

MONITOR MIKROPOČÍTAČE

IQ-151

Monitorem nazýváme program, který zajišťuje spolupráci jednotlivých částí mikropočítače a umožňuje vkládat a spouštět program ve strojovém kódu mikroprocesoru 8080.

Monitor je uložen na adresách F000 až F9C2 v paměti EPROM, takže se po vypnutí počítače nevymaže a je vždy připraven k činnosti. Používá také některých buněk paměti RAM 32kB, která je adresována čísly 0000-7FFF.

Veškerá čísla uváděná v tomto přehledu jsou v hexadecimálním tvaru, pokud by mohlo dojít k nedorozumění, je za nimi připojeno písmeno H. Čísla v dekadickém tvaru jsou zakončena písmenem D, např.

$$20 = 20H = 32D.$$

Obsah

Zapojení klávesnice a tabulka kódů přiřazených tlačítkům	1
Seznam řídicích znaků	2
Obrazovka-přímý výstup, kurzor, podprogramy pro tisk	3
Přepínání periférií pomocí I/O bajtu	7
Adresy jednotlivých pozic na obrazovce	8
Příkazy monitoru	9
Přehled některých adres monitoru IQ-151	15
Přerušování programu	18
Přehled bran mikropočítače IQ 151	22
Přehled adres paměti 32 kB používaných monitorem	23
Přehled některých adres paměti 32 kB používaných BASICem 6	24
Spolupráce monitoru s basicem	25
Poznámka k práci s modulem BASIC-6	28
Tabulka kódů užívaných počítačem	29
Instrukční kód mikroprocesoru 8080	33
Jednoduchý zpětný assembler	34

Na SPŠ ve Varnsdorfu je dále k dispozici ASSEMBLER IQ pro vkládání, kontrolu, editaci a ladění programů ve strojovém kódu.

Tento přehled neprošel terminologickou ani jazykovou úpravou.

Klávesnice je zapojena jako matice 8x8 vodičů připojených z úrovně 1 na bránu 84 (A0-A7) a na bránu 85 (B0-B7). Při stisku tlačítka se postupně vyhodnotí uroveň 0 v příslušném sloupci a řádku dle následujících tabulek.

Kód stisknutého tlačítka můžeme vyhodnotit dvojným způsobem:

1. Podprogramem F8A s čekáním na stisk tlačítka; po stisku tlačítka se ozve zvukový signál a vyhodnocený kód podle následující tabulky je v registru A.
2. Podprogramem F8C9 bez čekání na stisk tlačítka; vyhodnocený kód je v registru C, při ne-stisknutém tlačítku je v A číslo 8A.

Řádu stisknutého tlačítka lze rychle zjistit instrukcí IN 85; v registru A jsou pak čísla uvedená v druhém sloupci tabulky. Při nestisknutém tlačítku je v A číslo FF.

Sloupec stisknutého tlačítka nelze zjistit pouhou instrukcí IN 84, protože obvod 8255 nemá bránu A naprogramovanou jako vstupní.

brána	kód tlačítka bez SH a CTRL								kód tlačítka a SH								kód tlačítka a CTRL							
84	A0	A1	A2	A3	A4	A5	A6	A7	A0	A1	A2	A3	A4	A5	A6	A7	A0	A1	A2	A3	A4	A5	A6	A7
85	31	32	33	34	35	36	37	38	21	22	23	24	25	26	27	28	31	32	33	34	35	36	37	38
86 - FE	1	2	3	4	5	6	7	8	!	"	#	\$	%	&	'	(1	2	3	4	5	6	7	8
87 - FD	51	57	45	52	54	59	55	49	71	77	65	72	74	79	75	69	11	17	05	12	14	19	15	09
	Q	W	E	R	T	Y	U	I	q	w	e	r	t	y	u	i	Q	W	E	R	T	Y	U	I
88 - F3	41	53	44	46	47	48	4A	43	61	73	64	66	67	68	6A	6B	01	13	04	06	07	08	0A	0B
	A	S	D	F	G	H	J	K	a	s	d	f	g	h	j	k	A	S	D	F	G	H	J	K
89 - F7	5A	58	43	56	42	4E	4D	2C	7A	78	73	76	62	6E	6D	3C	1A	18	03	16	02	0E	0D	2C
	Z	X	C	V	3	N	M	,	z	x	c	v	b	n	m	<	Z	X	C	V	3	N	M	,
90 - EF	39	4F	4C	2E	0D	1D	1C	01	29	6F	6C	3E	0D	1D	1C	01	39	0F	0C	2E	0D	1D	1C	01
	9	O	L	.	CR	DC	IC	F1)	o	I	>	CR	DC	IC	F1	9	O	L	.	CR	DC	IC	F1
91 - DF	30	50	33	2F	5C	1E	08	02	30	70	23	3F	7C	1E	08	02	30	10	33	2F	1C	1E	08	02
	0	P	;	/	\	DL	←	F2	0	P	+	?		DL	←	F2	0	P	;	/	\	DL	←	F2
92 - 3F	5E	53	5D	1A	05	03	0C	04	7E	73	7D	1A	05	03	0C	04	1E	1B	1D	1A	05	03	0C	04
	^	[]	↓	F5	IL	↖	F4	~	{	}	↓	F5	IL	↖	F4	^	[]	↓	F5	IL	↖	F4
93 - 7F	2D	40	3A	20	5F	19	ZK	03	3D	60	2A	20	7F	19	ZK	03	2D	00	3A	20	1F	19	ZK	03
	-	@	:	SP	—	↑	→	F3	=	\	*	SP		↑	→	F3	-	@	:	SP	—	↑	→	F3

Při stisku tlačítka → je v registru A kód znaku, na který ukazuje kurzor (i když je blikání kurzoru potlačeno). Kód tohoto znaku je uložen na adrese 0B.

Seznam řídicích znaků

tlačítko	kód	význam
CTRL A	F1	01 (použito v basicu-6; odpovídá, nevídeň příkaz)
CTRL B	F2	02 (v basicu-6 zablokuje nebo odblokuje klávesnici)
CTRL C	F3	03 (v basicu 6 zastaví výpis programu)
CTRL D	F4	04
CTRL E	F5	05
CTRL F		06
CTRL G		07 skok na podprogram F973 - zvukový signál,
CTRL H	←	08 posuv kurzoru vlevo
CTRL I		09 tabulátor s krokem 8 míst; posune kurzor na po - zice ve sloupcích 00,08,10,18 při modulu VIDEO-32, při modulu VIDEO 64 navíc na 20, 28, 30,38.
CTRL J		0A
CTRL K	IL	0B
CTRL L	↖	0C kurzor na pozici 0,0 (HOME)
CTRL M	CR	0D odřádkování a zrušení inverzního i grafického re - žimu; počet vynechaných řádek se řídí číslem na adrese 14; pokud je číslo vkládaného řádku rovno číslu na adrese 13 (délce stránky), text roluje.
CTRL N		0E zrušení grafického režimu
CTRL O		0F přepnutí do grafického režimu
CTRL P		10
CTRL Q		11
CTRL R		12 zrušení inverzního režimu
CTRL S		13 přepnutí do inverzního režimu
CTRL T		14
CTRL U		15
CTRL V		16
CTRL W		17
CTRL X	→	18 posuv kurzoru vpravo
CTRL Y	↑	19 posuv kurzoru nahoru
CTRL Z	↓	1A posuv kurzoru dolů
CTRL [1B v basicu-6 ruší režim AUTO
CTRL \	IC	1C vytvoření mezery mezi znaky na místě kurzoru; následujících n znaků se posune vpravo (n je na adrese 12, může být maximálně 80H = 128D); pokud je mezi znaky CR (kód 0D, na obrazovce me - zera), končí posuv tímto znakem.
CTRL]	DC	1D zrušení znaku na místě kurzoru; následujících n znaků se posune zpět (viz poznámku u IC).
CTRL ^	DL	1E
CTRL _		1F smazání obrazovky a návrat kurzoru na pozici 0,0

Řídicí znaky se generují výše uvedeným způsobem v pod programech
 F8AA - kód je v registru A, na stisk tlačítka se čeká,
 F8C9 - kód je v registru A a C, bez čekání na stisk tlačítka,
 F000 - kód je v registru C, příkaz se realizuje,
 v klidovém stavu, kdy monitor čeká na vstup příkazu (zobrazena
 vrána > a bliká kurzor); příkaz se provede.

Lze použít i podprogramy F62B a F643 přepínajících periférie.

Řídicí příkaz lze dále provést podprogramy

- F003 - kód příkazu nutno předem uložit do registru A,
- F007 - kód příkazu nutno předem uložit do registru C,
- F647 - kód příkazu musí následovat jako bajt za voláním podpro - gramu, tedy CD 47 F6 kód .

Kód řídicího příkazu může být i součástí vypisovaného textu v pod - programu F488, v basicu lze obdobně použít příkazu PRINT CHR\$(kód).

Obrazovka

Veškeré obvody pro výstup znaku na obrazovku jsou umístěny ve výměnných modulech VIDEO 32 (32 řádků po 32 znacích) nebo VIDEO 64 (32 řádků po 64 znacích). Uvedené moduly obsahují vlastní paměť RAM a znakové generátory.

Jednotlivé pozice na obrazovce jsou adresovány přímo adresovou sběrnicí mikroprocesoru nebo nepřímo, pomocí souřadnic kurzoru. Modul VIDEO 32 používá adres EC00 - EFFF (1 kB), modul VIDEO 64 pak adres E800 - EFFF (2 kB). Počáteční adresu obrazovkového pole můžeme též zjistit na adrese 0020 - 0021, počet znaků na řádek na adrese 001F. Další údaje se týkají většinou standardně dodávaného modulu VIDEO 32.

1. Přímý výstup na obrazovku

Znakový generátor přiřazuje přirozeným číslům 0 - FF 255D různých znaků. Kódy 0 - 1F dávají znaky grafické, 20 - 7F odpovídají zhruba kódu ASCII. Pokud je 7. bit kódu roven 1, zobrazí se generovaný znak inverzně, tj. černý na bílém pozadí. Např. kód

0100 0001 B = 41 H = 65 D odpovídá písmenu A,

1100 0001 B = C1 H = 193 D zobrazí A v inverzi.

Pokud kód 0 až FF vložíme na adresu odpovídající pozice obrazovky, znak se na tomto místě přímo zobrazí. Vyzkoušet si to můžeme např. substitucí kódu na zadanou adresu, tj.

BR S ECC0 SP 41 SP C1 ... ,

na začátku 6. řádku se zobrazí A normálně i v inverzi.

Více znaků lze zobrazit přesunem úseků paměti příkazem M, např.

BR M C8D7,CA33,ED00 CR

zobrazí od 8. řádku část tabulky klíčových slov BASICu 6. Všimněte si, že poslední znak každého slova je zobrazen v inverzi.

Obráceně lze programově testovat, zda na daném místě obrazovky je nebo není určitý znak. Je-li v registru HL adresa pozice na obrazovce, pak instrukcemi

MOV A,M

CPI 03

testujeme, zda na sledovaném místě je grafický znak "šipka směřující nahoru".

2. Nepřímý výstup na obrazovkuPodprogram F007

Tento podprogram používá pro označení místa na obrazovce tzv. kurzoru (ukazatele). Kurzor ukazuje blikáním na místo, na které se vytiskne další znak. Jeho poloha se určuje kromě adresy na obrazovce i souřadnicemi, odpovídajícími řádku (0 - 1F) a sloupci (0 - 1F).

Výchozí poloha v levém rohu náhara (adresa 0000) má souřadnice 0,0. 4

Informace o kurzoru najdeme na těchto adresách:

- 0007 ... 0 - kurzor bliká, jinak je blikání blokováno
- 000B ... kód znaku, na který ukazuje kurzor
- 000C ... nižší bajt adresy místa na obrazovce
- 000D ... vyšší bajt adresy místa na obrazovce
- 000E ... sloupec, ve kterém je právě kurzor
- 000F ... řádek, ve kterém je právě kurzor

Příklad:

Jestliže jsou na těchto adresách čísla

0007 000B 000C 000D 000E 000F

00 58 11 ED 11 08

ukazuje blikající kurzor v 17. sloupci (11H) 8. řádku na znak X.

Pro práci s kurzorem jsou důležité ještě tyto adresy:

0013 - zde je umístěna tzv. délka stránky, což je přirozené číslo od 1 do 1F. Udává číslo řádky, která se zobrazí na obrazovce jako poslední.

0014 - číslo na této adrese udává, o kolik řádek se posune kurzor při stisku tlačítka CR nebo při průběhu podprogramu odpovídajícímu řídicímu znaku 0D.

