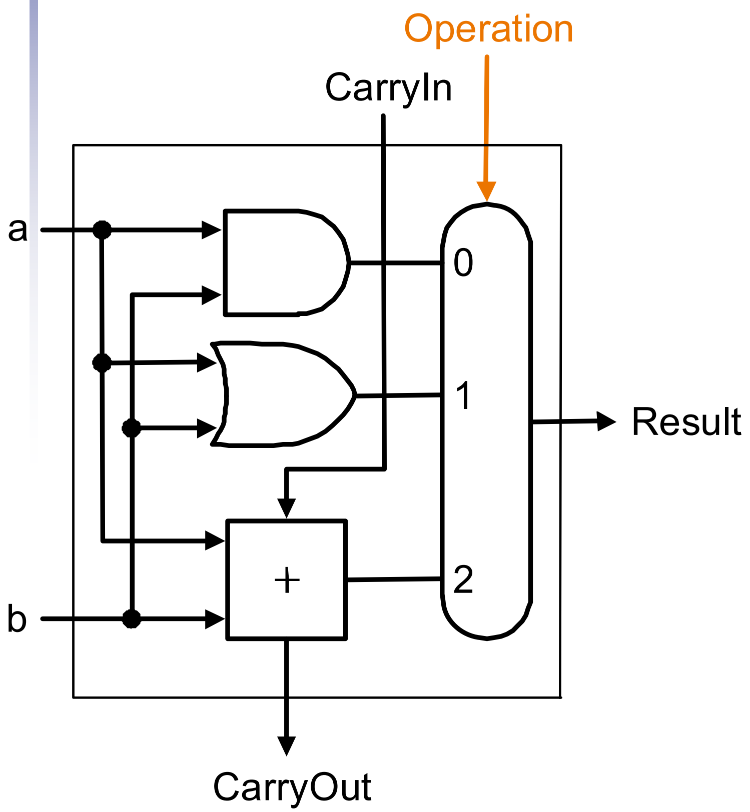
Chapter 3

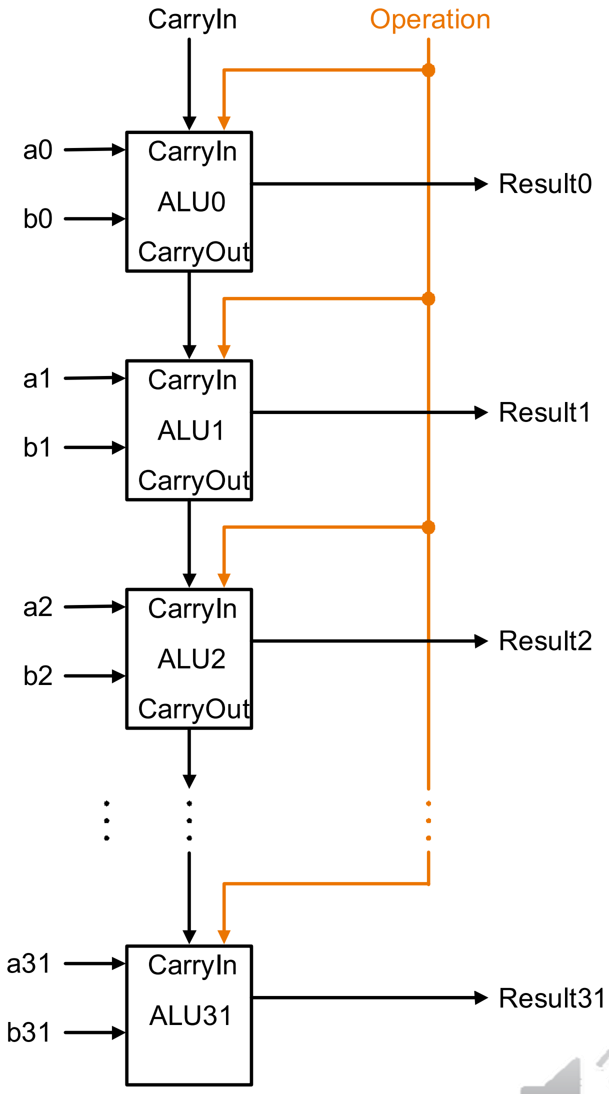
