Design of the MIPS ALU

Based on slides from Jeremy R. Johnson and William M. Mongan

Building Blocks

1. AND gate (c = a . b)

a	b	c = a . b
0	0	0
0	1	0
1	0	0
1	1	1

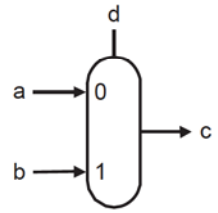
2. OR gate (c = a + b)

a	b	c = a + b
0	0	0
0	1	1
1	0	1
1	1	1

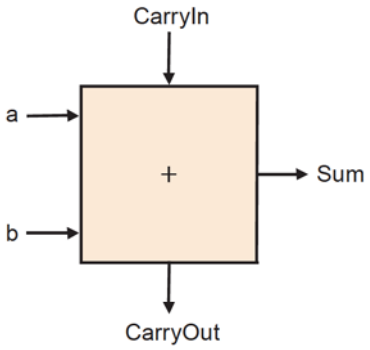
3. Inverter (c = $\bar{a}$)

a	c = $\bar{a}$
0	1
1	0

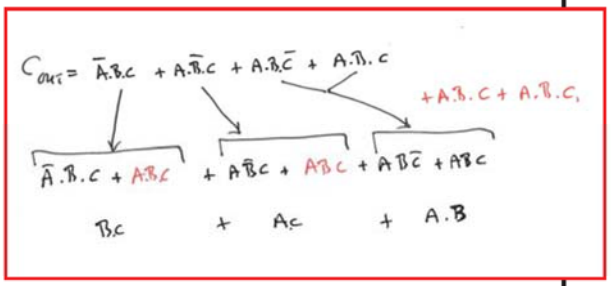
4. Multiplexor
(if d = = 0, c = a;
else c = b)



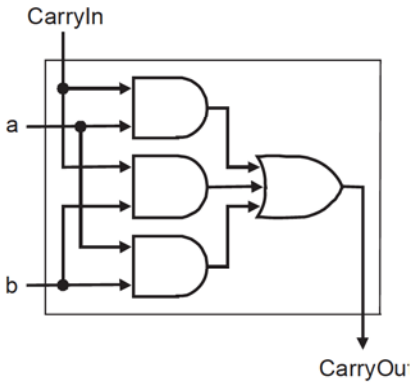
d	c
0	a
1	b



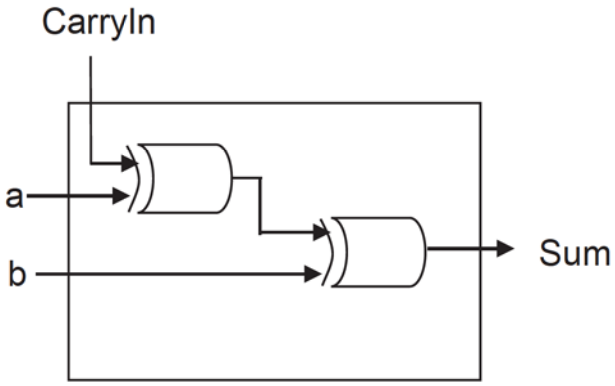
Full Adder



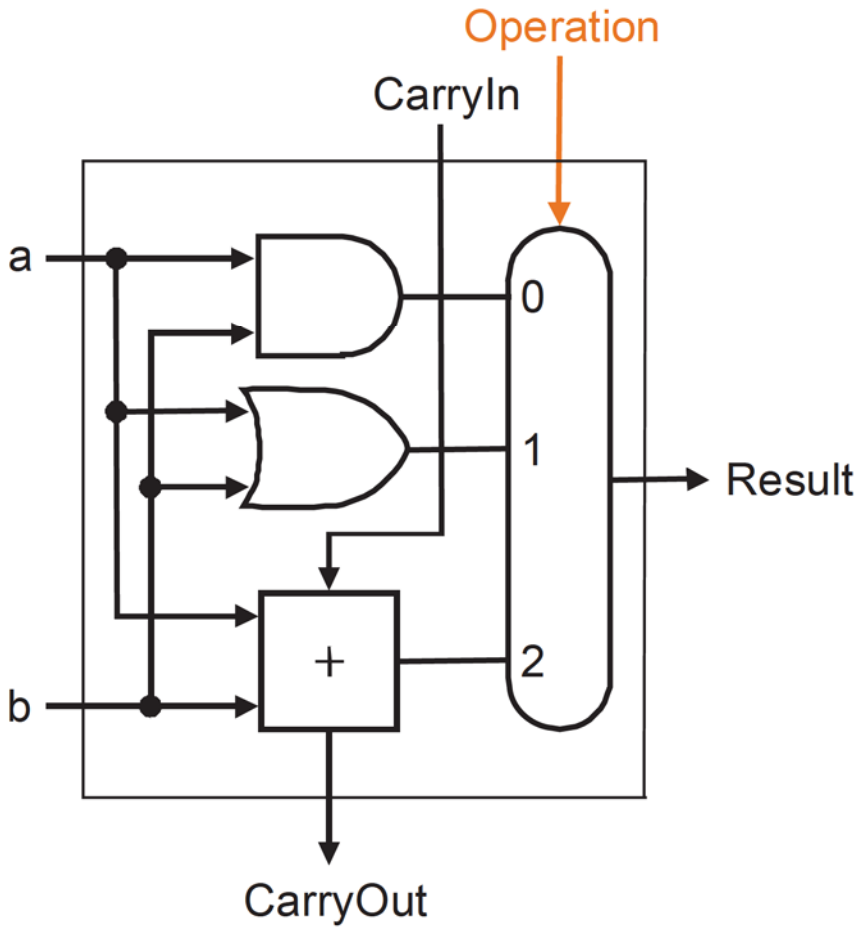
- **Sum = parity(a, b, CarryIn)**
  - **a xor b xor c + a•b•c ≡ a xor b xor c**
- **CarryOut = majority(a, b, CarryIn)**
  - **b•CarryIn + a•CarryIn + a•b + a•b•CarryIn ≡**
  - **b•CarryIn + a•CarryIn + a•b**



