

MISKOLCI EGYETEM

Villamosmérnöki Intézet

Automatizálási Tanszék

MIKROPROCESSZOROK ÉS ALKALMAZÁSUK

Oktatási segédlet (javított és bővített kiadás)

*Gépész informatikus, anyagmérnök automatizálási, gépész mechatronikai, földtudomány levegő tisztaság védelem
szakirányos és villamos hallgatók részére*

<http://mazzola.iit.uni-miskolc.hu/~gardus>

Dr. Gárdus Zoltán Ph.D.

egyetemi adjunktus

Miskolc

2005.

Tartalomjegyzék

1. Bevezetés a mikroprocesszor technikába (INTEL 8085 mikroprocesszor).....	3
2. INTEL 8255 PIO (paralel I/O).....	13
3. Programozási feladatok 8085 mikroprocesszorral.....	15
4. INTEL 8086-8088 mikroprocesszor hardver.....	18
5. Félvezetős memóriák.....	21
6. Memóriák bővítése.....	24
7. Programozási feladatok 8086 mikroprocesszorral.....	26
8. Általános kérdések a mikroprocesszorok tárgyköréből.....	28
9. Programozási feladatok mikroprocesszorokkal.....	31
10. Minta zárthelyi feladatok.....	33
11. INTEL 8085 mikroprocesszor utasításkészlete.....	34
12. Kidolgozott szoftverek 8085 és 8086 (8088) mikroprocesszorokra.....	43

Az alábbi anyag szorosan kapcsolódik az előadások és gyakorlatok témaköreihez, amelyek megértéséhez elengedhetetlen az előadások és a gyakorlatok látogatottsága

1. Bevezetés a mikroprocesszor technikába

Didaktikai okokból elsőként egy INTEL gyártmányú 8 bites rendszerű mikroprocesszorral ismerkedünk meg.

INTEL 8085 mikroprocesszor

A mikroszámítógép az alábbi funkcionális egységekből áll:

CPU: Central Processor Unit (INTEL 8085 μ p);
RAM: Random Access Memory;
ROM: Read Only Memory;
I/O egység [INPUT/OUTPUT (8212, 8251, 8255)].

A fenti egységeket BUS (sínrendszer) vonalak kötik össze:

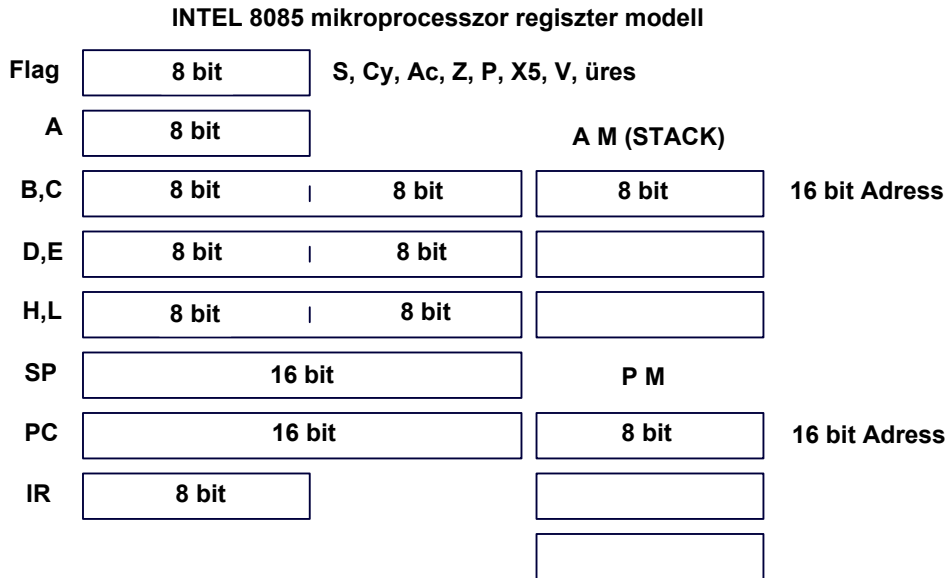
Cím BUS 16 bit);
Adat BUS (8 bit);
Vezérlő BUS (8 bit).

A CPU felépítése:

regiszterek és a flag-regiszter;
ALU: aritmetikai/logikai egység;
vezérlő áramkörök;
belső BUS rendszer.

A CPU funkciói:

utasítás lehívás a PM-ből (Fetch);
az utasítás dekódolása;
a dekódolt utasítások végrehajtása;
adatokkal I/O (I/OW, I/OR) műveletek végrehajtása;
AM Write/Read (MW, MR);
INTERRUPT;
WAIT.



A feni ábra az INTEL 8085 µp regiszter és flag modelljét szimbolizálja

Tipikus számítógép műveletek:

időzítés;
utasítás lehívási ciklus (Fetch);
Memory Read (M/R);
Memory Write (M/W);
I/O Read (I/OR);
I/O Write (I/OW);
STACK Read (S/R);
STACK Write (S/W);
WAIT (várakozás);
INTERRUPT (megszakítás);
HOLD;
HLT;
HLT INTERRUPT.

Az INTEL 8085 µp CMOS technológiával készült VLSI áramkör, egyetlen tápfeszültséget igényel, ami TTL szintű + 5 V ±5%.

Regiszterek:

A: AKKUMULÁTOR az egyik legfontosabb regiszter, mivel az I/O műveletek, az aritmetikai/logikai műveletek, eredmények legnagyobb része itt történik, tehát forrás és célregiszterként szolgál;
a B,C - D,E regiszterpárok általános célúak, a munkaregiszterek funkcióit látják el együttesen;
a H,L regiszterpár a memória funkcióját biztosítja, ami azt jelenti, hogy a H,L regiszterpár által megcímezett AM 1 byte-os tartalmát címezi;
SP: STACK POINTER (stack mutató), ami többszöri program interrupt elfogadását és végrehajtását teszi lehetővé, számára a lefoglalt hely az AM-ból származik;

PC: PROGRAM COUNTER (program számláló), a PC-ben lévő címhez tartozó utasítást, egy utasítás lehívási cikluson (Fetch-en) keresztül a CPU lehívja a PM-ből, majd a végrehajtást követően a PC tartalmát növeli 1-gyel;

IR: INSTRUCTION POINTER REGISTER (utasítás dekódoló regiszter), dekódolja a PM-ből lehívott utasítást.

Flag-regiszter:

S: SIGNUM előjel flag;

C_y: CARRY átvitel flag;

A_c: járulékos átvitel flag a 3. biten;

Z: ZERO flag;

P: paritás flag;

X₅: alól/felül csordulás flag;

V: túlcsordulás flag.

INTEL 8085 µp KIT használata, működése:

bekapcsolást követően SUBST MEM billentyű megnyomása után beállítjuk a szabad RAM terület (3000h, ill. géporientált 2000 h) címet;

ez után kiküldünk a 80h (P_A, P_B, P_C portok a 8255 PIO MODE „0”-ás üzemmódjában kimenetként szolgálnak, lásd később) adatot, a 8255 PIO C3 parancsregiszterének címére, az alábbiak szerint:

3000: MVI	A,80	3000 3E
OUT	C3	3001 80
		3002 D3
		3003 C3

Az assembly utasítások lefordítása gépi kódra
Byte-onként (utasítás-adat)

- beírjuk a programot byte-onként, amit a NEXT billentyűvel aktivizálunk;
- a program utolsó soraként EXEC billentyűt nyomunk;
- futtatás a GO billentyű segítségével történik, majd ezt követően a program kezdőcímét (3000h, 2000h) billentyűzzük be;
- az EXEC billentyű megnyomását követően a program futása történik.

INTEL 8085 µp szoftver:

Címzési módok:

direkt;

regiszter;

regiszter indirekt;

közvetlen.

Jelzőbitek kombinációi a feltételes vezérlésátadó utasítások magyarázatához:

NZ: Z=0 (Zéró);

Z: Z=1;

NC: C_y=0 (Carry);

C: C_y=1;

PO: P=0 (Paritás);

PE: P=1;

P: S=0 (Signum);

M: S=1;

NX₅: X₅=0 kizárólag feltételes ugrásoknál (Alól / Felül csordulás);

X₅: X₅=1. kizárólag feltételes ugrásoknál.

DMA: (Direct Memory Access) közvetlen memória elérés, hozzáférés. A közönséges I/O műveleteknél a CPU vezérli az adatáramlást. DMA esetében a perifériális berendezések gyorsabban végzik el mint a CPU. Ilyen módon megnövelhető a rendszer átbocsátó képessége, sebessége, ha az I/O eszköz maga valósítja meg a direkt átvitelt. A tartáskérés nyugtázása után (a CPU HLDA lábán keresztül történik) a CPU buszrendszerét nagyimpedanciás állapotba helyezi.

INT: (Interrupt) megszakítás kérés. A 8085 mikroprocesszor megszakításkérő bemenete az INT láb, amely szoftver úton az EI (Enable interrupt) és a DI (Disable interrupt) utasítások révén aktivizálható. Ezek az utasítások a CPU-ban lévő INTE F.F.-ot elfogadáskor 1-be, tiltáskor pedig 0-ba billentik.

INTA: megszakítás kérés nyugtázása

INTERRUPT: (SUBROUTINE) önálló program a programban, programmegszakítás. A CPU hatásfokát javítja. Általánosságban egy CALL A16 (feltétel nélküli szubrutin hívás) utasítás végrehajtását követően a főprogram futása ideiglenesen felfüggesztődik, majd a PC tartalma átíródik az A16 címre. Az A16 címen folytatódik a végrehajtás (ez a programrészlet a szubrutin), majd egy RET (feltétel nélküli visszatérés a szubrutinból) utasítás hatására a PC-be visszatöltődik az a cím, ahol folytatódnia kell majd a főprogramnak. A szubrutin futása alatt a Stack-ben (a hely a Stack számára AM-ből különítődik el) tárolódik 2 byte-on, ssss-1, ssss-2 címeken (LIFO) a főprogram azon címe, ahol a program futása felfüggesztődött. A szubrutinok egymásba ágyazhatók, skatulyázhatók. A szubrutinra egy 16 bites címmel hivatkozunk.

