

OSOBNÍ MIKRO POČÍTAČ PMD-85

UŽIVATELSKÁ PŘÍRUČKA

IV. OUTPUT/ENTER



TESLA - ELEKTRONICKÉ SOUČÁSTKY

Obsah :

Úvod	7
I. Všeobecná část	8
1. Příkaz OUTPUT	11
1.1. Výstupní rutina znaku	13
2. Příkaz ENTER	14
2.1. Vstupní rutina znaku	16
II. Kanál 4 - paralelní	17
1. Hardware	18
2. Popis režimů kanálu 4	20
2.a) Režim 0	20
2.b) Režim 1	21
2.c) Režim 2	25
2.d) kombinované režimy	27
3. Software pro OUTPUT	27
4. Software pro ENTER	31
5. Programování kanálu 4	36
6. Status kanálu 4	39
III. Kanál 1 - sériový	42
1. Hardware	42
2. Popis obvodu 8251	46
3. Software vstup/výstup pro MG	48
4. Software pro vstup/výstup pro V24	52
5. Programování kanálu 1	56
5.1. Programování asynchronního provozu	58
5.2. Programování synchronního provozu	59
5.3. Nastavení povelové instrukce	60
6. Status kanálu 1	61
IV. Kanál 5 - časovač	71

1. Hardware	71
1.1. Režim 0	73
1.2. Režim 1	74
1.3. Režim 2	75
1.4. Režim 3	76
1.5. Režim 4	77
1.6. Režim 5	78
2. Software kanálu 5	78
3. Programování kanálu 5	79
V. Kanál 7 - IMS - 2	83
1. Základní informace o systému IMS-2	86
1.1. Struktura sběrnice IMS-2	89
2. Odchyly IMS-2 v počítači PMD-85	98
3. Software pro OUTPUT 7	99
4. Software pro ENTER 7	104
4.1. Modifikace ENTER 7	106
5. Jiné využití kanálu 7	109
6. Povelů HP-IB	110
7. Příklady obsluhy přístrojů IMS-2 (HP-IB) .	113
VI. Aplikační konektor	119
VII. Závěr	129
VIII. Přílohy	131

Seznam příkladů :

1. Nastavení řídícího registru kanálu 8. 4	3
2. Výstupní rutina znaku	14
3. Vstupní rutina znaku	17
4. Vstupní rutina pro jeden znak	17
5. Vytvoření výstupní rutiny pro kanál 406	30
6. Úprava vstupní rutiny znaku	34
7. Vytvoření vstupní rutiny pro kanál 406	35
8. Nastavení linek PC v kanálu 4	39
8a. Nastavení kanálu 4 pro daný režim	39
9. Programová obsluha tiskárny CONSUL 2111	41
10. Úprava příkazu ENTER 1	54
11. Nastavení kanálu 1 do asynchronního provozu	59
12. Nastavení kanálu 1 do synchronního provozu	60
13. Nastavení kanálu 1	61
14. - " -	62
15. Systémový návrh připojení T 100	64
16. Programování časovače 8253	81
17. - " -	82
18. - " -	82
19. Modifikace příkazu OUTPUT 7	102
20. Modifikace koncového znaku ENTER 7	106
20a. - " -	106
21. Vytvoření příkazu PPOLL	111
22. Obsluha voltmetru M1T330	114
23. Obsluha přístroje BM 559	114
24. Obsluha LCR - HP 4275A	116
25. Obsluha BM 526	118
26. Použití aplikačního konektoru	120
27. Připojení minifloppy disku	120

Úvod

Uživatelská příručka "OUTPUT/ENTER" je určena uživatelům osobního počítače PMD-85, kteří jej chtějí využít ve výpočetním případně měřicím systému ve spolupráci s periferními zařízeními. Mikropočítač PMD 85 poskytuje k tomuto účelu různé stykové kanály. V šesti kapitolách je podrobně probrána filosofie OUTPUT/ENTER pro jednotlivé kanály. Uživatel je v nich seznámen s možnostmi jejich modifikací. Ke studiu dané problematiky se předpokládají u uživatele znalosti programování v systému 8080 a základní vědomosti o stavebnicových prvcích rodiny 8080.

V příloze příručky jsou zahrnuty v jednoduché formě bez komentáře důležité tabulky, se kterými by uživatel přicházel často do styku a připojovací protokoly některých běžných periferních zařízení k mikropočítači PMD 85.

Pro ty uživatele, kteří si chtějí vytvářet vlastní příkazy BASIC pro ovládání periferních zařízení doporučujeme uživatelskou příručku část V - EXTENDED ROM BASIC, která obsahuje popis konstrukce programových řádků BASIC.

I. Všeobecná část

Základním programovým vybavením osobního mikropočítače PMD 85 je graficky orientovaný programovací jazyk BASIC. Jako komunikační vstupně-výstupní zařízení je použit běžný televizní přijímač a vestavěná ASCII klávesnice. Styk s okolím zabezpečuje interfaceová deska, na které jsou umístěny stykové kanály pro paralelní a sériový přenos, časovač s reálnými hodinami, standardní styk pro měřicí přístroje - sběrnice IMS - 2 (HPIB) a styk s kazetovým magnetofonem. Všechny tyto kanály mají podporu v jazyce BASIC-G, což umožňuje lehce pracovat s příslušným vstupně-výstupním (I/O) kanálem. Všeobecným příkazem pro výstup dat je příkaz

OUTPUT

a pro vstup je příkaz

ENTER

Tyto příkazy mohou podporovat příkazy nižšího stupně hierarchie, a to **CONTROL** a **STATUS** nebo úplně nejnižší **OUT** a **INP**.

Adresace I/O kanálů je určena konfigurační adresních bitů A_0 až A_7 . Na následujícím obrázku je znázorněn adresní výběr příslušných kanálů.

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
1	X	X	X	1	1	X	X

Adresace externích kanálů prostřednictvím aplikačního konektoru.

A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
Ø	X	X	X	1	1	X	X
Ø	Ø	Ø	Ø	1	1	X	X
Ø	Ø	Ø	1	1	1	X	X
Ø	Ø	1	Ø	1	1	X	X
Ø	Ø	1	1	1	1	X	X
Ø	1	Ø	Ø	1	1	X	X
Ø	1	Ø	1	1	1	X	X
Ø	1	1	Ø	1	1	X	X
Ø	1	1	1	1	1	X	X
1	X	X	X	Ø	1	X	X
1	X	X	X	1	Ø	X	X

Adresace interních kanálů,
příčemž :

kanál Ø	neobsazen
kanál 1	sériový kanál V24/MG
kanál 2	neobsazen
kanál 3	neobsazen
kanál 4	GPIO (paralelní)
kanál 5	TIMER
kanál 6	neobsazen
kanál 7	IMS - 2

Adresace ROM modulu

Adresace klávesnice

Poznámka : znak X představuje hodnotu, na které nezávisí adresace daného kanálu. Tyto adresní bity (A₀, A₁) mají význam až při určování daného registru uvnitř daného kanálu, např.

Ø	1	Ø	Ø	1	1	1	1
---	---	---	---	---	---	---	---

adresuje paralelní kanál č. 4,

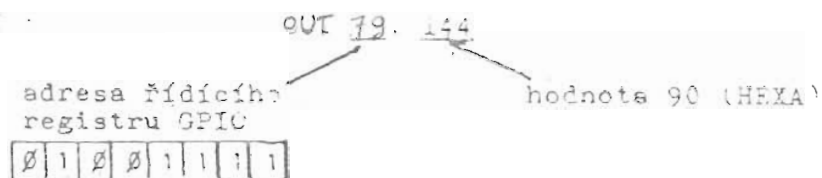
GPIO a zpřístupňuje řídicí registr tohoto kanálu.

PŘÍKLAD č. 1 : Nastavte řídicí registr kanálu č. 4 na hodnotu 90 (HEXA) t.j. 144 (decimálně).

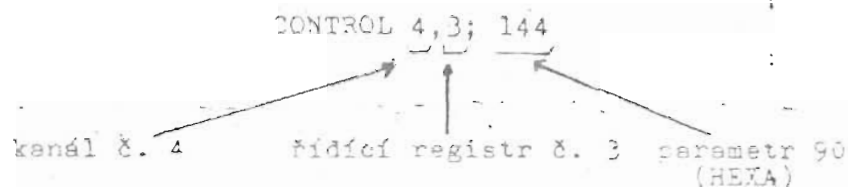
ŘEŠENÍ : Existují tři možné způsoby na různých úrovních; strojový kód, standardním příkazem BASIC a příkazem pro I/O kanály.

1. Strojový kód :	MVI A, 90	3E 90
	OUT CWR	D3 4F

2. Standard BASIC



3. Příkaz I/O



Poznámka : Jak je z příkladu zřejmé, podle způsobu 1. a 2. je nutné znát konkrétní hodnotu výběrové adresy příslušného kanálu. Způsob č. 3 však umožňuje pracovat s efektivním přístupem k daným kanálům.

Dříve než budeme probírat jednotlivé kanály, je nutné si objasnit filosofii příkazů I/O. Jsou to v podstatě tyto příkazy:

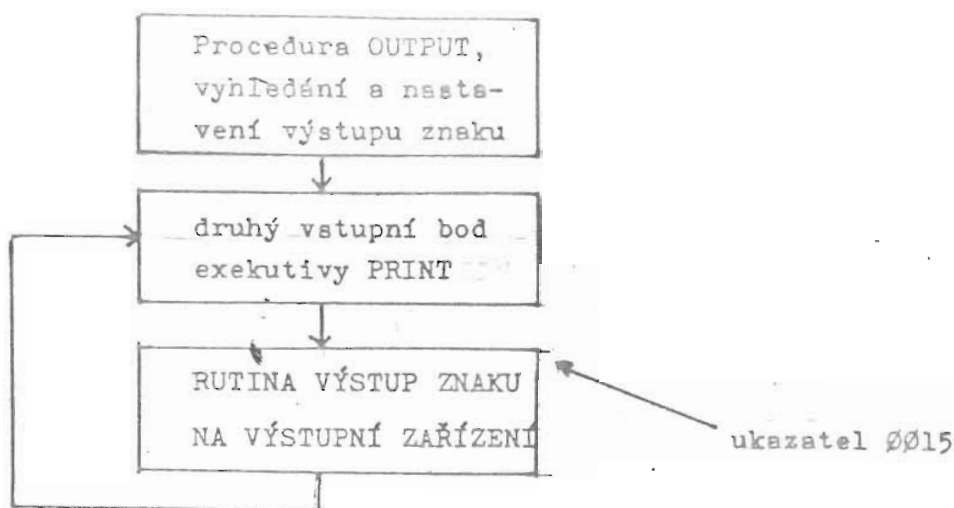
- **CONTROL** stanovení příslušného registru v daném kanálu
- **STATUS** funkce umožňující přečtení hodnoty příslušného registru v daném kanálu
- **OUTPUT** výstup výrazu na definovaný kanál
- **ENTER** vstup údajů do proměnné z definovaného kanálu

Příkazy CONTROL, STATUS se obvykle používají pro nastavení daného kanálu do požadovaného stavu pro danou periferní jednotku. Jsou jednoduchého provedení a je nutné je přirovnat k příkazům OUT a INP.

Příkazy OUTPUT a ENTER mají zvláštní význam, protože zabezpečují transfer bloku dat.

1.1. Příkaz OUTPUT

Umožňuje obdobně jako příkaz PRINT výstup výrazu na vybrané výstupní zařízení připojené na příslušný I/O kanál. Využívá se stejná rutina jako u příkazu PRINT. Následující obrázek znázorňuje jejich vzájemný vztah.



Úlohou procedury OUTPUT je "vsadit" do exekutivy PRINT příslušný vektor výstupní rutiny znaku. Adresní místo tohoto ukazatele je 0015 (HEXA). Tato hodnota se vypočítává v proceduře OUTPUT a závisí na jeho přidavných parametrech.

SYNTAX

OUTPUT Kr; výraz

kde - K znamená příslušný kanál I/O

r reprezentuje kód tzv. výstupního registru, přes který se bude uskutečňovat výstup znaků. Existuje max. 8 možných způsobů, které závisí na daném kanálu I/O.

Při interpretování delimitru ";" v příkazu OUTPUT je už nastaven ukazatel výstupní rutiny znaku. Vývojový diagram na následujícím obrázku znázorňuje průchodnost programu až po

daný delimetr středník.



Adresy vektorů výstupních exekutiv : v tabulce OUTPUT

Ukazatel (HEXA)	Kanál	Vektor (HEXA)
2088	0	FFFF
208A	1	2215
208C	2	FFFF
208E	3	FFFF
209D	4	209F
2092	5	FFFF
2094	6	FFFF
2096	7	2008

Iy kanály, které nejsou obsazeny příkazem OUTPUT se inicializují vektorem FFFF, což způsobí návrat do BASIC interpretu vstupem "COLD ENIAR" (adresa 2100) za předpokladu, že na tomto adresním vektoru je instrukce NOP (hodnota 00).

Poznámka : Kanál 3, 5 - TIMER není obsazen v příkaze OUTPUT, takže jej uživatel může modifikovat podle vlastní potřeby.

1.1.1. Výstupní rutina znaku

Úlohou výstupní rutiny znaku je odeslat na příslušný I/O kanál obsah akumulátoru, ve kterém je uložen znak ASCII. Korespondence odeslaného znaku na periferní zařízení závisí na zvoleném programovém režimu I/O kanálu. Nejjednodušší případ je odeslání znaku bez jakékoli korespondence :

OUT PORT
RET

Při vytváření této rutiny uživatelem je nutné si uvědomit, že výstup z ní je možný pouze instrukcí RET nebo jejími podmínkovými instrukcemi a nesmí být během ní zničeny jiné registry než akumulátor. Interpret BASIC zpracovává obsah akumulátoru do tvaru ASCII znaku, tedy maskuje jeho nejvyšší bit. Pokud uživatel požaduje poslat na I/O kanál úplný obsah akumulátoru, t.j. osm bitů, je nutné modifikovat interpret.

Na adrese :

03EE	ANI 7F	E6 7F
------	--------	-------

zrušení "ANI" POKE 1006, 0
nastavení "ANI" POKE 1007, 127

Následující příklad ukazuje vytvoření výstupní rutiny znaku pro shromáždění všech výstupních znaků ve výkonném BUFFERu, odkud budou ve vhodném okamžiku přebrány pomalým výstupním zařízením. Předpokládejme, že ukazatel zapisovaného místa je inicializován na začátku.

Příklad č. 2

Výstupní rutina znaku pro vyrovnávací BUFFER výstupních znaků.

OUTZNK:	PUSH H	; uchování reg. HL
	LHLD UKBUF	nastavení ukazatele
	MOV M,A	přesun dat do BUFFERu
	INX H	zvýšení ukazatele
	SHLD UKBUF	uchování nového ukazatele
	POP H	restaurování HL
	RET	návrat do BASICu

Ze struktury této výstupní rutiny znaku je zřejmé, že pokud bychom požadovali změnu typu výstupního znaku na zařízení, které nepracuje s ASCII znaky např. dálnopis, vsuneme do této rutiny podprogram, zabezpečující konvergenci znaku.

Poznámka : Vektory výstupních rutin znaku budou uvedeny u jednotlivých I/O kanálů v dalších kapitolách.

I.2. Příkaz ENTER

- umožňuje modifikovat obsah proměnné podobně jako příkaz INPUT.

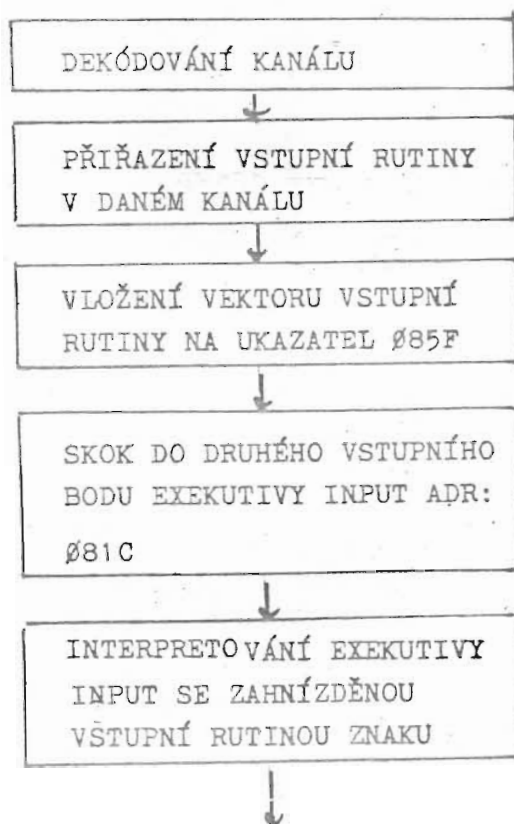
po zvolení příslušného I/O kanálu a vstupního registru, který určuje charakter vstupní rutiny znaku se vstupuje do interpretování příkazu INPUT. Pro příkaz ENTER tedy platí tatáž omezení, jako pro INPUT (ukončovací znak 00, oddělovač čárka).

Syntax : ENTER Kr; seznam proměnných

- kde "K" a "r" mají tentýž význam jako u příkazu OUTPUT

Zásadní rozdíl mezi architekturou příkazů OUTPUT, ENTER je, že u příkazu OUTPUT dochází k okamžitému (pokud uživatel nezmění charakter výstupní rutiny) výstupu znaku na I/O kanál, kdežto u příkazu ENTER se po přípravě příslušného kanálu všechna vstupní data shromáždí ve vstupním BUFFERU a až potom jsou přiřazovány dané proměnné. Ukončovacím znakem (END MARK) je byte 00. Vstupní BUFFER je umístěn od adresy 7F 01 (HEXA).

Schématicky znázorněná činnost příkazu ENTER je vidět na následujícím obrázku :



PŘÍŘAZENÍ OBSAHU VSTUPNÍHO BUFFERU PROMĚNNÉ

Adresy vektorů příslušných kanálů :

Ukazatel (HEXA)	Kanál	Vektor (HEXA)
2151	0	FFFF
2153	1	222C
2155	2	FFFF
2157	3	FFFF
2159	4	21B2
215B	5	FFFF
215D	6	FFFF
215F	7	2127

Údaje a poznámky pro tabulku vstupních rutin jsou stejné jako u příkazu OUTPUT.

I.2.1. Vstupní rutina znaku

Úlohou vstupní rutiny znaku je přijatý znak z I/O kanálu umístit do vstupního BUFFERu. V případě, že jde o jednoznakovou vstupní rutinu, činnost je ukončena zapsáním uvedené END MARKY (byte 00). Korespondenční vztahy mezi kanálem a periferním zařízením platí stejně jako u příkazu OUTPUT. Vstupní rutina znaku může pracovat se všemi registry CPU (8080). Vstupním argumentem rutiny je nastavený ukazatel BUFFERu. Tento ukazatel se musí před opuštěním vstupní rutiny znaku nastavit o jednu adresu nižší, tedy na hodnotu 7F 00 (HEXA). Tato skutečnost umožňuje uživateli vytvářet takové vstupní rutiny znaku, které budou dynamicky

mění vstupní BUFFER, s tím přiřazovat jeho obsahy proměnným.

Příklad č. 3

Předpokládejme, že vstupní údaje z periferního zařízení jsou umístěny v paměťové oblasti určené adresou 75 00 (HEXA). Vytvořte vstupní rutinu znaku pro přiřazení dané proměnné.

<u>Řešení :</u>	LXI H, 74FF	nastavení ukazatele dat
	RET	návrat do INPUT exekutivy

Tato vstupní rutina se musí v dekodovaném kanálu umístit na svůj ukazatel (ADR: 08 5F HEXA).

Příklad č. 4

Vytvořte vstupní rutinu pro vstup jednoho znaku

<u>Řešení :</u>	IN PORT	vstup znaku z kanálu
	INX H	inkrement ukazatele dat
	MVI M, 0	zápis END MARK
	LXI H, 7F 00	nastavení na začátek dat
	RET	návrat do INPUT exekutivy

Na základě těchto všeobecných údajů o filosofii příkazů I/O přistoupíme k podrobnému popisu jednotlivých kanálů. Pro jejich studium je předpokládána znalost periferních programovatelných obvodů typu 8255A, 8251, 8253.

II. Kanál 4 - Programovatelný paralelní vstupněvýstupní kanál (GPIO)

Prostřednictvím kanálu 4 je umožněna spolupráce s peri-

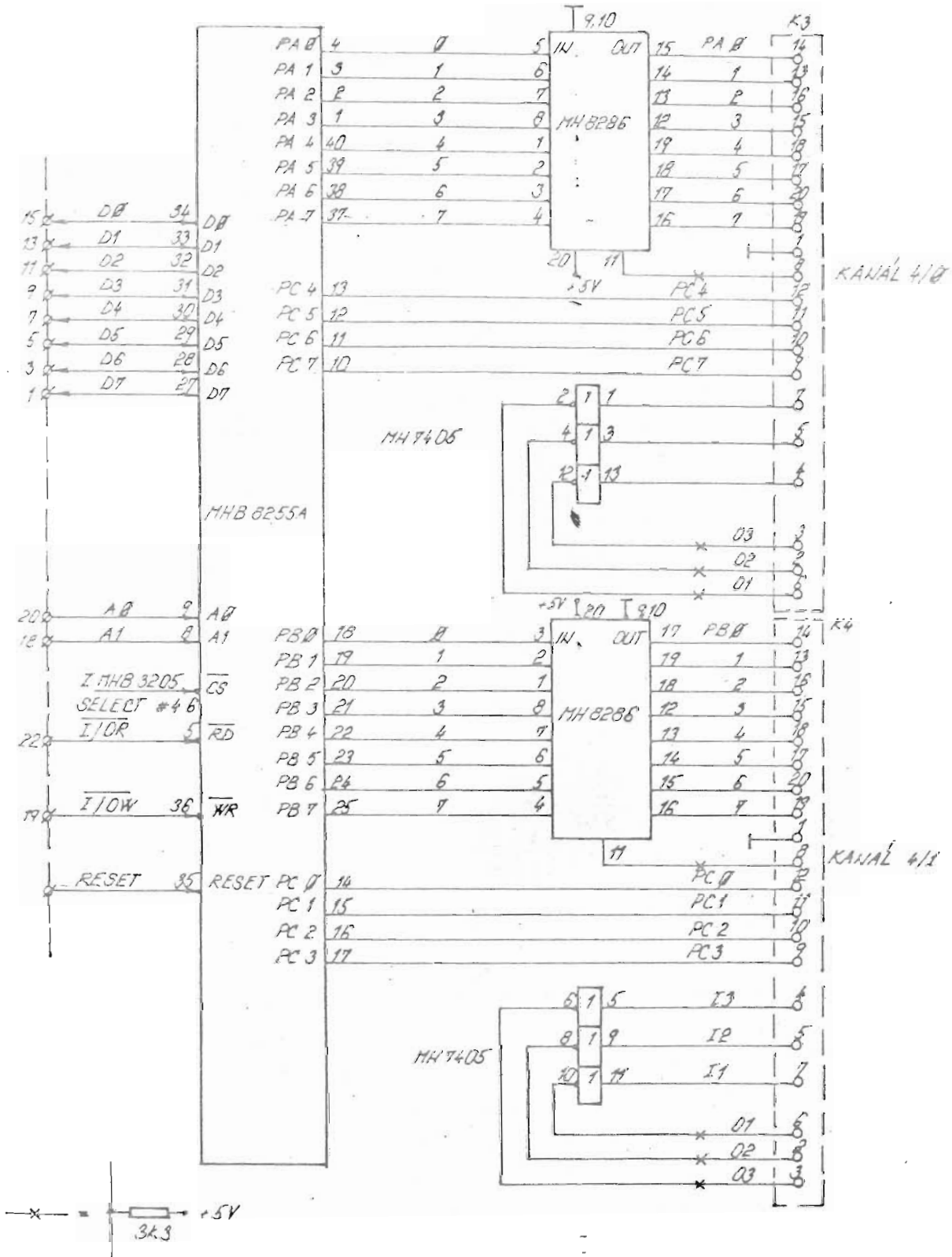
ferními vstupně-výstupními zařízeními, které pracují s paralelními osmibitovými datovými kanály.

II.1. HARDWARE

Základním prvkem kanálu je programovatelný periferní stykový obvod MHB 8255A. Na obrázku č. 1 je znázorněno schéma zapojení s ohledem na vstupně-výstupní stykové konektory. Jak je z obrázku vidět, obvod 8255A je rozdělen do dvou sekcí, skupina Ø a skupina I. Obě sekce jsou rovnocenné. Každé skupině je předurčen datový kanál; skupině Ø datový port A obvodu 8255A a skupině I port B. Datové kanály jsou odděleny výkonovými budiči TTL typu 8286, které je možno zatížit max. 30 mA na jednotlivý signál. Volba směru dat přes budič je realizována vnějším přepojem přes stykový konektor - pin 8. V klidovém stavu je na této lince úroveň LOG 1, což umožní výstup dat. Pro opačný směr je nutno připojit úroveň LOG Ø (pin č. 1). V případě, že uživatel chce ovládat směr dat programově, je možné připojit na tuto linku některý výstup

portu C. K datovému kanálu je přiřazena 1/2 portu C jako řídicí kanál. Linky tohoto portu nejsou výkonově přizpůsobené, ale mají k dispozici podporu ve třech invertorech TTL (MH 7405). Všechny uvedené linky jsou vyvedeny na 20-ti pinový FRB konektor.

Poznámka : V případě nutnosti použití úrovně LOG 1 pro inicializaci některé vstupní linky periferního zařízení můžeme použít výstup některého nepoužitého invertoru 7404, přičemž jeho vstup připojíme na úroveň LOG Ø.



OBR.Č 1 SCHEMA ZAPOJENÍ PARALELNÍHO INTERFEJSU (GPIO)

II.2. Popis režimů kanálu 4

Periferní programovatelný obvod 8255A má rozděleny své datové linky (port A, B, C) do dvou skupin tak, aby bylo možné je programovat ve třech režimech popřípadě jejich kombinacích. Datový port C má zvláštní postavení v tom, že jej můžeme rozdělit napůl a přiřadit portu A a portu B, přičemž jedna polovina může být zvolena jako výstupní a druhá jako vstupní. Při režimech 1, 2, které se vyznačují tím, že korespondence výměny a potvrzení dat mezi I/O kanálem a periferním zařízením se uskutečňuje prostřednictvím hardware za podpory činnosti vlastního obvodu 8255A. Tehdy některé linky portu C mají význam stejný jako řídící linky příslušného datového portu (A, B). V následujících částech si popíšeme jednotlivé režimy kanálu 4. Při každém režimu bude uvedena hodnota "k", která vyjadřuje řídící slovo pro nastavení kanálu do požadované funkce. Způsob programového nastavení kanálu do požadované funkce je popsán v kapitole II. 5.

II. 2. a) Režim 0 (MOD 0) - jednoduchý vstup/výstup

Režim 0 je nejjednodušší režim I/O kanálu pro datové linky portu A, B, C. Tímto režimem můžeme získat maximálně 24 linek výstupních nebo vstupních nebo jejich vzájemné kombinace. Následující tabulka č. 1 shrnuje všechny možné kombinace v režimu 0.

Tabulka č. 1

k	Skupina 0		Skupina I	
	port A	port C (vyšší)	port B	port C (nižší)
128	0	0	0	0
129	0	0	0	I

130	0	0	+ I	0
131	0	0	+ I	I
136	0	I	0	0
137	0	I	0	I
138	0	I	+ I	0
139	0	I	+ I	I
144	+ I	0	0	0
145	+ I	0	0	I
146	+ I	0	+ I	0
147	+ I	0	+ I	I
152	+ I	I	0	0
153	+ I	I	0	I
154	+ I	I	+ I	0
155	+ I	I	+ I	I

Vysvětlivky :

Znak I vstup dat (INPUT)

znak 0 výstup (OUTPUT)

znak + nutno přepojit svorky 1 - 8 na konektoru FRB (změna směru dat přes budič 8286)

Na portech, které jsou zvoleny jako výstupní, se zapsaná výstupní data uchovávají v jejich vnitřních registrech, takže můžeme s nimi pracovat jako s paměťovým místem (zápis příkazem CONTROL, čtení příkazem STATUS - viz v další kapitole).

II. 2. b) Režim 1 (MOD 1) - strobovaný vstup/výstup

Při naprogramování příslušného portu v dané skupině na tento režim, se vyhradí linky z portu C na řízení (handshake) dat.

