



# Kecskeméti Főiskola

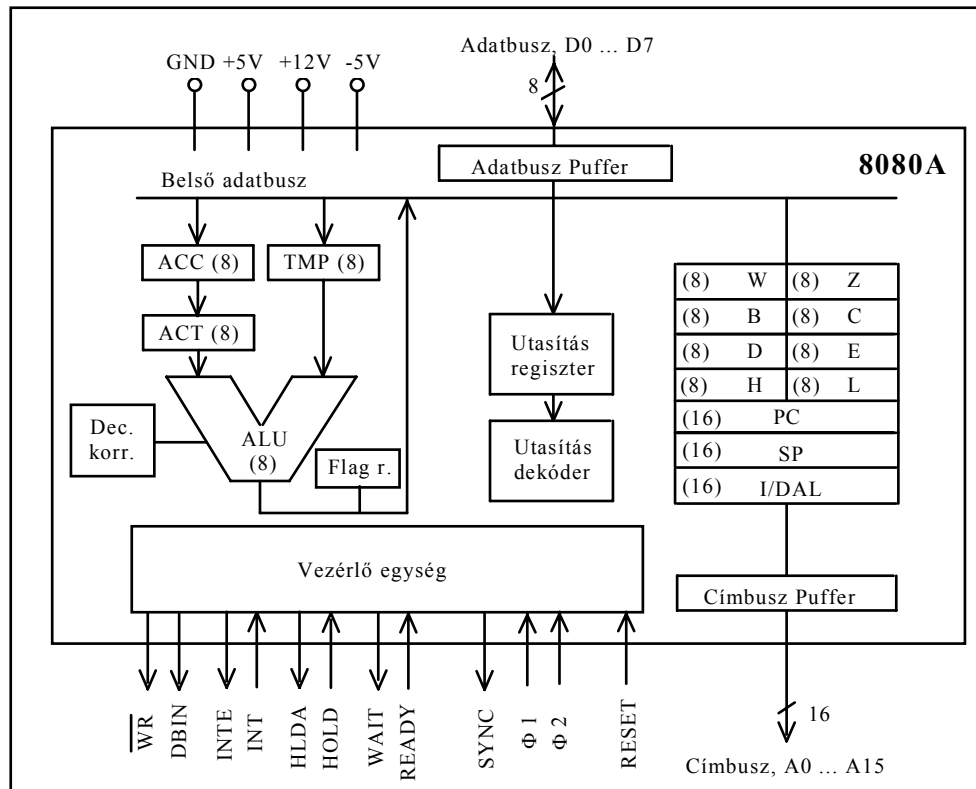
## Gépipari és Automatizálási

### Műszaki Főiskolai Kar



**DR. MADARÁSZ LÁSZLÓ**  
**TISZAVÖLGYI HENRIETTA**

## Az i8080 és az i8085 utasításai



Kecskemét  
2007

Dr. Madarász László – Tiszavölgyi Henrietta

Az i8080 és az i8085 utasításai

Oktatási segédlet

Kecskemét  
2007

## Bevezetés

Az Elektrotechnika és Kibernetika Tanszéken 2005-ben készült el Az i8080 utasításkészlet részletes ismertetése c. oktatási segédlet. Azóta az oktatási egységek átszervezése során a nevünk Elektrotechnika és Kibernetika Szakcsoportra változott. Közben az oktatási segédletet használva annak néhány sajtóhibája is kiderült, valamint egyes esetekben a magyarázatainak a túlzott tömörsége. Mindezekben igyekeztünk javítani, s ebben az átdolgozott, javított változatban törekvéseink szerint kevesebb hiba, viszont esetenként részletesebb magyarázat található. Alapvetően azonban a célkitűzéseink nem változtak meg, ezért megismételjük alább az eredeti Bevezetést is.

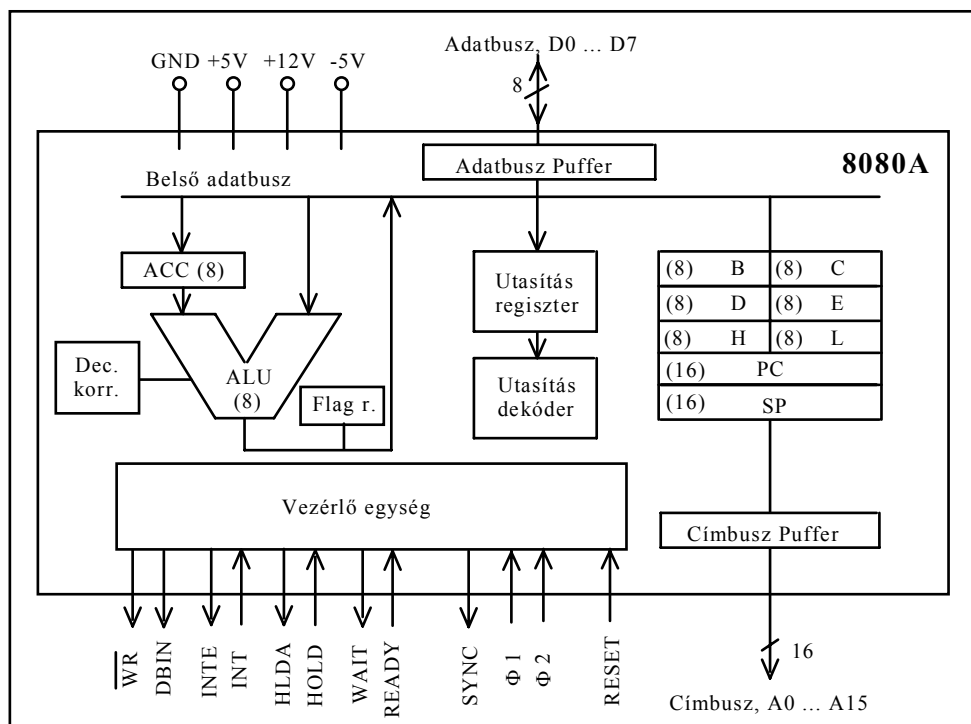
Kecskemét, 2007. december

A Szerzők

## Bevezetés „Az i8080 utasításkészlet részletes ismertetése” segédlethez

A mikroprocesszorokkal való ismerkedést egy jól áttekinthető, a legfontosabb elemeket és működéseket már tartalmazó hajdani mikroprocesszor, az Intel 8080 megismerésén keresztül kezdjük meg. Ennek a processzornak a hardver felépítését és az utasításkészletét az [1, 2] jegyzetből lehet megismerni. Az utasításkészletre vonatkozó általános elvek alapján azonban néhány utasítás működése nem érthető meg, másokban különleges megoldásokat alkalmazott az Intel. Az utasítások pontos megismerését kívánja segíteni ez az oktatási segédlet, amit elsősorban a témával foglalkozó oktatók számára készítettünk.

Az 1. ábrán látható az i8080 belső felépítése úgy, ahogyan a jegyzetekben is szerepel. Ez az ábra a programkészítő által elérhető részleteket mind tartalmazza. Az egyes belső elemek szerepét a jegyzetek ismertetik.



1. ábra.

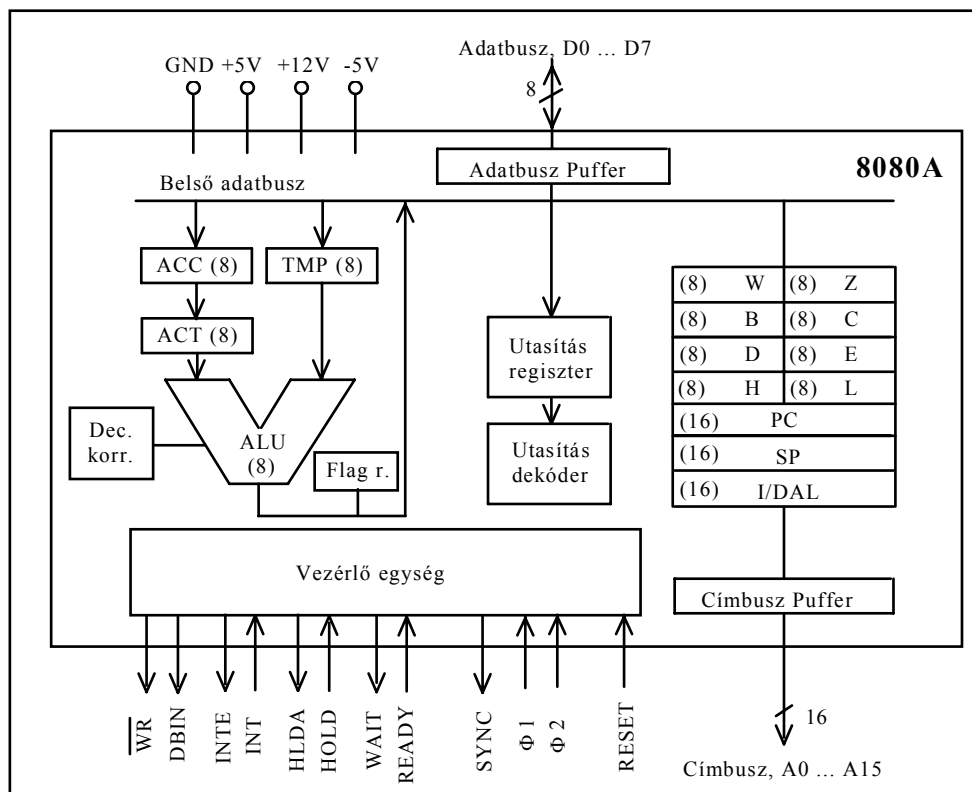
Az utasítások működésének megértéséhez azonban néhány további elemet is célszerű feltüntetni. Ezek a következők:

