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 to 1 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 **unsigned** 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 **signed** 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)`	`~ZF`
`jb  (jump iff below)`	`CF`
`jae (jump iff above or equal)`	`~CF`
`jbe (jump iff below or equal)`	`CF | ZF`
`ja  (jump iff above)`	`~(CF | ZF)`

Why does `jb` jump if and only if `CF`? Informal explanation:

```
(1) largenum - smallnum => correct result => ~CF => don't jump

(2) smallnum - largenum => incorrect result => CF => jump
```

So jump if and only if `CF`.

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)`	`~ZF`
`jl  (jump iff less than)`	`OF ^ SF`
`jge (jump iff greater than or equal)`	`~(OF ^ SF)`
`jle (jump iff less than or equal)`	`(OF ^ SF) | ZF`
`jg  (jump iff greater than)`	`~((OF ^ SF) | ZF)`

Why does `jl` jump if and only if `(OF ^ SF)`? Informal explanation:

```
(1) posnum - posnum => correct result => ~OF => jump iff SF

(2) negnum - negnum => correct result => ~OF => jump iff SF

(3) posnum - negnum, correct result => ~OF, ~SF => don't jump

(4) posnum - negnum, incorrect result => OF, SF => don't jump

(5) negnum - posnum, correct result => ~OF, SF => jump

(6) negnum - posnum, incorrect result => OF, ~SF => jump
```

So jump if and only if `(OF ^ SF)`.