Podprogram F007 používá čísla, které je umístěno v C registru. Pokud je kód v C registru z intervalu 0 - 1F, je chápán jako řídicí znak (ne tedy jako znak grafický) a proběhne určitý podprogram související s tiskem na obrazovku.

Příklad:

Při řídicím znaku

19 se kurzor posune o 1 řádku nahoru,

1F se smaže obrazovka (v paměti jsou kódy 20 - prázdný znak) a kurzor se nastaví na pozici 0,0.

Pokud je kód v C z intervalu od 20 do 7F, pak

aa) zobrazí se na místě kurzoru znak odpovídající kódu, je-li nastaven normální režim (mód) počítače (na adresách 10 a 11 jsou nuly),

ab) není-li na adrese 10 nula a je-li kód z intervalu 40 až 5F, upraví se číslo odečtením 40 na kód grafického znaku, grafický znak se zobrazí,

ac) není-li na adrese 11 nula, upraví se 7. bit kódu na 1, tím se znak zobrazí v inverzi.

Dále se kurzor

ba) posune o jedno místo doprava, pokud nebyl v posledním sloupci,

bb) posune na začátek další řádky, pokud byl v posledním sloupci.

Pokud je číslo tohoto řádku rovno délce stránky (viz výše), text na obrazovce odroluje, tzn., všechny řádky se posunou nahoru o počet daný číslem na adrese 14.

Kódy 80 - FF v tomto podprogramu nepoužíváme.

5

Podprogramy F000 a F003

O podprogramu F003 platí vše, co bylo uvedeno o podprogramu F007. Kód znaku však musí být v A registru.

Podprogramem F000 vypíšeme na obrazovku znak generovaný stiskem tlačítka na klávesnici (včetně tlačítek řídicích).

Podprogram F647

Podprogram F007 vyžadoval kód znaku uložený v registru C. Pro zobrazení dvojtečky bychom tedy použili

```
MVI C, 3A
```

```
CALL F007 , délka podprogramu je 5 bajtů.
```

Podprogram F647 umožňuje tuto délku zkrátit na bajty 4; kód znaku, který chceme zobrazit, dáváme za volání podprogramu, tedy

```
CALL F647
```

```
3A .
```

Podprogram F5B0

Tento podprogram je ekvivalentní delšímu programu

```
MVI C, 0D
```

```
CALL F007
```

nebo

```
CALL F647
```

```
0D .
```

Kód 0D vyvolá podprogram pro odřádkování, počet řádků se řídí bajtem na adrese 14.

Podprogram F488

Tento podprogram umožňuje výpis souvislého textu, umístěného ve formě kódů v tabulce. Adresa počátku tabulky se umísťuje do páru HL, konec tabulky označíme tak, že poslednímu znaku, který má být vypsan, změním 7. bit na 1 (znak má kód odpovídající inverznímu zobrazení, zobrazí se však normálně). Text se začne vypisovat od pozice dané polohou kurzoru, součástí textu mohou být i kódy řídicích znaků.

Příklad:

Obsahuje-li tabulka od adresy 3000 kódy

```
3000 1F 13 4E 41 5A 44 41 52 21 8D
```

```
      N A Z D A R !
```

pak podprogram

```
LXI H, 3000
CALL F488
```

smaže obrazovku (kód 1F),

přepne do režimu inverzního zobrazení (kód 13),

vypíše NAZDAR ! a

odřádkuje (kód 0D, zde se 7. jednotkovým bitem 8D ukončuje výpis).
Při odřádkování se zruší inverzní režim.

Podprogramy F5D0 a F5D5

Tyto podprogramy umožňují vypsat na obrazovku obsah registrového páru HL a střadače A v šestnáctkovém tvaru. Je-li např. v HL číslo 24A0, převede podprogram F5D5 jednotlivé cifry do ASC kódu a postupně je zobrazí na obrazovce od místa daného kurzorem (budou to kódy 32, 34, 41, 30).

Příklad:

Podprogram LXI H, 001C

CALL F5D0

CALL F647

20

MOV A,M

CALL F5D5

zobrazí na obrazovce

001C 21 (bajt na adrese 1C oddělený mezerou od adresy).

Poznámka:

Podprogramy F000, F003, F488, F5D0, F5D5, F5B0 a F647 se dají použít i pro tiskárnu, děrovač a snímač děrné pásky. Stačí změnit hodnotu I/O bajtu na adrese 0003 (vyhodnocují se 2 nejnižší bity). Pro výpis HL registru a pro výpis střadače na tiskárnu slouží též podprogramy F5FA a F5FF, u kterých se vyhodnocují dva nejvyšší bity I/O bajtu.

Přepínání periférií pomocí I/O bajtu na adrese 0003

Řada podprogramů monitoru je řešena tak, aby byla společná pro různé vstupní a výstupní periférie.

Jako vstupní periférie používá monitor klávesnici, snímač děrné pásky a vstup z magnetofonového pásku, výstupními perifériemi zde jsou obrazovka, děrovač pásky, tiskárna a magnetofon. Monitor dále počítá s případnými dalšími perifériemi uživatele a umožňuje skoky na vlastní obslužné podprogramy.

V popisovaných pod programech se vyhodnocují jednotlivé bity čísla na adrese 0003 a dle výsledku se směřují příslušné skoky.

Podprogram F62B:

I/O bajt	skok na adresu	podprogram
.....00	7FC2	neobsazen
.....01	F8AA	kód stisknutého tlačítka do A
.....10	F6EB	kód ze snímače děrné pásky do A
.....11	7FC5	neobsazen

Podprogram F689:

I/O bajt	skok na adresu	podprogram
....00..	F7D4	neobsazen
....01..	F6EB	kód ze snímače děrné pásky do A
....10..	F51F	znak z magnetofonové pásky do A
....11..	7FD7	neobsazen

Podprogram F64B:

I/O bajt	skok na adresu	podprogram
.....00	7FC8	neobsazen
.....01	F007	výpis znaku z C registru na obrazovku
.....10	F6A3	přepínač podle bitů A7,A6
.....11	7FC3	neobsazen

Podprogram F66B:

I/O bajt	skok na adresu	podprogram
..00....	7FCE	neobsazen
..01....	F6D9	výstup na děrovač z C registru
..10....	F56B	výstup na magnetofon z C registru
..11....	7FD1	neobsazen

Podprogram F6A3:

I/O bajt	skok na adresu	podprogram
00.....	7FDA	neobsazen
01.....	F64B	přepínač podle bitů A1,A0
10.....	F6C6	výpis znaku z C registru na tiskárnu
11.....	7FDD	neobsazen

Po inicializaci počítače se nastavuje I/O bajt na hodnotu 69H = 0110 1001 B, tj. vstupními perifériemi jsou klávesnice a magnetofon, výstupní je obrazovka.

Klávesnice používá brány 84, 85, 86 a 87 (obvod 8255), děrovač, snímač děrné pásky a tiskárna brány F9, FA, FB, pro kreslič a pro modul VIDEO 64 jsou rezervovány adresy F0, F1, F2, F3, F4 a FE.

Dále jsou obsazeny adresy 80 obvodem 3212 a adresy 88, 89 programovatelným řadičem přerušení 8259.

Adresy jednotlivých pozic na obrazovce (tabulka číslo 32)

hexadecimálně					desítkově		
	00	0F	10	1F		00	31
00	EC00 ...	EC0F	EC10 ...	EC1F	00	60416 ...	60447
01	EC20 ...	EC2F	EC30 ...	EC3F	01	60448 ...	60479
02	EC40 ...	EC4F	EC50 ...	EC5F	02	60480 ...	60511
03	EC60 ...	EC6F	EC70 ...	EC7F	03	60512 ...	60543
04	EC80 ...	EC8F	EC90 ...	EC9F	04	60544 ...	60575
05	ECA0 ...	ECAF	ECB0 ...	ECBF	05	60576 ...	60607
06	ECC0 ...	ECCF	ECD0 ...	ECDF	06	60608 ...	60639
07	ECE0 ...	ECEF	ECF0 ...	ECFF	07	60640 ...	60671
08	ED00 ...	ED0F	ED10 ...	ED1F	08	60672 ...	60703
09	ED20 ...	ED2F	ED30 ...	ED3F	09	60704 ...	60735
0A	ED40 ...	ED4F	ED50 ...	ED5F	10	60736 ...	60767
0B	ED60 ...	ED6F	ED70 ...	ED7F	11	60768 ...	60799
0C	ED80 ...	ED8F	ED90 ...	ED9F	12	60800 ...	60831
0D	EDA0 ...	EDAF	EDB0 ...	EDBF	13	60832 ...	60863
0E	EDC0 ...	EDCF	EDD0 ...	EDDF	14	60864 ...	60895
0F	EDE0 ...	EDEF	EDF0 ...	EDFF	15	60896 ...	60927
10	EE00 ...	EE0F	EE10 ...	EE1F	16	60928 ...	60959
11	EE20 ...	EE2F	EE30 ...	EE3F	17	60960 ...	60991
12	EE40 ...	EE4F	EE50 ...	EE5F	18	60992 ...	61023
13	EE60 ...	EE6F	EE70 ...	EE7F	19	61024 ...	61055
14	EE80 ...	EE8F	EE90 ...	EE9F	20	61056 ...	61087
15	EEA0 ...	EEAF	EEB0 ...	EEBF	21	61088 ...	61119
16	EEC0 ...	EECF	EED0 ...	EEDF	22	61120 ...	61151
17	EEE0 ...	EEEF	EEF0 ...	EEFF	23	61152 ...	61183
18	EF00 ...	EF0F	EF10 ...	EF1F	24	61184 ...	61215
19	EF20 ...	EF2F	EF30 ...	EF3F	25	61216 ...	61247
1A	EF40 ...	EF4F	EF50 ...	EF5F	26	61248 ...	61279
1B	EF60 ...	EF6F	EF70 ...	EF7F	27	61280 ...	61311
1C	EF80 ...	EF8F	EF90 ...	EF9F	28	61312 ...	61343
1D	EFA0 ...	EFAF	EFC0 ...	EFCF	29	61344 ...	61375
1E	EFC0 ...	EFCF	EFD0 ...	EFDf	30	61376 ...	61407
1F	EFE0 ...	EFEF	EFF0 ...	EFFF	31	61408 ...	61239

Informace o kurzoru najdeme na těchto adresách:

- 0007 ... 0, - kurzor bliká, jinak je blikání blokováno
- 000B ... kód znaku, na který ukazuje kurzor
- 000C ... nižší bajt adresy místa na obrazovce
- 000D ... vyšší bajt adresy místa na obrazovce
- 000E ... sloupec, ve kterém je právě kurzor
- 000F ... řádek, ve kterém je právě kurzor

Klíčovým stavem monitoru budeme označovat čekací smyčku na adrese F1DF, kdy se na obrazovce (nebo jiné výstupní periférii) zobrazí znak > (vrána) a kdy blikající kurzor oznamuje čekání na vstup některého příkazu monitoru.

Příkazy monitoru zadáváme z periférie, kterou určují 2 nejnižší bity I/O bajtu v podprogramu F62B; nemusí to tedy být vždy klávesnice počítače. Výstupní periférie se vyhodnocuje podprogramem F64B, může to být kromě obrazovky tiskárna nebo další zařízení uživatele. Po inicializaci počítače je nastaven I/O bajt (adresa 0003) na hodnotu

69H = 0110 1001 B, což odpovídá

vstupu znaku z klávesnice do C registru se současným výpisem na obrazovku.

Příkazy monitoru jsou bez parametru nebo s jedním či s více parametry. Parametry dále označované jako a značí adresu v délce 2 bajtů vkládanou jako 4 šestnáctkové číslice. Pokud vložíme více číslic, vyhodnocují se pouze poslední čtyři (můžeme tedy opravit chybně vložený údaj). Pokud adresu nevložíme, pracuje podprogram monitoru s adresou 0000 (s výjimkou příkazů G a C).

Příkazy označené jako d značí číslo v délce 1 bajtu vkládané jako 2 šestnáctkové číslice. Vložíme-li číslic více, vyhodnotí se pouze 2 poslední cifry.

Parametry oddělujeme čárkou nebo mezerou. Nevložíme-li po čárce nebo po mezeře parametr, dosazuje se automaticky nula.

Příkazy bez parametru:

R

... RETURN - skok na podprogram F484

návrat na adresu uloženou v registrech 1D-1E; při zasunutém modulu BASIC-6 je zde adresa teplého startu basicu CAD6. Uživatel zde může umístit vlastní adresu skoku na jiný podprogram.