- ACT Accumulator temporary, akkumulátor latch, átmeneti akkumulátor,
- W Temporary register W, W átmeneti regiszter,
- Z Temporary register Z, Z átmeneti regiszter.
- TMP Temporary register, átmeneti regiszter

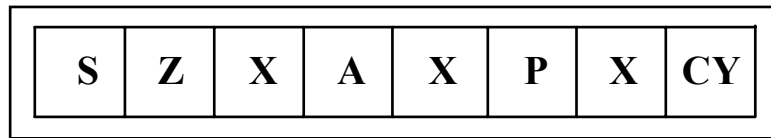
Az ACT, a W, a Z és a TMP mind 8 bites regiszterek. További belső egység a címek kialakításában közreműködő 16 bites regiszter:

I/DAL, Incrementer/decrementer address latch, inkrementáló/dekrementáló címtároló.

Ezek a regiszterek csak az utasítások végrehajtása közben játszanak szerepet, a felhasználó a programmal, az utasításokkal közvetlenül sem írni, sem olvasni nem képes ezeket. Ugyanakkor az utasítások működésének pontos megértéséhez szükséges ezek viselkedését is követni. A 2. ábrán megfigyelhető a most ismertetett kiegészítő regiszterek elhelyezkedése is.



Az utasítások egy része változtatja a flagbiteket, másoknál a működés feltétele egy-egy flag előírt értéke. A flagregisztert a 3. ábra mutatja be.



3. ábra

Az opkódban az egyes regiszterekre hárombites kódjukkal kell hivatkozni, ezeket az 1. táblázat mutatja be.

1. táblázat		Az önállóan hivatkozott regiszterek kódja az opkódon belül	
B: 000	C: 001	D: 010	E: 011
H: 100	L: 101	M: 110	A (ACC): 111

Ha az utasítás regiszterpárral dolgozik, a párosokat kétbites kóddal lehet kijelölni. A regiszterpárok kódértékeit a 2. táblázatban találjuk meg.

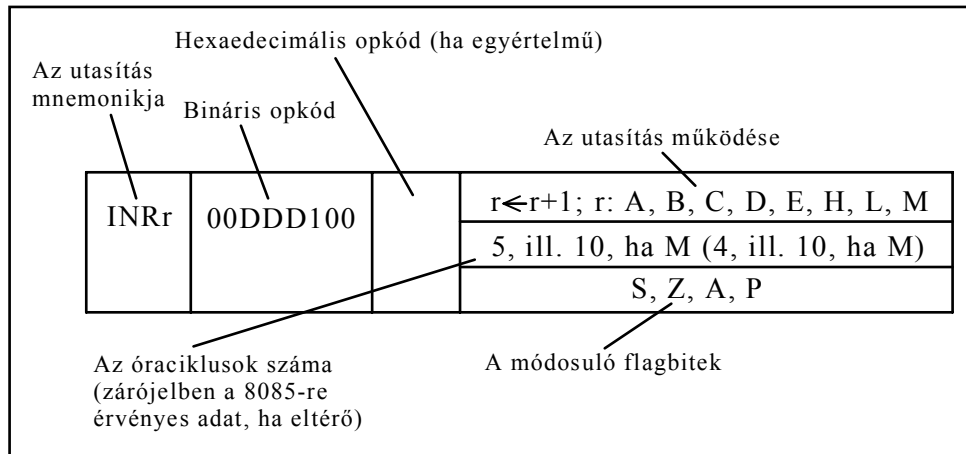
2. táblázat		A regiszterpárok kódja az opkódon belül	
B,C: 00	D,E: 01	H,L: 10	SP vagy PSW: 11

Az SP és a PSW kódja azonos, de az utasítás leírásából kiderül, hogy az adott utasítás melyikkel dolgozik. A PSW (Program Status Word, programállapot szó) az akkumulátort és a flagregisztert tartalmazza (A,F), a felső bájtja az A, az alsó az F.

A feltételes utasításokban a feltétel egy-egy flagbit meghatározott értéke. Az utasítás mnemonikájában ennek sajátos jele van (mnemonikjelölés), ami az utasítás nevének első betűje után áll (ugró utasítás esetén a J-t követi, pl. JNZ, szubrutinhívó utasításnál a C-t, pl. CPE, míg a visszatérő utasítás esetén az R-et, pl. RC). A feltételeket, azok kódját és jelentését a 3. táblázat mutatja be. A CCC-kód oszlopban lévő hárombites érték kerül az opkód bináris alakjába, a „mnemonikjelölés” az utasítás nevébe, a mnemonikba.

3. táblázat		A feltételek kódja az opkódon belül	
CCC-kód	Mnemonikjelölés	Értelmezés	Vizsgált flagérték
000	NZ	Nem zérus (Not Zero)	Z = 0
001	Z	Zérus (Zero)	Z = 1
010	NC	Nincs átvitel (Not Carry)	CY = 0
011	C	Átvitel (Carry)	CY = 1
100	PO	Páratlan paritás (Parity Odd)	P = 0
101	PE	Páros paritás (Parity Even)	P = 1
110	P	Pozitív (Plus)	S = 0
111	M	Negatív (Minus)	S = 1

A [2]-ben az utasításokat egységes formában egy táblázat mutatja be, a táblázat szerkezetét itt is bemutattuk (4. ábra).



4. ábra

Elöl áll a mnemonik, amit az assemblerek is használnak, mellette az opkód kettes számrendszerben. Ha az utasítás több bájtos, a második bájtra (B2) és a harmadikra (B3) csak utal a táblázat, hiszen azok tartalma változó. A harmadik rekeszben csak akkor szerepel hexadecimálisan az opkód, ha egyértelmű, azaz nem tartalmaz eseti beállítási biteket. Az utasítás működését leíró rekeszben, ha szükséges, felsoroljuk azokat a rekeszeket, melyekre az utasítás vonatkozhat. Alatta az utasítás végrehajtásának időigénye látható, óraciklusok számával megadva (zárójelben a 8085 időigénye, ha az eltér a 8080-tól). Legalul azt mutatja meg a táblázat, hogy az utasítás végrehajtása során melyik flagbit értéke változik.

A legtöbb utasítás esetében ez a táblázat, illetve a benne szereplő adatok elegendőek az utasítás megértéséhez, néhány esetben ki kell egészíteni az ott szereplő adatokat. Egyes utasításoknál pedig a végrehajtás, a gépi ciklusok ütemezése, a buszok viselkedése olyan különleges elemeket is tartalmaz, amit a táblázat nem tud megmutatni. Tulajdonképpen ezért készült el ez a segédlet. Bízunk benne, hogy használatával az i8080 utasításainak elemzése, a programműködések követése világosabb, érthetőbb lesz.

Az utasításokat a Függelékben egy másik táblázattal is bemutatjuk, ami az időbeli lefolyást is tartalmazza, óraciklusonként megadva az i8080 processzor működését.

Ezúton szeretném megköszönni oktató munkatársaimnak, hogy észrevételeikkel, kérdéseikkel segítettek ennek a segédletnek az összeállítását. Külön megköszönöm Tiszavölgyi Henrietta tanszéki demonstrátor munkáját, aki a Függelék táblázatának elkészítését vállalta magára.

Kecskemét, 2005. december

Dr. Madarász László

## Az utasítások

Ez a segédlet a Digitális technika II tantárgy oktatásához készült, ezért a hardver működésének megértését kívánja segíteni. Az utasításokkal kapcsolatban a tantárgyi követelmény az, hogy a CPU és a buszok viselkedésén keresztül bármelyik utasítás végrehajtását tudni kell követni, bemutatni. A belső működések, a kezelt adatok alakulása ezért ebben a segédletben sem téma, csak röviden említjük; a mikroprocesszor működésére, a gépi ciklusokra vonatkozóan tartalmaz kiegészítő ismereteket ez a fejezet.

Az utasításokat a [2]-ben szereplő táblázatos formában adjuk meg, majd kiegészítő megjegyzéseket fűzünk hozzájuk. A [2]-ben részletesen tárgyalt általános tudnivalókat itt nem ismételjük meg. A feldolgozás javasolt menete:

- a [2]-ből az utasításkészletre vonatkozó fejezet áttanulmányozása,
- azt követően a jelen segédlet használata.

