

# Princeton University

## COS 217: Introduction to Programming Systems

### IA-32 Condition Codes

#### Condition Codes

Bits in the EFLAGS register

```
cmpl src, dest
```

Performs the subtraction  $dest - src$ , and sets the condition codes depending upon the difference:

Condition Code	Set When
ZF (zero flag)	Mathematically: The difference was 0. Physically: All bits of the difference were 0.
SF (sign flag)	Mathematically: The difference was negative. Physically: The most significant bit of the difference was 1.
CF (carry flag)	Mathematically: The difference was incorrect when we view the operands and difference as <b>unsigned</b> integers. Physically: A borrow occurred into the most significant bit.
OF (overflow flag)	Mathematically: The difference was incorrect when we view the operands and difference as <b>signed</b> integers. Physically: The borrow into the most significant bit differed from the borrow out of the most significant bit.

#### Conditional Control Transfer Instructions (Used After Comparing Unsigned Numbers)

Instruction	Jump if and only if
je (jump iff equal)	ZF
jne (jump iff not equal)	$\sim$ ZF
jb (jump iff below)	CF
jae (jump iff above or equal)	$\sim$ CF
jbe (jump iff below or equal)	CF   ZF
ja (jump iff above)	$\sim$ (CF   ZF)

Examples (assuming a 5-bit computer for simplicity):

Comparison	Subtraction	Resulting Condition Codes	Execution of jb
12 and 6	<pre> 01100  12 -00110  -6 ----- 00110   6 </pre>	CF = 0 (unsigned diff was correct)	CF == 0 So don't jump
6 and 12	<pre> 00110   6 -01100 -12 ----- 11010  26 </pre>	CF = 1 (unsigned diff was incorrect)	CF == 1 So jump

## Conditional Control Transfer Instructions (Used After Comparing Signed Numbers)

Instruction	Jump if and only if
je (jump iff equal)	ZF
jne (jump iff not equal)	$\sim$ ZF
jl (jump iff less than)	SF ^ OF
jge (jump iff greater than or equal)	$\sim$ (SF ^ OF)
jle (jump iff less than or equal)	(SF ^ OF)   ZF
jg (jump iff greater than)	$\sim$ ((SF ^ OF)   ZF)

### Examples (assuming a 5-bit computer for simplicity):

Comparison	Subtraction	Resulting Condition Codes	Execution of jl
12 and 6	<pre> 01100  12 -00110  -6 ----- 00110   6           </pre>	SF = 0 (diff was positive) OF = 0 (signed diff was correct)	$(SF \wedge OF) == 0$ So don't jump
-6 and -12	<pre> 11010  -6 -10100 -12 ----- 00110   6           </pre>	SF = 0 (diff was positive) OF = 0 (signed diff was correct)	$(SF \wedge OF) == 0$ So don't jump
6 and 12	<pre> 00110   6 -01100 -12 ----- 11010  -6           </pre>	SF = 1 (diff was negative) OF = 0 (signed diff was correct)	$(SF \wedge OF) == 1$ So jump
-12 and -6	<pre> 10100 -12 -11010 -6 ----- 11010  -6           </pre>	SF = 1 (diff was negative) OF = 0 (signed diff was correct)	$(SF \wedge OF) == 1$ So jump
-12 and 6	<pre> 10100 -12 -00110 -6 ----- 01110  14           </pre>	SF = 0 (diff was positive) OF = 1 (signed diff was incorrect)	$(SF \wedge OF) == 1$ So jump
-6 and 12	<pre> 11010  -6 -01100 -12 ----- 01110  14           </pre>	SF = 0 (diff was positive) OF = 1 (signed diff was incorrect)	$(SF \wedge OF) == 1$ So jump
6 and -12	<pre> 00110   6 -10100 -12 ----- 10010 -14           </pre>	SF = 1 (diff was negative) OF = 1 (signed diff was incorrect)	$(SF \wedge OF) == 0$ So don't jump
12 and -6	<pre> 01100  12 -11010 -6 ----- 10010 -14           </pre>	SF = 1 (diff was negative) OF = 1 (signed diff was incorrect)	$(SF \wedge OF) == 0$ So don't jump

Copyright © 2009 by Robert M. Dondero, Jr.