

TESLA PIEŠŤANY, KONCERNOVÝ PODNIK

SKOLSKÝ MIKROPOČÍTAČ PMI 80

Užívateľská príručka

Časť II: Popis základného monitora

UPOZORNENIE I

V školskom mikropočítači PMI 80 je použitá nová verzia riadiaceho programu MONITOR. Nahrávanie z magnetofónu nie je kompatibilné s dosiaľ používanou verziou.

Piešťany, apríl 1982

1. Úvod

Monitor PMI-80 verzia V1 je základný rezidentný riadiaci program umiestnený v pevnej pamäti EPROM typu MHB8708 a je určený k riadeniu a obsluhu školského mikropočítača PMI-80.

Monitor zaistuje tieto základné funkcie systému :

- inicializáciu systému
- možnosť spracovania vonkajšieho prerušenia
- prehľadávanie a zmenu obsahu registrov
- prehľadávanie a zmenu obsahu pamäte
- štart užívateľského programu
- trasovanie programu /break point/
- čítanie a zápis údajov na kazetový magnetofón

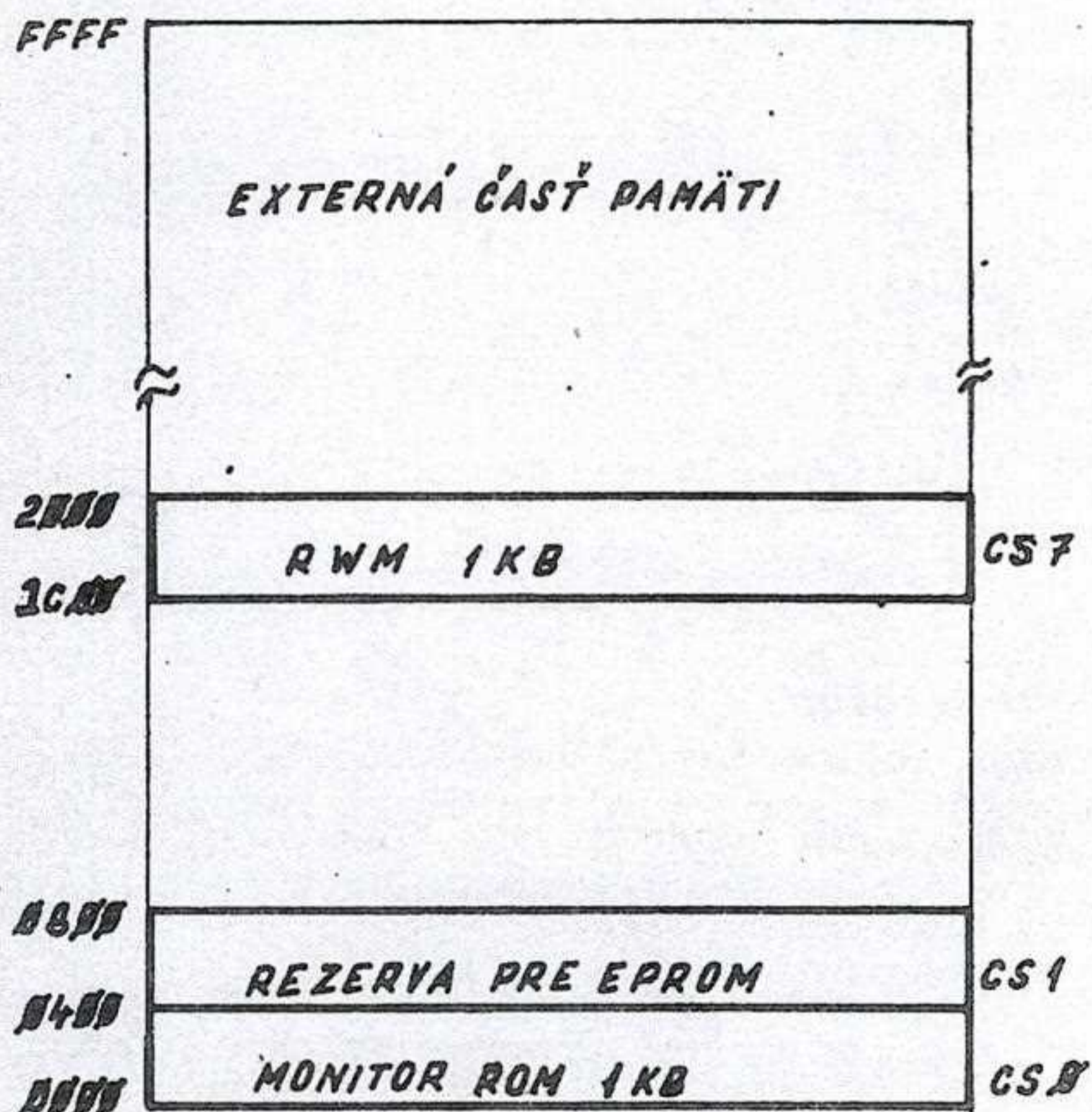
Okrem uvedených základných funkcií poskytuje monitor užívateľovi radu vedľajších funkcií tým, že zahŕňa väčší počet vnútorných podprogramov dostupných užívateľovi napr. vstup z klávesnice, výstup na segmentový displej atď.

Všetky funkcie mikropočítačového systému sú implementované programovo okrem funkcií RESET a INTERRUPT, ktoré sú vyvolané hardwareovo priamo z príslušnej klávesy.

Ako základné operátorské zariadenie používa monitor 25 prvkovú klávesnicu a deväťmiestny sedemsegmentový displej, ktoré sú riadené prostredníctvom paralelného interfejsového obvodu MHB8255A. Pre užívateľa zostáva k dispozícii jeden osembitový kanál PB, ktorý je inicializovaný ako vstupný.

2. ADRESOVANIE VNÚTORNÝCH PAMÄTÍ A INTERFEJSU

Táto kapitola stručne obsahuje informácie nutné pre programátora, ktoré sú implementované v hardware systéme.



Na obr. č.1 je znázornená mapa pamäte. Časti vybraté CS 0, 1 a CS 7 sa nachádzajú v konfigurácii systému. Ostatné časti pamäte sú externé. Operačná časť RWM je rozdelená nasledovne :
 1C00 až 1FD8 je k dispozícii užívateľovi. 1FD9 až 1FFF - rezervované pre monitor. Časť RWM rezervovaná pre monitor tvorí zázpisník monitora, ktorý sa používa pri interpretovaní prebiehajúcej funkcie.