MACRO: assembly szintű utasítások meghatározott sorozata, a MACRO és ENDM utasítások közé írjuk. A fentiekből következően a címével hivatkozunk rá és hosszával annyiszor épül be a programba, ahányszor meghívjuk. A makrók használata akkor célszerű, ha kevés utasításokból állnak. Természetesen a makrók tartalmazhatnak szubrutint is.

WAIT: várakozás. Ha a perifériális eszköz (pl. memória) sebessége alacsonyabb a CPU sebességénél, akkor a CPU addig várakozó állapotba kerül, amíg a perifériális eszköz nem reagál a CPU által megküldött kiolvasandó címre. Azonban a legtöbb memória alkalmas, hogy a CPU-val azonos sebességgel működjön. Ezért a CPU tartalmaz egy szinkronizálási lehetőséget, mely lehetővé teszi a memória számára, hogy kérjen egy WAIT állapotot. A memória jelet generál a CPU READY vezetékeire, eredményeképpen a CPU WAIT állapotba kerül. Miután memória elvégzi a művelete a CPU READY vezetéke felszabadul.

HOLD: tartás kérés (a DMA-kor)

HLDA: tartás kérés, nyugtázás

Az INTEL 8085 μ p illesztése I/O eszközökhöz:

Izolált I/O: az I/O rendszer elválasztja a memória címtérét az I/O címtérétől. Ez általában az A regiszterrel való közlekedést jelent (IN port, OUT port utasítások).

Memóriában leképzett I/O: a memória címtér egy részét hozzárendeljük az I/O címtérhez, akkor a programozás hatáskora javul, ugyanis az I/O-t ugyanazon utasítások használatával tudja kezelni a rendszer, melyeket a memória kezelésére használunk. Előállítunk új invertált MEMR, MEMW jeleket az A15 címvezeték kapuzása révén. Így az A15-öt I/O jelzőbitként kapjuk és:

ha A15=0, a memória aktív;

ha A15=1, a memória inaktív.

A INTEL 8085 μ p utasításkészlete:

Adatmozgató utasítás csoport:

MVI r,d8: r regiszter által specifikált helyre közvetlen 8 bites adat betöltése, „r” lehet:

A, B, C, D, E, H, L;

LXI rp,d16: közvetlen 16 bites adatbetöltés „rp” által specifikált regiszterpárba, „rp” lehet

BC, DE, HL, SP;

LDA A16: az akkumulátor tartalmának indirekt betöltése az AM A16-al specifikált címéről;

STA A16: az akkumulátor tartalmának indirekt tárolása az AM A16-al specifikált címére;

LHLD A16: H,L betöltése indirekt úton AM-ból, A16 címről L-be, A16₊₁címről H-ba;

SHLD A16: H,L tárolása indirekt úton AM-ba, A16 címre L, A16₊₁ címre H tárolódik;

LDAX rp: rp (lehet: B,C; D,E), akkumulátor indirekt betöltése, rp által specifikált címről;

STAX rp: rp (lehet: B,C; D,E), akkumulátor indirekt tárolása, rp által specifikált címre;

XCHG: H,L \leftrightarrow D,E regiszterpár tartalmának cseréje;

MOV r₁, r₂: regiszter \leftarrow regiszter

MOV r,M: regiszter \leftarrow memória (AM)

MOV M,r: memória (AM) \leftarrow regiszter közötti adatmozgatások;

PCHL: H,L regiszterpár tartalma PC-be töltődik.

Ha a program SUBROUTINE tartalmaz, a Stack-et inicializálni kell.

Az inicializálás az alábbiak szerint történik:

Inicializálás	3000	LXI	SP, 20C2	3000	31
		MVI	A, 80	3001	C2
		OUT	C3	3002	20
				3003	3E
				3004	80
					C3

Példa az adatmozgató utasítások használatára:

A MOV utasítás használata:

MOV r ₁ , r ₂	r ₁ ←r ₂
MOV r, M	r←M (H,L)
MOV M, r	M (H,L)←r
	M=H,L regiszterpár által megcímezett AM 1 byte-os tartalma
3000 MVI A, 3E	az akkumulátor feltöltése 3Eh adattal
LXI H, 2B3C	a memória címének a beállítása az AM-ban
MOV M, A	A tartalma a M-be kerül
LDA 2B3C	visszatölti indirekt úton az M tartalmát a 2B3C címről A-ba

LDAX rp utasítás működése:

```
MVI A, 80
LXI H, 2000
MOV M, A
LXI B, 2000
LDAX B
```

Aritmetikai utasítás csoport:

1.	2.	3.	4.	
ADD r	ADC r	SUB r	SBB r	
ADD M	ADCM	SUB M	SBB M	DAD rp
ADI d8	ACI d8	SUI d8	SBI d8	

Aritmetikai utasítás csoport magyarázata:

összeadás átvitel nélkül, r (lehet: A, B, C, D, E, H, L) által specifikált regiszter tartalma hozzáadja az A tartalmához, az eredmény A-ban képződik;

összeadás átvitel nélkül M (H,L regiszterpár által megcímezett AM 1 byte-os) tartalmát hozzáadja A tartalmához, az eredmény A-ban képződik;

összeadás átvitel nélkül, a d8-as adatot hozzáadja A tartalmához, az eredmény A-ban képződik;

ugyanaz mint az mint a fenti csak az összeadás figyelembe veszi a C_y flag értékét;

kivonás átvitel nélkül, r (lehet: A, B, C, D, E, H, L) által specifikált regiszter tartalmát kivonja A tartalmából, az eredmény A-ban képződik;

kivonás átvitel nélkül, M (H,L regiszterpár által megcímezett AM 1 byte-os) tartalmát kivonja A tartalmából, az eredmény A-ban képződik;

kivonás átvitel nélkül, a d8-as adatot kivonja A tartalmából, az eredmény A-ban képződik;

ugyanaz mint a fenti csak a kivonás figyelembe veszi a C_y flag értékét.

DAD rp 16 bites összeadás, rp (lehet: B,C; D,E; H,L; SP) által specifikált 16 bites adatot hozzáad a H,L rp tartalmához és az eredmény H,L-ben képződik.

DAA decimális korrekció, az A regiszterben lévő 8 bites hexadecimális számot 2 db négy bites NBCD számmá konvertálja a következő módon:

ha az A regiszter legalacsonyabb 4 bitjének tartalma > 9 és $A_c=1$, akkor hozzáad 6-ot;
 ha az A regiszter legmagasabb 4 bitjének tartalma > 9 és $C_y=1$, akkor hozzáad 6-ot;
 Kizárólag összeadás esetében alkalmazható.

Logikai utasítás csoport:

1.	2.	3.	
ANA r	ORA r	XRA r	STC
ANA M	ORA M	XRA M	CMC
ANI d8	ORI d8	XRI d8	CMA

Logikai utasításcsoport magyarázata:

közvetlen bitenkénti AND kapcsolat r (lehet: A, B, C, D, E, H, L) által specifikált regiszter és az A regiszter között, az eredmény A-ban képződik;

közvetlen bitenkénti AND kapcsolat M (H,L által megcímezett AM 1 byte-os tartalma) és az A tartalma között, az eredmény az A-ban képződik;

közvetlen bitenkénti AND kapcsolat a d8-as adat és az A tartalma között, az eredmény A-ban képződik;

ugyanaz mint a fenti, csak bitenkénti OR kapcsolat;

ugyanaz mint az előző, csak bitenkénti ANTIVALENCIA (KIZÁRÓ VAGY) kapcsolat.

STC: CARRY FLAG 1-be állítása.

CMC: CARRY FLAG törlése.

CMA: A-tartalmának komplementálása.

Inkrementáló utasítás csoport:

INR r: r (lehet: A, B, C, D, E, H, L) által specifikált regiszter tartalmát növeli 1-gyel.

INR M: M tartalmát (H,L által megcímezett AM 1 byte-os rekesze) növeli 1-gyel.

INX rp: rp (lehet: B,C; D,E; H,L; SP) által specifikált regiszterpár tartalmát növeli 1-gyel.

Dekrementáló utasítás csoport:

DCR r: r (lehet: A, B, C, D, E, H, L) által specifikált regiszter tartalmát csökkenti 1-gyel.

DCR M: M tartalmát (H,L által megcímezett AM 1 byte-os rekesze) csökkenti 1-gyel.

DCX rp: rp (lehet: B,C; D,E; H,L; SP) által specifikált regiszterpár tartalmát csökkenti 1-gyel.

Beviteli-kiviteli (I/O) utasítás csoport:

IN port: (a port 8 bites hexadecimális címet szimbolizál) az utasítás végrehajtását követően a port 1 byte-os címéről az 1 byte-os adat beolvassódik az A regiszterbe.

OUT port: (a port 8 bites hexadecimális címet szimbolizál) az utasítás végrehajtását követően a port 1 byte-os címére az 1 byte-os adat kiíródik.

Komparáló utasítás csoport:

CMP r: r (lehet: A, B, C, D, E, H, L) által specifikált regiszter tartalmát kivonja A-ból.

CMP M: M tartalmát (H,L által megcímzett AM 1 byte-os rekesze) kivonja A-ból.

CPI d8: a d8-as adatot kivonja A-ból.

A <; >; = relációk részletes működése:

ha A=adat, akkor a Z=1 (Zero flag) a THEN ágon JZ A16, CZ A16 és RZ utasítások alkalmazhatók;

ha A<adat, akkor a C_y=1 (Carry flag) a THEN ágon JC A16, CC A 16, és RC utasítások alkalmazhatók;

ha az „=” és a „<” relációk nem teljesülnek, akkor A>adat.

Rotáló utasítás csoport:

RLC: az A tartalmának balra rotálása a C_y flag érintésével.

RRC: az A tartalmának jobbra rotálása a C_y flag érintésével.

RAL: az A tartalmának balra rotálása a C_y flag-en keresztül.