a	b	CarryIn	Sum	CarryOut
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

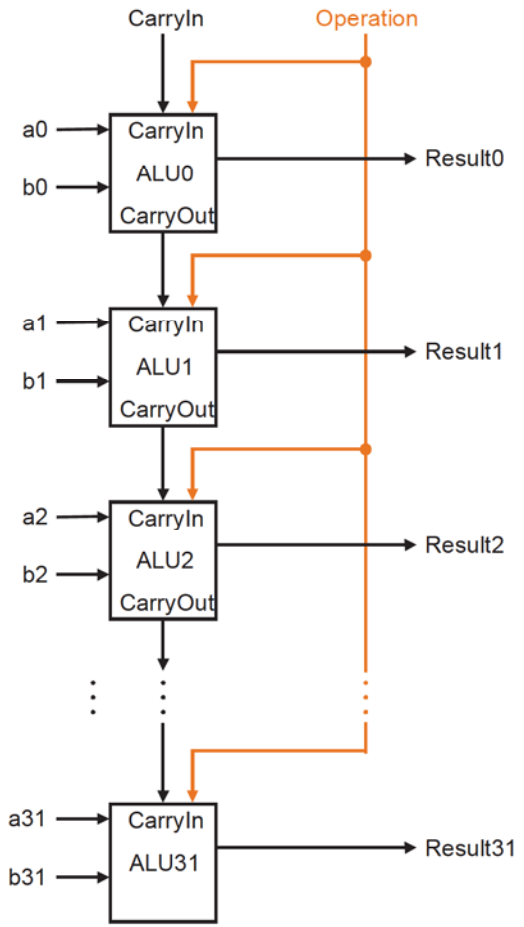


One-Bit ALU



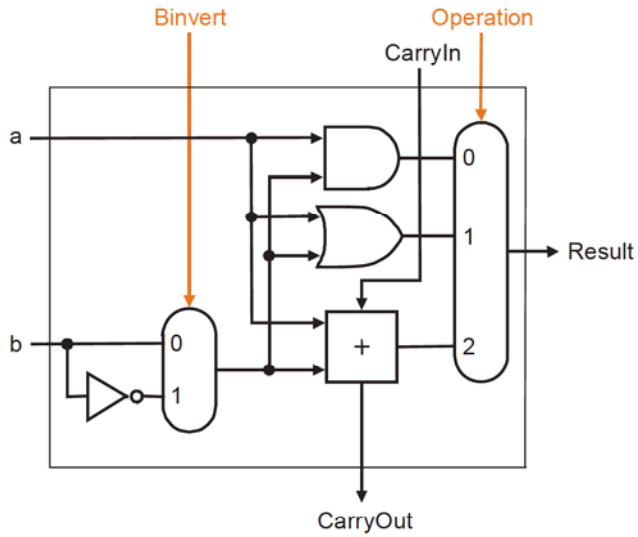
Building a 32-Bit ALU

- **Chain 32 1-Bit ALUs**



Supporting Subtraction

- **Subtraction is equivalent to adding the inverse**
  - **In two's complement a + $\bar{b}$ + 1**



Overflow and SLT

- **Modify last 1-Bit ALU**
  - **SLT set if (a < b) ⇔ a - b < 0**
  - **Check sign bit after subtraction**
  - **Check overflow in last 1-Bit ALU**
  - **Need to take overflow into account for SLT**

$-3 < 7$? IN FOUR BITS

$-3 - 7 < 0$?

