

Lecture 14 - Control

"jump" instruction. (X) (1)

* Condition Code Registers

single bit registers. —

describes attributes of most recent arithmetic or logical operations.

CF:

ZF:

SF:

OF:

$$t = a + b$$

1). Carry Flag

$$t < a$$

$t, a \in \text{unsigned}$

eg.

a	b	t	CF
00	11	11	0
01	11	00	1
10	11	01	1
11	11	10	1

2) Zero Flag

②

$$t == 0$$

$$t = a + b$$

eg.

a	b	t	ZF
00	00 11	11	0
01	11	00	1
10	11	01	0
11	11	10	0

00 = 0
 01 = +1
 10 = -2
 11 = -1

3) Sign Flag

$$t = a + b$$

$$t < 0$$

a	b	t	SF
00	11	11	1
01	11	00	0
10	11	01	0
11	11	10	1

4) Overflow Flag (Signed numbers)

$$\boxed{w=2}$$

$$2^w = 2^2 = \underline{4}$$

-2 -1 0 1

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

①
$$\begin{array}{r} 10 \text{ } (-2) \rightarrow a \\ + 11 \text{ } (-1) \rightarrow b \\ \hline 01 \text{ } (+1) \rightarrow t \end{array}$$

②
$$\begin{array}{r} 01 \text{ } (+1) \rightarrow a \\ + 01 \text{ } (+1) \rightarrow b \\ \hline 10 \text{ } (-2) \rightarrow t \end{array}$$

③
$$\begin{array}{r} 10 \text{ } (-2) \rightarrow a \\ + 01 \text{ } (+1) \rightarrow b \\ \hline 11 \text{ } (-1) \rightarrow t \end{array}$$

OF: $(a < 0 == b < 0) \Leftrightarrow (t < 0 != a < 0)$

\swarrow
 a and b should have the same sign.

\searrow
 t and a should have opposite signs.

(X) All arithmetic and logical operations except leal set the condition codes.

* Comparison and Test

1. CMP S2, S1 \Rightarrow similar to SUB.
sets condition code registers based on the condition ~~S1 - S2~~ $S1 - S2$.

* Does not alter any other registers.

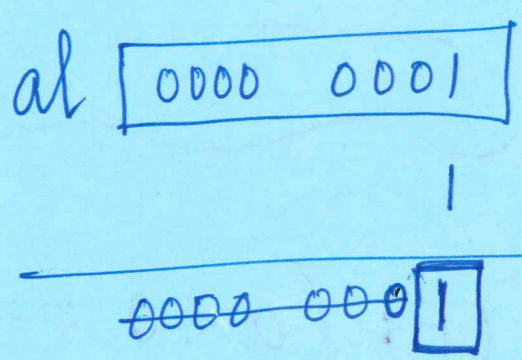
2. TEST S2, S1 \Rightarrow similar to AND.

Condition: S1 & S2

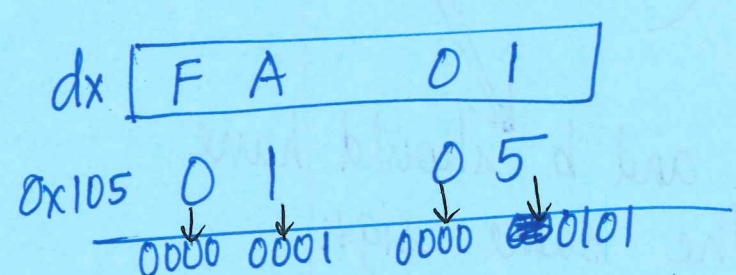
eg. To test bits.

test al, 1

test bit 0 of al.
(LSB)



test dx, 105h
(001)
0x105



Tests bits 0, 2, and 8.

Accessing Condition Codes ⁽⁵⁾

CC - 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 $\overset{D}{\downarrow}$ 1 byte register.

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

$$\boxed{t = a - b}$$

CMP $\begin{matrix} b & a \\ a & b \end{matrix}$

$$t = a - \frac{t}{w} b$$

SET (unsigned comparisons)

$$t = a - b$$

if $a - b < 0 \Rightarrow CF = 1$

if $a - b > 0 \Rightarrow CF = 0$

⑥

seta D	$D \leftarrow NCF \ \& \ NZF$	Above (unsigned $>$)
setae D	$D \leftarrow NCF$	Above or equal (unsigned $>=$)
setb D	$D \leftarrow CF$	Below (unsigned $<$)
setbe D	$D \leftarrow CF \ \& \ ZF$	Below or equal (unsigned $<=$)

~~SET~~ (signed comparisons 2's complement.)

$t = a - \frac{t}{w} b$

1) sete $a = b$, $\Rightarrow t = 0$. $ZF = 1$ \Rightarrow equality.

2) setl (set when less) - signed comparison.

i) No overflow ($OF = 0$)

$a < b$ when $a - \frac{t}{w} b < 0$ (ie) $SF = 1$

$a \geq b$ when $a - \frac{t}{w} b \geq 0$ (ie) $SF = 0$

ii) With overflow ($OF = 1$)

$a < b$ when $a - \frac{t}{w} b > 0$ (positive overflow) (ie) $SF = 0$

$a > b$ when $a - \frac{t}{w} b < 0$ (negative overflow) $SF = 1$

set 1 D

(7)

$$t = a - \frac{t}{w}b$$

a	b	t
00 (0)	01	11 (-1)
01 (1)	01	00 (0)
10 (-2)	01	01 (-3) → +1
11 (-1)	01	10 (-2)

$$\begin{array}{r} 10 \\ -01 \\ \hline 01 \end{array}$$

$$\begin{array}{r} 00 \\ 01 \\ \hline 101 \end{array}$$

$$\begin{array}{r} a = 10 \quad (-2) \\ - b = 01 \quad (+1) \\ \hline \end{array}$$

$$a < b$$

With Overflow (ie) OF = 1

$$t = a - b = 01 \quad (+1)$$

$$t > 0$$

SF = 0

$$-3 + 4 = (+1)$$

Without overflow

$$\begin{array}{r} a = 00 \quad (0) \\ - b = 01 \quad (1) \\ \hline \end{array}$$

$$a < b$$

OF = 0

$$t = a - b = 11 \quad (-1)$$

$$t < 0$$

SF = 1

∴ setl D

ⓑ

$$D \leftarrow SF \wedge OF$$

(signed <)

setle D

$$D \leftarrow (SF \wedge OF) \vee ZF$$

(signed <=)

setg D

$$D \leftarrow \neg(SF \wedge OF) \wedge \neg ZF$$

(signed >)

setge D

$$D \leftarrow \neg(SF \wedge OF)$$

(signed >=)

Jump instruction

jmp
jmp label

Direct jump
indirect jump.

jmp *operand

eg. jmp .L1
jmp *%eax
jmp *(%eax)

Jumps
absolute PC relative.