BR

... BREAK - skok na podprogram F792

Příkaz lze použít při probíhajícímu programu (včetně basicu) pro zastavení (přerušování) výpočtu. Na obrazovce se vypíše za hrádka # s adresou, před kterou se výpočet zastavil; tato adresa se zároveň ukládá do registrů 15-16, dále se uloží obsahy všech registrů mikroprocesoru (A, F, H, L, B, C, D, E, SP, PC), odřadí se a počítač přejde do klidového stavu. Nyní lze použít všech příkazů monitoru, modifikovat (upravit) obsahy některých registrů (včetně registrů mikroprocesoru) a pak pokračovat v přerušovaném programu příkazem G s uvedením na začátku vypsání adresy přerušování. Uvedený postup lze použít i pro pokračování přerušovaného programu v basicu.

CTEL ■

... smazání obrazovky, návrat kurzoru na pozici 0,0
odřádkování a přechod do klidového stavu

↵

... HOME

návrat kurzoru na pozici 0,0, odřádkování a přechod do klidového stavu

X CR

... CHANGE - skok na podprogram F2C4

Tímto příkazem vypíšeme na periférii určenou I/O bajtem v podprogramu F64B čísla uložená na adresách

23 22 25 24 27 26 29 28 2B 2A 2D 2C .

Čísla na těchto adresách se využívají dvojitým způsobem:

1. Při startu programu příkazem C nebo G se uloží po řadě do registrů

A F B C D E H L SP

mikroprocesoru; na zbývajících 2 adresy se ukládá startovací adresa uvedená v příkazu G nebo C. Výše uvedená čísla tedy tvoří výchozí (počáteční) hodnoty, se kterými mikroprocesor pracuje. Výchozí hodnoty můžeme změnit přeepsáním dat na těchto adresách příkazy

XA, XF, XB, XC, XD, XE, XH, XL, XS a XP ,

ve kterých druhé písmeno označuje název registru mikroprocesoru. Po vložení příkazu se zobrazí obsah registru a pomlčka, blikající kurzor vyzývá k vložení žádaného čísla. Vložené číslo potvrdíme

- a) tlačítkem CR, program se vrací do klidového stavu monitoru,
- b) tlačítkem SP, zobrazí se obsah dalšího registru s pomlčkou, postup můžeme opakovat pro další registr.

Bez vloženého čísla nemají tlačítka CR a SP na obsahy registrů vliv.

2. Po ukončení programu vyvolaného příkazem C se na tyto adresy přesunují stávající obsahy registrů mikroprocesoru. Na adrese 2C-2D zůstává původní startovací adresa.

Příkazu X můžeme použít i programově jako podprogramu F2CD - výpis obsahů registrů na obrazovku.

Příklad:

Chceme zjistit obsah registrů mikroprocesoru v určitém místě programu. Uložíme tedy na toto místo instrukci RST 6 - uložení obsahu čítače PC do zásobníku a skok na adresu 0030 (kód F7). Zde umístíme podprogram

CD 0D F6 CALL F60D ; uložení registrů na adresy 22 - 2B
C3 CD F2 JMP F2CD ; výpis a skok do klidového stavu monitoru.

G CR

... GOTO - skok na podprogram F405

Příkaz nejdříve uloží do registrů mikroprocesoru čísla z adres 22 až 2D takto:

22 23 24 25 26 27 28 29 2A 2B 2C 2D
F A C B E D L H SP PC

Program pak pokračuje od adresy uložené do registru PC, startujeme tedy příkazem G program od adresy uložené v buňkách 2C-2D. Po ukončení programu instrukcí k návratu program náhodně zběhne na adresu pod vrcholem zásobníku.

Příkaz startuje program stejně jako příkaz G. Po ukončení programu instrukcí k návratu však navíc uloží okamžité hodnoty registrů na výše uvedené adresy 22 - 2B (PC se neukládá) a obnoví se klidový stav monitoru. Příkazem X si tyto hodnoty můžeme zobrazit.

Pozor na adresu vrcholu zásobníku v buňkách 2A-2B! Při nevhodném čísle může systém havarovat, i když spouštěný program je správný. Před použitím příkazu C je lépe zobrazit si výchozí hodnoty příkazem X a případně vrchol zásobníku upravit příkazem XS.

Tímto příkazem nahrajeme do paměti počítače data z periférie určené I/O bajtem v podprogramu F689 (magnetofon, děrná páska). Data jsou nahrávána po blocích tak, jak byla uložena příkazem W. Pokud byla uložena i startovací adresa a3, pokračuje program od této adresy, byla-li a3 nulová, přejde počítač do klidového stavu monitoru.

Na začátku příkazu L se blokuje přerušování 50 Hz, posune se kurzor tak, aby nebyl na řádkách 0 až 8, na kterých se zobrazují nahrávaná data, a nastaví se bit C1 brány 86 ovládající start a stop magnetofonu (viz poznámku u příkazu W). Program pak vyčkává na označení počátku bloku (kód 3A - dvojtečka), které se zobrazí jako první. Pak následuje délka bloku (50 a méně), adresa prvního nahrávaného bajtu, 00, blok 50 bajtů, kontrolní součet a odřádkování. Nahráný kontrolní součet se porovná se součtem nahrávaných bitů, pokud se součty neshodují, přejde počítač do klidového stavu monitoru (se zobrazeným ?). V kladném případě se nahrává další blok. Protože v příkazu W končil blok bajtem 0A (návrat vozíku) na obrazovce nepoužívaným, zobrazí se na začátku dalšího bloku znak 0A - skobička.

Po nahrání posledního bloku se ještě zobrazí bajt 00, startovací adresa, 01 a kontrolní součet. Zastavení magnetofonu (C1=0) chybí.

Příkazu L můžeme použít i programově jako podprogramu F3BA, pokud uložíme do zásobníku 0 nebo adresu z příkazu L a (viz dále):

```
21 00 00 LXI H,0000
E5      PUSH H
C3 BA F3 JMP F3BA
```

Tímto způsobem lze nahrát do počítače více oddělených úseků uložených příkazem W.

Postup při špatném nebo při nespolehlivém nahrávání

Závady při nahrávání dat z magnetofonu do paměti počítače bývají následující:

1. Na obrazovce se objevují jiné znaky než šestnáctkové číslice. V tomto případě jsou data na kazetě nahrána s opačnou polaritou signálu. Změníme proto 7. bit čísla na adrese 1C na jedničku. Standardně zde bývá číslo 56 (po inicializaci počítače), změnou 7. bitu dostaneme číslo D6. Pro vstup signálu z magnetofonu se využívá bit C7 brány 86, se kterým program nejdříve utvoří exkluzivní součet s výše uvedeným bitem v buňce 001C. Výše uvedenou upravou změníme i hodnotu součtu a tím i tedy polaritu signálu.

2. Mezi správně nahrávanými číslicemi se občas objeví jiný znak; program přechází do klidového stavu monitoru (?).

Závada bývá způsobena jinou rychlostí magnetofonu proti rychlosti při nahrávání příkazem W, modulovaný kmitočet 1kHz se vyhodnocuje v nesprávný okamžik. Délce periody modulovaného kmitočtu odpovídá číslo 56 v již uvedené buňce 001C, kterým se řídí délka čekací smyčky v programu. Zvětšením nebo zmenšením tohoto čísla lze dosáhnout spolehlivého nahrávání.

3. Chybný znak se objevuje vždy na stejném místě.

V tomto případě byl program špatně nahrán, na pásku je vadné (hlučné) místo nebo byla znečištěna nahrávací hlava magnetofonu. Pokud používáme stereofonní magnetofon (např. M710 A), nahráváme vždy současně na obě stopy, spolehlivost záznamu se tak zvýší. Nutno spojit na straně magnetofonu (stačí v přívodní šňůře) špičky 1-4 a 3-5. Výrobce opravu nezajišťuje!

Příkazy s jedním parametrem:

C a CR ... CALL - skok na podprogram F3FC
O tomto příkaze platí vše, co bylo řečeno o příkazu C. Parametr a se však uloží do buněk 2C-2D jako startovací adresa.

G a CR ... GOTO - skok na podprogram F405
O tomto příkaze platí vše, co bylo řečeno o příkazu G. Parametr a se však uloží do buněk 2C-2D jako startovací adresa. Příkaz používáme tehdy, pokud chceme pokračovat v programu přerušeném tlačítkem BR nebo po jiném přerušení, pokud známe adresu, před kterou k přerušení došlo.

D a CR ... DISPLAY - skok na podprogram F385
Příkaz vypíše na periférii určenou I/O bajtem v podprogramu F6AB (na obrazovku, tiskárnu) adresu a a blok 8 dat od této adresy (dat může být i méně, adresa posledního údaje v řádce končí číslicí 7 nebo F); na další řádce výpis pokračuje následující adresou a blokem dat. Počet vypsaných řádek se řídí délkou stránky na adrese 13. Pro standardní délku stránky 1E je počet řádek 0E (14D), chceme-li vypsat n řádek (např. na tiskárnu), vložíme délku stránky 2n + 2. Pokračování výpisu způsobí stisk libovolného tlačítka, výpis ukončíme tlačítkem CR; vrátíme se tak do klidového stavu monitoru.

L a CR ... LOAD
O tomto příkazu platí vše, co bylo uvedeno o příkazu L bez parametru. Data se však ukládají do počítače na adresu o a větší. Příklad:
Příkazem W byl na periférii uložen blok od adresy 2000 až 2100. Příkazem L tento blok uložíme zpět do počítače na původní adresu. Příkazem L 1000 budou data na adresách 3000 až 3100.

S a SP ... SUBSTITUCE na adrese a -skok na podprogram F438
Příkaz vypíše na periférii určenou I/O bajtem v podprogramu F64B adresu a, obsah buněk na této adrese a pomlčka. Uvedený obsah buněk nyní můžeme
a) ponechat (tlačítka SP, ↑ CR),
b) změnit vložení bajtu (2 šestnáctkové číslice, při více číslicích se vyhodnotí poslední 2) a jeho potvrzením tlačítky jako v a).
Tlačítkem SP zvětšíme adresu a o 1 a uvedený postup můžeme opakovat, tlačítkem ↑ zmenšíme adresu o 1 (krokujeme v paměti zpět), tlačítkem CR se vrátíme do klidového stavu monitoru.

Příklad:
Na adresu 1C máme vložit číslo 02. Postup:
S 1C SP ,
zobrazí se
001C 21- , nyní napíšeme 2 SP , zobrazí se další řádek
001D D6-
Tlačítkem ↑ se můžeme vrátit a přesvědčit se, že skutečně
001C 02- odpovídá úloze. Tlačítkem CR vkládání ukončíme.

F a1, a2, d CR ... FILL - skok na podprogram F251

Tímto příkazem naplníme paměťový prostor počítače od adresy a1 do adresy a2 bajtem (číslem) d.

Příklad:

CTL M	mažeme obrazovku.
FEC00,EC1F,11 CR	na nultém řádku vodorovná čára (kód 11)
FEC40,EC5F,35 CR	na druhém řádku čísllice 5 (kód 35)
FEC60,EC7F,41 CR	na třetím řádku písmeno A (kód 41)

Příkazu F můžeme použít i programově jako podprogramu F258 - naplnění buněk od adresy HL do DE bajtem v C

M A1, A2, A3 CR ... MOVE - skok na podprogram F240

Tímto příkazem přesuneme blok dat od adresy a1 do adresy a2 na nové místo od adresy a3. Adresa a3 nesmí být adresou žádného bajtu z přesouvaného bloku, aby nedošlo k překrytí a přepsání některých dat.

Příklad:

F3000,306F,0 CR	adresovaný blok jsme naplnili nulami
D3000 CR	vypis bloku od adresy 3000
CR	
MF000,F02F,3000	přesun bloku dat F000 až F02F na 3000
D3000	vypis bloku, od adresy 3000 jsou místo nul data z adresy F000 (přesvědčí se)

Příkazu M můžeme použít i programově jako podprogramu F247 - přesun useků od HL do DE na místo od BC (adresy a1, a2, a3 musí být předem v párech HL, DE a BC)

W a1,a2,a3 CR ... WRITE - skok na adresu F260

Tímto programem nahrajeme blok dat (program) od adresy a1 na adresy a2 na periférii určenou I/O bajtem v podprogramu F66B (magnetofon, děrovač). Data se nahrávají v blocích po 80 (50H) bajtech.

Na konci programu se nahraje adresa a3; pokud není rovna nule, pokládá ji počítač za startovací adresu nahrávaného programu, tzn., že po uložení programu z periférie do počítače příkazem L se od této adresy program samočinně spustí. Pokud je a3 rovno nule, přejde počítač do klidového stavu monitoru.