-3 1101

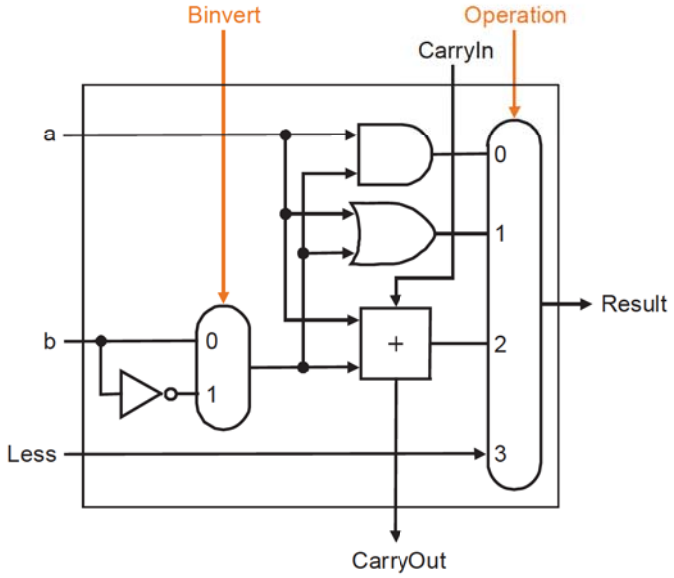
-7 1001

6 0110

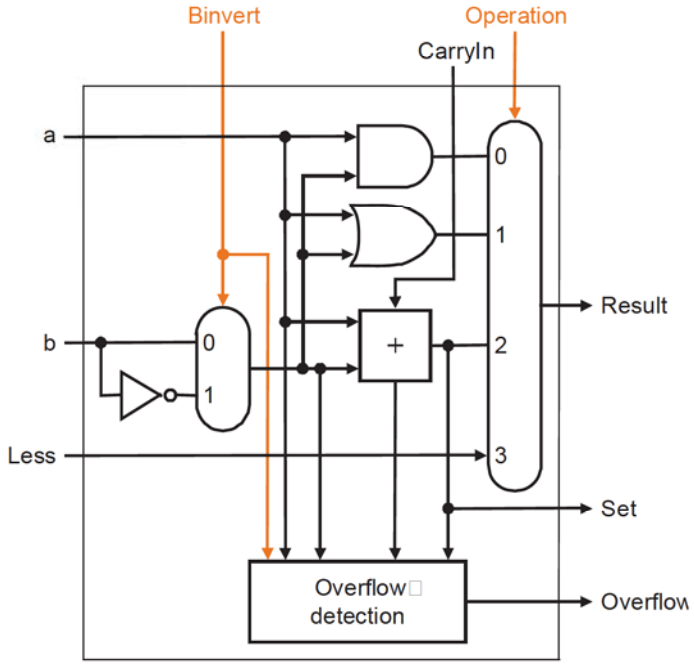
↑ SIGN BIT = 0

SO $-3 - 7 < 0$ IS FALSE

