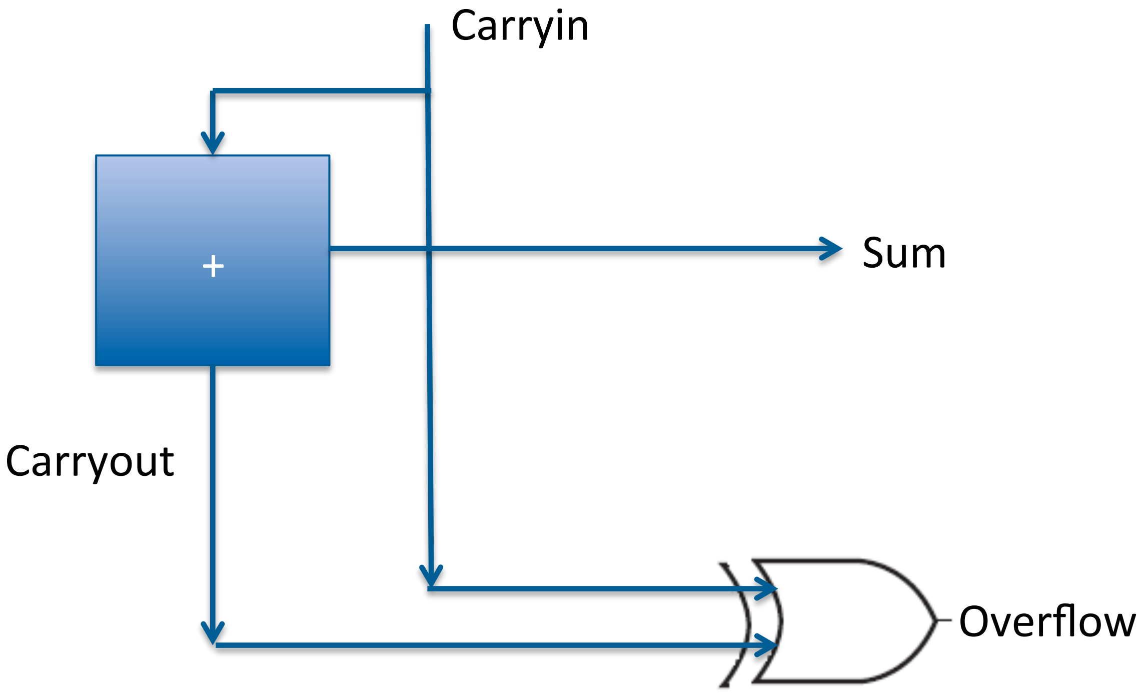
Overflow Detection

- Overflow exists if the result is larger than the register
- Mismatching carry-in and carry-out of the MSB indicates overflow
- Carry out alone signals overflow only for unsigned numbers
- Signed arithmetic requires a separate overflow indicator bit
- Negative results have MSB = 1, if no overflow occurs
- Positive results have MSB = 0, if no overflow occurs

Overflow Detection



For MSB, compare carryin with carryout

Mismatch indicates signed overflow

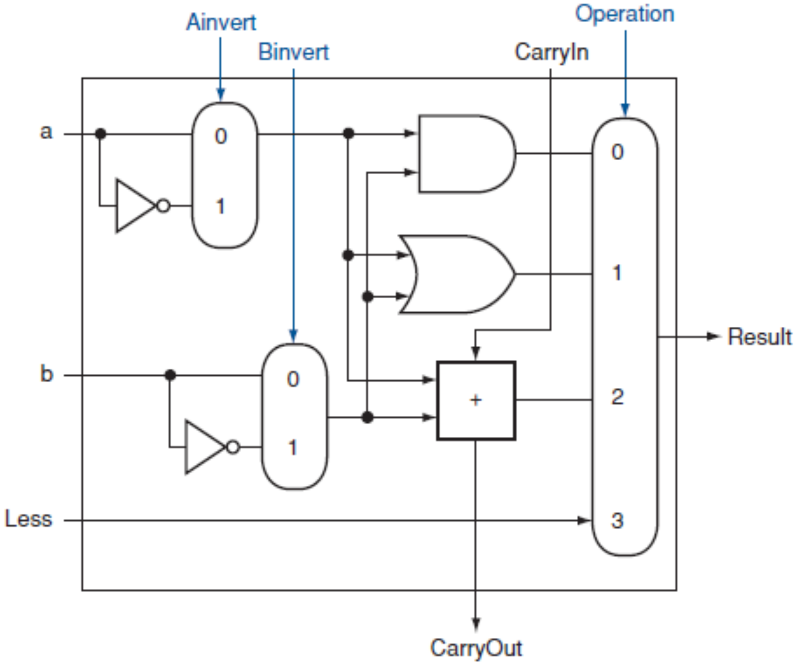
ALU for MIPS Core

- The ALU must also support slt (set on less than) instruction
- The slt instruction generates 1 in the result if a < b
- Otherwise, it generates 0
- If a-b < 0, then a < b (indicated by sign bit in result)
- Assumes no overflow occur