Příkaz W má následující strukturu:

1. Start magnetofonu nebo děrovače; na pásce se vyděruje 64 prázdných znaků (podprogram F6FC).
2. Počátek bloku se vyznačí znakem o kódu 3A (dvojtečka).
3. Nahraje se 1 bajt označující délku bloku (50H znaků mimo bloku posledního); pro nahrání bajtu v A je určen podprogram F5E8).
4. Nahraje se adresa 1. bajtu v bloku (obsah registrů HL - podprogram F5E3).
5. Nahraje se nulový bajt.
6. Následují jednotlivá data bloku.
7. Nahraje se kontrolní součet vytvořený z jednotlivých dat bloku (data se sčítají do jednoho bajtu, ze kterého se utvoří dvojkový doplněk).
8. Konec bloku se vyznačí nahráním kódů 0D (odřádkování); pak následuje pauza, jejíž délka se řídí číslem na adrese 001C. Po inicializaci počítače zde bývá číslo 21, pro příkaz W zde předem dáváme číslo 02, nahrávání se tak podstatně zkrátí.
9. Nahraje se kód 0A - návrat vozu na tiskárně, na obrazovce bez významu.
10. Není-li nahrán poslední blok dat, opakuje se postup od bodu 2.

11. Nahraje se 00, startovací adresa a3, bajt 01, kontrolní součet a znovu bajty 0D (pauza), 0A. 14
12. Zastavení magnetofonu chybí.

Poznámka:

Pro záznam na magnetofon se používá nosný kmitočet 1kHz, který se přivádí na vstup magnetofonu a na bit C5 brány 86 počítače. Tento kmitočet je amplitudově modulován nahrávanými bity přiváděnými na bit C0 téže brány. Nahrávaný bajt se nejdříve upraví na dvě šestnáctkové číslice v kódu ASC, jejich bity se invertují a sedmý bit se vynuluje. Každý bajt se tedy nahraje jako 2 osmibitová čísla s nulovým 7. bitem (obdobně se i děrují).

Příklad:

8A se nejdříve rozloží na čísla 38 a 41 (viz tabulku ASC kódu) a upraví se na

38 = 0011 1000 ... 0100 0111
41 = 0100 0001 ... 0011 1110 .

Jednotlivými bity (od nejnižšího řádu) se moduluje postupně nosný kmitočet .

Bitem C1 brány 86 se ovládá spínací tranzistor určený pro dálkové ovládání magnetofonu (START, STOP) M1, bitem C2 se obdobně ovládá magnetofon M2. Protože naše magnetofony nemívají dálkové ovládání, není vývod spínacího tranzistoru připojen ke konektoru pro magnetofon, je však připraven uvnitř počítače k montáži. V příkazu W výrobce zapomněl na pauzu potřebnou k rozběhu magnetofonu, není tedy ovládání v této formě použitelné.

Doplňky k nahrávání:

Příkaz W se dá použít programově jako podprogram F267. Adresy a1, a2, a3 musí být předem po řadě v registrech HL, DE a BC.

Obsahuje-li program několik oddělených úseků, můžeme jej uložit na kazetu (děrovač) tak, aby mohly být zpět nahrány jediným příkazem L.

Příklad:

Program v basicu používá podprogramu ve strojovém kódu uloženého na adrese 2500 až 25A0. Chceme-li nahrát s tímto podprogramem současně i program v basicu, najdeme nejdříve adresu jeho konce na adrese D0-D1. Začátek oblasti, kterou basic používá, je vždy na adrese 45. Příkazem

S D0 SP SP ĆR

zjistíme např. adresu konce basicu 1345 (na adresách D0-D1 to budou bajty 45 13, tedy čísla v obráceném pořadí).

Dále si vytvoříme příkazem S 30 pomocný podprogram

0030 21 00 00 E5 C3 BA F3 ... programový skok na příkaz L .

Pak stačí nahrát dvěma příkazy W

W 30,1345,30

W 2500,25A0,CAD6

oba programové bloky za sebe. Při zpětném přehrávání příkazem L se nejdříve nahraje úsek basicu s pomocným podprogramem 30 až 1345, nahraný program se hned spustí od adresy 30, čímž dosáhneme stavu, jako kdyby byl opět použit příkaz L. Nahraje se tedy úsek 2500 až 25A0 a po nahrání začne běžet program od adresy CAD6; tato adresa se nazývá teplým startem basicu a zajistí nám takový stav počítače, jaký dostáváme při nahrávání programu příkazem MLOAD (vypíše se READY). Obdobně se dá nahrát více oddělených úseků.

Poznámka:

Na adrese D0-D1 nemusí být vždy adresa konce basicu, je zde začátek paměti proměnných. Při nahrávání to však nevádí, nahrává se však dále. O upravě adresy viz poznámku v kapitole o spolupráci s basicem

- F000 Znak z klávesnice na obrazovku při I/O bajtu = 69
jinak:.....00 skok na adresu 7FC2
.....01 z klávesnice na obrazovku
.....10 ze snímače děrné pásky na periférii, určenou
bity 00..... na adresu 7FDA
01..... nelze použít
10..... na tiskárnu
11..... na adresu 7FDD
.....11 skok na adresu 7FC5
- F003 Znak o kódu ASC v C registru na periférii určenou I/O bajtem
v podprogramu F64B
- F007 Znak o kódu ASC v C registru na obrazovku včetně řídicích znaků:
07 - F973 - pípnutí
08 - F06F - kurzor vlevo
09 - F0B3 - tabulátor 8 znaků
0C - F065 - kurzor na pozici 0,0 (HOME)
0D - F137 - odřádkování a zrušení grafického i inverzního režimu (CR)
0E - F104 - přepnutí z grafiky do normálního režimu
0F - F104 - přepnutí do grafického režimu
12 - F103 - přepnutí z inverze do normálního režimu
13 - F103 - přepnutí do inverzního režimu
18 - F09E - kurzor vpravo
19 - F0A5 - kurzor nahoru
1A - F0AB - kurzor dolů
1C - F0C0 - IC (vytvoření mezery mezi znaky na místě kurzoru a posun n znaků; n je na adrese 12; posun se zastavuje také na místě, kde je řídicí znak 0D)
1D - F109 - DC (smazání znaku na místě kurzoru s posuvem)
1F - F052 - mazání obrazovky (na adresách EC00-EFFF kód 20)
ostatní kódy (0,1,2,3,4,5,6,A,3,10,11,14,15,16,17,1B,1E) RET
- F1B3 Programování obvodu 8259 - programovatelného řadiče přerušení
- F1D4 Obnova vrcholu zásobníku (7FC2 pro paměť 32 kB) a čekání na vstup příkazů (vrána): R S C G L D X W M F.
Při stisku jiné klávese skok na
- F227 otazník, odřádkování a zpět na F1D4
- F240 M s uložením adres do zásobníku
- F247 M: přesun úseků od HL do DE na místo od BC
- F251 F s uložením adres a kódu znaku do zásobníku
- F258 F: naplnění úseku od HL do DE znakem v C
- F260 W s uložením adres do zásobníku
- F267 W: nahrání dat od adresy v HL do adresy v DE; startovací adresa v BC
- F2C4 X: výpis registrů na obrazovku
obsahy registrů na adresách
22 23 24 25 26 27 28 29 2A 2B 2C 2D
F A C B E D L H SP PC
- F2CD X: vlastní výpis
- F2ED X: modifikace obsahu registrů
- F385 D s uložením adresy do zásobníku
- F38A D: od adresy v HL
- F3B6 L s uložením adresy do zásobníku
- F3BA L: s adresou v zásobníku
Pro nahrání oddělených úseků použít program
W a1,a2,a3 ... a3: 21 00 00 E5 C3 BA F3

- F3FC C s vložením adresy nebo bez; do zásobníku uloží návratovou adresu F432 (uložení registrů mikroprocesoru)
- F405 G s vložením adresy nebo bez; neukládá registry při RET, možnost zběhnutí programu při návratu
- Při C i G se registry mikroprocesoru naplní daty z 22-2B
- F432 uložení registrů na adresy 22-2B, skok do klidového stavu vlastní uložení na adrese F60D
- F438 S: substituce s udáním adresy
- F484 R: návrat na adresu uloženou v 1D-1E (teplý start basicu)
- F488 Výpis textu v tabulce, adresa počátku tabulky v HL, poslední znak textu v inverzi (A7=1), periférie podle F64B
- F495 Výpis znaku podle ASC kódu v A, periférie podle F64B
- F49B INX H a porovnání HL větší DE; v případě splnění CY=1
- F4A0 Porovnání HL větší DE; v případě splnění je CY=1
- F4A5 Převod ASC znaku v A na HEX číslo; pokud v A není číslo, je CY = 1
- F4B7 Vyhodnocení SP , CR
pro SP , CR je ZR=1
pro CR je navíc CY=1
- F4C3 HL=0; vstup čísla z klávesnice do HL s rotací HEXA čísel
návrat při CR , SP
vyhodnocují se 4 naposled vložené číslice
při CR je CY = 1
při nečíselném znaku ?, vrána, skok do MONITORU
- F4C9 Vstup z C do HL a vstup dalších čísel z klávesnice jako v F4C3
- F4E5 Vstup parametrů do zásobníku: v B počet parametrů, návratová adresa vždy na vrcholu zásobníku, pokud není po posledním parametru CR, zobrazí se ? a vrána
- F4F4 Vstup 2 číslic v ASC kódu z periférie určené I/O bajtem v podprogramu F689 a utvoření 1 bajtu v A
- F50B Vstup bajtu z periférie určené I/O bajtem v podprogramu F689 do A
- F51F Vstup jednoho bajtu z magnetofonu do A
- F56B Výstup bajtu z C registru na magnetofon
- F5A2 Pauza po dobu n.20 ms; n je číslo na adrese 001C převedené do desítkové soustavy
- F5A5 Pauza jako v podprogramu F5A2, n je číslo v A
- F5B0 Odřádkování na periférii určené I/O bajtem v podprogramu F64E
- F5B9 Odřádkování na periférii určené I/O bajtem v podprogramu F6AB
- F5D0 Výpis HL hexadecimálně (4 číslice) podle F64B
- F5D5 Výpis A hexadecimálně (2 číslice) podle F64B
- F5E3 Výstup (nahrání) HL na periférii dle F66B
- F5E8 Výstup (nahrání) A na periférii dle F66B
- F5FA Výpis HL dle F6AB (na tiskárnu)
- F5FF Výpis A dle F6AB (na tiskárnu)
- F60D Uložení registrů mikroprocesoru na adresy 22-2B, SP na 7FC2
- F62B Přepínač periférií podle 2 nejnižších bitů I/O bajtu
- F647 Podprogram pro výstup bajtu následujícího za voláním tohoto podprogramu na periférii určenou podprogramem F64B

2

- F64B Výstup na periférii určenou 2 nejnižšími bity I/O bajtu
 F667 Podprogram pro výstup bajtu následujícího za voláním tohoto podprogramu na periférii určenou podprogramem F66B
 F66B Výstup na periférii určenou 4. a 5. bitem I/O bajtu
 F689 Bajt z periférie určené 2. a 3. bitem I/O bajtu do A
 F6A7 Podprogram pro výstup bajtu následujícího za voláním tohoto podprogramu na periférii určenou podprogramem F6AB
 F6AB Výstup na periférii určenou 6. a 7. bitem I/O bajtu

Poznámka:

Formát uvedených podprogramů je

- CD 47 F6 bajt
- CD 67 F6 bajt
- CD A7 F6 bajt .

Vystupující znak v podprogramech F64B, F66B a F6AB je uložen v C.

- F6C6 Výstup na tiskárnu z C registru
 F6D9 Výstup na děrovač z C registru
 F6EB Vstup z děrné pásky do A
 F6FC Start magnetofonu nebo děrovače při nahrávání
 F711 Start magnetofonu a příprava obrazovky při vstupu dat do počítače
 F748 Podprogram při přerušení 50 Hz - blikání kurzoru, inkrementace čítače na adresách 8,9 a A, časování klávesnice
 F792 Podprogram pro přerušení tlačítkem BR - návratová adresa do buňky 0015-16 a na obrazovku, uložení registrů mikroprocesoru, skok do klidového stavu monitoru
 F800 (F818) Inicializace počítače po zapnutí nebo po stisku RES
 F803 (F8AA) Znak z klávesnice do A s čekáním
 F806 (F6EB) Vstup z děrné pásky do A
 F809 (F007) Výstup na obrazovku z C registru
 F80C (F6D9) Výstup z C registru na děrovač
 F80F (F6C6) Výstup z C registru na tiskárnu
 F812 (F51F) Vstup z magnetofonu do A
 F815 (F56B) Výstup z C registru na magnetofon
 F8AA Znak z klávesnice do A s čekáním a s pípnutím
 F8C9 Znak z klávesnice do C bez čekání (bez pípnutí)
 F973 Pípnutí s délkou v 18 a výškou v 17
 F995 Rolování textu na obrazovku, pokud je číslo zobrazovaného řádku rovno číslu v 0013 (délce stránky)
 F9C2 Konec paměti EPROM monitoru
 F9C3 - FFFF
 63D (1596 D) volných buněk v paměti EPROM k dispozici uživateli

Počítač IQ-151 používá méně známý programovatelný řadič přerušeni 8259 umožňující adresovatelnost podprogramů pro zpracování přerušeni, maskování jednotlivých žádostí o přerušeni a řadu režimů pro zpracování 8 žádostí o přerušeni.