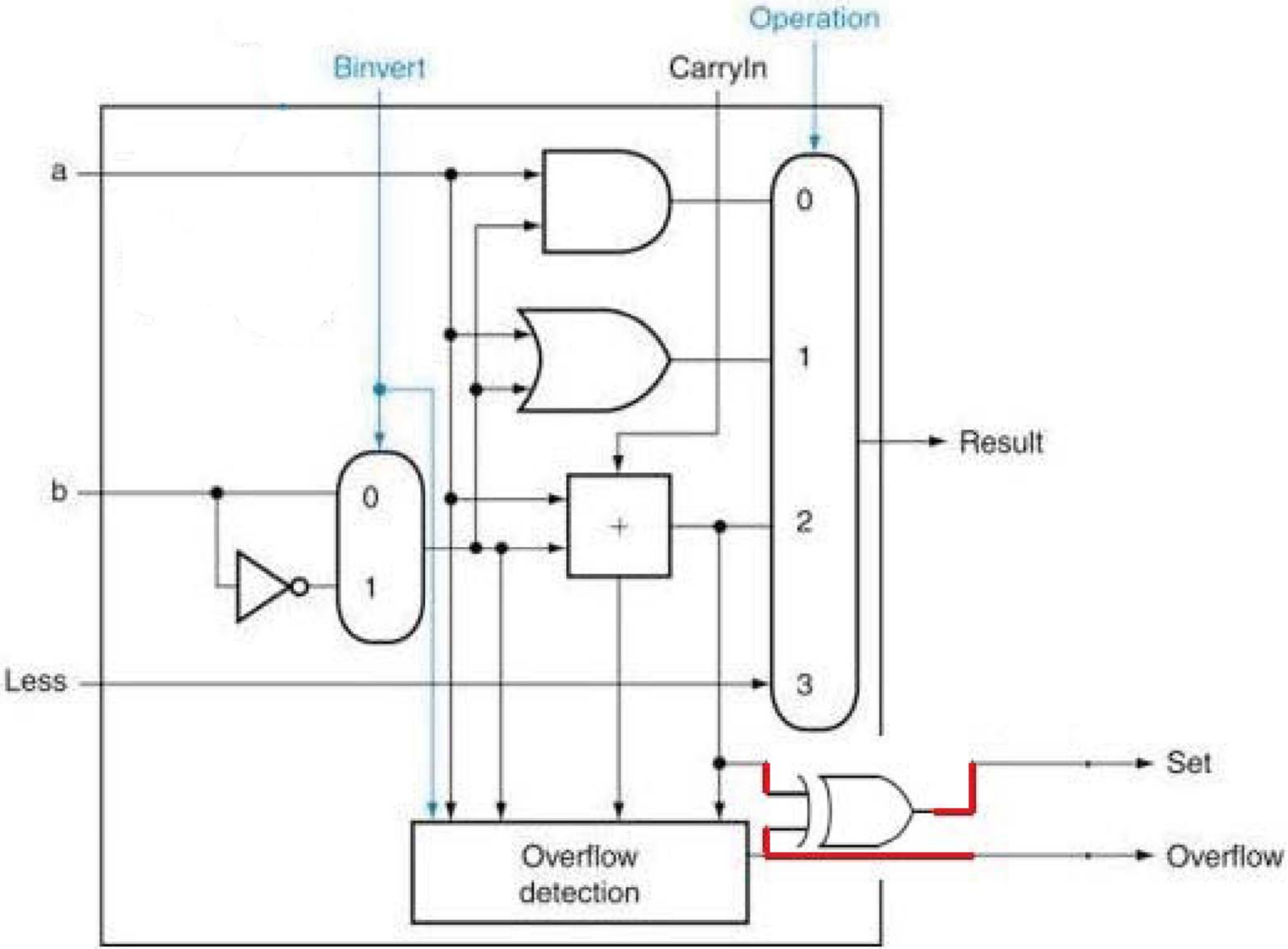
$-3 < 7$ IS FALSE

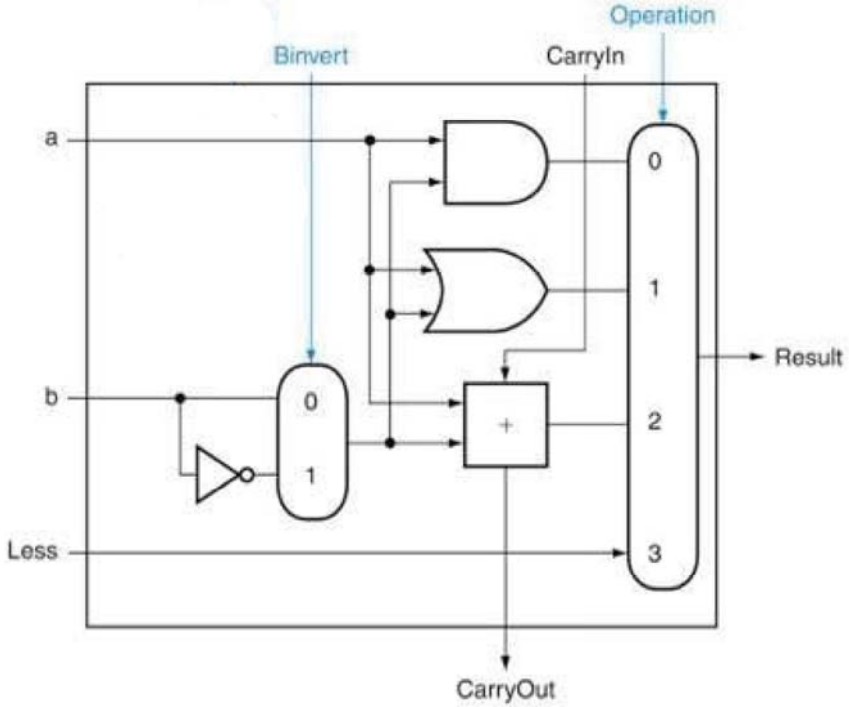


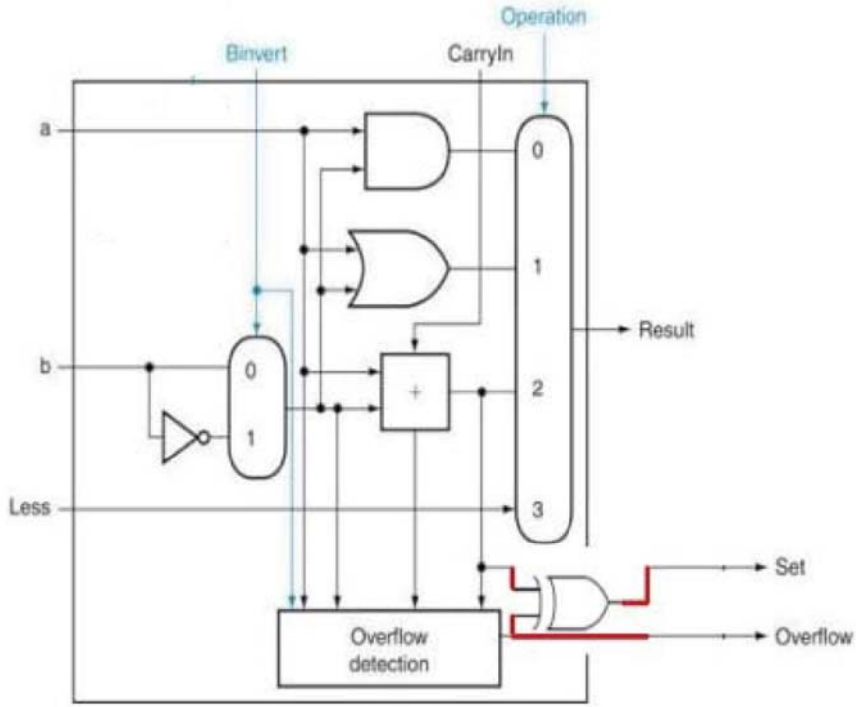
a.

