

P r o g r a m o v a n i e
v jazyku symbolických adries na PMD 85
(určené pre účastníkov kurzov výpočtovej techniky)

vypracoval : Ing.Milan Žofaj

recenzoval : Ing.Peter Kupkovič

O b s a h

Úvod.....	2
1. Programovanie mikroprocesora 8080A.....	3
2. Inštrukčný súbor mikroprocesora 8080A.....	4
3. MRS pre PMD 85-2.....	10
4. Podprogramy modulu knižnice systému MRS.....	15
5. Podprogramy monitora PMD 85-2.....	19
6. Súbor príkladov /zadania/.....	24
Príloha 1 : Tabuľka ASCII kódov.....	26
Príloha 2 : Abecedný zoznam inštrukcií.....	27
Príloha 3 : Súbor inštrukcií podľa kódov.....	28
Príloha 4 : Súbor programov.....	30
7. Záver	
Zoznam použitej literatúry	

Úvod

Táto publikácia je určená užívateľom mikropočítača PMD 85-2, účastníkom kurzov výpočtovej techniky ako aj lektorom. Poskytuje im popis programových prostriedkov potrebných na programovanie v jazyku symbolických adries a súbor vytvorených programov.

Obsah je rozdelený do viacerých kapitol. V nich užívateľ nájde stručný popis programátorského modelu mikroprocesora MHB 8080A, popis inštrukcií uvedeného mikroprocesora. Časť publikácie je venovaná systému MRS, jeho podprogramom a taktiež podprogramom monítora PMD 85-2.

V poslednej časti je súbor zadaných príkladov a k nim vytvorených programov v JSA. Väčšina zadaní príkladov obsahuje aj námety na doplnenie alebo zmenu programu, čo je možné využiť pri samostatnej práci najmä na kurzoch výpočtovej techniky.

Snahou autora bolo vytvoriť dielko, ktoré by bolo vhodnou pomôckou nielen začínajúcim záujemcom o programovanie v JSA, ale všetkým účastníkom kurzov.

1. Programovanie mikroprocesora MHB 8080 A

Na ovládanie programovania mikropočítačov netreba poznať všetky podrobnosti ich zapojenia, prípadne vnútorné zapojenia vlastného mikroprocesora. Obyčajne stačí programátorský model mikroprocesora, ktorý definuje pracovné registre, príznakové bity a spôsob spracovania programu.

Pred programovaním sa najskôr oboznámime s programátorským modelom mikroprocesora 8080 A, ktoré má programátor k dispozícii, sú schematicky naznačená na obr. Sedem registrov s dĺžkou 1 slabiky (8 bitov) tvorí pole pracovných registrov, čo je vlastne zápisníková pamäť pre medzivýsledky alebo konštanty. Registre sa symbolicky označujú veľkými písmenami A, B, C, D, E, H, L.

8	8
A	F
B	C
D	E
H	L
SP	
PC	

Register A má medzi ostatnými registrami celkom výnimočné postavenie, pretože môže pracovať ako zhromažďovač (akumulátor). Mikroprocesor umožňuje určité spracovanie údajov s dvojnásobnou dĺžkou, ktoré sa zachováávajú v dvojiciach registrov. Tieto sa označujú písmenom registra, v ktorom je uložená horná slabika (vyššie rády). Máme tri registrové dvojice:

dvojica B (B, C)

dvojica C (D, E)

dvojica E (H, L)

Osobitné postavenie má dvojica H, ktorá pracuje ako zhromažďovač pri spočítaní dvojíc registrov. Mikroprocesor 8080 A má ešte päť preklápacích obvodov združených do registra podmienok F, hodnoty ktorých sa nastavujú podľa výsledku niektorých operácií (príznakové bity, ako aj stavové alebo podmienkové bity, anglicky flags). Preklápacie obvody sú označené takto:

S - znamienkový bit

Z - bit indikujúci nulový obsah

AC - pomocný prenosový bit

CY - (hlavný) prenosový bit

PE - paritný bit s párnou paritou

Príznaky možno používať pri vetvení programu. Dvojica registrov (A, F) tvorí stavové slovo programu PSW (Processor Status Word).

Čítač inštrukcií, ktorý sa označuje PC (program counter), je samostatný šestnásťbitový register, ktorý po zakončení každej inštrukcie ukazuje miesto operačnej pamäti, odkiaľ sa má zobrať nasledujúca inštrukcia. Pretože sú v súbore inštrukcie s dĺžkou 1 až 3 slabiky, získa sa adresa nasledujúcej inštrukcie. Pri skokových inštrukciách sa do čítača inštrukcií dosadí adresa, kde sa má pokračovať, a to nepodmienenne, alebo v závislosti od splnenia určitej podmienky. Pri zahájení programu treba do čítača inštrukcií nastaviť začiatočnú adresu programu.

Kapacita pamäti adresovateľná čítačom inštrukcií je $2^{16} = 65\,536$ slabík. Adresovaný priestor je ohraničený od adresy 0000H až do FFFFH.

Posledný register, ktorý sa nazýva ukazovateľ zásobníka (stack pointer - SP) je tiež 16-bitový. Udáva adresu, na ktorej sa naplní osobitná zóna v pamäti označovaná ako zásobník alebo zásobníková pamäť. Zásobník sa používa na dočasné odloženie údajov a možno ho umiestniť v ľubovoľnej oblasti pamäti.

Podrobnejší popis mikroprocesora MHB 8080A je v [1].

2. Inštrukčný súbor mikroprocesora MHB 8080 A

Inštrukčný súbor mikroprocesora MHB 8080 A obsahuje 78 typov inštrukcií, pričom počet inštrukcií jedného druhu je rozličný. Súbor tvorí spolu 244 inštrukcií (z 256 možných operačných kódov je 12 nevyužitých). Súbor inštrukcií možno rozdeliť do piatich základných skupín:

- presunové inštrukcie
- aritmetické inštrukcie
- logické inštrukcie
- inštrukcie vetvenia programov
- inštrukcie pre prácu so zásobníkom a špeciálne inštrukcie

Pri opise inštrukcií budú použité znaky a symboly:

- r - označuje registre A, B, C, D, E, H, L
- M - označuje nepriamu adresáciu pomocou dvojice registrov HL
- rp - označuje dvojicu registrov B, D, H, PSW, SP
- a - označuje 16-bitovú adresu
- d - označuje 8-bitový údaj
- p - označuje 8-bitovú adresu vstupného alebo výstupného portu
- n - celé číslo z rozsahu $0 \leq n \leq 7$
- <x> - obsah prvku x
- <<x>> - obsah pamäťového miesta určeného adresou x
- <-- - presun
- <=> - vzájomná výmena
- :: - symbol pre porovnanie obsahov

Podrobnejší popis inštrukcií je v literatúre [1], [2].

2.1. Presunové inštrukcie

Tieto inštrukcie patria k najpoužívanejším inštrukciám. Používajú sa na presun údajov a adres pred a po vykonaní iných inštrukcií. Umožňujú naplnenie registrov, registrových párov, presun údajov z registra do registra, z pamäti do registra alebo opačne.

MOV r1,r2 r1 <-- <r2>

Inštrukcia urobí presun obsahu registra r2 do r1. V prípade použitia symbolu M namiesto r1 resp. r2 pôjde o presun údajov z registra do pamäti alebo opačne.

MVI r,d r <-- d

Priame naplnenie registra r údajom d, ktorý je priamo v druhej slabike inštrukcie.

STA a a <-- <A>

Odpamätanie obsahu akumulátora do pamäti na adresu a.

LDA a A <-- <a>

Naplnenie akumulátora obsahom pamäťového miesta s adresou a.

STAX rp <<rp>> <-- <A>

Odpamätanie obsahu akumulátora na adresu danú registrovým párom BC alebo DE.

LDAX rp A <-- <<rp>>

Naplnenie akumulátora obsahom pamäťového miesta s adresou v reg. páre BC alebo DE.

LHLD a L <-- <<a>>, H <-- <<a+1>>

Naplnenie páru H a L obsahom pamäťového miesta s adresou priamo v inštrukcii.

SHLD a a <-- <L>, a+1 <-- <H>

Odpamätanie H a L na pamäťové miesto s adresou priamo v pamäti.

LXI rp,a rp <-- a

Naplnenie registrového páru 16-bitovou adresou, ktorá je priamo v inštrukcii.

XCHG <HL> <=> <DE>

Výmena obsahov registrových párov HL a DE.

2.2. Aritmetické inštrukcie

Používajú sa pre aritmetické operácie s údajmi v registroch a v pamäti, pričom všetky inštrukcie ovplyvňujú príznakové bity.

ADD r A <-- <A> + <r>

Aritmetický súčet akumulátora a registra r (prípadne M). Výsledok ostane v A.

ADI d A <-- <A> + d

Priame sčítanie obsahu akumulátora a údajov v druhej slabike inštrukcie.

ADC r A <-- <A> + <r> + CY

Sčítanie s prenosovým bitom.

ACI d $A \leftarrow A + d + CY$

Priame sčítanie s prenosovým bitom.