RAR: az A tartalmának jobbra rotálása a C_y flag-en keresztül.

Feltétel nélküli vezérlésátadó utasítások:

Feltételnélküli ugrás Feltételnélküli SUBROUTINE hívás Feltételnélküli visszatérés

JMP A16

CALL A16

RET

Feltételes vezérlésátadó utasítások:

Feltételes ugrások

Feltételes SUBROUTINE hívások

Feltételes visszatérések

JNZ A16

CNZ A16

RNZ

JZ A16

CZ A16

RZ

JNC A16

CNC A16

RNC

JC A16

CC A16

RC

JPO A16

CPO A16

RPO

JPE A16

CPE A16

RPE

JP A16

CP A16

RP

JM A16

CM A16

RM

JNX₅ A16

JX₅ A16

Megjegyzés, a flag-ek kombinációinak jelentése a fentiekben már ismertetésre került!

Programmegszakítás (szubrutin) bemutatása:

Példa: Stack inicializálására, szubrutin hívásra, működésre a programban:

```
3000 LXI SP, 20C2
      MVI A, 80
      OUT C3
hurok: IN E0 (E0 portról adatot olvas be az A-ba)
      CALL 3050 (SUBROUTINE meghívása a 3050 címen)
3050: ACI 03 (a 03 adatot hozzáadja az A tartalmához)
      RET (visszatérés a SUBROUTINE-ből)
      OUT C0
      MOV B,A
      JMP hurok
```

A STACK műveletek utasítás csoportja:

PUSH rp: (írás a Stack-be) rp (lehet: B,C; D,E; H,L; PSW) regiszterpár magasabb helyiértékű regiszterének tartalmát átviszi arra a memóriahelyre, melynek címe 1-gyel alacsonyabb (ssss-1), mint az SP regiszter (ssss) tartalma. Az alacsonyabb helyiértékű regiszter tartalmát pedig arra a memória helyre viszi át, melynek címe 2-vel kevesebb (ssss-2), mint az SP regiszter tartalma.

POP rp: (olvasás a Stack-ből) rp (lehet: B,C; D,E; H,L; PSW) a memória tartalmát visszaírja rp alacsonyabb helyiértékű regiszterébe, arról a memória helyről, melynek a címét SP (ssss) regiszter határozza meg. A magasabb helyiértékű regiszter tartalmát visszaírja rp magasabb helyiértékű regiszterébe, arról a memória helyről, melynek címe 1-gyel magasabb (ssss+1), mint az SP regiszter tartalma.

PSW: Program Status Word (az A regiszter és a flag-regiszter tartalmából képzett szó)

Programmegszakítás (szubrutin) utasítás csoport:

EI: Enable Interrupt (megszakítás engedélyezés);
 DI: Disable Interrupt (megszakítás tiltás);
 SIM: Set Interrupt Mask;
 RIM: Read Interrupt Mask;
 RSTV: Restart On Overflow (megszakítás, ha a V flag értéke „1”).

Leállítás, üres utasítás csoport:

HLT: megállás, leállítás.
 NOP: üres utasítás (pl. késleltetések készítésére alkalmazható).

Példa: ~1s-os késleltetés programozására:

```
LXI D, FFAA (késleltetési érték beállítása a D,E regiszterpárba)
hurok: DCX D
      JNX5 hurok
```

Speciális megszakítás utasítás csoport:

RESTART megszakítás kérés

Ez egy speciális 1 byte-os CALL utasítás, hatására elsőként a PC tartalma bekerül az SP által meghatározott memória címre úgy, hogy az SP-1 címre a magasabb, az SP-2 címre pedig az alacsonyabb helyiértékű byte kerül. Az SP tartalma (2 byte-os cím) kettővel csökken, ezt követően a vezérlés átadódik arra a címre, mely „n”-nek 8 szorososa.

Példa RST 3 beugrasi címének kiszámítására:

$3 \times 8 = 24_{(10)} = 1\ 1000_{(2)} = 18_{(h)}$. Tehát ezt követően a program a 0018h címen fog folytatódni.

Speciális megszakítások 8085 μ p esetében:

RST n: (n=8-at jelent)

RST 0

RST 1

RST 2

RST 3

RST 4

RST 5

RST 6

RST 7

Nem maszkolható megszakítások a fentiek.

RST 5.5

RST 6.5

RST 7.5

Maszkolható megszakítások

TRAP: nem programozott feltételes ugrás (csabda)

2. INTEL 8255 PIO (paralel I/O)

Az INTEL 8255 PIO (Párhuzamos I/O) MODE „0”-ás (alapvető bevétel/kivétel) üzemmódjának programozása

A fenti áramkör három (A, B, C) egyenként 8 bites porttal rendelkezik, ami együttesen $3 \times 8 = 24$ I/O vonalat jelent. A C port meg van felezve C_a , és C_f jelölésű alsó (0-3) és felső (4-7) fél byte-ra. Így a bemenetek számának megfelelően 16 kombináció áll a rendelkezésünkre. Ebből következik az, hogy a portok kombinációi az O, O, O, O-I, I, I, I (O: output logikai „0”, I pedig input logikai „1”-et fog szimbolizálni).

Az INTEL 8255 PIO MODE „0”-ás üzemmód bitkiosztása

A három port elhelyezkedése:

	A	C_f	B	C_a
D_7	D_6	D_5	D_4	D_3
D_2	D_1	D_0		

A MODE „0”-ás üzemmódban a $D_7 = 1$, a $D_6, D_5, D_2 = 0$.

A fentiek értelmében az INTEL 8255 PIO MODE „0”-ás üzemmódjának a kapocstáblázata az alábbi.

D_4	D_3	D_1	D_0
A	C_f	B	C_a
O	O	O	O
O	O	O	I
O	O	I	O
O	O	I	I
O	I	O	O
O	I	O	I
O	I	I	O
O	I	I	I
I	O	O	O
I	O	O	I
I	O	I	O
I	O	I	I
I	I	O	O
I	I	O	I
I	I	I	O
I	I	I	I

Példa a 8255 PIO MODE „0”-ás üzemmódjának programozására:

A port: kimenet (O);
 B port: bemenet (I);
 C port: kimenet (O).

Vezérlő byte: $\begin{matrix} 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{matrix} = 82h$.

A PIO portjainak beállítása a fentiek kijelölése alapján, 8085 mikroprocesszorral:

```
3000 MVI A, 82  
      OUT C3 (PIO parancsregiszterének címe, INTEL 8085 $\mu$ p KIT).
```

A PIO portjainak beállítása a fentiek kijelölése alapján, 8086 mikroprocesszorral:

```
MOV DX, A16 (a PIO parancsregiszterének címe, ami a PC ISA BUS-ba helyezett I/O  
             kártyán lévő 8255 PIO parancsregiszterének a címét 0303h jelenti)  
MOV AL, 82  
OUT DX, AL
```

3. Programozási feladatok 8085 mikroprocesszorral

Az alábbi mintaprogramok a legáltalánosabb vezérlési és az IF THEN ELSE struktúrák assembly listáit tartalmazzák:

Feladat: 8 bites futófény programozására, változtatható futási sebességgel INTEL 8085 μ p-ral:

```

3000      LXI SP, 20C2h
          MVI A, 80h
          OUT C3h          (emulátor esetén nem kell)
          MVI A, 01h
          OUT C2h  kijelzés (világít a 0-adik biten lévő első LED)
          CALL rutin      (rutin 16 bites címet jelent)
hurok:   RLC              az A tartalmát balra rotálja
          OUT C2h        ciklusban történő kijelzés
          CALL rutin
          JMP  hurok

rutin:   MOV B, A
          IN  E1h        E 1 portról állítható a program futási sebessége
          MOV D, A
          MVI E, 00h     E regiszter lenullázása, mivel induláshoz véletlen-
hurok1:  DCX D           } D,E regiszterpárt csökkentés 1-gyel
          JNX5 hurok1
          MOV A, B
          RET

```

Az alábbi programrészlet nyomógomb megnyomására késleltet, ami csak a 0.-adik biten lévő nyomógomb aktivizálására működik:

```

          rutin: MOV B, A
hurok 3: IN  E2h
          ANI  01h  a 0.-adik biten elhelyezkedő nyomógomb kimaszkolása
          CPI  01h  összehasonlítás (komparálás)
          JZ   hurok 2
          JMP  hurok 3
hurok 2: LXI D, érték késleltetési érték (FFAA) beállítása (nyomógomb prellmentesítése)
hurok 4: DCX D
          JNX5 hurok4
          MOV A, B
          RET

```

E 2 port 4 bitjének felfutó élére induljon meg a futófény

```

rutin:   MOV B, A
hurok   IN  E2h
          ANI 10h
          CPI 10h
          JZ  hurok 1
          JMP hurok

```

hurok1: LXI D, érték
 hurok 2: DCX D
 JNX5 hurok 2
 MOV A, B
 RET

Feladat: 8 bites JOHNSON kódú számláló készítése, INTEL 8085 μ p-ral:

A 4 bites JOHNSON kód AM-ba (adatmemóriába) való tárolása

	JOHNSON kód
A memória táblázat kezdő címe (16 bites cím): 00h	0000
	01h
	03h
	07h
	0Fh
	0Eh
	0Ch
	08h
	1000

INTEL 8085 μ p program:

```

    LXI SP, 2002h
    MVI A, 80h
    OUT C3h
    LXI H, a memória táblázat kezdő címe (JOHNSON kód 00h)
    MOV A, M
    OUT C2h
    CALL rutin
hurok 1: INX H
    MOV A, M
    OUT C2h
    CALL rutin
    CPI 08 az akkumulátor tartalmának összehasonlítása 08-al?
    JZ hurok
    JMP hurok 1
hurok:  HLT
  
```

Feladat: szorzótábla készítése 0-99-ig INTEL 8085 μ p-ral:

```

hurok1: IN  E1h
    MOV C, A          C szer kell összeadni A tartalmát B tartalmával
    IN  E2h
    MOV B, A
    XRA A            kinullázzuk az A tartalmát
hurok:  ADC B        B-t hozzáadjuk A tartalmához
    DAA
    DCR C
    JNZ hurok
    OUT C0h
    JMP hurok1
  
```


Feladat: 8 bites SHIFT regiszter készítése 8085 μ p-ral, a 04h port szolgáltatja az információt

1000	LXI SP, 2000	2000	MOV B, A
1003	IN 04	2001	LXI D, AA00
1005	ANI 03	2004	DCX D
1007	OUT 05	2005	JNZ 2004
1009	CALL 2000		MOV A, B
100C	MVI L, 08		RET
100E	RLC		
	OUT 05		
	CALL 2000		
	DCR L		
	JNZ 100E		

Feladat: komparátor programozása INTEL 8085 μ p-ral:

```

1000: IN 04
      CPI 08
      JZ 1018
      JC 1011
      MVI A, 04
      OUT 05
      JMP 1000
1011: MVI A, 01
      OUT 05
      JMP 1000
1018: MVI A, 02
      OUT 05
      JMP 1000

```

A programozási feladatok gyakorolhatók az mcs85.exe file indítását követően. A szoftver ezenkívül tartalmaz komplett 8085 mikroprocesszor hardvert és szoftvert, valamint az elsajátításhoz és a gyakorláshoz elengedhetetlen TESZT kérdéseket. Lehetőséget szolgáltat az utasítások és a programok végrehajtási ciklusonként (Fetch) történő megtekintésére is.