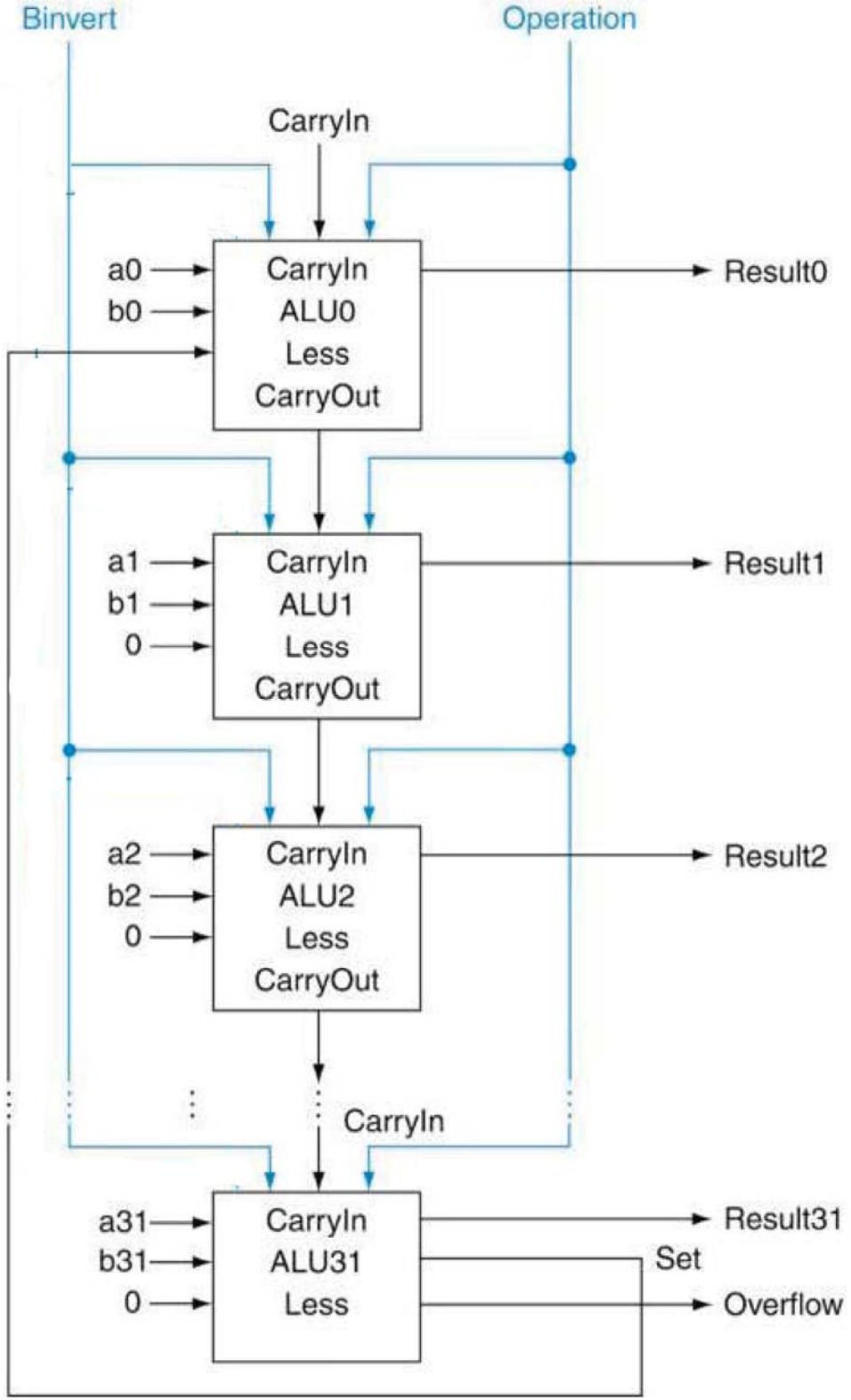


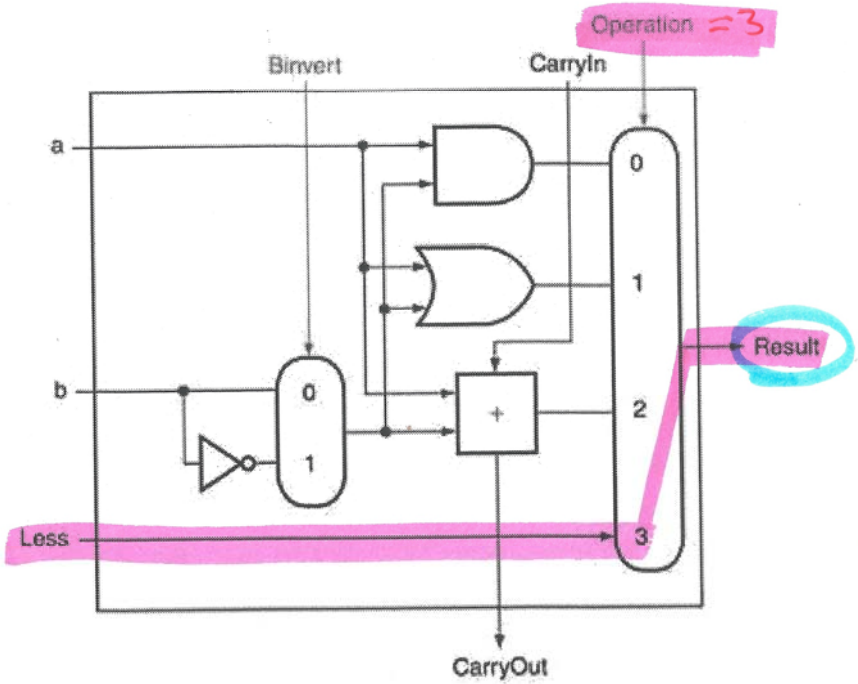
b.