A mnemonikákat egy általános assembly formátumban adjuk meg, egy konkrét assembler esetében kisebb eltérések lehetnek, pl. a számrendszerek megjelenítésében. Fontos a 16 bites értékeknél a bájt sorrend! Az i8080 esetében a memóriában (és így a memóriában tárolt utasításokon belül is) mindig a 16 bites adat alsó bájtja áll a kisebb című rekeszben, a felső bájt a magasabb címűben. Az assembly jellegű leírásban megjelenő 16 bites adatot azonban a szokásos módon írjuk. Az utasítások végrehajtásához szükséges gépi ciklusok számát az utasítás mérete és feladata határozza meg. Az opkódot egy M1 gépi ciklus olvassa be, az utasítás esetleges további elemeit további memóriaolvasási gépi ciklusok, ezután a végrehajtáshoz még annyi gépi ciklus szükséges, ahány bájtot még fogad vagy kiad a mikroprocesszor az adatbuszon. Van azonban kivétel is. A DAD utasítás végrehajtásában különleges gépi ciklusok jelennek meg, melyek során csak belső működéseket hajt végre a mikroprocesszor. Más esetben a végrehajtás még a következő utasítás opkódjának beolvasása (M1 gépi ciklus) alatt is folytatódik.

Az utasítások magyarázatában a hexadecimális számrendszert használjuk. Az opkód bináris alakjának megadásakor természetesen bináris érték szerepel, ezt nem jelöljük külön. A regiszterre vagy regiszterpárra hivatkozó utasítások az assembly programban a mnemonikban a regiszter, regiszterpár megjelölése előtt egyes esetekben előírják a szóközt (pl. INR C), más esetben tiltják (INRC), de az a leggyakoribb, hogy az assembler a szóközt nem veszi figyelembe és így mindkét alak használható. Ha az utasítás regiszterpárral dolgozik, a mnemoniknál a regiszterpár szerepel (pl. DAD BC), de egyes assemblerek megelégszenek ilyen esetben a regiszterpár első elemének megjelölésével (DAD B).

### Adatmozgató utasítások

<b>MOV</b> r1,r2	01DDSSSS		<b>r1-r2</b> ; r1,r2: A, B, C, D, E, H, L, M
			5, ill. 7, ha M (4, ill. 7, ha M)

Az utasítás egy regisztertartalmat (forrás, SSS) átmásol egy másik regiszterbe (cél, DDD), a forrás tartalma nem változik. A cél és a forrás lehetséges esetei a felsoroltak, a mnemonikban a felsorolásnak megfelelő jellel szerepelhetnek (pl. MOV H,A, ha az A tartalma kerül be a H-ba). Az M a mnemonikban azt jelenti, hogy az adott regiszter (cél vagy forrás) a H,L pillanatnyi tartalmával címzett külső memóriarekesz. Minden kombináció lehetséges, kivéve a MOV M,M-et, ilyen utasítás nincs (az így kialakuló opkód a HLT utasítás opkódja!). Általában egyetlen M1 gépi ciklusból áll, kivéve, ha M szerepel. SSS=M esetén ugyanis az M1-et egy memóriaolvasás, DDD=M esetén egy memóriairás követi.

<b>MVI</b> r B2	00DDD110 B2		<b>r-B2</b> ; r: A, B, C, D, E, H, L, M
			7, ill. 10, ha M

Az utasítás a második bájtjában szereplő adatot bemásolja a megadott regiszterbe (cél, DDD). A cél lehetséges esetei a felsoroltak, a mnemonikban a felsorolásnak megfelelő jellel szerepel (pl. MVI C,A6

akkor, ha az A6h értéket kell beírni a C-be). Az M a mnemonikban azt jelenti, hogy az adott „regiszter” a H,L pillanatnyi tartalmával címzett külső memóriarekesz.

Általában két gépi ciklusból áll (M1 és memóriaolvasás), kivéve, ha M szerepel, akkor az előbbieket egy memóriairás követi.

<b>LXI</b>	rp	00rp0001		<b>rh←B3, r1←B2; rp: BC, DE, HL, SP</b>
B2		B2		10
B3		B3		

Az utasítás második és harmadik bájtyában egy 16 bites adat található. Ennek felső bájtya (B3) a kijelölt regiszterpár felső oldali elemébe (a regiszterpár nevében balra álló regiszterbe) kerül, a 16 bites adat alsó bájtya (B2) pedig a regiszterpár alsó oldali regiszterébe. A cél lehetséges esetei a felsoroltak, a mnemonikban a felsorolásnak megfelelő jellel szerepel a regiszterpár (pl. LXI DE, ABCD esetén az ABh értéket kell beírni a D-be, a CDh értéket az E-be; az utasításban a második bájty (B2) értéke CDh, a harmadik bájty (B3) értéke ABh).

A végrehajtás három gépi ciklusból áll (M1 és két memóriaolvasás).

<b>LDA</b>		00111010	3A	<b>A←(B3, B2)</b>
B2		B2	B2	13
B3		B3	B3	

Az utasításban (pl. LDA 3456) szereplő 16 bites címmel kijelölt külső memóriarekesz tartalma betöltődik az akkumulátorba. Az utasítás második bájtya (B2) a cím alsó bájtya, az utasítás harmadik bájtya (B3) a cím felső bájtya.

A végrehajtás négy gépi ciklusból áll, az M1-et és a két memóriaolvasást a végrehajtáshoz egy további memóriaolvasás követi.

<b>STA</b>		00110010	32	<b>(B3, B2)←A</b>
B2		B2	B2	13
B3		B3	B3	

Az utasításban (pl. STA 29BE) szereplő 16 bites címmel kijelölt külső memóriarekeszbe betöltődik az akkumulátor tartalma. Az utasítás második bájtya (B2) a cím alsó bájtya, az utasítás harmadik bájtya (B3) a cím felső bájtya.

A végrehajtás négy gépi ciklusból áll, az M1-et és a két memóriaolvasást a végrehajtáshoz egy memóriairás követi.

<b>LHLD</b>		00101010	2A	<b>L←(B3, B2), H←(B3, B2 + 1)</b>
B2		B2	B2	16
B3		B3	B3	

Az utasításban (pl. LHLD 2468) szereplő 16 bites címmel kijelölt külső memóriarekesz tartalma bemásolódik az L regiszterbe, az eggyel magasabb című memóriarekesz tartalma pedig a H-ba.

Öt gépi ciklusból áll, az M1-et és a két memóriaolvasást a végrehajtáshoz szükséges két további memóriaolvasás követi.

<b>SHLD</b>		00100010	22	<b>(B3, B2)←L, (B3, B2 + 1)←H</b>
B2		B2	B2	16
B3		B3	B3	

Az utasításban (pl. SHLD 3579) szereplő 16 bites címmel kijelölt külső memóriarekeszbe bemásolódik az L regiszter tartalma, az eggyel magasabb című memóriarekeszbe pedig a H tartalma.



Öt gépi ciklusból áll, az M1-et és a két memóriaolvasást a végrehajtáshoz szükséges két memóriairás követi.

<b>LDAX</b> rp	00rp1010		<b>A←(rp)</b> ; rp:BC, DE
			7

Az utasításban (pl. LDAX BC) szereplő regiszterpár tartalmával, mint címmel kijelölt külső memóriarekesz tartalma bemásolódik az A regiszterbe. A lehetséges regiszterpárokat a táblázat felsorolja.

Két gépi ciklusból áll, az M1-et a végrehajtáshoz szükséges memóriaolvasás követi.

<b>STAX</b> rp	00rp0010		<b>(rp)←A</b> ; rp: BC, DE
			7

Az utasításban (pl. STAX DE) szereplő regiszterpár tartalmával, mint címmel kijelölt külső memóriarekeszbe bemásolódik az A regiszter tartalma. A lehetséges regiszterpárokat a táblázat felsorolja.

Két gépi ciklusból áll, az M1-et a végrehajtáshoz szükséges memóriairás követi.

<b>XCHG</b>	11101011	EB	<b>H↔D, L↔E</b>
			4

Az XCHG utasítás a H és a D regiszterek tartalmát felcseréli, továbbá felcseréli egymással az L és az E regiszterek tartalmát is.

Egy gépi ciklusból áll, az M1 végén a belső működés megtörténik.

### Aritmetikai utasítások

<b>ADD</b> r	10000SSS		<b>A←A+r</b> ; r: A, B, C, D, E, H, L, M
			4, ill. 7, ha M
			S, Z, A, P, CY

Az akkumulátor és a kijelölt regiszter tartalmának összegét az akkumulátorba helyezi. A regiszter lehetséges esetei a felsoroltak, a mnemonikban a felsorolásnak megfelelő jel szerepel (pl. ADD C akkor, ha az A és a C összegét kell beírni az A-ba). Az M a mnemonikban azt jelenti, hogy az adott regiszter a H,L pillanatnyi tartalmával címzett külső memóriarekesz.

