Assembly utasítások listája

Bevezetés:

Ebben a segédanyagban a fontosabb assembly utasításokat szedtem össze. Az utasítások csoportosítva vannak.

- A „fontos” kategóriába azok az utasítások tartoznak, amiknek az ismerete nélkül valószínűleg a ZH-kat sem lehet megoldani.
- Az egyéb kategóriába esnek azok az utasítások, amik szintén előkerülnek, és megkönnyíthetik a feladatok megoldását, de nem feltétlenül elengedhetetlenek.

FONTOS DOLGOK:

1. Az utasítok megtanulása még nem jelenti, hogy az ember használni is tudja őket. Minden utasítás működését érdemes átgondolni, és kipróbálni 1-2 példán.
2. A segédanyag még nem végleges, és nem hivatalos. Bővül, változik. Amíg így van, addig csak jelzés értékkel kezelendő. Ha ez a helyzet változik, akkor ezt a pontot ki fogom törölni. ☺
3. Hibákkal, megjegyzéseekel, észrevételekkel kapcsolatban senki ne fogja vissza magát. Nyugodtan lehet szólni. Alkalomadtán javítani is fogom a jegyzetet.
4. Illetve, ha valaki érez erőt és kedvet a jegyzet írásába való bekapcsolódásba, akkor bártan jelentkezzen. Sok a dolgom Nekem is így lassan haladok a segédanyagokkal.

Fontos utasítások

MOV

- MOV op1, op2 ; op1 <= op2
  ; op1, vagy op2 regiszter kell, hogy legyen
  ; a két operandus mérete megegyező kell, hogy legyen
- Kétoperandusú utasítás. Az első operandusába átrakja a második operandus által jelölt adatot. A két operandus közül legalább egy regiszter kell, hogy legyen.

ADD

- ADD op1, op2 ; op1 <= op1 + op2
  ; op1, vagy op2 regiszter kell, hogy legyen
  ; a két operandus mérete megegyező kell, hogy legyen

SUB

- SUB op1, op2 ; op1 <= op1 - op2
  ; op1, vagy op2 regiszter kell, hogy legyen
  ; a két operandus mérete megegyező kell, hogy legyen

ADC

- ADC op1, op2 ; op1 <= op1 + op2 + carry
  ; op1, vagy op2 regiszter kell, hogy legyen
  ; a két operandus mérete megegyező kell, hogy legyen

SBB

- SBB op1, op2 ; op1 <= op1 - op2 - carry
  ; op1, vagy op2 regiszter kell, hogy legyen
  ; a két operandus mérete megegyező kell, hogy legyen

CBW

- CBW ; **Nincs operandusa**
  ; Az AL-ben tárolt számot terjeszti ki **ELŐJELESEN** AX -be

CWD

- CWD ; **Nincs operandusa**
  ; Az AX-ben tárolt számot terjeszti ki **ELŐJELESEN** DX:AX -be

MUL

- MUL op ; **1 operandusú, az operandus nem lehet közvetlen**
  ; előjel nélküli számok szorzására használatos
  ; Ha op 8 bites, akkor:
  ; AX <= AL * op
  ; Ha op 16 bites, akkor:
  ; (DX:AX) <=AX * op

IMUL

- IMUL op ; Ugyanaz, mint a MUL, csak előjelesen kezeli a számokat.
  ; FONTOS: az előjeles és előjel nélküli számok kezelése meg
  ; van különböztetve

DIV

- DIV op ; **1 operandusú, az operandus nem lehet közvetlen**
  ; előjel nélküli számok osztására használatos
  ; Ha op 8 bites, akkor:
  ; AL <= AX / op (AL-be a hányados)
  ; AH <= AX % op (AH-ba a maradék)
  ; Ha op 16 bites, akkor:
  ; AX <= (DX:AX) / op (AX-be a hányados)
  ; DX <= (DX:AX) % op (DX-ba a maradék)

IDIV

- IDIV op ; Ugyanaz, mint a DIV, csak előjelesen kezeli a számokat.
  ; FONTOS: az előjeles és előjel nélküli számok kezelése meg
  ; van különböztetve