ALU for MIPS Core

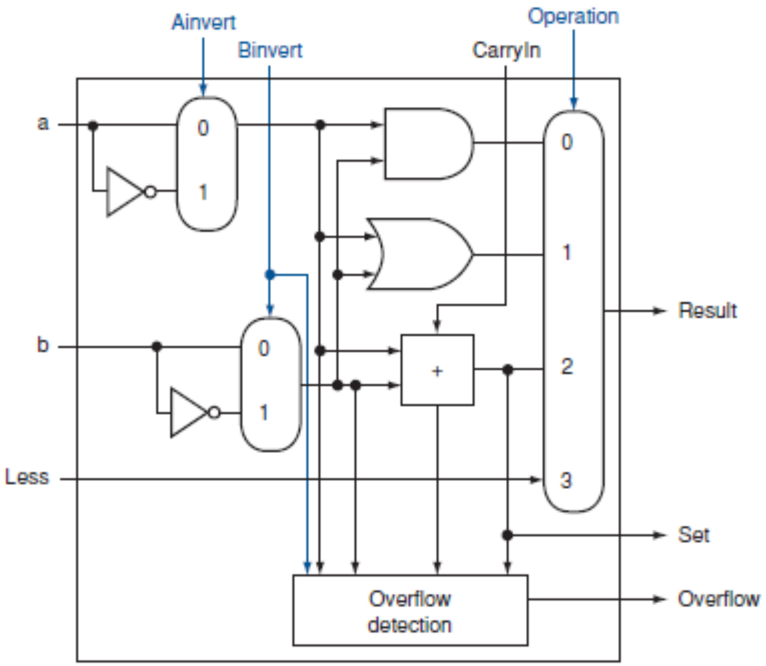
- A new input, Less, is included for ALU
- Less = 0 for the high order 31 bit positions
- LSB of result = 1 if a < b, otherwise LSB = 0
- Logic is only needed in the ALU for the MSB to generate both Overflow and Set (replaces LSB of result)

ALU for MIPS Core



ALU for all but the MSB position

ALU for MIPS Core



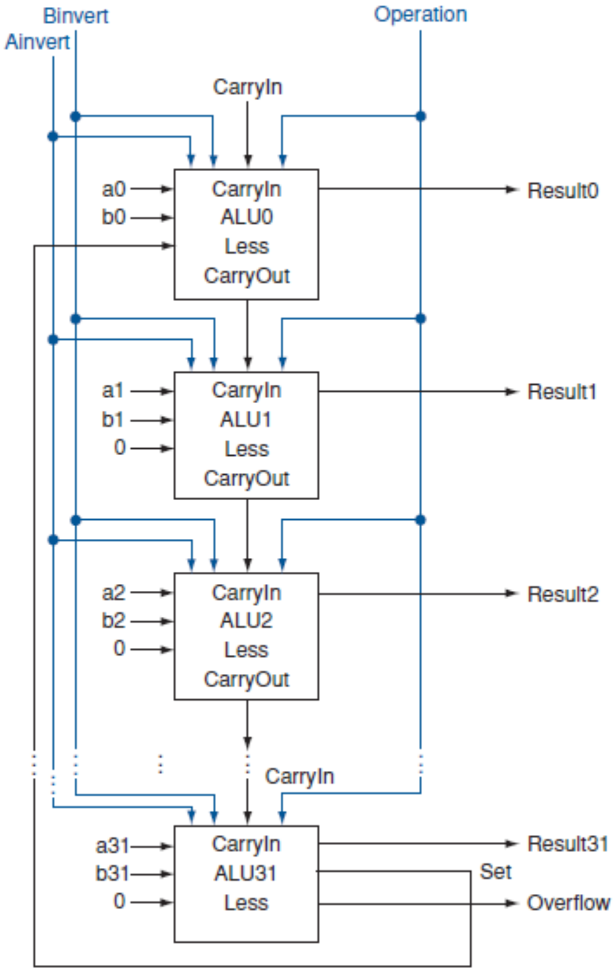
ALU for the MSB position

Includes logic to generate Set bit for slt instruction
Also generate overflow flag