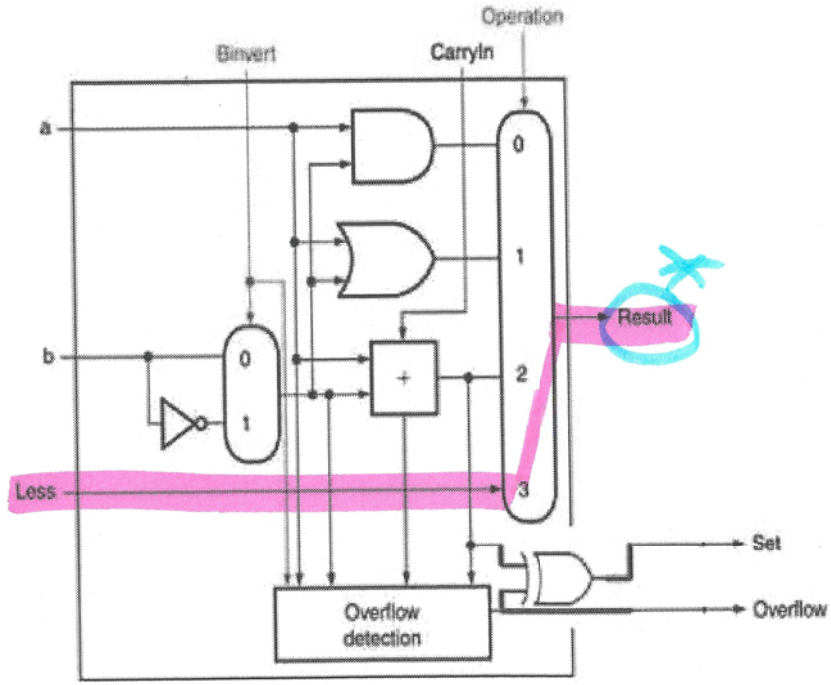

Operation
Binvert
CarryIn
a
b
Less
0
1
2
3
0
1
+
Result
Overflow
detection
Set
Overflow

