

Kivonás: **SUB** op1, op2
Működése: op1 ← op1 - op2

Előjelkiterjesztés 16 bitről 32 bitre: **CWD**

0 operandusú.

Működése: DX:AX ← AX (előjelhelyesen), azaz:

- ha AX legfelső bitje 1, akkor DX ← FFFFh
 - ha AX legfelső bitje 0, akkor DX ← 0000h
-

Előjeltelen osztás: **DIV** op

Működése:

- ha op 8 bites
 - AL ← AX / op
 - AH ← AX % op
 - ha op 16 bites
 - AX ← DX:AX / op
 - DX ← DX:AX % op
-

Túlsordulás esetén abortál a program. „op” nem lehet közvetlen operandus.

Előjeles osztás: **IDIV** op

Működése a DIV-ével megegyező, de előjeles számokra működik. A nem 0 maradék előjele az osztandóéval egyezik meg.

Összehasonlítás: **CMP** op1, op2

Működése: op1 - op2 az eredmény tárolása nélkül, a flagek viszont beállnak a kivonásnak megfelelően.

Feltételes ugrások előtt használjuk.

Feltételes ugrások:

Mind egyik 1 operandusú, 8 bites IP-relatív értéket vár. Ezt nem szoktuk a kódban kézzel megadni („bedrótozni”), helyette címkeket írunk. A tényleges IP-relatív értéket az assembler számolja ki. Az operandus méretéből következik, hogy visszafelé 128, előre pedig 127 bájtal ugorhatunk, a következő utasítás címétől számítva.

Összehasonlítás (CMP) eredménye alapján:

előjeles	JG	JL	JGE	JLE	JNG	JNL	JNGE	JNLE
op1 ? op2	>	<	>=	<=	!>	!<	!>=	!<=
előjeltelen	JA	JB	JAE	JBE	JNA	JNB	JNAE	JNBE

JE: ==, JNE: !=

Betűk eredete: **Greater, Less, Above, Below, Equal, Not**

Közvetlenül flagek alapján:

JZ, JC, JS, JO, JP (ugrás, ha a flag igaz); JNZ, JNC, JNS, JNO, JNP (ha a flag hamis)

A 8086 P flagje páros paritás esetén 1, ezért:

- JPE (**J**ump if **P**arity **E**ven) ugyanaz, mint JP
- JPO (**J**ump if **P**arity **O**dd) ugyanaz, mint JNP

A CMP kivonás alapján állítja be a flageket, így Z akkor és csak akkor 1, ha a két szám egyenlő volt. Ezért:

- JZ ugyanaz, mint JE
- JNZ ugyanaz, mint JNE

CX regiszter értéke alapján:

JCXZ (**J**ump if **CX** **Z**ero): akkor ugrik, ha CX értéke 0.

Csere: **XCHG** op1, op2

Működése: megcseréli op1 és op2 értékét

Inkrementálás: **INC** op

Működése: op ← op + 1

Dekrementálás: **DEC** op

Működése: op ← op - 1

Ciklusszervezés: **LOOP** ipr8

Működése:

1. CX ← CX - 1
 2. Ha CX != 0, akkor ugrás ipr8 IP-relatív értékkel
-