3. INTEL 8086-8088 mikroprocesszor hardver

A 8088 CPU 8 bites külső adatsínnel rendelkezik, míg a 8086 CPU egyidejűleg 16 bites adatok átvitelére is alkalmas. A 8 MHz frekvenciával működő 8086 μ p ~ 10-szer hatékonyabb elődénél. Két különálló egységből áll:

Végrehajtó egység EU (Execution Unit);

BUS illesztő egység BIU (BUS-Interface Unit), kapcsolatot tart a külvilággal.

Az EU részei:

általános célú regiszterek;
operandusok;
ALU;
flag-ek.

A BIU részei:

regiszter készlet (ES, CS, DS, SS, IP);
címösszedómű;
FIFO (First-In First-Out, ahol a lekérdezés, a beérkezésnek megfelelő sorrendjében történik) regiszter.

A 8086 CPU címezési módjai:

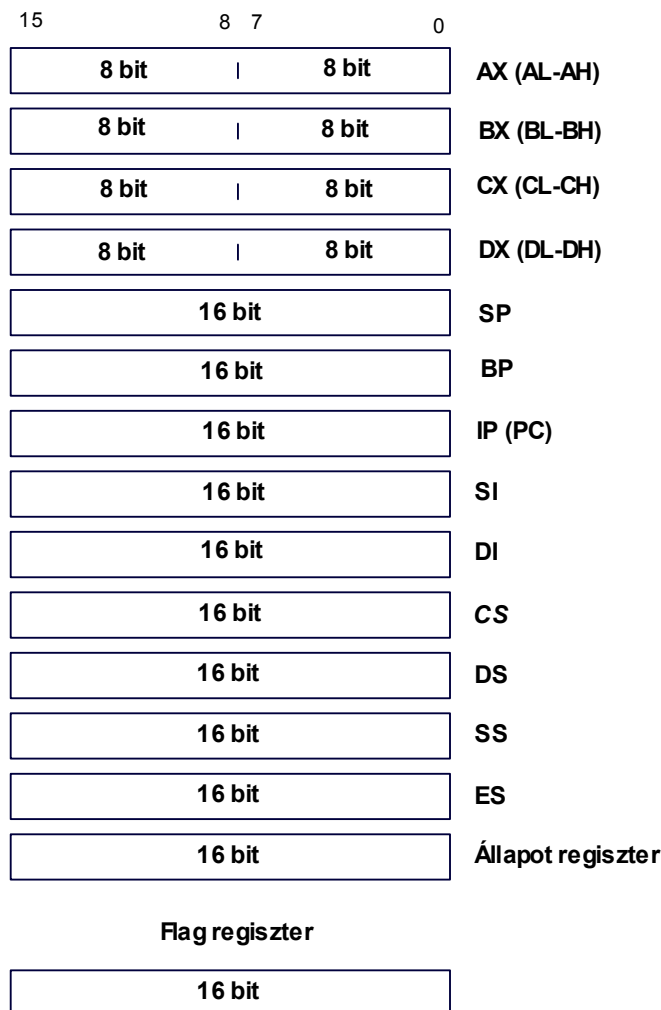
Minden 8086 memóriacím valamely **szegmens regiszter tartalmának** és egy **effektív memóriacímnek** az összege;
az effektív memóriacím kiszámítható úgy, hogy a szegmens regiszter tartalma 4 bittel balra lépve adódik hozzá az effektív memória címhez, így képződik a 20 bites fizikai cím.

A 8086 CPU programozási címezési módjai:

direkt;
direkt indexelt;
implicit;
bázis relatív;
stack.

A 8086 μ p regiszter és flag-modellje:

INTEL 8086 mikroprocesszor regiszter modell



Flag-ek: C, P, A, Z, S, T, I, D, O
A felmaradó bitek üresek.

Regiszterek:

4 db 16 bites általános célú regiszter (AX, BX, CX, DX), amik AL, AH; BL, BH; CL, CH; DL, DH egyenként 8 bites regiszterekre bonthatók, AX akkumulátor, minden I/O művelet ezen keresztül bonyolódik le, BX bázisregiszter, a címképzésben vesz részt, CX Counter, értéke dekrementálódik a string és a ciklus szervező utasítások használatakor, DX adatregiszter, aritmetikai műveletekben használatos és az I/O műveletek címét is tartalmazza;

SP Stack Pointer, minden SP-vel történő címzés implicit hivatkozás az SS regiszterre;

BP Base Pointer, a Stack-ben történő paraméterállítás a felhasználási területe;

indexregiszterek (SI, DI), szerepük a string műveleteknél van;

programszámláló (IP, PC Program Counter);

szegmensregiszterek (DS, CS, SS, ES).

A flag-regiszter (16 bites):

C: CARRY (átvitel) flag;
 P: paritás (párosság ellenőrző) flag;
 A: (járulékos CARRY) flag;
 Z: ZERO (zérus)flag;
 S: SIGNUM (előjel) flag;
 T: TRAP flag;
 I: INTERRUPT (megszakítás engedélyezés/tiltás) flag;
 D: DIRECTION (irány) flag;
 O: OVERFLOW (túlcsordulás) flag.

A PC párhuzamos portjának programozás szempontjából való leírása:

A PC párhuzamos (paralel), CENTRONIX szabványos felületű portjának programozását, az alábbiak szerint sajátítható el. A BIOS négy logikai periféria kialakítására ad módot: LPT1, LPT2, LPT3 és LPT4. A BIOS a megadott sorrendben keresi az egységeket, ha az adott egység be van építve, akkor azt alaphelyzetbe állítja és a soron következő perifériához rendeli. Egy gépben egyszerre három illesztőt lehet használni, amelyek báziscímei 378h, 278h és régebben a 3BCh a Hercules kártya paralel portjának a címét szolgáltatta.

A kártya regiszterei:

Cím+0 írható/olvasható, az illesztő adatregiszterébe a kimenetre kerülő adatot lehet beírni, az utoljára kiírt adat, vagy a kétirányú változatnál a vett adat olvasható be;
 Cím+1 csak olvasható állapotregiszter, a bitek kiosztása:0-2 nem használt, 3 hibajelző invertált ERROR jel értéke, 4 a berendezés kiválasztott SELECT jel értéke, 5 a nyomtatóban nincs papír PE jel értéke, 6 a berendezés a külön adatot átvette invertált ACK jel értéke, 7 a berendezés kész az adatátvitelre BUSY jel invertáltja;
 Cím+2 írható/olvasható, parancsregiszter, a regiszter aktuális értéke visszaolvasható, a bitek kiosztása:0 adat érvényesítése, STROBE, 1 automatikus soremelés engedélyezése, AUTO FD, 2 alaphelyzetbe állítás, invertált INT, minimum 50 ms ideig alacsony szint, 3 az átvitel irányának jelzése, SEL_IN, 0 adatvonalak bemenetek, 1 adatvonalak kimenetek, 4 megszakítás engedélyezés, 5 kétirányú változatnál a kimenet meghajtásának tiltása, 6-7 nem használt.

A 8086 mikroprocesszor utasításkészletét és programozási lehetőségeit, a „**Mikroprocesszorok és alkalmazásuk**” című jegyzet tartalmazza.

5. Félvezetős memóriák

A félvezetős memóriák funkció szerinti felosztása

RAM: Random Access Memory (közvetlen hozzáférésű, változtatható tartalmú, írható/olvasható memóriák RWM azon csoportját alkotják, amelyeknél az információ hozzáférési ideje annak címétől független). Sztatikus és dinamikus kivitelekben készülnek. A sztatikus RAM memóriák bipoláris tranzisztorokkal, illetve MOS (Metal Oxid Semiconductor) tranzisztorokkal épülnek fel, a dinamikus RAM memóriák pedig, MOS elemekkel.

ROM: Read Only Memory [csak kiolvasásra szánt ROM, PROM (felhasználáskor elektromosan programozható), EPROM, REEPROM, EAROM, RMM (elektromosan többször programozható, törölhető, illetve újraprogramozható) jelölésű memóriák]. Bipoláris és MOS tranzisztorokkal építhetők fel.

A bipoláris tranzisztorokkal felépített memóriák lehetnek:

- dióda mátrix;
- maszk ROM;
- PROM.

A MOS tranzisztorokkal felépített memóriák az alábbi típusúak:

MASZK programozott MOS ROM;

EPROM;

-FAMOS (a vezérlő elektródán ellentétes polaritású jellel törölhető);

-MNOS (Metal-Nitrid-Oxid-Silicon);

-MAOS (Metal-Aluminium-Oxid-Semiconductor).

