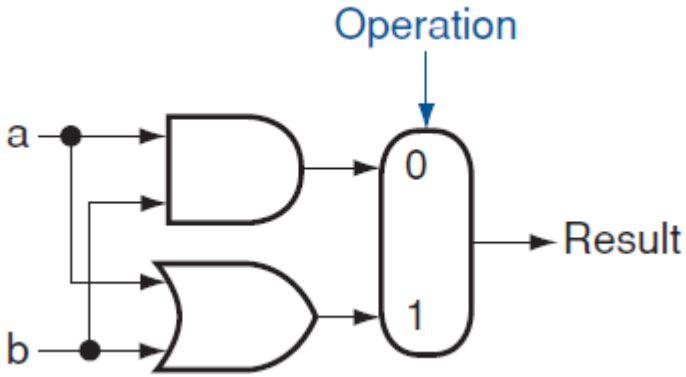
Arithmetic Logic Unit

- The ALU is the brawn of the computer
- Performs integer arithmetic operations
  - Addition and subtraction
  - Multiplication and division
- Performs logical operations
  - AND, OR, XOR
- Acts on commands from control unit

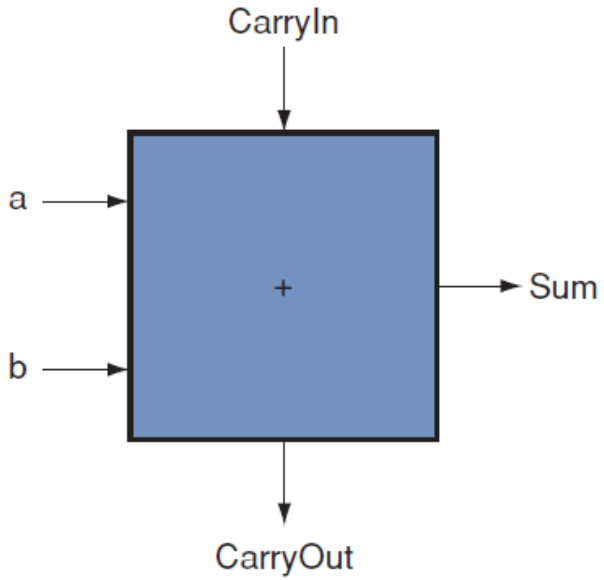
Building an ALU

- Multiple 1-bit ALUs can be used to build a 32-bit ALU
- This is called a bit sliced design
- The AND and OR operations map directly to gates



Building an ALU

- From module 3 we know how to build a full adder

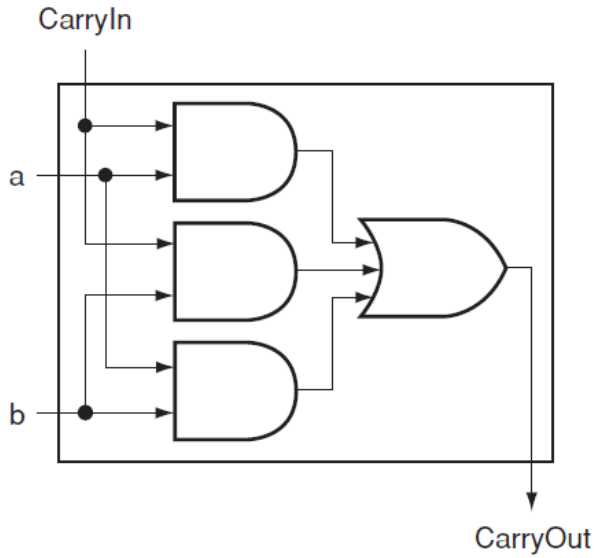


- The single bit inputs (a and b) together with the Carryin are added to produce the Sum and Carryout

Building an ALU

- Truth table and possible circuit to generate CarryOut

Inputs		
a	b	CarryIn
0	1	1
1	0	1
1	1	0
1	1	1

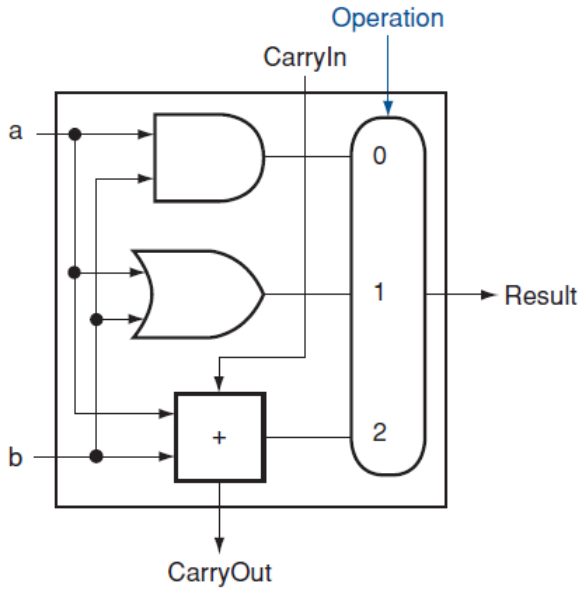


CarryOut = (a • CarryIn) + (a • b) + (b • CarryIn)