Arithmetic for Computers

32-bit ALU

- Ripple Carry ALU



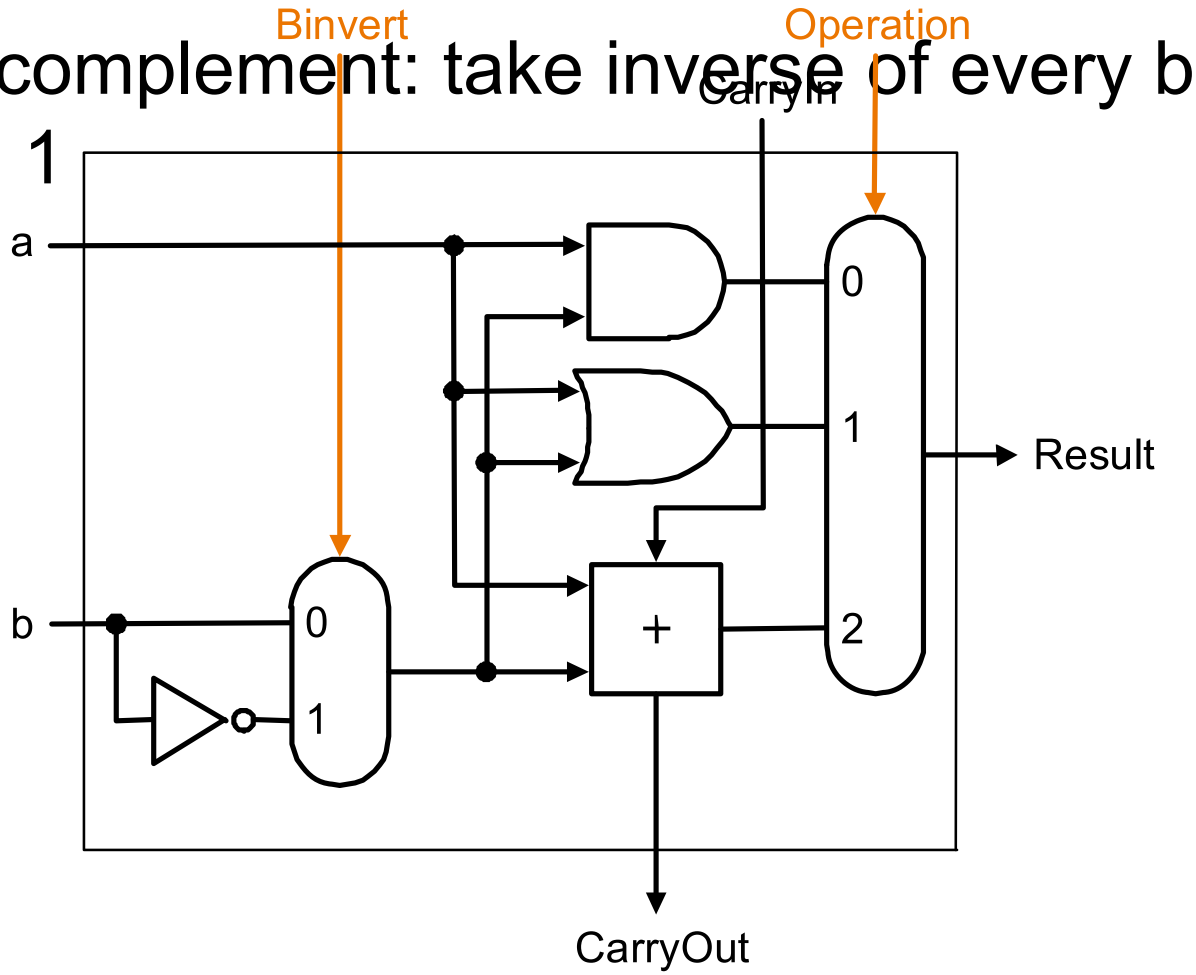
