

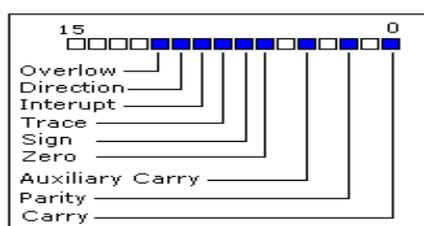
SISTEMI di elaborazione e trasmissione delle informazioni

I.T.I.S. "Euganeo" Este, A.S. 2004/2005

Jxx - Jump Instructions Table

Mnemonic	Meaning	Jump Condition
JA	Jump if Above	CF=0 and ZF=0
JAЕ	Jump if Above or Equal	CF=0
JB	Jump if Below	CF=1
JBE	Jump if Below or Equal	CF=1 or ZF=1
JC	Jump if Carry	CF=1
JCXZ	Jump if CX Zero	CX=0
JE	Jump if Equal	ZF=1
JG	Jump if Greater (signed)	ZF=0 and SF=OF
JGE	Jump if Greater or Equal (signed)	SF=OF
JL	Jump if Less (signed)	SF != OF
JLE	Jump if Less or Equal (signed)	ZF=1 or SF != OF
JMP	Unconditional Jump	unconditional
JNA	Jump if Not Above	CF=1 or ZF=1
JNAE	Jump if Not Above or Equal	CF=1
JNB	Jump if Not Below	CF=0
JNBE	Jump if Not Below or Equal	CF=0 and ZF=0
JNC	Jump if Not Carry	CF=0
JNE	Jump if Not Equal	ZF=0
JNG	Jump if Not Greater (signed)	ZF=1 or SF != OF
JNGE	Jump if Not Greater or Equal (signed)	SF != OF
JNL	Jump if Not Less (signed)	SF=OF
JNLE	Jump if Not Less or Equal (signed)	ZF=0 and SF=OF
JNO	Jump if Not Overflow (signed)	OF=0
JNP	Jump if No Parity	PF=0
JNS	Jump if Not Signed (signed)	SF=0
JNZ	Jump if Not Zero	ZF=0
JO	Jump if Overflow (signed)	OF=1
JP	Jump if Parity	PF=1
JPE	Jump if Parity Even	PF=1
JPO	Jump if Parity Odd	PF=0
JS	Jump if Signed (signed)	SF=1
JZ	Jump if Zero	ZF=1

Most Arithmetic and Logic Instructions affect the processor status register (or Flags)



As you may see there are 16 bits in this register, each bit is called a flag and can take a value of 1 or 0.

Carry Flag (CF) - this flag is set to 1 when there is an unsigned overflow. For example when you add bytes 255 + 1 (result is not in range 0...255). When there is no overflow this flag is set to 0.

Zero Flag (ZF) - set to 1 when result is zero. For none zero result this flag is set to 0.

Sign Flag (SF) - set to 1 when result is negative. When result is positive it is set to 0. Actually this flag take the value of the most significant bit.

Overflow Flag (OF) - set to 1 when there is a signed overflow. For example, when you add bytes 100 + 50 (result is not in range - 128...127).

Parity Flag (PF) - this flag is set to 1 when there is even number of one bits in result, and to 0 when there is odd number of one bits. Even if result is a word only 8 low bits are analyzed!

Auxiliary Flag (AF) - set to 1 when there is an unsigned overflow for low nibble (4 bits).

Interrupt enable Flag (IF) - when this flag is set to 1 CPU reacts to interrupts from external devices.

Direction Flag (DF) - this flag is used by some instructions to process data chains, when this flag is set to 0 - the processing is done forward, when this flag is set to 1 the processing is done backward.

Dato che molto spesso capita di dover scegliere la sequenza di istruzioni da eseguire in base al risultato di un confronto tra due valori (per esempio, salta se $A > 0$) i progettisti del microprocessore 8086 hanno pensato di inserire nel set di istruzioni un particolare tipo di operazione, detta *compare*, che permette di effettuare la sottrazione tra i due operandi, modificando i flag, *senza però memorizzare il risultato*.

Si tratta, in pratica di mettere a confronto aritmetico i due valori e registrare, *senza modificare i valori stessi*, l'esito del confronto a livello dei flag.

La forma sintattica dell'istruzione è:

CMP *op1, op2*

ove *op1* e *op2* devono avere la medesima dimensione e dove *op1* non può mai essere una costante.

Per esempio, l'istruzione:

CMP AX, BX

modifica i flag come SUB AX, BX ma il risultato della sottrazione non viene memorizzato in AX.

L'istruzione di confronto viene generalmente posta subito prima di un'istruzione di salto condizionato, in modo che questo venga effettuato in base al risultato del confronto stesso.

In questo caso, alcune tra le istruzioni di salto possono assumere un significato logico simile a quello usato nei linguaggi evoluti.

Formato dell'istruzione		Descrizione	
JA	<i>indirizzo di riferimento</i>	salta se $op1 > op2$	<i>op1 e op2 interi senza segno</i>
JAE	<i>indirizzo di riferimento</i>	salta se $op1 \geq op2$	<i>op1 e op2 interi senza segno</i>
JB	<i>indirizzo di riferimento</i>	Salta se $op1 < op2$	<i>op1 e op2 interi senza segno</i>
JBE	<i>indirizzo di riferimento</i>	Salta se $op1 \leq op2$	<i>op1 e op2 interi senza segno</i>
JE	<i>indirizzo di riferimento</i>	Salta se $op1 = op2$	
JG	<i>indirizzo di riferimento</i>	Salta se $op1 > op2$	<i>op1 e op2 interi con segno</i>
JGE	<i>indirizzo di riferimento</i>	Salta se $op1 \geq op2$	<i>op1 e op2 interi con segno</i>
JL	<i>indirizzo di riferimento</i>	Salta se $op1 < op2$	<i>op1 e op2 interi con segno</i>
JLE	<i>indirizzo di riferimento</i>	Salta se $op1 \leq op2$	<i>op1 e op2 interi con segno</i>
JNA	<i>indirizzo di riferimento</i>	Salta se $op1 \leq op2$	<i>op1 e op2 interi senza segno</i>
JNAE	<i>indirizzo di riferimento</i>	Salta se $op1 < op2$	<i>op1 e op2 interi senza segno</i>
JNB	<i>indirizzo di riferimento</i>	Salta se $op1 \geq op2$	<i>op1 e op2 interi senza segno</i>
JNBE	<i>indirizzo di riferimento</i>	Salta se $op1 > op2$	<i>op1 e op2 interi senza segno</i>
JNE	<i>indirizzo di riferimento</i>	Salta se $op1 \neq op2$	
JNG	<i>indirizzo di riferimento</i>	Salta se $op1 \leq op2$	<i>op1 e op2 interi con segno</i>
JNGE	<i>indirizzo di riferimento</i>	Salta se $op1 < op2$	<i>op1 e op2 interi con segno</i>
JNL	<i>indirizzo di riferimento</i>	Salta se $op1 \geq op2$	<i>op1 e op2 interi con segno</i>
JNLE	<i>indirizzo di riferimento</i>	Salta se $op1 > op2$	<i>op1 e op2 interi con segno</i>
JNZ	<i>indirizzo di riferimento</i>	Salta se $op1 \neq op2$	
JZ	<i>indirizzo di riferimento</i>	Salta se $op1 = op2$	