= (a • b) + (a + b) • CarryIn

Building an ALU

- 1-Bit ALU containing AND gate, OR gate and Full adder



- Full 32-Bit ALU can be builtup from multiple copies

Building an ALU

Subtraction can be included by noting that:

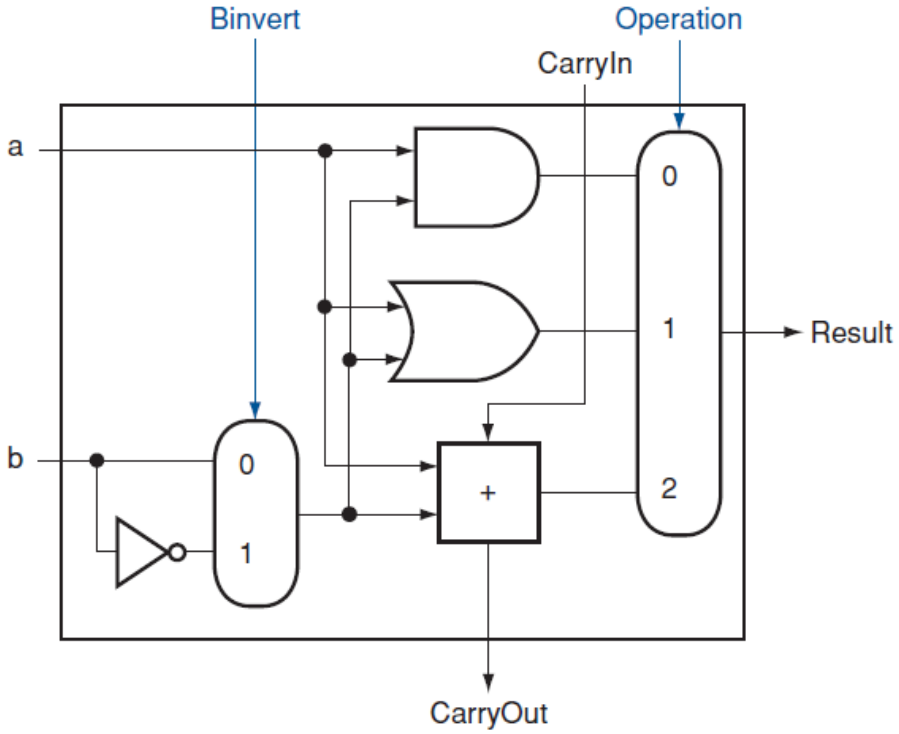
$$a + \bar{b} + 1 = a + (\bar{b} + 1) = a + (-b) = a - b$$

$\bar{b} + 1$ is the negative of b

Subtract b from a by adding the negative of b to a

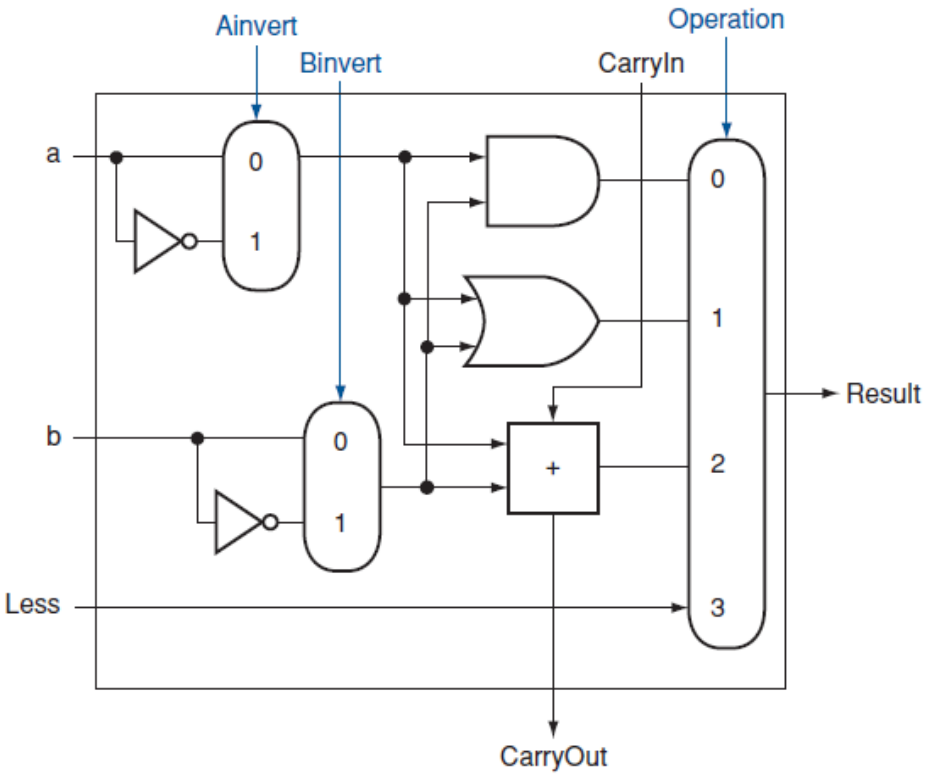
Uses two's complement of b

Building an ALU



1-Bit ALU that subtracts, adds and performs AND and OR

Building an ALU



Ainvert allows the computation of (b – a) and other functions