SUB r $A \leftarrow A - \langle r \rangle$

Odčítanie obsahu registra r od akumulátora. Výsledok ostane v A.

SUI d $A \leftarrow A - d$

Priame odčítanie údajov v druhej slabike inštrukcie od obsahu akumulátora.

SBB r $A \leftarrow A - \langle r \rangle - CY$

Odčítanie s prenosom.

SBI d $A \leftarrow A - d - CY$

Priame odčítanie s prenosom.

INR r $r \leftarrow r + 1$

Zvýšenie obsahu registra r resp. M/HL o 1.

INX rp $rp \leftarrow \langle rp \rangle + 1$

Zvýšenie obsahu registrového páru o 1. Príznakové bity sa nenastavujú.

DCR r $r \leftarrow r - 1$

Zmenšenie obsahu registra r resp. M/HL o 1.

DCX rp $rp \leftarrow \langle rp \rangle - 1$

Zmenšenie obsahu registrového páru o 1. Príznakové bity sa nenastavujú.

DAA $A \leftarrow \text{desiatková korekcia } A$

Inštrukcia pripočíta k akumulátoru číslo 6 vtedy, ak je hodnota dolných 4 bitov akumulátora väčšia ako 9, alebo ak je po dvojkovom spočítaní nastavený bit AC (pomocný prenos). Ak po pripočítaní čísla 6 k dolným bitom je hodnota horných 4 bitov väčšia ako 9, alebo ak nastal prenos zo 7. bitu $CY=1$, potom sa pripočíta 6 k horným 4 bitom.

DAD rp $HL \leftarrow \langle HL \rangle + \langle rp \rangle$

Sčítanie obsahu registrových párov. Výsledok ostáva v HL.

CMP r $\langle A \rangle :: \langle r \rangle$

Porovnanie obsahu A s obsahom registra r alebo pamäťovej bunky M/HL. Nastavia sa príznaky:

$Z=1$ ak $A=r$
 $Z=0$ ak $A \neq r$
 $CY=1$ ak $A < r$
 $CY=0$ ak $A \geq r$

CPI d $\langle A \rangle :: d$

Porovnanie obsahu akumulátora A s údajom, ktorý je priamo v druhej slabike inštrukcie.

2.3. Logické inštrukcie

Umožňujú pracovať s logickými funkciami AND, OR, NOT, XOR, ktoré sú definované ako operácie prebiehajúce medzi dvoma slabikami údajov tak, že príslušná logická funkcia sa vykoná s každou dvojicou bitov zodpovedajúceho si rádu.

ANA r A <-- <A> AND <r>
Logický súčin.

ANI d A <-- <A> AND d
Logický súčin, obsahu akumulátora s obsahom druhej slabiky inštrukcie.

ORA r A <-- <A> OR <r>
Logický súčet.

ORI d A <-- <A> OR d
Logický súčet obsahu akumulátora s obsahom druhej slabiky inštrukcie.

XRA r A <-- <A> EXOR <r>
Funkcia nerovnoznačnosti.

XRI d A <-- <A> EXOR d
Logická funkcia nerovnoznačnosti medzi obsahom akumulátora a obsahom druhej slabiky inštrukcie.

CMA A <-- NOT <A>
Negácia akumulátora.

STC CY <-- 1
Nastavenie prenosového bitu CY na 1.

CMC CY <-- NOT <CY>
Negácia prenosového bitu CY.

RLC
Rotačný posuv doľava. Posunie všetky bity akumulátora o jedno miesto doľava, bit 7 do bitu 0 a súčasne do príznakového bitu CY.

RRC
Rotačný posuv doprava. Posunie všetky bity akumulátora o jedno miesto doprava, bit 0 sa presunie do bitu 7 a súčasne do CY.

RAL
Rotačný posuv akumulátora doľava cez CY. Posunie každý bit o jedno miesto doľava, bit 7 sa presunie do CY a obsah CY do bitu 0.

RAR

Rotačný posuv akumulátora doprava cez CY. Posunie obsah každého bitu o jedno miesto doprava, bit 0 sa posunie do CY a obsah CY do bitu 7.

2.4. Inštrukcie vetvenia programu.

JMP a PC <-- a
Nepodmienený skok na adresu a.

Jxx a
Podmienený skok. Ak je splnená podmienka xx, program pokračuje na adrese a, ak táto podmienka nie je splnená, program pokračuje na nasledujúcej adrese, teda na adrese <PC> +3. Podmienky sú tieto:

- NZ - nenulový výsledok
- Z - nulový výsledok
- NC - nevznikol prenos
- C - vznikol prenos
- PO - nepárny počet jednotiek výsledku
- PE - párny počet jednotiek výsledku
- P - kladný výsledok
- M - záporný výsledok.

CALL a SP <-- <SP> - 2, zásobník <-- <PC>, PC <-- a
Volanie podprogramu. Inštrukcia spôsobí skok na adresu a, so súčasným uložením návratovej adresy do zásobníka, na ktorej bude pokračovať po skončení podprogramu.

Cxx a
Podmienené volanie podprogramu. Ak je splnená podmienka xx, potom sa volá podprogram na adrese a, v opačnom, prípade sa pokračuje nasledujúcou inštrukciou programu. Príznaky sú také isté ako u inštrukcie Jxx.

RET PC <-- návratová adresa zo zásobníka,
 SP <-- <SP> + 2
Návrat z podprogramu. Inštrukcia spôsobí skok na adresu, ktorá je uložená na vrchole zásobníka.

Rxx
Podmienený návrat z podprogramu. Ak je splnená podmienka xx, vykoná sa návrat z podprogramu ako pri inštrukcii RET. Ak podmienka nie je splnená, vykoná sa nasledujúca inštrukcia za inštrukciou Rxx.

PCHL PC <-- <HL>
 Naplnenie počítadla inštrukcií obsahom HL.

RST n SP <-- <SP> - 2, zásobník <-- <PC>, PC <-- 8xn
 Inštrukcia spôsobí volanie podprogramu na adrese n x 8, pričom n je z rozsahu 0 až 7.

2.5. Inštrukcie pre prácu so zásobníkom a špeciálne inštrukcie.

Zásobníková pamäť je určený súvislý úsek pamäti RWM, ktorý sa používa na prechodné uloženie medzivýsledkov a adries, predovšetkým pri spracovaní prerušenia, volaní a návratov z podprogramov a pod. Vytvára sa programovými prostriedkami od zvolenej adresy (dno zásobníka) smerom k nižším adresám, vždy po dvoch slabikách. V dôsledku toho má naposledy uložený údaj do zásobníka (vrchol zásobníka), najnižšiu adresu, ktorá je obsiahnutá vo zvláštnom 16-bitovom registri SP (smerník zásobníka). To znamená, že naposledy vložený údaj sa musí vyberať ako prvý.

PUSH rp SP <-- <SP> - 2, zásobník <-- rp
 Vlož do zásobníka.

POP rp rp <-- <zásobník>, SP <-- <SP> + 2
 Vyber zo zásobníka.

SPHL SP <-- <HL>
 Naplnenie smerníka zásobníka obsahom dvojice registrov HL.

XTHL <L> <=> <<SP>>, <H> <=> <<SP + 1>>
 Výmena obsahu vrcholu zásobníka.

IN p A <-- <p>
 Vstup. Inštrukcia presunie 8 bitov údajov z určenej brány s adresou p do akumulátora.

OUT p p <-- <A>
 Výstup. Inštrukcia presunie obsah akumulátora na údajovú zbernicu a adresu p na adresovú zbernicu.

EI
 Povolenie prerušenia.

DI
 Zákaz prerušenia.

HLT
 Zastavenie procesora.

NOP
 Prázdna operácia. Používa sa na rezervovanie miesta v pamäti, prípadne v časových slučkách.

3.1. EDITOR

```

INI - inicializuje pracovnú oblasť a pripraví ju na vstup nového
      textu
LN - posuv, ukazovateľa po texte
     LN+X - o X riadkov dopredu
     LN-X - o X riadkov späť
     LN=X - nastaví ukazovateľ na X-tý riadok
           (ako X sa môže použiť aj návestie v texte)
     LN-0 - ukazovateľ na prvý riadok
     LN+0 - ukazovateľ na posledný riadok
     LN=0 - ukazovateľ na posledný editovaný riadok
SAV MMMMMMMMM - uloženie zdrojového textu na MGF
                (MMMMMMMM - meno)

```

LOA MMMMMMMM - nahratie zdrojového textu z MGF

POSTUP: V obrazovkovom režime nastav kurzor na riadok, za ktorý chceš vsunúť nahraný zdrojový text. Ak sa pred nahratím pracovná oblasť inicializuje, text sa nahrá od začiatku.

MRS - návrat do riadiaceho modulu MRS

ALD - spusti sa postupnosť volaní jednotlivých modulov:

ASM, LNK, DBG.

Ak niektorý nájde chybu, uskutoční sa návrat späť do EDITORA a ukazovateľ sa nastaví na chybný riadok.

