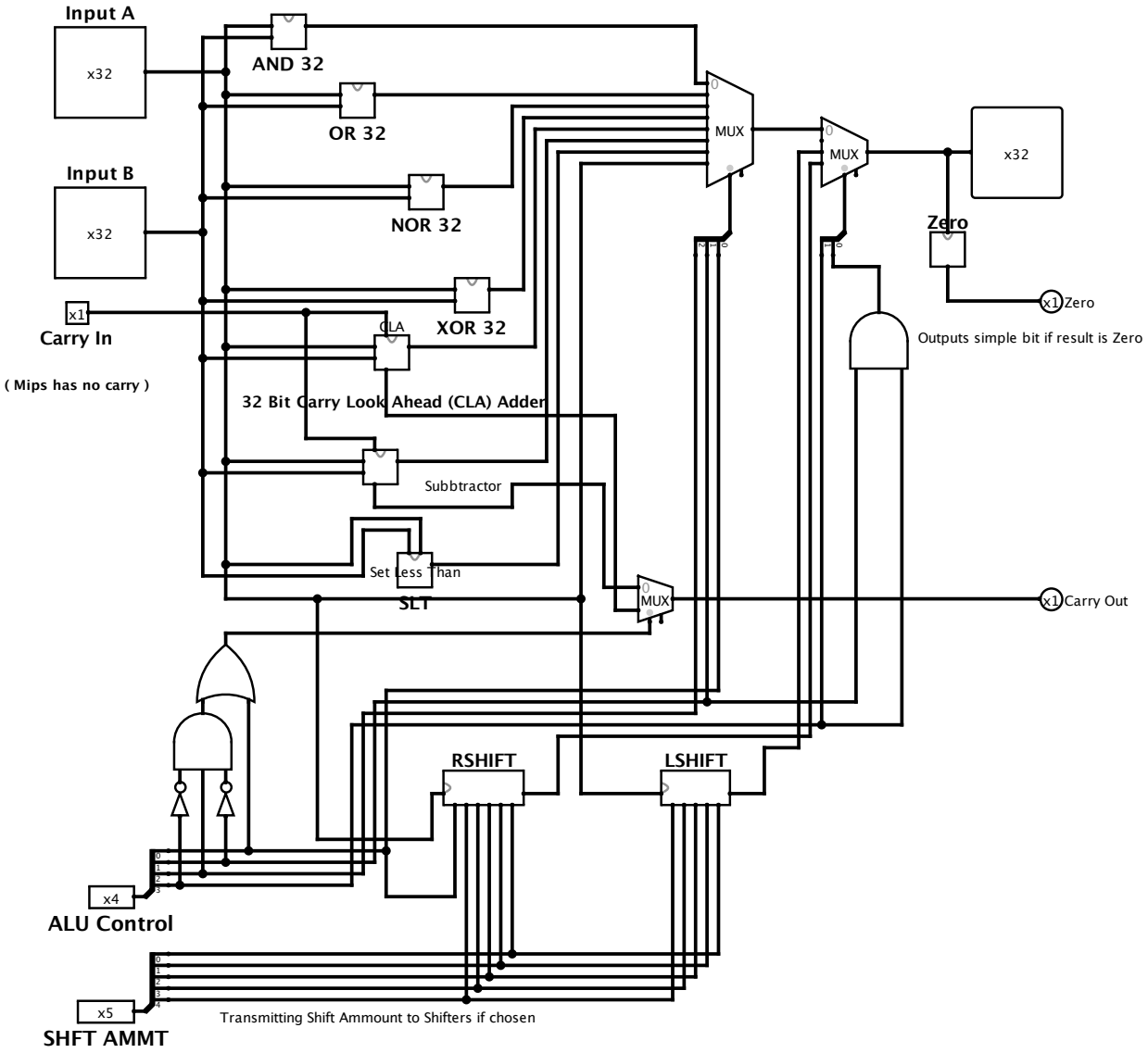
4 bit CLA (15 of 17)



CLA (Carry lookahead adder) (16 of 17)



main (17 of 17)


Input A
x32
AND 32
OR 32
Input B
x32
NOR 32
XOR 32
x1
Carry In
(Mips has no carry)
CLA
32 Bit Carry Look Ahead (CLA) Adder
Subbtractor
Set Less Than
SLT
MUX
MUX
x32
Zero
x1 Zero
Outputs simple bit if result is Zero
x1 Carry Out
RSHIFT
LSHIFT
x4
ALU Control
x5
SHFT AMMT
Transmitting Shift Ammount to Shifters if chosen