PLA programozható logikai áramkör, az FPLA felhasználó által programozható logikai áramkör.

A PLA-k felfoghatók hiányos címmezű ROM memóriákként. Ismeretes, hogy a ROM, PROM, REEPROM memóriák egyik jellegzetes alkalmazási területe a kombinációs áramkörök, hálózatok, logikák helyettesítése. Ilyen esetekben a bemeneti változók a címvezetékekre kapcsolódnak, a kimeneti logikai függvényeket pedig az adatkimeneteken kapjuk meg. A PLA egy AND ROM-ból és egy OR ROM-ból épül fel. A PLA-k programozása gyárilag maszkolással történik.

PAL: Programmable Array Logic (programozható mátrix áramköri elrendezés)

GAL: Gate Array Logic (kapu szintű mátrix áramköri elrendezés)

FPGA: Field Programmable Gate Array (felhasználó által programozható kapu mátrix, tömb) A XILINX alapcella CMOS technológiájú áramkör, melynek angol neve LCA (Logic Cell Array), azaz logikai cellatömb. A MOS technikák közül mind a P-MOS, N-MOS és a CMOS alkalmas LSI egységek gyártására.

Speciális MOS technológiák:

- SILICON GATE P-MOS (szilícium vezérlő elektródás P-MOS);
- V-MOS (Vertical MOS);
- D-MOS (Double Diffused MOS);
- SOS (Silicon on Saphir);
- CCD (Charge Coupled Device) töltéscsatolt eszköz.

A legújabb fejlesztések közül néhányat megemlítve, amelyek széles sávban kapható tárcapacitással rendelkeznek az igénynek megfelelően, kis tömegűek és használatkor a PC USB portjára csatlakoztathatók:

- Flash RAM;
- Pen Drive;
- Traveling Disc.

A fentiekben említették mellett harmadik nagy memória típus a SERIAL (soros), ide tartoznak a SHIFT regiszterek.

A mágnesesség elvén működő technológiák:

- MBD (Magnetic Bubble Devices) mágneses buborék eszközök;
- FD (Floppy Disc);
- HD (Hard Disc).

Optikai úton működő technológia: CD (Compact Disc)

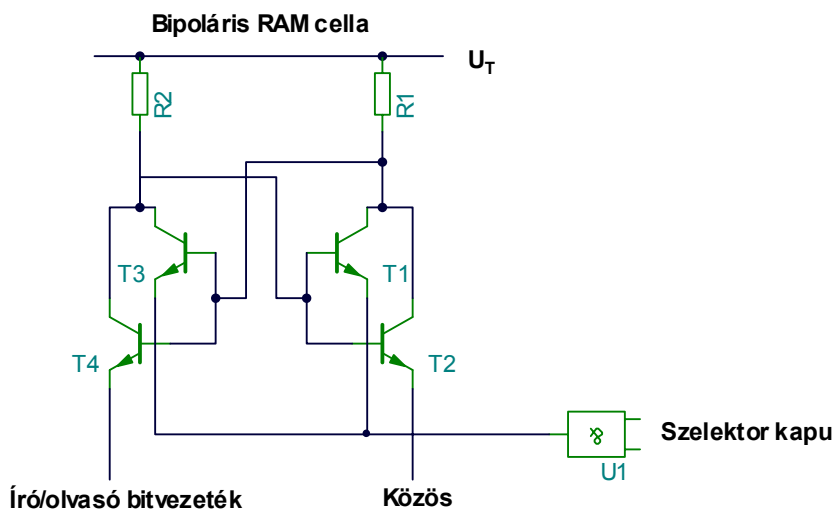
A memóriák jellemezhetők:

tárkapacitás: rekeszek száma;

cím: a rekeszek sorszám;

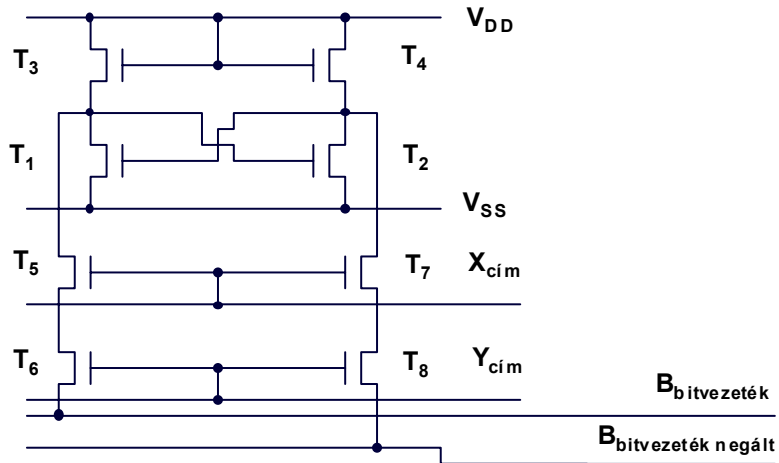
hozzáférési idő: a címzés pillanata és a tárolt információ elérése közötti idő;

szószervezés 1 byte=8 bit, 1 kbyte=1024 byte.



A fenti ábra egy bipoláris tranzisztorokkal felépített elemi RAM cellát szemléltet, ami tulajdonképpen egy bistabil multivibrátornak (RS F.F.) fogható fel.

Nyolc tranzisztossal felépített sztatikus MOS RAM cella



A nyolc tranzisztossal felépített sztatikus MOS RAM cellában a T₁, T₂ tranzisztorok a bistabil multivibrátor szerepét töltik be, a T₃, T₄, pedig MOS ellenállások. A T₅...T₈ tranzisztorok pedig a cella kiválasztását biztosítják.

6. Memóriák bővítése

Bővítésre akkor van szükség, ha az egy tokba integrált tárkapacitás nem elegendő az adott feladat megoldásához.

A bővítések lehetséges alternatívái:

- címnövelés (pl. 256x1 bit→512x1 bit);
- szóhosszúság növelés (pl. 256x1 bit→256x2 bit);
- vegyes a fenti két lehetőség együttes használata (256x1 bit→512x2 bit).

A memóriák bővítését megkönnyíti, hogy a memória tokokat ellátják CE (CHIP ENABLE) chip engedélyező, T.S. (Three State) három állapotú, valamint O.C. (Open Collector) nyitott kollektoros kimenetekkel.

A bővítési lehetőségek kivitelezése:

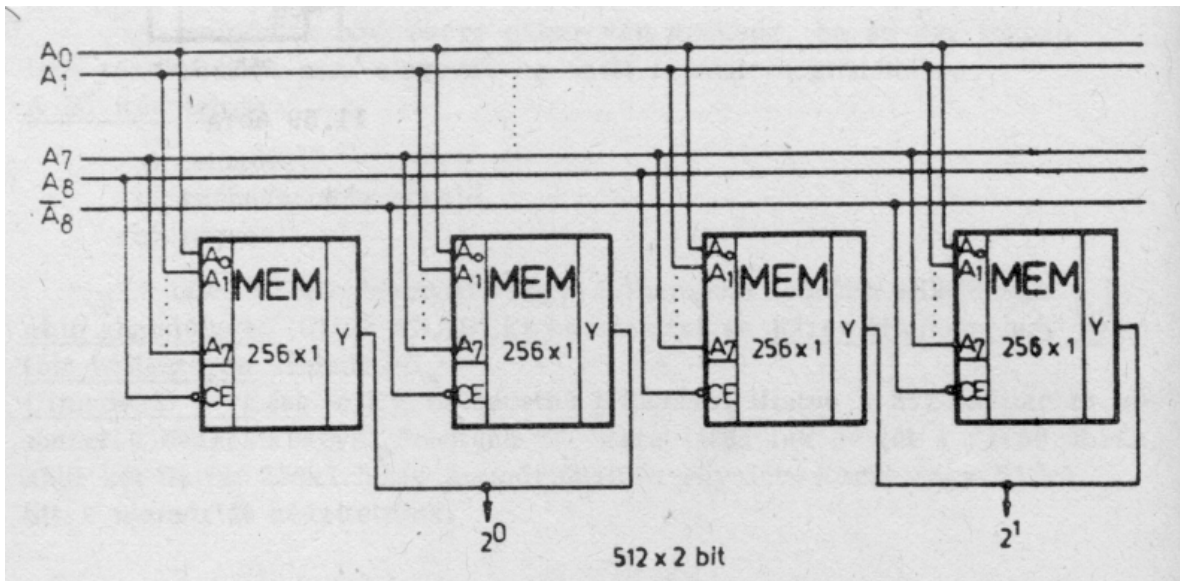
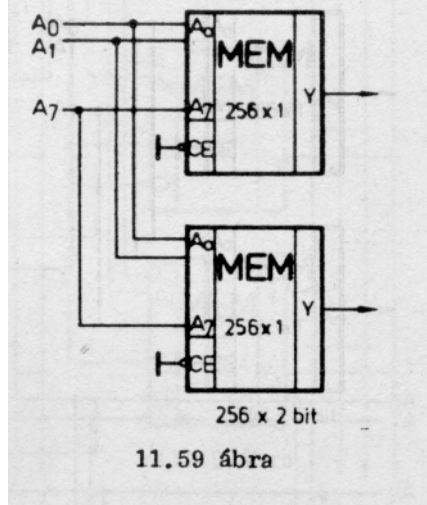
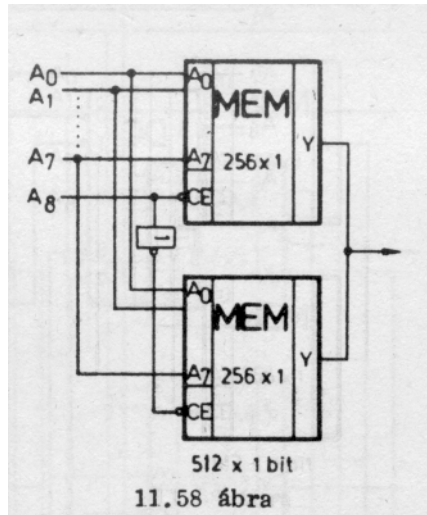
- **címnövelés esetében**, a fenti példát tekintve, 2 db 256x1 bites memóriára van szükség, az $A_0...A_7$ címvezetékeket ki kell egészíteni az 512x1 bit előállításához egy A_8 és egy inverteren keresztül előállított A_8 negált címvezetékekkel, ezt követően az A_8 címvezeték az első memória CE lábára csatlakozik, az A_8 negált címvezeték pedig, a második memória CE lábára kerül, a címvezetékeket párhuzamosítjuk és a T.S. kimeneteket összekötjük;
- **szóhosszúság növeléskor**, a fenti példát nézve, 2 db 256x1 bites memóriára van szükség, a címvezetékeket párhuzamosítjuk és a CE bemeneteket engedélyezzük;
- **vegyes bővítés esetében**, a fenti példának megfelelően 4 db 256x1 bites memóriát kell felhasználnunk, a címvezetékeket párhuzamosítjuk, illetve kibővítjük A_8 és A_8 negált címvezetékekkel, majd alkalmazzuk a címnövelés esetében leírtakat és ezt követően 2-2 db memória T.S. kimeneteit összekötjük.