Obrázokový režim:

Do tohto režimu prejde EDITOR po vykonaní príkazov INI, RST alebo LN.

Rozdelenie riadku:

- 7 znakov - pole návestia
- 5 znakov - pole inštrukcie
- 16 znakov - pole adresy
- 16 znakov - pole poznámky

POLE NÁVESTIA je vyplnené medzerami alebo obsahuje návestie začínajúce písmenom o max. dĺžke 6 znakov.

POLE INŠTRUKCIE môže obsahovať INŠTRUKCIE 8080 alebo PSEUDOINŠTRUKCIE:

- ORG - nastavuje veľkosť čítača adres (v poli adresy môže byť výraz)
- DB - ukladá do pamäti hodnoty výrazov z adresnej časti. Tie-to hodnoty majú dĺžku najviac 1 bytu. Ako výraz môže byť reťazec v apostrofoch.
- DS - rezervuje pamäť
- DW - ako DB, len hodnota výrazu zaberá 2 byty
- EQU - priradí návestiu hodnotu danú výrazom, ktorý je v poli adresy
- ENT - popisuje návestie ako bod vstupu (ENTRY POINT)
- EXT - popisuje návestie ako vonkajší bod (EXTERNAL POINT) pre prácu s knižnicou
- END - pseudoinštrukcia pre ukončenie prekladu

POLE ADRESY môže obsahovať prvky predpísané pre inštrukciu alebo pseudoinštrukciu alebo výrazy, ktoré sa vyhodnocujú počas prekladu. Výraz v systéme MRS môže mať tvar:

- SYMBOL
- SYMBOL + SYMBOL
- SYMBOL - SYMBOL
- SYMBOL
- >SYMBOL (hodnota vyšších 8 b zo 16 b hodnoty)
- <SYMBOL (hodnota nižších 8 b zo 16 b hodnoty)

SYMBOL môže mať jeden z tvarov:

- NÁVESTIE

- DECIMÁLNA KONŠTANTA ; reťazec číslíc 0-9
- HEXADECIMÁLNA KONŠTANTA ; reťazec číslíc 0 - 9 a písmen A - F, zakončený písmenom H, Ak začína písmenom, musí byť pred ním číslíca 0
- ZNAKOVÁ KONŠTANTA ; ľubovoľný reťazec znakov v apostrofoch

POLE POZNÁMKY môže obsahovať ľubovoľné znaky. Ak začína riadok ';' (bodkočiarka), ide o poznámku. Prekladač poznámku ignoruje.

Zvláštne inštrukcie pre riadenie prekladu začínajú '*':

- *A - povolenie výpisu
- *E - zakázanie výpisu - assembler pri zakázanom výpise vypisuje len chybné riadky
- *L - počas prekladu sa vytvára výpis na tlačiareň
- *CXXXX
 - XXXX=0 - prekladač nevytvára výsledný kód v pamäti, len výpis prekladu
 - XXXX=adresa HEXA - prekladač ukladá výsledný kód od tejto adresy, platnosť zruší ORG

V obrazovkovom režime sa využívajú klávesy nasledujúcim spôsobom:

SPACE	- skoky kurzora po znakoch
-->, <--	- štandardne
SH -->, SH <--	- skoky po zónach
DEL, INS	- štandardne
SH DEL	- vymazanie riadku
SH INS	- opakovanie predchádzajúceho riadku
SH <--	- o obr. späť (o 23 riadkov)
SH -->	- o obr. dopredu
SH \	- riadok s kurzorom sa stane prvým na obrazovke
SH END	- návrat do príkazového režimu

Chybové hlásenia

- MEM FULL - oblasť určená pre text je plná. Treba text skrátiť, alebo vymazať moduly z knižnice (pamäť má kapacitu asi 4000 riadkov)
- ST FULL - zaplnená pamäť určená pre návestia (max. 255)

3.2. A S S E M B L E R

Spustí sa príkazom: ASM MMMMMM (MMMMMM - meno programu).

Preložený text sa uloží do knižnice. Ak je program bez mena, vznikne modul s prázdny menom a každý ďalší preklad ho vymaže. Ak sa počas prekladu vyskytnú chyby, modul sa do knižnice nezaradí.

Prekladač vymazáva, nepoužité návestia. Ak sa nájde znak *A (povolenie výpisu), vypisuje sa celý preklad, až kým nepríde znak *E (zákaz výpisu). Výpis možno prerušiť klávesom WRK a opäť spustiť klávesom SH WRK, Kláves STOP s okamžitou platnosťou ukončí činnosť ASSEMBLERu. Po preklade sa do dialógového riadku vypíše počet chýb; vyžaduje sa potvrdenie klávesom EOL.

Chybové oznamy, ktoré ukončia okamžite činnosť ASSEMBLERa:

NO END - nedefinovaný koniec programu (chyba END)
 MEM FULL - pokus uložiť program tam, kde nesmie byť (obe chyby okamžite ukončia činnosť ASSEMBLERa)

Chyby zistené počas prekladu:

U - v adresnej časti riadka sa vyskytlo návestie, ktoré nebolo definované t.j. nie je v poli návestia. Indikuje sa pri každom výskyte.
 M - v poli návestia sú dve rovnaké návestia ; indikuje sa každý ich výskyt.
 D - v adresnej časti je dvakrát definované návestie (už označené chybovým kódom M => inf. charakter).
 R - výraz v adresnej časti prekročil hodnotu 255
 E - návestie popísané ako EXTERNAL sa vyskytlo ako súčasť výrazu, alebo je v adresnej časti inštrukcie, ktorá pripúšťa len 8-bitovú hodnotu.

Zobrazenie chybného riadku : 1. znak - chybový kód
 2.-5. Znak - číslo chybného riadku

3.3. L I N K E R

Spojovací modul nájde ku každému, vonkajšiemu návestiu zodpovedajúce vstupné návestie a priradí mu hodnotu. Ak v knižnici ne-nájde žiadané návestie, vypíše počet chýb. Chybové hlásenie treba potvrdiť klávesom EOL. LINKER netreba spustiť vždy, ale len ak sú v programe vonkajšie návestia. Spúšťa sa príkazom LNK.

3.4. D E B U G G E R

Spúšťa sa príkazom DBG. Po vyvolaní sa prihlási výpisom:

```
X PPPP INŠTRUKCIA AAAA BBBB DDDD HHHH SSSS
  PC              AF   BC   DE   HL   SP
```

Stlačením SH & INS sa zabezpečí alebo zakáže opis dialógového riadku do pracovnej oblasti obrazovky.

X...Status - informácia o stave lad. programu (obyčajne medzera).
Pri prvom spustení lad. programu sú tieto hodnoty náhodné.

Činnosť DEBUGGERa sa riadi klávesami:

- R - pôjde o vkladanie údajov do registrového paru. Očakáva sa kláves, ktorý určuje registrový pár a hodnota vkladateľných údajov.
 - P - Program Counter (RP)
 - A - Accumulátor, Flag (RA)
 - B - reg. pár BC (RB)
 - D - reg. pár DE (RD)
 - H - reg. pár HL (RH)
 - S - Stack Pointer (RS)
- S - (STEP) krokovanie programu (vykoná sa len 1 inštrukcia)
- I - (INSERT BREAK) vloženie hodu prerušenia do programu. Očakáva sa adresa. Po spustení programu sa prerušenie indikuje písmenom B.
- O - vymazanie vloženého BREAKu
- W - (WINDOW) pôjde o definovanie pamäťových zón.
Očakávajú sa znaky:
 - M - (MEMORY) do dialógového riadku sa vypíšu začiatkové a koncové adresy štyroch chránených zón, do ktorých sa nesmie zapisovať. Ďalším klávesom sa určí, ktorú adresu chceme zmeniť (kláves 0 až 7).
 - P - (PC) vypíšu sa zóny, v ktorých sa môže pohybovať PC. Zmeny sa robia vyššie popísaným spôsobom.
- T - (TRACE) ladený program sa spustí, pričom sa vypisujú hodnoty všetkých registrov a kontrolujú sa všetky obmedzenia v zónach. Pri prekročení obmedzenia sa program zastaví. Kláves STOP umožňuje okamžité zastavenie.
- N - (NO TRACE) to isté ako T-režim, len sa nevypisujú hodnoty registrových párov.
- G - (GO) príkaz na spustenie programu ; kontroluje sa len BREAK POINT, koniec sa signalizuje písmenom E.
- D - (DUMPING) výpis pamäti po 8 byte v riadku od adresy, ktorá sa napíše za D.
Posúvanie výpisu: <-- - späť
--> - dopredu
EOL - návrat do DEBUGGERu
- M - (MEMORY) mod. pamäti od zadanej adresy; posuvy podobne ako v D-režime. Umožňuje meniť obsah 12 bytov, ktoré sa vypíšu do dialógového riadku.
- Q - (QUIT) návrat do modulu MRS.
- C - program sa krokuje, ale podprogramy sa vykonávajú v G-režime.