1-bit ALU



32-bit ALU

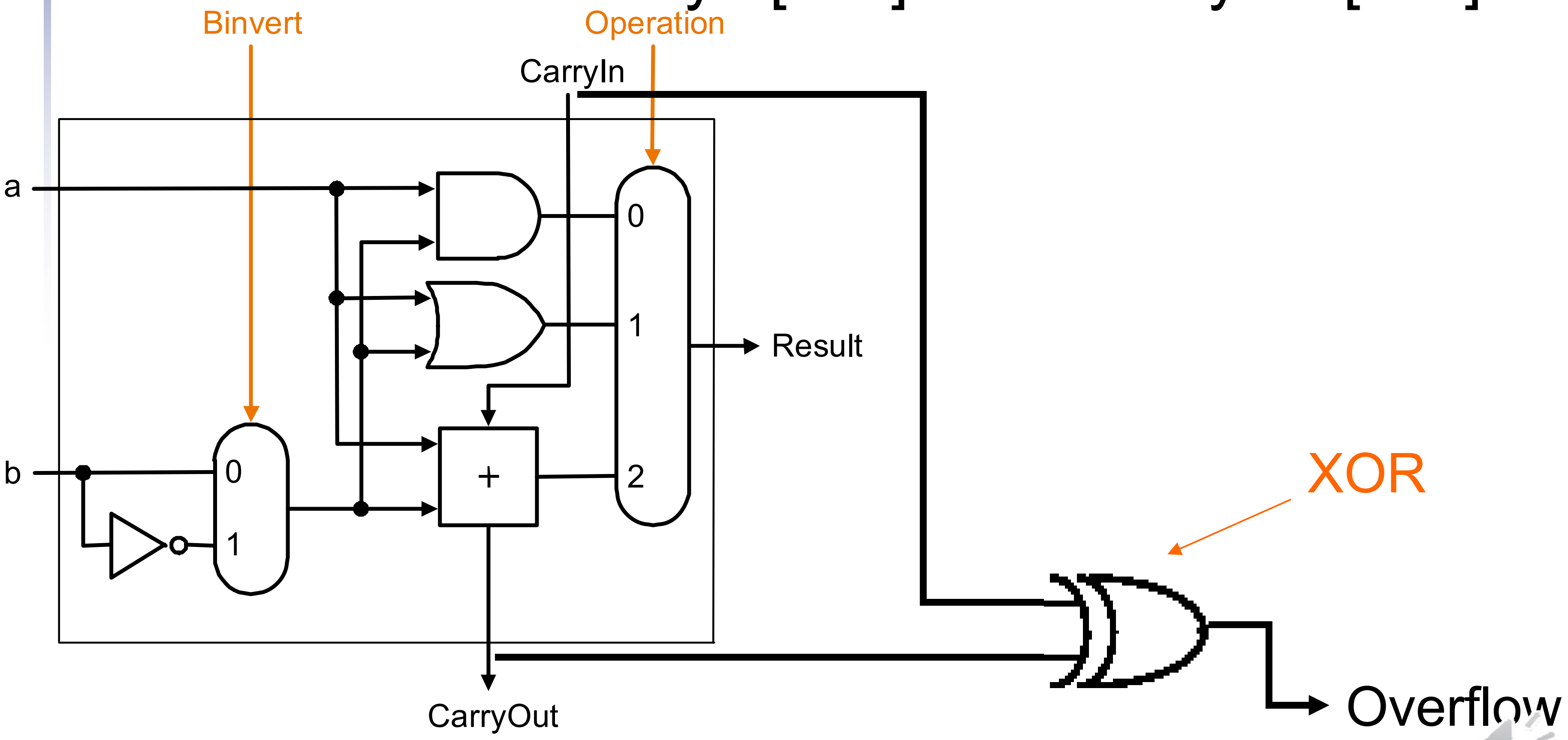
Subtraction?