Nagyobb címnövelő bővítés esetében, célszerű DC (dekódoló) áramkört használni. Ebben az esetben a minterm generátor kimenetein megvalósított mintermek, a chip szelektálást végzik, azaz a memória tokok CE (CHIP ENABLE) lábaira csatlakoznak. A dekódoló áramkör bemeneteire, pedig a kibővített címvezetékek kerülnek.

Az alábbi ábrák:

- címnövelő;
- szóhosszúság növelő;
- vegyes

bővítési alternatívákra mutatnak példákat.



7. Programozási feladatok 8086 mikroprocesszorral

Feladat: a paralel port 2. lábán (DATA0) elhelyezett LED aktivizálása:

```
MOV DX, 37A (a 378 címen lévő paralel port inicializálása Cím+2 írható/olvasható)
MOV AL, 04 (az adatvonalak kimenetek, a 2. bit értéke logikai „1”)
OUT DX, AL
MOV DX, 378 (a paralel port címe)
MOV AL, FE (a kimenetek invertáltak, a DATA0 lábán elhelyezett LED működtetése)
OUT DX, AL
```

Feladat: Tartály töltés/ürítés megvalósítása 8086 μ p-ral.

Z1 szelep tölt E1-ig, majd Z2 tölt E2-ig ezután Z3 szelepen keresztül leeresztés E0-ig és ez 13x-szor ismétlődik.

Az INTEL 8255 PIO MODE „0”-ás üzemmódjában a portok bitkiosztását szemlélteti az alábbi részletezés:

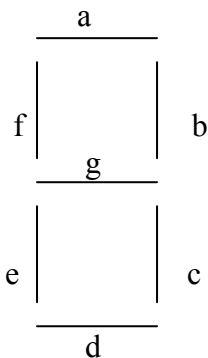
A port (kimenet): 0.,1.,2. biten Z1, Z2, Z3 szelepek

B port (bemenet): 0.,1.,2. biten E0, E1, E2 érzékelők

C port (kimenet):0.,1.,2.,3.,4.,5.,6. bitek a hétszegmenses kijelző (a,b,c,d,e,f,g) szegmenseit tartalmazza.

Vezérlő byte meghatározása:

		A	C _f	B	C _a		
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	0	0	1	0: 82h



t(töltés):

ü(ürítés):

Program:

```

MOV DX, 0303 (PIO konfigurálása)
MOV AL, 82
OUT DX, AL
MOV BL, 0D (13x tölt)
cím 6: MOV AL, 78 (t kijelzése)
MOV DX, 0302
OUT DX, AL
MOV AL, 01 (Z1 nyit)
MOV DX, 0300
OUT DX, AL
MOV DX, 0301
cím: IN AL, DX (?E1)
AND AL, 07
CMP AL, 03
JZ cím1
JMP cím
cím 1: MOV DX, 0300
MOV AL, 02 (Z2 nyit)
OUT DX, AL
MOV DX, 0301
cím 2: IN AL, DX (?E2)
AND AL, 07
CMP AL, 07
JZ cím3
JMP cím2
cím 3: MOV DX, 0302 (ü kijelzése)
MOV AL, 1D
OUT DX, AL
MOV DX, 0300
MOV AL, 04 (Z3 nyit)
OUT DX, AL
MOV DX, 0301
cím 4: IN AL, DX (?E0)
AND AL, 07
CMP AL, 00
JZ cím5
JMP cím4
DEC BL
JNZ cím6
cím 5: MOV DX, 0302 (0 kijelzése)
MOV AL, 3F
OUT DX, AL
MOV DX, 0300 (szelepek zárása)
MOV AL, 00
OUT DX, AL

```

8. Általános kérdések a mikroprocesszorok tárgyköréből

Az alábbiakban kérdések és programozási feladatok láthatók a mikroprocesszorok című fejezetből.

Alapvető és egyben minimális szintű kérdések, a mikroprocesszorok tárgyköréből

A kérdések a főiskolai szintű villamos (nappali 3 éves képzés és levelező képzés), az informatikus (nappali 5 éves képzés), valamint a kohász automatizálási szakirányos (nappali 5 éves képzés) hallgatói számára készültek.

1. Milyen alapvető fő funkcionális egységekből épül fel a mikroszámítógép?
2. Sorolja fel a BUS rendszert és magyarázza meg, hogy milyen célt szolgál?
3. Rajzolja fel a 8085 mikroprocesszor regisztermodelljét!
4. Rajzolja fel a 8086 mikroprocesszor regisztermodelljét!
5. Milyen célt szolgálnak a mikroprocesszor HOLD és HLDA lábai?
6. Mit ért DMA alatt?
7. Hogyan határozza meg a 8255 PIO MODE „, MODE 0”-ás üzemmódjában, a vezérlő byte-ot, ha **A** port kimenet, **B** port bemenet, C_a = kimenet és C_f = bemenet. Adja meg a vezérlő byte-ot hexadecimálisan és binárisan is!
8. Milyen alapvető memória típusokat ismer és sorolja fel a memória bővítésének lehetőségeit!
9. Részletesen ismertesse és magyarázza meg a 8085 mikroprocesszor CMP utasítás csoport működését!
10. Mit ért FETCH alatt? Rajzolja meg a 3000h címen kezdődő LXI H, 2B3Ch utasítás végrehajtási modelljét!
11. Mi a különbség a szubrutin és a makró között?
12. Rajzoljon MR-nek megfelelő részletes végrehajtási modell!
13. Adja meg az RST 5.5 megszakítás címét hexadecimálisan!
14. Rajzoljon általánosan végrehajtási modellt, az STAX rp utasításnak megfelelően!
15. Hány byte a STACK igénye tíz egymásba ágyazott szubrutinnak?
16. Magyarázza meg a PCHL utasítást!
17. Hogyan működik a 8085 mikroprocesszor esetében a decimális korrekció és erre milyen utasítás szolgál?

18. Sorolja fel a 8086 mikroprocesszor flag-jeit!
19. Sorolja fel a 8085 mikroprocesszor flag-jeit!
20. Milyen műveletet hajt végre az XCHG utasítás?
21. A DCX rp utasítás melyik flag-et befolyásolja?
22. Mit ért izolált I/O alatt?
23. Rajzoljon végrehajtási modelleket a PUSH rp, valamint a POP rp utasítások működéseinek megfelelően!
24. Hogyan és milyen címeken tárolódik a Stack-ben a PC azon tartalma (301Fh), ahol majd a főprogramnak folytatódnia kell (a szubrutin meghívása után, futása alatt, a RET utasítás végrehajtásáig)? A program elején a Stack, az LXI SP, 20C2h utasítással lett inicializálva!
25. Mit ért LIFO alatt?
26. Magyarázza meg a memóriában leképzett I/O fogalmát!
27. Sorolja fel a feltétel nélküli és a feltételes szubrutinból való visszatéréshez szükséges utasításokat (8085)!
28. Rajzoljon végrehajtási modellt, az XRA M utasítás működésének megfelelően!
29. Mit ért a bit maszkolása alatt?
30. Milyen utasítás alkalmazásával olvas be a 8085 mikroprocesszor esetében, az akkumulátorba a memóriából?
31. Milyen célt szolgál az INSTRUCTION REGISTER (IR)?
32. Ismertesse a CMA, STC és CMC utasításokat!
33. Hogyan működtetik az EI, DI utasítások az INTE F.F.-ot?
34. Írja fel a feltételes vezérlésátadó utasítások képzéséhez szükséges flag-ek kombinációit (0 és 1 esetekre: Z, C_y, P, S, X₅) 8085 mikroprocesszor esetében!
35. Mit ért *Direkt, Regiszter, Regiszter indirekt* címzési módok alatt?
36. Mire szolgál az ALU?
37. Rajzoljon részletes végrehajtási modellt, a 3000h címen kezdődő LHLD 20C0h utasításnak megfelelően!
38. Mit ért WAIT állapot alatt?

39. Sorolja fel általánosan egy mikroprocesszor belső funkcionális egységeit!
40. A tárolás milyen módjaira alkalmasak a RAM és ROM memóriák, valamint a rövidítések mit jelentenek?
41. Rajzoljon végrehajtási modellt, a 3000h címen kezdődő LDA 2B3Ch utasításnak megfelelően!
42. Ismertesse az r_1-r_2 , r-M és M-r közötti adatmozgató utasításokat!
43. Rajzoljon részletes végrehajtási modellt, a 3000h címen kezdődő SHLD 1B2Ch utasításnak megfelelően!
44. Sorolja fel a 8085 mikroprocesszor feltétel nélküli és a feltételes ugró utasításait!
45. Sorolja fel a 8085 mikroprocesszor feltétel nélküli és a feltételes szubrutin hívó utasításait!
46. Magyarázza meg a 8085 mikroprocesszor NOP és a HALT utasításait!
47. Rajzoljon részletes végrehajtási modellt, a 3000h címen kezdődő ORI 80h utasításnak megfelelően!
48. Sorolja fel és magyarázza meg a működését, a 8085 mikroprocesszor rotáló utasításainak!
49. Milyen utasítással valósítja meg a 8085 mikroprocesszor a 16 bites összeadást, majd ismertesse az utasítás működését!
50. Ismertesse a 8085 mikroprocesszor I/O utasításait és rajzolja le általánosan mindkét esetben a végrehajtási modellt!