4. Podprogramy modulu knižnice systému MRS

Modul knižnice sa prihlási : LIB>

Očakávajú sa príkazy :

LST - výpis modulov v knižnici. Riadiace klávesy:
 WRK - zastavenie výpisu
 SH WRK- opätovné pokračovanie výpisu
 STOP - návrat do modulu LIB
 DEL - vymazanie posledného nahratého programu z knižnice (základný modul nie je možné, vymazať).
 SAV MMMMMMMM - vloženie knižnice na magnetofón pod menom MMMMMMMM
 LOA MMMMMMMM - nahranie novej knižnice z magnetofónu

Základný modul knižnice SYSMOD obsahuje podprogramy systému MRS. Pred použitím sa na začiatku programu označia použité podprogramy pomocou pseudoinštrukcie:

EXT meno podprogramu

Zoznam podprogramov:

MEMORY - obsahuje adresu prvej voľnej pamäťovej bunky za knižnicou.

LASTLIN - obsahuje adresu prvej obsadenej pamäťovej bunky za voľnou pamäťou - umožňuje to prácu s dynamickou pamäťou. Ak užívateľ v programe vykoná inštrukcie:

EXT MEMORY
 EXT LASTLIN
 LXI D, MEMORY
 LHLD LASTLIN

tak získa v registroch <DE> a <HL> hranice použiteľnej pamäti - bežiaci program má k dispozícii pamäť od <DE>+1 do <HL>-1.

DCBN - konverzia reťazca decimálnych číslíc na 16-bitovú binárnu hodnotu

VSTUP: <HL> = adresa reťazca

VÝSTUP: <HL> = adresa konca reťazca (reťazec končí ľubovoľným znakom okrem číslíc 0-9)

<DE> = konvertovaná binárna hodnota

<A> = ukončovací znak

Mení sa: <AF>, <DE>, <HL>

HXBN - konverzia reťazca hexadecimálnych číslíc na 16-bitovú binárnu hodnotu

VSTUP: <HL> = adresa reťazca

VÝSTUP: <HL> = adresa konca reťazca (reťazec končí ľubovoľným symbolom okrem 0-9 a A-F)

<DE> = konvertovaná binárna hodnota

Mení sa: <AF>, <DE>, <HL>

- BNHX - konverzia 8-bitovej hodnoty na dve hexadecimálne číslice
 VSTUP: <A> = konvertované číslo
 <HL> = adresa pamäti, kde sa má výsledok konverzie uložiť
 VÝSTUP: <HL> = adresa pamäti za uloženou dvojicou
 Mení sa: <AF>, <HL>
- FBNDC - konverzia 16-bitovej hodnoty na reťazec decimálnych číslic
 VSTUP: <BC> = konvertované číslo
 <HL> = adresa pamäti, kam sa má uložiť výsledok
 VÝSTUP: <HL> = adresa pamäti za reťazcom decimálnych číslic, číslo je uložené v pamäti ako reťazec ASCII znakov
 Mení sa: <AF>, <DE>, <HL>
- INKEY - čítanie jedného znaku z klávesnice s automatickým opakovaním (autorepeat)
 VÝSTUP: <A> = kód stlačeného klávesu
 Mení sa: <AF>
- SCAN - ohmatanie klávesnice. Modul zisti, či je stlačený niektorý kláves
 VÝSTUP: <A> = 0 ak nie je stlačené nič
 <A> = ASCII kód stlačeného klávesu
 Mení sa: <AF>
- CLEAR - vymazanie obrazovky
 Mení sa: <AF>
- PIXEL - rozkreslenie znaku na obrazovke
 VSTUP: <A> = ASCII kód znaku
 <HL> = adresa videopamäte, kde má byť uložený pravý dolný roh rozkresľovaného znaku. Na vlastné rozkreslenie znaku využíva modul MONITORa.
 <C03AH> = jas. úroveň, blikanie
 Mení sa: <AF>
- OUTCHR - vypisuje znak na pozíciu označenú kurzorom a kurzor posunie doprava. Na konci riadku prejde na nový riadok, na konci obrazovky posunie obrazovku hore. OUTCHR správne ošetrí tieto riadiace znaky:
 83H - posun jazdca vľavo <--
 85H - posun jazdca vpravo -->
 89H - ukončenie riadku a prechod na nový riadok
 8DH - zmaže obrazovku a jazdec vľavo hore
 VSTUP: <A> = ASCII kód znaku
 Mení sa: <AF>

- CURSOR - adresa pamäti, ktorá obsahuje políciu kurzora na obrazovke pre modul OUTCHR. Sú to dva byty, v prvom sa uchová číslo stĺpca (0-43), v druhom číslo riadku (0-22).
- OUTBUF - výpis jedného riadku na obrazovku.
 VSTUP: <H> = číslo riadku (0-22), kam sa má vypísať daný text. Vypisovaný text je uložený od adresy LINE a vypíše sa vždy celý počet znakov (44).
 Mení sa: <AF>
- LINE - adresa pamäti, ktorá obsahuje text vypisovaný modulom OUTBUF, táto pamäť má veľkosť 44 znakov, využíva ju aj systém MRS vrátane ladiaceho programu, s čím treba počítať pri ladení programov.
- BEEP - vydá sa zvukový signál
 Mení sa: <AF>
- MRS - práca s dialógovým riadkom, modul vypíše požadovaný text do dialógového riadku, načíta vstupný text a podľa požiadavky ho analyzuje v trojznakové príkazy. Vyvolaním tohto modulu má užívateľ k dispozícii všetky funkcie popísané v popise práce s modulom MRS, To znamená, že na editovanie textu možno použiť klávesy -->, <--, INS, DEL. Vstup textu vždy ukončuje kláves EOL.
 VSTUP: <A> = číslo pozície v dialógovom riadku, od ktorej sa má text vypisovať (0-43)
 <HL> = adresa vypisovaného textu
 = dĺžka vypisovaného textu, ak B=0, text sa nevypisuje
 <C> = maximálna dĺžka čítaného textu včítane EOL. Ak <C>=0 text sa nečíta (požiadavka len na výpis)
 <DE> = adresa tabuľky trojznakových príkazov. Ak <DE>=0, načítaný text sa neanalyzuje. V tabuľke nasledujú vždy za sebou trojznaková skratka príkazu a adresa obslužného programu. Tabuľka je zakončená binárnou 0. Ak je požadovaná táto analýza, modul nevráti riadenie, kým užívateľ nenapíše riadok, ktorý začína prípustným príkazom.
 VÝSTUP: = počet načítaných znakov bez EOL
 načítaný reťazec je ukončený bin. 0
 <DE> = adresa načítaného reťazca, pokiaľ bola požadovaná analýza trojznakových príkazov, ukazuje <DE> na prvý znak za príkazom
 <HL> = ak bola požadovaná analýza trojznakových príkazov obsahuje adresu príslušného podprogramu
 Mení sa: <AF>, <BC>, <DE>, <HL>

- MRS2 - umožní opraviť text načítaný modulom MRS
 musí mu predchádzať volanie modulu MRS s čítaním reťazca
 VSTUP: použije informácie získané pri práci modulu MRS.
 Modul MRS2 sa musí volať inštrukciou: JMP MRS2
 VÝSTUP: rovnaký ako pri module MRS
 Mení sa: <AE>, <BC>, <DE>, <HL>
- SENTRY - Vyhľadá výstupný bod v knižnici. Dá sa využiť, ak napr.
 jeden program produkuje dáta, ktoré má spracovať iný modul
 knižnice. Pomocou SENTRY môže daný program nájsť tabuľku v
 inom module, kam má uložiť vstupné dáta.
 VSTUP: <HL> = adresa reťazca, ktorý udáva požadované
 vstupné návěstie.
 VÝSTUP: <S> = 1 - chybné zadanie meno
 <Z> = 1 - v knižnici nie je také meno
 <HL> = adresa mena v knižnici. Na tejto adrese je 6
 znakov meno a ďalšie dva znaky adresa, ktorú
 toto meno označuje (napr. adresa tabuľky,
 ktorej meno je popísané ako ENT)
 Mení sa: <AF>, <BC>, <DE>, <HL>
- GETMEM - Určí symbol. Ak symbol je návěstie, potom sa práca modulu
 GETMEM týka naposledy prekladaného modulu a jeho využitie
 je obdobné ako modulu SENTRY.
 VSTUP: <HL> = adresa reťazca označujúceho symbol
 VÝSTUP: = -1 - syntakticky chybný symbol
 = 0 - symbol je konštanta a <DE> = jej hodnota.
 >1 - symbol je návěstie
 <HL> = adresa znaku za symbolom
 Ak symbol je návěstie, potom
 = dĺžka návěstia v znakoch
 <C> = kód návěstia pre modul VSTOAS
 Mení sa: <AF>, <BC>, <DE>, <HL>
- VSTOAS - V spojení s modulom GETMEM umožňuje nájsť ľubovoľnú adresu
 pamäti označenú návěstím v naposledy prekladanom zdrojovom
 texte.
 VSTUP: <C> = kód návěstia získaný v module GETMEM
 VÝSTUP: <HL> = adresa návěstia. Na tejto adrese je 6 znakov,
 meno návěstia a ďalšie dva znaky obsahujú adresu v preloženom
 programe, ktorá je označená daným návěstím.
 Mení sa: <AF>, <HL>
- PNTCHR - Vytlačenie jedného znaku na tlačiareň.
 VSTUP: <A> = kód tlačeneho znaku.
 Mení sa: <AF>
 Prechod na nový riadok: CR (0DH), LF (0AH)

5. PODPROGRAMY MONITORA PMD 85-2

Modul MONITOR mikropočítača PMD 85-2 poskytuje celý rad užitočných podprogramov. V tejto kapitole sú uvedené bežne používané podprogramy. Každý z nich je označený menom, popisom jeho funkcie, volacou adresou, požadovaným vstupom, výstupom a informáciou o tom, ktoré obsahy registrov sa zmenia.