- Expand our 1-bit ALU to include an inverter
  - 2's complement: take inverse of every bit and add 1



Overflow

- For N-bit ALU
  - Overflow = CarryIn[N-1] XOR CarryOut[N-1]

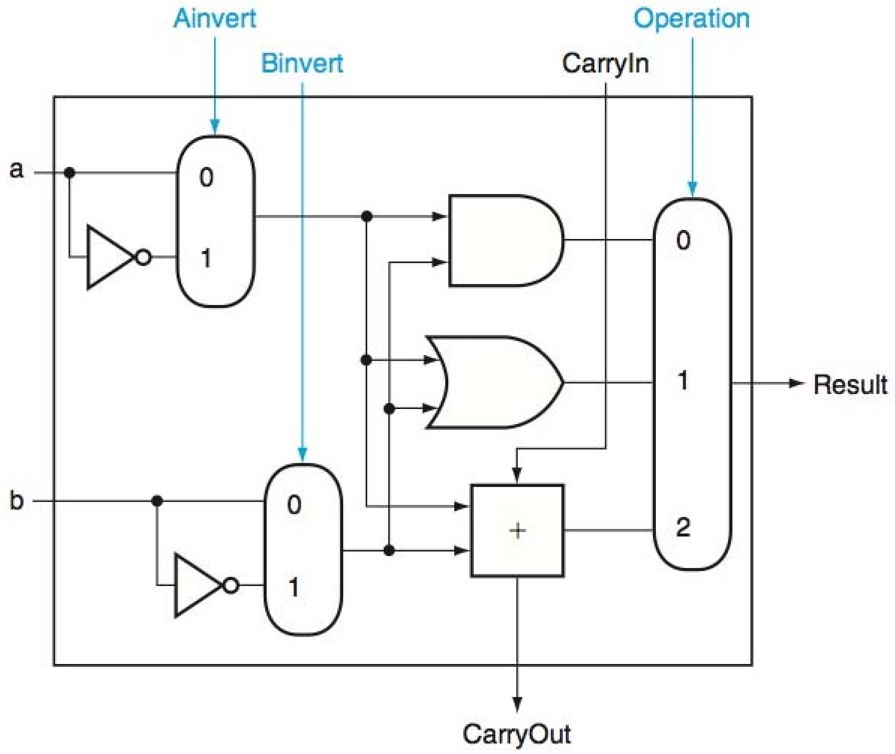


Most significant (N-1) bit ALU

Zero Detection

- Conditional Branches
- One big NOR gate
- Zero = $(\text{Result}_{N-1}+\text{Result}_{N-2}+\ldots.\ \text{Result}_1+\text{Result}_0)$
- Any non-zero result will cause zero detection output to be zero