V každé skupině jsou to tři signálové linky. Ostatní dva můžeme využít v jednoduchém režimu jako vstup nebo výstup. Je to inteligentní režim, protože odlehčuje software od vytváření strobovacích a potvrzovacích signálů, navíc ho můžeme použít i v přerušovacím systému. Protože "handshake" má pevně určenou polaritu linek, může se stát, že pro žádané periferní zařízení bude právě nutný opačný signál, proto v dané skupině jsou k dispozici tři volné inventory 7405. Následující tabulka č. 2 uvádí všechny kombinace pro datové porty A, B včetně řídicího kódu a určení linek PC pro handshake.

Tabulka č. 2

kód	skupina 0					skupina I				
	PORT A	PC4	PC5	PC6	PC7	PORT B	PC0	PC1	PC2	PC3
164	0	0	0	ACK	OB $\overline{\text{F}}$	0	INTR	OB $\overline{\text{F}}$	ACK	-
172	0	I	I	ACK	OB $\overline{\text{F}}$	0	INTR	OB $\overline{\text{F}}$	ACK	-
166	0	0	0	ACK	OB $\overline{\text{F}}$	+I	INTR	IBF	STB	-
174	0	I	I	ACK	OB $\overline{\text{F}}$	+I	INTR	IBF	STB	-
180	+ I	STB	IBF	0	0	0	INTR	OB $\overline{\text{F}}$	ACK	-
188	+ I	STB	IBF	I	I	0	INTR	OB $\overline{\text{F}}$	ACK	-
182	+ I	STB	IBF	0	0	-I	INTR	IBF	STB	-
190	+ I	STB	IBF	I	I	+I	INTR	IBF	STB	-

Vysvětlivky : I INPUT DATA

0 OUTPUT DATA

+ nutno přepojit směrem k perifernímu zařízení

INTR linke pro přerušování

ACK vyslané data jsou převzaty

OB $\overline{\text{F}}$ data pro vstup jsou přetčena

STB zápis "STROBE" dat do periferního zařízení

IBF vstupní data jsou převzata

\overline{ACK} a \overline{IBF} jsou handshake signály pro výstup dat

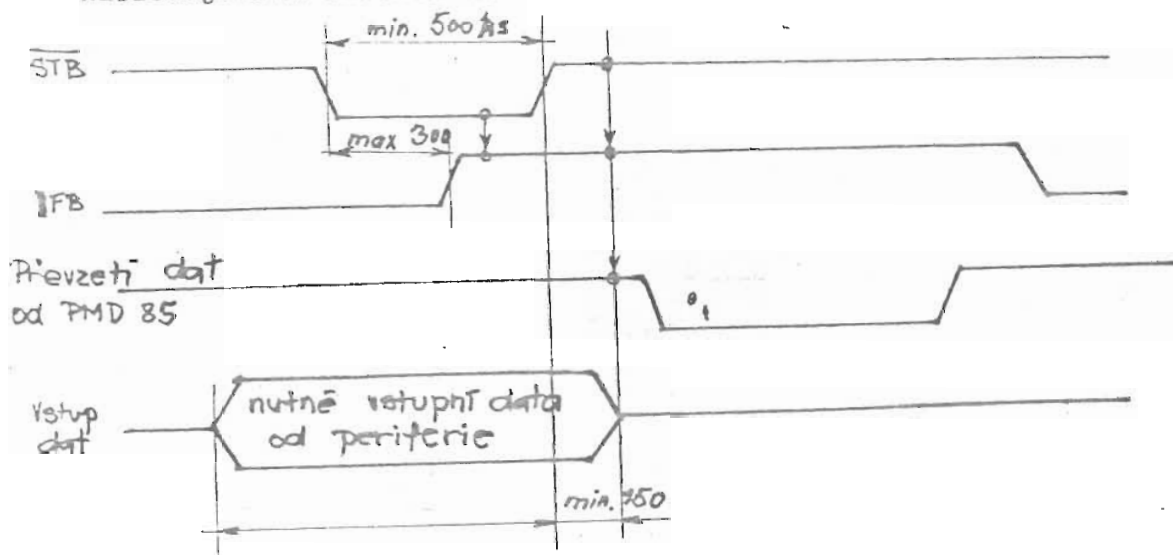
\overline{STB} a \overline{IBF} jsou handshake signály pro vstup dat

Vstupní a výstupní rutiny znaku v režimu 1, 2 testují připravenost převzetí, popřípadě odeslání dat prostřednictvím těchto linek :

- vstup dat; testování na signál IBF (INPUT BUFFER FULL)
- výstup dat; testování na signál INTR (INTERRUPT)

Signál INTR (PC0 nebo PC3) je aktivován až po jeho uvolnění. Toto uvolnění se realizuje set/reset funkcí bitů PC2 u skupiny 0 a PC6 u skupiny 1.

Vzájemné časové průběhy pro režim 1 vstup/výstup jsou uvedeny na následujících obrázcích.



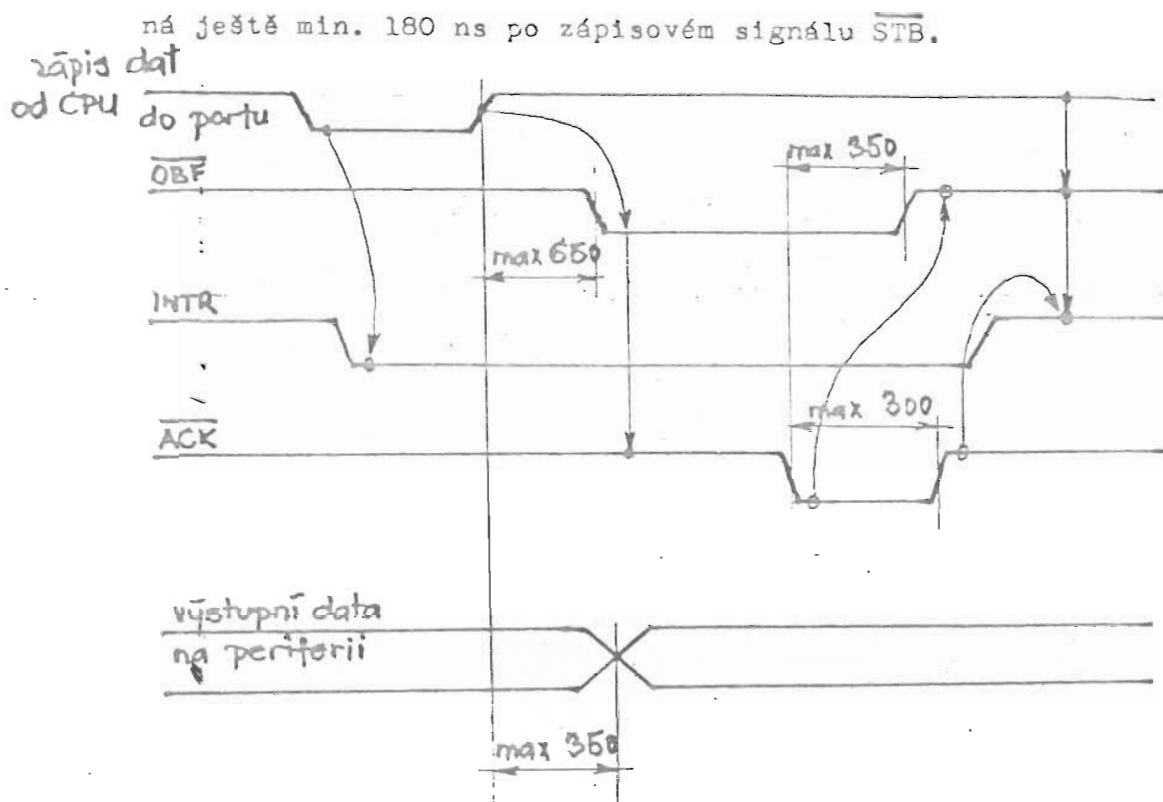
Obr. č. 2 - Časový průběh vstupu dat v režimu 1.

\overline{STB} ... zápisový signál vstupních dat z periferie

\overline{IBF} ... potvrzovací signál ze vstupního kanálu, převzetí dat

Tento signál bude shoden až po převzetí přijatých dat

signálem \overline{RP} z CPU. Vstupní data z periferie musí být plat-



Obr. č. 3 - Časový průběh výstupu dat v režimu 1.

Jak je z časového průběhu zřejmé, je potřebné testovat převzetí vyslaných dat na lince INTR, která musí mít úroveň LOG 1.

\overline{OBF} ... výstupní port má data vyslaná do periférie uložena

\overline{ACK} ... potvrzení periférie, že převzala data

INTR .. signál pro přerušení, v případě, že není uvolněno nastavením linek PC2, PC6, jeho úroveň je LOG 1.

11.2.c) Režim (MÓD 2) - obousměrný strobovaný vstup/výstup

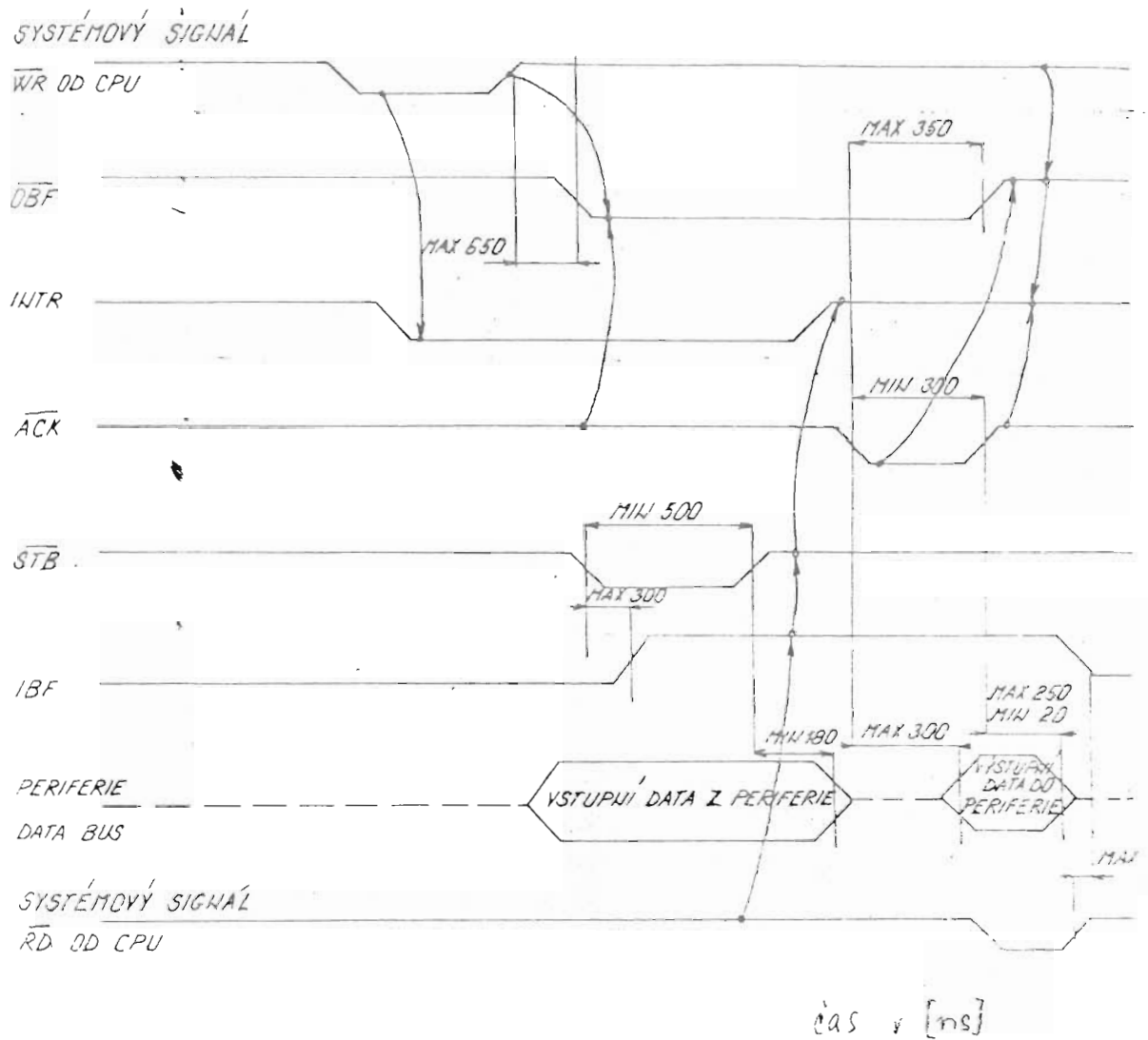
Režim 2 poskytuje kanálu 4 speciální použití v případech, kdy chceme využívat datový kanál obousměrně a za podpory "handshake" Tuto vlastnost má jen kanál ve skupině Ø. Praktické použití je při paralelním spojení dvou počítačů. V tabulce č. 3 jsou uvedeny kombinace v případě, že PORT A bude inicializován v režimu 2.

Tabulka č. 3

kód	skupina Ø - data obousměrně				skupina I			poznámka	
	PC4	PC5	PC6	PC7	PORT B	PCØ	PC1	PC2	
194	STB	IBF	ACK	OBF	+ I	0	0	0	Port B v režimu Ø
192	↓	↓	↓	↓	0	0	0	0	
195	↓	↓	↓	↓	+ I	I	I	I	
193	↓	↓	↓	↓	0	I	I	I	
198	↓	↓	↓	↓	+ I	0	0	0	Port B v režimu 1
196	↓	↓	↓	↓	0	0	0	0	
199	↓	↓	↓	↓	+ I	I	I	I	
197	↓	↓	↓	↓	0	I	I	I	

Význam signálů u skupiny Ø je obdobný jako u režimu 1, vstup/výstup.

Vzájemný vztah korespondujících signálů v režimu 2 jsou v časovém průběhu na obr. č.4.



OBRAZ 4 ČASOVÉ PRŮBĚHY V REŽIMU 2

II.3.3. Kombinované režimy

Z uvedených režimů 0, 1, 2 je možno sestavit i kombinované režimy mezi skupinami 0 a I. Platí zde tytéž časové průběhy a závislosti, které byly uvedeny u jednotlivých režimů. Mění se pouze číslo (kód) pro nastavení řídicího registru. Tento parametr lze získat v části 5, která popisuje způsob programování režimů kanálu 4. Závěrem uvádíme souhrnnou tabulku pro port A, B, C, v režimech, ve kterých mohou být dané linky využívány,

Sk.	PORT	MÓD 0		MÓD 1		MÓD 2
		INPUT	OUTPUT	INPUT	OUTPUT	LEN PORT A
0	A	IN	OUT	IN	OUT	Obousměrně
	PC4	IN	OUT	STB	IN/OUT	STB
	PC5	IN	OUT	IBF	IN/OUT	IBF
	PC6	IN	OUT	IN/OUT	ACK	ACK
	PC7	IN	OUT	IN/OUT	OBF	OBF
I	B	IN	OUT	IN	OUT	-
	PC0	IN	OUT	-	-	IN/OUT
	PC1	IN	OUT	IBF	OBF	IN/OUT
	PC2	IN	OUT	STB	ACK	IN/OUT
	PC3	IN	OUT	-	-	-

Tabulka č. 4 - Souhrnná tabulka využití portu A, B, C

II.3. Software pro OUTPUT

Programovatelný interface 8255A, který je základním prvkem kanálu č. 4 může pracovat ve třech režimech (MÓD 0, 1, 2). Ke každému režimu je vytvořena výstupní rutina znaku. Na obr. č. 5 je znázorněna ve strojovém kódu exekutiva, která zabezpe-

čuje "zahníždění" zvolené výstupní rutiny.

209F : POP H	návrat programového ukazatele	
INX H		2
CALL 16D7	matematický výraz do akumulátoru (0 - 255) mozku	
ANI 07		2
PUSH H	uchování programového ukazatele	
LXI H, TABOUT	nastavení tabulky výstupních ru- tin	2
MVI B, 0		
MOV C, A	offset	
RAD B	výpočet místa v tabulce	
MOV D, H	adresa příslušné výstupní rutiny	
MVI H, 20	znaku	
SHLD 0015	uložení na ukazatel 0015	2
POP H	návrat prog. ukazatele	2
CALL 006D	kontrola syntaxi středník	
DB ";"	ASCII kód "středník"	
JMP 20 3F	skok do pokračování PRINT	

TABOUT: C4	MÓD 0 port A	
C7	MÓD 0 port B	
CA	MÓD 0 port C	2
CD	MÓD 1 port A	
D9	MÓD 1 port B	2
C	MÓD 2 port A	
E5	uživatel	0
E8	uživatel	

20C4 : OUT 4C
RET

PORT A

20C7 : OUT 4D

RET

PORT B

20CA : OUT 4E

RET

PORT C

20CD : MOV C, A

uchování výstupních dat

20CE : IN 4E

čtení portu C

ANI 08

maska pro bit INTR (PC3)

JZ 20CE

MOV A, C

vrácení výstupních dat

OUT 4C

výstup dat na port A

RET

20D9 : MOV C, A

20DA : IN 4E

ANI 01

JZ 20DA

MOV A, C

OUT 4D

RET

pro port B

20E5 : JMP FFFF

20E8 : JMP FFFF

uživatelské výstupní rutiny znaku

Obr. č. 5 - Výstupní rutiny znaku pro kanál č. 4

OUT 40

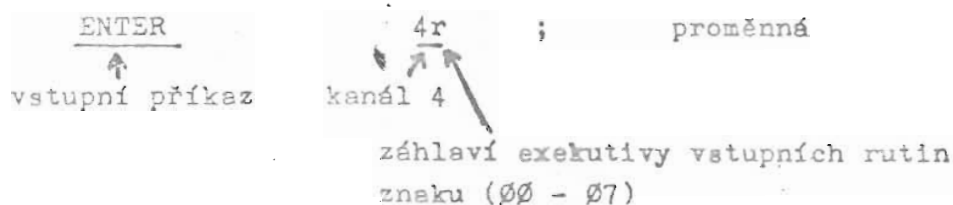
výstup na port A

RET

2. krok - na adresu 20E6 uložíme vektor rutiny
VYSRUT.

II. 4. Software pro ENTER

Filosofie vstupní exekutivy je obdobná jako u příkazu OUTPUT.
Její úlohou je vložit žádanou vstupní rutinu znaku do ukazatele
085F.



Exekutiva pro vkládání zvoleného režimu vstupu znaku do
datového portu je ve strojovém kódu uvedena na obr. č. 6.

```
21B2 : POP H
      INX H
      CALL 16D7
      ANI 07
      PUSH H
      LXI H, TABENT
      MVI B, 0
      MOV C, A
      DAD B
      MOV L, M
      MOV H, 21
```

komentář stejný jako u OUTPUT

OUT 4C

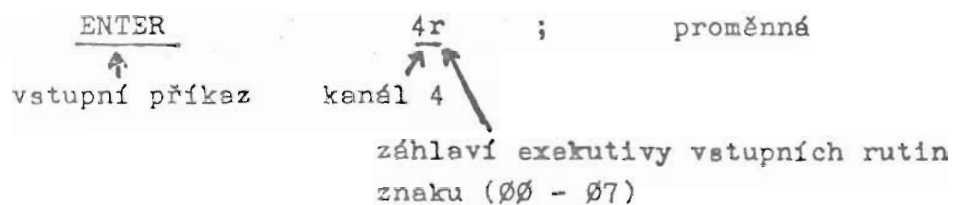
výstup na port A

RET

2. krok - na adresu 20E6 uložíme vektor rutiny
VYSRUT.

II. 4. Software pro ENTER

Filosofie vstupní exekutivy je obdobná jako u příkazu OUTPUT.
Její úlohou je vložit žádanou vstupní rutinu znaku do ukazatele
085F.



Exekutiva pro vkládání zvoleného režimu vstupu znaku do
datového portu je ve strojovém kódu uvedena na obr. č. 6.

```
21B2 : POP H
      INX H
      CALL 16D7
      ANI 07
      PUSH H
      LXI H, TABENT
      MVI B, 0
      MOV C, A
      DAD B
      MOV L, M
      MIV H, 21
```

komentář stejný jako u OUTPUT

SHLD 085F

POP H

CALL 006D

DB ";"

JMP 214E

komentář stejný jako u OUTPUT

skok do pokračování příkazu INPUT

TABENT : D7

DA

DD

E4

FD

E4

F7

FA

MÓD 0 PORT A

MÓD 0 PORT B

MÓD 0 PORT C

MÓD 1 PORT A

MÓD 1 PORT B

MÓD 2 PORT A

uživatel

uživatel

21D7 : IN 4C

DB "01"

21DA : IN 4D

DB "01"

21DD : IN 4E

MOV M, A

INX H

JMP 220F

výstupní rutiny znaku pro MÓD 0

21E4 : IN 4E

ANI 20

JZ 21E4

IN 4C

MOV M, A

INX H

handshake signál IBF

zesnímání dat port A

uložení do vstupního BUFFERu

zvýšení ukazatele BUFFERu

21EF	:	CPI 0A	}	test na ukončovací znak "LF"
		JNZ 21E4		
		JMP 220D		skok na závěr vstupu
21F7	:	JMP FFFF		uživatelská rutina
21FA	:	JMP FFFF		uživatelská rutina
21FD	:	IN 4E	}	
		ANI 02		handshake signálu
		JZ 21FD		
		IN 4D		zesnímaní dat - port B
		MOV M, A		uložení do vstupního BUFFERu
		INX H		zvýšení ukazatele BUFFERu
2208	:	CPI 0A	}	test na "LF"
		JNZ 21FD		
220D	:	DCX H		snížení ukazatele BUFFERu
		DCX H		snížení ukazatele BUFFERu
220F	:	MVI M, 0		zapsání END MARK (00)
		LXI H, 7F00		nastavení na začátek
		RET		

Obrázek č. 6 Vstupní rutiny znaku pro kanál č. 4

Záhlaví exekutivy nebudeme vysvětlovat, protože jeho úloha je stejná jako u příkazu OUTPUT, tedy zjistit na základě obsahu výrazu "r" v příkazu ENTER vektor dané vstupní rutiny znaku a uložit na ukazatel 085F. Rozebereme si blíže tvar vstupní rutiny, aby bylo možno vytvářet vlastní uživatelské vstupy.

Nejjednodušší vstup je v režimu (MOD) 0. Zde se seřadí stav vstupního kanálu (portu) a uloží se do vstupního BUFFERu. Za tímto vstupním údajem se vloží (ADR. 220F) ukončovací znak (END MARK) s hodnotou 00. Pro přebrání vstupního BUFFERu procedurou příkazu ENTER se nastaví začátek BUFFERu minus 1 (adr. 7F00). Je-li zvolený režim vstupního kanálu MOD 1, který se používá pro vstup bloku dat je situace vstupu odlišná ve dvou bodech.

- a) Pro každá vstupní data existuje korespondence (handshake) s periferním zařízením, které zabezpečuje obvod 8255A svou interní logikou. Platnost přebíraných vstupních dat se testuje na lince INTR (pin PC0, PC3).
- b) Vstupní blok dat z periferního zařízení musí být ukončen znaky CRLF (0D0A HEXA). Test ukončení se uskutečňuje na přijetém znaku LF. Po úspěšné komparaci je nutno tyto dva znaky zrušit ze vstupního bufferu a nahradit je ukončovací značkou (END MARK) 00 HEXA, která je žádoucí z hlediska interpreteru BASIC. V případě, že vstupní periferní zařízení poskytuje jiné ukončovací znaky (ne CRLF) je možno modifikovat příslušnou vstupní rutinu znaku příkazem POKE.

Příklad č. 6 : Upravte vstupní rutinu znaku pro režim (MOD) 1, port A pro periferní zařízení, které ukončuje vstupní blok znaky "SPCR" (200D) HEXA.

Řešení : Nahradíme na adrese 21F0 ukončovací znak LF znakiem

POKE	8688	,	13
			
	adresa 21F0		znak CR

Upozornění : Vstupními daty do kanálu 4 prostřednictvím vstupních

an znaku ^ až 5 mohou být data jen ve tvaru ASCII pro přiřazení numerickým proměnným. Pro řetězcové proměnné mohou být vstupní data ve tvaru binárním mimo těchto byte :

00	END MARK
2C	čárka (ASCII)
0A	ukončovací znak

protože byte 00 nám předčasně ukončí přebírání dat ze vstupního bufferu a byte 2C nám rozdělí vstupní buffer na dva vstupní bloky. Obrana proti této činnosti je možná jen v případě vytvoření speciální uživatelské vstupní rutiny znaku.

Příklad č. 7 : Vytvořte vstupní rutinu pro kanál 406, který by snímal 4 BCD čísla z převodníku, v režimu MÓD 1.

Řešení : Použijeme celý kanál 4 (skupina 0 + skupina I) pro připojení převodníku. Korespondenční linky budou použity ze skupiny I. Pro převod obsahu 2 x BCD do tvaru 2 x ASCII použijeme rutinu z operačního systému (viz příručka část III) s adresním vektorem 813B. Na uživatelskou pozici (406), adr. 21F8 umístíme vektor následující rutiny :

BCDENT :	IN 4E	}	handshake
	ANI 02		
	JZ BCDENT		
	IN 4C		vstup dat na portu A
	CALL 813B		převod do 2 x ASCII
	INX H		zvýšení ukazatele
	IN 4D		vstup dat na portu B
	CALL 813B		převod do 2 x ASCII

INX H	zvýšení ukazatele
JMP 220F	závěr vstupní rutiny znaku

Syntax : ENTER 406; A (I)
nebo ENTER 406; A

II. 5. Programování režimu kanálu 4 pro OUTPUT/ENTER

Pracovní režim (MODE) periferního obvodu 8255A je potřeba nastavit do inicializované pozice. Tato pozice se nuluje stisknutím tlačítka RESET + SHIFT, přičemž všechny jeho porty (A, B, C) jsou inicializovány jako vstupní linky. Avšak jak už bylo řečeno, na portu A, B jsou připojeny TTL budiče, které mají nastaven přenos dat na výstup z kanálu, takže na jejich výstupu bude v konečném důsledku úroveň LOG 1. Cílem této části je popsat jednotlivé pracovní režimy obvodu 8255A a způsob jejich nastavení.

Příkaz, který nám umožňuje tuto činnost je :

CONTROL 4, r; k

kde "r" představuje přístup k datovým portům, nebo k řídicímu registru pracovního režimu - viz následující tabulka.

Tabulka č. 5

r	Význam	Poznámka
0	PORT A	Zápis dat
1	PORT B	Zápis dat
2	PORT C	Zápis dat
3	SET/RESET LINKY PORTU C	Nastavení bitu na port C
	ŘÍDÍCÍ REGISTR	Zápis řídicího slova

znamená celočíselnou hodnotu v rozsahu 0 - 255 a má význam podle hodnoty "r". V případě, že r = 3 jde o hodnotu řídicího slova, nebo o SET/RESET bit linky portu C, jinak představuje hodnotu dat pro zápis do daného portu.

Nastavení pracovního režimu se uskutečňuje zápisem vhodného řídicího slova do řídicího registrů :

CONTROL 4, 3 ; k [1, m]

- hodnotu "k" získáme z následující tabulky č. 6 jako součet jednotlivých vah požadovaného úkonu kanálu 4

Nastav řídicí slovo	Skupina 0				Skupina I		
	Obousměrný přenos na portu A v režimu 2	Režim 1 - podpora v handshake	Volba směru dat na portu A výstup dat = 0 vstup dat = 16	Volba směru dat na portu C výstup dat = 0 vstup dat = 8	Režim 1 - podpora v handshake	Volba směru dat na portu B výstup dat = 0 vstup dat = 2	Volba směru dat na portu C výstup dat = 0 vstup dat = 1
28	64	32	16	8	4	2	1

Tabulka č. 6 - Nastavení řídicího slova režimu práce kanálu 4

- parametry "l, m" uvedené v hranaté závorce, mají speciální význam. Uvádějí se pouze při volbě režimu 1 nebo 2 pro příslušnou skupinu. Jejich hodnoty umožňují bitově nastavit linky portu C, které jsou nutné v interní logice obvodu 8255A pro podporu handshake - viz popisovaný režim 1, 2.

Všeobecně můžeme bitově manipulovat se všemi linkami portu C.

Následující tabulka uvádí všechny možné kombinace pro bitovou manipulaci linek portu C.