START	<p>Studený štart systému.</p> <p>ADRESA : 8000 H</p> <p>VSTUP : 0</p> <p>VÝSTUP : 0</p> <p>MENÍ : obsahy všetkých registrov</p>
ROLL	<p>Posun obrazu smerom hore.</p> <p>ADRESA : 808E H</p> <p>VSTUP : 0</p> <p>VÝSTUP : 0</p> <p>MENÍ : obsahy všetkých registrov</p>
BEPUK	<p>Invertovanie zvukového bitu a žltej LED diódy na klávesnici.</p> <p>ADRESA : 80D8 H</p> <p>VSTUP : 0</p> <p>VÝSTUP : 0</p> <p>MENÍ : AF</p>
HEX1	<p>Konverzia ASCII znaku (s adresou v HL) na 16-bitové číslo.</p> <p>ADRESA : 80DF H</p> <p>VSTUP : <HL> = prevádzaný znak v ASCII kóde</p> <p>VÝSTUP : CY = 1 nie je 16-ová číslica</p> <p style="padding-left: 100px;">CY = 0 je 16-ová číslica a <A> = 0...15</p> <p>MENÍ : AF, HL</p>
HEX	<p>Konverzia ASCII znaku v A na 16-bitové číslo.</p> <p>ADRESA : 80E0 H</p> <p>VSTUP : <A> = prevádzaný ASCII znak</p> <p>VÝSTUP : CY = 1 nie je 16-ová číslica</p> <p style="padding-left: 100px;">CY = 0 je 16-ová číslica a <A> = 0...15</p> <p>MENÍ : AF, HL</p>
PAIRIN	<p>Mení 2 ASCII znaky na dve 16-ové číslice v A</p> <p>ADRESA : 80F7 H</p> <p>VSTUP : <HL> = adresa textu (2 ASCII znaky)</p> <p>VÝSTUP : CY = 1 nie sú 16-ové číslice</p> <p style="padding-left: 100px;">CY = 0 sú 16-ové číslice</p> <p style="padding-left: 100px;"><A> = 0...FF H</p> <p style="padding-left: 100px;"><HL> = ukazuje na druhý znak</p> <p>MENÍ : AF, HL, B</p>

ADRIN Mení text (4 ASCII znaky) na 16-ové číslice.
 ADRESA : 8109 H
 VSTUP : <HL> = text (4 ASCII znaky)
 VÝSTUP : CY = 1 v texte nie sú 16-ové číslice
 CY = 0 v texte sú 16-ové číslice
 <DE> = 0...FFFF H
 <HL> = adresa 4. znaku
 MENÍ : AF, HL, DE, B

PREVO1 Vypíše 2 16-ové číslice v A na obrazovku.
 ADRESA : 8125 H
 VSTUP : <A> = číslo 0...FF H
 C03F = nastaviť adresu kurzoru
 VÝSTUP : 0
 MENÍ : AF, B

PREVO0 Mení 1 16-ovú číslicu na ASCII znak.
 ADRESA : 8888 H
 VSTUP : <A> = 16-ova číslica v bitoch 0 až 3
 VÝSTUP : <A> = ASCII znak číslica
 MENÍ : AF

PREVO2 Mení dve 16-ové číslice v A na dva ASCII znaky.
 ADRESA : 813B H
 VSTUP : <A> = 0...FF H
 <HL> = adresa buffera (na 2 ASCII znaky)
 VÝSTUP : <HL> = adresa 2. znaku
 MENÍ : AF, HL, B

MODBEP Zapnutie/vypnutie pípnutia pri stlačení klávesu.
 ADRESA : 8167 H
 VSTUP : 0
 VÝSTUP : 0
 MENÍ : AF

BECLR Vypnutie pískania.
 ADRESA : 81EE H
 VSTUP : 0
 VÝSTUP : 0
 MENÍ : AF

MODCHR Zvýšenie kódu znaku nad kurzorom o 1.
 ADRESA : 846F H
 VSTUP : 0
 VÝSTUP : 0
 MENÍ : AF

XCHCUR Vykreslenie/zmazanie kurzora.
 ADRESA : 8481 H
 VSTUP : 0
 VÝSTUP : 0
 MENÍ : AF, BC, HL

INKLAV Načítanie znaku z klávesnice s čakaním na stlačenie.
 ADRESA : 84A1 H
 VSTUP : 0
 VÝSTUP : <A> = kód stlačeného klávesu
 MENÍ : AF

PRTOUT Napíše znak na pozíciu danú kurzorom.
 ADRESA : 8500 H
 VSTUP : <A> = ASCII znak (1CH=CLEAR, 0DH=CR, 0AH=LF)
 <C03E>&<C03F> = pozícia kurzora
 <C03A> = kód farby (00H, 40H, 80H, 0C0H)
 VÝSTUP : 0
 MENÍ : AF

MODNEG Zapnutie/vypnutie inverzného výpisu + mazanie displeja.
 ADRESA : 859F H
 VSTUP : 0
 VÝSTUP : 0
 MENÍ : AF, BC, DE, HL

ERASE Mazanie displeja.
 ADRESA : 85A7 H
 VSTUP : 0
 VÝSTUP : 0
 MENÍ : AF, BC, DE, HL

CERASE Mazanie časti displeja.
 ADRESA : 85B3 H
 VSTUP : <HL> = adresa kurzora
 <DE> = adresa konca najvyššej mazanej linky + 1
 = počet mazaných liniek
 VÝSTUP : 0
 MENÍ : AF, BC, DE, HL

RPOINT Test rozsvietenia bodu.
 ADRESA : 85E6 H
 VSTUP : <C170> = X-ová súradnica 0...255
 <C172> = Y-ová súradnica 0...255
 VÝSTUP : Z=1, <A> = 0 bod nesvieti
 Z=0, <A> = 1 bod svieti
 MENÍ : AF, BC, DE, HL

EDIT Vykonanie zmeny v dialógovom riadku.
 ADRESA : 8800 H
 VSTUP : <C134 H> = kód ASCII znaku alebo riadiaci znak
 VÝSTUP : zmenený buffer dialóg. riadku je vypísaný na
 displej
 MENÍ : AF

MODCOL Cyklická zmena atribútov obrazovky displeja.
 ADRESA : 8C6B H
 VSTUP : 0
 VÝSTUP : 0
 MENÍ : AF

STOP Test stlačenia klávesu STOP.
 ADRESA : 8C74 H
 VSTUP : 0
 VÝSTUP : Z=1, <A>=03H kláves STOP je stlačený
 Z=0, <A>=40H kláves STOP nie je stlačený
 MENÍ : AF

POINT Vykreslenie bodu v danom grafickom móde.
 ADRESA : 8C7D H
 VSTUP : <C170> = súradnica X 0...255
 <0172> = súradnica Y 0...255
 <C03A> = kód farby (00H, 40H, 80H, 0C0H)
 <C1F3> = mód kódu (0A8H=negácia bodu,
 0AFH=nulovanie, 0B0H=nastavenie bodu)
 VÝSTUP : 0
 MENÍ : AF

POSPOINT Výpočet adresy bodu v displeji.
 ADRESA : 8C94 H
 VSTUP : <L> = súradnica X 0...255
 <H> = súradnica Y 0...255
 VÝSTUP : <HL> = adresa bytu na displeji
 = bitová maska bodu 1, 2, 4, 8, 10H, 20H
 MENÍ : AF, BC, DE, HL

INPOL Lineárny interpolátor.
 ADRESA : 8CD0 H
 VSTUP : <C170> = súradnica X počiatočného bodu
 <0172> = súradnica Y počiatočného bodu
 <0173> = súradnica X koncového bodu
 <C174> = súradnica Y koncového bodu
 <C03A> = kód farby (00H = plný jas
 40H = polojas
 80H = bliká plný jas
 C0H = bliká polojas)
 <C1F4> = mód kódu (A8H = negácia
 AFH = nulovanie
 B0H = nastavenie)
 VÝSTUP : 0
 MENÍ : AF, BC, DE, HL

BINBCD Mení dvojkové číslo na 2 BCD číslice.
 ADRESA : 8E73 H
 VSTUP : <A> = dvojkové číslo
 VÝSTUP : <A> = 2 BCD číslice
 MENÍ : AF, B, HL

PIO Nastaví obvod 8255 do režimu tlačiarne.
 ADRESA : 8F4FH
 VSTUP : 0
 VÝSTUP : 0
 MENÍ : A

6. Súbor príkladov (zadania)

1. Príklad

Program nuluje časť pamäti RWM. Vyžaduje ako vstupné argumenty adresu začiatku nulovanej zóny v HL a dĺžku nulovanej zóny v B. Doplňte úlohu tak, aby vstupné hodnoty boli zadávané z klávesnice a zobrazované na displeji.