Az utasítás (ha nem M a mnemonikban megadott regiszter) egyetlen gépi ciklus (M1) alatt megvalósul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!). A külső memória tartalmát felhasználó ADD M esetén az M1-et memóriaolvasás követi, ezzel a gépi ciklusok sora lezárul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!).

<b>ADI</b> B2	11000110	C6	<b>A←A+B2</b>
	B2	B2	7
			S, Z, A, P, CY

Az ADI utasítás (pl. ADI F3) az akkumulátor tartalmának és a második bájtnban (B2) megadott értéknek (a példában F3h) az összegét az akkumulátorba helyezi.

Az utasítás két gépi ciklus (M1 és egy memóriaolvasás) alatt megvalósul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!).

<b>ADC</b> r	10001SSS		<b>A←A+r+CY</b> ; r: A, B, C, D, E, H, L, M
--------------	----------	--	---

			4, ill. 7, ha M
			S, Z, A, P, CY

Az akkumulátor tartalmának, a kijelölt regiszter tartalmának és a CY flagbitnek az összegét az akkumulátorba helyezi. A regiszter lehetséges esetei a felsoroltak, a mnemonikban a felsorolásnak megfelelő jel szerepel (pl. ADC B akkor, ha az A és a B regiszterek és a CY flagbit összegét kell beírni az A-ba). Az M a mnemonikban azt jelenti, hogy az adott regiszter a H,L pillanatnyi tartalmával címzett külső memóriarekesz.

Az utasítás (ha nem M a mnemonikban megadott regiszter) egyetlen gépi ciklus (M1) alatt megvalósul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!). A külső memória tartalmát felhasználó ADC M esetén az M1-et memórialvasás követi, ezzel a gépi ciklusok sora lezárul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!).

<b>ACI</b>	11001110	CE	<b>A-A+B2+CY</b>
B2	B2	B2	7
			S, Z, A, P, CY

Az ACI utasítás az akkumulátor tartalmának és a második bájtban (B2) megadott értéknek, valamint a CY flagbitnek az összegét az akkumulátorba helyezi.

Az utasítás két gépi ciklus (M1 és egy memórialvasás) alatt megvalósul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!).

<b>SUB</b> r	10010SSS		<b>A-A-r; r: A, B, C, D, E, H, L, M</b>
			4, ill. 7, ha M
			S, Z, A, P, CY

Az akkumulátor és a kijelölt regiszter tartalmának különbségét az akkumulátorba helyezi. A regiszter lehetséges esetei a felsoroltak, a mnemonikban a felsorolásnak megfelelő jel szerepel (pl. SUB D akkor, ha az A és a D különbségét kell beírni az A-ba). Az M a mnemonikban azt jelenti, hogy az adott regiszter a H,L pillanatnyi tartalmával címzett külső memóriarekesz.

Az utasítás (ha nem M a mnemonikban megadott regiszter) egyetlen gépi ciklus (M1) alatt megvalósul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!). A külső memória tartalmát felhasználó SUB M esetén az M1-et memórialvasás követi, ezzel a gépi ciklusok sora lezárul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!).

<b>SUI</b>	11010110	D6	<b>A-A-B2</b>
B2	B2	B2	7
			S, Z, A, P, CY

Az SUI utasítás (pl. SUI D2) az akkumulátor tartalmának és a második bájtban (B2) megadott értéknek (a példában D2h) a különbségét az akkumulátorba helyezi.

Az utasítás két gépi ciklus (M1 és egy memórialvasás) alatt megvalósul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!).

<b>SBB</b> r	10011SSS		<b>A-A-r-CY</b> ; r: A, B, C, D, E, H, L, M
			4, ill. 7, ha M
			S, Z, A, P, CY

Az akkumulátor tartalmából kivonja a kijelölt regiszter tartalmát és a CY flagbit értékét. A regiszter lehetséges esetei a felsoroltak, a mnemonikban a felsorolásnak megfelelő jel szerepel (pl. SBB H akkor, ha az A-ból a H és a CY értékét kell kivonni, s az eredményt kell beírni az A-ba). Az M a mnemonikban azt jelenti, hogy az adott regiszter a H,L pillanatnyi tartalmával címzett külső memóriarekesz.

Az utasítás (ha nem M a mnemonikban megadott regiszter) egyetlen gépi ciklus (M1) alatt megvalósul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!). A külső memória tartalmát felhasználó SUB M esetén az M1-et memóriaolvásás követi, ezzel a gépi ciklusok sora lezárul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!).

<b>SBI</b> B2	11011110 B2	DE B2	<b>A-A-B2-CY</b>
			7
			S, Z, A, P, CY

Az SBI utasítás (pl. SBI 79) az akkumulátor tartalmából kivonja a második bájtban (B2) megadott értéket (a példában 79h), továbbá a CY értékét, majd az eredményt az akkumulátorba helyezi.

Az utasítás két gépi ciklus (M1 és egy memóriaolvásás) alatt megvalósul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!).

<b>INR</b> r	00DDD100		<b>r-r+1</b> ; r: A, B, C, D, E, H, L, M
			5, ill. 10, ha M (4, ill. 10, ha M)
			S, Z, A, P

A regiszter tartalmát megnöveli 1-gyel (inkrementálja). A regiszter lehetséges esetei a felsoroltak, a mnemonikban a felsorolásnak megfelelő jel szerepel (pl. INR B akkor, ha a B tartalmát kell növelni). Az M a mnemonikban azt jelenti, hogy az adott regiszter a H,L pillanatnyi tartalmával címzett külső memóriarekesz.

Az utasítás (ha nem M a mnemonikban megadott regiszter) egyetlen gépi ciklus (M1) alatt megvalósul. A külső memória tartalmát megnövelő INR M esetén az M1-et memóriaolvásás követi, majd a megnövelt értéket egy memóriairás ciklusban helyezi vissza a H,L által címzett rekeszbe.

<b>DCR</b> r	00DDD101		<b>r-r-1</b> ; r: A, B, C, D, E, H, L, M
			5, ill. 10, ha M (4, ill. 10, ha M)
			S, Z, A, P

A regiszter tartalmát csökkenti 1-gyel (dekrementálja). A regiszter lehetséges esetei a felsoroltak, a mnemonikban a felsorolásnak megfelelő jel szerepel (pl. DCR L akkor, ha az L tartalmát kell csökkenteni). Az M a mnemonikban azt jelenti, hogy az adott regiszter a H,L pillanatnyi tartalmával címzett külső memóriarekesz.

Az utasítás (ha nem M a mnemonikban megadott regiszter) egyetlen gépi ciklus (M1) alatt megvalósul. A külső memória tartalmát csökkentő INR M esetén az M1-et memóriaolvásás követi, majd a csökkentett értéket egy memóriairás ciklusban helyezi vissza a H,L által címzett rekeszbe.

<b>INX</b> rp	00rp0011		<b>rp←rp+1; rp: BC, DE, HL, SP</b>
			5 (6)

A regiszterpár tartalmát megnöveli 1-gyel (inkrementálja). A regiszterpár lehetséges esetei a felsoroltak, a mnemonikban a felsorolásnak megfelelő jel szerepel (pl. INX BC akkor, ha a BC 16 bites teljes tartalmát kell növelni).

Az utasítás egyetlen gépi ciklus (M1) alatt megvalósul.

<b>DCX</b> rp	00rp1011		<b>rp←rp-1; rp: BC, DE, HL, SP</b>
			5 (6)

A regiszterpár tartalmát csökkenti 1-gyel (dekrementálja). A regiszterpár lehetséges esetei a felsoroltak, a mnemonikban a felsorolásnak megfelelő jel szerepel (pl. DCX SP akkor, ha az SP 16 bites teljes tartalmát kell csökkenteni).

Az utasítás egyetlen gépi ciklus (M1) alatt megvalósul.

<b>DAD</b> rp	00rp1001		<b>HL←HL+rp; rp: BC, DE, HL, SP</b>
			10
			CY

A HL regiszterpár 16 bites teljes tartalmának és a kijelölt regiszterpár tartalmának összegét a HL regiszterpárba helyezi. A regiszterpár lehetséges esetei a felsoroltak, a mnemonikban a felsorolásnak megfelelő jel szerepel (pl. DAD BC akkor, ha a HL és a BC összegét kell beírni a HL-be).

Az utasítás három gépi ciklus (M1 és két speciális gépi ciklus) alatt valósul meg (ld. a Függelék táblázatát!). A második és harmadik gépi ciklus 3-3 órajelből áll, és nem keletkezik közben SYNC jel, csak belső működések zajlanak. A címbuszon a második gépi ciklus során ugyanaz a cím áll, mint az elsőben, a harmadik gépi ciklusban a címbusz lebeg. Az adatbusz a második és a harmadik gépi ciklusban is lebeg.