Bit	Nastavení bitu	Nulování bitu
0	1	0
1	3	2
2	5	4
3	7	6
4	9	8
5	11	10
6	13	12
7	15	14

Tabulka č. 7 - parametry řídícího slova pro bitové ovládání linek portu C

Poznámka : Jak je vidět z tabulky, platí pro set/reset bitu vzorec (příslušný bit x 2) + žádaný stav bitu

Hodnoty l, m pro zvolený režim 1, 2 a danou skupinu určíme z tabulky č. 8.

Režim	skupina 0		skupina I	
	vstup	výstup	vstup	výstup
1	9	13	5	5
2	13,9		-	-

Tabulka č. 8 - Parametry řídicího slova pro režim 1 a 2.

Příklad č. 8 : Nastavte linky 4, 5 portu C ve skupině 0 na úroveň LOG 1.

Řešení : CONTROL 4, 3; 9, 11

Příklad 8 a : Nastavte kanál 4 pro výstup dat ve skupině 0 a vstup dat ve skupině I. Oba datové kanály budou mít podporu v handshake (korespondenční linky pro odevzdávání platných dat).

Řešení : Použijeme příkaz CONTROL, pro který vyhledáme v tabulce č. 6 (2) příslušnou hodnotu "k" a v tabulce č. 8 parametr "l, m".

CONTROL 4, 3; $\underbrace{128 + 32 + 4 + 2}_{\substack{\text{do řídicího} \\ \text{registru} \\ \text{(režim)}}}, \underbrace{13}_{\substack{\text{set bit} \\ \text{PC6}}}, \underbrace{5}_{\substack{\text{set bit} \\ \text{PC2}}}$

II. 6. STATUS kanálu 4

K datovým kanálům a portům můžeme přistupovat jednoduchým příkazem - funkcí, která umožňuje přiřazovat dané zvolené proměnné obsah žádaného portu. Syntax funkce je :

$\xrightarrow{\text{proměnná}} P = \text{STATUS} \xrightarrow{\text{kanál 4}} 4, r \xleftarrow{\text{registr - port A, r = 0}} r$
port B, r = 1

port C, r = 2

Poznámka : Obsah řídícího registru (r = 3) není možno číst.

V případě režimu MOD 0 poskytne nám funkce STATUS hodnotu žádaného datového portu. Bývá zvykem v tomto režimu obsadit řídící nebo stavové signály periferního zařízení na port C. Potom stav periferního zařízení získáme funkcí :

$$S = \text{STATUS } 4, 2$$

Dalším zpracováním proměnné S můžeme zajistit datový přístup z periferie. Aby bylo možné lépe testovat jednotlivé bity na určitý stav, interpreter BASIC - G nám poskytuje funkci :

$$P = \text{BIT } K, M$$

kde - P je přiřazovací proměnná dané funkci

- K je celočíselná hodnota v rozsahu 0 - 255

- M je váha bitu (2^M) na proměnné K, kterou funkce BIT testuje.

Poznámka : Proměnná P může nabýt jen dvou hodnot; 0 nebo 1

Obě funkce můžeme sloučit do tvaru např.:

$$P = \text{BIT } (\text{STATUS } 4, 2), 3$$

přičemž proměnná P nebude hodnoty signálu na lince PC3.

V ostatních režimech (1,2) nám může port C poskytnout stavové slovo, které obsahuje signály, které korespondují s periferním zařízením. Stavová slova pro jednotlivé režimy jsou uvedena v následující tabulce č. 9.

režim
MOD 1-vstup
MOD 1-výstup
MOD 2

Poznámka

Příklad č. 9

Řešení : 1. 1

CON
říd
rež
vst
úvo
2.

OUT
kar
sku
MOI

režim	BIT 7	6	5	4	3	2	1	0
MOD 1-vstup	I/O	I/O	IBF ₀	INTE ₀	INTR ₀	INTE _I	IBF _I	INTR _I
MOD 1-výstup	OBF ₀	INTE ₀	I/O	I/O	INTR ₀	INTE _I	OBF _I	INTR _I
MOD 2	OBF ₀	INTE ₀	IBF ₀	INTE _I	INTR ₀	XXXX	XXX	XXXX

Poznámka : Indexový znak znamená příslušnou skupinu.

Příklad č. 9 : Naprogramujte skupinu 0 pro připojení tiskárny např. CONSUL 2111 v režimu 1 (podpora handshake) a testujte signál připravenosti "READY PRINTER" na lince portu PC 5.

Řešení : 1. krok : naprogramování kanálu portu A ve skupině 0 na režimu 1, sečtení volných linek na portu C v režimu INPUT, uvolnění INTR pro testování "handshake".

CONTROL 4, 3; 128 + 32 + 8, 13
řídící slovo
režim 1
vstup PC 4, 5
uvolnění INTR linky

2. krok : použití příkazu OUTPUT pro výstup dat na kanál portu A ve skupině 0

OUTPUT 403; výraz
kanál 4
skupina 0, handshake
MOD 1

3. krok : testování připravenosti tiskárny

IF (BIT STATUS 4, 2, 5) = THEN PRINT "ERROR"

Pokud je hodnota uvedená v závorce rovna 0 vypíše se text ERROR, jinak se pokračuje v dalším interpretování.

První krok používáme na začátku programu, kde provádíme inicializaci programu. Krok 3 je nejlépe sestavit do programu, který nám bude testovat připravenost periferního zařízení před každým jeho použitím. Může mít takovýto tvar :

```
1000 IF (BIT STATUS 4, 2, 5) THEN RETURN
```

```
1010 DISP "PRINTER IS NOT READY"
```

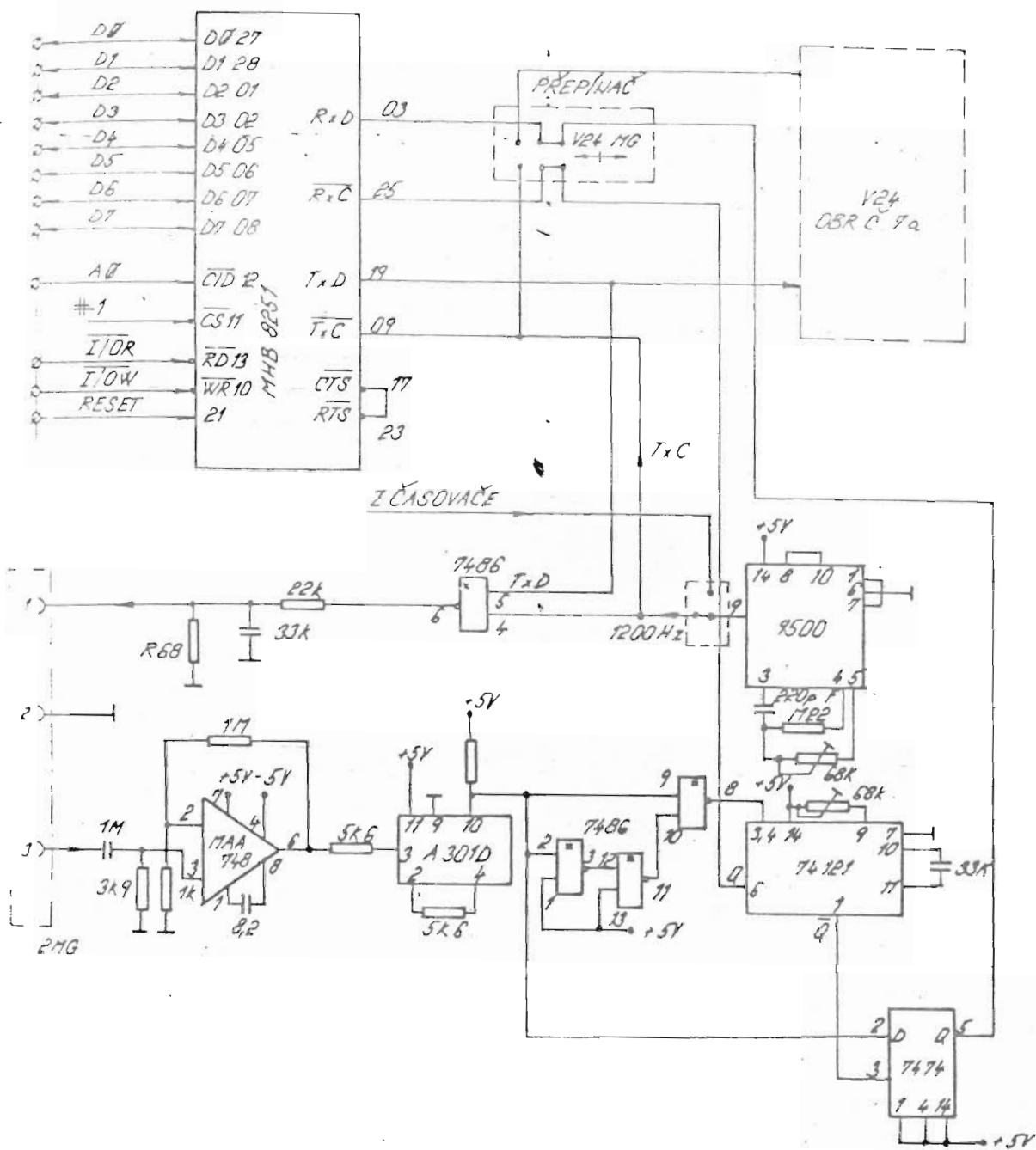
```
1020 GOTO 1000
```

Touto částí kapitoly II byl uzavřen kanál 4, který umožňuje připojit periferní zařízení po paralelním datovém kanálu např. děrovač DT 105, snímáč FS 1 503, tiskárnu Consul 2111, tiskárnu DARO, A/D převodník, JOYSTICK, atd. V příloze této příručky uživatel najde několik praktických zapojení uvedených periferních zařízení.

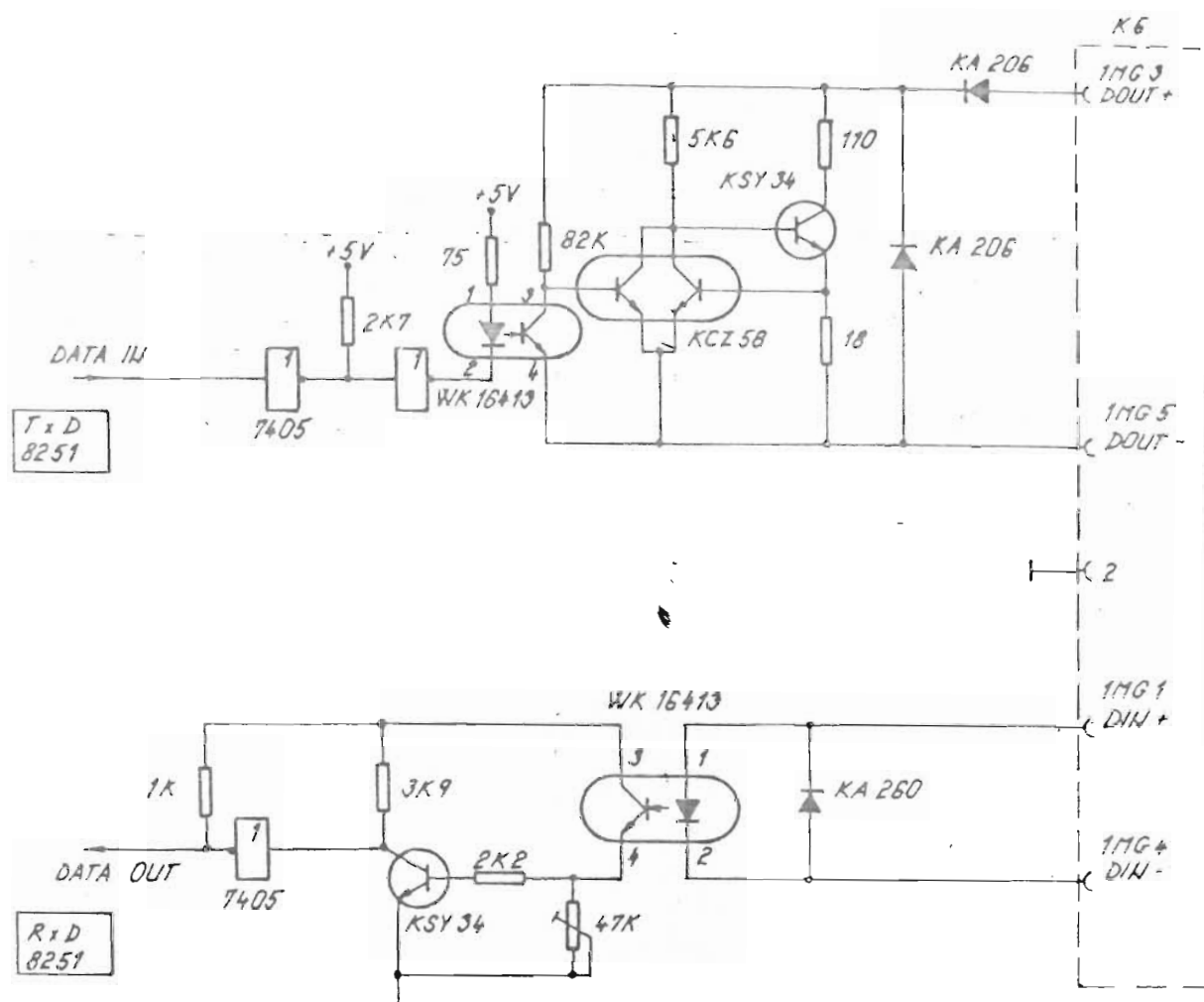
III. Kanál 1 - sériový vstupně-výstupní kanál

III. 1. HARDWARE

Sériový vstupně-výstupní kanál 1 je charakterizován vlastnostmi obvodu USART MHB 8251. K tomuto obvodu se připojují dva samostatné bloky viz obr. č. 7, které se prostřednictvím přepínače aktivují k obvodu USART. Jeden blok je pro interface s ka-

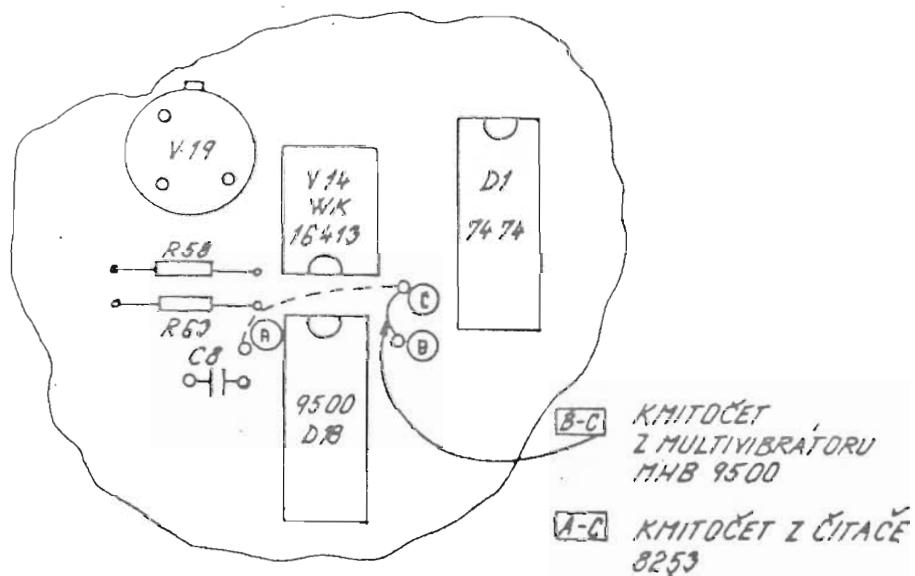


OBR.Č. 7 SCHÉMA ZAPOJENÍ INTERFEJSU PRO MAGNETOFON



OBRAZKA 7a SCHEMA ZAPOJENÍ INTERFEJSU V.24

INTERFACE KARTA



OBR. 8 ÚPRAVA GENERÁTORU HODIN PRO USART

zetovým magnetofonem sloužícím jako vnější paměť a druhý blok je interface V 24. Tento blok je ve zjednodušené formě, a to jen pro data OUT/INP oddělené optočleny. Jak ze schématu vidíte, vzorkovací frekvence je generovaná pevně s podporou obvodu MHB 9500 na výstup 1 200 Hz. V případě, že uživatel chce tento kmitočet změnit, může využít připojení časovače 8253 jako generátoru pulzů. Tento požadavek však vyžaduje provést dvě činnosti; první je zásah do hardware viz obr. č. 8, kde je nutno odpojit generátor pulzů od MHB 9500 a připojit 8253, druhý krok je programové nastavení čítače 8253 na požadovanou frekvenci. Způsob programování čítače 8253 je podrobně vysvětlen v kapitole 4.

Styk V 24 obsahuje jen datové linky, které jsou pasivní. Pro některé aplikace lze použít potenciály - 5V a + 12V, které jsou vyvedeny na aplikačním konektoru I/O. Pro lepší pochopení vlastností těchto kanálů popíšeme si technické a programové možnosti obvodu USART MHB 8251.

III. 2. Popis obvodu 8251

MHB 8251 je programovatelný obvod pro připojení periférií se sériovým přenosem dat. Lze jej použít pro vysílání a příjem dat v synchronním nebo asynchronním režimu. Provozní režim, počet údajových bitů, způsob ukončení poritů jsou dány programem. Stejně je dáno programem použití interní nebo externí synchronizace a počet synchronizačních znaků při synchronizovaném provozu a poměr mezi délkou údajového bitu a periodou hodinových pulzů a délka STOP bitu při synchronním provozu.

Samotná architektura vysílače a přijímače umožňuje plně duplexní provoz při zachování plné kompetibility s obvody TTL.

Řídící slovo, které určí režim činnosti obvodu, se skládá

se svou částí : instrukce pro druh provozu a povelová instrukce. Instrukce pro druh provozu musí následovat po uvedení obvodu do čekacího - nefunkčního stavu signálem RESET nebo stavem $IR = 1$ povelové instrukce. V případě, že instrukcí pro druh provozu byl nastaven synchronní režim, musí po ní následovat zápis daného počtu synchronních znaků. Zápis-řídícího slova je ukončen zápisem povelové instrukce, kterou je možno zapsat i během přenosu údajů, a tak změnit režim obvodu. Během přenosu údajů můžeme přečíst i stavovou informaci, vyhodnocením které lze sledovat stav přenosu. Rozlišení údajů (dat) a řídícího slova nebo stavového slova je podle vstupu C/\bar{D} (pin 12). Obvod 8251 má pro řízení údajů mezi periferním zařízením modémového typu vyvedeny řídící linky \overline{DSR} , \overline{DTR} , \overline{CTS} a \overline{RTS} (piny 22, 24, 23, 17). Stavy těchto linek jsou obsaženy v povelové instrukci resp. stavovém slově a vstup \overline{CTS} slouží pro hardwareové blokování vysílání sériového znaku. V mikropočítači PMD 85 jsou spojeny řídící linky \overline{CTS} - \overline{RTS} (pin 17 - 23).

Vysílač a přijímač jsou řízeny samostatnými hodinovými pulzy $T \times C$, $R \times C$ (piny 9, 25), jejichž frekvence spolu s obsahem instrukce pro druh provozu udává časové parametry sériového přenosu. Při vysílání údajů je sériový údaj doplněn resp. upraven podle obsahu registru instrukce pro druh provozu. V případě synchronního provozu během nevysílání údajů dochází k vysílání synchronizačních znaků a při asynchronním provozu výstup $T \times D$ (pin 19) je stav LOG 1. Stav vysílací paměti a vysílací vyrovnávací paměti obvodu 8251 jsou indikovány signály $T \times E$ a $T \times RDY$ (piny 18, 15). Vysílání sériového znaku je uvolněno resp. blokováno stavem $T \times EN$ (pin 18) povolené instrukce a vstupem \overline{CTS} přičemž pro uvolnění vysílání musí platit log. součin $\overline{CTS} \cdot T \times EN = 1$, v opačném případě je $T \times D$ ve stavu 1. Přijímač

ze sériové posloupnosti znaků přivedený na vstup R x D (pin 3) sestaví paralelní údaj, který uloží v přijímací paměti obvodu. Řídící logika podle stavu zápisové instrukce pro druh provozu a povolené instrukce přijatý znak vyhodnotí a nastaví chybové příznaky PE, FE, OE stavového slova. Stav paměti přijímače je indikován signálem R x RDY (pin 14). Uvolnění a blokování přijímače je řízeno stavem R x EN povelové instrukce. Na vynechání příznakových chyb stavového slova slouží signál EN povelové instrukce.

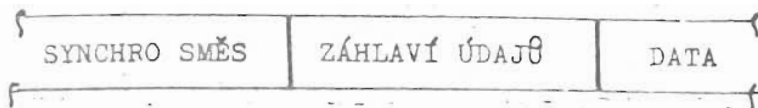
Pokud je naprogramován synchronní provoz s interní synchronizací, přijímač začne pracovat ve vyhledávacím režimu (HUNT MODE), který spočívá v porovnání přijatého znaku (přijatého po jedno-tlivých bitech) s obsahem registru synchronizačního znaku. Po dosažení shody vyhledávací režim se ukončí a dosažení synchronizace se indikuje nastavením signálu SYNDY (pin 16) do stavu LOG 1. Při ztrátě synchronizace vyhledávací režim lze spustit signálem EH povelové instrukce. V případě externí synchronizace, dosažení shody synchronizace se musí indikovat přivedením vstupu SYNDY do stavu LOG 1.

Pracovní režim USARTu je dán vhodným naprogramováním. Druhy pracovních režimů a jejich způsob naprogramování si detailněji probereme ve zvláštní části této kapitoly. Nyní si vysvětlíme programové procedury pro vstup/výstup dat. Protože tyto jsou odlišné pro jednotlivé interface (V 24 - MG) probereme je jednotlivě.

III. 3. Software vstup/výstup pro magnetofon

Filosofie formátu zápisu údajů na MG pásku je postavena

na zápise uceleného bloku dat s patřičnou údajovou hlavičkou a synchronizační směsí. Poloupnost těchto částí je znázorněna na následujícím obrázku :



Tyto tři základní části jsou vytvořeny následovně :

- SYNCHRO SMĚS : je to konstantní zápis skládající se z 16 byte 00 a 16 byte 55 (HEXA), které mají sloužit jako synchro směs pro záhlaví údajů
- ZÁHLAVÍ ÚDAJŮ: je to proměnlivá složka, jejíž obsah udává důležité údaje o následujícím datovém bloku. Má konstantní délku 14 + 1 byte, přičemž poslední byte je dán implicitně a představuje kontrolní sumu nad 14 byte.
Podmínkou začátku čtení datového bloku je současné nalezení obsahu synchro směsí a záhlaví údajů včetně kontrolní sumy.

Záhlaví údajů je koncipováno na tyto údaje :

číslo záznamu	typ	adresa	délka	komentář
1 byte	1 byte	2 byte	2 byte	8 byte

- číslo záznamu : je identifikátor záznamů na pásce MG. Tento parametr udává uživatel, a proto je nutno ho zaznamenávat písemně na vizitku kazety, aby byla zachována kontinuita číslování záznamů.
- typ záznamu : je parametr, který implicitně vkládá systém pod kterým právě uživatel pracuje. V současné době jsou k dispozici 3 typy záznamů :

- a) záznam operačního systému - znak ?
- b) záznam programu BASIC - G - znak
- c) záznam dat pro program BASIC - G - znak D

Podle toho, o který typ záznamu jde, jsou určovány další parametry záhlaví.

- adresa + délka záznamu obsahují parametry, které určují konkrétní počáteční adresu operační paměti, kam budou data uložena (kromě typu dat) a jejich délka.

Pro uvedené tři typy záznamů má adresa a délka záznamu následující význam :

- a) typ "?" - adresa a délka je dána přímo uživatelem
- b) typ ">" - adresa a délka je určena implicitně BASICem a má hodnotu :

adresa = 2400 (HEXA); délka = hodnota ukazatele 5E7A (HEXA)

- c) typ "D" - adresa nemá význam, délka je určena implicitně z délky pole.

- komentář - je nepovinná část záhlaví, která obsahem má blíže charakterizovat záznam v případě, že komentář není dán, v tomto prostoru se nachází znak SPACE (20 HEXA).

- data - tento blok představuje konkrétně zaznamenaná data v HEXA formě. Velikost bloku je dána hodnotou DÉLKA v záhlaví údajů. Na konci datového bloku je kontrolní suma zabírající 1 byte.

Pro všechny typy záznamů jsou pro vstup/výstup magnetofon tyto společné rutiny :

(adresa 8D89), které inicializuje obvod USART pro asynchronní režim, 1 STOP bity, 1 x vzorkovací frekvence, bez parity, osm bitů.

SAVE (adresa 8D85), které zabezpečuje záznem synchro směsí (16 x 66 - 16 x 55), záhlaví údajů a vlastního datového bloku. Vstupním parametrem této rutiny je obsah 14 byte registru, který je umístěn v zápisníku operačního systému od adresy C1B2 (HEXA).

SAVE (adresa 8D6C), je rutina, která zabezpečuje zápis datového bloku. Vstupním argumentem je registr HL, který určuje paměťové místo a registr DE, který udává počet (délku) dat.

READY (adresa 8D81) testovací rutina připravenosti obvodu USART pro zápis dalšího byte.

- TREADY (adresa 8E0E) testovací rutina připravenosti obvodu USART pro čtení dalšího byte.

- LHLAV (adresa 8DE2) rutina zabezpečuje přečtení a uložení záhlaví záznamu do pomocného registru (14 byte), začínající adresou C1B2 (HEXA).

- LOADD (adresa 8DC2) zabezpečuje přečtení dat s možností jejich uložení do operační paměti. Vstupními argumenty jsou registr HL, určující paměťové místo, registr D, udávající počet dat, registr C, ve funkci semaforu pro možnost ukládání dat do paměti.

Pro využití magnetofonu jako vnější paměti jsou uvedené rutiny postačující. Systémový příkaz musí zabezpečit při záznamu dat naplnění registru pro záhlaví s požadovanými argumenty

a pro čtení záznamu budeme tento registr kontrolovat na číslo záznamu, typ, popřípadě i na obsah komentáře, ve kterém může být uložen název záznamu.

Pro doplnění uvádíme seznam systémových příkazů pro práci s magnetofonem pod operačním systémem a pod BASICem v příkladech.

OS: - MGSV 01 1000 1400 TESLA

Záznam dat od adresy 1000 do 1400 (HEXA) s komentářem TESLA a záznam má mít evidenční číslo 01. (znak _ představuje delimetr mezeru - SPACE).

- MGLD 01

vyhledání a uložení záznamu 01 z pásky MG do operační paměti.

- MGEND 01

kontrola správnosti záznamu s číslem 01 současně umožňuje vyhledat jeho fyzický konec.

BASIC : - SAVE 1 "TESLA

záznam programu BASIC pod evidenční číslo 1 s komentářem TESLA, úvodní uvozovky je nutné použít, jinak se v komentáři nesmí vyskytnout žádné klíčové slovo BASICu.

- LOAD 1

vyhledání a zavedení programu BASIC

- CHECK 1

vyhledání a kontrola záznamu s číslem 1, současně umožňuje vyhledání jeho fyzického konce:

- DSAVE 1; AŽ (Ø) "komentář"

zápis obsahu jednorozměrného řetězcového pole AŽ pod evidenčním číslem 1.

- DLOAD 1; BŽ (Ø)

naplnění řetězcového pole BŽ datovým záznamem s číslem 1.

Upozornění :

Při práci s datovými záznamy je nutné mít na zřeteli dimenzi daného pole v programu BASIC.

III.4. Software vstup/výstup pro styk V 24

Programové exekutivy pro vstup/výstup dat stykem V 24 jsou jednoduchého tvaru. S přihlédnutím na všeobecné zásady příkazů OUTPUT a ENTER můžeme uvést, že nasměrování výstupu znaků je pod příkazem

OUTPUT 1; výraz

který umožňuje přemodifikování příkazu PRINT tak, jak to uvádí následující procedura (adresa 2215) :

V 24 OUT: LXI H, USARTOUT

modifikace výstupu

SHLD ØØ15

ukazatel výstupní rutiny

POP H

návrat programového čítače

INX H	inkrement programového čítače
CALL 006D	kontrola syntaxi - znak
DB ';	středník
JMP 203F	skok do rutiny PRINT
USARTOUT: MOV C, A	uchování výstupního znaku
CALL 8DB1	testování připravenosti USARTu
MOV A, C	návrat výstupního znaku
VÝSTUP : OUT 1E	výstup znaku na V 24
RET	návrat do BASICu

V případě žádosti o úpravu výstupního znaku na jiný kód, je možno uložit na adresu VÝSTUP (2229 HEXA) rutiny USARTOUT (adr. 2224 HEXA) skok do vlastního podprogramu, který provede konverzi kódu a výstup znaku na USART. Připomínáme, že je nutno uchovat registr HL, protože tento obsahuje návratovou adresu programového čítače BASIC.

