Lecture 14 - Control

"jump" instruction: \( \times \) 0

*Condition Code Registers

- single bit registers - describes attributes of most recent arithmetic or logical operations.

CF:

ZF:

SF:

OF:

1). Carry Flag

\[ t = a + b \]

\[ t < a \]

t, a \in \text{unsigned}

<table>
<thead>
<tr>
<th>( a )</th>
<th>( b )</th>
<th>( t )</th>
<th>CF</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>11</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>11</td>
<td>00</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>01</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>
2) **Zero Flag**

\[ t = a + b \]

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>t</th>
<th>ZF</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00</td>
<td>11</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>01</td>
<td>11</td>
<td>00</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>11</td>
<td>01</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>11</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ \begin{align*}
00 &= 0 \\
01 &= +1 \\
10 &= -2 \\
11 &= -1 
\end{align*} \]

3) **Sign Flag**

\[ t = a + b \]

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>t</th>
<th>SF</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00</td>
<td>11</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>01</td>
<td>01</td>
<td>11</td>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>11</td>
<td>01</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>11</td>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>
4) **Overflow Flag** (Signed numbers).

\[ w = 2 \]

| -2 | -1 | 0 | 1 |

1. \((-2) + (-1) = -3 \Rightarrow \) negative overflow
2. \((+1) + (+1) = 2 \Rightarrow \) positive overflow
3. \(-2 + 1 = -1 \Rightarrow \) no overflow.

<table>
<thead>
<tr>
<th>(10 \quad (-2) \rightarrow a)</th>
<th>(01 \quad (+1) \rightarrow a)</th>
<th>(10 \quad (-2) \rightarrow a)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(+11 \quad (-1) \rightarrow b)</td>
<td>(+01 \quad (+1) \rightarrow b)</td>
<td>(+01 \quad (+1) \rightarrow b)</td>
</tr>
<tr>
<td>(01 \quad (+1) \rightarrow t)</td>
<td>(10 \quad (-2) \rightarrow t)</td>
<td>(11 \quad (-1) \rightarrow t)</td>
</tr>
</tbody>
</table>

**OF:** \((a < 0 \iff b < 0) \land (t < 0 \iff a < 0)\)

\[ a \text{ and } b \text{ should have the same sign.} \]

\[ t \text{ and } a \text{ should have opposite signs.} \]
All arithmetic and logical operations except lead set the condition codes.

*Comparison and Test*

1. CMP $S_2, S_1 \Rightarrow$ similar to SUB.
   Sets condition code registers based on the condition $S_1 - S_2$.
   * Does not alter any other registers.

2. TEST $S_2, S_1 \Rightarrow$ similar to AND.

Condition: $S_1 \& S_2$

*eg. To test bits.*

test al, 1

Test bit 0 of al.

(LSB)

test dx, 105h

(0x105)

Test bits 0, 2, and 8.
Accessing Condition Codes

1. Set a single byte to 0 or 1 based on CCs.
2. Conditionally jump to some other part of the program.
3. Conditionally transfer data.

SET \( D \rightarrow 1 \) byte register.

\* SET works based on the prev A/L operation or the CMP instruction.

\[ t = a - b \]

\[ t = a + \frac{b}{w} \]

SET (unsigned comparisons)

\[ t = a - b \]

if \( a - b < 0 \) \( \Rightarrow \) CF = 1

if \( a - b > 0 \) \( \Rightarrow \) CF = 0
seta D  
setae D  
setb D  
setbe D  

\[ D \leftarrow nCF \]  
\[ D \leftarrow \neg ZF \]  
\[ D \leftarrow nCF \]  
\[ D \leftarrow CF \]  
\[ D \leftarrow CF \oplus ZF \]  

Above (unsigned >)  
Above or equal (unsigned ≥)  
Below (unsigned <)  
Below or equal (unsigned ≤)  

\( \times \) SET (signed comparisons)  
2's complement.  
\[ t = a - \frac{t}{w} b \]  

1) \( a = b \), \( \Rightarrow t = 0. \)  
\[ \boxed{ZF = 1} \Rightarrow \text{equality.} \]  

2) \text{setl (set when less)} - signed comparison.  
   
i) No overflow (OF = 0)  
\[ a < b \text{ when } a \frac{t}{w} b < 0 \]  
\[ a \geq b \text{ when } a \frac{t}{w} b \geq 0 \]  
\[ \boxed{SF = 1} \]  
\[ \boxed{SF = 0} \]  

ii) With overflow (OF = 1)  
\[ a < b \text{ when } a \frac{t}{w} b > 0 \]  
\[ a > b \text{ when } a \frac{t}{w} b < 0 \]  
\[ \boxed{SF = 0} \]  
\[ \boxed{SF = 1} \]  

\[ SFI \]
Set D

\[ t = a \frac{t}{w} b \]

<table>
<thead>
<tr>
<th>a</th>
<th>00</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>01</td>
<td>01</td>
</tr>
<tr>
<td>t</td>
<td>11</td>
<td>-1</td>
</tr>
</tbody>
</table>

10  
-01  
---

00  
01  
101

\[ a = 10 \quad (-2) \]
\[ b = 01 \quad (+1) \]
\[ t = a - b = 01 \quad (+1) \]

With Overflow (le) \[ OF = 1 \]
\[ t > 0 \]
\[ -3 + 4 = +1 \]

\[ a = 00 \quad (0) \]
\[ b = 01 \quad (1) \]
\[ t = a - b = 01 \quad (-1) \]

Without overflow \[ OF = 0 \]
\[ a < b \]
\[ t < 0 \]

SF = 0

SF = 1
.; setl D
setle D
setg D
setge D

D ← SF ∧ OF
      (signed <)

D ← (SF ∧ OF) | ZF
      (signed ≤)

D ← ¬(SF ∧ OF) ∧ ¬ZF
      (signed >)

D ← ¬(SF ∧ OF)
      (signed ≥)

Jump instruction

jmp
jmp label
jmp *operand
e.g., jmp .L1
    jmp *%eax
    jmp *(%eax)

Direct jump

Indirect jump.

Jumps

absolute

PC relative.