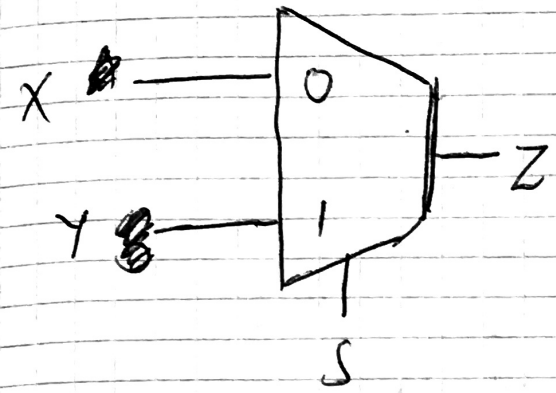
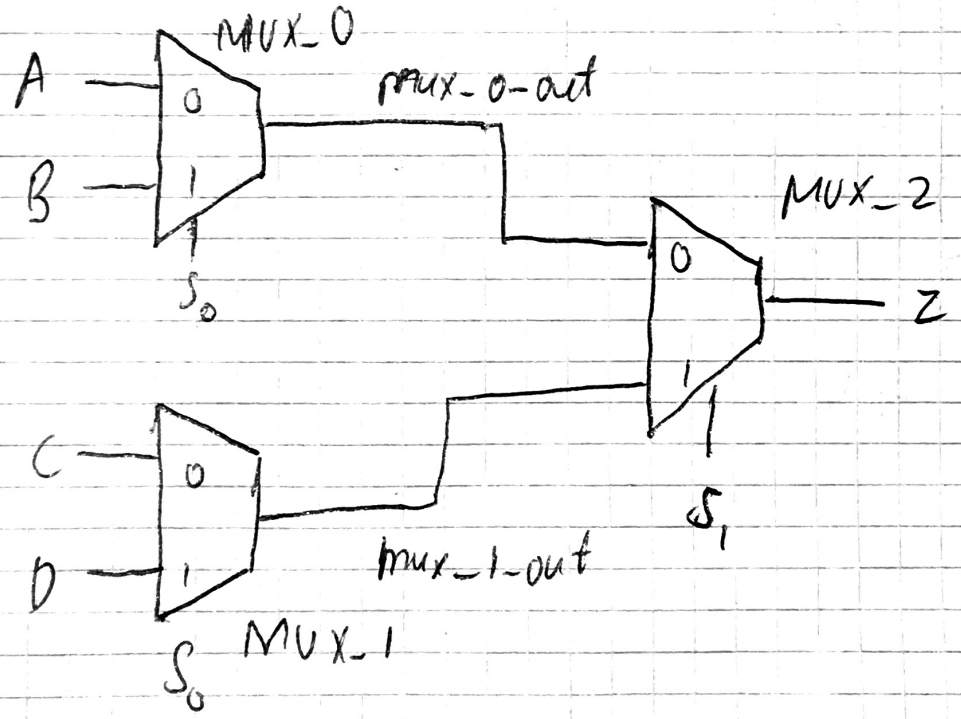
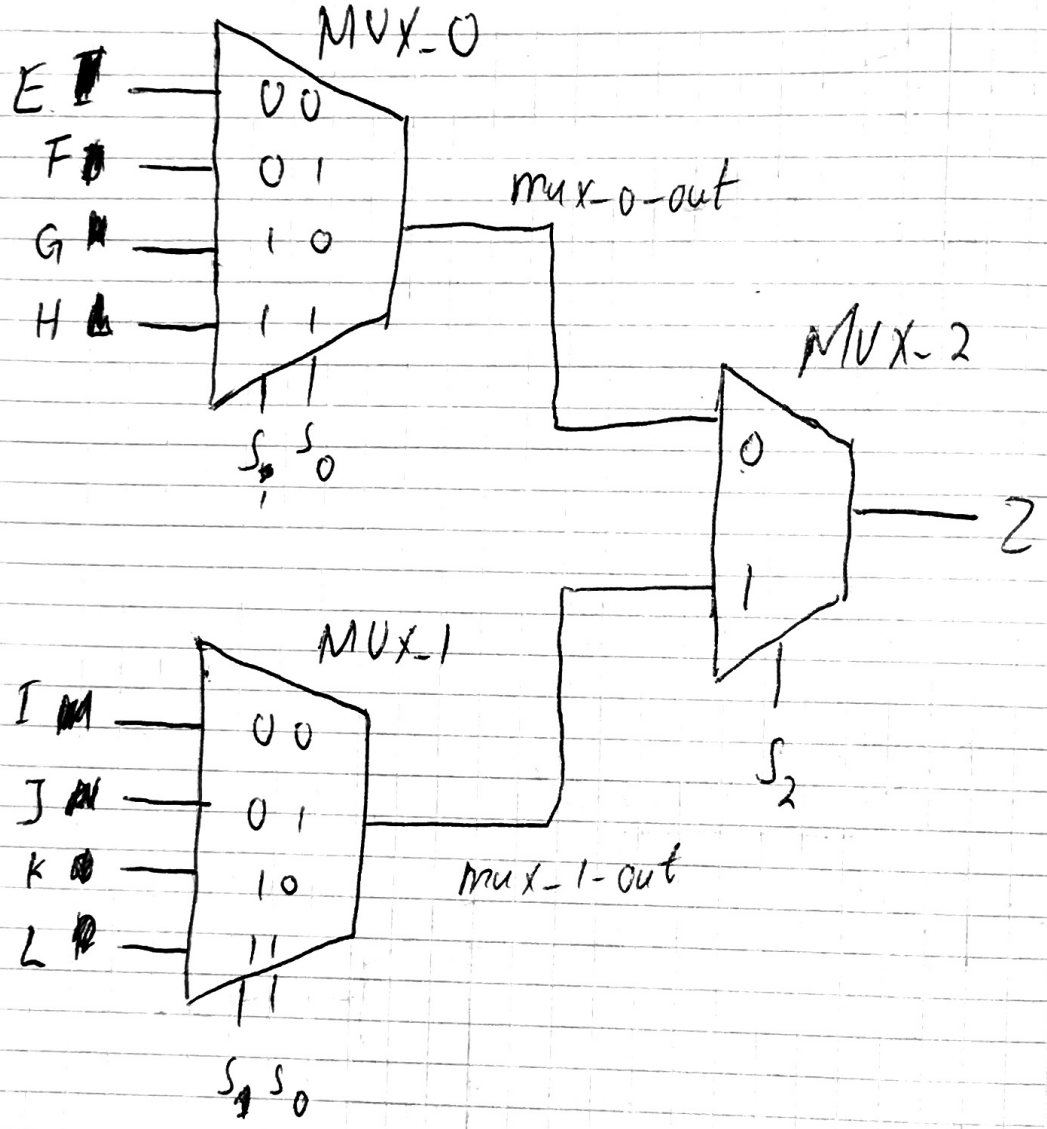
2 to 1 multiplexer