Použití příkazu pro výstup znaku ASCII na kanál 1 je jednoduché, jak je vidět z následujícího příkladu, který demonstruje výstup znaků z řetězce AŠ, rychlostí 1200 Bd, bez parity, osm údajových bitů a 2 STOP bity /stejně podmínky u výstupu na MG/ :

OUTPUT 1; AŠ

vstup dat k proměnné je umožněn pod příkazem

ENTER 1; seznam proměnných

který v záhlaví interpretování zmodifikuje rutinu INPUT pro přiřazení vstupní rutiny znaku stykem V 24. Opět si napíšeme

její tvar v assembleru 8080 (adresy 222C, 223B HEXA).

V 24 INP :	LXI H, USARTINP	zmodifikování ukazatele vstupní
	SHLD 085F	rutiny
	POP H	návrat programového čítače BASIC
	INX H	zvýšení programového čítače
	CALL 006D	kontrola syntaxi, znak ';
	DB ;	
	JMP 214E	pokračování v rutině INPUT
USARTINP:	DCX H	ukazatel vstupního BUFFERu
LOOP:	INX H	
	CALL 8E0E	testování připravenosti USARTu
	IN 1E	vstup znaku z USARTu
	MOV M, A	zápis znaku do vstupního bufferu
	CPI 0A	testování koncového znaku
	JNZ LOOP	skok na přiřazení proměnné k ru-
		tině INPUT

Opět i v tomto případě lze upravovat vstupní rutinu o další možnosti podle druhu periferní jednotky (převod kódu na ASCII kód, změna koncového znaku apod). V některých případech je potřeba přijímat předem stanovený počet znaků, následující příklad nám ukáže, jak se postupuje při modifikaci vstupní rutiny znaku.

Příklad č. 10 : Upravte příkaz ENTER pro vstup 128 znaků stykem
V 24.

Řešení : 1. krok - vytvoříme vstupní rutinu s počítadlem pro
128 znaků

USART 128 :	LXI D, 0080	nastavení počítadla
	DCX H	ukazatel vstupního bufferu
LOOP :	INX H	
	CALL SE0E	testování připravenosti USARTu
	IN 1E	vstup znaku v V 24
	MOV M, A	zápis znaku do bufferu
	DCXD	snížení počítadla
	MOV A, D	} počítadlo prázdné ?
	CPI FF	
	JNZ LOOP	pokud ne, pokračuj v cyklu
	INX H	zvyš ukazatel bufferu
	INX H	
	JMP 220E	skok na přiřazení k proměnné rutiny INPUT

Poznámka : Zvýšení ukazatele vstup. bufferu o 2 adresy je nutné z důvodu dalšího zpracování, které v původní vstupní rutině maskuje koncové znaky CRLF (0D 0A HEXA) a zapíše koncový znak 00.

2. krok - v rutině "V 24INP" nahradíme na adrese

Provedeme modifikaci rutina ENTER pro V 24, která zabezpečuje přiřazení vstupní rutiny znaku. Na adresu 222D uložíme vektor nově vytvořené rutiny USART 128.

Počet přijatých znaků v našem příkladu byl stanoven na konkrétní hodnotu. Je však možné vytvořit i takovou modifikaci příkazu ENTER, ve které bude moci uživatel přímo uvést počet přijímaných vstupních znaků. Pro doplnění uvádíme upravení rutiny "V 24 INP" se vstupní rutinou uvedenou v tomto příkladu.

V 24 INP :	LXI H, USART 128	modifikace ukazatele vstupní rutiny
	SHLD 085F	
	POP H	návrat programového čítače BASIC

```

INX H      zvýšení programového čítače
CALL 006D  testování znaku čárka
DB ', '
CALL 16D7  výpočet hodnoty v rozsahu 1 byte
STA (USART 128) + 2   ulož parametr počet znaků
CALL 006D  testování znaku středník
DB ', '
JMP 214E   pokračování rutiny INPUT
    
```

Syntax pro nově vytvořenou rutinu ENTER bude :

```
ENTER 1, P; AŽ
```

- kde P je parametr příkazu pro nastavení počtu vstupních znaků z kanálu V24

III. 5. Programování kanálu 1

Pro nastavení kanálu 1 se všeobecně používá příkaz

```
CONTROL 1, r; seznam výrazů
```

- parametr r = 1
- seznam výrazů obsahuje parametry, které specifikují kanál 1

První parametr, který musí být odeslán na USART je parametr určující režim provozu (MODE INSTRUCTION) a druhým je parametr určující povelovou instrukci USARTu /COMMAND/.

V případě, že je zvolen režim synchronního provozu, tak mezi tyto dva parametry se vsouvá jeden nebo dva synchronizační znaky SYNC. Na následujícím obrázku je schématicky znázorněna

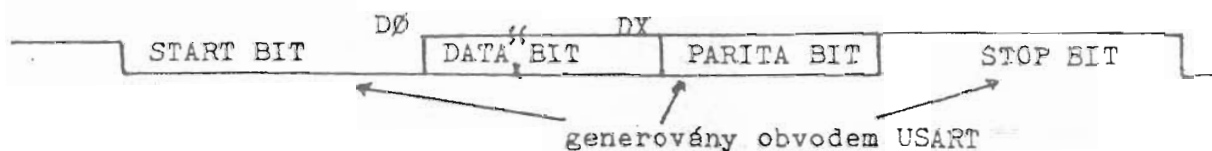
posloupnost parametrů - dat na USART.

význam r	sled vyslaných dat po RESET USARTu
r = 1	MODE INSTRUCTION
r = 1	SYNC 1
r = p	SYNC 2
r = 1	COMMAND
r = ∅	DATA
r = 1	COMMAND
r = ∅	DATA

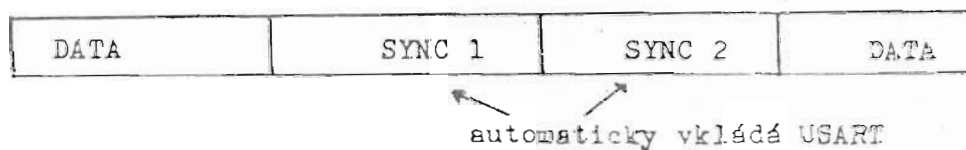
- hodnota r uvádí, zda se jedná o nastavení kanálu nebo přesun dat.

Rozdíl v přenosu dat asynchronního a synchronního režimu vystihují následující obrázky :

ASYNCHRONNÍ PŘENOS :



SYNCHRONNÍ PŘENOS :



Při asynchronním režimu každý odeslaný datový byte začíná START bitem, vlastními daty, jejichž délka může být naprogramována dalším paritním bitem a uzavírají ho jeden nebo dva STOP bity.

Při synchronním režimu je odeslán celý datový blok, který začíná jedním nebo dvěma synchronizačními znaky. Tento přenos je podstatně rychlejší než asynchronní.

Při volbě synchron znaku je třeba dbát, aby tento charakter se nevyskytoval v datovém poli.

III. 5. 1. Programování kanálu 1 pro asynchronní provoz

Vlastnosti obvodu USART v asynchronním režimu provozu jsou popsány v tabulce číslo 10. Jsou v ní uvedeny decimální hodnoty, které uživatel musí pro svůj určený provoz spočítat (hodnota "k") a odeslat na kanál 1 prostřednictvím příkazu :

CONTROL 1, 1; k

Tabulka č. 10

Podmínky provozu režimu kanálu 1		hodnota "k"
Počet STOP bitů	1	64
	1 1/2	128
	2	192
Generování parity	NE	0
	ANO	16
Parita	sudá	32
	lichá	0
Počet datových bitů	5	0
	6	4
	7	8
	8	12
Faktor BAUD (vzorkovací frekvence)	1 x	1
	16 x	2
	64 x	3

Poznámka : Při nastavení vzorkovací frekvence 1 x není možné volit 1 1/2 STOP BIT.

Příklad č. 11 : Nastavte kanál 1 do provozu asynchronní režim, počet datových bitů je 8, bez parity, počet STOP bitů 2, vzorkovací frekvence 1 x.

Řešení : Podle tabulky 10 spočteme jednotlivé hodnoty za daných podmínek a vyšleme příkazem CONTROL. Tedy :
 CONTROL 1, 1; 12 + 0 + 192 + 1
 CONTROL 1, 1; 215

III. 5. 2. Programování obvodu USART v synchronním režimu

Nastavení kanálu do synchronního režimu se uskutečňuje obdobně jako u asynchronního režimu, s tím rozdílem, že je použita jiná tabulka - viz tabulka číslo 11.

Podmínky provozu synchronního režimu		"k" hodnota
Počet datových bitů	5	0
	6	4
	7	8
	8	12
Generování parity	NE	0
	ANO	16
Parita	sudá	32
	lichá	0
Externí synchronizace	ANO	64
	NE	0
Počet SYNC charakterů	1	128
	2	0

Příklad č. 12 :

Nastavte kanál 1 do provozu synchronního, počet datových bitů - 8, sudá parita, interní synchronizace, počet SYNC = 2, hodnota SYNC značky FF (HEXA).

Řešení :

Pro určení parametru režimu provozu použijeme tabulku č.11, ze které vybereme jednotlivé hodnoty podmínek a spočítáme. Tuto hodnotu včetně hodnoty SYNC značky odešleme na kanál 1 prostřednictvím příkazu CONTROL. Tedy :

CONTROL 1, 1; 12 + 16 + 32 + 0 + 0, 255

CONTROL 1, 1; 60, 255

Poznámka : SYNC značku můžeme odeslat i samostatným příkazem. Podmínkou je, aby tato značka byla zapsána do USARTu, bezprostředně potom, když se nastaví jeho režim provozu.

Pokud už byl zvolen režim provozu, můžeme přistoupit k nastavení povelové instrukce (COMMAND). Toto nastavení platí pro oba druhy režimů (synchronní i asynchronní).

III. 5. 3. Nastavení povelové instrukce kanálu 1.

Způsob nastavení je opět stejný jako v části 5.1. nebo 5.2. Následující tabulka uvádí jednotlivé hodnoty u požadovaných signálů, které je nutno spočítat a odeslat na kanál 1 příkazem CONTROL.

Povelové signály	režim	hodnota
Povolení vysílání T x EN	ano	1
	ne	0
Povolení příjmu R x EN	ano	4
	ne	0
$\overline{DTR} = 0$ Datový terminál připraven	ano	2
	ne	0
Vnitřní nulování INTERNAL RESET	ano	64
	ne	0
Výzva k vysílání $\overline{RTS} = 0$	ano	32
	ne	0
Nulování příkazových chyb (PE, OE, FE)	ano	16
	ne	0
Start vyhledávání SYNC (HUNT, MODE)	ano	128
	ne	0
Nastavení T x D do stavu LOG 0	ano	8
	ne	0

Nastavení povelové instrukce kanálu 1 můžeme provádět v libovolném okamžiku, tedy i mezi přenosem dat. Pokud si uživatel blíže všimne jednotlivých signálů, zjistí, že některé se nacházejí přímo jako výstupní linky obvodu USART.

Příklad č. 13 :

Nastavte kanál č. 1 do stavu povolení vysílání i příjmu, datový terminál připraven.

Řešení :

S přihlédnutím k hodnotám uvedeným v tabulce č. 12 můžeme psát :

```
[ CONTROL 1, 1; 1 + 4 + 2
  CONTROL 1, 1; 7
```

Závěrem lze konstatovat, že naprogramování kanálu č. 1 zůstává vysláním několika parametrů v určitém pořadí. Toto pořadí stanoví jejich význam.

Příklad č. 14 :

Nastavte kanál 1 do provozu uvedeného v příkladu č. 11 a 13,

Řešení : CONTROL 1, 1; 215, 7

Je-li žádoucí změnit provozní režim kanálu 1, je nutno nejprve odeslat povelové instrukce pro interní nulování a až potom provést vlastní nastavení režimu. V opačném případě řídicí logika USARTu považuje každý údaj jako povelovou instrukci. Tuto podmínku je nutno akceptovat okamžitě, pokud chceme využívat sériový kanál jinak než jej využívá interface magnetofonu.

Příkaz pro interní nulování kanálu 1 je :

CONTROL 1, 1; 64

III. 6. Status kanálu č. 1

Obvod 8251 nám umožňuje přečíst stavové slovo, které obsahuje několik důležitých vstupních linek interface i stav provozu zvoleného režimu.

D7	D6	D5	D4	D3	D2	D1	D0
DSR	SYNDET	CE	FE	PE	TxE	RxRDY	TxRDY

Význam jednotlivých signálů u jednotlivých dat je :

TxRDY : výstup pro indikaci stavu vysílací vyrovnávací paměti.

Je nulován zapsáním do vysílací vyrovnávací paměti (pokud $TxRDY = 1$ znamená, že je možno přijímat další údaj na vyslání).

RxRDY : výstup pro indikaci stavu výstupní paměti přijímače.

(pokud $RxRDY = 1$ znamená to, že je možno přečíst údaj z kanálu).

TxE : výstup pro indikaci stavu vysílací paměti. Pokud je TxE rovno 1 znamená to, že vysílání dat bylo ukončeno.

DSR : signál označující připravenost přenosového zařízení /programově zabezpená úroveň LOG 0/.

SYNDET : Detekce synchronizace (pouze pro synchronní režim).

Pokud byl nalezen znak SYNC, pak $SYNDET = 1$. Jeho nulování se uskuteční přečtením stavového slova.

PE : parita přijatého znaku nesouhlasí s paritou nastavenou instrukcí pro provozní režim.

FE : ukončení přijatého znaku nesouhlasí s nastavenou délkou STOP bitu (platí pouze pro asynchronní režim).

OE : obsah příjmové paměti obvodu 8251, který nebyl přečten, je nahrazen právě přečteným novým údajem.

Přiřazení stavového slova proměnné je umožněno příkazem -
- funkcí :

$$S = STATUS\ 1, 1$$

Po jeho vykonání nabude proměnná "S" hodnoty stavového

slova v rozsahu 0 + 255. Další zpracování může zjistit stav jednotlivých linek, popřípadě chybu při přenosu.

Závěrem popisu kanálu 1 uvedíme příklad připojení a modifikace výstupní rutiny znaku pro dálnopis T 100, který je využíván ve funkci tiskárny.

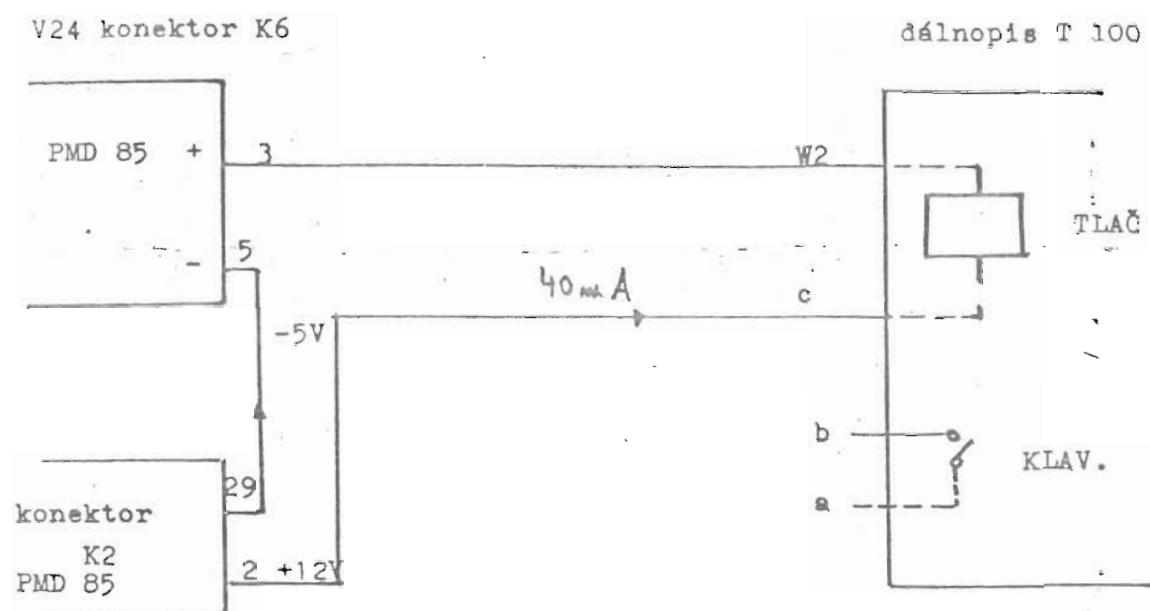
Příklad č. 15

Proveďte systémový návrh připojení čsl. dálnopisu T 100 ve funkci tiskárny sériovým stykem V 24 - kanál 1.

Řešení se bude skládat z následujících kroků :

1. krok - přepneme přepínač na volbu interface pro V 24
2. krok - podle připojovacího protokolu - viz obr. č. 9 - připojíme výstup V 24 na vstup dálnopisu T 100. Tato výstupní linka má charakter proudové smyčky (40 mA) a dálnopis představuje relovou zátěž. Proudový zdroj získáme z napětí - 5 V a + 12 V, která jsou vyvedena na aplikačním konektoru mikropočítače PMD 85.
3. krok - naprogramujeme kanál 1 pro požadavky dalekopisného kódu CCITT 2. Jsou to tyto podmínky :
 - asynchronní provozní režim
 - počet datových bitů 5
 - 2 STOP bity
 - přenosová rychlost Bd 75
 - bez parity

Poznámka : Požadovanou přenosovou rychlost 75 Bd získáme z vestavěného generátoru vzorkovací frekvence 1 200 Hz, který slouží pro interface magnetofonu. V tomto případě



Obr. č. 9 : Připojení dálnopisu T 100 k PMD 85'

však musíme zvýšit frekvenci vzorkování na 16 x.

Na základě těchto podmínek vybereme pro dané podmínky z tab. č. 14 hodnoty režimu pro příkaz CONTROL.

Potom :

CONTROL 1, 1; 64, 194, 35
 ↑
 interní nulování kanálu
 ↑
 požadovaný režim (MODE)
 ↑
 povelová instrukce (COMMAND)

Od tohoto okamžiku je kanál 1 připraven vysílat přes V 24 výstupní znaky, avšak pouze podmnožinu (5 bitů) znaků ASCII. Pro správné vysílání znaků, který bude v souladu s normou CCITT 2, na kterou je stavěn dálnopis T 100, je nutno modifikovat výstupní rutinu znaku. Tuto činnost uskutečnime v dalším kroku.

4. krok - dálnopisný kód CCITT 2 se skládá z 5 bitů. Jejich pomocí je obsaženo 49 znaků ASCII a 5 znaků řídících. Přepínání (shift) znaků je vysláním řídícího kódu pro danou změnu. V následující tabulce je uveden kompletní kód podle CCITT 2 pro písmena a číslice ve tvaru (HEXA).

HEXA KÓD	PÍSMENO	ČÍSLO
03	A	-
19	B	2
0E	C	:
09	D	
01	E	3
0D	F	
1A	G	

14	H	
06	I	8
0B	J	Ω
0F	K	(
12	L)
1C	M	.
0C	N	,
18	O	9
16	P	Ø
17	Q	1
0A	R	4
05	S	'
10	T	5
07	U	7
1E	V	=
13	W	2
1D	X	/
15	Y	6
11	Z	+
08	CR	CR
02	LF	LF
1B	SHIFT NAHOŘE	SHIFT NAHOŘE
1F	SHIFT DOLE	SHIFT DOLE
04	SPACE	SPACE
00	nepoužito	nepoužito

Vysvětlivky :

CR - návrat vozu
 LF - posun o řádek

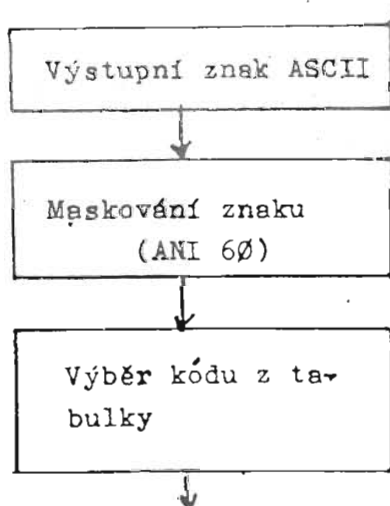
SHIFT NAHOŘE písmenná mezera
SHIFT DOLE číselná mezera
SPACE - mezera

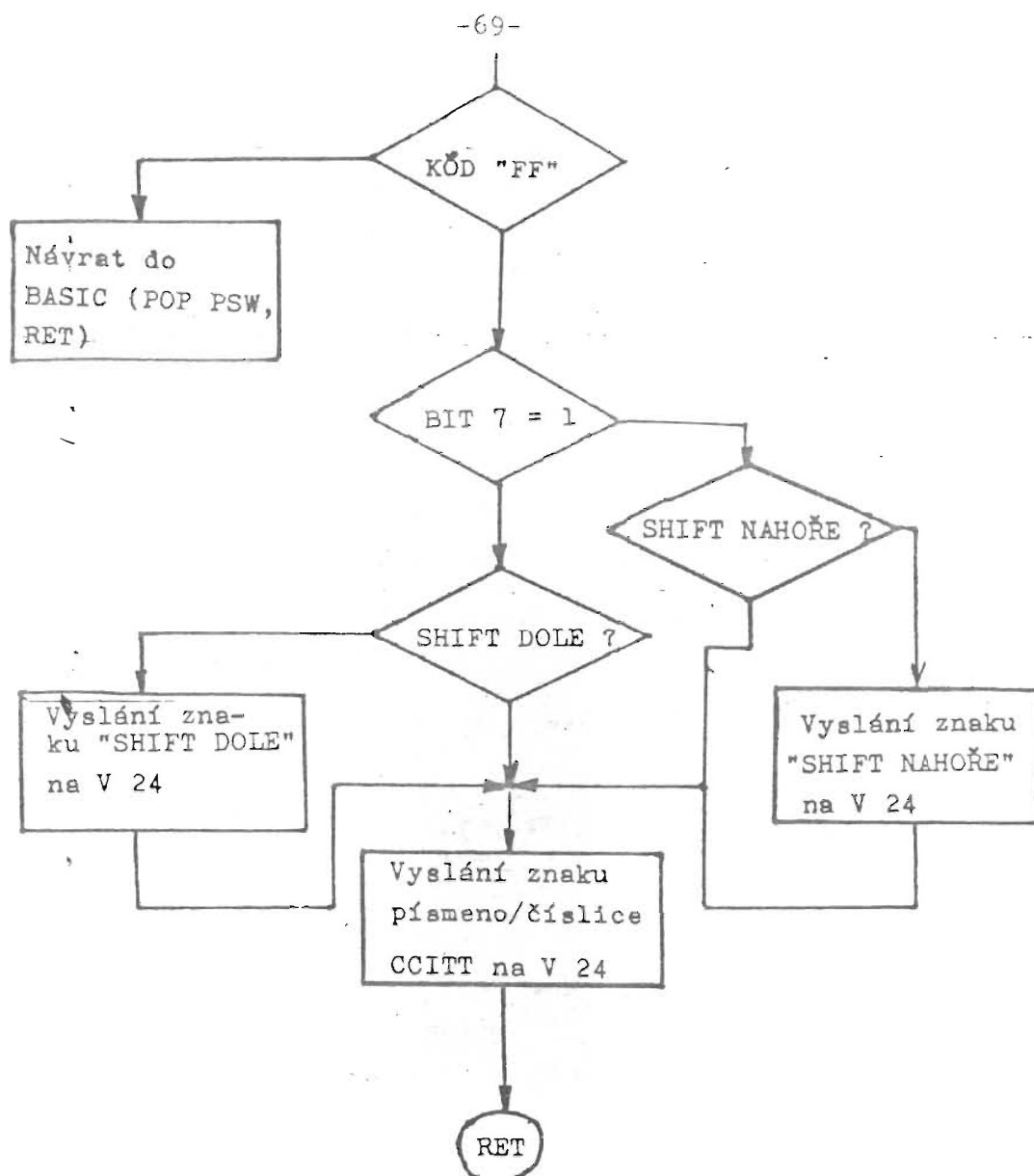
Pro znaky ASCII v mřížce HEXA od 20 do 60 (SPACE až /) je nutno vytvořit tabulku, která bude obsahovat 5 bitový kód CCITT 2 příslušných znaků. Kromě těchto kódů je nutno vložit tzv. semafor určující zda se jedná o písmeno nebo o číslici. Tento význam přiřadíme poslednímu (sedmému) bitu. V případě, že nemíníme přiřadit některému kódu ASCII kód dálnopisný, neboť není, vyhradíme si kód FF.

Takže obsah tabulky může mít tvar byte :

0	0	0	X	X	X	X	X	dálnopisný kód pro písmeno
1	0	0	X	X	X	X	X	dálnopisný kód pro číslici
1	1	1	1	1	1	1	1	prázdný kód

Samotný převod ASCII/CCITT 2 provádíme principem "LOOK UP TABLE" tj. převodem přes tabulku, která bude mít, jak bylo uvedeno, pro danou adresu (kód ASCII) přiřazenu hodnotu kódu CCITT 2. Konverzní rutinu můžeme vytvořit podle vývojového diagramu obr. 8. 10.





Obr. č. 10 - Vývojový diagram pro stavbu převodní rutiny znaku ASCII/CCITT 2.

Vyslání znak ASCII vymaskujeme pro rozsah znaků SPACE až znak / tj. 20 - 60 HEXA. Tato hodnota představuje offset pro převodní tabulku. Z ní obdržíme byte, který podrobíme tes-

testování, zda je prázdný, číselný nebo písmenný znak. V případě číselného nebo písmenného znaku testujeme, jaký znak byl předcházející, aby nebylo třeba pokaždé vysílat kód SHIFT, před vlastním kódem písmeno/číslice. Při vytváření této převodní rutiny je nutno uchovat hodnotu registru HL.

Pro lepší manipulaci doporučujeme uložit převodní tabulku CCITT 2 s výstupní rutinou znaku CCITT 2 do uživatelské ROM-ky, která při aktivování (příkaz ROM) provede uložení tabulky a výstupní rutiny na interpretační místo v operační paměti a nakonec uskuteční přemodifikování původní procedury OUTPUT 1. (viz popis uživatelské příručky II, příkaz ROM).

Modifikace výstupní rutiny se uskutečňuje na adrese 2216 (HEXA), kde je ukazatel výstupní rutiny znaku. Na toto místo uložíme vektor uživatelské výstupní rutiny "CCITT 2".

Tvar výstupní rutiny znaku CCITT 2 :

CCITT 2 : MOV C, A	uchování znaku ASCII
CALL 8DB1	testování připravenosti USARTu
MOV A, C	návrat znaku ASCII
CALL PREVOD	převod znaku případně vyslání kódu SHIFT
OUT 1E	výstup znaku
RET	návrat do BASICu

Poznámka : Převodní rutina zabezpečuje kromě vlastního převodu kódu, i vyslání změny SHIFT pro písmeno/číslici.

5. krok - systémový přístup k dálnopisu je umožněn příkazem

OUTPUT 1; seznam výrazů
nebo LIST # 1; -
pro výpis programu BASIC.

IV. Kanál 5 - časovač/čítač

Tento kanál umožňuje uživateli zpracovávání časových znaků, reálného času, generování časových signálů, tedy dává podporu pro software, který požaduje zpracovávání signálů.