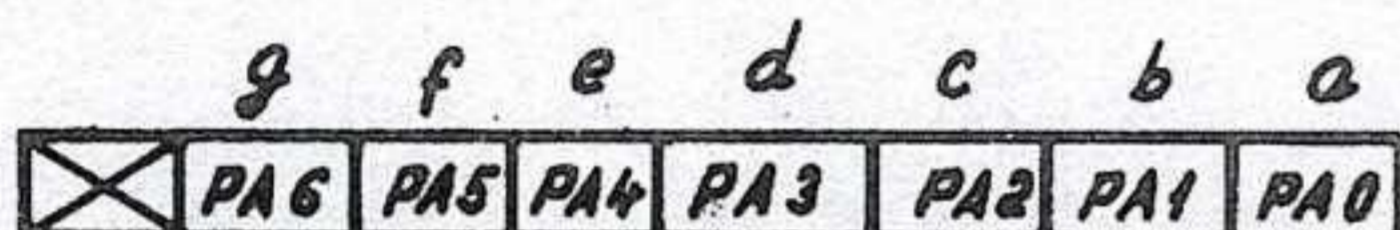
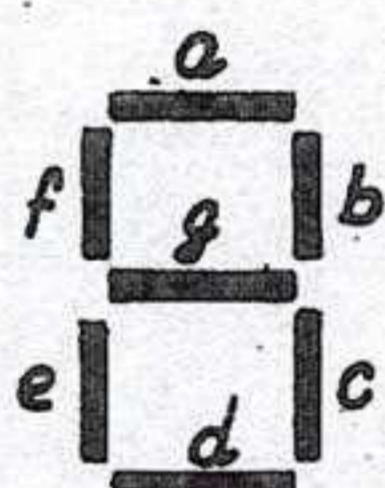
Obr. č.1 Mapa pamäte

Z užívateľského hľadiska najdôležitejšie miesta zázpisníka monitora sú zhrnuté v tabulke č.1.

Adresa	Význam
1FD9	vrchol ukazovateľa zásobníka /STACK/
1FE6 + 1FE8	miesto pre uloženie JMP od RST 7
1FEF + 1FF7	výstupný register údajov pre zobrazenie na displej
1FF8 - 1FF9	bežná vstupná adresa
1FFA	bežné vstupné dáta
1FFC - 1FFD	ukazovateľ výstupného registra pre zobrazenie /od pozície 0/

Tab. č.1 Najdôležitejšie miesta v zázpisníkovej pamäti monitora.

Výstupné údaje sa zobrazujú na deväťmiestnom sedemsegmentovom displeji. Ľavá krajná pozícia displeja je označená ako pozícia 0, pravá krajná ako pozícia 8. Pre zobrazenie adresy sú vymedzené pozície 2, 3, 4 a 5 zľava a pre dáta posledné pozície 7, 8. Na pozícii 0 sa zobrazuje znak informujúci o druhu prebiehajúcej monitorskej funkcie. Na displeji možno zobrazovať ľubovoľnú konfiguráciu segmentov v daných pozíciách. Obr. č.2 zobrazuje priradenie výstupných liniek interfejsového obvodu MHB8255A kanálu PA k jednotlivým segmentom displeja.



Monitor obsahuje tabuľku znakov, ktoré možno zobrazovať na displeji - viď tab.č.2. Pre zobrazenie žiadaného znaku na danú pozíciu displeja postačuje vložiť jeho poradovú adresu v tabuľke do príslušnej adresy výstupného registra.

Obr. č.2 Priradenie segmentov k výstupnému kanálu PPI.

Príklad : Je žiadúce zobraziť znak "H" na nultú pozíciu displeja.

V tabuľke znakov /tab. č.2/ sa znak "H" nachádza na adrese 1A, a preto bude výstupný register obsahovať na svojom prvom mieste /adresa 1FEF/ údaj 1A.

Zápisník monitora obsahuje ukazateľ výstupného registra údajov, a tým je možné dynamicky meniť zobrazované údaje buď ako konštantné reťazce, alebo ako premenlivé modifikovateľné údaje /adresa, dáta a pod./.

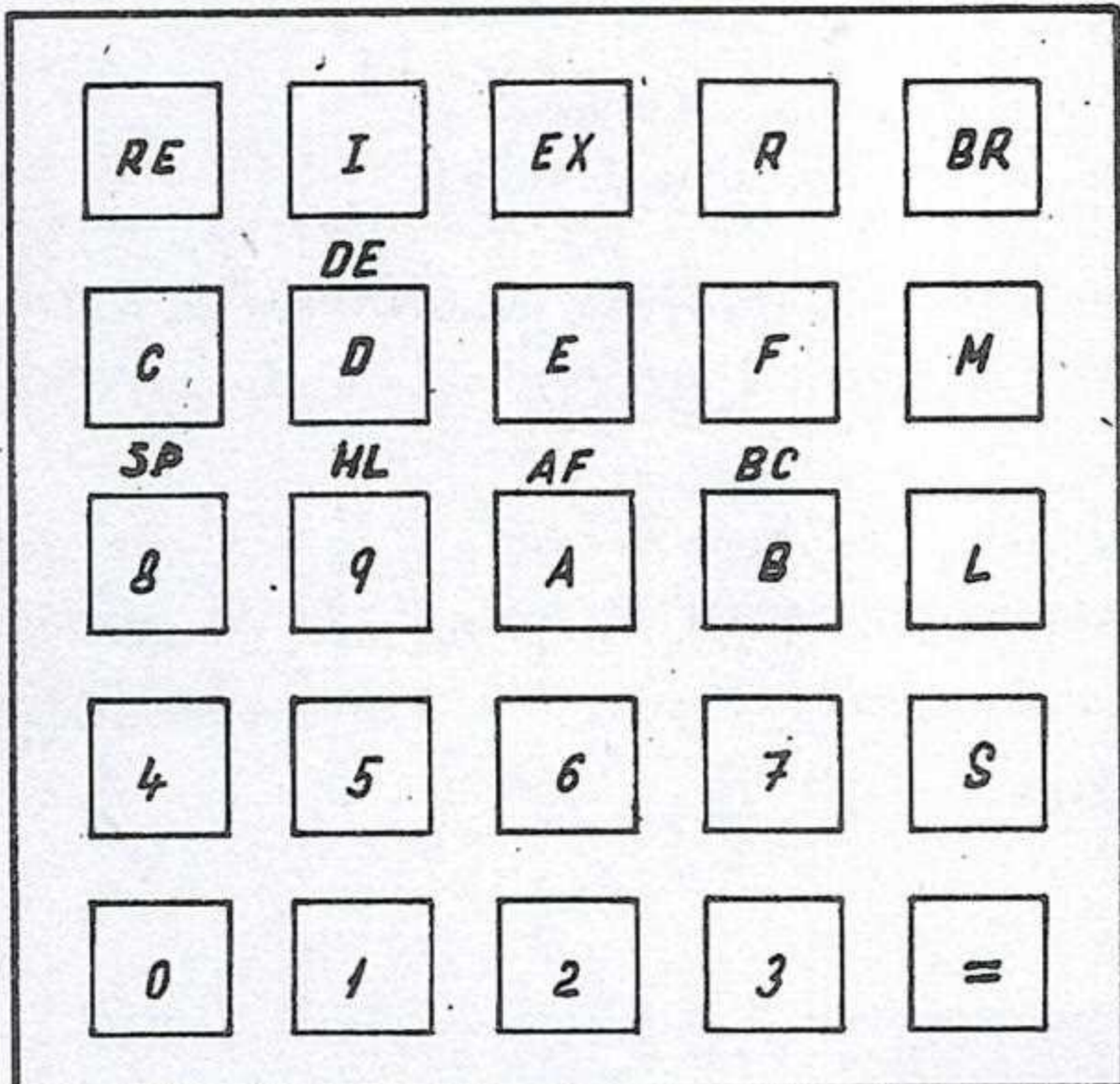
ADRESA	ZNAK	ADRESA	ZNAK	ADRESA	ZNAK	ADRESA	ZNAK	ADRESA	ZNAK
00	0	07	7	0E	E	15	u	1C	h
01	1	08	8	0F	F	16	n	1D	?
02	2	09	9	10	t	17	j	1E	"
03	3	0A	A	11	o	18	=	1F	-
04	4	0B	b	12	r	19	blank	20	G
05	5	0C	c	13	p	1A	H	21	u
06	6	0D	d	14	l	1B	n	22	i

Tab. č.2 Znaký, ktoré sú uložené v monitore pre zobrazovanie na displej.