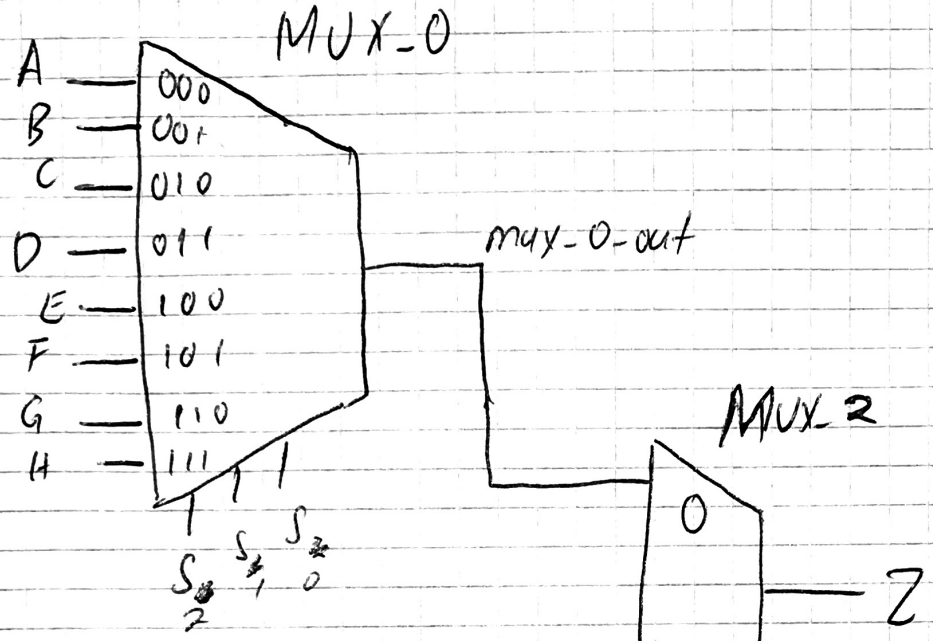
4 to 1 multiplexer

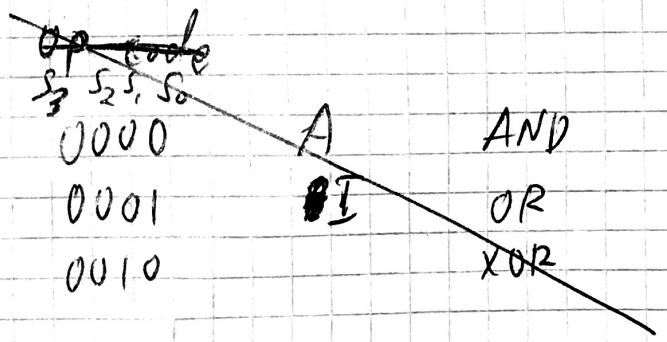


8 to 1 multiplexer



16 to 1 multiplexer

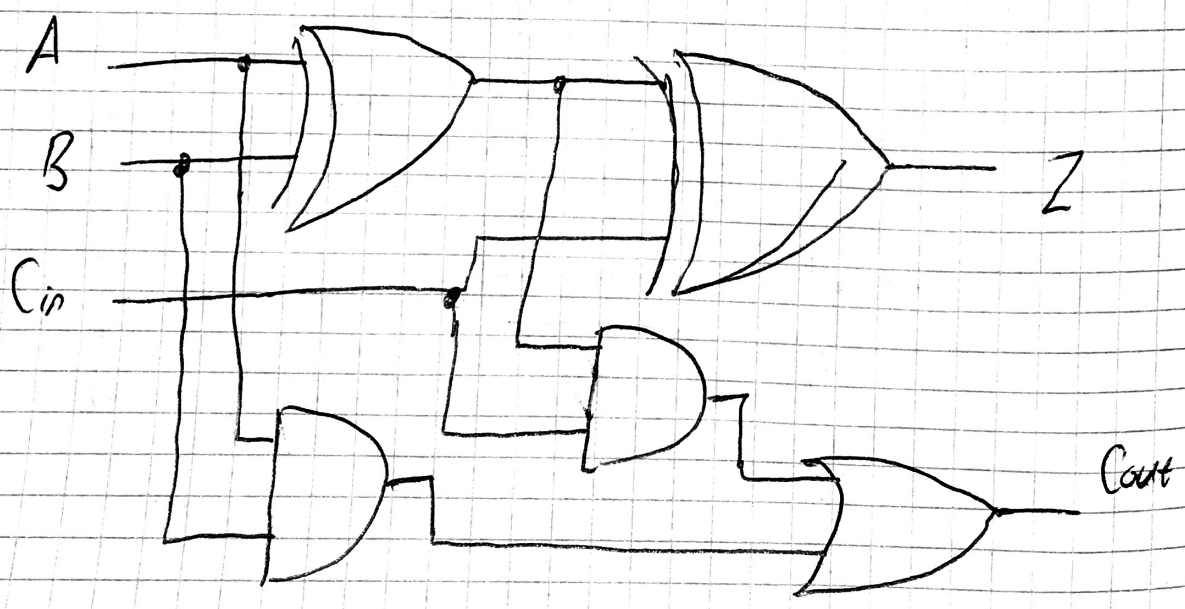




Full Adder

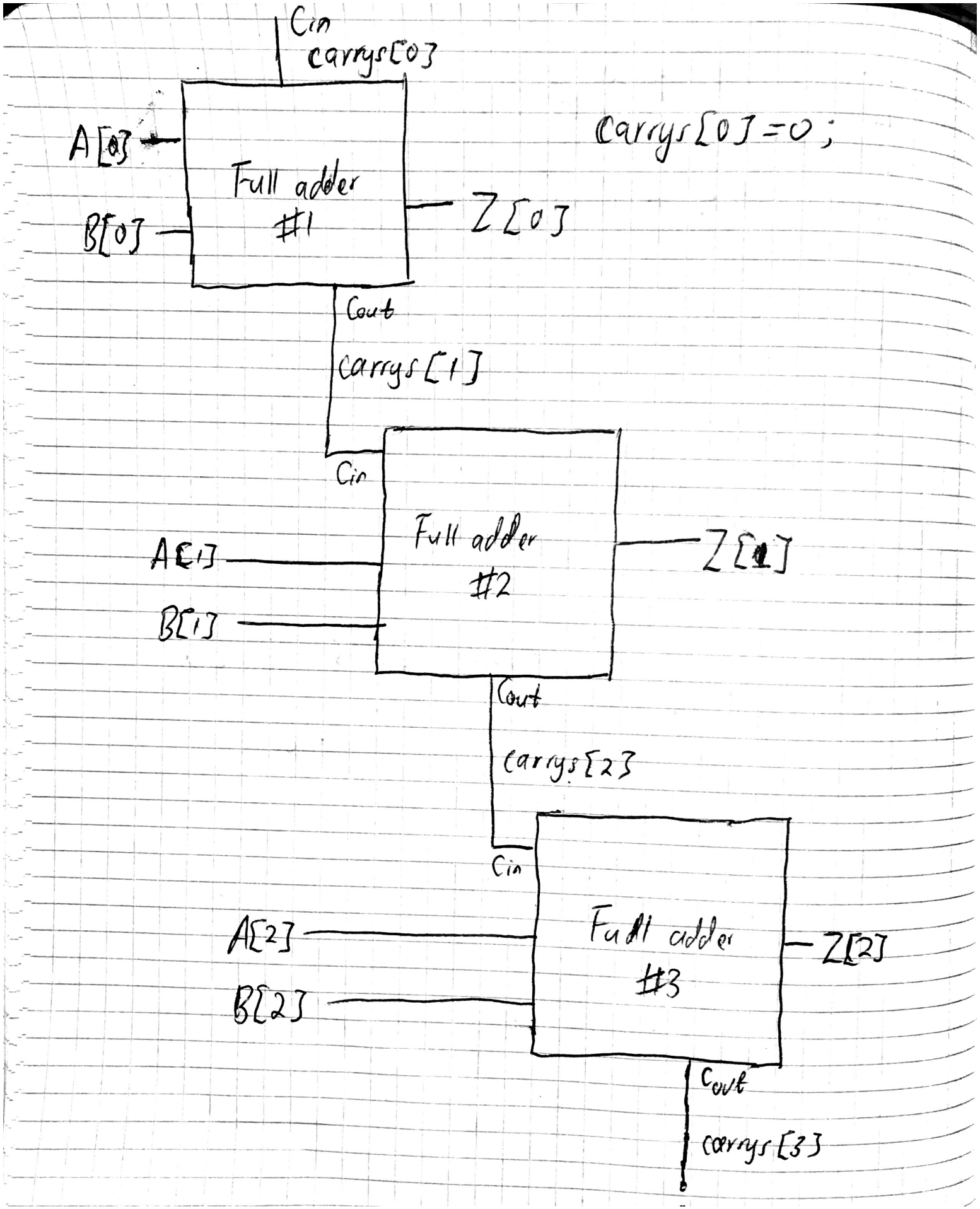
Truth Table

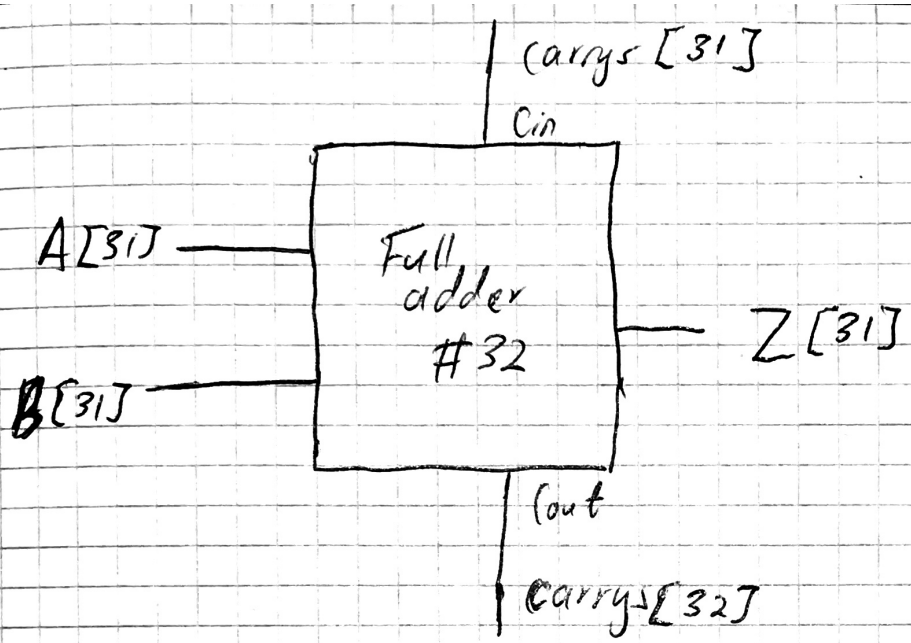
C_{in}	A	B	Z	C_{out}
0	0	0	0	0
0	1	0	1	0
0	0	1	1	0
0	1	1	0	1
1	0	0	1	0
1	1	0	0	1
1	0	1	0	1
1	1	1	1	1



$$Z = (A \oplus B) \oplus C_{in}$$

$$C_{out} = AB + C_{in}(A \oplus B)$$


Cin
carrys[0]
carrys[0]=0;
A[0]
B[0]
Full adder
#1
Z[0]
Cout
carrys[1]
Cin
A[1]
B[1]
Full adder
#2
Z[1]
Cout
carrys[2]
Cin
A[2]
B[2]
Full adder
#3
Z[2]
Cout
carrys[3]



overflow = carrys[32]

SLT

X MSB	Y MSB	X−Y MSB C	z out
0	0	0	0
0	0	1	1
0	1	x	0
0	1	x	0
1	0	x	1
1	0	x	1
1	1	0	0
1	1	1	1

x MSB and y MSB indicate whether they are negative numbers ⇒ the only cases we need to care about are both positive or both negative ⇒ look at the MSB of the calculation X−Y.