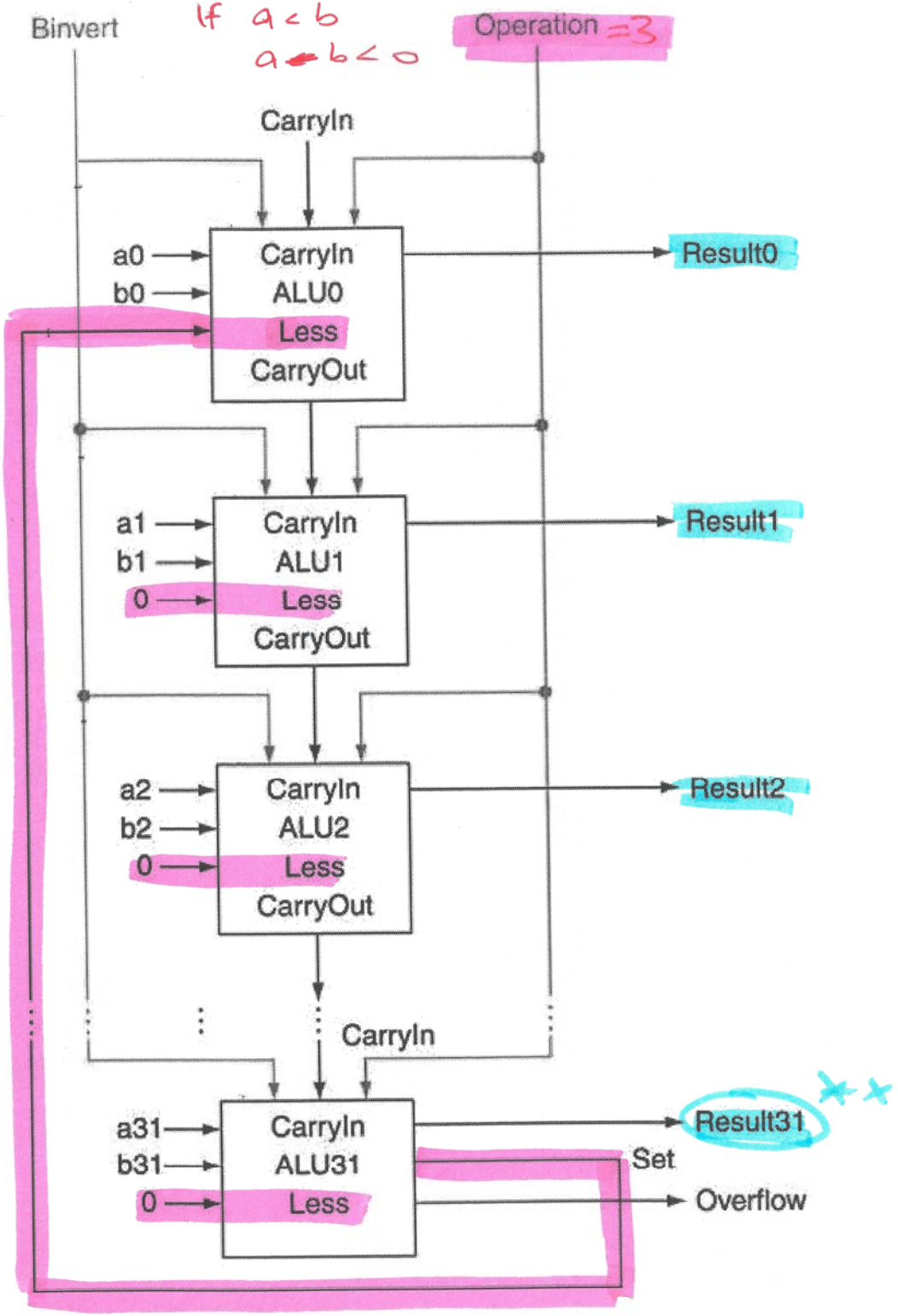

Operation
Binvert
CarryIn
a
b
Less
0
1
2
3
+
Result
CarryOut


Operation
Binvert
CarryIn
a
b
Less
0
1
2
3
+
Result
Set
Overflow
Overflow detection

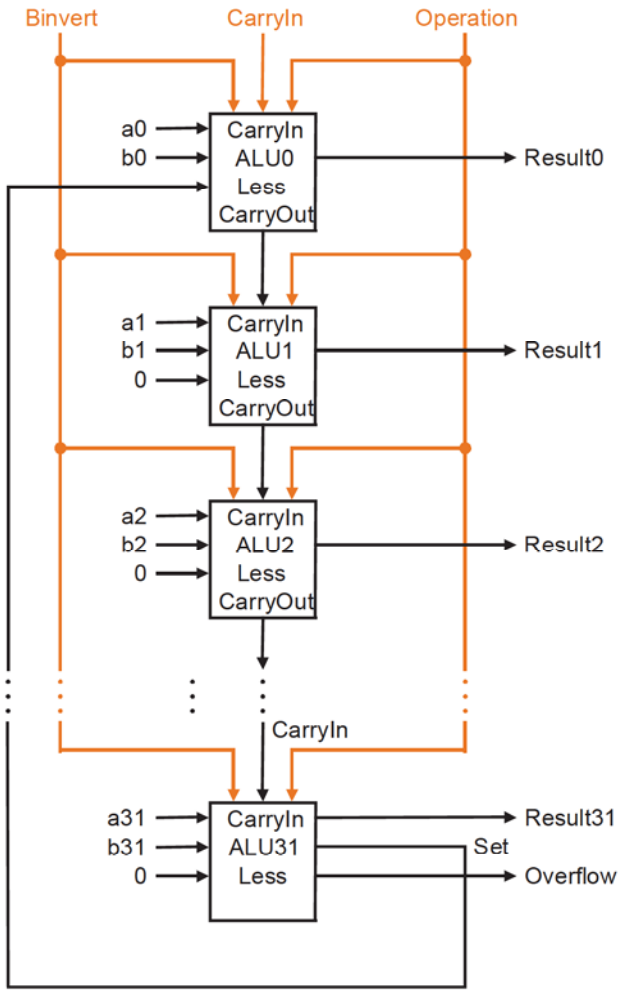

Binvert
Operation
CarryIn
a0
b0
CarryIn
ALU0
Less
CarryOut
Result0
a1
b1
0
CarryIn
ALU1
Less
CarryOut
Result1
a2
b2
0
CarryIn
ALU2
Less
CarryOut
Result2
CarryIn
a31
b31
0
CarryIn
ALU31
Less
Result31
Set
Overflow


Operation =3
Binvert
CarryIn
a
b
0
1
2
3
Result
Less
CarryOut


Operation
Binvert
CarryIn
a
b
0
1
2
3
Result
Less
Set
Overflow
Overflow detection


Binvert
If a<b
a-b<0
Operation =3
CarryIn
a0
b0
CarryIn
ALU0
Less
CarryOut
Result0
a1
b1
0
CarryIn
ALU1
Less
CarryOut
Result1
a2
b2
0
CarryIn
ALU2
Less
CarryOut
Result2
CarryIn
a31
b31
0
CarryIn
ALU31
Less
Result31
Set
Overflow

