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

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

\[
\text{slt \ rd, rs, rt}
\]

– \text{slt} is an arithmetic instruction
– produces a 1 if \( rs < rt \) and 0 otherwise
– use subtraction: \((a-b) < 0 \) implies \( a < b \)

• Need to support test for equality (\text{beq \$t5, \$t6, label})

– use subtraction: \((a-b) = 0 \) implies \( a = b \)
Supporting slt

```
Supporting slt

[Diagram of circuit diagram with labeled inputs and outputs, showing the flow of data and operations such as Binvert, CarryIn, Operation, ALU0, Less, CarryOut, Result0, ALU1, Less, CarryOut, Result1, ALU2, Less, CarryOut, Result2, ALU31, Less, CarryOut, Result31.]
```

```
[Diagram of circuit diagram with labeled inputs and outputs, showing the flow of data and operations such as Binvert, CarryIn, Operation, ALU0, Result, CarryOut, ALU1, Less, CarryIn, CarryOut, ALU2, Less, CarryIn, CarryOut, ALU31, Less, CarryIn.]
```

```
Set
Overflow detection
```

```
Set
```

```
Result
```

```
Overflow
```

```
Overflow
```

```
Set
```

```
Result
```

```
Overflow
```

```
Set
```

```
Result
```

```
Overflow
```

```
Set
```

```
Result
```

```
Overflow
```

```
Set
```

```
Result
```

```
Overflow
```

```
Set
```

```
Result
```

```
Overflow
```

```
Set
```

```
Result
```

```
Overflow
```
Test for equality and complete ALU

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!