Pre vstupy údajov je systém vybavený 25 prvkovou klávesnicou. Konfigurácia kláves je znázornená na obr. č.3. V podstate možno rozdeliť klávesy do dvoch skupín.

V prvej skupine /farba modrá/ sú to klávesy s riadiacou funkciou, pričom dve z nich sú, ako bolo spomínané, zabudované do systému /RE, I/.

Druhá skupina kláves reprezentuje množinu hexaznakov, pričom klávesy 8, 9, A, B a D pri činnosti výpisu vnútorných registrov CPU reprezentujú jednotlivé páry registrov. Funkcia kláves a im priradené dáta je znázornené v tabuľke č.3.



Obr. č.3 Pohľad na klávesnicu

dáta	klávesa	význam
80 + 8F	0 + F	hexadecimálne znaky 0 + 9, A, B, C, D, E, F
90	=	ukončenie činnosti /NEXT/
91	EX	štart užívateľského programu
92	M	modifikovanie obsahu pamäti
93	L	čítanie údajov z MG
94	S	zápis údajov na MG
97	BR	trasovanie programu /break point/
9A	R	modifikovanie registrov CPU
-	RE	reset systému
-	I	vyvolanie prerušenia

Tab. č.3 Funkcie kláves

Mikropočítačový systém PMI-80 obsahuje v základnej zostave jeden interfejsový obvod MHB8255A. Jeho adresovanie je nasledovné :

kanál PA adresa F8
kanál PB adresa F9
kanál PC adresa FA
riadiaci reg. adresa FB

Pretože užívateľ má k dispozícii v základnej zostave len jeden kanál PB, ktorý môže byť buď vo funkcii vstupu alebo výstupu, riadiace slovo môže byť pre tento kanál len :

- a./ kanál PB ako vstupný - riadiace slovo je 8A
- b./ kanál PB ako výstupný - riadiace slovo je 88

Pre rozšírený systém t.j. pri použití aj druhého PPI obvodu sú adresy jeho jednotlivých kanálov nasledovné :

kanál PA adresa F4
kanál PB adresa F5
kanál PC adresa F6
riadiaci reg. adresa F7

Tieto kanály sú plne k dispozícii užívateľovi a môžu pracovať v režimoch 0, 1 a 2.

3. POPIS PRÍKAZOV MONITORU

Jednotlivé príkazy pre monitor sa zadávajú stisknutím príslušnej klávesy. Klávesy **RE** a **I** negenerujú žiaden kód, ale vyvolávajú príslušné signály v hardware. Ostatné klávesy generujú kód, ktorý je spracovaný monitorom ako monitorský príkaz.

Použitá symbolika v definícii príkazov.

- a./ parametre uvedené v špicatých zátvorkách < > sú povinné
- b./ parametre /adresa/ resp. /dáta/ predstavujú adresu t.j. 2 Byty = 4 znaky, resp. dáta t.j. 1 Byte = 2 znaky hexadecimálnych čísiel
- c./ parametre uvedené v špicatých i guľatých zátvorkách sú nepovinné

Monitorský príkaz, ktorý je otvorený, je možno ukončiť stlačením klávesy, ktorá nepatrí do množiny hexa kláves mimo kláves **RE** a **I**. V príkladoch je takto označovaná klávesou ako . Monitor vydá na tento úkon správu o chybe týkajúcu sa adresného poľa alebo dátového poľa displeja. /Viď kapitola 4/. Každá správa vydaná monitorom sa akceptuje ľubovoľným stiskom klávesy /okrem **RE** a **I** /. Tento úkon je označovaný klávesou .

3.1 RE - inicializácia systému /RESET/

Stisknutím klávesy **RE** sa vydáva hardwarový signál, ktorý uvedie systém do počiatočného stavu. To sa týka nielen procesoru 8080A, ale i ostatných obvodov. Riadenie sa odovzdá na inštrukciu umiestnenú na adrese 0000, čím sa spustí riadiaci program monitora. Monitor prevedie svoju vlastnú inicializáciu, nastaví ukazovateľ zásobníka na vrchol a vypíše správu :

"	P	N	I	-	E	D	"
---	---	---	---	---	---	---	---

Potvrdenie správy /ako i každej inej/ je stlačením ľubovoľnej klávesy. Po tomto úkone sa objaví len v ľavej časti displeja sprievodný znak "P", čím monitor dáva najavo, že je pripravený prijať niektorý z jeho príkazov.

3.2 INT - vonkajšie prerušenie /Interrupt/

Stlačením klávesy **I** sa vydáva požiadavka na vonkajšie prerušenie procesora. Prerušenie je akceptované len v prípade jeho povolenia. Ak k nemu došlo, odovzdá procesor riadenie na adresu 38 a uloží návratovú adresu do zásobníka rovnako ako u inštrukcie RST 7.

Aby mal užívateľ možnosť prerušenie programovo obslúžiť, je na adrese 38 umiestnený skok do zápisníka monitora na adresu 1FE6. Na túto adresu musí teda užívateľ umiestniť svoj vektor obslužnej rutiny. Pre tento účel má k dispozícii 3 byte.

3.3 REG - modifikácia obsahu registrov CPU /Registen/

Tvar príkazu :