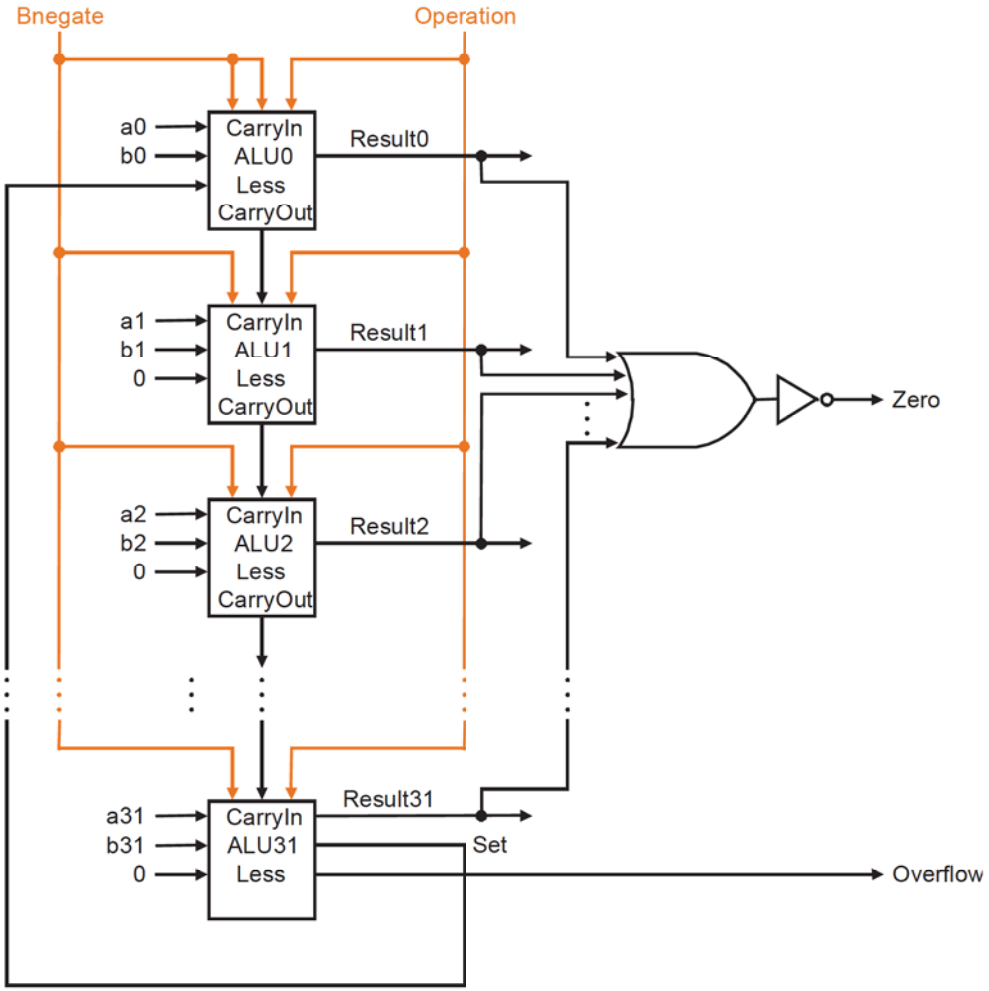
32-Bit ALU with Sub and Slt



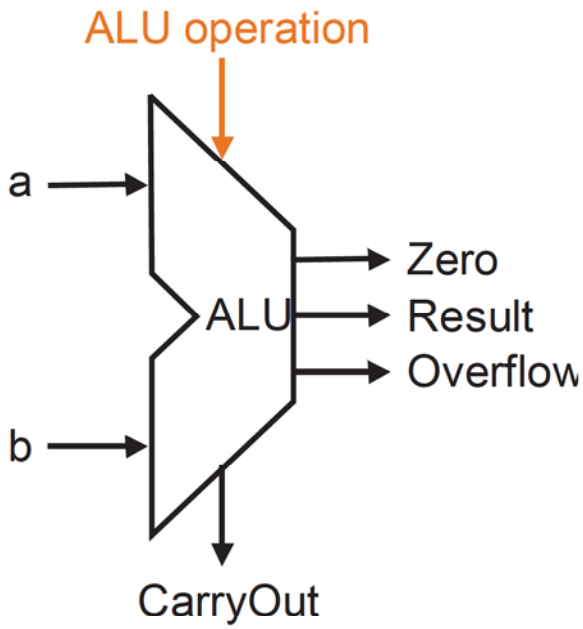
Support Beq

- $a = b \Leftrightarrow a - b = 0$
- $\text{Zero} = \overline{(\text{Result31} + \cdots + \text{Result0})}$

Final 32-Bit ALU



Control	Function
000	and
001	or
010	add
110	sub
111	slt



Exercise 5: 32-bit ALU

Using the diagram on the next page, add necessary connections and show the values of all signals given that the two numbers at the inputs of the ALU are A = 1 and B = 3, and Binvert = 1, and Operation = 3.

Exercise 5
a=1
b=3
Binvert=1
Operation=3
Show all signals