Using SOP on the MSBs

$$z = \bar{x}\bar{y}c + x\bar{y} + xyz$$

Equal

Want 1s for equal bits and 0s for unlike bits

X	X	C
0	0	1
0	1	0
1	0	0
1	1	1

← XNOR truth table

$\Rightarrow$ X XNOR Y = 32'b1

Collapse to 1 if all bits are 1 $\Rightarrow$ AND the bits

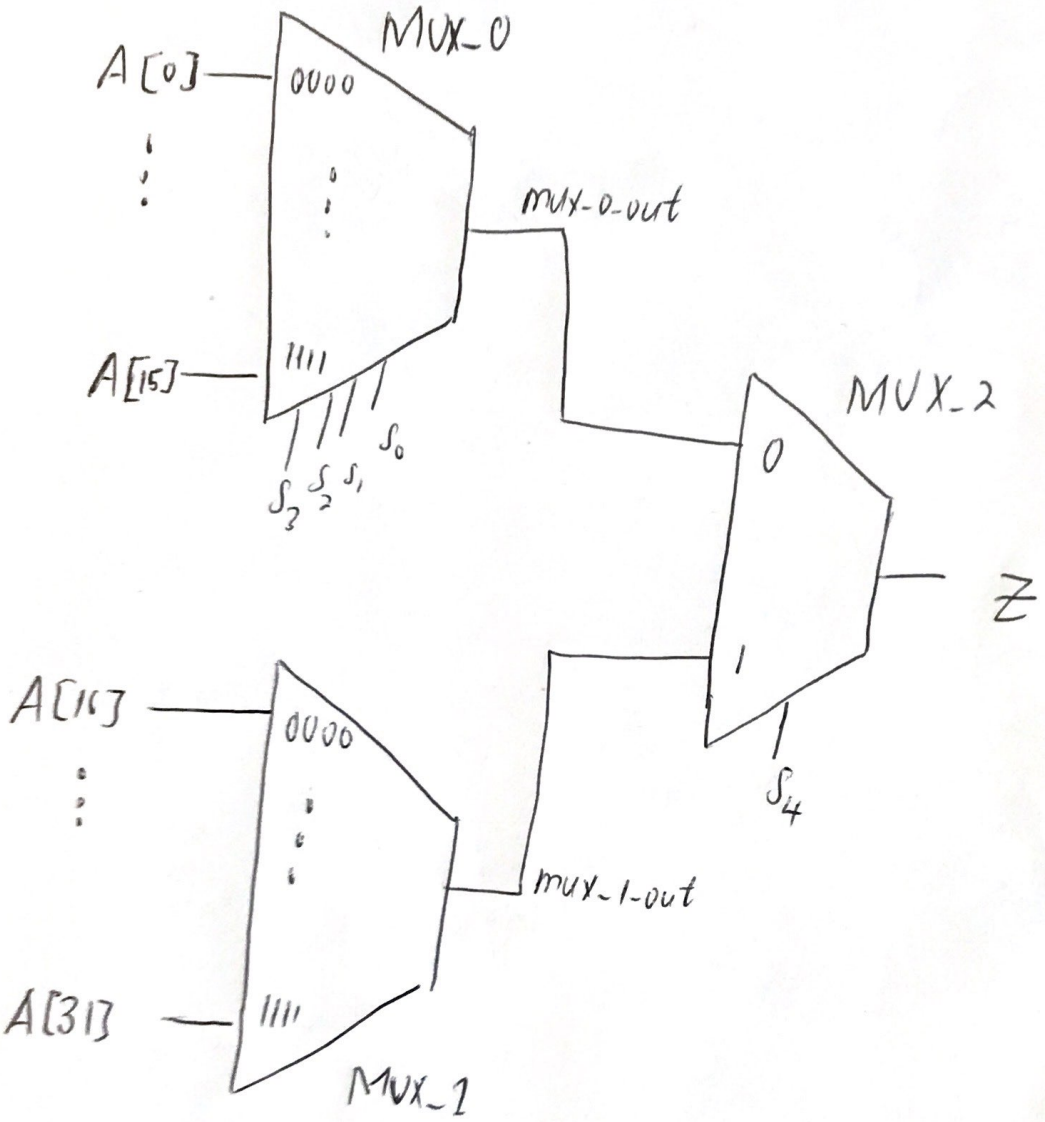
$\therefore$ **AND (X XNOR Y)**

Zero

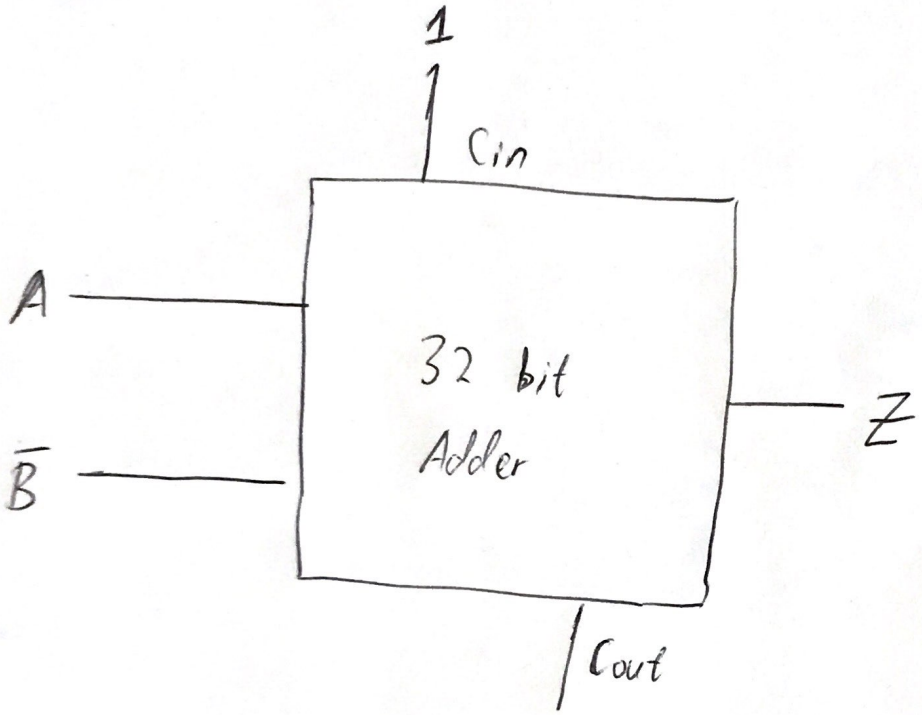
If all output bits are zero, set. (i.e., do not set if only 1 bit is 1)

NOR (output)