2. Príklad

Program vymení obsah dvoch pamäťových zón o dĺžke 256 byte. Vyžaduje vstupy: HL - adresa začiatku zdrojovej zóny
 DE - adresa začiatku cieľovej zóny
 C - dĺžka zóny
 Doplňte úlohu tak, aby vstupy boli zadávané z klávesnice a aby sa vypísali hodnoty oboch zón pred výmenou a po výmene na obrazovku.

3. Príklad

Program presunie obsah časti pamäti na iné miesto. Vyžaduje vstupy: HL - začiatok zdrojovej zóny
 DE - začiatok cieľovej zóny
 B - dĺžka zóny
 Doplňte úlohu tak, aby vstupy boli zadávané z klávesnice a aby sa vypísali hodnoty cieľovej zóny pred presunom a po presune na obrazovku.

4. Príklad

Program hľadá maximum v súbore čísel. Vyžaduje vstupy:

HL - adresa začiatku súboru, B - dĺžka súboru.

Doplňte úlohu so zadávaním vstupov z klávesnice a s výpisom súboru a maxima na obrazovku.

5. Príklad

Program roluje časť obrazovky doľava. V podstate ide o modifikovanie obsahu videopamäte, posuv jej časti doľava.

Úlohu doplňte tak, aby časť obrazovky rolovala doprava a iná časť doľava.

6. Príklad (a, b, c, d)

Programy pracujú s obsahom videostránky a zároveň ovládajú výpis na obrazovku. Ide o mazanie, invertovanie, zmenu atribútov obrazovky alebo jej časti a o prenos časti obrazovky na iné miesto.

Zmeňte v úlohe vstupné argumenty a overte správnu činnosť programu. Doplňte úlohy tak, aby vstupné argumenty boli zadávané z klávesnice.

7. Príklad

Program vynásobí obsahy dvoch registrov. Metóda násobenia je založená na algoritme postupného pripočítavania.

8. Príklad

Program generuje melódiu podľa tabuľky. V podstate, pôjde o využívanie 3. bitu portu PC, na ktorý je pripojený reproduktor. Striedavým nastavovaním a nulovaním tohto bitu sa generuje určitý tón, ktorého výška závisí od toho, ako dlho je 3. bit nastavený resp. nulovaný. Zmeňte v úlohe obsah tabuľky tak, aby bola generovaná iná melódia.

9. Príklad

Program generuje melódiu podľa tabuľky. Využíva podprogram monitora BEEP. Tabuľka obsahuje výšky a dĺžky tónov a musí byť ukončená ukončovacím znakom 255. Výšky tónov môžu byť: 0, 1, 2, 3 a dĺžka 0...255. V programe sa dĺžky tónov modifikujú.

Príloha 1 - Tabuľka ASCII kód

Hodnota znaku (16) (10)		Znak	Hodnota znaku (16) (10)		Znak	Hodnota znaku (16) (10)		Znak
00	0	NUL	2C	44	,	57	87	w
01	1	SOH	2D	45	-	58	88	x
02	2	STX	2E	46	.	59	89	y
03	3	ETX	2F	47	/	5A	90	z
04	4	EOT	30	48	0	5B	91	[
05	5	ENQ	31	49	1	5C	92	\
06	6	ACK	32	50	2	5D	93]
07	7	BEL	33	51	3	5E	94	^
08	8	BS	34	52	4	5F	95	_
09	9	HT	35	53	5	60	96	`
0A	10	LF	36	54	6	61	97	a
0B	11	VT	37	55	7	62	98	b
0C	12	FF	38	56	8	63	99	c
0D	13	CR	39	57	9	64	100	d
0E	14	SO	3A	58	:	65	101	e
0F	15	SI	3B	59	;	66	102	f
10	16	DLE	3C	60	<	67	103	g
11	17	DC1	3D	61	=	68	104	h
12	18	DC2	3E	62	>	69	105	i
13	19	DC3	3F	63	?	6A	106	j
14	20	DC4	40	64	@	6B	107	k
15	21	NAK	41	65	A	6C	108	l
16	22	SYN	42	66	B	6D	109	m
17	23	ETB	43	67	C	6E	110	n
18	24	CAN	44	68	D	6F	111	o
19	25	EM	45	69	E	70	112	p
1A	26	SUB	46	70	F	71	113	q
1B	27	ESC	47	71	G	72	114	r
1C	28	FS	48	72	H	73	115	s
1D	29	GS	49	73	I	74	116	t
1E	30	RS	4A	74	J	75	117	u
1F	31	US	4B	75	K	76	118	v
20	32	SP	4C	76	L	77	119	w
21	33	!	4D	77	M	78	120	x
22	34	"	4E	78	N	79	121	y
23	35	#	4F	79	O	7A	122	z
24	36	\$	50	80	P	7B	123	{
25	37	%	51	81	Q	7C	124	
26	38	&	52	82	R	7D	125	}
27	39	,	53	83	S	7E	126	~
28	40	(54	84	T	7F	127	DEL
29	41)	55	85	U			
2A	42	*	56	86	V			
2B	43	+						

Príloha 2 - Abecedný zoznam inštrukcií mikroprocesora 8080A

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

Inštrukcia	Kód	Inštrukcia	Kód	Inštrukcia	Kód	Inštrukcia	Kód
DCR A	3D	MOV C,A	4F	ORA H	B4	XRA B	A8
DCR B	05	MOV C,B	48	ORA L	B5	XRA C	A9
DCR C	0D	MOV C,C	49	ORA M	B6	XRA D	AA
DCR D	15	MOV C,D	4A	ORI d	F6	XRA E	AB
DCR E	1D	MOV C,E	4B	OUT p	D3	XRA H	AC
DCR H	25	MOV C,H	4C	PCHL	E9	XRA L	AD
DCR L	2D	MOV C,L	4D	POP B	C1	XRA M	AE
DCR M	35	MOV C,M	4E	POP D	D1	XRI d	EE
DCX B	0B	MOV D,A	57	POP H	E1	XTHL	E3

Príloha 3 - Súbor inštrukcií mikroprocesora 8080 podľa kódov

Kód	Inštrukcia	Kód	Inštrukcia	Kód	Inštrukcia	Kód	Inštrukcia
00	NOP	40	MOV B,B	80	ADD B	C0	RNZ
01	LXI B,a	41	MOV B,C	81	ADD C	C1	POP B
02	STAX B	42	MOV B,D	82	ADD D	C2	JNZ a
03	INX B	43	MOV B,E	83	ADD E	C3	JMP a
04	INR B	44	MOV B,H	84	ADD H	C4	CNZ a
05	DCR B	45	MOV B,L	85	ADD L	C5	PUSH B
06	MVI B,d	46	MOV B,M	86	ADD M	C6	ADI d
07	RLC	47	MOV B,A	87	ADD A	C7	RST 0
08	-	48	MOV C,B	88	ADC B	C8	RZ
09	DAD B	49	MOV C,C	89	ADC C	C9	RET
0A	LDAX B	4A	MOV C,D	8A	ADC D	CA	JZ a
0B	DCX B	4B	MOV C,E	8B	ADC E	CB	-
0C	INR C	4C	MOV C,H	8C	ADC H	CC	CZ a
0D	DCR C	4D	MOV C,L	8D	ADC L	CD	CALL a
0E	MVI C,d	4E	MOV C,M	8E	ADC M	CE	ACI d
0F	RRC	4F	MOV C,A	8F	ADC A	CF	RST 1
10	-	50	MOV D,B	90	SUB B	D0	RNC
11	LXI D,a	51	MOV D,C	91	SUB C	D1	POP D
12	STAX D	52	MOV D,D	92	SUB D	D2	JNC a
13	INX D	53	MOV D,E	93	SUB E	D3	OUT p
14	INR D	54	MOV D,H	94	SUB H	D4	CNC a
15	DCR D	55	MOV D,L	95	SUB L	D5	PUSH D
16	MVI D,d	56	MOV D,M	96	SUB M	D6	SUI d
17	RAL	57	MOV D,A	97	SUB A	D7	RST 2
18	-	58	MOV E,B	98	SBB B	D8	RC
19	DAD D	59	MOV E,C	99	SBB C	D9	-
1A	LDAX D	5A	MOV E,D	9A	SBB D	DA	JC a
1B	DCX D	5B	MOV E,E	9B	SBB E	DB	IN p
1C	INR E	5C	MOV E,H	9C	SBB H	DC	CC a
1D	DCR E	5D	MOV E,L	9D	SBB L	DD	-
1E	MVI E,d	5E	MOV E,M	9E	SBB M	DE	SBI d
1F	RAR	5F	MOV E,A	9F	SBB A	DF	RST 3
20	-	60	MOV H,B	A0	ANA B	E0	RPO
21	LXI H,a	61	MOV H,C	A1	ANA C	E1	POP H
22	SHLD a	62	MOV H,D	A2	ANA D	E2	JPO a
23	INX H	63	MOV H,E	A3	ANA E	E3	XTHL
24	INR H	64	MOV H,H	A4	ANA H	E4	CPO a
25	DCR H	65	MOV H,L	A5	ANA L	E5	PUSH H
26	MVI H,d	66	MOV H,M	A6	ANA M	E6	ANI d
27	DAA	67	MOV H,A	A7	ANA A	E7	RST 4