IV. 1. HARDWARE

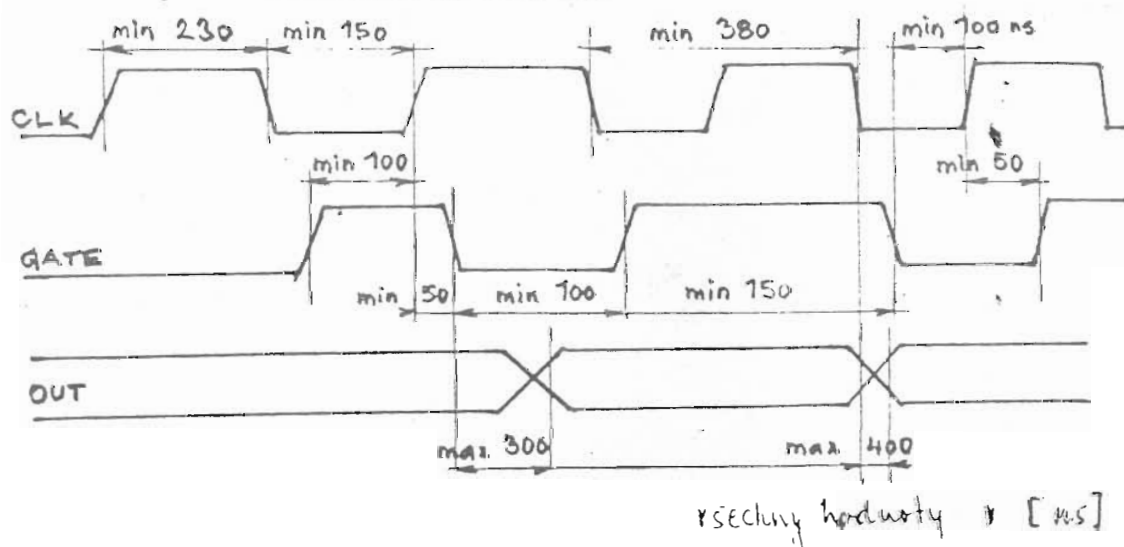
Základním stavebním prvkem kanálu 5 je programovatelný čítač/časovač realizovaný obvodem 8253. Obvod 8253 představuje tři samostatně pracující 16-ti bitové programovatelné čítače vzad se svým 16-ti bitovým registrem počáteční hodnoty a vlastní hradlovací logiky. Úlohou hradlovací logiky je hradlovat vstup hodin čítače, nastavovat počáteční hodnotu čítače a generovat požadovaný výstup v závislosti na zvoleném režimu časovače. Obvod 8253 nevyužívá systémový signál RESET, takže uživatel musí podle charakteru aplikace zjistit pro příslušný časovač danou inicializaci. Uživatel má přístup k časovačům ze dvou stran :

- programový přístup s podporou příkazů BASIC
- hardwareový přístup prostřednictvím tří linek
 - a) vstupní linka "CIK" představuje základní vstupní signál, který umožňuje dekrementovat hodnotu programovatelného 16-ti bitového čítače vzad o jednotkový stav. Maximální frekvence

signálu CLK je 2 MHz.

- b) vstupní linka "GATE" umožňuje blokovat dekrementování čítače od přivedených hodin CLK. Tento signál úrovně LOG.1 uvolňuje činnost čítače.
- c) výstupní signál "OUT" signalizuje dosažený stav čítače/časovače a jeho časový průběh závisí na naprogramovaném režimu.

Základní časové průběhy signálů CLK, GATE a OUT jsou znázorněny na následujícím obrázku.



Zapojení obvodu 8253 v architektuře interface počítače PM8 85 je znázorněno na obr. č. 11.

Jednotlivé čítače (T0, T1 a T2) - jejich I/O linky jsou přístupné uživateli následovně :

Čítač T0 : je k dispozici úplně uživateli prostřednictvím aplikačního konektoru K2.

Signál GATE je inicializován na úroveň LOG.1.

čítač T1 : Hodinový signál tohoto čítače je pevně připojen na systémové hodiny Ø2 (TTL). Linky GATE, OUT jsou k dispozici uživateli na konektoru K2. Navíc výstupní linka OUT je připravena pro možnost využití vzorkovací frekvence u kanálu 1 (obvod 8251). Provedení této úpravy si může udelat na základě obr. č. 8.

čítač T2 : Tento čítač je využit pouze pro programový přístup. Je předurčen pro programové zpracovávání reálného času. Sekundové pulzy jsou generovány hodinovým obvodem MHB 1116 a přivedeny na vstup CLK tohoto čítače.

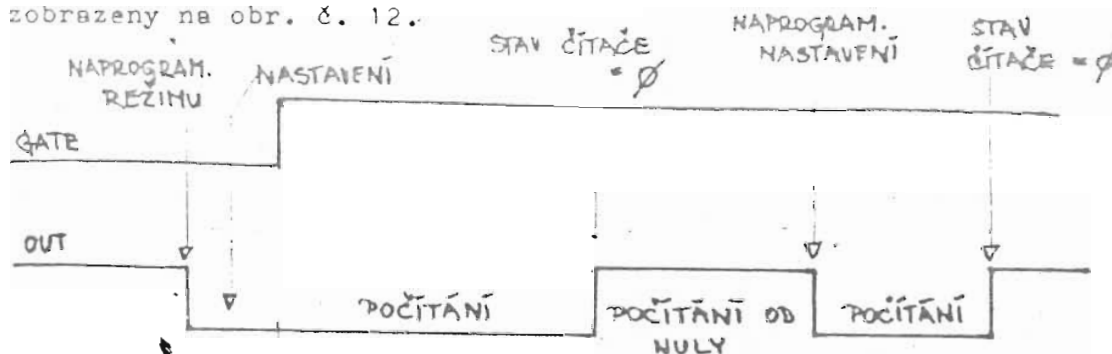
Obvod 8253 poskytuje celkově 6 režimů práce, z nichž každý může být použit pro kterýkoli časovač. Jednotlivé režimy se v zásadě odlišují v reakci na hradlovací vstup a chování ve výstupu. V následujících částech kapitoly se probereme podrobně každý pracovní režim časovače.

IV. 1. 1. Režim Ø - generátor přerušení s programovým spouštěním

Čítač odpočítává hodinové impulzy od počáteční hodnoty směrem k nule po programovém nastavení a uvolnění signálu GATE. Čítač pokračuje v odpočítávání i po překročení nuly. Pokud se během počítání časovače nastaví nová počáteční hodnota, počítání se zastaví, výstup OUT přejde do nuly a začne znovu počítat hned po vložení prvního byte.

Režim Ø je vhodný pro generování jednotlivých časových zpoždění nebo časová měření, která jsou inicializována programovými nebo technickými událostmi za předpokladu, že měřený čas je

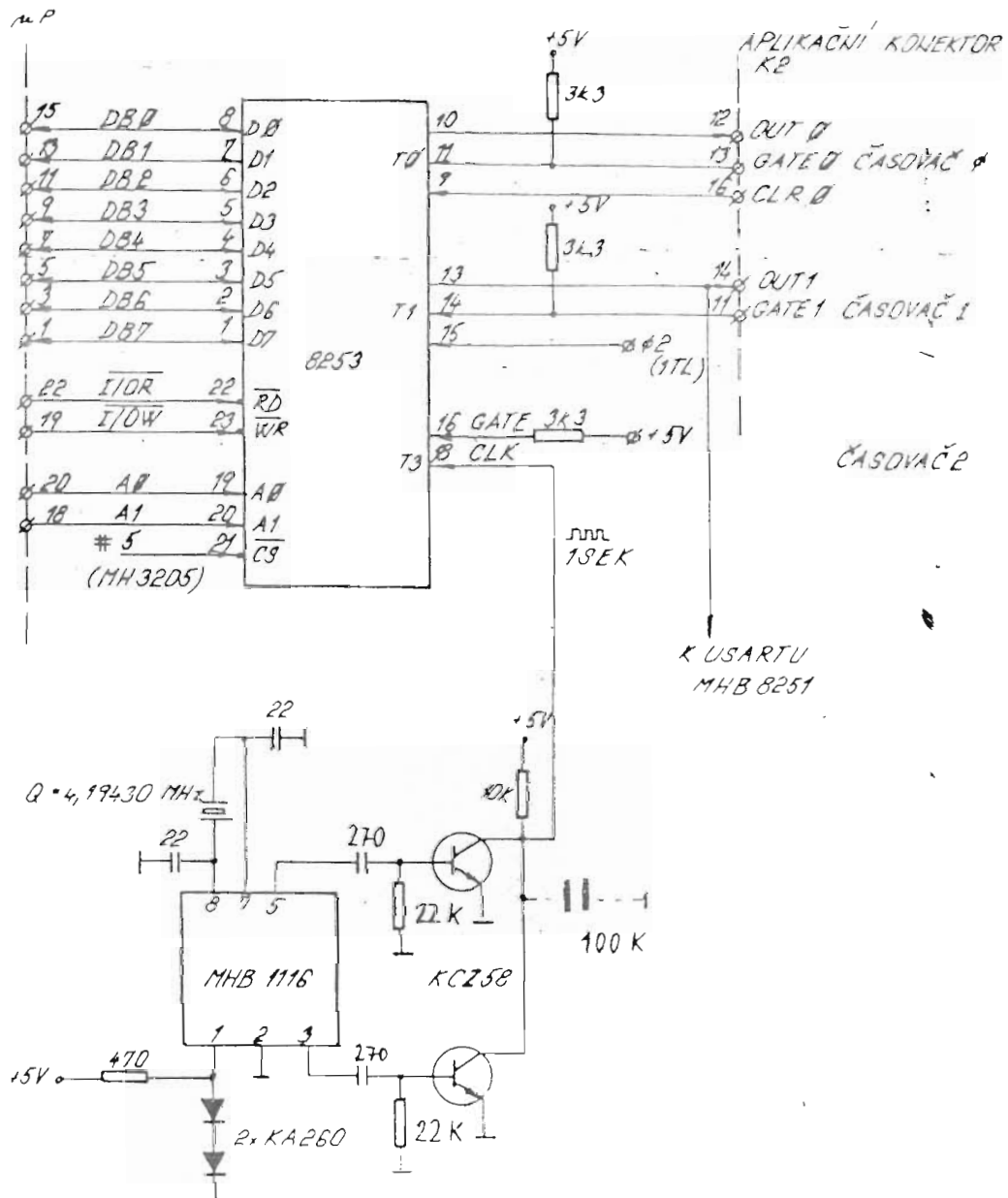
menší než čas odpovídající 16-ti bitové kapacitě čítače. Použití režimu na měření délky externího signálu umožňuje hradlovací vstup, který povoluje počítání pouze během jeho jedničkové úrovně. Časové průběhy signálů v režimu \emptyset jsou zobrazeny na obr. č. 12.

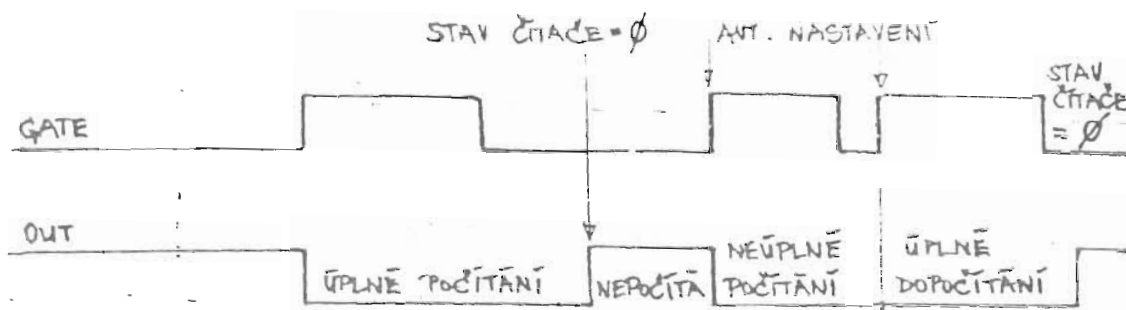


Obr. č. 12 - Časové průběhy signálů v režimu \emptyset

IV. 1. 2. Režim 1 - generátor přerušení s hardwareovým spouštěním

Odpočítávání hodinových impulzů od počáteční hodnoty směrem k nule inicializuje každá náležitá hrana hradlovacího signálu GATE, když se zároveň nuluje výstup OUT. Po odpočtu přechází výstup OUT do úrovně LOG 1 a nové počítání odstartuje další náběžná hrana hradlovacího signálu. Použití režimu 1 je vhodné při generování časového zpoždění nebo měření času, které vyvolává externí událost, zvláště tehdy, pokud je to krátký impuls. Časový průběh signálů je na obr. č. 13.

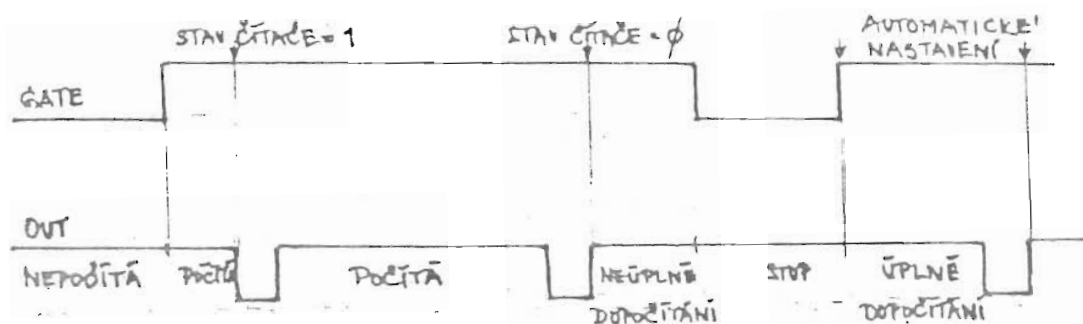




obr. 8. 13 - časové průběhy signálů v režimu 1

IV. 1. 3. Režim 2 - generátor impulzů odčítání s automatickým spouštěním

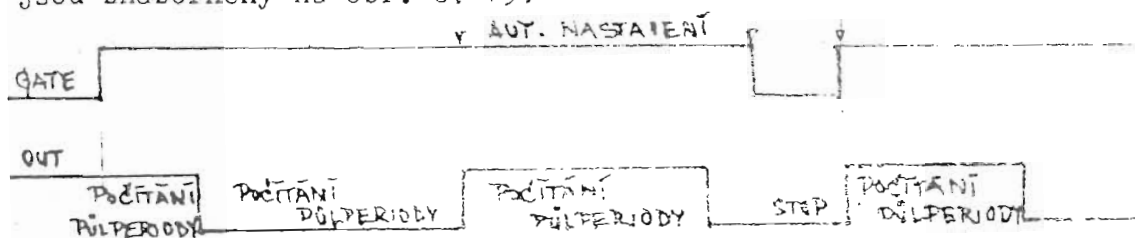
PO uložení počáteční hodnoty začne čítač opakovaně odpočítávat vložený počítaný interval a generovat impulzy, trvající právě jeden hodinový interval, během jednotkového stavu čítače. Po dekrementaci čítače na nulu se počáteční hodnota automaticky nastaví ze stavu nastaveného 16-ti bitového registru počáteční hodnoty : Režim 2 je obzvlášť vhodný pro generování časových funkcí, pro které nepostačuje 16-ti bitová kapacita čítačů. Jestliže se během počítání vloží do registru počáteční hodnoty nová počáteční hodnota, neovlivní to právě probíhající cyklus se starou hodnotou, ale po něm následující cykly budou už počítat s novou hodnotou. Nulová úroveň signálu GATE zastavuje počítání čítače od počáteční hodnoty. Na obr. 8. 14 jsou zobrazeny pro tento režim časové průběhy signálů.



Obr. č. 14 - Časové průběhy signálů v režimu 2

IV. 1. 4. Režim 3 - generátor obdélníkového signálu

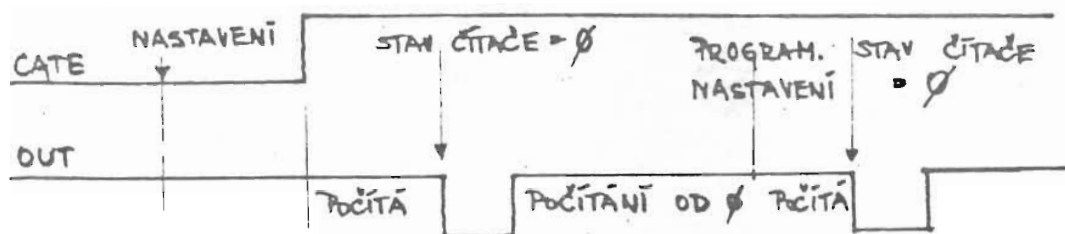
Po uložení počáteční hodnoty začne čítač opakovaně odpocítávat vložený počítací interval a generovat obdélníkový signál s periodou 1 : 1. Výstup se na začátku každého počítacího intervalu nastavuje na úroveň LOG 1 a v půlce počítacího cyklu přechází do úrovně LOG 0. V případě, že počáteční hodnota je lichým číslem, jednotkový výstup trvá o 1 hodinový interval déle. Pokud se během počítání vloží do registru počáteční hodnoty nová počáteční hodnota, nevlivní to právě probíhající půlcyklus se starou hodnotou, ale ze ním následující půlcyklus a další cykly budou už s novou hodnotou. Hradlovací vstup GATE zastavuje počítání, a při jeho náběžné hraně se čítač rozběhne od počáteční hodnoty. Časové průběhy signálů jsou znázorněny na obr. č. 15.



Obr. č. 15 - Časové průběhy signálů GATE, OUT v režimu 3

IV. 1. 5. Režim 4 - generování impulzu odpočítání s programovým spuštěním

Čítač odpočítává hodinové impulzy od počáteční hodnoty směrem k nule po programovém nastavení a hardwareovém odhradlování. Při nulovém stavu čítače výstup OUT přechází na 1 hodinový interval do 0. Čítač pokračuje v počítání, ale výstup zůstává na úrovni LOG 1 až do dalšího nastavení a dopočítání se čítače. Pokud během počítání vloží do registru počáteční hodnoty nová počáteční hodnota, neovlivní to právě probíhající cyklus, ale hned po dosažení nuly začne cyklus s novou počáteční hodnotou. Proto se režim 4 zvlášť hodí na generování posloupnosti zpoždění, jejichž délka se liší. Počítání se zastavuje při nulové hodnotě signálu GATE. Časové průběhy signálů GATE a OUT jsou znázorněny na obr. č. 16.

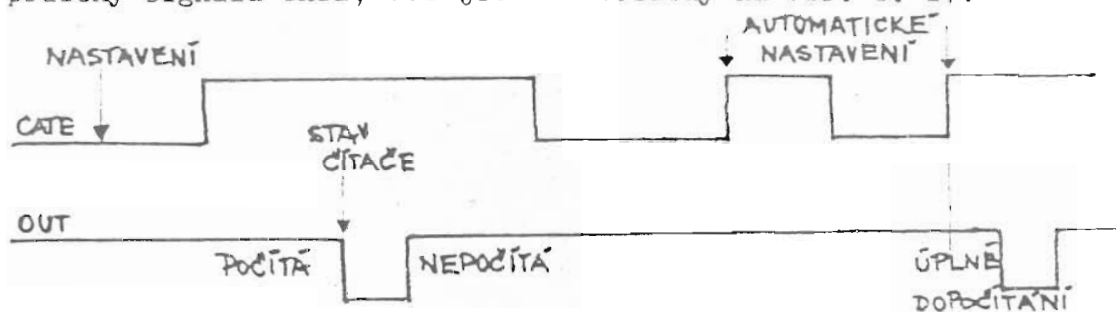


Obr. č. 16 - Časové průběhy signálů režimu 4

IV. 1. 6. Režim 5 - generátor dopočítání s hardwareovým spouštěním

Odpočítávání hodinových impulzů od počáteční hodnoty směrem k nule inicializuje každá náběžná hrana hradlovacího signálu GATE. Výstup OUT nabývá hodnoty LOG 1 po naprogramování režimu, úroveň LOG 0 při nulovém stavu čítače.

Délka trvání úrovně LOG 0 je jeden hodinový interval. Pokud se během počítání vloží nová počáteční hodnota, neovlivní to právě probíhající cyklus. Nový cyklus s novou počáteční hodnotou začne vždy při příchodu náběžné hrany signálu GATE. Časové průběhy signálů GATE, OUT jsou zobrazeny na obr. č. 17.



Obr. č. 17 - Časové průběhy signálů v režimu 5

IV. 2. Software

Pro kanál 5 - čítač/časovač není programová počítací (OUTPUT/ENTER) v jazyce BASIC tak, jak je tomu u ostatních kanálů. Tuto "nevýhodu" je možné kompenzovat vytvořením vlastního

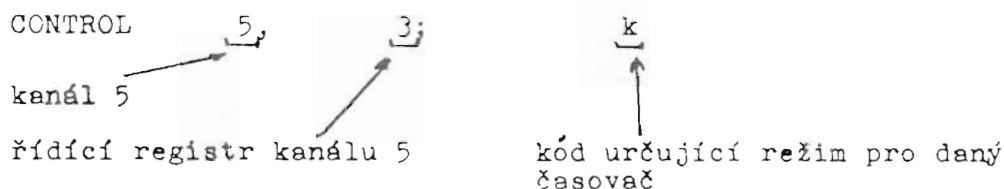
uživatelských exekutiv pro konkrétní využití časovače a adresní vektory umístít do tabulky skoků. Pro vstup je to ukazatel 2092 (HEXA) a pro vstup ukazatel na adrese 215B (HEXA).

Určitým průvodcem při tvorbě těchto exekutiv mohou být rutiny uvedené v kapitole II - kanál 4. Pro programový styk s kanálem můžeme využívat příkazy a funkce :

CONTROL, OUT, STATUS, INP

IV. 3. Programování kanálu 5

V části I této kapitoly byly podrobně popsány pracovní režimy čítače/časovače 8253, nastavení obvodu, tedy kanálu do zvoleného režimu vykonáváme příkazem



Parametr "k" určíme z tabulky č. 13, ve které je uvedena konkrétní hodnota pro daný časovač a režim.

V dané tabulce se vyskytují 4 funkce pro daný režim časovače.

Jejich význam je následující :

- VZORKUJ tato funkce nám umožní prostřednictvím řídicího registru daného časovače provést jeho vzorkování během čtení jeho stavu, které čtení čítače v okamžiku jeho měnících se výstupů.

- ČTI/NASTAV - je funkce, kterou chceme vykonat nad zvoleným časovačem v daném zvoleném režimu. Stav "ČTI" se uplatňuje při přebírání hodnoty z časovače a stav "NASTAV" je využit pro nastavení počáteční hodnoty časovače.

Datový přístup k časovači může být 16-ti bitový nebo 8-mi bitový, a to konkrétně k dolnímu nebo hornímu byte hodnoty. Při 16-ti bitovém přístupu použijeme duplicitní přístup k údajům (nejprve dolní byte a potom horní - vyšší byte). Všechny hodnoty platí pro binární počítání. Pro desítkové počítání je nutné k dané hodnotě připočíst ještě hodnotu 1.

Časovač	Funkce	Režim / kód "k"					
		0	1	2	3	4	5
0	VZORKUJ	0	0	0	0	0	0
	ČTI/NASTAV JEN DOLNÍ BYTE	16	18	20	22	24	26
	ČTI/NASTAV POUZE HORNÍ BYTE	32	34	36	38	40	42
	ČTI/NASTAV OBA BYTE (DOLNÍ JAKO PRVNÍ)	48	50	52	54	56	58
1	VZORKUJ	64	64	64	64	64	64
	ČTI/NASTAV POUZE DOLNÍ BYTE	80	82	84	86	88	90
	ČTI/NASTAV POUZE HORNÍ BYTE	96	98	100	102	104	106
	ČTI/NASTAV OBA BYTE (DOLNÍ JAKO PRVNÍ)	112	114	116	118	120	122
2	VZORKUJ	128	128	128	128	128	128

2	ČTI/NASTAV POUZE DOLNÍ BYTE	144	146	148	150	152	154
	ČTI/NASTAV POUZE HORNÍ BYTE	160	162	164	166	168	170
	ČTI/NASTAV OBA BYTE (DOLNÍ JAKO PRVNÍ)	176	178	180	182	184	186

Tabulka č. 13 - Řídící kód pro nastavení kanálu 5 do požadovaného režimu,

Příklad č. 16. Nastavte časovač 0 v kanálu 5 pro režim 2, s dvoubitovým čtením/nastavováním registru počáteční hodnoty a binárním počítáním. Počáteční hodnota časovače má být 256.

Řešení : Pro dané zadání vyhledáme hodnotu "k" = 52 v tab. č. 13 .
Potom vydáme dva příkazy :

CONTROL 5, 3; 52 příkaz pro zvolený režim
CONTROL 5, 0; 0, 1 příkaz pro nastavení počáteční
kanál 5 počáteční hodnoty 256 (DEC),
časovač 0 0100 (HEXA)
dolní byte = 00
horní byte = 01

Poznámka : Pro desítkové počítání by hodnota "k" byla zvýšena o 1, tj. $52 + 1 = 53$

Pro zjištění hodnoty stavu čítače/časovače poskytuje nám BASIC funkci STATUS, kterou přiřazujeme hodnotu čítače dané proměnné.

Všeobecně platí :

S = STATUS 5, T

kde "T" je parametr, který adresuje daný časovač a může mít hodnoty 0, 1, 2. (časovač 0, 1, 2).

"S" je proměnná, které se má hodnota časovače přiřadit.

Příklad č. 17 : Demonstrace příkazů pro čtení počítačového časovače 0, který byl nastaven v př. č. 16.

Řešení : Pro daný časovač vyšleme příkaz pro vzorkování časovače.

OUTPUT 5, 3; 128

dále přečteme nižší hodnotu časovače:

L = STATUS 5, 0

a potom vyšší hodnotu :

H = STATUS 5, 0

Příklad č. 18 : Nastavte časovač 1 jako generátor obdélníkových pulzů s frekvencí 1200 .

Řešení : Dané úloze odpovídá režim 3, pro který vyhledáme hodnotu "k" z tabulky č. 13.

Potom nastavení časovače provedeme příkazem :

CONTROL 5, 3; 118

časovač 1 nastavíme do počáteční hodnoty, která se vypočítá z následujícího :

- základní kmitočet CLK = 02 (TTL) = 18, 432/9 = 2.048 (MHz)
- počáteční hodnota = $2.048 \cdot 10^6 / 1200 \approx 1707$
- hodnotu 1707 převedeme na HEXA hodnotu t.j. 06AB (HEXA)
- hodnotu y 06AB (HEXA) převedeme na dvě decimální hodnoty (nižší, vyšší byte) t.j. 06; 17 (decimální)

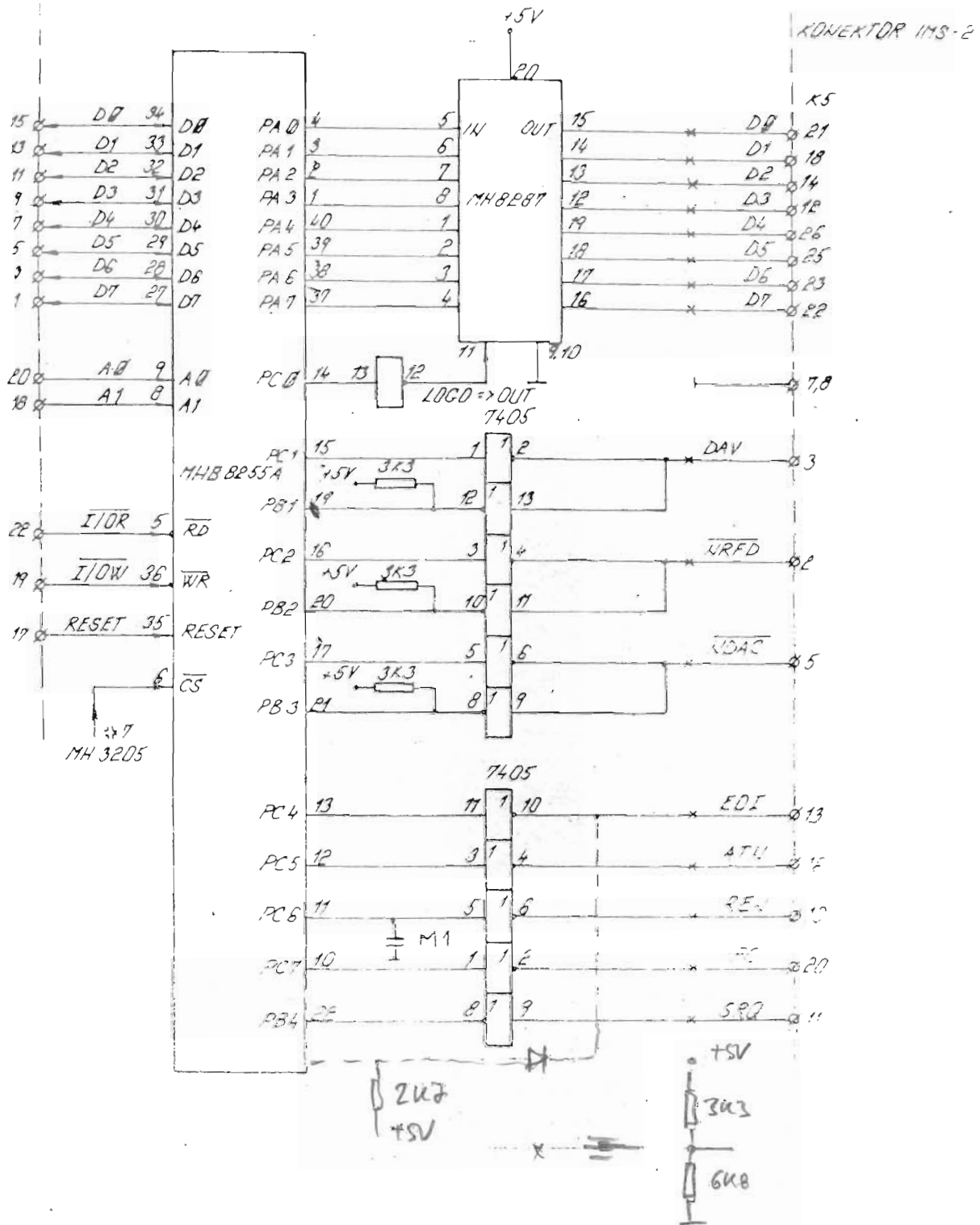
Tyto dvě hodnoty vyšleme příkazem CONTROL na časovač 1.