ALU for MIPS Core

Includes support for slt

Set from MSB routed back to LSB of result



ALU for MIPS Core

- Branch on equal (beq) requires zero flag output from ALU
- Branch is taken if zero flag = 1 (PC is loaded with target address)
- Otherwise PC+4 is used as address of next instruction
- If a-b = 0, a and b are equal, so zero flag is set to 1
- NOR of all result bits yields zero flag

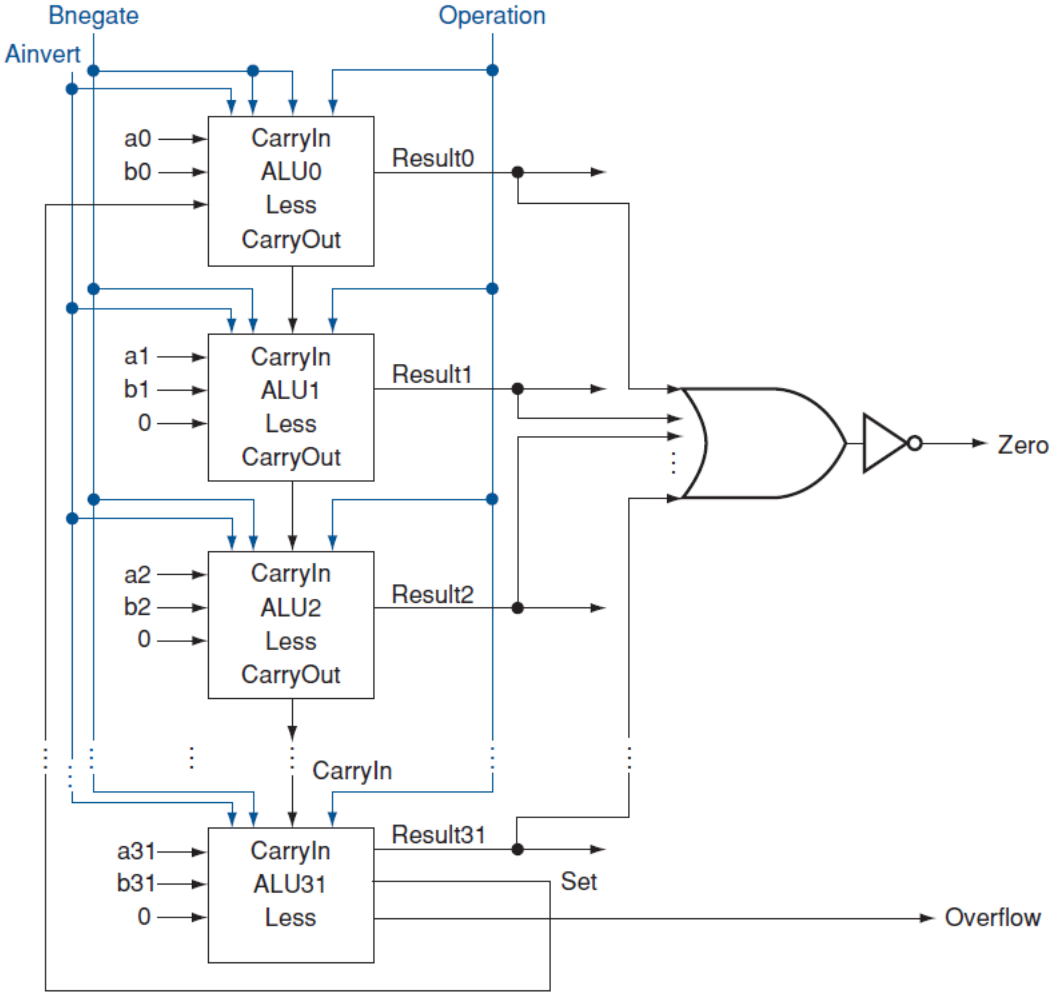
$$\text{Zero} = (\overline{\text{Result31} + \text{Result30} + \ldots + \text{Result2} + \text{Result1} + \text{Result0}})$$

ALU for MIPS Core

Final 32-bit ALU

Bnegate negates b (Carryin for LSB = 1 and all b bits are inverted)

So a-b is computed



Supports all operations needed for the core MIPS instruction subset