R < reg > < (dáta) > **=**

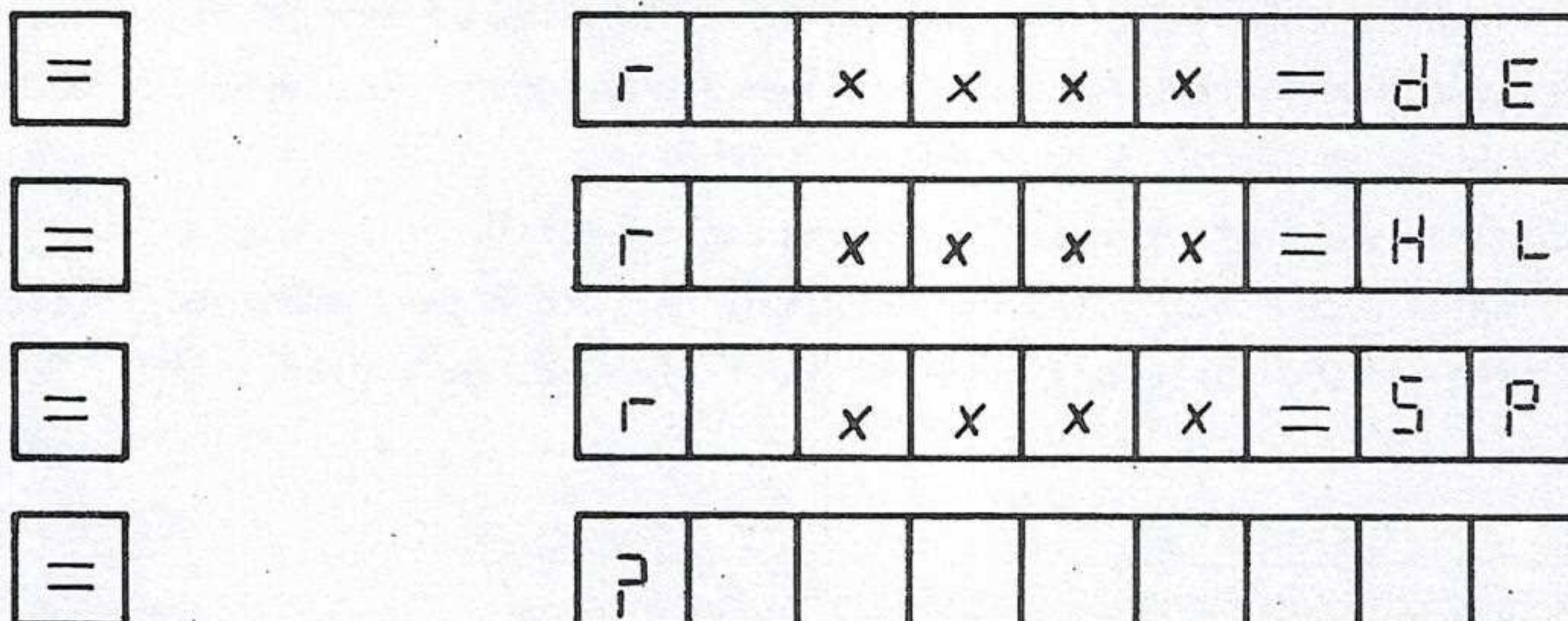
Stisknutím klávesy **R** sa zahajuje prevádzanie funkcie prehľadávania popriprade zmeny obsahu registrov CPU, čo dáva i náznakový symbol "r". Obsahy registrov sa budú zobrazovať v pároch po poradí : AF, BC, DE, HL a SP počínajúc od navoleného registrového páru. Stisknutím klávesy **=** sa ukončuje manipulácia s obsahom daného reg. páru a prechádza v poradí k ďalšiemu a v prípade posledného k žiadosti o monitorský príkaz. (**R**)

Príklad č.1 Prehľadavanie obsahov registrov a zmena reg. páru BC = 22 33.

klávesnica

displej

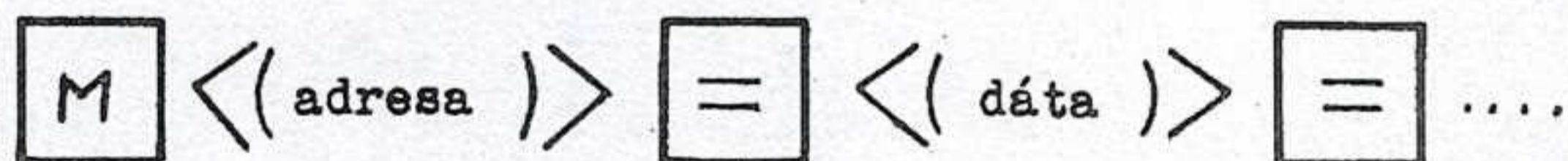
	R	r							
	A	r	x	x	x	x	=	A	F
	=	r	x	x	x	x	=	b	c
	2	r	x	x	x	2	=	b	c
	2	r	x	x	2	2	=	b	c
	3	r	x	2	2	3	=	b	c
	3	r	2	2	3	3	=	b	c



Poznámka : znak "X" v displejovej časti reprezentuje momentálny obsah -
- údaj

3.4 MEM - modifikácia obsahu pamätového miesta

Tvar príkazu :



Stisknutím klávesy M sa zahajuje prevedenie príkazu pre prehľadávanie a zmenu obsahu pamäti, čo monitor potvrdzuje náznakovým symbolom "P" v ľavej krajnej pozícii displeja. Bezprostredne po tom sa zobrazí v adresnej časti displeja posledná /bežná/ adresa, ktorú možno ponechať alebo modifikovať. Stiskom klávesy = sa ukončuje časť týkajúca sa vloženia adresy a prechádza sa k modifikácii už len dát na tejto adrese poprípade nasledujúcich vzostupných adries. Potvrdenie platnosti dát je opäť klávesou =.

Príklad č. 2 Vložte na príslušné pamätové miesto nasledovný programový fragment :

adresa	dáta	symbolický zápis
1C00	21	LXI H, 22 00
1C01	00	
1C02	22	
1C03	11	LXI D, 33 00
1C04	00	
1C05	33	
1C06	76	HLT

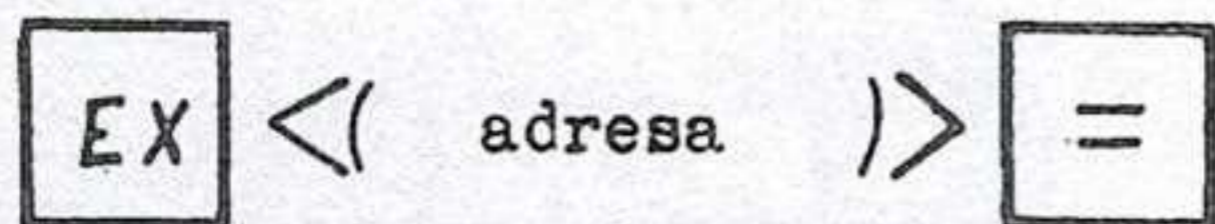
Vkladanie dát od adresy 1C00 :

krok	klávesnica	displej
		P
1.	M	M x x x x
2.	1	M x x x 1
3.	C	M x x 1 C
4.	0	M x 1 C 0
5.	0	M 1 C 0 0
6.	=	M 1 C 0 0 = x x
7.	2	M 1 C 0 0 = x 2
8.	1	M 1 C 0 0 = 2 1
9.	=	M 1 C 0 1 = x x
10.	0	M 1 C 0 1 = x 0
11.	0	M 1 C 0 1 = 0 0
12.	=	M 1 C 0 2 = x x

Ďalšie úkony sa opakujú podľa krokov 10, 11, 12 až do požadovanej adresy.

3.5 GO - štart programu

Tvar príkazu :



Stisknutím klávesy **EX** je možné spustiť program uložený v pevnej pamäti EPROM alebo RWM, do ktorej bol zavedený ručne alebo z MG. Bezprostredne po stisknutí klávesy **EX** sa zobrazí náznakový symbol "□" a v adresnom poli displeja stávajúci obsah čítača adres PC. Teraz je možné modifikovať adresu spôsobom ako je uvedené v príklade 2 krokmi 2, 3, 4 a 5 a klávesou **=** ukončiť príkaz GO alebo ponechať zobrazovanú adresu a akceptovať ju klávesou **=**.