<b>DAA</b>	00100111	27	<b>A←A<sub>BCD</sub>, BCD összeadás után</b>
			4
			S, Z, A, P, CY

A DAA utasítás a megelőző összeadás eredményét alakítja át. Akkor alkalmazzák, ha a program BCD számokkal dolgozik. Mivel az ALU kettes számrendszerben ad össze, az összeadás végrehajtása után egy hamis eredmény keletkezik, amit azonban át lehet alakítani a helyes BCD összegre. Ezt az átalakítást végzi el a DAA.

Egyetlen M1 gépi ciklust igényel.

### Logikai utasítások

<b>ANA</b> r	10100SSS		<b>A←A AND r; r: A, B, C, D, E, H, L, M</b>
			4, ill. 7, ha M
			S, Z, A, P, CY=0

Az akkumulátor és a kijelölt regiszter tartalmának ÉS kapcsolatát (azaz a két tartalom bitenkénti ÉS kapcsolataként kialakuló nyolcbites eredményt) az akkumulátorba helyezi. A regiszter lehetséges esetei a felsoroltak, a mnemonikban a felsorolásnak megfelelő jel szerepel (pl. ANA D akkor, ha az A és a D bitenkénti ÉS kapcsolatát kell beírni az A-ba). Az M a mnemonikban azt jelenti, hogy az adott regiszter a

H,L pillanatnyi tartalmával címzett külső memóriarekesz. A CY flag az utasítás végrehajtása után mindig 0 értékű.

Az utasítás (ha nem M a mnemonikban megadott regiszter) egyetlen gépi ciklus (M1) alatt megvalósul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!). A külső memória tartalmát felhasználó ANA M esetén az M1-et memórialvasás követi, ezzel a gépi ciklusok sora lezárul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!).

<b>ANI</b>	11100110	E6	<b>A-A AND B2</b>
B2	B2	B2	7
			S, Z, A=0, P, CY=0

Az utasítás (pl. ANI 9B) az akkumulátor és a második bájtban (B2) megadott nyolc bitnek (példánkban 9B) az ÉS kapcsolatát (azaz a két érték bitenkénti ÉS kapcsolataként kialakuló nyolcbites eredményt) az akkumulátorba helyezi. A CY és az A flag az utasítás végrehajtása után mindig 0 értékű.

Az utasítás két gépi ciklus (M1 és egy memórialvasás) alatt megvalósul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!).

<b>XRA r</b>	10101SSS		<b>A-A EXOR r; r: A, B, C, D, E, H, L, M</b>
			4, ill. 7, ha M
			S, Z, A=0, P, CY=0

Az akkumulátor és a kijelölt regiszter tartalmának ANTIVALENCIA (kizáró VAGY, EXOR, XOR) kapcsolatát (azaz a két tartalom bitenkénti ANTIVALENCIA kapcsolataként kialakuló nyolcbites eredményt) az akkumulátorba helyezi. A regiszter lehetséges esetei a felsoroltak, a mnemonikban a felsorolásnak megfelelő jel szerepel (pl. XRA L akkor, ha az A és az L bitenkénti ANTIVALENCIA kapcsolatát kell beírni az A-ba). Az M a mnemonikban azt jelenti, hogy az adott regiszter a H,L pillanatnyi tartalmával címzett külső memóriarekesz. A CY és az A flag az utasítás végrehajtása után mindig 0 értékű.

Az utasítás (ha nem M a mnemonikban megadott regiszter) egyetlen gépi ciklus (M1) alatt megvalósul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!). A külső memória tartalmát felhasználó XRA M esetén az M1-et memórialvasás követi, ezzel a gépi ciklusok sora lezárul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!).

<b>XRI</b>	11101110	EE	<b>A-A EXOR B2</b>
B2	B2	B2	7
			S, Z, A=0, P, CY=0

Az utasítás (pl. XRI D5) az akkumulátor és a második bájtban (B2) megadott nyolc bitnek (példánkban D5) az ANTIVALENCIA (kizáró VAGY, EXOR, XOR) kapcsolatát (azaz a két érték bitenkénti ANTIVALENCIA kapcsolataként kialakuló nyolcbites eredményt) az akkumulátorba helyezi. A CY és az A flag az utasítás végrehajtása után mindig 0 értékű.

Az utasítás két gépi ciklus (M1 és egy memórialvasás) alatt megvalósul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!).

<b>ORA r</b>	10110SSS		<b>A-A OR r; r: A, B, C, D, E, H, L, M</b>
			4, ill. 7, ha M
			S, Z, A=0, P, CY=0

Az akkumulátor és a kijelölt regiszter tartalmának VAGY kapcsolatát (azaz a két tartalom bitenkénti VAGY kapcsolataként kialakuló nyolcbites eredményt) az akkumulátorba helyezi. A regiszter lehetséges

esetei a felsoroltak, a mnemonikban a felsorolásnak megfelelő jel szerepel (pl. ORA H akkor, ha az A és a H bitenkénti VAGY kapcsolatát kell beírni az A-ba). Az M a mnemonikban azt jelenti, hogy az adott regiszter a H,L pillanatnyi tartalmával címzett külső memóriarekesz. A CY és az A flag az utasítás végrehajtása után mindig 0 értékű.

Az utasítás (ha nem M a mnemonikban megadott regiszter) egyetlen gépi ciklus (M1) alatt megvalósul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!). A külső memória tartalmát felhasználó ORA M esetén az M1-et memóriaolvásás követi, ezzel a gépi ciklusok sora lezárul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!).

<b>ORI</b>	11110110	F6	<b>A-A OR B2</b>
B2	B2	B2	7
			S, Z, A=0, P, CY=0

Az utasítás (pl. ORI F2) az akkumulátor és a második bájtban (B2) megadott nyolc bitnek (példánkban F2h) a VAGY kapcsolatát (azaz a két érték bitenkénti VAGY kapcsolataként kialakuló nyolcbites eredményt) az akkumulátorba helyezi. A CY és az A flag az utasítás végrehajtása után mindig 0 értékű.

Az utasítás két gépi ciklus (M1 és egy memóriaolvásás) alatt megvalósul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!).

<b>CMP r</b>	10111SSS		<b>A-r alapján a flagbitek beállnak</b>
			r: A, B, C, D, E, H, L, M
			4, ill. 7, ha M
			S, Z, A, P, CY

Az akkumulátor tartalmából a kijelölt regiszter tartalmát kivonja, a különbségnek megfelelően beállítja a flagregiszter bitjeit, de az akkumulátor eredeti tartalma nem változik meg! A regiszter lehetséges esetei a felsoroltak, a mnemonikban a felsorolásnak megfelelő jel szerepel (pl. CMP D akkor, ha az A és a D különbsége alapján kell beállítani a flageket). Az M a mnemonikban azt jelenti, hogy az adott regiszter a H,L pillanatnyi tartalmával címzett külső memóriarekesz.

Az utasítás (ha nem M a mnemonikban megadott regiszter) egyetlen gépi ciklus (M1) alatt megvalósul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!). A külső memória tartalmát felhasználó CMP M esetén az M1-et memóriaolvásás követi, ezzel a gépi ciklusok sora lezárul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!).

<b>CPI</b>	11111110	FE	<b>A-B2 alapján a Flagbitek beállnak</b>
B2	B2	B2	7
			S, Z, A, P, CY

Az CPI utasítás (pl. CPI D9) az akkumulátor tartalmának és a második bájtban (B2) megadott értéknek (a példában D9h) a különbségét képezi, a különbségnek megfelelően beállítja a flagregiszter bitjeit, de az akkumulátor eredeti tartalma nem változik meg!

Az utasítás két gépi ciklus (M1 és egy memóriaolvásás) alatt megvalósul, bár a belső működések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!).

<b>RLC</b>	000001111	07	<b>CY←A7, Ai+1←Ai, A0←A7</b>
			4
			CY

Az akkumulátor (A) tartalmának körléptetése balra, egy bittel. Az A7 bit az A0 pozícióra kerül, de egyúttal beíródik a CY flagbitre is. Más flagbitek nem változnak.

Az utasítás egyetlen (M1) gépi ciklus alatt végrehajtódik, bár a belső műkődések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!).

<b>RRC</b>	000011111	0F	<b>A7←A0, Ai-1←Ai, CY←A0</b>
			4
			CY

Az akkumulátor (A) tartalmának körléptetése jobbra, egy bittel. Az A0 bit az A7 pozícióra kerül, de egyúttal beíródik a CY flagbitre is. Más flagbitek nem változnak.

Az utasítás egyetlen (M1) gépi ciklus alatt végrehajtódik, bár a belső műkődések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!).

<b>RAL</b>	000101111	17	<b>Ai+1←Ai, A0←CY, CY←A7</b>
			4
			CY

Az akkumulátor (A) tartalmának és kilencedik bitként a CY flagbitnek a körléptetése balra, egy bittel. Az A7 bit a CY-ba kerül, CY az A0 pozícióra. Más flagbitek nem változnak.