INC

- INC op ; op <= op + 1
  ; csak növeli egyel az operandusa értékét

DEC

- DEC op ; op <= op - 1
  ; csak csökkenti eggyel az operandusa értékét

NEG

- NEG op ; op <= -op
  ; negálja az operandus értékét

XCHG

- XCHG op1 op2 ; op1 = op2 , op2 = op1
  ; megcseréli a két operandus értékét
  ; legalább az egyik operandusa regiszter
  ; a két operandus mérete megegyező kell, hogy legyen

PUSH

- PUSH op ; op 16 bites (word)
  ; csökkenti SP értékét kettővel
  ; beteszi a 16 bites regisztert (op) a verem tetejére

POP

- POP op ; op 16 bites (word)
  ; növeli SP értékét kettővel
  ; a verem tetején lévő értéket op-ba tölti

CMP

- CMP op1, op2 ; két operandus, legalább az egyik regiszter
  ; a két operandus mérete megegyező kell, hogy legyen
  ; a flags regisztert állítja be „op1-op2” szerint
  ; gyakorlatilag a két operandusának a relációját számítja ki
  ; utána feltételes ugrásokat lehet végezni, ha
  ; (op1 > op2), (op1=op2), stb.
  ; fontos, hogy csak addig van hatása amíg egy másik utasítás
  ; meg nem változtatja a FLAGS tartalmát

Feltételes ugró utasítások: (Általában – de nem feltétlenül – CMP utasítás után állnak.)

- Formájuk általában:
  - JE címke ; ugrik a címkére, ha egy feltétel teljesül
    ; ebben az esetben, ha az előző aritmetikai művelet
    ; eredménye 0
- CMP után, számok összehasonlításakor (egy ugrásra több utasítás is lehet):

Előjeles számoknál	Ugrik ha …	Előjel nélküli számoknál
JE, JZ	=	JE, JZ
JNE, JNZ	≠	JNE, JNZ
JG, JNLE	>	JA, JNBE
JGE, JNL	≥	JAE, JNB, JNC
JL, JNGE	<	JB, JNAE, JC
JLE, JNG	≤	JBE, JNA

- A FLAGS regiszter bitjei alapján ugró utasítások:

Ugrik, ha a flag értéke 1	Vizsgált flag	Ugrik, ha a flag értéke 0
JZ		JNZ
JC		JNC
JS		JNS
JO		JNO
JP		JNP

JCXZ

- JCXZ címke ; ugró utasítás. A vezérlés ugrik a címkére, ha CX=0

LOOP

- LOOP címke ; ciklusszervező utasítás
  ; a CX értékét csökkenti egyel,
  ; és ugrik a címkére, ha a CX elérte a 0-t
  ; A CX-et számlálóként használva a címke és a LOOP közötti
  ; kódrész le lehet futtatni adott számban.

JMP

- JMP címke ; ugró utasítás. A vezérlés feltétel nélkül ugrik a cimkére

CALL

- CALL címke ; a visszatérési címet a verem tetejére teszi
  ; ugrik a címkére

RET

- RET ; nincs operandusa
  ; kiveszi a verem tetejéről a visszatérési címet és odaugrik

MOVSB

- MOVSB ;sztring mozgatása, byte
  ;átvitel az (ES:DI) által mutatott cimre
  ; a (DS:SI) által mutatott címrõl

MOVSW

- MOVSW ;sztring mozgatása, word
  ;átvitel az (ES:DI) által mutatott cimre
  ; a (DS:SI) által mutatott címrõl

LODSB

- LODSB ; nincs operandusa (8 bit)
  ; a DS:[SI] által mutatott helyről betölti a string/számsorozat
  ; következő elemét az AL-be, és a D flag alapján lépteti SI-t
  ; (ha D=0, akkor +1 ha D=1, akkor -1)

LODSW

- LODSW ; nincs operandusa (16 bit)
  ; a DS:[SI] által mutatott helyről betölti a string/számsorozat
  ; következő elemét az AX-be, és a D flag alapján lépteti SI-t
  ; (+2 vagy -2)

STOSB

STOSW

CMPSB

CMPSW

SCASB

SCASW

AND

TEST

OR

XOR

NOT

RCR

RCL

ROR

ROL

SHR

SHL

SAR

SAL

További utasítások

PUSHF

POPF

XLAT

LDS

LES

LEA

LAHF

SAHF

AAA

DAA

AAS

DAS

AAM

Készítették:

- Antal Gábor