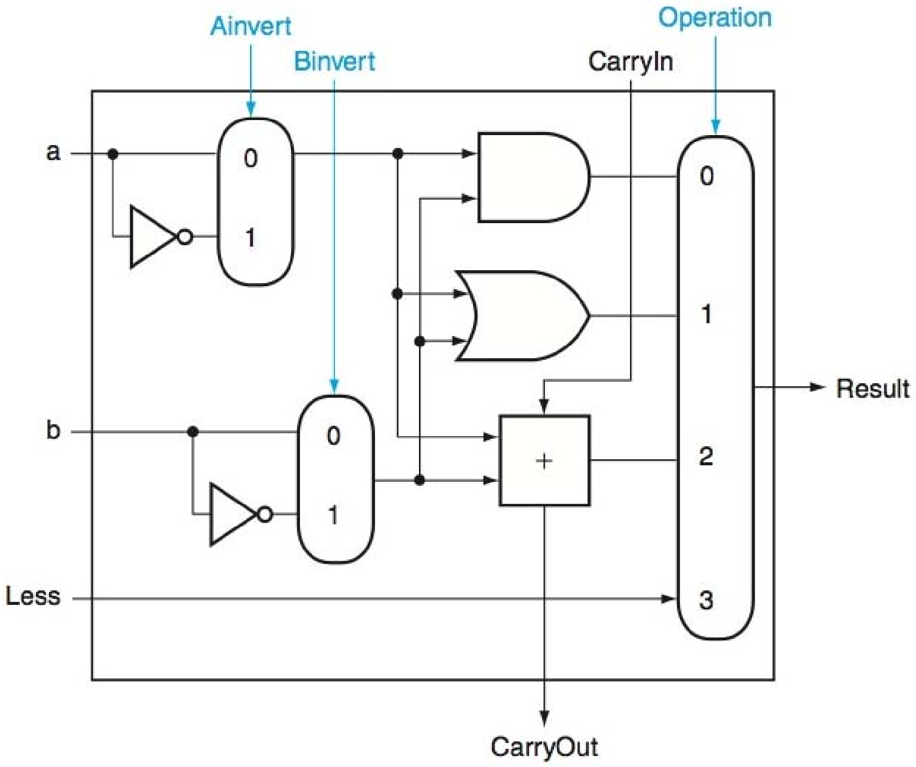
NOR


Ainvert
Binvert
CarryIn
Operation
a
b
0
1
0
1
+
0
1
2
Result
CarryOut

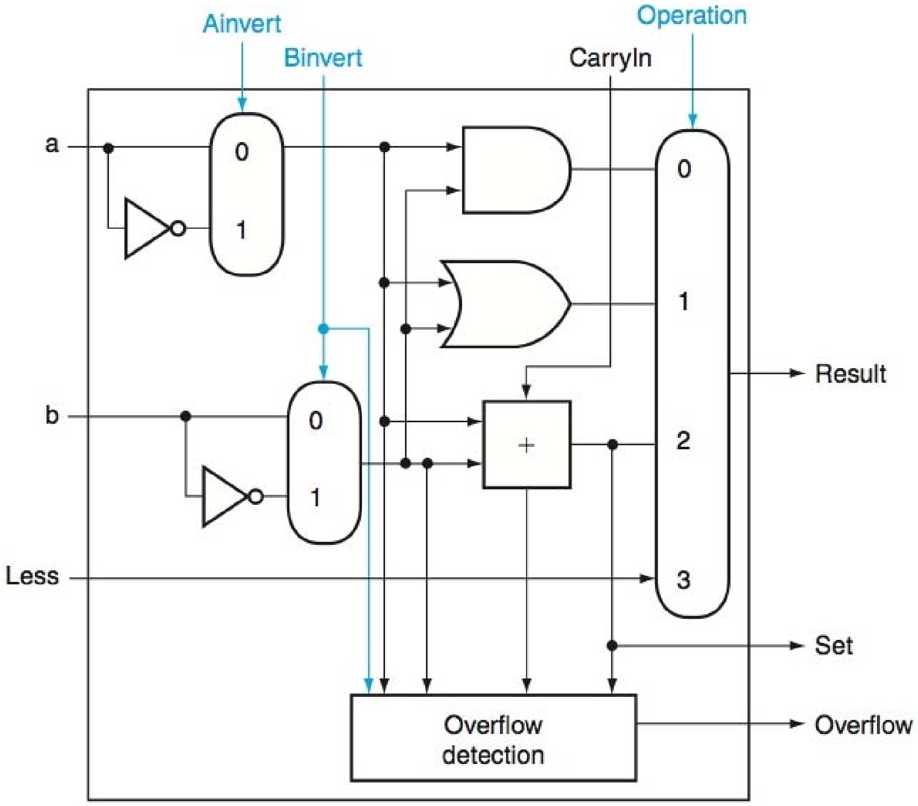
Set-On-Less-Than (SLT)