Po tomto deje sa vynuluje displej a v ľavej krajnej pozícii sa zobrazuje silnejším svitom znak "E".

Riadenie opustilo monitor a je pod užívateľským riadením. Z tohto stavu môže byť návrat späť do monitora len stiskom **RE** reset systému alebo ak je povolené prerušenie a uložený prerušovací vektor stiskom klávesy **I** - interrupt.

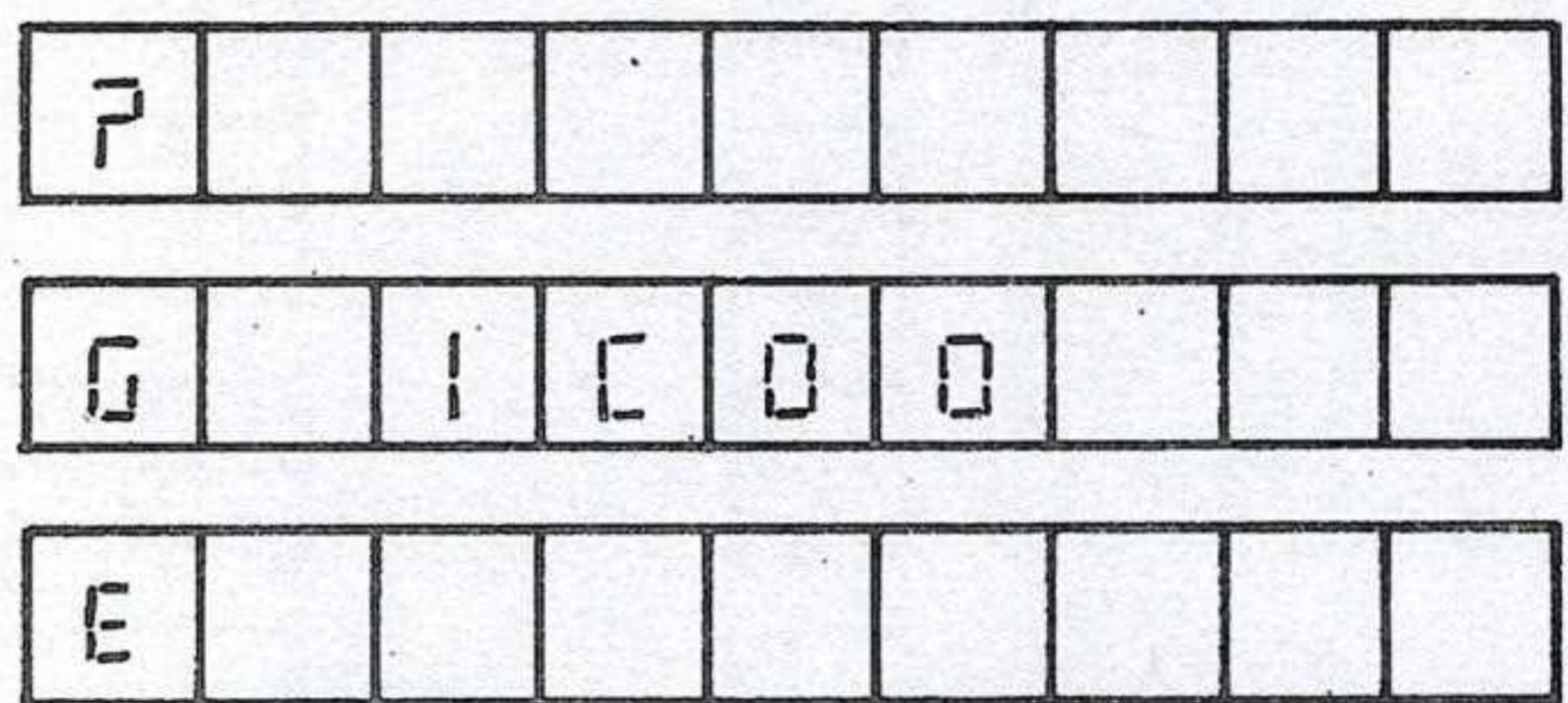
Príklad č.3 - Štart programu od adresy 1C00H, ktorá sa nachádza v PC.

klávesnica

displej

EX

=



3.6 BREAK - zastavenie programu /Break point/

Tvar príkazu :

BR <(adresa)> = <(adresa)> =

Tento monitorový príkaz slúži na odlaďovanie programov umiestnených v pamäti RWM. Filozofia spôsobu odlaďovania programov týmto spôsobom spočíva vložení "zarážky" /break point/ - zastavovacieho bodu reprezentujúci adresu inštrukcie. Pri interpretovaní programu a nabehnutí na tento programový bod, prevedie sa skok do monitora s uchovaním vnútorného stavu procesora. Tento stav možno monitorovým príkazom modifikovať. Posúvaním tohto zastavovacieho programového bodu možno previesť trasovanie programu a tým odlaďovať jednotlivé stavy programu. Treba však upozorniť na tú skutočnosť, že vkladanie "zarážky" musí byť uskutočnené na tú adresu inštrukcie, ktorá bude interpretovaná.

Bezprostredne po stisku klávesy BR a akceptovaní danej adresy "zarážky" klávesou = sa zobrazí štartovacia adresa, ktorú možno modifikovať. Pri trasovaní programom je táto adresa adresou "zarážky", čo uľahčuje odlaďovanie programu.

Príklad č.4 Je požadované zistiť, či v programovom fragmente uvedeného v príklade č.2 sa previedla inštrukcia LKI H, 22 00.

Riešenie : - štartovacia adresa = 1C00
 - zarážka /break point/ = 1C03

krok

klávesnica

displej

1.

BR

?

b x x x x

2.

1

b x x x i

3.

c

b x x i c

4.

0

b x i c 0

5.

3

b i c 0 3

6.

=

c x x x x

7.

1

c x x x i

8.

c

c x x i c

9.

0

c x i c 0

10.

0

c i c 0 0

11.

=

" b r - 5 t o p "

12.

X

?

13.

R

r

14.

HL
9

r 2 2 0 0 = H L

15.

X

E r r - A d r e s

16.

X

?