9. Programozási feladatok mikroprocesszorokkal

Programozási feladatok INTEL 8085, 8086-8088 mikroprocesszorokra

A programozási feladatok a főiskolai szintű villamos (nappali 3 éves képzés és levelező képzés), az informatikus (nappali 5 éves képzés), valamint a kohász automatizálási szakirányos (nappali 5 éves képzés) hallgatói számára készültek.

1. Írjon programot 0-9-ig, a STIBITZ/AIKEN kódátalakító realizálására! Bemenet az INPUT 0 port 4 db kapcsolója (0, 1, 2, 3. bit), kimenet az OUTPUT 1 port 4 db LED-je (0, 1, 2, 3. bit).
2. Készítsen RS F.F.-ot mikroprocesszorral! Bemenet az INPUT 2 port 2 db nyomógombja (0, 1. bit), kimenet az OUTPUT 1 port 2 db LED-je (0, 1. bit).
3. Írjon programot az alábbi komparátor megvalósítására! Olvasson be folyamatosan adatokat az INPUT 1 portról! Ha a beolvasott adat = 6, akkor az OUTPUT 1 port 1., ha a beolvasott adat < 6, akkor a 0., valamint ha a beolvasott adat > 6 a 2. biten elhelyezkedő LED világítson.
4. Működtessen a PC párhuzamos portjának 2. bitjén egy LED-et kimenetként! A PC paralel portjának címe: 378h. Ez az Adress + 2 (MODE 2) üzemmódban valósítható meg. Ebben az üzemmódban a 3. bit értéke 1. Figyelem a kimenetek invertáltak!
5. Készítsen decimális előreszámlálót mikroprocesszorral 0-90-ig! Az órajel frekvenciája változtatható legyen, az INPUT 1 port-ról. A kijelzés az OUTPUT 0 porton történjen.
6. Készítsen fix frekvenciájú órajel alkalmazásával, 9-0-ig STIBITZ kódú visszaszámlálót! A kijelzést az OUTPUT 1 porton valósítsa meg!
7. Írjon programot, ami az OUTPUT 1 port 0. bitjén elhelyezkedő LED állapotát megváltoztatja, az INPUT 2 port 0. bitjén lévő nyomógomb minden negyedik megnyomására!
8. Készítsen 8 bites jobbra léptető SHIFT regisztert mikroprocesszorral! Az információt az INPUT 0 port 0. bitjén lévő kapcsoló aktivizálása szolgáltassa. A kijelzés az OUTPUT 1 port-on keresztül történjen meg. Az órajel frekvenciáját az INPUT 2 port 4. biten lévő RC oszcillátor szolgáltassa. A léptetés a 4. bit logikai „1” állapota esetében történjen.
9. Készítsen hanggenerátort a 4 bites GRAY kódnak megfelelően! A 4 bites GRAY kódot az AM-ben helyezze el! A kijelzés az OUTPUT 6 portra kerüljön, valamint a megszólaló hangok közötti sebesség változtatható legyen az INPUT 4 porton keresztül.
10. Készítsen 4 bites JOHNSON kódú visszaszámlálót mikroprocesszorral! A kijelzés az OUTPUT 1 porton legyen. A számlálás az INPUT 2 port 0. Bitjén elhelyezkedő nyomógomb logikai „1” helyzetében működjön.
11. Töltsön fel egy tartályt Z_1 szelepen keresztül X_k szintig, majd ezt követően a Z_2 és Z_3 szelepeken keresztül X_f folyadékszintig! Ez követően végezzen ürítést a Z_4 és Z_5 szelepeken keresztül. A szelepek a 8255 PIO PA porton (0, 1, 2, 3, 4. bit), az érzékelők pedig a PB portra (0, 1, 2. bit) kerültek elhelyezésre. A PIO PC portja közömbös. A PIO

MODE 0-ás üzemmódban legyen, ahol a parancsregiszter címe: 303h, az A port címe: 300h, valamint a B port címe: 301h.

12. Folyamatosan szorozza össze az INPUT 0 és az INPUT 1 portokon beállított adatokat 0-90-ig! Az eredményt jelezze ki decimálisan az OUTPUT 0 porton!
13. Készítsen 4 bites AIKEN kódú előreszámlálót mikroprocesszorral! A számlálás az INPUT 2 port 0. bitjén lévő nyomógomb logikai „1” helyzetében történjen. A kijelzést az OUTPUT 1 portra küldje.
14. Készítsen JK F.F.-ot mikroprocesszorral! Bemenet INPUT 2 port 2 db nyomógomb (0, 1. bit), kimenet az OUTPUT 1 port 2 db LED-je (0, 1. Bit). Az órajelet az INPUT 2 port 5. bitjén elhelyezkedő RC oszcillátor szolgáltatassa. A működés az 5. bit logikai „1” helyzetében valósuljon meg.
15. Írjon programot 16 bites futófény megvalósítására! A kijelzés folyamatosan történjen az OUTPUT 1, ill. az OUTPUT 2 portokon keresztül. A futási sebesség legyen változtatható, amit az INPUT 1 porton lehessen beállítani.

10. Minta zárthelyi feladatok

Az alábbiakban közölt feladatsor a II. minta zárthelyi, a mikroprocesszorok témakörből, a III. évfolyam informatikus hallgatói részére

Digitális rendszerek II. Zh.

Informatikus hallgatók részére III. évfolyam

Kidolgozási idő: 60 perc

1. Milyen utasítás használatos 8085 mikroprocesszor esetében, a decimális korrekció elvégzésére és milyen esetben? Magyarázza meg részletesen az utasítás működését!
2. Mit ért FETCH alatt és rajzoljon végrehajtási modellt, a 3000h címen kezdődő LHLD 20C0h utasításnak megfelelően!
3. Mit jelent a WAIT állapot?
4. Rajzolja le a 8086 mikroprocesszor regiszter és Flag modelljét!
5. Készítsen programot 8086 mikroprocesszorral, ami az alsó és a középső érzékelők állapotát vizsgálja! Abban az esetben, ha mindkettő logikai „1” helyzetben van, akkor olvasson be adatot az AL regiszterbe, a 301h címről! Az érzékelők a 0-adik és az 1-ső biteken kerültek elhelyezésre. Az érzékelők a 302h címen helyezkednek el.
6. Rajzolja le és ismertesse a 8085 mikroprocesszor regiszter és Flag modelljét!
7. Sorolja fel a típusait és ismertesse működését, a 8085 mikroprocesszor komparáló utasításainak!
8. Készítsen 8085 mikroprocesszorral RS F.F.-ot! Bemenet 2 db nyomógomb (E2h port 0-adik és 1-ső bit), kimenet 2 db LED (C1h port 0. és 1-ső bit)!
9. Készítsen 8085 mikroprocesszorral 4 bites STIBITZ kódú visszaszámológót 9-0-ig!
Az órajelet az E2h port 4-ik bitjén lévő RC oszcillátor szolgáltatassa!
Memóriatáblázatot a program nem tartalmazhat!
10. Mit ért DMA alatt?

11. INTEL 8085 mikroprocesszor utasításkészlete

INTEL 8085 mikroprocesszor utasításkészlete, hexadecimális kódkészlettel

Jelölésmagyarázat:

- d_8 : byte;
- d_{16} : d byte (double byte);
- A_8 : adr_8 : 1 byte-os cím (8 bites cím);
- A_{16} : adr_{16} : 2 byte-os cím (16 bites cím).

Adatmozgató utasítások

MVI

1. A, byte: 3E
2. B, byte: 06
3. C, byte: 0E
4. D, byte: 16
5. E, byte: 1E
6. H, byte: 26
7. L, byte: 2E
8. M (H,L), byte: 36

LXI

1. B, d byte: 01
2. D, d byte: 11
3. H, d byte: 21
4. SP, d byte: 31

LDA adr_{16} : 3A

STA adr_{16} : 32

LDAX

1. B: 0A
2. D: 1A

STAX

1. B: 02
2. D: 12

LHLD adr_{16} : 2A

SHLD adr_{16} : 22

XCHG: EB

PCHL: E9

LDHI byte: 28

LDSI byte: 38

LHLX: ED

SHLX: D9

MOV

1. A, A: 7F
2. A, B: 78
3. A, C: 79
4. A, D: 7A
5. A, E: 7B
6. A, H: 7C
7. A, L: 7D
8. A, M (H,L): 7E
9. B, A: 47
10. B, B: 40
11. B, C: 41
12. B, D: 42
13. B, E: 43
14. B, H: 44
15. B, L: 45
16. B, M (H,L): 46
17. C, A: 4F
18. C, B: 48
19. C, C: 49
20. C, D: 4A
21. C, E: 4B
22. C, H: 4C
23. C, L: 4D
24. C, M (H,L): 4E
25. D, A: 57
26. D, B: 50
27. D, C: 51
28. D, D: 52
29. D, E: 53
30. D, H: 54
31. D, L: 55
32. D, M (H,L): 56
33. E, A: 5F
34. E, B: 58
35. E, C: 59
36. E, D: 5A
37. E, E: 5B
38. E, H: 5C
39. E, L: 5D
40. E, M (H,L): 5E
41. H, A: 67
42. H, B: 60
43. H, C: 61

- 44. H, D: 62
- 45. H, E: 63
- 46. H, H: 64
- 47. H, L: 65
- 48. H, M (H,L): 66
- 49. L, A: 6F
- 50. L, B: 68
- 51. L, C: 69
- 52. L, D: 6A
- 53. L, E: 6B
- 54. L, H: 6C
- 55. L, L: 6D
- 56. L, M (H,L): 6E
- 57. M (H,L), A: 77
- 58. M (H,L), B: 70
- 59. M (H,L), C: 71
- 60. M (H,L), D: 72
- 61. M (H,L), E: 73
- 62. M (H,L), H: 74
- 63. M (H,L): L: 75