CONTROL 5, 1; 171, 06

Závěrem této kapitoly uvádíme, že použití časovače je velmi všestranné. Můžeme jej použít pro vyvolání přerušení od dosažení časového limitu, nebo k počítání externích pulzů. Umožňuje to propojení výstupu časovače s linkou INT, která je vyvedena na aplikačním konektoru K2. Příkaz BASIC pro nastavení časovače může uživatel vytvořit na základě popisu uživatelské příručky č. V, ve které je uveden postup modifikace interpretu.

V. Kanál 7 - styk IMS - 2 (HPIB)

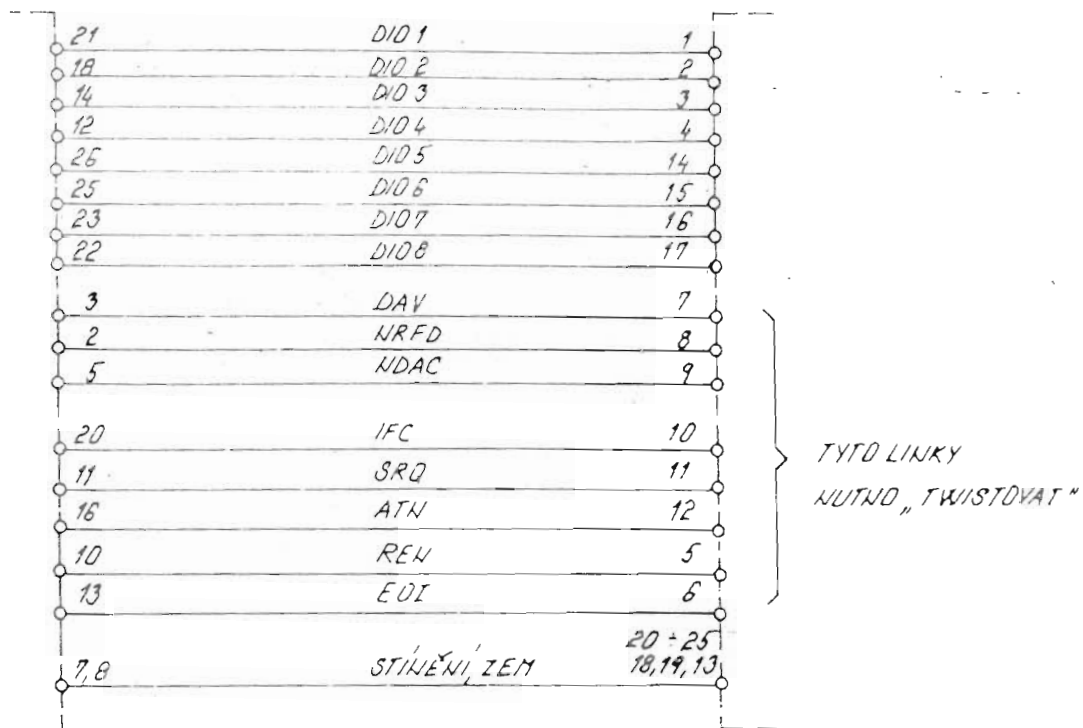
Kanál 7 představuje samostatný interface, určený pro připojení měřicích přístrojů, které jsou vybaveny standardní sběrnici IMS-2, popřípadě připojení periferních a měřicích přístrojů fy TEKTRONIX, HEWLETT PACKARD. Vůči ostatním kanálům je inicializován a obsahuje komunikační procedury řídicího členu (CONTROLLERu). Obvodové řešení interface IMS-2 je uvedeno na obr. č. 18, ze kterého je vidět, že celá činnost bude podporována programovými prostředky. Při jeho návrhu se vycházelo z minimálního universálního řešení. Styk s kanálem je prostřednictvím 38 pinového konektoru FRB (není standardní podle IMS-2), a jeho rozložení a přiřazení konektoru CANNON (standard IMS-2) je na obr. č. 19. Pro lepší pochopení průběhu vzájemné komunikace po kanálu IMS-2 popíšeme v následující části základní informace styku IMS-2, tak jak je doporučuje příslušná norma IEC. V části 2. se potom zmíníme o odlišnostech, které platí pro PMD 85.



OBROČ 19. 04. 1984 ZAPOJENI INTERFEJSU IMS-2, HP-IB,

PMD 85, KONEKTOR K5

CANNON (IMS-2)



OBR. Č. 19 PŘÍPOJENÍ POČÍTAČE PMD-85 NA STANDARDNÍ SBĚRNICI IMS-2

V. 1. Základní informace o systému IMS - 2

IMS - 2 je v podstatě stykový (přepojovací) systém, který slouží pro pružnou a ekonomickou výstavbu různých automatizovaných měřicích (zkušebních) pracovišť (sestav, kompletů), a to na základě mezinárodního normovaného propojení číslicově programovatelných (ovladatelných) přístrojů vybavených a doplněných příslušnými stykovými jednotkami.

IMS - 2 je v podstatě tvořen :

- a) normovaným kabelem tzv. sběrnicí
- b) přístrojovými styky, t.j. doplňkovými přístrojovými jednotkami odpovídajícími normě IMS - 2
- c) normami a doporučeními, které zajišťují pohodlné propojení, spolupráci a slučitelnost přístrojů různého původu.

IMS - 2 je původně určen pro propojování max. 15 přístrojů s maximální délkou sběrnice 20 m na jednom pracovišti.

Doplňkovým zařízením je však možno toto omezení eliminovat.

Zprávy mezi přístroji se přenášejí po sběrnici v číselném tvaru bit paralelně, byte (8 bitů) sériově. Pro kódování stykových zpráv je normován kód 1807 (ASCII). Maximální rychlost asynchronního přenosu je 250 až 500 k byte/s. Rychlost přenosu je automaticky přizpůsobena nejpomalejší jednotce, která se podílí na příjmu resp. vysílání zpráv.

Sběrnice má celkem 25 (případně) 24 vodičů, z toho je 16 tzv. signálních, ostatní jsou zpětné a zemnicí vodiče. Oboustranné konektory sběrnice jsou vyřešeny tak, že se dají jednoduše napojovat na sebe a na přístrojové styky a tak vytvářet

různé konfigurace, protože všechna zařízení jsou připojena na sběrnici paralelně.

Pro organizaci spojení mezi přístroji se používají rozlišovací adresy. Je k dispozici 31 různých znaků pro aktivování vysílání znaků na sběrnici a dalších 31 různých znaků pro aktivování přijímání znaků ze sběrnice. Při použití dvouznakových adres je k dispozici 961 adres pro aktivování vysílání a dalších 961 adres pro aktivování příjmu.

Protože jsou všechny přístroje připojeny ke sběrnici paralelně, byly pro uspořádání a jednoznačný tok zpráv mezi přístroji zavedeny funkce "přijímač" (listener), "vysílač" (talker) a "řízení" (controller). Tyto funkce mohou být zabudovány v přístrojových stycích a potom výběrově aktivovány pomocí tzv. adresování.

Přijímač má schopnost přijímat přístrojové zprávy ze sběrnice. Tato funkce je potřebná např. u tiskárny, děrovače děrné pásky. U minipočítače pro příjem naměřených hodnot do paměti, u měřicích přístrojů pro příjem programových kódů pro nastavení režimu práce apod. Více aktivních přijímačů může přijímat zprávy současně.

Vysílač má schopnost vysílat zprávy na sběrnici. Tato funkce je nutná např. u multimetru, aby mohl vysílat naměřené údaje na sběrnici. Minipočítač ve funkci vysílače nastavuje podle vloženého programu např. multimetr na měření např. napětí či nastavuje hodnoty frekvence na úrovní signálu u signálového generátoru. V daném čase může být aktivní pouze jediný vysílač. Systém IMS-2 zabezpečuje automaticky zásadu jediného vysílače. Pokud je aktivován nový vysílač, tak je do té doby aktivní vysílač pasívován zrušen.

Blok s funkcí řízení má schopnost vysílat různé stykové zprávy, např. adresy pro aktivaci výběrového spojení mezi přístroji, různé stykové povely pro celé řízení stykového systému apod. V daném čase může být aktivní pouze jediný blok s funkcí řízení. Na složitých pracovištích může být více řídících jednotek, které si mohou řízení postupně předávat. V těchto případech je nutné jednu řídící jednotku prohlásit za hlavní neboli systémovou jednotku.

V. 1. 1. Struktura sběrnice IMS - 2

Na přenos úplné informace, interfaceových zpráv a blokových zpráv je určena sada 16-ti vodičů, která je rozdělena do hod. skupin signálních vodičů (obr. č. 20).

Údajová sběrnice (DATA BUS), 8 vodičů DI01 + DI08, určená na přenos 7-mi bitových interfaceových a blokových zpráv ve formě bit - paralelní a byte sériové, asynchronně a obousměrně.

Řídící sběrnice přenosu údajů (data transfer control bus), 3 vodiče se používá na uskutečnění (zabezpečení) přenosu každého byte údajů po DIO signálních vodičích.

DATA VALID (DAV) - platné údaje, potvrzuje platnost údajů na DIO vodičích, aktivní úroveň "L" (nízká úroveň)

NOT READY FOR DATA (NRFD) - nepřipravený na přijímání údajů, používá se na indikaci podmínky připravenosti zařízení přijmout údaje, aktivní úroveň "H" (vysoká úroveň)

NOT DATA ACCEPTED (NDAC) - nepřijaty údaje, používá se na indikaci podmínky přijetí informace v zařízení, aktivní "H".

Vodiče DAV, NRFD, NDAC pracují v tzv. třídrátovém komunikačním (handshake) procesu, který slouží k přenosu každého byte údajů systémem sběrnice.

Obyčejné ovládání sběrnice (general interface management bus) - 5 vodičů - se používá na ovládání pořadí toku informací sběrnici :

ATTENTION (ATN) - pozor, používá se na stanovení interpretace údajů na DIO vodičích.

ATN = "H" (vysoká úroveň) po DIO vodičích se přenášejí informační údaje (blokové zprávy).

ATN = "L" (nízká úroveň) po DIO vodičích se přenášejí adresované nebo universální příkazy nebo adresy (interfejsové zprávy).

INTERFACE CLEAR (IFC) - nulování interfejsu, uvedení systému do známého klidového stavu, aktivní je úroveň "L".

SERVICE REQUEST (SRQ) - žádost o obsluhu, vysílá zařízení, aby indikovalo potřebu obsluhy resp. potřebu žádat o přerušování běžícího procesu, aktivní úroveň "L".

REMOTE ENABLE (REN) - dálkově možný, používá se (ve spojení s ji-

nými zprávami) na výběr mezi dvěma možnými zdroji programových údajů zařízení místně - panel, dálkově), aktivní úroveň "L".

END OR IDENTIFY (EOI) - konec nebo identifikace, používá se na indikaci konce přenosu vícenásobného byte nebo ve spojení s ATN na vykonání návaznosti paralelního hlášení, aktivní úroveň "L".

Přiřazení kontaktů doporučeného konektoru sběrnice IMS - 2
je následující :

Kontakt	vodič	kontakt	vodič
1	DIO 1	14	DIO 5
2	DIO 2	15	DIO 6
3	DIO 3	16	DIO 7
4	DIO 4	17	DIO 8
5	REN	18	ZEM
6	EOI	19	LOG ZEM (6)
7	DAV	20	LOG ZEM (7)
8	NRFD	21	LOG ZEM (8)
9	NDAC	22	LOG ZEM(9)
10	IFC	23	LOG ZEM(10)
11	SRQ	24	LOG ZEM (11)
12	ATN	25	LOG ZEM (12)
13	STÍNĚNÍ		

Log zem (n) se vztahuje na signální zpětnou zem příslušného ko-

taktu. Jako doporučený konektor je zvolen 25 pinový konektor CANNON. Každé zařízení má mít vidlicový konektor.

- Přiřazení adres zařízení v kódu ISO - 7

Adresy slouží pro organizaci a aktivaci, případně i pasivaci spojení mezi přístroji. Adresy se vybírají ze seznamu adresových znaků povolených normou a tyto nastavujeme na přístrojových stykích. Znaky vyčleněné z kódu ISO 7 jsou uvedeny v tab. č. 14.

Jako adresa se nesmí použít znak ? (otazník) a znak _ (podtrhnutí). Interfejsová zpráva "?" se používá k universální pasivaci funkce přijímač a zpráva "_" byla vyhrazena k pasivaci funkce vysílač. Hodnoty bitů (0, 1) uvedených v tab. 14 se nastavují obyčejně pěti přepínači na styku zařízení.

Uživatel nastavuje sice jen prvních pět bitů adresy, ale řídicí blok vysílá do sběrnice hodnoty sedmi bitů. Hodnotami zbývajících dvou bitů (b_6 , b_7) se zabezpečuje odlišení adresy pro vysílač a přijímač, i když jsou hodnoty bitů b_1 až b_5 společné.

- Operační posloupnosti (sekvence) na sběrnici IMS - 2

Většina komunikačních úloh si vyžaduje posloupnost kódovaných zpráv. To jsou doporučené posloupnosti (vhodné, vybrané z mnoha dalších). Uvedeme si některé z nich.

- Přenos dat

$ATN, UNL, LAD_1, \dots, LAD_n, TAD, ATN, DAB_1, \dots, DAB_n$
--

$\overline{\text{ATN}}$ - nastavení signálu ATN do úrovně "L"

UNL - pasivace funkcí přijímač (vyslání znaku ?)

LAD_n - vyslání adresy n-tého přijímače

TAD - vyslání adresy zařízení, na které mohu vyslat data ihned po přechodu signálu ATN do "H"

ATN - nastavení signálu ATN do "H"

DAB_n - vyslány byte, ukončení opětovným přechodem signálu ATN do "L", nebo na současné nastavení signálu EOI do "L" současně s vysíláním posledního byte, popř. po vyslání dohodnutých ukončovacích znaků v kódu ISO 7, např. CR nebo CR, LF.

- Sériové hlášení - kdykoliv je SRQ v úrovni "L".

$\overline{\text{ATN}}$, UNL, SPE, TAD_n , ATN, SBA nebo SBN, $\overline{\text{ATN}}$, SPD, ATN

$\overline{\text{ATN}}$, ATN, UNL podobně jako v předcházejícím

SPE, - uvolnění možnosti sériového hlášení

<p>→ TAD_n - $\overline{\text{ATN}}$ SBA nebo SBN $\overline{\text{ATN}}$ SBN</p>	<p>vyslání adresy zařízení, které má odevzdat stavový byte. byte stavů vyslaný zvoleným zařízením, pokud bylo vysláno SBN, smyčka se opakuje do přijetí SBA zvoleným zařízením, které vyslalo SRQ - "L"</p>
---	--

SPD - blokování možnosti sériového hlášení

SBN - stavový byte zařízení, které neoznámilo požadavek na obsluhu ($b_7 = \emptyset$)

SBA - stavový byte vyslaný zařízením, které indikovalo požadavek na obsluhu.

- 10 -
Paralelní hlášení

\overline{ATN} . \overline{EOI} , SBA , \overline{ATN} . \overline{EOI}

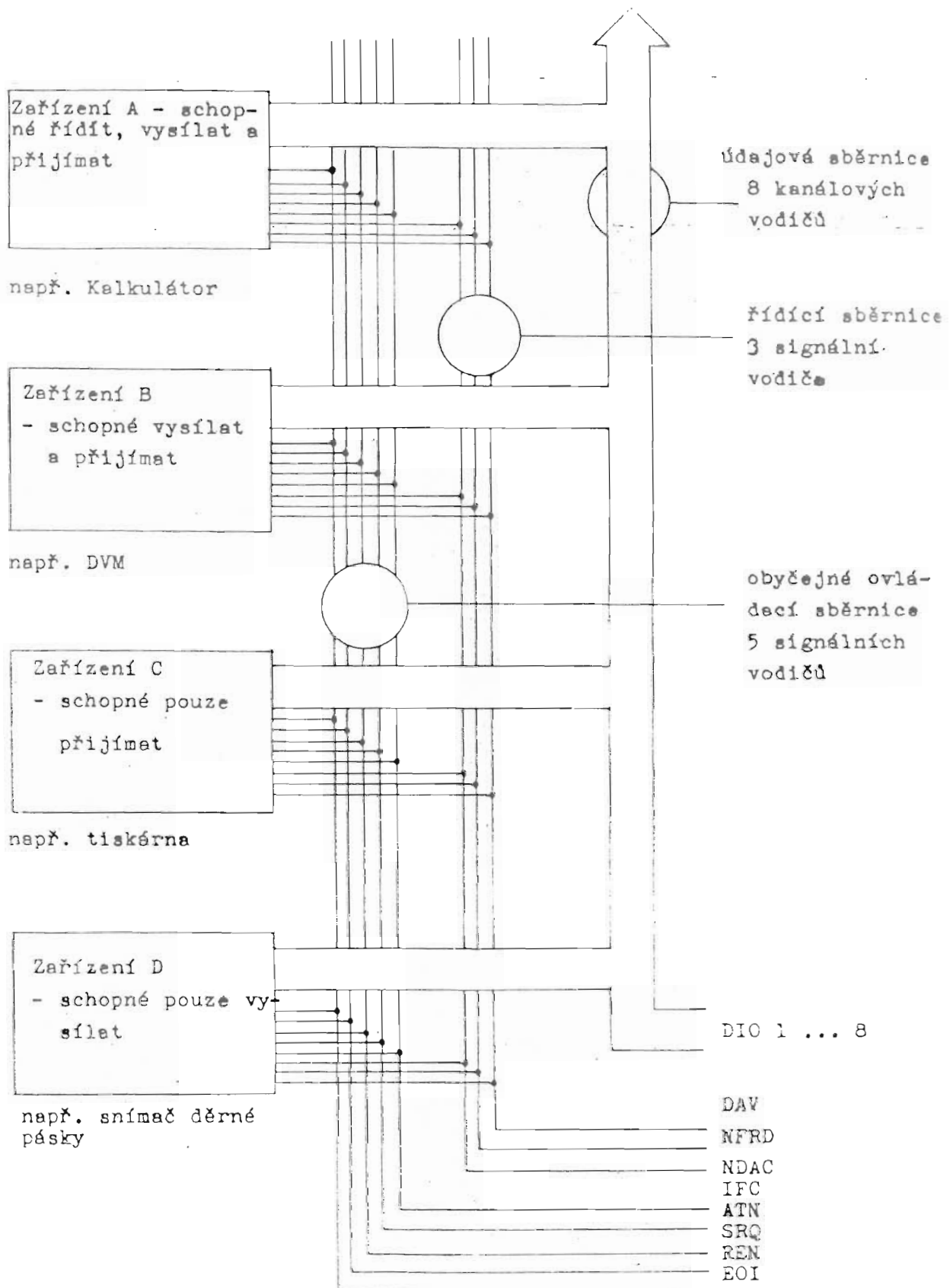
\overline{ATN} . \overline{EOI} - signály jsou současně na úrovni "L"

SBA - předurčené zařízení umístí požadavky na specifický DIO vodič. Tento stav by se přenášel do bloku zařízení bez komunikačního (handshake) procesu.

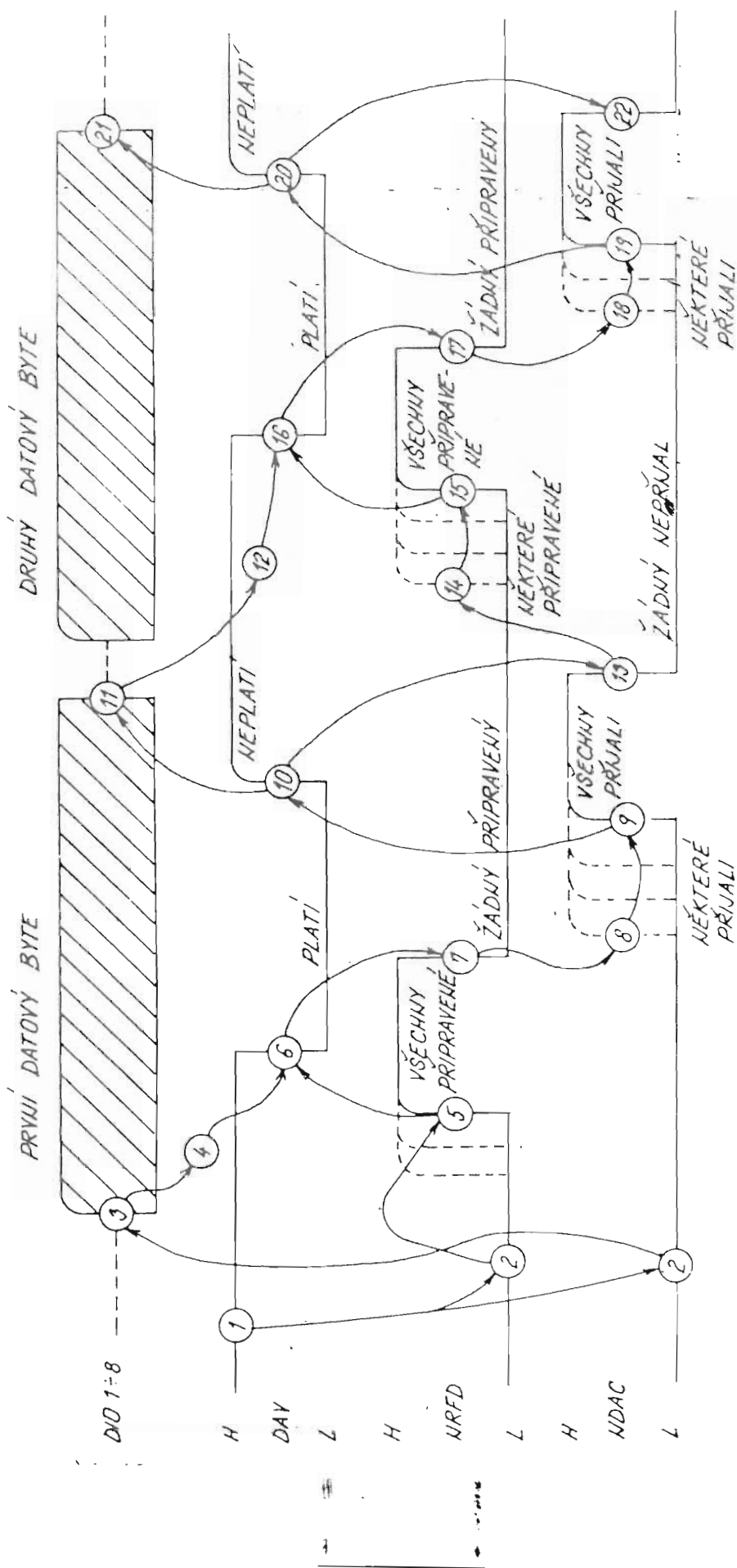
\overline{ATN} . \overline{EOI} - signály jsou uvedeny na úroveň "H".

Časový a tokový diagram komunikačního (handshake) procesu.

Každý byte údaj, přenesených na sběrnici IMS - 2, využívá komunikační (handshake) proces na výměnu dat mezi vysílačem a přijímačem. Na obr. č. 21 je komunikační proces s označením skutečných impulzních průběhů signálů vodičů DAV, NRFD, NDAC. Signály NRFD a NDAC představují sdružené průběhy, které jsou výsledkem dvou nebo více přijímačů, přijímajících tentýž byte v rozdílném čase. Na obr. č. 22 je stejná posloupnost dějů ve formě tokového diagramu na přenos bytu údajů mezi zdrojem a přijímačem. Stejná čísla na tokovém diagramu a na časovém diagramu posloupnosti dějů se týkají stejného děje.

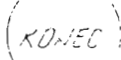


Obr. č. 20 - struktura IMS - 2

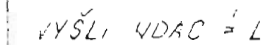


OBR. Č. 21 ČASOVÝ DIAGRAM KOMUNIKAČNÍHO PROCESU

ČINNOST BLOKU VYSÍLAČ



ČINNOST BLOKU PŘIJÍMAČ



b4	b3	b2	b1	s	r	b7	0	0	0	0	0	1	1	1	1	1
0	0	0	0	0	0	NUL	0	DLE ¹⁶	SP ³²	0	48	64	P	80	96	P ¹¹²
0	0	0	1	1	1	SOH	GT ¹	DC1	LLO ¹⁷	!	33	A ⁶⁵	Q	81	97	q ¹¹³
0	0	1	0	2	2	STY	2	DC2	18	"	34	B ⁶⁶	R	82	98	r ¹¹⁴
0	0	1	1	3	3	ETY	3	DC3	19	35	35	C ⁶⁷	S	83	99	s ¹¹⁵
0	1	0	0	4	4	EOT	SD ⁴	DC4	DCL ²⁰	36	4	D ⁶⁸	T	84	100	t ¹¹⁶
0	1	0	1	5	5	ENQ	PP ⁵	NAK	PPU ²¹	%	37	E ⁶⁹	U	85	101	u ¹¹⁷
0	1	1	0	6	6	ACK	6	SYN	22	8	38	F ⁷⁰	V	86	102	v ¹¹⁸
0	1	1	1	7	7	BEL	7	ETB	23	39	7	G ⁷¹	W	87	103	w ¹¹⁹
1	0	0	0	8	8	BS	GE ⁸	CAN	SPE ²⁴	(40	H ⁷²	X	88	104	x ¹²⁰
1	0	0	1	9	9	HT	TCT ⁹	EM	SPD ²⁵)	41	I ⁷³	Y	89	105	y ¹²¹
1	0	1	0	10	10	LF	10	SUB	26	42	58	J ⁷⁴	Z	90	106	z ¹²²
1	0	1	1	11	11	VT	11	ESC	27	43	59	K ⁷⁵		91	107	
1	1	0	0	12	12	PF	12	FS	28	44	60	L ⁷⁶		92	108	
1	1	0	1	13	13	CR	13	GS	29	45	61	M ⁷⁷		93	109	
1	1	1	0	14	14	SO	14	RS	30	46	62	N ⁷⁸		94	110	
1	1	1	1	15	15	SI	15	US	31	47	63	O ⁷⁹	UNT	95	111	DEL ¹²⁷

Tabulka č. 14 - znaky ISO 7. pro IMS - 2

Vysvětlivky tab. č. 14

1. Skupina primárních povelů (PCG - Primery Command Group)
2. Skupina sekundárních povelů (SCG - Secundary Command Group)
3. Adresní povely (ACG - Addressed Command Group)
4. Universální povely (USG - Universal Command Group)
5. Adresy pro přijímače (LAG - Listener Adress Group)
6. Adresy pro vysílače (TAG - Talker Adress Group)

Podtržené znaky neodpovídají kódu ISO 7, jsou to mnemotechnické pomůcky, označení jejich významu v IMS - 2.