3.7 LOAD - čítanie dát z kazetového magnetofónu do pamäti


Tvar príkazu :

L <(adresa)> = <dáta > = =

Stisknutím klávesy L sa vyvolá obslužný program monitora pre čítanie bloku dát z MG do pamäti mikropočítača. Bezprostredne po stisknutí L je možné explicitne zadať adresu, od ktorej sa budú ukladať data do pamäti. Pokiaľ adresa nebola zadaná, budú sa dáta ukladať do pamäti od adresy dané stavom čítača adres PC. Po akceptovaní adresy stiskom klávesy = je nutné zadať do dátového poľa displeja adresu bloku, ktorý sa bude čítať z MG. Po jeho vložení a potvrdení klávesou = sa zobrazí na displeji žiadosť pre zapnutie MG. Monitor očakáva potvrdenie tejto žiadosti stiskom ľubovolnej klávesy /okrem RE a I /. Od toho okamžiku sa vyhľadá príslušný blok a prenesie sa do pamäti mikropočítača. Po ukončení prenosu monitor vypíše správu pre ukončenie činnosti MG a po jej potvrdení sa vracia do svojho vstupného bodu / " " /. V prípade, že by sa k žiadanému bloku nepribližovalo, obslužný program túto udalosť rozpozná a vydáva žiadosť o spätné previnutie MG, ktoré treba taktiež akceptovať klávesou = . Po tejto činnosti sa vracia obslužný program do bodu, kde vydáva žiadosť o spustenie MG a činnosť sa opakuje. Treba pripomenúť, že prenos dát z MG do operačnej pamäti mikropočítača končí zapísaním posledného bytu stránky. Stránkou sa rozumie súvislý úsek max. 256 bytov pamäti, adresovaný H adresou, pričom dĺžka stránky - adresa L nemusí začínať nulou.

Pri približovaní sa k žiadanému bloku dát na páske MG, MONITOR vypíše adresu bloku /MARK/. Obsluha počas tohto zobrazovania môže manipulovať s MG /vpred, vzad/ a kvitovaním ľubovolnou klávesou /okrem RE a I / sa vracia opäť do režimu snímania dát z MG. Týmto spôsobom je umožnená obsluha rýchla orientácia blokov na páske MG.

Príklad č.5 Je požadované čítanie dát z MG od bloku s adresou 10.
 Dáta sa majú ukladať do stránky operačnej pamäti 1D20
 /čiže začiatok stránky bude od 20/.

krok	klávesnica	displej										
0.		<table border="1"> <tr> <td>?</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>	?									
?												
1.	L	<table border="1"> <tr> <td>L</td> <td></td> <td>x</td> <td>x</td> <td>x</td> <td>x</td> <td></td> <td></td> <td></td> <td></td> </tr> </table>	L		x	x	x	x				
L		x	x	x	x							
2.	1	<table border="1"> <tr> <td>L</td> <td></td> <td>x</td> <td>x</td> <td>x</td> <td>1</td> <td></td> <td></td> <td></td> <td></td> </tr> </table>	L		x	x	x	1				
L		x	x	x	1							
3.	0	<table border="1"> <tr> <td>L</td> <td></td> <td>x</td> <td>x</td> <td>1</td> <td>0</td> <td></td> <td></td> <td></td> <td></td> </tr> </table>	L		x	x	1	0				
L		x	x	1	0							
4.	2	<table border="1"> <tr> <td>L</td> <td></td> <td>x</td> <td>1</td> <td>0</td> <td>2</td> <td></td> <td></td> <td></td> <td></td> </tr> </table>	L		x	1	0	2				
L		x	1	0	2							
5.	0	<table border="1"> <tr> <td>L</td> <td></td> <td>1</td> <td>0</td> <td>2</td> <td>0</td> <td></td> <td></td> <td></td> <td></td> </tr> </table>	L		1	0	2	0				
L		1	0	2	0							
6.	=	<table border="1"> <tr> <td>L</td> <td></td> <td>1</td> <td>0</td> <td>2</td> <td>0</td> <td></td> <td>x</td> <td>x</td> <td></td> </tr> </table>	L		1	0	2	0		x	x	
L		1	0	2	0		x	x				
7.	1	<table border="1"> <tr> <td>L</td> <td></td> <td>1</td> <td>0</td> <td>2</td> <td>0</td> <td></td> <td>x</td> <td>1</td> <td></td> </tr> </table>	L		1	0	2	0		x	1	
L		1	0	2	0		x	1				
8.	0	<table border="1"> <tr> <td>L</td> <td></td> <td>1</td> <td>0</td> <td>2</td> <td>0</td> <td></td> <td>1</td> <td>0</td> <td></td> </tr> </table>	L		1	0	2	0		1	0	
L		1	0	2	0		1	0				
9.	=	<table border="1"> <tr> <td>"</td> <td>1</td> <td>0</td> <td></td> <td>2</td> <td>0</td> <td></td> <td>"</td> <td></td> <td></td> </tr> </table>	"	1	0		2	0		"		
"	1	0		2	0		"					
10.	=	<table border="1"> <tr> <td>E</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>	E									
E												
11.		<table border="1"> <tr> <td>"</td> <td>1</td> <td>0</td> <td></td> <td>5</td> <td>E</td> <td>0</td> <td>P</td> <td></td> <td>"</td> </tr> </table>	"	1	0		5	E	0	P		"
"	1	0		5	E	0	P		"			

V kroku 10 dochádza k vyhľadávaniu a prenosu dát z MG a po jeho prenose vypisuje monitor správu o zastavení MG, čo obsluha akceptuje v kroku 11 zatlačením klávesy a návrat do kroku 0.

3.8 SAVE - zápis dát z operačnej pamäti na MG

Tvar príkazu :

\boxed{S} <(adresa)> $\boxed{=}$ <dáta> $\boxed{=}$ $\boxed{=}$

Stisknutím klávesy \boxed{S} sa vyvolá obslužný program monitora pre zápis bloku dát z pamäti na MG. Činnosť obsluhy je podobná ako pri príkaze LOAD až na to, že je nutné predom nastaviť kazetu MG na požadované miesto. Vzhľadom na to, že monitor neobsahuje knižnicu blokov umiestnených na kazete MG, je nutné dodržať vzostupnosť číslovania /adresovania/ blokov na páske, a uchovať ich písomne aj s približným fyzickým umiestnením.

Pri žiadosti zápisu bloku dát na pásku MG za posledný blok dát je možno postupovať tak, že sa prečíta posledný blok dát z MG do pamäte ROM napr. na adresu 0000 a tým sa zistí fyzické miesto pásky pre zápis ďalšieho bloku. Takýmto spôsobom je možné i prepisovanie blokov dát na páske MG, ale je tu určité nebezpečenie zničenia nasledujúceho bloku. Toto riziko možno vylúčiť vhodným odstupom blokov na páske MG.

4. SPRÁVY HLÁSENÉ MONITOROM

Nasledujúce správy signalizuje monitor na displeji v rámci dialógu s obsluhou. Monitor prijíma akceptovanie tejto správy stiskom ľubovoľnej klávesy /okrem **RE** a **I** /.

"		P	N	I	-	A	O	"
---	--	---	---	---	---	---	---	---

Správa sa zobrazuje pri inicializácii systému.

"		E	r	r	O	r	"
---	--	---	---	---	---	---	---

Chybný úkon, bez bližšieho určenia, závisí od predošlého riešenia.

E	r	r	=		d	A	I	R
---	---	---	---	--	---	---	---	---

Pri vkladani do dátového poľa bola zatlačená klávesa nepatriaca do množiny hexaznakov.

E	r	r	=	A	d	r	E	S
---	---	---	---	---	---	---	---	---

Pri vkladani do adresného poľa bola zatlačená klávesa nepatriaca do množiny hexaznakov.

"	b	r	=	S	t	O	P	"
---	---	---	---	---	---	---	---	---

Program bol zastavený na programovej "zarážke" /break point/

"	N	G		r	U	n	"
---	---	---	--	---	---	---	---

Žiadosť o spustenie MG.