- SLT produces a 1 if rs < rt, and 0 otherwise
    - all but least significant bit will be 0
    - how do we set the least significant bit?
    - can we use subtraction?
        - rs - rt < 0
    - set the least significant bit to the sign-bit of (rs - rt)
- New input: LESS
- New output: SET

SLT Implementation

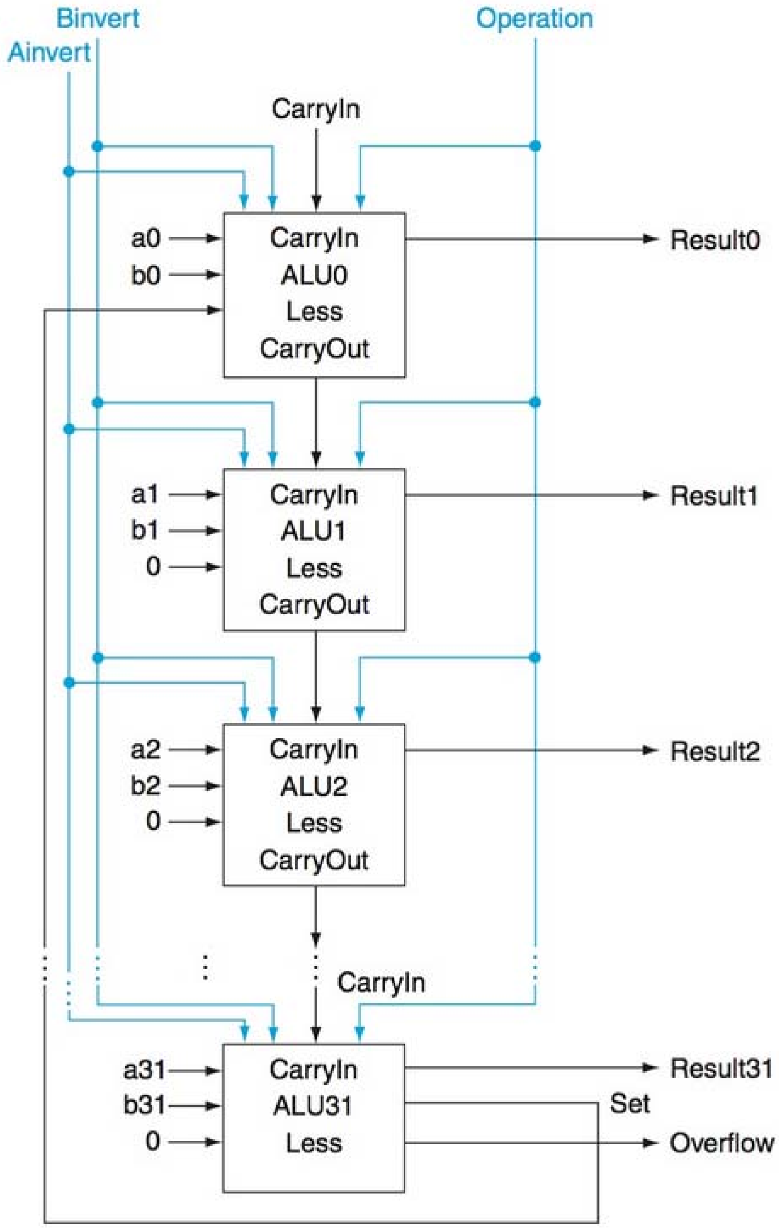