Obvod 8259 (K 580 IK 59) je připojen ke sběrnici mikropočítače jako brána s adresou 88 a 89. Z 8 přerušovacích vstupů jsou počítačem využity 3 pro přerušeni způsobené kmitočty 16 kHz, 50 Hz a tlačítkem BR (brejk), ostatních 5 vstupů je vyvedeno na sběrnici přes oddělovací hradla (špičky 39 - 43 konektoru).

Po inicializaci počítače (zapnutím nebo tlačítkem RES) je řadič programován 3 příkazovými slovy ICW

```
ICW1: MVI A, F7
      OUT 88
ICW2: MVI A, 7F
      OUT 89
ICW4: MVI A, 00
      OUT 89
```

na režim s těmito vlastnostmi:

- Přerušeni je způsobeno náběžnou hranou signálu přivedeného na příslušný přerušovací vstup.
- Po převzetí žádosti o přerušeni se automaticky potlačují všechny žádosti o přerušeni téže a všech nižších úrovní.
- Na sběrnici počítače se zasílá instrukce CALL s uvedením adresy podprogramu takto (instrukce ke skoku v následující tabulce se na tyto adresy ukládají při inicializaci počítače):

úroveň	špička kon.	adresa CALL	podprogram	význam
0	39	7FE0	C3 00 F8 FF	inicializace počítače
1	40	7FE4	C3 00 F8 FF	inicializace počítače
2	41	7FE8	C3 00 F8 FF	inicializace počítače
3	42	7FEC	C3 00 F8 FF	inicializace počítače
4	43	7FF0	C3 00 F8 FF	inicializace počítače
5	není	7FF4	C3 92 F7 FF	přerušeni tlačítkem BR - skok na podprogram F792
6	není	7FF8	C3 48 F7 FF	přerušeni kmitočtem 50 Hz skok na podprogram F748
7	není	7FFC	C3 00 F8 FF	inicializace počítače

Po dokončení podprogramu dojde k návratu na adresu původního programu, před kterou došlo k přerušeni.

- Po návratu z podprogramu vyvolaného instrukcí CALL zůstávají žádosti o přerušeni stejné i nižší úrovně potlačeny, pokud ne - naprogramujeme konec přerušeni příkazem

```
OCW2: MVI A, 20
      OUT 88
```

- Jednotlivé žádosti o přerušeni se dají zamaskovat nebo povolit operačním slovem

```
OCW1: MVI A, MASKA
      OUT 89
```

kde bajt MASKA vytvoříme tak, že na bitech odpovídajících potlačení úrovně vložíme 1, na bity povolených úrovní přerušeni 0.

Příklad:

Při inicializaci řadiče přerušeni se vkládá maska

```
9F = 1 0 0 1 1 1 1 1 ,
```

jsou tedy povoleny úrovně přerušeni 6 a 5, odpovídající přerušeni 50 Hz a tlačítku BR. Ostatní vstupy jsou zamaskovány, přerušeni 50 Hz má přitom vyšší úroveň než přerušeni BR.

Uživatelské žádosti o přerušeni zajistíme tak, že na odpovídající špičku konektoru sběrnice přivedeme uroveň 0 přes hradlo s otevřeným kolektorem. Přerušeni urovní 5, 6 a 7 jsou realizována v zapojení počítače.

Obvod 8259 je složitý obvod, obsahující mimo jiné

1. registr žádostí o přerušeni IRR, na jehož bity přivádíme žádost o přerušeni,
2. registr masky přerušeni IMR, do kterého vkládáme masku instrukcí

MVI A, MASKA
OUT 89

(1 maskuje, 0 povoluje přerušeni odpovídající úrovní),

3. registr žádostí v obsluze ISR, ve kterém nastavené bity indikují právě obsluhovaná přerušeni.

Spolupráce řadiče přerušeni s mikroprocesorem během zpracování přerušeni je následující:

1. Žádost o přerušeni nastaví odpovídající bit v IRR registru obvodu 8259, pokud není daná uroveň přerušeni zamaskována.
2. Pokud má tato žádost vyšší prioritu než úroveň přerušeni právě obsluhovaná, zašle obvod 8259 do mikroprocesoru signál INT.
3. Má-li mikroprocesor povolené přerušeni, potom je žádost potvrzena vysláním pulzu INTAN řadičem 8228.
4. Po obdrženi signálu INTAN je v ISR registru nastaven bit odpovídající žádosti s nejvyšší prioritou a odpovídající bit v IRR registru je nulován. Obvod 8259 zašle po datové sběrnici do mikroprocesoru operační kód instrukce CALL - CD.
5. Systémový řadič 8228 zašle po obdrženi kódu CD dva další pulzy INTAN, které umožní zaslání 2 bajtů adresy podle výše uvedené tabulky.
6. Proběhne podprogram volaný instrukcí CD odpovídající obsluhované úrovni. Po návratu z podprogramu pokračuje přerušeni program.
7. Ve zvoleném režimu práce zůstává bit v ISR registru nastaven do té doby, než zašleme příkaz konce přerušeni

MVI A, 20
OUT 88.

Příklad:

Vytvořme systém, který by čítal impulzy přiváděné do počítače zvnějšku nezávisle na právě probíhajícím programu.

Obvodové řešení:

Výstup čítače připojíme přes hradlo s otevřeným kolektorem na špičku konektoru 39 (vstup INTON) a 5-6 (0 V).

Programové řešení:

Počet impulzů ukládáme jako 2 bajty na adresy 0000-0001, podprogram pro obsluhu přerušeni úrovní 0 vytvořme od adresy 002E. Na začátku programu, jehož součástí bude vyhodnocování údaje čítače, pak kromě nulování čítače uložíme na adresu 7FE0 (adresa instrukce CALL při přerušeni úrovní 0) příkaz ke skoku C3 2E 00 - JMP 002E (instrukce C3 je tam vytvořena již při inicializaci počítače).

Ve strojovém kódu bude tento úvod

21 00 00	LXI H, 0000	; nulování páru HL
22 00 00	SHLD 0000	; nulování bajtů na adresách 0000-0001
21 2E 00	LXI H, 002E	; adresa podprogramu do HL a její
22 E0 7F	SHLD 7FE0	; uloženi do tabulky přerušeni úrovní 0
3E 9E	MVI A, 9E	; odmaskování přerušeni úrovní 0
D3 89	OUT 89	; do IMR registru obvodu 8259

basicu bude tento program

POKE 0,0:POKE 1,0

POKE HEX(7FE1),46:POKE HEX(7FE2),0

OUT HEX(89),158

Podprogram pro obsluhu přerušeni úrovně 0 vytvořime ve strojovém kodu:

```
002E 2A 00 00  LHL 0000  ; stav čítače do HL
0031 23        INX H    ; zvětšení HL o jednu
0032 22 00 00  SHLD 0000  ; uložení stavu čítače do HL
0035 3E 20     MVI A, 20
0037 D3 88     OUT 88   ; programování ukončení přerušeni
0039 C9        RET      ; návrat z podprogramu
```

Každá změna úrovně 1 na 0 na vstupu konektoru 39 způsobí přerušeni programu a zvětšení čísla na adrese 0000-0001 o 1.

Podprogram pro přerušeni 50 Hz každých 20 ms zajišťuje (pokud není přerušeni zamaskováno, např. v příkaze L)

- a) zvětšení čísla v registrech 000A, 0009, 0008 o 1,
- b) obsluhu blikání kurzoru, pokud není blikání potlačeno,
- c) ošetření zámků právě stisknutého tlačítka na klávesnici, případně vytváří interval mezi opakovaným vstupem znaku při trvale stisknutém tlačítku.

Pokud zamaskujeme toto přerušeni, zablokujeme klávesnici.

Přerušeni 50 Hz můžeme využít i jako uživatelé pro vlastní program (např. blikání na určitém místě obrazovky, průběžné zobrazení času nebo vytvoření jiného časovače), jestliže obdobně jako v předcházejícím příkladě upravíme v tabulce přerušeni instrukci JMP F748 skokem na vlastní podprogram. Pozor, tento podprogram musíme vždy ukončit skokem na původní podprogram F748. Na konci celého programu upravíme tabulku přerušeni do původního stavu.

Podprogram pro přerušeni programu tlačítkem BR uloží obsahy registrů mikroprocesoru a adresu, před kterou k přerušeni došlo. Počítač pak přejde do klidového stavu monitoru (podrobněji viz o příkazech monitoru).

Další režimy činnosti řadiče přerušeni

Obvod 8259 rozeznává 2 druhy příkazových slov zasílaných z mikroprocesoru:

1. Inicializační příkazová slova ICW1, ICW2 a ICW4 zasílaná po řadě při inicializaci počítače. Volí se jimi adresa počátku tabulky přerušeni, způsob volby žádosti o přerušeni a způsob ukončení přerušeni. Slova použitá při inicializaci počítače IQ-151 již byla uvedena. Zvolený režim lze však upravit takto:

- a) Upravíme-li slovo ICW1 na

```
ICW1: MVIA, FF
      OUT 88 (ICW2 a ICW4 ponecháme),
```

bude počítač pracovat v režimu přerušeni úrovně (ne v režimu přerušeni náběžnou hranou) signálu přivedeného na přerušovací vstup.

- b) Pokud upravíme slovo ICW4 na

```
ICW4: MVI A, 02
      OUT 89
```

naprogramujeme automatický konec přerušeni. V tomto režimu je automaticky nulován odpovídající bit v ISR registru hned po vyslání instrukce CALL; nemusíme tedy konec přerušeni programovat operačním slovem OCW2.

2. Operační příkazová slova OCW1, OCW2 a OCW3 zasílaná kdykoliv po inicializaci. Nastavují se jimi masky přerušeni, rotace priorit přerušeni a priority při ukončení přerušeni. Poznali jsme již slovo

OCW1: MVI A, MASKA
OUT 89

kterým zamaskujeme (bit=1) nebo povolíme (bit=0) příslušnou žádost o přerušeni.

Další operační slova jsou:

OCW2: Toto slovo zasíláme na bránu 88 a určujeme jím rotaci priorit přerušeni a konec přerušeni včetně nastavení úrovně. Formát tohoto slova je následující:

D7 D6 D5 D4 D3 D2 D1 D0
R SL EOI 0 0 L2 L1 L0

Bity R, SL, EOI určují rotaci priorit přerušeni a konec přerušeni takto:

R	SL	EOI	význam
0	0	1	příkaz nespecifikovaného konce přerušeni
0	1	1	příkaz specifikovaného konce přerušeni
1	0	1	rotace priorit při nespecifikovaném konci přerušeni
1	0	0	nastavení rotace při automatickém konci přerušeni
0	0	0	rušení rotace při automatickém konci přerušeni
1	1	1	rotace priorit při specifikovaném konci přerušeni
1	1	0	příkaz nastavení priority
0	1	0	bez významu

Automatický konec přerušeni je programován v úvodu slovem ICW4= 02. Při ICW4= 00 (normální režim IQ-151) je nutno konec přerušeni zajistit příkazem OCW2, ve kterém je EOI = 1. Konec přerušeni zde může být

- a) specifikovaný, kdy bity L2,L1,L0 určují úroveň, ve které má dojít ke konci přerušeni (nuluje se příslušný bit v ISR registru,
- b) nespecifikovaný, kdy se nuluje v ISR bit odpovídající přerušeni s nejvyšší prioritou.

Rotace priorit umožňuje během činnosti obvodu měnit priority přerušeni tak, že

- a) naposledy obsluhované úrovni přerušeni přiřazena priority nej - nižší,
- b) úroveň, které má být přiřazena nejnižší priority, je určena bity L2,L1,L0.

Rotace priorit je cyklická a priority vzrůstají se snižujícím se číslem úrovně přerušeni.

Operačním slovem OCW3 (zde neuvédíme) lze zajistit čtení registrů IRR a ISR a nastavení režimu speciální masky, kdy zamaskovaná úroveň blokuje všechna přerušeni nižší priority.

Masku přerušeni lze zjistit instrukcí
IN 89 ; obsah registru IMR do A .