Az utasítás egyetlen (M1) gépi ciklus alatt végrehajtódik, bár a belső műkődések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!).

<b>RAR</b>	000111111	1F	<b>Ai-1←Ai, CY←A0, A7←CY</b>
			4
			CY

Az akkumulátor (A) tartalmának és kilencedik bitként a CY flagbitnek a körléptetése jobbra, egy bittel. Az A0 bit a CY-ba kerül, CY az A7 pozícióra. Más flagbitek nem változnak.

Az utasítás egyetlen (M1) gépi ciklus alatt végrehajtódik, bár a belső műkődések tovább tartanak, a következő utasítás opkódját beolvasó M1 alatt is (ld. a Függelék táblázatát!).

<b>CMA</b>	001011111	2F	<b>A←<math>\overline{A}</math></b>
			4

Az akkumulátor (A) tartalmának bitenkénti negálása. A flagbitek nem állnak be a megfelelő új értékekre!

Az utasítás egyetlen (M1) gépi ciklus alatt megvalósul.

<b>CMC</b>	001111111	3F	<b>CY←<math>\overline{CY}</math></b>
			4
			CY

A CY flagbit értékének negálása. Más flagbitek nem változnak!

Az utasítás egyetlen (M1) gépi ciklus alatt megvalósul.

<b>STC</b>	00110111	37	<b>CY←1</b>
			4
			CY

A CY flagbit értékének beállítása 1 értékre. Más flagbitek nem változnak!

Az utasítás egyetlen (M1) gépi ciklus alatt megvalósul.

### Vezérlésátadó utasítások

<b>JMP</b>	11000011	C3	<b>PC←B3, B2</b>
			10

Az utasítás (feltétlen ugrás, pl. JMP 6B7D, azaz ugrás a 6B7D címre) a PC tartalmát átírja az utasításban szereplő címre. A cím alsó bájta (most 7Dh) az utasítás második bájta (B2), a cím felső bájta (példánkban 6Bh) az utasítás harmadik bájta (B3). Az új címérték a WZ regiszterpárba kerül.

Az utasítás három ciklusos, az M1-et két memóriaolvasás gépi ciklus követi. A PC tartalmának felülírása a következő utasítás opkódját beolvasó M1 alatt történik meg, ezért a címbuszra kivételesen nem a PC, hanem a WZ tartalma kerül ki ekkor, majd a WZ + 1 érték töltődik be a PC-be (ld. a Függelék táblázatát!).

<b>J...</b>	11CCC010		<b>PC←B3, B2 ha igaz, egyébként folytat</b>
			10 (7/10)

Az utasítás (feltételes ugrás, pl. JNZ 9FC5, azaz JMP ha Z = 0 a 9FC5 címre) egy flagbit értékétől függően vagy átírja a PC tartalmát az utasításban szereplő címre, vagy belső működés nélkül befejeződik. A cím alsó bájta (most C5h) az utasítás második bájta (B2), a cím felső bájta (példánkban 9Fh) az utasítás harmadik bájta (B3). A feltétel az S, Z, P, CY flagbitek egyikének 0 vagy 1 értéke lehet, ezt a nyolcféle lehetőséget három bites kód jelzi az opkódon belül (CCC). A további részletek a 3. táblázatban találhatóak. Az új címérték a WZ regiszterpárba kerül.

Az utasítás három ciklusos, az M1-et két memóriaolvasás gépi ciklus követi, akár teljesül a feltétel és átíródik a PC, akár nem. Ha a feltétel nem teljesül, a következő utasítás opkódját beolvasó M1 alatt a PC tartalma kerül a címbuszra, és az előzőekben beolvasott új címérték nem kerül felhasználásra. Ha a feltétel teljesül, a PC tartalmának felülírása a következő utasítás opkódját beolvasó M1 alatt történik meg, ezért a címbuszra kivételesen nem a PC, hanem a WZ tartalma kerül ki ekkor, majd a WZ + 1 érték töltődik be a PC-be (ld. a Függelék táblázatát!).

<b>CALL</b>	11001101	CD	<b>(SP-1)←PCh, (SP-2)←PC1, SP←SP-2, PC←B3, B2</b>
			17 (18)

Az utasítás (feltétlen szubrutinra ugrás, szubrutinhívás, pl. CALL B7D2, azaz ugrás a B7D2 címen kezdődő szubrutinra) a PC tartalmát átírja az utasításban szereplő címre, de előzőleg a PC értékét elmenti a verembe. A cím alsó bájta (most D2h) az utasítás második bájta (B2), a cím felső bájta (példánkban B7h) az utasítás harmadik bájta (B3). Az új címérték a WZ regiszterpárba kerül. A PC felső bájta az SP-1 címre íródik be, a PC alsó bájta az SP-2 címre, és az SP-2 lesz az SP értéke az utasítás végrehajtása után.

Az utasítás öt ciklusos, az M1-et két memóriaolvasás gépi ciklus követi a címérték beolvasására, majd két veremírás gépi ciklus következik, a címbájtok elmentésére (előbb az SP-1 címre a PC felső bájta, majd az SP-2 címre a PC alsó bájta kerül kiírásra). A PC tartalmának felülírása a következő utasítás



opkódját beolvasó M1 alatt történik meg, ezért a címbuszra kivételesen nem a PC, hanem a WZ tartalma kerül ki ekkor, majd a WZ + 1 érték töltődik be a PC-be (ld. a Függelék táblázatát!).

<b>C...</b>	11CCC100		<b>CALL, ha igaz, egyébként folytat</b>
B2	B2		11/17 (9/18)
B3	B3		

Az utasítás (feltételes szubrutinra ugrás, feltételes szubrutinhívás, pl. CC DEF6, azaz ugrás a DEF6h címen kezdődő szubrutinra, ha ha CY=1) egy flagbit értékétől függően vagy átírja a PC tartalmát az utasításban szereplő címre, vagy belső működés nélkül befejeződik. Ha teljesül a feltétel, a PC átírása előtt annak értékét elmenti a verembe. A cím alsó bájttja (most F6h) az utasítás második bájttja (B2), a cím felső bájttja (példánkban DEh) az utasítás harmadik bájttja (B3). Az új címérték a WZ regiszterpárba kerül. A PC felső bájttja az SP-1 címre íródik be, a PC alsó bájttja az SP-2 címre, és az SP-2 lesz az SP értéke az utasítás végrehajtása után. A feltétel az S, Z, P, CY flagbitek egyikének 0 vagy 1 értéke lehet, ezt a nyolcféle lehetőséget három bites kód jelzi az opkódon belül (CCC). A további részletek a [2] jegyzetben találhatóak.

Az utasítás három ciklusos, ha a feltétel nem teljesül (M1, majd a címet beolvasó két memóriaolvasás gépi ciklus), a következő utasítás opkódját beolvasó M1 alatt a PC tartalma kerül a címbuszra, és az előzőekben beolvasott új címérték nem kerül felhasználásra. Ha a feltétel teljesül, az utasítás öt ciklusos, az M1-et és a két memóriaolvasás gépi ciklust két veremírás gépi ciklus követi a címbájtok elmentésére (előbb az SP-1 címre a PC felső bájttja, majd az SP-2 címre a PC alsó bájttja kerül kiírásra). A PC tartalmának felülírása a következő utasítás opkódját beolvasó M1 alatt történik meg, ezért a címbuszra kivételesen nem a PC, hanem a WZ tartalma kerül ki ekkor, majd a WZ + 1 érték töltődik be a PC-be (ld. a Függelék táblázatát!).

<b>RET</b>	11001001	C9	<b>PC<sub>1</sub>←(SP), PC<sub>h</sub>←(SP+1), Sp←SP+2</b>
			10

A RET utasítás (feltétlen visszatérés szubrutinról) a PC tartalmát átírja a veremből kiolvasható, oda utoljára betöltött címre. Az új címérték a WZ regiszterpárba kerül. A PC alsó bájttja az SP címről olvasódik ki, a PC felső bájttja az SP+1 címről, és az SP+2 lesz az SP értéke az utasítás végrehajtása után.

Az utasítás három ciklusos, az M1-et két veremolvasás gépi ciklus követi a címérték visszaolvasására, (előbb az SP címről a PC alsó bájttja, majd az SP+1 címről a PC felső bájttja kerül beolvasásra). A PC tartalmának felülírása a következő utasítás opkódját beolvasó M1 alatt történik meg, ezért a címbuszra kivételesen nem a PC, hanem a WZ tartalma kerül ki ekkor, majd a WZ + 1 érték töltődik be a PC-be (ld. a Függelék táblázatát!).