All but MSB



Most Significant Bit

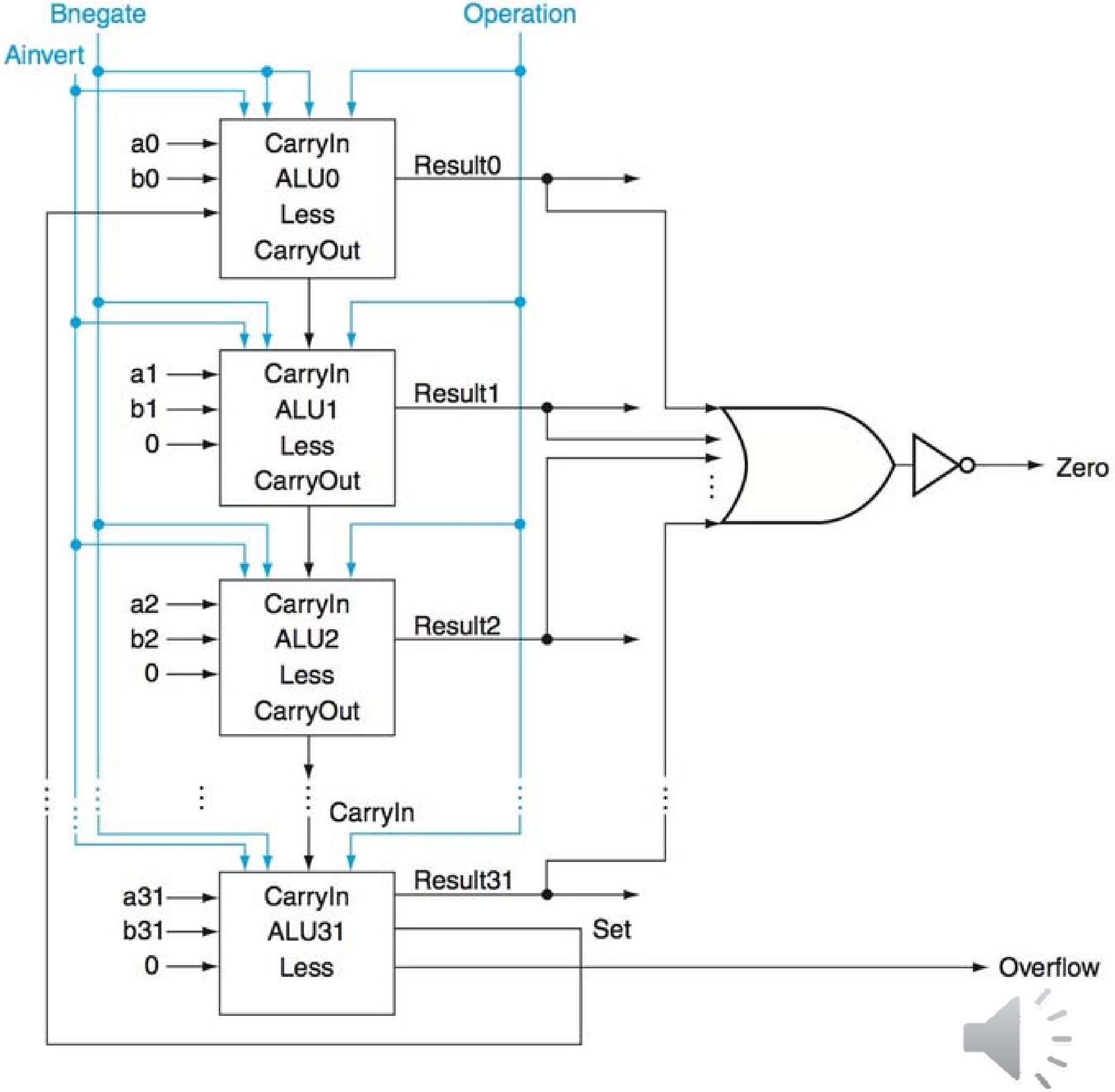
SLT Implementation



- Set of MSB is connected to Less of LSB!

Final ALU

- You should feel comfortable identifying what signals accomplish:
  - add
  - sub
  - and
  - or
  - nor
  - slt


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