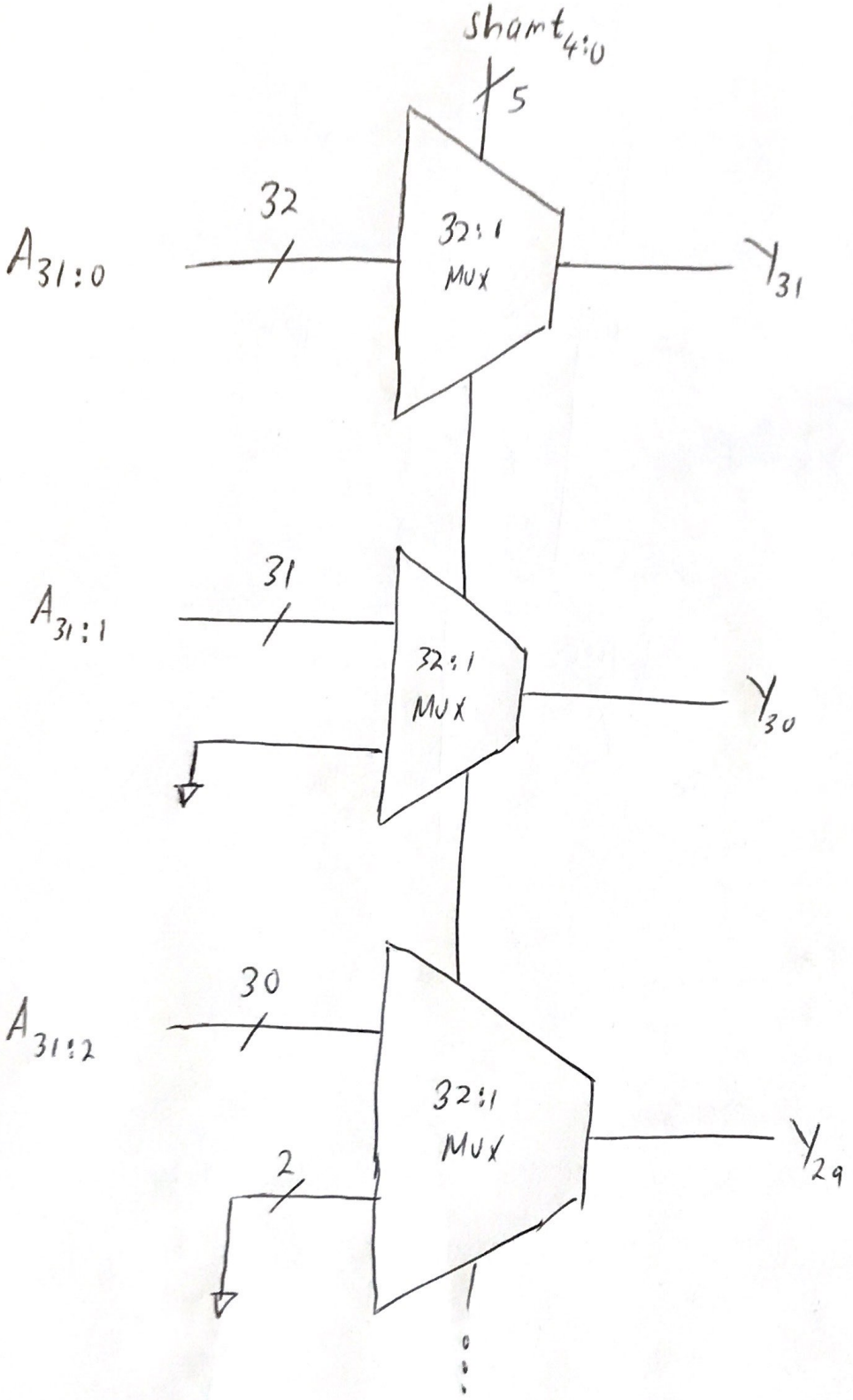
32 to 1 multiplexer

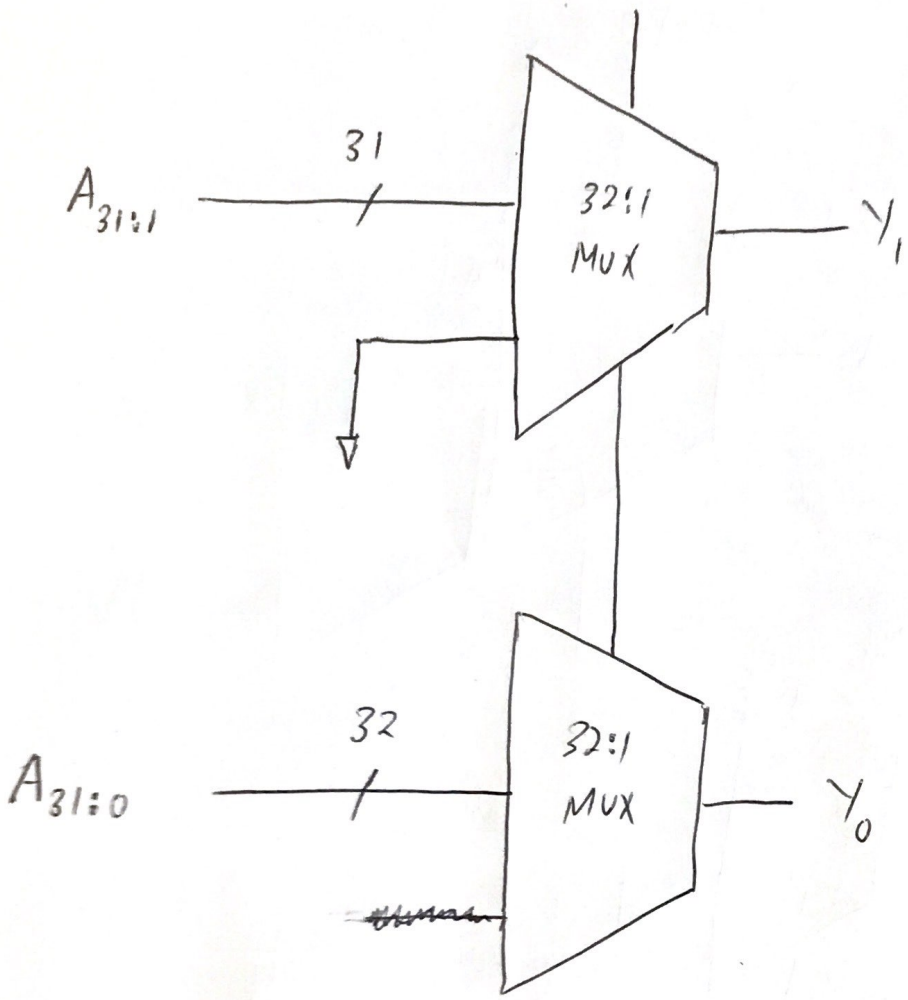