<b>R...</b>	11CCC000		<b>RETURN, ha igaz, egyébként folytat</b>
			5/11 (6/12)

Feltételes szubrutinból való visszatérés utasítás (pl. RZ, RET ha Z = 1) egy flagbit értékétől függően vagy átírja a PC tartalmát a veremből kiolvasható, oda utoljára betöltött címre, vagy belső működés nélkül befejeződik. Az új címérték a WZ regiszterpárba kerül. A PC alsó bájttja az SP címről olvasódik ki, a PC felső bájttja az SP+1 címről, és az SP+2 lesz az SP értéke az utasítás végrehajtása után. A feltétel az S, Z, P, CY flagbitek egyikének 0 vagy 1 értéke lehet, ezt a nyolcféle lehetőséget három bites kód jelzi az opkódon belül (CCC). A további részletek a [2] jegyzetben találhatóak.

Az utasítás egy ciklusos, ha a feltétel nem teljesül (M1), a következő utasítás opkódját beolvasó M1 alatt a PC tartalma kerül a címbuszra. Ha a feltétel teljesül, az utasítás három ciklusos, az M1-et két veremolvasás gépi ciklus követi a címérték visszaolvasására, (előbb az SP címről a PC alsó bájta, majd az SP+1 címről a PC felső bájta kerül beolvasásra). A PC tartalmának felülírása a következő utasítás opkódját beolvasó M1 alatt történik meg, ezért a címbuszra kivételesen nem a PC, hanem a WZ tartalma kerül ki ekkor, majd a WZ + 1 érték töltődik be a PC-be (ld. a Függelék táblázatát!).

<b>RST n</b>	11NNN111		$(SP-1) \leftarrow PCh, (SP-2) \leftarrow PCl, SP \leftarrow SP-2, PC \leftarrow 8 \cdot n$
			11

Az utasítás egybájtos, feltétlen szubrutinra ugrás, szubrutinhívás (pl. RST 6), a PC tartalmát átírja az utasításban szereplő bitek által meghatározott címre, de előzőleg a PC értékét elmenti a verembe. A cím 0000 0000 00NN N000b értékű, példánkban – mivel  $6d = 110b = 0000 0000 0011 0000b$ , azaz 0030h). Az új címérték a WZ regiszterpárba kerül. A PC felső bájta az SP-1 címre íródik be, a PC alsó bájta az SP-2 címre, és az SP-2 lesz az SP értéke az utasítás végrehajtása után.

Az utasítás három ciklusos, az M1-et két veremírás gépi ciklus követi, a címbájtok elmentésére (előbb az SP-1 címre a PC felső bájta, majd az SP-2 címre a PC alsó bájta kerül kiírásra). A PC tartalmának felülírása a következő utasítás opkódját beolvasó M1 alatt történik meg, ezért a címbuszra kivételesen nem a PC, hanem a WZ tartalma kerül ki ekkor, majd a WZ + 1 érték töltődik be a PC-be (ld. a Függelék táblázatát!).

<b>PCHL</b>	11101001	E9	$PCh \leftarrow H, PCl \leftarrow L$
			5 (6)

A PCHL utasítás a HL regiszterpár tartalmát betölti a PC-be (ugrás a HL tartalmára, mint címre). Az utasítás egyetlen gépi ciklust igényel, az M1 alatt végre is hajtódik.

### Veremkezelő utasítások

<b>PUSH rp</b>	11rp0101		$(SP-1) \leftarrow rh, (SP-2) \leftarrow rl, SP \leftarrow SP-2;$ rp: BC, DE, HL, PSW
			11 (13), ill. 12, ha PSW

Az utasításban megjelölt regiszterpár tartalmát a verembe menti. A regiszterpár lehetséges esetei a felsoroltak, a mnemonikban a felsorolásnak megfelelő jel szerepel (pl. PUSH DE akkor, ha a DE regiszterpár 16 bites tartalmát kell elmenteni). A regiszterpár felső bájta (BC esetén a B, DE esetén a D, HL esetén a H, PSW esetén az A) az SP-1 címre íródik be, a regiszterpár alsó bájta (BC esetén a C, DE esetén az E, HL esetén az L, PSW esetén az F) az SP-2 címre, és az SP értéke az utasítás végrehajtása után ez az SP-2 lesz.

Az utasítás három gépi ciklust igényel, az M1 után a két veremírás gépi ciklus következik.

<b>POP</b> rp	11rp0001		<b>rl←(SP), rh←(SP+1), SP←SP+2;</b> rp: BC, DE, HL, PSW
			10
			Csak PSW esetén

Az utasításban megjelölt regiszterpár tartalmát a veremből kiolvasott értékekkel felülírja. A regiszterpár lehetséges esetei a felsoroltak, a mnemonikban a felsorolásnak megfelelő jel szerepel (pl. POP HL akkor, ha a HL regiszterpárba kell betölteni a veremből kiolvasott 16 bites értéket). A regiszterpár alsó bájta (BC esetén a C, DE esetén az E, HL esetén az L, PSW esetén az F) az SP címről kap értéket. A regiszterpár felső bájta (BC esetén a B, DE esetén a D, HL esetén a H, PSW esetén az A) az SP+1 címről kiolvasott értéket kapja. Az SP értéke az utasítás végrehajtása után SP+2 lesz. Az utasítás nem befolyásolja a flagbiteket, kivéve, ha a kijelölt regiszterpár a PSW.

Az utasítás három gépi ciklust igényel, az M1 után a két veremolvasás gépi ciklus következik.

<b>XTHL</b>	11100011	E3	<b>L ↔ (SP), H ↔ (SP+1)</b> végrehajtva SP eredeti értékű
			18 (16)

Az XTHL utasítás a HL regiszterpár tartalmát és a veremben legfelül lévő 16 bites értéket egymással felcseréli. Az L tartalma az SP értékkel címzett memóriarekesz tartalmával cserélődik fel, a H tartalma pedig az SP+1 címmel kijelölt rekesz tartalmával cserélődik ki. Az SP értéke az utasítás végrehajtása után az eredeti érték lesz, azaz az utasítás következtében nem változik.

Az utasítás öt gépi ciklust igényel, az M1 után a két veremolvasás gépi ciklus következik, majd két veremírás.

<b>SPHL</b>	11111001	F9	<b>SP←H, L</b>
			5 (6)

Az SPHL utasítás hatására a HL regiszterpár 16 bites teljes tartalma beíródik az SP regiszterbe, a HL tartalma nem változik meg.

Az utasítás egyetlen, M1 gépi ciklus alatt lezajlik.

### Perifériakezelő utasítások

<b>IN</b> n	11011011 B2	DB B2	<b>A-B2 című bemeneti PORT</b>
			10

Bemenetet olvasó utasítás, a bemenetet egy nyolcbites érték, bemeneti cím jelöli ki (n). Az IN 2A esetében a bemeneti egység címe 2Ah. A beolvasott érték az akkumulátorba kerül, a flagbitek nem változnak.

Három gépi ciklust igényel az utasítás. Az M1 után a címet egy memóriaoLVASÁS gépi ciklus olvassa be, majd a végrehajtáshoz egy I/O olvasás gépi ciklus szükséges, ekkor a címbusz alsó és felső bájta is kikerül az n érték.

<b>OUT</b> <b>n</b>	11010011	D3	<b>B2 című kimeneti port-A</b>
	B2	B2	10

Kimenetre író utasítás, a kimenetet egy nyolcbites érték, kimeneti cím jelöli ki (n). Az OUT C7 esetében a kimeneti egység címe C7h. Az adatbuszon az akkumulátor tartalma jelenik meg a kimenet írásakor, így az akkumulátor tartalma léphet ki a kimeneten.

Három gépi ciklust igényel az utasítás. Az M1 után a címet egy memóriaolvasás gépi ciklus olvassa be, majd a végrehajtáshoz egy I/O írás gépi ciklus szükséges, ekkor a címbusz alsó és felső bájttára is kikerül az n érték, az adatbuszra az A tartalma.

### Megszakításkezelő utasítások

<b>EI</b>	11111011	FB	<b>Lásd: később</b>
			4

Megszakítást engedélyező utasítás, végrehajtása után a következő utasítás végén még nem fogad a mikroprocesszor megszakítást, csak az azutáni utasítás végén (késleltetett engedélyezés).

Egyetlen (M1) gépi ciklus szükséges az utasítás végrehajtásához.

<b>DI</b>	11110011	F3	<b>Lásd: később</b>
			4

Megszakítást tiltó utasítás, végrehajtása után a következő utasítás végén már nem fogad el a mikroprocesszor megszakítást.

Egyetlen (M1) gépi ciklus szükséges az utasítás végrehajtásához.

### Egyéb utasítások

<b>HLT</b>	01110110	76	<b>Lásd: később</b>
			7 (5)

A HLT utasítás hatására a mikroprocesszor befejezi az utasítások beolvasását, végrehajtását.

Az utasítás végrehajtásához egyetlen (M1) gépi ciklus szükséges, a következő speciális gépi ciklusban a PC tartalma kilép a címbuszra, az adatbuszon megjelenik a HALT státusz-szó, ezután a címbusz és a vezérlőbusz harmadik állapotba kerül.