Přehled bran mikropočítače IQ 151

80 obvod 3212

bit 0 - přepínání paměti EPROM při inicializaci
(na adresu F800)

bit 1, 2 - použity pro vnitřní řízení

bit 3, 4, 5, 6, 7 - volné pro uživatele

84 kanál A obvodu 8255 - dekódování klávesnice

85 kanál B obvodu 8255 - dekódování klávesnice

86 kanál C obvodu 8255

bit C7 - vstup z magnetofonu 1 přes zesilovač
- tlačítko FBbit C6 - vstup z magnetofonu 2 (není zpracováván programem)
- tlačítko FAbit C5 - vstup kmitočtu 1kHz
- tlačítko CTRLbit C4 - zem
- tlačítko SH

bit C3 - výstup na reproduktor přes obvod MBA 810

BIT C2 - ovládání spínacího tranzistoru pro start a stop
magnetofonu 2, není připojeno ke konektorubit C1 - ovládání spínacího tranzistoru pro start a stop
magnetofonu 1, není připojeno ke konektoru (KC 507)

bit C0 - výstup na magnetofon M1 přes klopný obvod

87 řídicí registr obvodu 8255

88 programovatelný řadič přerušení 8259, adresa A=0

89 programovatelný řadič přerušení 8259, adresa A=1

registr žádostí o přerušení zapojen takto:

bit 7 - kmitočet 16 kHz

bit 6 - kmitočet 50 Hz

bit 5 - tlačítko BR (přes Schmidtův klopný obvod)

bit 4 - volný - konektor sběrnice, špička 43

bit 3 - volný - konektor sběrnice, špička 42

bit 2 - volný - konektor sběrnice, špička 41

bit 1 - volný - konektor sběrnice, špička 40

bit 0 - volný - konektor sběrnice, špička 39

žádosti o přerušení na špičky konektoru přes hradlo s otevřeným kolektorem spínané do úrovně L

F0 - FE rezervováno pro kreslič, tiskárnu, děrovač a snímač
ostatní adresy volné, doporučuje se užívat adresy 00 - 7F

Přehled adres paměti 32 kB používaných monitorem

- 0000-0002 volné 3 bajty, možno použít při instrukci RST 0
- 0003 I/O bajt pro přepínání periférií, standardně 69
- 0004-0005 adresa konce paměti RAM - 7FFF pro 32 kB
- 0006 bajt pro časování klávesnice a blikání kurzoru
- 0007 0 - kurzor bliká, jinak blikání potlačeno
- 0008-000A 3 bajty pro čítač 20 ms; čítání je zastaveno, je-li zamaskováno přerušování 50 Hz u obvodu 8259
- 000B kód znaku, na který ukazuje kurzor (kód pro přímý vstup na obrazovku)
- 000C-000D adresa kurzoru na obrazovce (EC00-EFFF nebo E800-EFFF)
- 000E-000F sloupec a řádek kurzoru (00-1F, pro sloupec s modulem Video 64 00-3F)
- 0010 1 - grafické znaky, 0 - normální režim
- 0011 1 - znaky v inverzi, 0 - normální režim
- 0012 počet posunovaných znaků v příkazech DC a IL
- 0013 délka stránky (01-1F) udávající číslo posledního řádku, který lze zobrazit na obrazovce
- 0014 řádkování po vložení příkazu CR
- 0015-0016 návratová adresa pro příkaz R monitoru
- 0017-0018 číslo udávající výšku a délku tónu v podprogramu F973
- 0019-001A adresa počátku zásobníku při nahrávání z magnetofonu a děrné pásky - EC00 pro monitor, 0100 pro basic
- 001B 7. bit ovládá polaritu signálu nahrávaného z magnetofonu, zbývající číslo je úměrné délce periody modulovaného kmitočtu (1 kHz)
- 001C číslo úměrné době v čekací smyčce podprogramu F5A2
- 001D-001E adresa teplého startu basicu (případně jiného jazyku v připojeném modulu)
- 001F maximální počet znaků na řádku - 20 pro VIDEO 32,
40 pro VIDEO 64
- 0020-0021 adresa začátku paměti obrazovky - EC00 pro VIDEO 32
E800 pro VIDEO 64
- 0020-002D oblast, do které se ukládají hodnoty registrů mikroprocesoru 8080 v podprogramu F60D
- volná oblast
- 7FC0-7FC1 adresa vrcholu zásobníku
- 7FC2-7FDF tabulka skoků na podprogramy obsluhující periférie
- 7FE0-7FFF tabulka skoků na podprogramy obsluhující přerušování
-
- EC00-EFFF paměť pro obrazovku v modulu VIDEO 32
- E800-EFFF paměť pro obrazovku v modulu VIDEO 64
- F000-F9C2 monitor v paměti EPROM, zůstává i po vypnutí počítače
- F9C3-FFFF 63D H = 1596 D volných buněk paměti EPROM pro uživatele

Přehled některých adres paměti 32 kB používaných BASICem 6

- 0050-009F oblast využívaná při vkládání příkazového a programového řádku basicu - maximálně 50 H znaků
- 00A4-00A5 adresa začátku oblasti USR deklarované 2. parametrem příkazu CLEAR
- 00A6-00A7 adresa konce oblasti CLEAR (USR+STRING)
- 00BA-00BB konec oblasti STRING deklarované 1. parametrem příkazu CLEAR
- 00CC-00CD adresa počátku zásobníku BASICu
- 00CE-00CF počátek programového bufferu basicu (016A bez kresliče)
- 00D0-00D1 adresa počátku oblasti, do které se ukládají hodnoty jednoduchých proměnných
- 00D2-00D3 adresa počátku oblasti, do které se ukládají identifikátory a hodnoty indexovaných proměnných
- 00D4-00D5 adresa počátku prázdné oblasti
- 0100-015A oblast využívaná při nahrávání programu z magnetofonu nebo z děrovače

C800-E7FF Basic 6 v paměti PROM

Spolupráce monitoru s basicem

BASIC-6 obsahuje několik příkazů a operátorů, kterými lze volat podprogramy ve strojovém kódu (podprogramy monitoru nebo uživatelské) a se kterými lze přímo pracovat s čísly v jednotlivých registrech. Veliké adresy a data v basicu vkládáme a získáváme v dekadickém tvaru, strojový kód však používá čísel v šestnáctkové soustavě.

Příklad 1:

Do registru na adrese 21AE H máme příkazem v basicu uložit číslo 41 H.

Řešení:

Příkaz POKE a,d vloží číslo d na adresu a. Číslo a,d můžeme vkládat dvojím způsobem:

1. Převědeme je do dekadického tvaru, tj.

$$\begin{aligned} a &= 21AE H = (2 \cdot 16^3 + 1 \cdot 16^2 + 10 \cdot 16^1 + 14 \cdot 16^0) D = \\ &= (2 \cdot 4096 + 1 \cdot 256 + 10 \cdot 16 + 14) D = 8622 D, \\ d &= 41 H = (4 \cdot 16 + 1) D = 65 D, \end{aligned}$$

tvar příkazu bude

POKE 8622,65

2. Pro převod použijeme operátoru HEX, který převede číslo ze šestnáctkové soustavy do dekadické:

POKE HEX(21AE),HEX(41)

Příklad 2:

Zjistěte stav čítače 50 Hz, který je tvořen čísly na adresách 0008-0009.

Řešení:

Číslo na adrese a zjistíme operátorem PEEK(a). Protože číslo v registru 0009 se zvýší o 1 právě tehdy, když se číslo v 0008 mění z hodnoty FF H (255 D) na 00 H, vynásobíme obsah registru 0008 číslem 256 D, tedy

```
PRINT PEEK(8) + 256*PEEK(9)
```

vytiskne skutečný obsah čítače jako dekadické číslo.

Poznámka:

Číslo d v předcházejících příkladech může být přirozené číslo menší než 255, číslo a pak menší než

$$10^{16}, \text{ tj. } 65536 D = (FFFF + 1) H.$$

Podprogramy ve strojovém kódu lze z basicu volat příkazy a operátory

```
CALL a,USR(a),WORD(a) a BYTE(a),
```

kde a je adresa počátku volaného podprogramu. Příkaz k návratu z podprogramu vrátí program ze strojového kódu do basicu.

Číslo získané operátory USR a BYTE nabývají hodnoty rovné číslu v registru A mikroprocesoru, operátorem WORD získáme číslo z registru A a mikroprocesoru, operátorem WORD získáme číslo z registru A a mikroprocesoru, operátorem WORD získáme číslo z registru A a mikroprocesoru, operátorem WORD získáme číslo z registru A a mikroprocesoru, operátorem WORD získáme číslo z registru A a mikroprocesoru.

Příklad 2:

Příkazem

```
K = USR(HEX(F8AA))
```

spouštíme podprogram F8AA H, ve kterém se čeká na stisk tlačítka, po jeho stisku je kód odpovídajícího znaku v registru A. Po návratu do basicu proto bude tento kód přiřazen identifikátoru K.

Příkaz je tedy ekvivalentní delšímu programu

```
... KŠ=INKEY$: IF KŠ="" THEN ...
    K=ASC(KŠ)
```

Kód K pak lze testovat např. na kódy řídicích znaků ap.

Příkazy a operátory CALL, WORD a BYTE mohou mít i více parametrů, které se předávají do volaného podprogramu ve strojovém kódu jako určité výchozí hodnoty registrů mikroprocesoru.

Příklad 4:

Příkazem S monitoru vložíme následující program:

```
1000 CD 0D F6 CALL F60D ; podprogram uloží okamžité hodnoty
                        ; registrů mikroprocesoru
1003 C3 CD F2 JMP F2C3 ; výpis registrů, návrat do monitoru
```

V basicu pak vyzkoušejme program

```
10 CALL 4096,257 .
```

Zjistíme tak, že v okamžiku vyvolání podprogramu jsou v párech BE a DE čísla 0101 H= 257 D. Stejný výsledek bychom zjistili i v příkazech

```
10 A=WORD(4096,257) nebo
10 A=BYTE(4096,257) .
```

Závěr:

Parametr w se v příkazech

```
CALL a,w
WORD(a,w)
BYTE(a,w)
```

ukládá před skokem do podprogramu na adrese a do registrových párů BC a DE.

Poznámka:

Protože po výpisu registrů přechází počítač do klidového stavu monitoru, nevrací se program zpět do basicu.

Příklad 5:

Se stejným podprogramem vyzkoušejme příkaz

```
10 CALL 4096,257,258 .
```

Zjistíme, že v příkazech s parametry

```
a, w1,w2
```

se ukládá w1 do páru BC a w2 do páru DE.

Poněkud složitější je situace, kdy vložíme více parametrů, např.

```
10 CALL 4096,257,258,259,260,261 .
```

Zjistíme, že do BC a DE se ukládají dva poslední parametry, zbývající jsou v zásobníku uspořádány takto:

0101 H	257 D	
0102 H	258 D	BC = 0104 H = 260 D
0103 H	259 D	DE = 0105 H = 261 D

E55D H návratová adresa do basicu po instrukci k návratu.

Parametry nutno při zpracování postupně ze zásobníku vybírat, např. instrukcemi

```
POP H
XTHL ; parametr v HL, návratová adresa opět v zásobníku .
```

Příklad 6:

27

Sestav program na převod dekadického čísla 0-65535 do hexadecimálního tvaru.

Řešení:

Hexadecimální výpis páru HL zajistí podprogram monitoru F5D0 v programu

```
0030 EB          XCHG          ; číslo z DE do HL
0031 C3 D6 F5    JMP F5D0      ; výpis HL a návrat do basicu .
```

Programové řádky v basicu budou

```
100 INPUT D
200 CALL 48,D .
```

Dekadické číslo D předáváme podprogramu ve strojovém kódu do páru DE, podprogram je vypíše.

Příklad 7:

Operátor WORD použij v programu z příkladu 2.

Řešení:

```
0030 2A 08 00    LHLD 0008     ; 2 bajty čítače do HL
0033 C9          RET           ; návrat do basicu

50 PRINT WORD(48)
```

V řádku 50 se vytiskne obsah páru HL, tedy obsah 2 bajtů čítače.

Příklad 8:

Zjednoduš příkaz

$L = \text{PEEK}(\text{HEX}(a)) + 256 \times \text{PEEK}(\text{HEX}(a+1))$,

kterým přiřazujeme identifikátoru L číslo z paměťových buněk a, a+1, pomocí operátoru WORD.