GTL - Go To Local - návrat k místnímu ovládání
SDC - Selectet Derice Clear - selektivní nulování přístroje
PPC - Paralel Pool Configme - nastavení sestavy paralelního hlášení
GET - Group Execute Trigyer - skupinové spouštění
TCT - Take Con Troe - převzetí řízení
LLO - Local Locout - místní blokování
DCL - Derice Clear - nulování zařízení
PPU - Paralel Pool Unconfigme - rušení sestavy paralelního hlášení
SPE - Serial Pool Enale - povolení sériového hlášení
SPD - Serial Pool Disable - blokování sériového hlášení
UNL - UNListen - nepřijímá
UNT - UNTalk - nevysílá

V. 2. Odchyly IMS - 2 v mikropočítači IMS-2

Při návrhu mikropočítače - počítače IMS-2 bylo nutno vzít v úvahu několik argumentů, z kterých vyplynul tento kompromis pro sběrnici IMS - 2 v mikropočítači PMD 85: →

a) na sběrnici IMS má být ve funkční konfiguraci jeden účastník, a to osobní mikro počítač PDP-11

b) připojovací styl IMS - 2 není realizován konektorem CANNON (16-vozt), ale 30-ti pinovým konektorem FRB s vlastním rozložením signálů.

Uživatelský přístup k sběrnici IMS-2 je podporován příkazy BASIC - G, které jsou podmnožinou příkazů HP-IB (příkazy OUTPUT/ENTER).

V. 3. Software pro sběrnici kanálu 7

Výstup dat na komunikační sběrnici IMS - 2 se uskutečňuje příkazem OUTPUT, který má následující syntax

OUTPUT Z [ZK] ; výrez

kde "Z" nebo "ZK" znamenají adresy "LISTENER" přístrojů, které jsou připojeny na sběrnici IMS - 2. V případě adresování více přístrojů oddělujeme tyto adresy čárkou (viz hranatá závorka v příkazu OUTPUT).

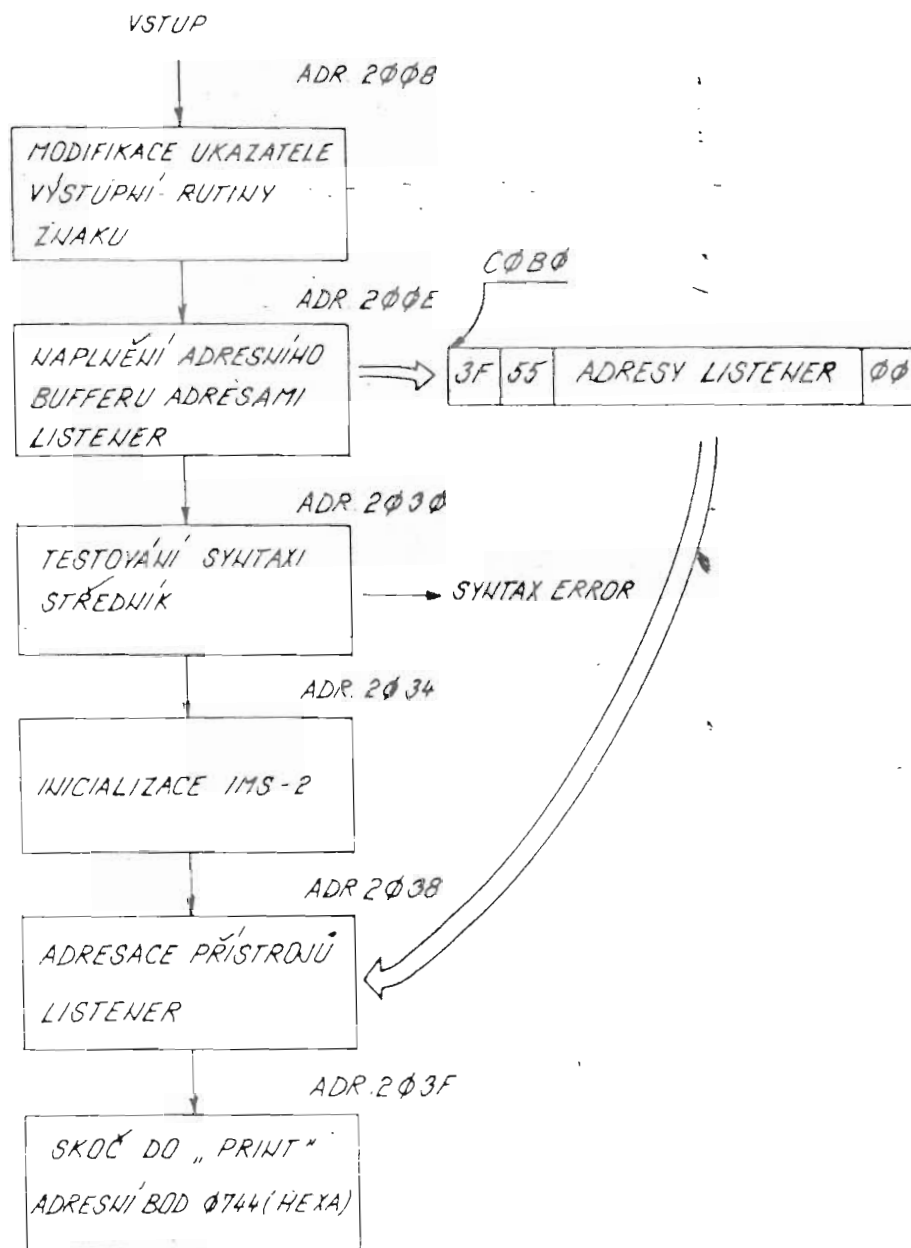
- výrez má obdobný význam jako u příkazu PRINT

Zpracování tohoto příkazu se uskutečňuje na několika úrovních. Po systémovém dekodování tohoto příkazu se začne vyhodnocovat výraz :

Z [ZK] ;

Vývojový diagram této rutiny (IMSPRINT) je zobrazen na obr. 8. 23. Adresový vektor této rutiny je 2008 (HEXA).

V záhlaví rutiny IMSPRINT dochází k přemodifikování ukazatele výstupního znaku na adresu 0015 (HEXA). Na tento ukazatel



OBRAZ 23 VÝVOJOVÝ DIAGRAM ČINNOSTI OUTPUT 7

- INICIMS (2042 HEXA) inicializace sběrnice IMS - 2

Tyto podprogramy může uživatel využít při svých modifikacích. Jako příklad může sloužit vzor systémového přístupu změny výstupu znaků.

Příklad č. 19 : Proveďte systémový přístup pro modifikaci příkazu OUTPUT 7 s možností binárního výstupu (8 bitů) znaků na sběrnici IMS - 2. Požadovaná syntax příkazu :

OUTPUT 7Z; X, Y

kde "Z" je adresa přístroje LISTENER

"X" je počáteční adresa uložených výstupních dat

"Y" počet byte, které mají být vyslány na IMS - 2.

Řešení :

1. krok - vytvoříme rutinu pro přebrání parametrů X, Y, které nastaví hodnoty ukazatelů ADDRESS X a POČET Y.

```
PAR XY : CALL 0933      } přebrání parametru X
          CALL 05C3      } do registru DE
          XCHG
          SHLD ADDRESS X   uložení na ukazatel ADDRESS X
          XCHG
          CALL 006D        testování syntaxí "čárka"
          DB ', '
          CALL 0933      } přebrání parametru Y
          CALL 05C3      } do registru DE
          XCHG
          SHLD POČET Y     uložení na ukazatel POČET Y
          XCHG
          RET
```

2. krok - vytvoříme výstupní rutinu pro výstup bloku dat na sběrnici IMS - 2 s využitím rutiny pro výstup znaku na IMS - 2 (rutina OUTIMS).

BLOKIMS:	LHLD POČET Y	počet byte v reg. DE
	XCHG -	-
	LHLD ADRESS X	počáteční adresa v HL
LOOP :	MOV A, M	vezmi znak z bufferu
	CALL OUTIMS	výstup znaku na IMS - 2
	INX H	zvyš adresu
	DCXD	sniž počet
	MOV A, D	
	CPI FF	} test posledního znaku
	JNZ LOOP	
	RET	návrat

3. krok - provedeme modifikaci původního výstupu z rutiny OUTPUT 7 na vstup do rutiny PRINT. Toto adresní místo (1FDE) nasměrujeme na naši novou rutinu BINIMS, ze které bude návrat zpět do interpreteru BASIC.

BINIMS :	CALL PAR XY	převzetí parametrů X, Y
	PUSH H	uchování ukazatele progr. BASIC
	CALL BLOKIMS	výstup bloku dat na IMS - 2
	POP H	návrat prog. čítače BASIC
	RET	návrat do BASIC

4. krok - pokud jsme nashromáždili výstupní binární data v paměťovém prostoru např. 26 000 (dec.) s počtem 1 000 byte, použijeme příkaz v následujícím tvaru :

OUTPUT 701; 26 000, 1000

kanál 7 → 701
adresa LISTENER - 01 → 26 000
počet přenášených byte → 1000
počáteční adresa bloku dat → 26 000

Závěrem uvádíme, že tento příklad může sloužit uživatelské modifikaci příkazu OUTPUT 7, kterou můžeme taktéž uložit do uživatelské ROMky, která se "postará" o jeho příslušnou modifikaci v interpreteru BASIC.

Používání standardního příkazu OUTPUT 7 pro výstup znaků na sběrnici IMS - 2 nepřináší žádné komplikace. V jeho příkazu se mohou nacházet i adresy nepřipojených účastníků LISTENER. Ukončení příkazu je dáno vysláním posledního znaku, který se zpracovával pod rutinou PRINT.

V. 4. Software pro ENTER kanálu 7

Vstup údajů po sběrnici IMS - 2 do operační paměti a jejich přiřazení dané proměnné je uskutečňováno příkazem

ENTER 7 adr ; proměnná (seznam proměnných)

Kde - "adr" představuje adresu periferního zařízení připojeného na sběrnici IMS - 2, které je v režimu vysílání dat (TALKER)

- proměnná nebo seznam proměnných představuje danou proměnnou, které se mají přiřadit vstupní přijímaná data. Syntax a vlastnosti výrazu za středníkem jsou stejné jako u příkazu INPUT. Tato poznámka platí i pro hodnotu znaků, které vstupují k dané proměnné.

Příkaz ENTER 7 představuje vůči příkazu OUTPUT určité nebezpečí, protože není systémově ošetřeno (časový limit) zvolení neexistujícího zařízení TALKER, od kterého jsou očekávána vstupní data. Dále je nutno dodržet přijímací protokol dat, který musí být ukončen domluveným znakem. V BASIC - G jsou to ukončovací znaky CRLF (ØD, ØA HEXA).

Pro možnost uživatelské modifikace příkazu ENTER 7 (zakončovací znaky, změna vstupního bufferu apod.) vysvětlíme si jeho činnost na vývojovém diagramu na obrázku č. 24.

Filosofie příkazu ENTER 7 vychází z činnosti interpretování příkazu INPUT, který v první fázi vyhledá ukazatel dané proměnné a v další po naplnění vstupního bufferu ho přiřadí.

Pro vstup dat po sběrnici IMS - 2 je vypracována zvláštní rutina vstupního znaku. Vektor této rutiny INIMS (adresa 20 EB) se umísťuje do ukazatele vstupní rutiny znaku (adresa 085F). Adresní pozice vstupního bufferu je dána parametrem na adrese 085C, a je inicializována na hodnotu 7F01. Po naplnění vstupního bufferu, dochází k přiřazování znaků proměnné, a proto je nutné nastavit ukazatel vstupního bufferu na počátek. Adresní místo tohoto parametru je 2212 (HEXA), na němž je umístěna hodnota 7F00, tedy o jednu pozici nižší než při jeho vstupu. Ukončovacím znakem naplňování znaků ASCII do vstupního bufferu je dvojice znaků CRLF (0D 0A), přičemž se testuje pouze poslední znak LF. Testování znaku je dáno maskou, která je umístěna na adrese 2113 (HEXA). V případě komparace se ukončuje vklad a znak předcházející, tedy znak "CR" se přepíše ukončovacím znakem 00, který je dán nutností při přebírání znaků při interpretování příkazu INPUT.

Stejně jako u příkazu OUTPUT 7 je třeba i v příkazu ENTER 7 před vlastním přenosem dat uskutečnit adresaci příslušného periferního zařízení, které je připojeno na sběrnici IMS - 2. Používá se stejná rutina (vektor 1FB2) se vstupním registrem začínajícím na adrese C0B0. Délka tohoto adresního registru je pevná a má 5 znaků. Následující obrázek znázorňuje jeho strukturu.

UNL		adr.přístroje	ukončení registru
3F	35	např. 41	00

Výstupním bodem z rutiny ENTER 7, ve které se provedly uvedené změny na exekutivě INPUT, je skoková instrukce do druhého vstupního bodu rutiny INPUT (adresa 081C). Skoková instrukce (opuštění záhlaví rutiny ENTER 7) je na adrese 214F. Tento bod je možno modifikovat, pokud bychom požadovali vložit další parametry do příkazu ENTER 7.

V. 4. 1. Modifikace příkazu ENTER 7

Modifikace příkazu ENTER 7 připadá v úvahu v podstatě ve dvou bodech.

- a) změna ukončovacích znaků vstupního protokolu
- b) změna typu přenášených znaků tj. přenos binárních dat (8 bitů)

Tyto požadavky může uživatel zvládnout vhodnou modifikací rutiny ENTER 7 - viz následující příklady.

Příklad č. 20 - Změňte koncový znak pro vstupní data ze sběrnice IMS - 2. Novým znakem má být znak CR (0D HEXA).

Řešení : Na adrese 2113 (HEXA) změníme masku "0A" na masku "0D".

POKE 8467, 13
 ↗ ↖
 adresa 2113 (HEXA) Znak CR (0D HEXA)

Příklad č. 20 a - Proveďte systémový návrh změny typu přenosu dat ASCII na binární (8 bitů) data.

Řešení : Při binárním přenosu dat nelze ukončovat přenos předem dohodnutým ukončovacím znakem (např. CRLF). Přenos musí mít jednoznačně určen počet přenášených dat a počet parametrů bude ukončovacím faktorem přenosu dat. Dále je třeba si uvědomit, že vyhrazený prostor pro vstup dat (254 znaků) může být v některých případech nedostatečný. Z tohoto důvodu je výhodnější určovat daný prostor přímo v příkazu ENTER 7. Tedy nová syntax pro náš příklad by měla být :

ENTER 7 adr ; X , Y

kde X představuje počáteční adresu vstupního bufferu binárních dat
Y je délka vstupního bufferu (počet binárních dat)

Jednotlivé fáze modifikace jsou následující :

1. krok : modifikujeme vstupní rutinu znaku INIMS pro ukončení na počet přijatých znaků. Předpokládáme, že parametr Y (počet dat) je uložen v registru DE.

```

BINDAT : DCX D          snížení počtu přijatých dat
          MOV A, D
          ORA A          } testování posledního
          JP 20F3        } přijatého znaku
          RET            návrat
    
```

Na uvedený programový fragment musíme nasměrovat testování znaku ve vstupní rutině INIMS. Tento bod má adresu 2114 (HEXA), na kterou umístíme skokovou instrukci s vektorem BINDAT. Tedy

C3 adr BINDAT

2. krok : Modifikujeme výstup z rutiny ENTER 7, který je nasměrován na vstupní adresní bod rutiny INPUT. Doplňme rutinu

ENTER 7 o převzetí parametrů X, Y a naplnění vstupního bufferu binárními daty. Tvar tohoto doplňujícího programového fragmentu je následující :

<u>ENTER BIN</u> :	CALL 0933	}	převzetí parametru X
	CALL 05C3		do registru DE
	PUSH D		uchování parametru X
	CALL 00 6D		testování syntaxi čárka
	DB ", "		
	CALL 0933	}	převzetí parametru Y
	CALL 05C3		do registru DE
	POP B		
	PUSH H		uchování programového čítače BASIC
	PUSH B		
	POP H		vrácení parametru X
	CALL INIMS		naplnění vstupního bufferu
	POP H		návrat progr. čítače BASIC
	RET		návrat do interpreteru

Adresní vektor rutiny ENTER BIN uložíme do skokové instrukce na adrese 214E. Těmito úkony jsme provedli požadované modifikace v příkazu ENTER 7.

Používání nového příkazu nám umožňuje zadaný paměťový prostor naplnit daty z přístroje TALKER po sběrnici IMS - 2. Převzetí dat a manipulace s nimi v množině proměnných je umožněna jednoduchým příkazem - funkcí PEEK.

Tedy např. příkazem ENTER 701; 20 000, 1 000 naplníme vstupní buffer, který je definován v paměťovém prostoru začínajícím adresou 20 000 (dec.) a končícím adr. 21 000.

prezetí binárních hodnot (rozsah 0 + 255) je umožněno programo-
vým kódem :

```
DIM A (10000)
FOR I = 0 TO 999
  A (I) = PEEK (20000 + I)
NEXT I
```

Po skončení programového cyklu se vstupní hodnoty z přístroje
s adresou 01 nacházejí v numerickém poli A.

Závěrem poznamenáváme, že v případě nutnosti obou příkazů
ENTER 7 (nemodifikovaný i modifikovaný) je možno použít (vyhradit)
jeden ze tří volných kanálů (2, 3, 6) a zopakovat nutné progra-
mové rutiny podle vzoru kanálu 7.

V. 5. Jiné využití kanálu 7

Když si nalistujeme obr. č. 18, který znázorňuje hardwareové
uspořádání kanálu 7, vidíme, že je charakterizován programovatelným
paralelním obvodem 8255A. Datové linky jsou vyvedeny od portu
A přes budič 8287, který je nageguje přímo na konektor K5. Ovláda-
cí směr toku dat je signálem linky PC0. Je-li na úrovni LOG 0, je
budič nasměrován pro výstup dat. Ostatní linky portu C, B jsou
vzájemně svázané, takže závisí na konkrétní periférii jak budou
využity. Není možné v plném rozsahu využít je jako u kanálu 4.
V zásadě je možno kanál 7 využít pro jedno periferní zařízení
s paralelním tokem dat (děrovač, snímač, A/D převodník atd.).
Programový přístup v tomto případě je možný s podporou příkazů

```
CONTROL 7, r; výraz
STATUS 7, r
```

které mají stejnou syntax a význam jako u kanálu 4.

V. 6. Povelý IMS - 2 (HPIB)

Pro komunikaci s přístroji na sběrnici IMS - 2 obsahuje BASIC-G pouze dva základní povelý pro transfer dat (OUTPUT, ENTER).

Kromě těchto datových povelů existují další agregované povelý, které ulehčují činnost s přístroji IMS - 2. Pokud chceme dodržet syntax podle fy HP, mohly by to být např. :

STATUS 7XX : A Příjem stavového slova přístroje s adresou XX při sériovém hlášení a přiřazení jeho decimální hodnoty proměnné A. Vyslány jsou zprávy podle popisu IMS - 2.

A=PPOLL (7) Příjem paralelních hlášení ze sběrnice přístrojů, kterým bylo umožněno paralelní hlášení a přiřazení stavového slova proměnné A.
Vyslány jsou zprávy podle popisu IMS - 2.

ABORTIO 7 Příkaz nuluje interfejsové funkce všech přístrojů na sběrnici. Vysláno je IFC do úrovně "L" po dobu cca 100 ms.

TRIGGER 7 spustí všechny přístroje adresované jako přijímače; (vysílá interfejsovou zprávu GET)

TRIGGER 7XX

XX - adresa přístroje
spustí přístroj s adresou XX (vysílá interfejsovou zprávu GET přístroji s adresou XX).

CLEAR 7 Nastaví všechny přístroje adresované jako přijímače do základního stavu (vysílá interfejsovou zprávu CLEAR).

CLEAR XX nastaví přístroj s adresou XX do základního stavu
(vysílá interfejsovou zprávu SDC)

LOCAL 7 nastaví všechny přístroje na místní ovládání
(interfejsová zpráva GTL)

LOCAL 7XX nastaví přístroj s adresou XX na místní ovládání.

Z důvodu omezené kapacity operační paměti nebylo možné přijmout tyto povely. Avšak ti uživatelé, kteří by požadovali některý povel ve své aplikaci, mohou si ho doplnit za podpory nižších příkazů (CONTROL, STATUS, USR, CODE) a uživatelské rutiny pro danou funkci. Jako vzor může sloužit následující příklad, který demonstruje vytvoření (náhradu) příkazu PPOLL.

Příklad č. 21 : Vytvořte náhradu příkazu PPOLL

Příkaz PPOLL umožňuje dotazování (testování) sběrnice IMS - 2 zda některé připojené periferní zařízení nepožaduje obsluhu (linka SRQ). V případě, že ano, tak se na datové sběrnici objeví jeho příslušná hodnota 1 z 8. Tato funkce nám umožňuje sledovat přerušování pro 8 přístrojů. Uživatelská rutina, která ve smyslu normy IMS realizuje dotaz a přebrání adresy periférie má následující tvar :

PPOLL : MVI A, 92

CALL 2044 inicializace sběrnice IMS - 2

MVI A, 71

OUT 7E nastavení ATN = 0, EOI = 0, REN = 0

XTHL zpoždění odezvy

IN 7C převzetí hodnoty - data

PUSH PSW uchování hodnoty - data

CALL 2046 inicializace IMS

POP PSW	návrat dat
STA ADR PPOLL	uložení do zápisníku
RET	návrat

Zahníždění do uživatelského programu BASIC můžeme provést dvěma formami.

a) Použití funkce USR

Tato forma vyžaduje mít umístěnu rutinu PPOLL v operační paměti a s podporou příkazu USR se vykonává její interpretování. Funkce USR nám umožní odevzdat obsah akumulátoru (tedy kód periferie) do příslušné proměnné. Potom programový fragment BASIC má tvar :

100	A = USR (adr PPOLL)
110	ON A GO TO
120	GO TO 100

Komentář : 100 provedení příkazu PPOLL

110 testování obsahu A na příslušnou hodnotu přístroje

(můžeme použít i příkaz IF - THEN)

120 návrat na nový dotaz PPOLL

b) použitím příkazu CODE

V tomto případě uživatel nemusí vkládat rutinu PPOLL do operační paměti zvláštními prostředky. Využije možnosti příkazu CODE, které umožňují vykonávat binární podprogramy. Nosiči těchto binárních podprogramů jsou řetězcové proměnné. Jejich naplnění můžeme uskutečnit během interpretování programu BASIC.

Dalže, jestliže přiřadíme buňku s adresou 9215 (dec.) tj. 23FF (HEXA) jako schránku pro převzetí výstupního argumentu z rutiny PPOLL, můžeme vytvořit programový fragment BASIC následovně :

```
10 A$ = " 3E92CD44203E71D37EE3E3DB7C32FF23C34620"  
20 CODE A$  
30 B = PEEK (9215)  
40 ON B GO TO .....  
50 GO TO 20
```

Komentář : 10 uložení rutiny PPOLL do řetězce A\$
(ve zjednodušené formě)
20 provedení rutiny PPOLL
30 přiřazení výsledné hodnoty PPOLL proměnné B
40 testování hodnoty B
50 návrat na nový dotaz sběrnice IMS (PPOLL)

V. 7. Příklady programové obsluhy přístrojů IMS - 2 (HPIB)

V této části si vysvětlíme na příkladech programové obsluhy některých periferních zařízení, které mají sběrnici IMS - 2 popř. HPIB. Pro připojení a jejich programovou obsluhu je třeba znát příslušnou adresu zařízení a komunikační protokol pro přijímání výzvy nebo odevzdávání výsledků. Tyto protokoly jsou uvedeny v uživatelských příručkách příslušných periferních zařízení. Je třeba dbát hlavně na formu, v jaké odevzdávají výstupní data - jejich ukončovací znaky. V případě, že periferní zařízení nemají možnost volby typů ukončovacích znaků, je nutno modifikovat interpreter BASIC - G podle předcházejících zásad.

Příklad č. 22 : Vytvoření programových řádků pro obsluhu číslí-
cového voltmetru M1T 330 (adresa přístroje 01).

Řešení : Na základě řídicích povelů obsažených v tab. č. 1 str. 12
v uživatelské příručce přístroje M1T 330 nastavíme
přístroj do pracovního režimu příkazem

OUTPUT 701 ; "RQDQWQC"

- kde RQ je rozsah 300 mW

DQ ovládá filtru vypnout

WQ bez zpoždění startu

C kalibrace jednorázová

Start měření uskutečníme příkazem

OUTPUT 701 ; "E"

Po tomto povelu můžeme uskutečnit sčítání naměřené hodnoty
napětí.

ENTER 701 ; A\$

Naměřená hodnoty se nachází v řetězci A\$ ve tvaru :

V ± Z . XXXX E ± Y , CRLF

znak napětí

číslo v exponenc.

ukončovací znaky

tvaru

Přístroj M1T 330 je řešen novou koncepcí ovládání orientovanou
na mikropočítačový systém 8080. To mu umožňuje inteligentní
práci se sběrnici IMS - 2.

Příklad č. 23 : Obsluha RLCG MOST - VOLTMETER BM559 prostřed-
nictvím sběrnice IMS - 2. (adresa přístroje 01).

Řešení : Přístroj BM 559 má řešenu logiku interface IMS - 2
diskrétními prvky. Tento nedostatek přináší některá
omezení, Jedno z nejvýznamnějších je, že není možná dál-
ková změna rozsahu měření.

Spuštění měření se uskuteční příkazem :

OUTPUT 701 ; "M1E"

```

├── "exekute" (proved)
├── 1 kanál A
├── 2 kanál B
└── programování kanálu
    
```

Pro převzetí naměřené hodnoty do proměnné použijeme příkazu :

ENTER 701 ; A\$

kde hodnota v A\$ je pevného tvaru a má následující strukturu :

R	rozměr měřené veličiny (Farad, Henry, SIEmens, OHM, Volt)
+	znaménko měřené veličiny
x	mantisa měřené veličiny
.	pevná desetinná čárka
E	znak oddělující exponent
+	znaménko exponentu
Y	číselná hodnota exponentu
;	ukončovací znak (podle IFC "mezera - středník")

Jak je ze struktury vstupních naměřených hodnot vidět, ukončovací znaky, jimiž jsou uzavřena vstupní data, jsou jiné než obsahuje vstupní rutina znaku v rutině ENTER 7. Máme dvě možnosti, buď změnu znaků přímo v přístroji nebo vyslat příkaz pro změnu masky koncového znaku. Tento příkaz má tvar :

POKE 8467, 59

└─ znak středník

Až po této úpravě můžeme provést přečtení dat. Jinak nám může tento úkon zničit interpreter i vytvořený program a jediná možnost další činnosti vede přes manuální RESET počítače.

Příklad č. 24 : Připojení LCR-metru fy HP (typ HP 4275 A)

Řešení :

1. Programování napájení měřeného prvku - str. 3 - 56

4275 MULTI FREQUENCY LCR METER - OPERATING MANUAL

OUTPUT 7 $\frac{XX}{1}$; "BI + $\frac{NNN}{2}$ E + $\frac{NNV}{3}$ "

1. XX adresa LCR metru
2. NNN tři digity pro mantisu napětí
3. NN dva digity pro exponent napětí

Vzor programového řádku :

OUTPUT 717; "BI + 010 E + 01V"

2. Programování funkcí LCR-metru

(strana 3 - 56 OPERATING MANUAL)

OUTPUT 7 $\frac{XX}{1}$; "řetězec"

1. XX adresa LCR-metru
2. řetězec skládající se z programových kódů LCR-metru (tab.
3. - 11. OPERATING MANUAL)

Vzor programového řádku :

OUTPUT 717; "A2 B1 C1 F17 H0 I0 R31 T2"

Význam :

- A2 - měření kapacity
- B1 - měření ztrátového činitele
- C1 - automatická volba náhradního obvodu
- F17 - nastavení frekvence generátoru 1 MHz
- H0 - měření s menší přesností
- I0 - vypnutí signálu SRQ (žádosti o obaluhu)
- R31 - automatické přepínání rozsahů měřených veličin
- T2 - externí spouštění měření

Pro spouštění měření se použije kódový řetězec "E" (execute).

Vzor programového řádku :

```
OUTPUT 717; "E"
```

Je možno vložit řetězec "E" i na konec programového řetězce např. :

```
OUTPUT 717; "A1 B1 C1 F16 H1 IØ R31 T2 E"
```

3. Vstup naměřených údajů do počítače

HP 4275 A má dva formáty výstupu naměřených hodnot. Tyto se nastavují na zadním panelu přístroje.