<b>NOP</b>	00000000	00	<b>PC←PC+1</b>
			4

A NOP utasítás végrehajtása nem idéz elő semmilyen belső működést, csak a PC szokásos automatikus inkrementálódása történik meg.

Egyetlen (M1) gépi ciklus alatt megvalósul.

A 8080 mikroprocesszor utasításait a 22. oldalon összefoglaló táblázatból látszik, hogy 12 opkódot nem használtak ki a tervezők. Tulajdonképpen eredetileg ezeket a kódokat is értelmezte volna a 8080. Stanley Mazor, aki az utasítások részletes megvalósítási menetét dolgozta ki, a 256 opkódot nehézségi sorrendbe állította, s a legegyszerűbbel kezdte a munkát, onnan haladt az összetettebbek felé. Így ért el az XTHL utasításhoz, amelyik a legnehezebb opkód felől a 13. volt. Amikor ennek felvázolta a folyamatábráját és azt átadta a fejlesztést irányító mérnöknek, Masatoshi Shima-nak, ezzel a rájegyzéssel kapta vissza: „No more”. Így maradt szabadon az utolsó 12 opkód, s így vált lehetségessé, hogy a 8085, később a Z80 többletutasításokat használhasson!

### A 8085 többletutasításai

A 8085-nek mindössze két többlet utasítása van a 8080 utasításkészlethez képest. Ezeket a [2] részletesen leírja.

<b>SIM</b>	00110000	30	<b>Maszkregiszter-A</b>
			(4)

A SIM utasítás az akkumulátorból betöltheti a 8085 megszakítási maszkbiteket tároló regiszterét (ha A3=0), törölheti az RST 7.5 megszakításkérést, engedélyezheti a soros kimenetet és ekkor az A7 kilép az SOD kimeneti ponton.

Az utasítás végrehajtásához egyetlen (M1) gépi ciklus szükséges.

<b>RIM</b>	00100000	20	<b>A-Maszkregiszter</b>
			(4)

A RIM utasítás az akkumulátorba tölti a 8085 megszakítási maszkbitjeinek az értékét, az általános megszakítás-engedélyezés bitjét, a megszakításkérő bemeneteken esetlegesen várakozó megszakítások jelzéseit, valamint az A7-be helyezi a soros adatbemeneten (SID) lévő bitértéket.

Az utasítás végrehajtása egyetlen (M1) gépi ciklust igényel.

## A 8080, 8085 mikroprocesszorok utasításkészletének összefoglalása

00	NOP	-	36	MVI	M, bájt	6C	MOV	L,H	A2	ANA	D	D8	RC	-
01	LXI	B, dupla	37	STC	-	6D	MOV	L,L	A3	ANA	E	D9	-	-
02	STAX	B	38	-	-	6E	MOV	L,M	A4	ANA	H	DA	JC	Cím
03	INX	B	39	DAD	SP	6F	MOV	L,A	A5	ANA	L	DB	IN	Bájt
04	INR	B	3A	LDA	Cím	70	MOV	M,B	A6	ANA	M	DC	CC	Cím
05	DCR	B	3B	DCX	SP	71	MOV	M,C	A7	ANA	A	DD	-	-
06	MVI	B, bájt	3C	INR	A	72	MOV	M,D	A8	XRA	B	DE	SBI	Bájt
07	RLC	-	3D	DCR	A	73	MOV	M,E	A9	XRA	C	DF	RST	3
08	-	-	3E	MVI	A, bájt	74	MOV	M,H	AA	XRA	D	E0	RPO	-
09	DAD	B	3F	CMC	-	75	MOV	M,L	AB	XRA	E	E1	POP	H
0A	LDAX	B	40	MOV	B,B	76	HLT	M,M	AC	XRA	H	E2	JPO	Cím
0B	DCX	B	41	MOV	B,C	77	MOV	M,A	AD	XRA	L	E3	XTHL	-
0C	INR	C	42	MOV	B,D	78	MOV	A,B	AE	XRA	M	E4	CPO	Cím
0D	DCR	C	43	MOV	B,E	79	MOV	A,C	AF	XRA	A	E5	PUSH	H
0E	MVI	C, bájt	44	MOV	B,H	7A	MOV	A,D	B0	ORA	B	E6	ANI	Bájt
0F	RRC	-	45	MOV	B,L	7B	MOV	A,E	B1	ORA	C	E7	RST	4
10	-	-	46	MOV	B,M	7C	MOV	A,H	B2	ORA	D	E8	RPE	-
11	LXI	D, dupla	47	MOV	B,A	7D	MOV	A,L	B3	ORA	E	E9	PCHL	-
12	STAX	D	48	MOV	C,B	7E	MOV	A,M	B4	ORA	H	EA	JPE	Cím
13	INX	D	49	MOV	C,C	7F	MOV	A,A	B5	ORA	L	EB	XCHG	-
14	INR	D	4A	MOV	C,D	80	ADD	B	B6	ORA	M	EC	CPE	Cím
15	DCR	D	4B	MOV	C,E	81	ADD	C	B7	ORA	A	ED	-	-
16	MVI	D, bájt	4C	MOV	C,H	82	ADD	D	B8	CMP	B	EE	XRI	Bájt
17	RAL	-	4D	MOV	C,L	83	ADD	E	B9	CMP	C	EF	RST	5
18	-	-	4E	MOV	C,M	84	ADD	H	BA	CMP	D	F0	RP	-
19	DAD	D	4F	MOV	C,A	85	ADD	L	BB	CMP	E	F1	POP	PSW
1A	LDAX	D	50	MOV	D,B	86	ADD	M	BC	CMP	H	F2	JP	Cím
1B	DCX	D	51	MOV	D,C	87	ADD	A	BD	CMP	L	F3	DI	-
1C	INR	E	52	MOV	D,D	88	ADC	B	BE	CMP	M	F4	CP	Cím
1D	DCR	E	53	MOV	D,E	89	ADC	C	BF	CMP	A	F5	PUSH	PSW
1E	MVI	E, bájt	54	MOV	D,H	8A	ADC	D	C0	RNZ	-	F6	ORI	Bájt
1F	RAR	-	55	MOV	D,L	8B	ADC	E	C1	POP	B	F7	RST	6
20	RIM	-	56	MOV	D,M	8C	ADC	H	C2	JNZ	Cím	F8	RM	-
21	LXI	H, dupla	57	MOV	D,A	8D	ADC	L	C3	JMP	Cím	F9	SPHL	-
22	SHLD	Cím	58	MOV	E,B	8E	ADC	M	C4	CNZ	Cím	FA	JM	Cím
23	INX	H	59	MOV	E,C	8F	ADC	A	C5	PUSH	B	FB	EI	-
24	INR	H	5A	MOV	E,D	90	SUB	B	C6	ADI	Bájt	FC	CM	Cím
25	DCR	H	5B	MOV	E,E	91	SUB	C	C7	RST	0	FD	-	-
26	MVI	H, bájt	5C	MOV	E,H	92	SUB	D	C8	RZ	-	FE	CPI	Bájt
27	DAA	-	5D	MOV	E,L	93	SUB	E	C9	RET	-	FF	RST	7
28	-	-	5E	MOV	E,M	94	SUB	H	CA	JZ	Cím			
29	DAD	H	5F	MOV	E,A	95	SUB	L	CB	-	-			
2A	LHLD	Cím	60	MOV	H,B	96	SUB	M	CC	CZ	Cím			
2B	DCX	H	61	MOV	H,C	97	SUB	A	CD	CALL	Cím			
2C	INR	L	62	MOV	H,D	98	SBB	B	CE	ACI	Bájt			
2D	DCR	L	63	MOV	H,E	99	SBB	C	CF	RST	1			
2E	MVI	L, bájt	64	MOV	H,H	9A	SBB	D	D0	RNC	-			
2F	CMA	-	65	MOV	H,L	9B	SBB	E	D1	POP	D			
30	SIM	-	66	MOV	H,M	9C	SBB	H	D2	JNC	Cím			
31	LXI	SP, dupla	67	MOV	H,A	9D	SBB	L	D3	OUT	Bájt			
32	STA	Cím	68	MOV	L,B	9E	SBB	M	D4	CNC	Cím			
33	INX	SP	69	MOV	L,C	9F	SBB	A	D5	PUSH	D			
34	INR	M	6A	MOV	L,D	A0	ANA	B	D6	SUI	Bájt			
35	DCR	M	6B	MOV	L,E	A1	ANA	C	D7	RST	2			

## Irodalomjegyzék

1. Dr. Madarász László: Mikroprocesszorok, mikroszámítógép elemek. Főiskolai jegyzet (H-276), GAMF, Kecskemét, 1998.
2. Dr. Madarász László: Mikroelektronikai gyakorlatok V. Főiskolai jegyzet (H-344), KF GAMF Kar, Kecskemét, 2004.
3. Intel: MC+-80 User's Manual (With Introduction to MCS-85). Intel Corporation. 98-153D.