Řešení:

```
60 L=WORD(48,HEX(a))
```

ve strojovém kódu

```
0030 EB          XCHG          ; a do HL
0031 5E          MOV E,M       ; bajt z adresy a do E
0032 23          INX H         ; zvětšení adresy v HL o 1
0033 56          MOV D,M       ; bajt z adresy a+1 do D
0034 EB          XCHG          ; číslo z DE do HL, předáváno do basicu
0035 C9          RET           ; návrat do basicu
```

Poznámka:

Při návratu programu do basicu může při nevhodném vrcholu zásobníku dojít k havarii, někdy basic vyčkává na stisk klávesy.

Ze strojového kódu se vždy můžeme dostat do klidového stavu basicu, tj. do stavu, kdy počítač vyčkává na vkládání příkazů, instrukcí

```
C3 D6 CA    JMP CAD6 .
```

Parametry v operátorech WORD a BYTE mají stejný význam, jako parametry v příkazu CALL.

Poznámka k práci s modulem BASIC 6

Programovatelný řadič přerušení K 580 IK 59 odmaskuje někdy samovolně zamaskované žodosti o přerušení nebo špatně vyhodnotí požadavek na přerušení (např. při stisku tlačítka BR). Výsledkem této nežádoucí činnosti je skok na podprogram F800, kterým inicializujeme počítač po zapnutí nebo při stisku červeného tlačítka RES. Tím se zdánlivě vymaže již vložený, obvykle dlouhý, program v basicu. Zevadě předejdeme, když opravíme adresy skoků v tabulce přerušení na

C3 D6 CA

místo v tabulce uvedených

C3 00 F8.

V případě nežádoucího přerušení tak program začíná teplým startem basicu.

Pokud dojde k vymazání vloženého programu v basicu (třeba nechtěným stiskem RES), lze jej obnovit vložením a spuštěním krátkého programu ve strojovém kódu:

0100	21	6E	01	7E	23	A7	C2	03
0108	01	22	6A	01	5E	23	56	73
0110	32	CA	18	01	E3	C3	0C	01
0118	23	22	D0	00	C3	D6	CA	

Program spouštíme příkazem

G 100 CR.

Pokud tento program nahrajeme příkazem

W 100,11E,100

proběhne obnovení basicu ihned po nahrání programu z kazety do počítače.

Poznámka:

1. Při použití kresliče se posouvá počátek bufferu z 016A na 021D. Musíme proto upravit program takto:

```
0100 21 21 02 ..
0108 .. 22 1D 02
```

2. Při editaci programu v basicu (kopírování řádek, vymazávání vložených řádek) dochází k posunu adresy počátku oblasti jednoduchých proměnných na adrese 00D0-00D1. Výše uvedený program také upraví tuto adresu.

HEX	DEK	obrazovka	význam	ASSEMBLER
20	32	SP	SP	LXI H, a
21	33	!"#\$%&,'	!"#\$%&,'	SHLD a
22	34			INX H
23	35			INR H
24	36			DCR H
25	37			MVI H, d
26	38			DAA
27	39			
28	40	() * + , - . /	() * + , - . /	DAD H
29	41			LHLD a
2A	42			DCX H
2B	43			INR L
2C	44			DCR L
2D	45			MVI L, d
2E	46			CMA
2F	47			
30	48	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	LXI SP, a
31	49			STA a
32	50			INX SP
33	51			INR M
34	52			DCR M
35	53			MVI M, d
36	54			STC
37	55			
38	56	8 9 : ; < = > ?	8 9 : ; < = > ?	DAD SP
39	57			LDA a
3A	58			DCX SP
3B	59			INR A
3C	60			DCR A
3D	61			MVI A, d
3E	62			CMC
3F	63			

HEX	DEK	obrazovka	význam	ASSEMBLER
00	0	●	není	NOP
01	1	□	F1 x	LXI B, a
02	2	↓	F2 x	STAX B
03	3	↑	F3 x	INX B
04	4	→	F4	INR B
05	5	←	F5	DCR B
06	6	♀	není	MVI B, d
07	7	♂	připnutí	RLC
08	8	◆	kurs.vlevo	DAD B
09	9	✱	tabulátor	LDAX B
0A	10	┌	návrat vo.	DCX B
0B	11	└	IL	INR C
0C	12	├	HOME	DCR C
0D	13	┤	CR	MVI C, d
0E	14	┘	zr.grafiky	RRC
0F	15	┙	grafika	
10	16	┌	není	LXI D, a
11	17	└	není	STAX D
12	18	├	zr.inverze	INX D
13	19	┤	inverze	INR D
14	20	┘	není	DCR D
15	21	┙	není	MVI D, d
16	22	■	není	RAL
17	23	■	není	
18	24	■	kurs.vpravo	DAD D
19	25	■	kurs.nahoru	LDAX D
1A	26	■	kursor dolů	DCX D
1B	27	■	zr.AUTO x	INR E
1C	28	■	IC	DCR E
1D	29	■	DC	MVI E, d
1E	30	■	DL	RAR
1F	31	■	mez.obraz.	

x ... řídicí znak v basicu

HEX	DEK	obrazovka	význam	ASSEMBLER
40	64	@	@	MOV B, B
41	65	A	A	MOV B, C
42	66	B	B	MOV B, D
43	67	C	C	MOV B, E
44	68	D	D	MOV B, H
45	69	E	E	MOV B, L
46	70	F	F	MOV B, M
47	71	G	G	MOV B, A
48	72	H	H	MOV C, B
49	73	I	I	MOV C, C
4A	74	J	J	MOV C, D
4B	75	K	K	MOV C, E
4C	76	L	L	MOV C, H
4D	77	M	M	MOV C, L
4E	78	N	N	MOV C, M
4F	79	O	O	MOV C, A
50	80	P	P	MOV D, B
51	81	Q	Q	MOV D, C
52	82	R	R	MOV D, D
53	83	S	S	MOV D, E
54	84	T	T	MOV D, H
55	85	U	U	MOV D, L
56	86	V	V	MOV D, M
57	87	W	W	MOV D, A
58	88	X	X	MOV E, B
59	89	Y	Y	MOV E, C
5A	90	Z	Z	MOV E, D
5B	91	[[MOV E, E
5C	92	\	\	MOV E, H
5D	93]]	MOV E, L
5E	94	<	<	MOV E, M
5F	95	—	—	MOV E, A

HEX	DEK	obrazovka	význam	ASSEMBLER
60	96	\	\	MOV H, B
61	97	a	a	MOV H, C
62	98	b	b	MOV H, D
63	99	c	c	MOV H, E
64	100	d	d	MOV H, H
65	101	e	e	MOV H, L
66	102	f	f	MOV H, M
67	103	g	g	MOV H, A
68	104	h	h	MOV L, B
69	105	i	i	MOV L, C
6A	106	j	j	MOV L, D
6B	107	k	k	MOV L, E
6C	108	l	l	MOV L, H
6D	109	m	m	MOV L, L
6E	110	n	n	MOV L, M
6F	111	o	o	MOV L, A
70	112	p	p	MOV M, B
71	113	q	q	MOV M, C
72	114	r	r	MOV M, D
73	115	s	s	MOV M, E
74	116	t	t	MOV M, H
75	117	u	u	MOV M, L
76	118	v	v	HLT
77	119	w	w	MOV M, A
78	120	x	x	MOV A, B
79	121	y	y	MOV A, C
7A	122	z	z	MOV A, D
7B	123	{	{	MOV A, E
7C	124			MOV A, H
7D	125	}	}	MOV A, L
7E	126	~	~	MOV A, M
7F	127	■	■	MOV A, A

HEX	DEK	obrazovka	význam	ASSEMBLER
80	128	◼	END	ADD B
81	129	inv ◻	FOR	ADD C
82	130	inv ↓	NEXT	ADD D
83	131	inv ↑	DATA	ADD E
84	132	inv →	INPUT	ADD H
85	133	inv ←	DIM	ADD L
86	134	inv ♀	READ	ADD M
87	135	inv ♂	LET	ADD A
88	136	inv ◆	GOTO	ADC B
89	137	inv ⬆	RUN	ADC C
8A	138	inv ⬇	IF	ADC D
8B	139	inv L	RESTORE	ADC E
8C	140	inv ⊥	GOSUB	ADC F
8D	141	inv ▬	RETURN	ADC L
8E	142	inv +	REM	ADC M
8F	143	inv +	STOP	ADC A
90	144	inv F	OUT	SUB B
91	145	inv -	ON	SUB C
92	146	inv T	GET	SUB D
93	147	inv F	POKE	SUB E
94	148	inv I	PRINT	SUB H
95	149	inv ◼	LPRINT	SUB L
96	150	inv ◼	DEF	SUB M
97	151	inv ◼	CONT	SUB A
98	152	inv ◻	LIST	SBB B
99	153	inv ◻	LLIST	SBB C
9A	154	inv ◻	CLEAR	SBB D
9B	155	inv ◻	PTAPE	SBB E
9C	156	inv ◻	MLOAD	SBB H
9D	157	inv ◻	PLIST	SBB L
9E	158	inv ◻	MSAVE	SBB M
9F	159	inv ◻	SCRATCH	SBB A

HEX	DEK	obrazovka	význam	ASSEMBLER
A0	160	inv ◼	CALL	ANA B
A1	161	inv !	BYE	ANA C
A2	162	inv "	WAIT (ANA D
A3	163	inv =	AUTO	ANA E
A4	164	inv S	MEM	ANA H
A5	165	inv %	MOV	ANA L
A6	166	inv &	VECT	ANA M
A7	167	inv /	PO INT	ANA A
A8	168	inv (ORG	XRA B
A9	169	inv)	SPEED	XRA C
AA	170	inv *	NARROW	XRA D
AB	171	inv +	WIDE	XRA E
AC	172	inv ,	WRITE	XRA H
AD	173	inv -	SIZE	XRA L
AE	174	inv .	FREE	XRA M
AF	175	inv /	UNFLOT	XRA A
B0	176	inv 0	PLOT	ORA B
B1	177	inv 1	CLS	ORA C
B2	178	inv 2	TAB (ORA D
B3	179	inv 3	TO	ORA E
B4	180	inv 4	SPC (ORA H
B5	181	inv 5	FN	ORA L
B6	182	inv 6	THEN	ORA M
B7	183	inv 7	NOT	ORA A
B8	184	inv 8	STEP	CMP B
B9	185	inv 9	+ (op.)	CMP C
BA	186	inv :	- (op.)	CMP D
B3	187	inv ;	* (op.)	CMP E
BC	188	inv <	/ (op.)	CMP H
BD	189	inv =	^ (op.)	CMP L
BE	190	inv >	AND	CMP M
BF	191	inv ?	OR	CMP A

HEX	DEK	obrazovka	v ýznam	ASSEMBLER
C0	192	inv	< (op.)	RNZ
C1	193	inv	= (op.)	POP 3
C2	194	inv	> (op.)	JNZ a
C3	195	inv	SGM	JMP a
C4	196	inv	INT	CNZ a
C5	197	inv	ABS	PUSH 3
C6	198	inv	USR	ADI d
C7	199	inv	INP	RST 0
C8	200	inv	FOS	RZ
C9	201	inv	SQR	.RET
CA	202	inv	RND	JZ a
CB	203	inv	LOG	...
CC	204	inv	EXP	CZ a
CD	205	inv	COS	CALL a
CE	206	inv	SIN	ACI d
CF	207	inv	TAN	RST 1
D0	208	inv	ATN	RNC
D1	209	inv	PEEK	POP D
D2	210	inv	LEN	JNC a
D3	211	inv	STRS	OUT d
D4	212	inv	VAL	CNC a
D5	213	inv	ASC	PUSH D
D6	214	inv	CHRS	SUI d
D7	215	inv	LEFTS	RST 2
D8	216	inv	RIGHTS	RC
D9	217	inv	MIDS	...
DA	218	inv	INKEYS	JC a
DB	219	inv	HEX (IN d
DC	220	inv	PI	CC a
DD	221	inv	BYTE (...
DE	222	inv	WORD (SBI d
DF	223	inv	PTR	RST 3

HEX	DEK	obrazovka	v ýznam	ASSEMBLER
E0	224	inv	`	RPO
E1	225	inv	a	POP H
E2	226	inv	b	JPO a
E3	227	inv	c	XTHL
E4	228	inv	d	CPO a
E5	229	inv	e	PUSH H
E6	230	inv	f	ANI d
E7	231	inv	g	RST 4
E8	232	inv	h	RPE
E9	233	inv	i	PCHL
EA	234	inv	j	JPE a
EB	235	inv	k	XCHG
EC	236	inv	l	CPE a
ED	237	inv	m	...
EE	238	inv	n	XRI d
EF	239	inv	o	RST 5
F0	240	inv	p	RP
F1	241	inv	q	POP PSW
F2	242	inv	r	JP a
F3	243	inv	s	DI
F4	244	inv	t	CP a
F5	245	inv	u	PUSH PSW
F6	246	inv	v	ORI d
F7	247	inv	w	RST 6
F8	248	inv	x	RM
F9	249	inv	y	SPHL
FA	250	inv	z	JM a
FB	251	inv	{	EI
FC	252	inv		CM a
FD	253	inv	}	...
FE	254	inv	~	CPI d
FF	255	inv	■	RST T