Beviteli/kiviteli utasítások

IN adr₈: DB

OUT adr₈: D3

Aritmetikai utasítások

ADD

- 1. A: 87
- 2. B: 80
- 3. C: 81
- 4. D: 82
- 5. E: 83
- 6. H: 84
- 7. L: 85
- 8. M (H,L): 86

ADI byte: C6

ADC

- 1. A: 8F
- 2. B: 88
- 3. C: 89
- 4. D: 8A
- 5. E: 8B
- 6. H: 8C
- 7. L: 8D
- 8. M (H,L): 8E

ACI byte: CE

SUB

1. A: 97
2. B: 90
3. C: 91
4. D: 92
5. E: 93
6. H: 94
7. L: 95
8. M (H,L): 96

SUI byte: D6

SBB

1. A: 9F
2. B: 98
3. C: 99
4. D: 9A
5. E: 9B
6. H: 9C
7. L: 9D
8. M (H,L): 9E

SBI byte: DE

DAD

1. B: 09
2. D: 19
3. H: 29
4. SP: 39

DSUB: 08

DAA: 27

Logikai utasítások

ANA

1. A: A7
2. B: A0
3. C: A1
4. D: A2
5. E: A3
6. H: A4
7. L: A5
8. M (H,L): A6

ANI byte: E6

ORA

1. A: B7
2. B: B0
3. C: B1
4. D: B2
5. E: B3
6. H: B4
7. L: B5
8. M (H,L): B6

ORI byte: F6

XRA

1. A: AF
2. B: A8
3. C: A9
4. D: AA
5. E: AB
6. H: AC
7. L: AD
8. M (H,L): AE

XRI byte: EE

CMA: 2F

STC: 37

CMC: 3F

Komparáló utasítások**CMP**

1. A: BF
2. B: B8
3. C: B9
4. D: BA
5. E: BB
6. H: BC
7. L: BD
8. M (H,L): BE

CPI byte: FE

Rotáló utasítások

RLC: 07

RRC: 0F

RAL: 17

RAR: 1F

ARLH: 10

RDEL: 18

Inkrementáló utasítások

INR

1. A: 3C
2. B: 04
3. C: 0C
4. D: 14
5. E: 1C
6. H: 24
7. L: 2C
8. M (H,L): 34

INX

1. B: 03
2. D: 13
3. H: 23
4. SP: 33

Dekrementáló utasítások

DCR

1. A: 3D
2. B: 05
3. C: 0D
4. D: 15
5. E: 1D
6. H: 25
7. L: 2D
8. M (H,L): 35

DCX

1. B: 0B
2. D: 1B
3. H: 2B
4. SP: 3B

STACK műveletekkel kapcsolatos utasítások

PUSH

1. B: C5
2. D: D5

3. H: E5
4. PSW: F5

POP

1. B: C1
2. D: D1
3. H: E1
4. PSW: F1

XTHL: E3

SPHL: F9

Egyéb utasításcsoport

EI: FB

DI: F3

NOP: 00

HLT: 76

SIM: 20

RIM: 30

Speciális megszakítások:**RST**

- | | | | |
|----|-------|------|---------|
| 1. | 0: C7 | 0000 | |
| 2. | 1: CF | 0008 | |
| 3. | 2: D7 | 0010 | |
| 4. | 3: DF | 0018 | |
| 5. | 4: E7 | 0020 | |
| 6. | 5: EF | 0028 | RST 5.5 |
| 7. | 6: F7 | 0030 | RST 6.5 |
| 8. | 7: FF | 0038 | RST 7.5 |

RSTV: CB 0040

Feltétel nélküli vezérlésátadó utasítások

JMP adr₁₆: C3

CALL adr₁₆: CD

RET: C9

Feltételes vezérlésátadó utasítások**JZ** adr₁₆: CA**JNZ** adr₁₆: C2**JC** adr₁₆: DA**JNC** adr₁₆: D2**JPE** adr₁₆: EA**JPO** adr₁₆: E2**JM** adr₁₆: FA**JP** adr₁₆: F2**JX₅** adr₁₆: FD**JNX₅** adr₁₆: DD**CZ** adr₁₆: CC**CNZ** adr₁₆: C4**CC** adr₁₆: DC**CNC** adr₁₆: D4**CPE** adr₁₆: EC**CPO** adr₁₆: E4**CM** adr₁₆: FC**CP** adr₁₆: F4**RZ**: C8**RNZ**: C0**RC**: D8**RNC**: D0**RPE**: E8**RPO**: E0

RM: F8

RP: F0

12. Kidolgozott szoftverek 8085 és 8086 (8088) mikroprocesszorokra

Az alábbi szoftverek, az Mcs85.exe file futtatását követően, betöltés után azonnal működtethetők egy lépéses üzemmódban. Az Mcs85.exe file indítása után, a „**Belépés a monitor programba**” menüpont kiválasztását követően, a szoftverek betöltése az alábbi szintaktika szerint történik:

L 1000 C:\ comp.

A fenti parancs végrehajtása után, az 1000h címre betöltődik a comp (comparátor) nevű program, a gyökér könyvtárból. A program lépésenkénti futtatása a:

J 1000

parancs begépelése után, az **F9** funkció billentyű aktivizálásával történik. Azok a szoftverek amik az **AM** (Adat Memória: LXI H, 3000h)-át használják 3000h címen, a memória feltöltésének folyamata, pl. 8 bites JOHNSON kóddal, a lentiekben megadottak szerint történik:

>3000 00 01 03 07 0F 0E 0C 08.

8085 mikroprocesszor emulátorra írt és futtatható szoftverek:

1. Comp;
2. Counter;
3. Counter1;
4. Counter2;
5. Counter3;
6. Counter4;
7. szorzás;
8. futóf;
9. kodatal;
10. Tank.

A szoftverek eredményeinek kijelzése a 00h portra kerülnek. Az INPUT port címe szintén 00h. A Counter, 1, 2, 3, 4, (Számláló) programok időzítéseinek lehetséges alternatívái: 1. fix; 2. változtatható (00h portról beolvasható adat); 3. nyomógomb megnyomására működtethető (00h port 0-adik bitjén elhelyezkedő nyomógomb).

A 8086 (8088) mikroprocesszorok assembly programozása Win98 operációs rendszerig bezárólag, DOS üzemmódban működtethetők. A Turbo Pascal 6.0, ill. a Borland Pascal 7.0 programnyelvek megfelelő installálásukat követően, futtatásuk lehetővé teszi, a 8086 (8088) mikroprocesszorok assembly szintű programozását.

A Windows NT, ME, 2000 és XP operációs rendszerek alatt, viszont futtatható a FreePascal programnyelv, ami szintén biztosítja a fenti mikroprocesszorok assembly nyelvű programozási lehetőségét. A programnyelv letöltéséhez és futtatásához, a DOS (GO32v2extender), dosw321010full.zip (40, 939 900 MB) file kicsomagolását követő install.exe, a <http://www.freepascal.org> Internet címen keresztül érhetőse nyújt segítséget.

A **Borland DELPHI 6** programnyelven írt PC paralel port I/O műveleteket kezelő és működtethető szoftverek futtatásához, meg kell hívni az **io.dll file**-t, az alábbiak szerint:

```

program Project1;

{$APPTYPE CONSOLE}

uses
  SysUtils;

var
  Port: Word;
  Data: Byte;

procedure PortOut (Port: Word; Data: Byte); stdcall; external 'C:\io\io.dll';

begin
  { TODO -oUser -cConsole Main : Insert code here }

  Port:=888;
  Data:=3;

  PortOut (Port, Data);

  writeln;
  writeln;
  Writeln ('      Kimeneti adat a paralel porton = ',Data);
  readln
end.

```

8086 (8088) mikroprocesszor programozási példa, PC paralel port kimeneti üzemmódjának a beállítására:

```

Procedure paralel; assembler;
asm
  MOV DX, $37A ( Cím+2 a PC paralel portja kimenet)
  MOV AL, 04h (A DATA0...DATA7 kimenetek)
  OUT DX, AL
  MOV DX, $378 ( a PC paralel portjának címe)
  MOV AL, $FE ( a PC paralel port DATA0 bit értéke „1”, a kimenetek invertáltak)
  OUT DX, AL
end;

```

8086 (8088) mikroprocesszorra írt assembly szoftverek Pascal programnyelv alá:

1. Keverőtartály.pas;
2. Ioncserélő.pas;
3. Marógép.pas.

Megjegyzés:

Az oktatási anyag együttes áttekintése a mikroprocesszor emulátor programok használatával szinkronban, megkönnyíti mind a hardver, mind pedig a programozási technika elsajátítását, valamint lehetőséget nyújt a hallgatók számára otthoni körülmények között, az un. „**Virtuális Mikroprocesszor Laboratórium**” megteremtéséhez.

Dr. Gárdus Zoltán Ph.D.
egyetemi adjunktus
Villamosmérnöki Intézet
Automatizálási Tanszék

Irodalomjegyzék

- [1] Ajtonyi István: DIGITÁLIS RENDSZEREK
MISKOLCI EGYETEM, 2002
- [2] Ajtonyi István: Vezérléstechnika I.-II.
Tankönyvkiadó, Budapest, 1991
- [3] Gál T: Programozható logikák BME
Tankönyvkiadó, 1994
- [5] INTEL SDK-85 System Design Kit User's Manual
INTEL Corporation, 1978
- [6] INTEL SDK-86 System Design Kit User's Manual
INTEL Corporation, 1978
- [7] XILINX The Programmable Gate Array Data Book San Jose, California 95124, 1991
- [8] XILINX The Programmable Logic Data Book San Jose, California 95124, 1994
- [9] László József: A PC hardver programozása
Valós és védett módban
PASCAL és ASSEMBLY nyelven
ComputerBooks
Budapest, 2003