Kód	Inštrukcia	Kód	Inštrukcia	Kód	Inštrukcia	Kód	Inštrukcia
28	-	68	MOV L, B	A8	XRA B	E8	RPE
29	DAD H	69	MOV L, C	A9	XRA C	E9	PCHL
2A	LHLD a	6A	MOV L, D	AA	XRA D	EA	JPE a
2B	DCX H	6B	MOV L, E	AB	XRA E	EB	XCHG
2C	INR L	6C	MOV L, H	AC	XRA H	EC	CPE a
2D	DCR L	6D	MOV L, L	AD	XRA L	ED	-
2E	MVI L, d	6E	MOV L, M	AE	XRA M	EE	XRI d
2F	CMA	6F	MOV L, A	AF	XRA A	EF	RST 5
30	-	70	MOV M, B	B0	ORA B	F0	RP
31	LXI SP, a	71	MOV M, C	B1	ORA C	F1	POP PSW
32	STA a	72	MOV M, D	B2	ORA D	F2	JP a
33	INX SP	73	MOV M, E	B3	ORA E	F3	DI
34	INR M	74	MOV M, H	B4	ORA H	F4	CP a
35	DCR M	75	MOV M, L	B5	ORA L	F5	PUSH PSW
36	MVI M, d	76	HLT	B6	ORA M	F6	ORI d
37	STC	77	MOV M, A	B7	ORA A	F7	RST 6
38	-	78	MOV A, B	B8	CMP B	F8	RM
39	DAD SP	79	MOV A, C	B9	CMP C	F9	SPHL
3A	LDA a	7A	MOV A, D	BA	CMP D	FA	JM a
3B	DCX SP	7B	MOV A, E	BB	CMP E	FB	EI
3C	INR A	7C	MOV A, H	BC	CMP H	FC	CM a
3D	DCR A	7D	MOV A, L	BD	CMP L	FD	-
3E	MVI A, d	7E	MOV A, M	BE	CMP M	FE	CPI d
3F	CMC	7F	MOV A, A	BF	CMP A	FF	RST 7

Táto strana príručky sa nedochovala...

Táto strana príručky sa nedochovala...

Táto strana príručky sa nedochovala...

```

;
;PRIKLAD 05
;
;ROLOVANIE CASTI OBRAZOVKY DOLAVA
;
;VSTUP : <HL> - ZACIAT. ADRESA 1. RIADKU
;          <E> - POCET RIADKOV
;          <B> - POCET ZNAKOV V RIADKU
;VYSTUP: 0
;
;
ZADR EQU 0C900H
POCR EQU 9
POCZ EQU 2FH
WAITS EQU 8BCBH
STOP EQU 8C74H
;
0132 21 00C9 L00 LXI H,ZADR
0135 1E 09 MVI E,POCR
0137 06 2F L0 MVI B,POCZ
0139 0E 11 MVI C,40H-POCZ DOPLNOK DO 40H
013B 56 MOV D,M UCHOVAT 1.BYTE
013C 23 L1 INX H DALSI BYTE
013D 7E MOV A,M VYBRAT BYTE
013E 2B DCX H POSUV VLAVO
013F 77 MOV M,A POSUNUTY BYTE
0140 23 INX H DALSI BYTE
0141 05 DCR B
0142 C2 3C01 JNZ L1 OPAKUJ AZ B=0
0145 72 MOV M,D 1.BYTE POSLEDNY
0146 09 DAD B NOVY RIADOK
0147 1D DCR E
0148 C2 3701 JNZ L0 VSETKY RIADKY ?
;
014B 16 14 MVI D,20 DLZKA ONESKORENIA
014D CD CB8B CALL WAITS CAK.SLUCKA
0150 CD 748C CALL STOP TEST STOP ???
0153 C2 3201 JNZ L00
0156 C9 RET
END

```

```

;
;PRIKLAD 6a
;
;MAZANIE OBRAZOVKY ALEBO JEJ CASTI
;
;VSTUP : <HL> - HORNA LAVA ADRESA
;         <DE> - DOLNA LAVA ADRESA
;         <B>  - SIRKA MAZANEJ CASTI
;
;
HLADR EQU 0C018H
DLADR EQU 0E018H
SIRKA EQU 24
;
0132 21 18C0      LXI  H,HLADR
0135 11 18E0      LXI  D,DLADR
0138 06 18        ZAC  MVI  B,SIRKA
013A 0E 28        MVI  C,40H-SIRKA    DOPLNOK DO RIAD.
;
013C 7C          MOV  A,H            VYSSIA CAST ADR.
013D 92          SUB  D              POROVNAJ ADRESY
013E C2 4401     JNZ  MAZE          AK ROZNE, MAZ
0141 7D          MOV  A,L            NIZSIA CAST ADR.
0142 93          SUB  E              POROVNAJ ADRESY
0143 C8          RZ                  KONIEC AK ROVNE
;
0144 E6 00     MAZE ANI  0            NULUJ A
0146 77        MOV  M,A            NULUJ VIDEOBYTE
0147 23        INX  H              DALSI VIDEOBYTE
0148 05        DCR  B              ZNIZ SIRKU
0149 C2 4401   JNZ  MAZE          OPAKUJ AZ B=0
;
014C 09        DAD  B              NOVY RIAD. VRAM
014D C3 3801   JMP  ZAC            OPAKUJ
;
END

```

```

;
;PRIKLAD 6b
;
;INVERZIA OBRAZOVKY ALEBO JEJ CASTI
;
;VSTUP : <HL> - HORNA LAVA ADRESA
;         <DE> - DOLNA LAVA ADRESA
;         <B>  - SIRKA INVERTOVANEJ CASTI
;
;
HLADR EQU 0ED40H
DLADR EQU 0EF40H
SIRKA EQU 30H
;
0132 21 40ED      LXI   H,HLADR
0135 11 40EF      LXI   D,DLADR
0138 06 30        ZAC   MVI   B,SIRKA
013A 0E 10        MVI   C,40H-SIRKA    DOPLNOK DO RIAD.
;
013C 7C          MOV   A,H            VYSSIA CAST ADR.
013D 92          SUB   D            POROVNAJ ADRESY
013E C2 4401     JNZ   INVER        AK ROZNE,ZMEN !
0141 7D          MOV   A,L            NIZSIA CAST ADR.
0142 93          SUB   E            POROVNAJ ADRESY
0143 C8          RZ                  KONIEC AK ROVNE
;
0144 7E          INVER MOV   A,M            CITAJ VIDEOBYTE
0145 EE 3F       XRI   3FH            INVERTUJ BYTE
0147 77          MOV   M,A            NASTAV VIDEOBYTE
0148 23          INX   H            DALSI VIDEOBYTE
0149 05          DCR   B            ZNIZ SIRKU
014A C2 4401     JNZ   INVER        OPAKUJ AZ B=0
;
014D 09          DAD   B            NOVY RIAD. VRAM
014E C3 3801     JMP   ZAC            OPAKUJ
;
END

```

```

;
;PRIKLAD 6c
;
;ZMENA ATRIBUTOV OBRAZOVKY ALEBO JEJ CASTI
;
;VSTUP : <HL> - HORNA LAVA ADRESA
;         <DE> - DOLNA LAVA ADRESA
;         <B>  - SIRKA ZMENENEJ CASTI
;
;
HLADR EQU 0C018H
DLADR EQU 0E018H
SIRKA EQU 24
PLNY EQU 0          PLNY JAS
POLO EQU 40H        POLO JAS
BPLNY EQU 80H        BLIK PLNY JAS
BPOLO EQU 0C0H       BLIK POLOJAS
;
0132 21 18C0          LXI H,HLADR
0135 11 18E0          LXI D,DLADR
0138 06 18            ZAC MVI B,SIRKA
013A 0E 28            MVI C,40H-SIRKA    DOPLNOK DO RIAD.
;
013C 7C              MOV A,H            VYSSIA CAST ADR.
013D 92              SUB D              POROVNAJ ADRESY
013E C2 4401         JNZ ZMENA          AK ROZNE,ZMEN !
0141 7D              MOV A,L            NIZSIA CAST ADR.
0142 93              SUB E              POROVNAJ ADRESY
0143 C8              RZ                KONIEC AK ROVNE
;
0144 7E              ZMENA MOV A,M      CITAJ VIDEOBYTE
0145 EE 3F           ANI 3FH            NULUJ ATRIBUTY
0147 F6 80           ORI BPLNY         BLIK PLNY JAS
0149 77              MOV M,A           NASTAV VIDEOBYTE
014A 23              INX H              DALSI VIDEOBYTE
014B 05              DCR B              ZNIZ SIRKU
014C C2 4401         JNZ ZMENA          OPAKUJ AZ B=0
;
014F 09              DAD B              NOVY RIAD. VRAM
0150 C3 3801         JMP ZAC            OPAKUJ
;
END