Instrukční kód 8080 abecedně dle mnemokódu

d ... konstanta 1byte, w ... konstanta 2byte
a ... adresa 2byte

* nastavuje všechny bity (CY, ZR, SG, PE, AC)
+ nastavuje pouze CY
! nastavuje všechny bity kromě CY (ZR, SG, PG, AC)

ACI	ADC	ADC	ADC	ADC	ADC	ADC	ADC	ADD	ADD	ADD	ADD	ADD	ADD	ADI	ANA	ANA	ANA	ANA	ANA	ANA	ANA	ANA	ANI	CALL	CC	CM	CMA	CMC	CMP	CMP	CMP	CMP	CMP	CMP	CMP	CNC						
d	A	B	C	D	E	H	L	M	A	B	C	D	E	H	L	M	A	B	C	D	E	H	L	M	A	a	a	A	B	C	D	E	H	L	M	A	B					
8F	88	89	8A	8B	8C	8D	8E	87	80	81	82	83	84	85	86	C6	A7	A0	A1	A2	A3	A4	A5	A6	E6	CD	DC	FC	2F	3F	BF	38	39	3A	3B	3C	3D	3E	D4			
"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"		
C4	R4	EC	FE	E4	CC	27	09	19	29	39	3D	05	0D	15	1D	25	2D	35	03	13	23	33	F3	F3	76	D3	3C	04	0C	14	1C	24	2C	34	03	13	23	33	DA	FA		
a	a	a	d	a	a	B	B	D	H	SP	A	B	C	D	E	H	L	M	B	B	B	B	B	B	B	B	A	B	C	D	E	H	L	M	B	B	B	B	B	B	a	a
JMP	JNC	JNZ	JP	JPE	JPO	JZ	LDA	LDAX	LDAX	LHLD	LXI	LXI	LXI	LXI	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	
C3	D2	C2	F2	EA	E2	CA	3A	0A	1A	2A	01	11	21	31	7F	78	79	7A	7B	7C	7D	7E	47	40	41	42	43	44	45	46	4F	48	49	4A	4B	4C	4D	4E	57	50		
D,D	D,D	D,E	D,H	D,L	D,M	E,A	E,B	E,C	E,D	E,E	E,H	E,L	E,M	E,A	H,B	H,C	H,D	H,E	H,H	H,L	H,M	H,A	B	L	L	L	L	L	L	L	M	M	M	M	M	M	M	M	M	M	M	M
MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV
E,d	H,d	L,d	M,d	NOP	ORA	ORA	ORA	CRA	ORA	ORA	ORA	ORA	ORA	ORI	OUT	FCHL	POP	POP	POP	POP	PUSH	PUSH	PUSH	RAL	RAR	RC	RET	RLC	RM	RNC	RNZ	RP	RPE	RPO	RRC	RST	RST	RST	RST	RST		
1E	26	2E	36	00	B7	B0	B1	B2	B3	B4	B5	B6	F6	D3	E9	C1	D1	E1	F1	C5	D5	E5	F5	17	1F	D8	C9	07	F8	D0	C0	F0	E8	E0	0F	C7	CF	D7	DF	E7		
MVI	MVI	MVI	MVI	NOP	ORA	ORA	ORA	CRA	ORA	ORA	ORA	ORA	ORI	OUT	FCHL	POP	POP	POP	POP	PUSH	PUSH	PUSH	RAL	RAR	RC	RET	RLC	RM	RNC	RNZ	RP	RPE	RPO	RRC	RST	RST	RST	RST	RST	RST		
RST	RST	RST	RZ	SBB	SBB	SBB	SBB	SBB	SBB	SBB	SBB	SBB	SBI	SHLD	SPHL	STA	STAX	STAX	STC	SUB	SUB	SUB	SUB	SUB	SUB	SUB	SUI	XCHG	XRA	XRA	XRA	XRA	XRA	XRA	XRA	XRI	XTHL					
5	6	7	A	B	C	D	E	H	L	M	d	a	a	a	a	B	D	D	A	B	C	D	E	H	L	M	d	A	B	C	D	E	H	L	M	d	A	B	E	E		
FF	C8	9F	98	99	9A	9B	9C	9D	9E	DE	22	P9	32	02	12	37	97	90	91	92	93	94	95	96	D6	EB	AF	A8	A9	AA	AB	AC	AD	AE	EE	E3						

Jednoduchý zpětný assembler - verze 2

Zpětný assembler umožňuje výpis programu ve strojovém kódu pomocí mnemotechnických zkratk. Začátek zpětného assembleru

1. je automaticky spuštěn po nahrání programu z magnetofonu nebo
2. voláme příkazem C 7500 CR.

Na začátku programu se zobrazí

"od adresy"

a program čeká na vložení adresy, od které chceme program vypsat.

Program se vypíše tak, aby byla zaplněná obrazovka. Ve spodním řádku se zobrazují znaky, odpovídající kódu pro přímý vstup na obrazovku (ne tedy instrukce mikroprocesoru).

Dále můžeme stisknout tlačítko

- a) SP ... výpis pokračuje dále,
- b) ↑ ... ve výpisu listujeme zpět až po vloženou adresu,
- c) V ... znovu se zobrazí " od adresy ",
můžeme vložit adresu novou,
- d) BR ... vracíme se zpět do klidového stavu monitoru.

Zpětný assembler lze vložit ručně příkazem S podle následující tabulky. Vložený program překontrolujeme a nahrajeme příkazem

C 78D6 CR.

Nahrání programu z kazety uložíme do počítače příkazem L CR.

Výpis zpětného assembleru - verze 2:

```

7500 C3 A3 77 CD D0 F5 C5 CD 47 F6 20 C1 C9 CD D5 F5
7510 C3 06 75 E5 2A 0E 00 6F 22 0E 00 CD 34 F5 E1 C9
7520 4F A7 CA 88 F4 7E 23 A7 F2 25 75 0D C3 22 75 7A
7530 CD 20 75 C3 06 75 3A 50 00 47 E6 07 5F 78 0F 0F
7540 0F E6 07 57 78 FE 40 D2 8D 75 21 64 75 E6 04 CA
7550 DD 75 73 FE 07 CA 60 75 C6 04 CD 30 75 21 85 75
7560 7A C3 30 75 52 4C C3 52 52 C3 52 41 CC 52 41 D2
7570 44 41 C1 43 4D C1 53 54 C3 43 4D C3 49 4E D2-44
7580 43 D2 4D 56 C9 C2 C3 C4 C5 C8 CC CD C1 21 B3 75
7590 FE 80 D2 D1 75 3E 06 BA C2 9F 75 B3 CA 88 F4 3E
75A0 01 CD 30 75 21 85 75 7A CD 20 75 CD 47 F6 2C 53
75B0 C3 5D 75 48 4C D4 4D 4F D6 41 44 C4 41 44 C3 53
75C0 55 C2 53 42 C2 41 4E C1 58 52 C1 4F 52 C1 43 4D
75D0 D0 FE C0 D2 40 76 7A C6 02 53 C3 5A 75 21 05 76
75E0 3B C2 EC 75 3A CA 88 F4 3C C3 20 75 21 09 76 3E
75F0 02 3B 7A CA 30 75 A7 1F 57 3E 07 83 CD 30 75 21

```

7600	3B	76	C3	2F	75	4E	4F	D0	BF	53	54	41	58	20	C2	4C
7610	44	41	58	20	C2	53	54	41	58	20	C4	4C	44	41	58	20
7620	C4	53	48	4C	C4	4C	48	4C	C4	53	54	C1	4C	44	C1	4C
7630	58	C9	44	41	C4	49	4E	D8	44	43	D8	C2	C4	C8	53	D0
7640	21	70	76	7B	FE	07	CA	D7	76	C6	06	FE	09	CA	F3	76
7650	C6	05	FE	11	CA	F3	76	21	A8	76	FE	0C	CA	E2	76	FE
7660	10	CA	DF	76	C6	09	0F	CD	20	75	21	C5	76	C3	2F	75
7670	52	53	D4	B0	B1	B2	B3	B4	B5	B6	B7	4A	4D	D0	BF	4F
7680	55	D4	49	CE	58	54	48	CC	58	43	48	C7	44	C9	45	C9
7690	41	44	C9	41	43	C9	53	55	C9	53	42	C9	41	4E	C9	58
76A0	52	C9	4F	52	C9	43	50	C9	52	45	D4	BF	50	43	48	CC
76B0	53	50	48	CC	50	4F	D0	43	41	4C	CC	BF	BF	BF	50	55
76C0	53	C8	D2	CA	C3	4E	DA	DA	4E	C3	C3	50	CF	50	C5	D0
76D0	CD	C2	C4	C8	50	53	D7	AF	CD	30	75	23	C3	2F	75	21
76E0	B7	76	7A	1F	57	DA	30	75	3E	04	CD	30	75	21	D1	76
76F0	C3	2F	75	82	C3	30	75	47	0E	02	FE	C3	C8	FE	CD	C8
7700	E6	EF	FE	22	C8	FE	2A	C8	E6	CF	FE	01	C8	E6	C7	FE
7710	C2	C8	FE	C4	C8	0D	78	E6	F7	FE	D3	C8	E6	C7	FE	06
7720	C8	FE	C6	C8	0D	C9	C5	CD	F7	76	79	C1	32	52	00	C9
7730	7A	94	C0	73	95	C9	C5	D5	E5	CD	36	75	E1	D1	C1	C9
7740	C5	D5	E5	CD	49	77	C3	3C	77	3E	03	CD	13	75	2A	54
7750	00	CD	03	75	3A	50	00	CD	0D	75	3A	52	00	47	A7	CA
7760	73	77	FE	01	2A	56	00	7D	CA	70	77	7D	CD	0D	75	7C
7770	CD	0D	75	3E	12	CD	13	75	C5	CD	36	77	C1	78	A7	C8
7780	2A	56	00	FE	01	CA	8C	77	CD	03	75	C9	7D	CD	0D	75
7790	C9	32	14	00	32	58	00	47	3E	0D	05	C2	9F	77	07	32
77A0	5A	00	C9	21	0B	79	22	5C	00	3E	02	CD	91	77	21	33
77B0	78	CD	88	F4	CD	C3	F4	EB	2A	5C	00	73	23	72	23	22
77C0	5C	00	EB	22	5E	00	CD	47	F6	1F	3A	5A	00	32	60	00
77D0	22	54	00	7E	32	50	00	23	CD	26	77	A7	CA	ED	77	5E
77E0	23	FE	01	CA	E8	77	56	23	EB	22	56	00	EB	CD	40	77
77F0	CD	30	F5	3A	60	00	3D	C2	CD	77	23	EB	01	A0	EF	2A
7899	5E	00	CD	47	F2	EB	23	CD	AA	F8	FE	56	CA	A3	77	FE
7810	03	DC	91	77	FE	19	C2	B7	77	CD	25	78	C2	BF	77	2A
7820	62	00	C3	07	78	22	62	00	2A	5C	00	2B	2B	11	FC	78
7830	CD	30	77	C8	2B	56	2B	5E	23	23	C9	1F	56	79	70	69
7840	73	20	6F	64	20	61	64	72	65	73	79	A0	CD	47	F6	1F
7850	21	03	0A	22	0E	00	3E	01	32	14	00	21	85	78	CD	88
7860	F4	CD	7A	78	21	88	78	CD	88	F4	CD	7A	78	21	32	78
7870	CD	88	F4	21	00	00	E5	C3	BA	F3	06	16	CD	47	F6	51
7880	05	C2	7C	78	C9	0F	4D	D0	5F	0E	0D	20	20	20	20	0F
7890	54	0E	44	49	53	41	53	53	45	4D	42	4C	45	52	20	2D
78A0	20	56	45	52	5A	45	20	32	0F	54	0E	0D	20	20	20	20
78B0	0F	C3	4A	0E	0D	0D	20	20	2A	20	43	20	4C	75	62	
78C0	6F	6D	69	72	20	4A	55	7A	65	6B	20	37	2F	38	34	20
78D0	56	64	66	20	2A	8D	21	56	02	22	1B	00	21	4C	78	11
78E0	F3	78	01	4C	78	CD	67	F2	3E	21	CD	A5	F5	21	00	75
78F0	11	43	78	01	A3	77	CD	67	78	F2	C3	D4	F1			