Condition Codes (Implicit Setting)

- **Implicitly set by arithmetic operations**
  - (think of it as side effect)

Example: `addq` *Src,Dest* ↔ `t = a+b`

Single bit registers

CF	Carry Flag (for unsigned)	**SF**	Sign Flag (for signed)
ZF	Zero Flag	**OF**	Overflow Flag (for signed)

- **CF set** if carry out from most significant bit (unsigned overflow)
- **ZF set** if `t == 0`
- **SF set** if `t < 0` (as signed)
- **OF set** if twos-complement (signed) overflow
  `(a>0 && b>0 && t<0) || (a<0 && b<0 && t>=0)`

Not set by `leaq` instruction (beware!)

Condition Codes (Explicit Setting: Compare)

- **Explicit Setting by Compare Instruction**

  `cmpq` *Src2,Src1*

  `cmpq b,a` like computing `a-b` without setting destination

Single bit registers

CF	Carry Flag (for unsigned)	**SF**	Sign Flag (for signed)
ZF	Zero Flag	**OF**	Overflow Flag (for signed)

- **CF set** if carry out from most significant bit (used for unsigned comparisons)
- **ZF set** if `a == b`
- **SF set** if `(a-b) < 0` (as signed)
- **OF set** if twos complement (signed) overflow
  `(a>0 && b<0 && (a-b)<0) || (a<0 && b>0 && (a-b)>0)`

Condition Codes (Explicit Setting: Test)

- **Explicit Setting by Test instruction**

  `testq` *Src2,Src1*

  `testq b,a` like computing `a & b` without setting destination

  - Sets condition codes based on value of *Src1* & *Src2*
  - Useful to have one of the operands be a mask

Single bit registers

CF	Carry Flag (for unsigned)	**SF**	Sign Flag (for signed)
ZF	Zero Flag	**OF**	Overflow Flag (for signed)

- ZF set if `a&b == 0`
- SF set if `a&b < 0`

- `testq %rax, %rax`
  - Sets SF and ZF, check if rax is +,0,-