```

```

;
;PRIKLAD 6d
;
;PRENOS CASTI OBRAZOVKY NA INE MIESTO
;
;VSTUP : <HL> - ADRESA LAVEHO HORNEHO ROHU
;          ZDROJ.CASTI ( ZLHR )
;          <DE> - ADRESA LAVEHO HORNEHO ROHU
;          CIEL.CASTI ( CLHR )
;          <200H>&<201H> - ADRESA LAVEHO DOL.
;          ROHU CIEL.CASTI ( CLDR )
;          <B> - SIRKA
;
;
;
CLDR EQU 0E000H
ZLHR EQU 0E018H
CLHR EQU 0C000H
SIRKA EQU 24
;
0132 21 00E0      LXI    H,CLDR
0135 22 0002      SHLD   BUFFER
0138 21 18E0      LXI    H,ZLHR
013B 11 00C0      LXI    D,CLHR
;
013E 06 18      ZNOVA   MVI    B,SIRKA
0140 0E 28      MVI    C,40H-SIRKA      DOPLNOK DO RIADKU
0142 7E      SPAT     MOV    A,M      VYBER BYTE ZDR.
0143 12      STAX    D      PRENES DO CIELA
0144 23      INX     H      DALSI ZDROJ.B.
0145 13      INX     D      DALSI CIEL.B.
0146 05      DCR     B      ZNIZ SIRKU
0147 C2 4201     JNZ    SPAT      OPAKUJ AZ B=0
014A 09      DAD     B      NOVY RIAD.ZDROJ.
014B EB      XCHG
014C 09      DAD     B      NOVY RIAD.CIELA
014D EB      XCHG      VYMEN HL A DE
014E 22 0202     SHLD   BUFFER+2     SKRY ZDROJ.ADR.
0151 2A 0002     LHLD   BUFFER      VYBER CLDR
0154 7C      MOV     A,H
0155 92      SUB     D      ZISTI ROVNOST:
0156 C2 5B01     JNZ    KONIEC      CLDR=CLHR
0159 7D      MOV     A,L
015A 93      SUB     E
015B C8      KONIEC   RZ      AK ROVNE KONIEC
;
015C 2A 0202     LHLD   BUFFER+2     AK NIE,OBNOV HL
015F C3 3E01     JMP     ZNOVA      OPAKUJ
;
0162      ORG     0200H
0200      BUFFER  DS     4
;
      END

```

```

;
;
;PRIKLAD 07
;
;NASOBENIE DVOCH 8-BITOVYCH CISEL
;
;VSTUP : <B> - 1.CISLO
;        <C> - 2.CISLO
;VYSTUP: <HL>- SUCIN
;
;
CIS1 EQU 4
CIS2 EQU 6
0132 1E 04      MVI E,CIS1
0134 0E 06      MVI C,CIS2
0136 16 08      MVI D,8          8 CIAST.SUCTOV
0138 06 00      MVI B,0
013A 21 0000     LXI H,0          1.CIAST. SUCET
013D 29         L0 DAD H          POSUVA DOLAVA HL
013E 7B         MOV A,E
013F 07         RLC              NASTAV CY
0140 5F         MOV E,A
0141 D2 4501     JNC L1          SKOK AK CY=0
0144 09         DAD B           CIAST.SUCET
0145 15         L1 DCR D        ZNIZ POCITADLO
0146 C2 3D01     JNZ L0         OPAKUJ AZ D=0
0149 C9         RET
END

```

```

;
;PRIKLAD 8
;GENEROVANIE MELODIE
;VSTUP : 0
;VYSTUP: 0
;
;-----
PC      EQU      0F6H
;
0132 21 0010  ZAC    LXI    H,TABTON    NASTAV TAB.
0135 4E        OPAT   MOV    C,M        DLZKA TONU
0136 23        INX    H                DALSI BYTE
0137 56        MOV    D,M                VYSKA TONU
0138 23        INX    H                DALSI TON
0139 3E 2A        MVI    A,'*'          TEST KONCA
013B B9        CMP    C                MELODIE
013C C8        RZ                      KONIEC PROGRAMU
013D CD 4301     CALL   TON             GENERUJE TON
0140 C3 3501     JMP    OPAT            OPAKUJ
;
;PODPROGRAM GENERUJE TON URCITEJ VYSKY A DLZ
;
0143 DB F6      TON    IN      PC        CITAJ PORT C
0145 EE 04      XRI    04H            NASTAV 3.BIT
0147 5A        MOV    E,D            VYSKA TONU
0148 D3 F6      OUT    PC            ZVUK.VYSTUP
014A 1D        L000   DCR    E        ONESKORENIE
014B C2 4A01    JNZ    L000
014E EE 04      XRI    04H            NEGUJ 3.BIT
0150 5A        MOV    E,D            VYSKA TONU
0151 D3 F6      OUT    PC            ZVUK.VYSTUP
0153 1D        L111   DCR    E        ONESKORENIE
0154 C2 5301    JNZ    L111
0157 0D        DCR    C                ZNIZ DLZKU TONU
0158 C2 4301    JNZ    TON
015B 06 02      MVI    B,2            PAUZA ZA TONOM
015D 0E FF      N00    MVI    C,255
015F 0D        N11    DCR    C
0160 C2 5F01    JNZ    N11
0163 05        DCR    B
0164 C2 5D01    JNZ    N00
0167 C9        RET                    KON.PPGM TON
;
;
;TABULKA TONOV V PORADI : DLZKA,VYSKA
;
0168          ORG    1000H
1000 82 FE      TABTON DB    130,254    C
1002 89 E2      DB    137,226    D
1004 91 CA      DB    145,202    E
1006 94 BE      DB    148,190    F
1008 9B AA      DB    155,170    G
100A A9 97      DB    169,151    A
100C AD 87      DB    173,135    H
100E B1 7F      DB    177,127    C
1010 2A 2A      DB    '***'      UKONC.ZNAK
;
END

```



```

;
;PROGRAM 9
;
;GENEROVANIE MELODIE
;
;VSTUP :
;VYSTUP: PROGRAM GENERUJE MELODIU PODLA TAB.
;
;-----
;
BEEP EQU 88A3H PIPA PODLA TAB.
STOP EQU 8C74H TEST STLAC.STOP
ADRTAB EQU 0C130H OBSAH.ADR.TABULK
;
0132 21 0010 LXI H,TAB NASTAV TAB.
0135 22 30C1 SHLD ADRTAB ULOZ ADR.TAB.
0138 CD A388 OPAKUJ CALL BEEP PIPAJ PODLA TAB.
013B 3A 0110 LDA TAB+1 VYBER DLZKU TONU
013E 3C INR A ZVYS DLZKU TONU
013F FE 0A CPI 10 MAX.DLZKA TONU
0141 C2 4601 JNZ LOOP
0144 3E 01 MVI A,1 MIN.DLZKA TONU
0146 32 0110 LOOP STA TAB+1 NASTAV DLZKY T.
0149 32 0310 STA TAB+3
014C 32 0510 STA TAB+5
014F CD 748C CALL STOP STLAC.STOP ??
0152 C2 3801 JNZ OPAKUJ
0155 C9 RET ANO --> KONIEC
;
;TABULKA TONOV OBSAHUJE: VYSKA T., DLZKA T.
;(VYSKY TONOV: 0,1,2,3 )(DLZKA: 0-255)
;NA KONCI TAB.MUSI BYT UKONCOVACI ZNAK
;
0156 ORG 1000H
1000 01 01 TAB DB 1,1 TON 1
1002 02 01 DB 2,1 TON 2
1004 03 01 DB 3,1 TON 3
1006 FF DB 255 UKONC.ZNAK
;
END

```

Záver

Vážení záujemcovia o programovanie v JSA. Dúfame, že sa Vám dostala do rúk publikácia, ktorá Vám pomôže "prelúskat", sa úskaliami tohto jazyka. Pre lepšiu prácu v JSA musíte mať k dispozícii okrem samotného PMD 85-2 aj ROM-modul MRS, ktorý si môžete zakúpiť v UP B.Bystrica. JSA vyžaduje náročnejšie štúdium, trpezlivosť a precíznosť pri tvorbe programov, k čomu Vám prajeme veľa úspechov.

Poznámka: Publikácia neprešla jazykovou úpravou.

Zoznam použitej literatúry: M.Babák : Mikroprocesorová technika
M.Daniš : Mikroprocesorová technika -cv
V.Dvořák : Automatizačné cvičenia