Our ultimate goal: building the MIPS datapath
Arithmetic Logic Unit (ALU)

ALU operation:

000 = and
001 = or
010 = add
110 = subtract
111 = slt

ALU operation:

000 = and
001 = or
010 = add
110 = subtract
111 = slt
What about subtraction \((a - b)\)?

- Two's complement approach: just negate \(b\) and add.
- How do we negate?

- The solution:
Tailoring the ALU to the MIPS datapath

• Need to support the set-on-less-than instruction
  \[ \text{slt } \text{rd, rs, rt} \]
  – \text{slt} is an arithmetic instruction
  – produces a 1 if \( \text{rs} < \text{rt} \) and 0 otherwise
  – use subtraction: \((a-b) < 0\) implies \(a < b\)

• Need to support test for equality (beq \$t5, \$t6, label)
  – use subtraction: \((a-b) = 0\) implies \(a = b\) \(\Rightarrow\) \text{Zero}=1
Supporting slt

Operation

CarryIn

Less

CarryOut

Result

Set

Overflow

Overflow detection

ALU0

ALU1

ALU2

ALU31

a0

b0

a1

b1

0

a2

b2

0

a31

b31

0

Binvert

CarryIn
Test for equality and complete ALU (3-bit control)

ALU operation:
- 000 = and
- 001 = or
- 010 = add
- 110 = subtract
- 111 = slt

Control lines

<table>
<thead>
<tr>
<th>Bnegate</th>
<th>Operation</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>00</td>
<td>and</td>
</tr>
<tr>
<td>0</td>
<td>01</td>
<td>or</td>
</tr>
<tr>
<td>0</td>
<td>10</td>
<td>add</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>sub</td>
</tr>
<tr>
<td>1</td>
<td>11</td>
<td>slt</td>
</tr>
</tbody>
</table>

- Note: zero is a 1 when the result is zero!
Add NOR and NAND operations
Final ALU (4-bit control)

ALU Control lines

<table>
<thead>
<tr>
<th>Ainvert</th>
<th>Bnega te</th>
<th>Operation</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>00</td>
<td>and</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>01</td>
<td>or</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>10</td>
<td>add</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>10</td>
<td>sub</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>11</td>
<td>slt</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>00</td>
<td>nor</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>01</td>
<td>nand</td>
</tr>
</tbody>
</table>