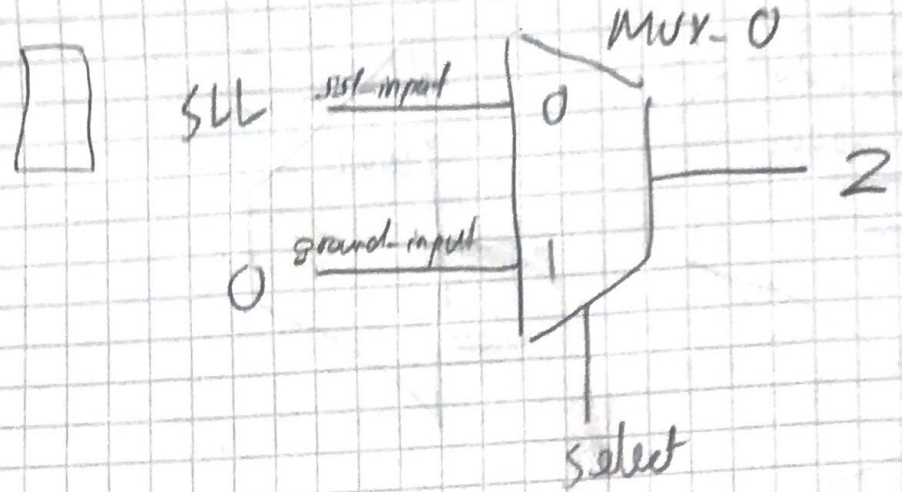
Subtractor. $Z = A - B$



Shift left logical (SLL)