"	N	G		S	t	O	P	"
---	---	---	--	---	---	---	---	---

Ukončenie činnosti s MG.

"	N	G		S	P	A	I	"
---	---	---	--	---	---	---	---	---

Je nutné pretočiť pásku späť.

Znak, ktorý monitor vypisuje v ľavej krajnej pozícii displeja, je daný priebehom monitorského príkazu. /náznakový symbol/

5. Podprogramy monitora dostupné užívateľovi

Monitor obsahuje celú radu podprogramov, ktoré sú dostupné i užívateľovi, t.j. môžu byť volené z užívateľského programu. Tým sa šetrí miesto v užívateľskom programe. Všetky monitorské podprogramy sa volajú štandardne inštrukciou CALL /alebo podmienenou inštrukciou CALL/. V priebehu svojej práce používajú užívateľský definovaný zásobník pre odkladanie dát alebo návratových adries. Zoznam týchto podprogramov je uvedený v tab. 4. Užívateľ, ak pozná vnútornú skladbu programových fragmentov, môže ich používať s ľubovoľným vstupným bodom, čím môže ovplyvniť vstupné argumenty fragmentu.

Meno	adr. /HEX/	Funkcia
CLEAR	00AB	Vymazanie displeja a zápis znaku do nulte pozície
ENTRY	0008	Vstup do MONITORA a uchovanie stavu procesora
TIN	0300	Prečítanie jedného osembitového slova z MG
TOUT	02D4	Zápis jedného osembitového slova do MG
OUTDA	00F2	Umiestni 1 byte do dátového poľa výstupného reg.
OUTAD	00BB	Umiestni 2 byte do adresného poľa výstupného reg.
MODDA	00FB	Modifikuj so zobrazením 1 byte v dátovom poli
MODAD	00D7	Modifikuj so zobrazením 2 byte v adresnom poli
OUTKE	0X16	Zobrazovanie údajov a snímanie znaku z klávesnice

Tab. č.4 Zoznam podprogramov MONITOR pre užívateľa.

5.1 Popis použitia podprogramov

U každého podprogramu je uvedená jeho volacia adresa, popis jeho funkcie, význam vstupných a výstupných argumentov a zoznam registrov, ktoré používa podprogram.

CLEAR adresa 00AB

Slúži k vynulovaniu výstupného registra /dĺžky 9 byte/ a zápis znaku, ktorého údaj je uložený v akumulátora do nulte pozície výstupného registra. Ukazovateľ výstupného registra je umiestnený na adrese 1FFC.

Vstup : A = znak určený adresou v tabuľke 2.

Adresa výstupného registra daná implicitne ukazovateľom na adrese 1FFC.

Výstup : definovaný výstupný register.

Používa reg.: HL, DE, A

Príklad č.6 Výstupný register začína adresou 1D00. Vynulujte ho a zapíšte do jeho nulte pozície znak P.

Riešenie :

LXI H, 1D00	;	Nastavenie výstup. registra
SHLD 1FFC	;	Zápis do jeho ukazateľa
MVI A, 13	;	Argument znaku P
CALL CLEAR	;	Exekutíva

ENTRY adresa 0008

Tento podprogram slúži pre vstup do MONITORa z užívateľského programu a zistenie v danom mieste užívateľského programu stav procesora. Návrat je možný monitorským príkazom - GO.

TIN adresa 0300

Je určený pre zosnímanie jedného osembitového byte z pásy MG a uloženie do registra C. Užívateľ musí zabezpečiť spustenie pásy od zvoleného miesta.

Vstup : Údaje na páske MG.

Výstup : Register C.

Používa reg. : BC, DE, A

TOUT adresa 02D4

Tento podprogram zabezpečí zápis obsahu registra C na kazetovú pásku MG. Užívateľ musí zabezpečiť spustenie pásky od zvoleného miesta.

Vstup : C - žiadaný údaj pre zápis.

Výstup : Páska MG.

Používa reg.: BC, DE, A

OUTDA adresa 00F2

Je určený pre uloženie jedného osembitového slova /byte/ do dátového poľa výstupného registra ako dva hexa-znaky. Vstupným argumentom je tu obsah adresy 1FFA v zápisníku MONITORa, ktorý si uchováva tzv. bežné dáta. Dátové pole výstupného registra MONITORa je pevne určené registrovým párom BC, avšak toto miesto možno udať aj explicitne do reg. páru BC a vyvolať podprogram s adresou + 3 t.j. 00F5.

Vstup : Dáta pre zobrazenie uložené na adrese 1FFA.

Výstup: Dva hexa-znaky umiestnené v dátovom poli výstupného registra MONITORa.

Používa reg.: BC, HL, D, A

Príklad č.7 Je žiadúce umiestniť do dátového poľa výstupného registra dáta 80.

Riešenie : MVI A, 80 ; vstupné dáta
 STA 1FFA ; zápis do zápisníka MONITORa
 CALL OUTDA ; exekutíva

OUTAD adresa 00BB

Podprogram je určený pre uloženie dvoch osembitových slov /2 byte/ do adresného poľa výstupného registra ako štyri hexa-znaky. Vstupným argumentom je tu obsah adres 1FFB /nižší byte/ a 1FF9. Adresné pole výstupného registra MONITORa je daná pevne reg. párom BC, avšak toto miesto možno zadať užívateľom explicitne do reg. páru BC a vyvolať podprogram s adresou + 3 t.j. 00BE.

Vstup : obsah adres 1FFB, 1FF9

Výstup : adresné pole výstupného registra MONITORa

Používa reg.: BC, HL, D, A

Príklad č.8 Umiestnite do výstupného registra MONITORA šestnásťbitový údaj 1C00 počnúc nultou jeho pozíciou.

Riešenie :

LXI B, 1FEF	;	nultá pozícia výstu reg.
LXI H, 1C00	;	vstupný argument
SHLD 1FF8	;	zápis arg. do zápisníka
CALL 00BE	;	exekutíva

MODDA adresa 00FB

Podprogram je určený pre zobrazenie obsahu výstupného registra určeného svojim ukazovateľom umiestneného na adresách 1FFC, 1FFD a modifikovanie obsahu dátového poľa výstupného registra MONITORA. Výstup z podprogramu je stiskom klávesy **[=]**. Obsah dátového poľa je uchovaný v zápisníku MONITORA ako bežné vstupné dáta na adrese 1FFA. V prípade stisku nehexaznaku okrem **[=]** **[RE]** **[I]** dochádza k vypísaniu správy o chybe a návrat do vstupného bodu MONITORA. Tento podprogram volá podprogramy OUTDA a OUTKE.

Vstup : klávesnica, obsah výstupného registra

Výstup : displej, dáta v zápisníku MONITORA na adrese 1FFA

Používa reg.: BC, HL, DE, A

MODAD adresa 00D7

Úlohou podprogramu je zobrazovať obsah výstupného registra určeného svojim ukazovateľom umiestneného na adresách 1FFC, 1FFD a modifikovanie obsahu adresného poľa výstupného registra MONITORA. Výstup z podprogramu je stiskom klávesy **[=]**. Obsah adresného poľa je uchovaný v zápisníku MONITORA ako bežná vstupná adresa na adresách 1FF8 /nižší byte/ a 1FF9. V prípade stisku nehexaznaku okrem kláves **[=]** **[RE]** **[I]** dochádza k vypísaniu správy o chybe a návrat do vstupného bodu MONITORA. Podprogram k svojej činnosti volá podprogramy OUTAD a OUTKE.

Vstup : klávesnica, obsah výstupného registra

Výstup : display, údaj v adresnom poli uchovaný na adresách 1FF8, 1FF9

Používa reg.: BC, HL, DE, A

OUTKE adresa 0116

Tento podprogram je hlavným podprogramom MONITORa. Jeho funkciou je obsah výstupného registra dekódovať na sedemsegmentový deväťmiestny display a súčasne zisťovať, či je zatlačený v danom stĺpci matice kláves, niektorá klávesa. Pri ukončení t.j. zobrazení poslednej pozície výstupného registra sa testuje stavové slovo klávesnice. V prípade, že došlo k stisku niektorej klávesy /okrem **RE** **I** / a jej uvoľneniu, je uskutočnený výstup z tohoto podprogramu. V akumulátore sa nachádza príslušný kód stisknutej klávesy /viď tab. č.3/. Ešte pred spustením podprogramu sa akumulátor porovnáva s hodnotou klávesy **=** t.j. údajom 90. Tým sa súčasne nastavujú testovacie /príznakové/ bity.

Vstup : klávesnica, výstupný register

Výstup : kód klávesy v A; nastavenie príznakových bitov

Používa reg.: všetky

Príklad č.9 Zostavte program pre testovanie klávesy **8** .

```
Riešenie : LOOP : CALL OUTKE          ;    exekutíva
                 CPI 88                 ;    kód 8
                 JNZ LOOP               ;    ak nie je, pokračuj na LOOP
```

PRÍLOHA - Súbor inštrukcií 8080

PRENOSNÉ	ARITMETICKÉ A LOGICKÉ	RIADIACE	I/O STACK	
7F AA MOV A,A 78 AB MOV A,B 79 AC MOV A,C 7A AD MOV A,D 7B AE MOV A,E 7C AM MOV A,M 7E AL MOV A,L 77 BA MOV B,A 78 BB MOV B,B 79 BC MOV B,C 7A BD MOV B,D 7B BE MOV B,E 7C BH MOV B,H 74 BL MOV B,L 75 BM MOV B,M 7F CA MOV C,A 78 CB MOV C,B 79 CC MOV C,C 7A CD MOV C,D 7B CE MOV C,E 7C CH MOV C,H 74 CL MOV C,L 75 CM MOV C,M 77 DA MOV D,A 78 DB MOV D,B 79 DC MOV D,C 7A DD MOV D,D 7B DE MOV D,E 7C DH MOV D,H 74 DL MOV D,L 75 DM MOV D,M 7F EA MOV E,A 78 EB MOV E,B 79 EC MOV E,C 7A ED MOV E,D 7B EE MOV E,E 7C EH MOV E,H 74 EL MOV E,L 75 EM MOV E,M 7F HA MOV H,A 78 HB MOV H,B 79 HC MOV H,C 7A HD MOV H,D 7B HE MOV H,E 7C HH MOV H,H 74 HL MOV H,L 75 HM MOV H,M 7F LA MOV L,A 78 LB MOV L,B 79 LC MOV L,C 7A LD MOV L,D 7B LE MOV L,E 7C LH MOV L,H 74 LL MOV L,L 75 LM MOV L,M 77 MA MOV M,A 78 MB MOV M,B 79 MC MOV M,C 7A MD MOV M,D 7B ME MOV M,E 7C MH MOV M,H 74 ML MOV M,L 75 MM MOV M,M XCHG EB	SE 5E MVI A, byte 06 5F MVI B, byte 07 58 MVI C, byte 08 59 MVI D, byte 09 5A MVI E, byte 0A 5B MVI H, byte 0B 5C MVI L, byte 0C 5D MVI M, byte 0D 5E MVI SP, byte 0E 5F MVI SP, byte 1E 5A MVI B, double 2E 5B MVI D, double 3E 5C MVI H, double 3F 5D MVI L, double 3E 5C MVI SP, double 01 5E LXI B, double 11 5F LXI D, double 21 5A LXI H, double 31 5B LXI SP, double 0A 5B LDAX B 1A 5C LDAX D 2A 5D LHLD adr 3A 5E LDA adr 02 5B STAX B 12 5C STAX D 22 5D SHLD adr 32 5E STA adr	C3 JMP adr C2 JNZ adr CA JZ adr D2 JNC adr DA JC adr E2 JPO adr EA JPE adr F2 JF adr FA JM adr E9 PCHL CD CALL adr C4 CNZ adr CC CZ adr D4 CNC adr DC CC adr E4 CPO adr EC CPE adr F4 CP adr FC CM adr C9 RET C0 ANZ C8 RZ D0 RNC D8 RC E0 RPO E8 RPE F0 RF F8 RM C7 RST 0 CF RST 1 D7 RST 2 DF RST 3 E7 RST 4 F7 RST 5 FF RST 6	A7 ANA A0 AB XRA A1 AC XRA A2 AD XRA A3 AE XRA A4 AF XRA A5 AG XRA A6 AH XRA A7 AI XRA A8 AJ XRA A9 AK XRA AA AL XRA AB AM XRA AC AN XRA AD AO XRA AE AP XRA AF AQ XRA B7 ORA B0 BR ORA B1 BS ORA B2 BT ORA B3 BU ORA B4 BV ORA B5 BW ORA B6 BX ORA BF BY ORA C0 BZ ORA C1 CA ORA C2 CB ORA C3 CC ORA C4 CD ORA C5 CE ORA C6 CF ORA C7 CH ORA C8 CL ORA C9 CM ORA D7 CMP D0 DA CMP D1 DB CMP D2 DC CMP D3 DD CMP D4 DE CMP D5 DF CMP D6 DH CMP D7 DL CMP D8 DM CMP D9 DN CMP DA DP CMP DB DQ CMP DC DR CMP DD DS CMP DE DT CMP DF DU CMP E7 DAA E8 CMA E9 STC EA CMC EB RLC EC RRC ED RAL EE RAR EF 07 F7 0F FF 17 FF 1F	B PUSH D H PSW C5 C5 D5 D5 E5 E5 F5 F5 C1 B POP D1 D E1 H F1 H PSW E3 XTHL F9 SPHL D3 OUT byte DB IN byte F3 DI FB EI 00 NOP 76 HLT

Pre dodávky nezáväzné! Všetky práva, zvlášť právo prekladu do cudzích jazykov vyhradené. Akékoľvek rozmnožovanie dovolené len s výslovným súhlasom vydavateľa.

Vypracoval Ing. Roman Kišš, I. vydanie v počte 300 výtlačkov.

Vydal: TESLA PIEŠŤANY, k. p. OTS, Piešťany, ČSSR 1982.