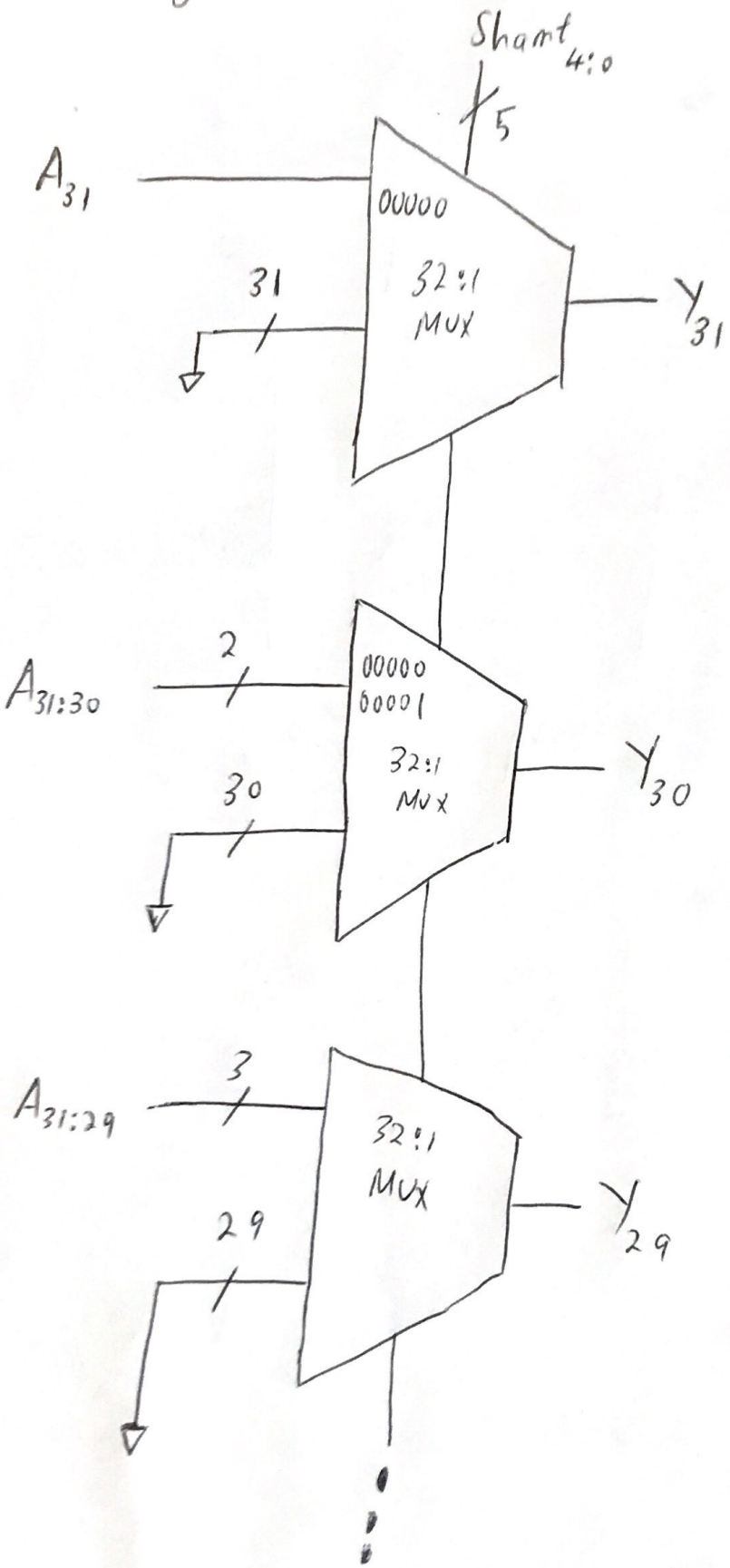


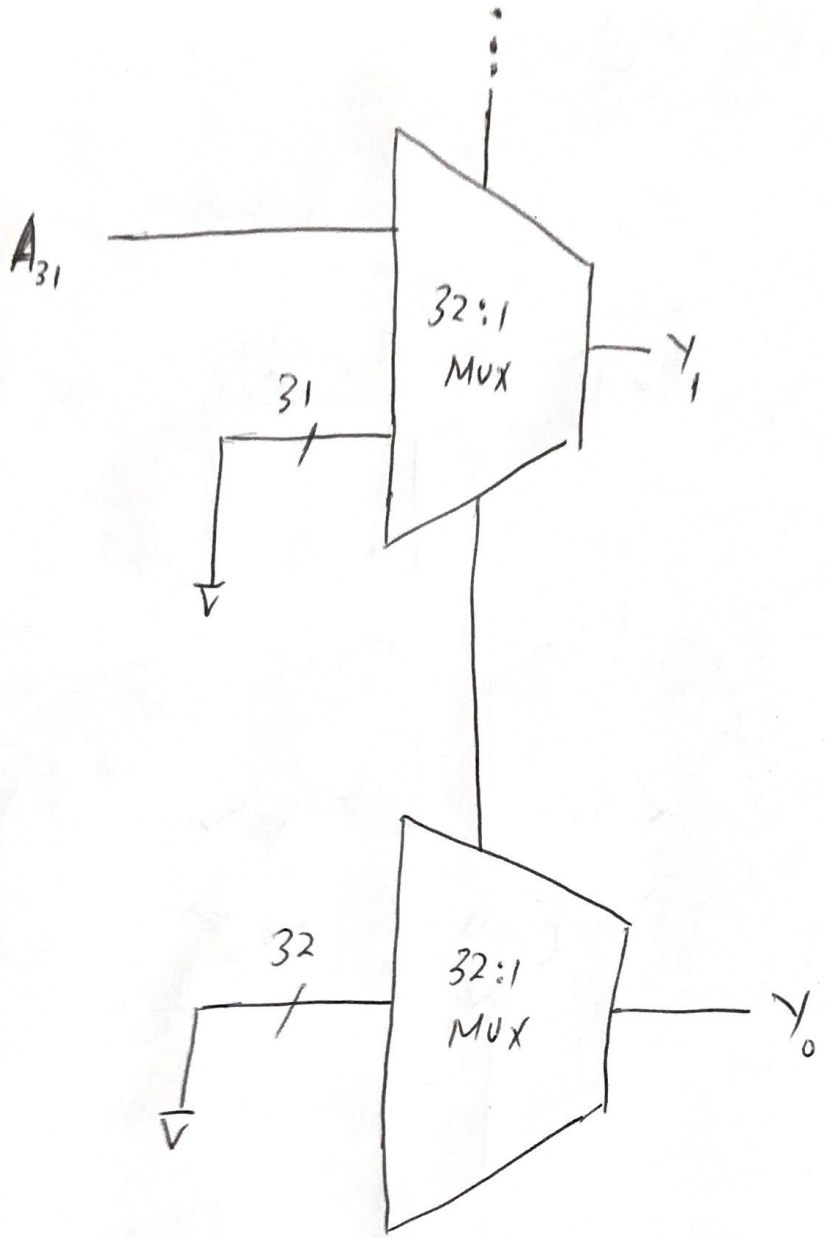

31
$A_{31:1}$
32:1
MUX
Y_1
32
$A_{31:0}$
32:1
MUX
Y_0



select = OR(B(N-1:4))

Shift right logical (SRL)




A_{31}
32:1
MUX
Y_1
31
V
32
32:1
MUX
Y_0
V