FORMÁT A : Obsah displeje A a displeje B jsou vysílány v následujícím formátu :

<u>X X X X</u>	<u>+</u>	<u>N N N N N N E</u>	<u>+</u>	<u>N N</u>	<u>X X</u>	<u>+</u>	<u>N N N N N N E</u>	<u>+</u>	<u>N N</u>	<u>CR LF</u>
2 3 4 5		6		7 8 9		10		11		

Například :

```
P N N C + L. 25600 E - 10, ND 8. 102 50 E - 02 CR LF
```

tento údaj se načítá do jednoho řetězce např. A\$ příkazem

Například :

```
ENTER 717; A$
```

Protože formát údajů je pevný, lze přenést naměřené údaje do číselných proměnných pomocí řetězcových funkcí :

vzor : $C = \text{VAL}(\text{MID\$}(A\$, 6, 12)) = C = 1.256 \text{ E} - 10$

$D = \text{VAL}(\text{MID\$}(A\$, 21, 12)) = D = 8.1025 \text{ E} - 2$

FORMÁT B :

Obrazy displejů A a B jsou vysílány v následujícím formátu :

<u>X X X X</u>	<u>+</u>	<u>N . N N N N N E</u>	<u>+</u>	<u>N M</u>	<u>CR LF</u>
1 2 3 4 5		6		11	

<u>X X</u>	<u>+</u>	<u>N . N N N N N E</u>	<u>+</u>	<u>N N</u>	<u>CR LF</u>
8 9		10		11	

1 - mezera

2 - druh náhradního obvodu

- | | |
|------------------------|-------------------------|
| 3 - měrná frekvence | 7 - čárka |
| 4 - stav displeje A | 8 - hodnota displeje B |
| 5 - funkce displeje A | 9 - funkce displeje B |
| 6 - hodnota displeje A | 10 - hodnota displeje B |
| | 11 - potvrzení údajů |

Vzor : S N N C + 8.256BB E - 9 CR LF

NQ + 1.257BB E + BB CR LF

Tento údaj je nutno načítat do dvou řetězcových proměnných
A\$, B\$.

Například : ENTER 717; A\$, B\$

Převod údajů řetězců na číselné proměnné provedeme :

C = VAL (MID \$ (A\$, 6, 12)) C = 8.256 E - 9

Q = VAL (MID \$ (B\$, 3, 12)) Q = 1.257 E - 2

Příklad č. 25 : Připojení čítače BM 526 pomocí interfejsové
jednotky BP 5269

Řešení : 1. Programování funkcí čítače (manuál INTERFACE /IMS - 2
BP 5269 str. 24)

OUTPUT 7XX; "M N T N E"
1 2 3 4 5 6

1. adresa přístroje
2. znak "M" pro funkci přístroje 3
3. druh funkce
4. znak "T" pro měrný rozsah
5. "E" znak pro vykonání /EXEKUTE/

Příklad - OUTPUT 720; "M 3 T 3 E"

M 3 - měření frekvence

T 3 - měrný interval 1 ms

- E - znak pro vykonávání

2. Zesnímaní naměřených hodnot

Tvar vstupních údajů je následující

<u>X</u>	<u>X</u>	⁺	<u>N</u>	<u>N</u>	<u>N</u>	<u>N</u>	<u>N</u>	<u>N</u>	<u>N</u>	<u>N</u>	<u>E</u>	⁺	<u>N</u>	<u>CR</u>	<u>LF</u>
1			2								3			4	

1 - fyzikální měřená jednotka

2 - mantisa

3 - exponent

4 - potvrzení údajů

Např.: Hz + 0 0 0 154 0 0 E + 3 CR LF

představuje hodnotu - frekvence (Hz) 15,4 MHz

vstup údajů do počítače uskutečnime příkazem např. ENTER 720; F8

Přeměna údajů z řetězce do číselné proměnné se provede operací :

F = VAL (MID\$(F8, 3, 12))

VI. Aplikační konektor pro I/O

Cílem této kapitoly je popsat využití aplikačního konektoru K2, na který je vyvedena část lokální sběrnice mikropočítače postačující pro práci s I/O podsystémem. Na obr. 25 je znázorněna obvodová architektura tohoto řešení. Jde v podstatě o budiče TTL, které oddělují dané okolí od počítačové sběrnice. Datová sběrnice je oddělena budičem 8286, který je aktivní pro množinu těchto adres :

A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
1	X	X	X	1	1	X	X

Z této adresní kombinace vychází možnost připojení 8 periferních zařízení typu 8255A, 16 typu 8251 atd.

Směr přepínání dat přes budič je uskutečňován novým signálem I/OB.

Nejjednodušším využitím aplikačního konektoru bez přidání zařízení je jako 8-mi bitový vstupní kanál, který bude aktivován některou povolenou adresou I/O např. FF (HEXA). Pro tuto množinu I/O zařízení interpreter BASIC nemá výkonné povely (OUTPUT, CONTROL, ENTER, STATUS). Jedinými příkazy pro obsluhu I/O zařízení jsou příkazy OUT a INP.

Příklad č. 26 : Napište příkaz pro vstup dat před aplikační konektor

Řešení : A = INP (255)

Proměnná A nabude hodnoty vstupních dat připojených přímo na aplikační konektor.

Aby bylo uživateli zřejmé, jak využít tento aplikační kanál uvádíme příklad, který poslouží jako vzor. V příkladu je uvedeno obvodové řešení vodiče minifloppy disku a jeho systémový návrh programové obsluhy.

Příklad č. 27 : Navrhněte připojení řadiče minifloppy diskové jednotky, který je orientován na obvod 8271.

Tento příklad není úplným stavebním návodem pro připojení minifloppy disku, včetně obslužných rutin R/W data. Má posloužit jako ideový návrh, ve kterém se klade důraz na systémový přístup k dané problematice. Pokud má uživatel zájem o tuto aplikaci, doporučujeme mu uživatelskou příručku pro minifloppy disky MFD 85 části V, VI, VII, které se připravují k vydání. Pro pochopení této aplikace je nutno prostudovat obvod FDC i 8271.

STRUČNÝ POPIS OBVODU 8271

Obvod 8271 je kompletní programovatelný řadič pro floppy diskovou jednotku velikosti 8" nebo 5 1/4". Umožňuje jednostranný

případně oboustranný záznam s jednoduchou hustotou záznamu ve formátu IBM-3740.

Jeho činnost je programovou záležitostí, která je charakterizována vydáním tzv. COMANND slova s příslušnými jeho parametry a čtením RESULT registru, který poskytuje všechny potřebné informace o provedené COMANND fázi. O stavu právě probíhající fáze hovoří obsah tzv. STATUS registru. Celkově má obvod 8271 pět registrů, ke kterým je adresní přístup.

Podle toho, jak se mají data přenášet do a z operační paměti můžeme zvolit obvod 8271 pro režim tzv. NON DMA a DMA MODE. K této činnosti je podporován svými linkami DRQ - žádost o DMA, DACK - potvrzení žádosti o DMA a INT - žádost o programovou obsluhu. Tyto tři linky jsou velmi důležité při našem obvodovém návrhu.

Pro úplnost ještě uvádíme, že jediným externím blokem obvodu 8271 je separátor čtených dat, který má za úkol oddělit data od hodinových pulzů. K tomu má vyvedeny vstupní linky DATA WINDOW a UNSEAPRATED DATA.

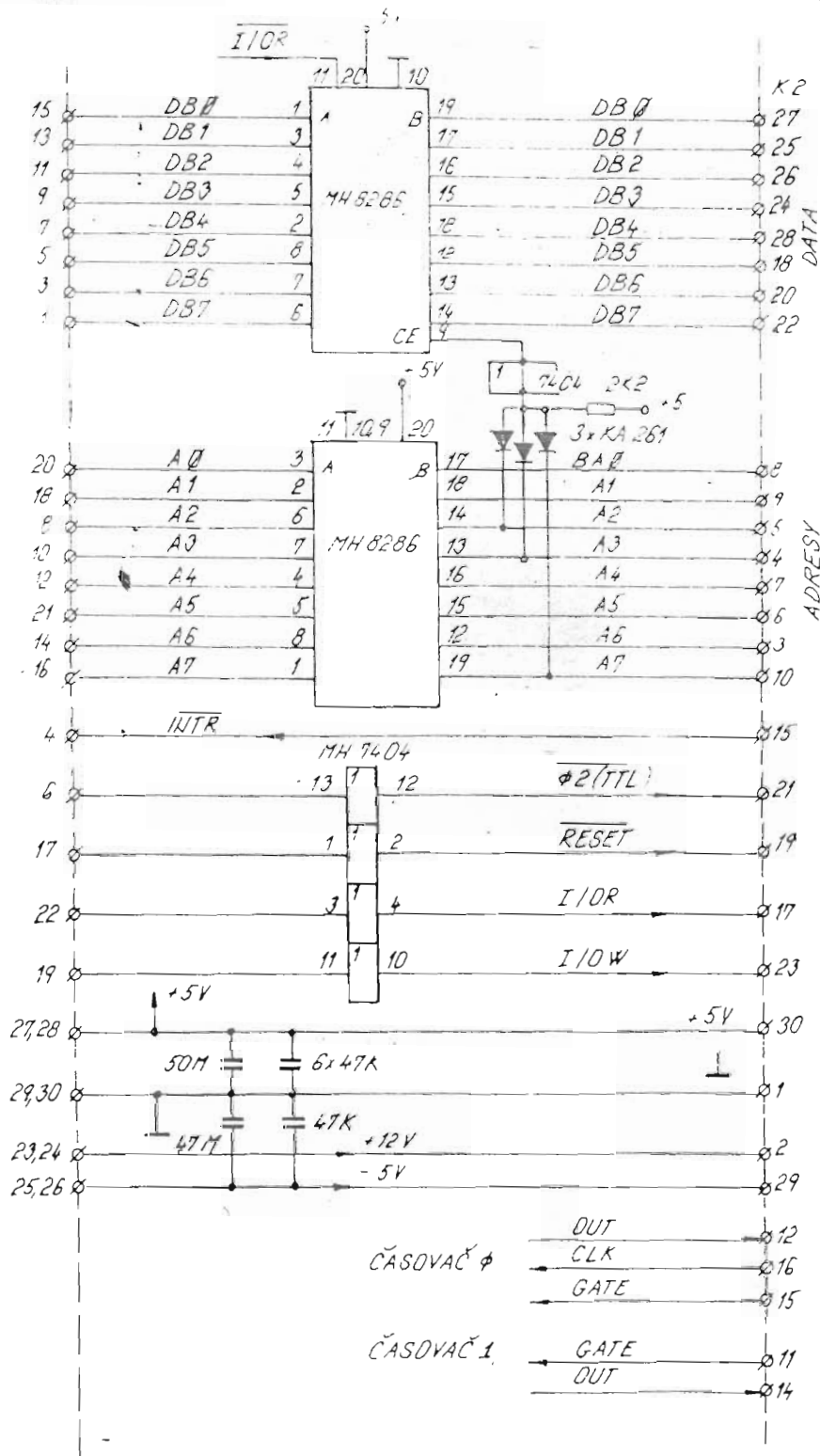
OBVODOVÉ ŘEŠENÍ ŘADIČE MINIFLOPPY DISKU

Při návrhu vodiče MFD jednotky se vycházelo z maximální obvodové jednoduchosti na úkor software. Na obr. č. 26 je kompletní interface, který umožňuje spolupráci se standardní mini a mikromechanikou (8", 5 1/4", 3 1/4"). Separátor dat je tvořen obvodem 74123, který vytváří tzv. WINDOW (okno) pro data. Časová konstanta pro minifloppy je přibližně 5,7 ns. Je to doba do které může padnout jeden datový pulz, který je třeba vyselektovat mezi dvěma hodinovými pulzy (základní FM).

-121a-

SYSTEMA KONEKTOR
M.P.

121a. KONEKTOR



OBR. Č 25 APLIKAČNI KONEKTOR

Návrhu je třeba se řídit *nevytáhnout* *radice* *přímého přístupu* dat DMA a v podstatě by to nebylo ani *protože* aplikační konektor neobsahuje plnou adresovou sběrnici, operační paměti. K obvodu 8271 přistupuje tedy *jako k běžnému perifernímu* obvodu, který požaduje nebo *poskytuje datový výke*. Je tedy o princip tzv. byte po byte. Tento *sériový přenos dat* je podmíněn přenosovou dobou, během které se musí uskutečnit jednobytový transfer dat. Pro minifloppy disk je to doba do 64 μ s a pro standard disk doba do 32 μ s. Pokud by tento dost kritický požadavek byl splněn s dostatečnou rezervou, přistoupilo se ke speciálnímu využití obvodu 8271 pro transfer dat. Princip spočívá v tom, že obvod 8271 se naspecifikuje *pro přenos dat v režimu DMA*. Při této činnosti se generuje linka *DRQ (žádost o DMA)* která se testuje v přenosové rutině. Potvrzením *DRQ* je \overline{DACK} je chápáno jako linka \overline{CS} periferního obvodu. Tehdy dojde ke čtení nebo zápisu dat. Tato činnost je stereotypní bez programového výstupu. Jediným možným výstupem je žádost o programovou obsluhu, tedy linka INT. Tato linka v režimu DMA je generována vždy na základě vnitřní události probíhající operační fáze v obvodu 8271. Může to být ukončení přenosu dat (načtení bloku dat) nebo předčasné ukončení činnosti v závislosti na některé příčině ať už technické nebo systémové (otevřená dvířka disku, nenačtený sektor atd.). Na obr. č. 27 je uvedena tabulka, která zahrnuje adresaci obvodu 8271, jeho vnitřních registrů. Testování linky DRQ je realizováno ve zjednodušené formě obvodem 7403 (otevřený kolektor), který je připojen na datovou linku D_2 .

PROGRAMOVÝ NÁVRH OBSLUHY ŘADIČE 8271

V této části si uvedeme systémový návrh pro realizaci potřeb-

ných obslužných rutin obvodu 8271. Jsou to v hierarchii diskového operačního systému ty nejspodnější rutiny, protože zabezpečují elementární práci s diskem.

Celá činnost obsluhy 8271 je postavena v podstatě na čtyřech rutinách.

1. Rutina pro čtení byte z 8271 a jeho popis do operační paměti

```

READ : IN DRQ
      RAR      } testování linky DRQ od 8271
      JC READ
      IN DATA      čtení dat z obvodu 8271
      STAX D        zápis dat do operační paměti
      INX D          zvýšení adresy operační paměti
      JMP READ       nový cyklus čtení dat z 8271
    
```

Vstupním argumentem této rutiny je registr DE, ve kterém je uložena adresa operační paměti, na kterou se mají čtená data z obvodu 8271 ukládat.

2. Rutina pro zápis byte do 8271 z operační paměti

```

WRITE : IN DRQ
      RAR      } testování linky DRQ
      JC WRITE
      LDAX D      čtení z operační paměti
      OUT DATA    zápis byte do 8271
      INX D        zvýšení adresového ukazatele
      JMP WRITE    nový cyklus zápisu dat do 8271
    
```

Opět vstupním argumentem je registr DE - ukazatel adresového místa operační paměti.

3. Rutina pro obsluhu programového přerušování

```

POP PSW      zvýšení návratu STACK od INT
    
```


ADRESA I/O	ČTENÍ	ZÁPIS
9C	STATUS REGISTER	COMMAND REG
9D	RESULT REGISTER	PARAMETER REG
9E	—	RESET REG

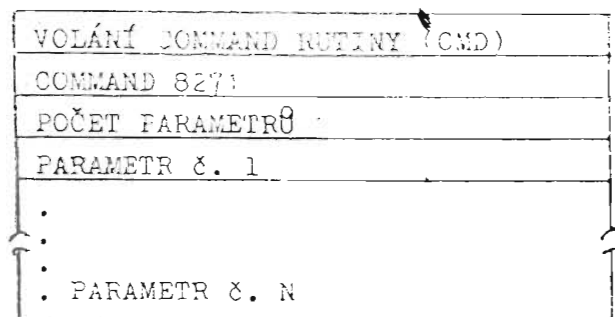
OBČ 27 ADRESOVÁNÍ REGISTRŮ OBVODU 8271

IN RESULT	přidělení result. registru 8271,
RAR	posunutí vpravo
STA STATUS	zapsání do zápisníku adr. STATUS
RET	návrat do hlavní rutiny

.. "COMMAND" rutina :

Tato rutina zabezpečuje interpretování požadované činnosti obvodu 8271. Činnost, která se má provést, je dána v "COMMAND BYTE" a v "PARAMETR BYTE", která může být i více podle druhu požadované činnosti.

Argumenty rutiny se ukládají za jejím voláním. Na následujícím obrázku je zobrazena struktura COMMAND RUTINY.



Poznámka : Požadovaný "COMMAND BYTE" a "PARAMETR BYTE" jsou uvedeny pro každou žádanou činnost v manuálu obvodu 8271. Pro některé činnosti si je uvedeme v dalším popisu.

Tvar rutiny pro nastavení obvodu do žádané činnosti a jeho provedení je následující :

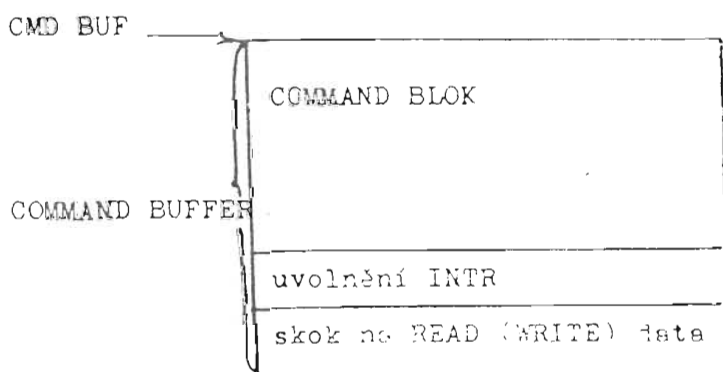
<u>CMD</u> :	IN STATUS	testování přístupu ke COMMAND
	ANI A0	registru 8271
	JNZ CMD	
	XTHL	nastavení HL na argumenty

```

MOV A, M          zápis COMMAND BYTE
OUT COMMAND
INX H             zvýš ukazatel
MOV B, M          počet parametrů
LOOP 1: INX H     zvýš ukazatel
LOOP 2: IN STATUS
ANI 20            testování přístupu k PARAMETR
JNZ LOOP 2        registru obvodu 8271
MOV A, M          zápis PARAMETR BYTE
OUT PARAMETR
DCR B             sniž počet
JNZ LOOP 1        cyklus
INX H             zvýš ukazatel adresy
XTHL              návrat do hlavní rutiny
RET

```

Pozice parametru "TRACK, SEKTOR" v COMMAND POVELOVÉM BLOKU je vždy na stejném místě při všech funkcích R/W data. Je výhodné proto vytvořit si tzv. "COMMAND BUFFER", do kterého by se přenesl pro požadovanou činnost příslušný povelový blok. Takový buffer by měl mít následující složení :



Na následujících příkladech si ukážeme jak je nutno postupovat při sektorovém čtení a zápisu dat.

```
***** DAT CE 128 BYTE SEKTORU DISKETY
```

<u>READ SEKTOR</u> :	IXI H, TABREAD	nastavení ukazatele tabulky
	RDSK MVI B, ØB	počet prvků v tab. TABREAD
	CALL MOVE	rutina pro přenos tab. do CMDBUF
	LHLD ADRDMA	nastavení ukládací adresy paměti
	XCHG	změna do registru DE
	LHLD TRCSEK	převzetí parametru TRACK/SEKTOR
	SHLD CMDBUF+Ø5	nastavení TRACK/SEKTORU v CMDBUF
	JMP CMD BUF	skok na interpretování CMDBUF
<u>TABREAD:</u>	CALL CMD	podprogram CMD
16		povelový byte pro čtení dat ze 128 byte sektoru
Ø2		počet parametrů
ØØ		TRACK
Ø1		SEKTOR
FB		uvolnění přerušení
JMP READ		skok na rutinu čtení z 8271

ZÁPIS DAT DO 128 BYTE SEKTORU DISKETY

<u>WRITE SEKTOR</u> :	LXIH, TABWRITE	nastavení ukazatele tab. pro zápis dat
	JMP RDSK	skok do rutiny READSEKTOR
<u>TABWRITE</u> :	CALL CMD	podprogram CMD
	ØA	povelový byte pro zápis dat (128 byte)
	Ø2	počet parametrů
	ØØ	TRACK
	Ø1	SEKTOR
	FB	uvolnění přerušení
	JMP WRITE	skok na rutinu zápis do 8271

NASTAVENÍ R/W HLAVY MECHANISMU NA STOPU (TRACK) NULA

<u>TRACK 0</u> :	CALL CMD	podprogram CMD
	29	povelový byte pro nastavení TRACK
	01	počet parametrů
	00	TRAGK = 0
	FB	uvolnění přerušení
<u>HERE</u> :	JMP HERE	čekání na přerušení (ukončení) činnosti

Jak je z příslušných rutin vidět, jde o velmi jednoduchou skladbu výkonných diskových rutin, které tvoří základ pro diskový operační systém. Doporučujeme uživateli uživatelskou příručku část VII, která detailně popisuje diskový operační systém jednotky MFD 85.

VII. Závěr

V pěti kapitolách jsme se snažili popsat možnosti využití jednotlivých kanálů I/O. Nelze popsat všechny varianty aplikací počítače PMD - 85 pro jeho okolí. Byly vzaty pouze typické příklady, jejich připojovací protokoly včetně programové obalů. Taktéž bylo uvedeno, jak postupovat při modifikaci interpreteru na straně I/O pro speciální typ aplikace. Pro ty, kteří se budou snažit o takového úpravy, uvádíme na závěr této příručky stručně některá volání rutin z interpreteru BASIC - G, které by měly ulehčit a zkompresovat vlastní uživatelské podprogramy :

Seznam rutin z interpreteru BASIC - G pro stavbu I/O uživatelských podprogramů

<u>Adresa</u>	<u>popis rutin</u>
006D	Kontrola znaku v syntaxi příkazu; testovací znak se umísťuje za dany podprogram : CALL 006D

0075	vyhledání prvního znaku po mezeře (SPACE)
007D	komparace obsahu HL s obsahem DE
0500	rutina testování po výkonu povelu BASIC
00FF	skok do rutiny výpisu "SYNTAX ERROR"
06A6	rutina pro přiřazení hodnoty proměnné, na kterou ukazuje reg. HL
0B07	rutina vyhledání adresního místa proměnné. Ukazatel proměnné v reg. HL, (vstup) adresní místo proměnné v reg. DE /výstup/
16D7	Přepoččet ASCII čísla, na které ukazuje reg. HL na HEXA hodnotu, které po výpočtu bude obsažena v registru A
0933	} převod integer čísla ASCII do 16-ti bitů. Ukazatel v reg. HL, výstupní data v reg. DE. Podprogramy je třeba vykonat po sobě.
05C3	

P Ř Í L O H A A

- Programování kanálu 1, 4, 5
- Zapojovací protokoly :
 - děrovač DT 105S
 - CONSUL 2111
 - DZM 180
 - JOYSTICK
 - BM 578
 - FS 1503
 - PMD - PMD

Kanáľ 1 - sériový, programovateľný styk

Naprogramování kanálu : CONTROL 1, 1;

asynchronní režim :

rychlost			délka znaku				parita			počet stop bitů		
1x	16x	64x	5	6	7	8	sudá	lichá	bez	1	1,5	2
1	2	3	0	4	8	12	32	16	0	64	128	192

synchronní režim :

počet znaků				parita			SYNDET		počet bitů SYNC	
5	6	7	8	sudá	lichá	bez	vstup	výstup	1	2
0	4	8	12	32	16	0	64	0	0	128

řídící slovo pro povel :

Vyhledávání SYNC (pouze pro synchronní režim)	Interní nulování	Požadavek na vysílání (RTS = 0)	Nulování chybových příznaků	Nastavení T x R na nulu	Povolení příjmu	Koncové zařízení připraveno	Povolení vysílání
128	64	32	16	8	4	2	1

Stavové slovo : A = STATUS 1, 1

7	6	5	4	3	2	1	BIT 0
DSR	SYNDET	FE	OE	PE	TxE	RxRDY	TxRDY

Kanál 4 - programovatelný paralelní styk

Naprogramování kanálu : CONTROL 4, 3; k

Nastav řídicí slovo	skupina 0				skupina I		
	Režim 2	Režim 1	Vstup - port A	Vstup - port C _H	Režim 1	Vstup - port B	Vstup - port C _L
128	64	32	16	8	4	2	1

Nastavení linky portu C (bit set/reset) : CONTROL 4, 3; b
 přičemž $b = (\text{příslušný bit} \times 2) + \text{žádáný stav bitu}$

Příkaz pro výstup : OUTPUT 4 r; výraz

Příkaz pro vstup : ENTER 4 r; proměnná

Číslo registru r	funkce
00	A - režim 0
01	B - režim 0
02	C - režim 0
03	A - režim 1
04	B - režim 1
05	A - režim 2
06, 07	uživatel

STATUS kanálu : A = STATUS 4, 2

režim	BIT 7	6	5	4	3	2	1	0
1- vstup	I/O	I/O	IBF _A	INTE _A	INTR _A	INTE _A	IBF _B	INTR _B
1-výstup	$\overline{\text{OBF}}_A$	INTE _A	I/O	I/O	INTR _A	INTR _B	$\overline{\text{OBF}}_B$	INTR _B
2	$\overline{\text{OBF}}_A$	INTE ₁	IBF _A	INTE ₂	INTR _A	I/O	I/O	I/O

Kanál 5 - programovatelný čítač/časovač

Naprogramování kanálu - CONTROL 5, 3; k

časovač	funkce	režim - kód "k"					
		0	1	2	3	4	5
0	vzorkuj	0	0	0	0	0	0
	čti/nastav pouze dolní byte	16	18	20	22	24	26
	čti/nastav pouze horní byte	32	34	36	38	40	42
	čti/nastav oba byte (dolní první)	48	50	52	54	56	58
1	vzorkuj	64	64	64	64	64	64
	čti/nastav pouze dolní byte	80	82	84	86	88	90
	čti/nastav pouze horní byte	96	98	100	102	104	106
	čti/nastav oba byte (dolní první)	112	114	116	118	120	122
2	vzorkuj	128	128	128	128	128	128
	čti/nastav pouze dolní byte	144	146	148	150	152	154
	čti/nastav pouze horní byte	160	162	164	166	168	170
	čti/nastav oba byte (dolní první)	176	178	180	182	184	186

Programový přístup :

nastavení : CONTROL 5, ČASOVAČ; byte , byte

čtení : A = STATUS 5, ČASOVAČ

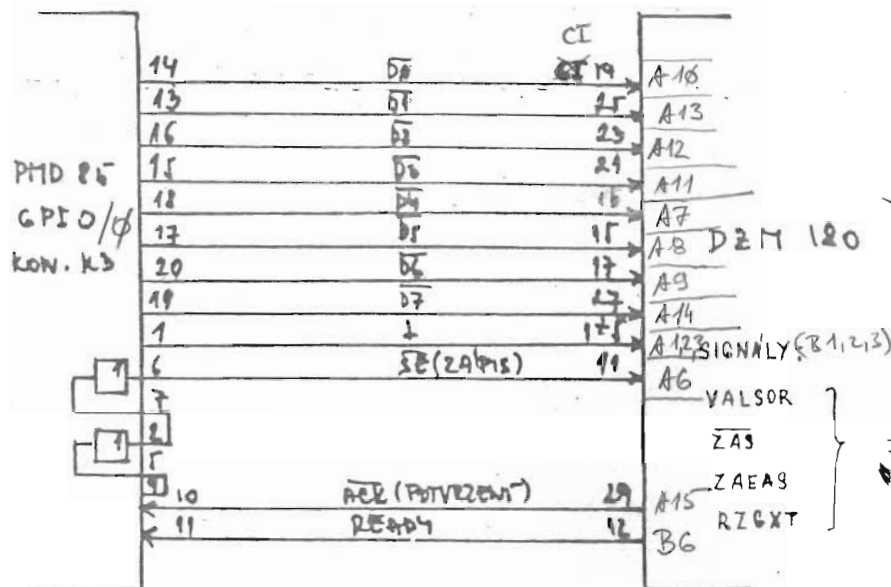
Tabulka ASCII

0	NUL	32	SP	64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	'	71	G	103	g
8	BS	40	(72	H	104	h
9	HT	41)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	Ø	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	ETB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z

27	ESC	59	;	91	[123	{
28	FS	60	<	92	\	124	!
29	GS	61	=	93]	125	}
30	RS	62	>	94	^	126	
31	US	63	?	95	-	127	DEL

Připojení tiskárny DZM 180

Využitý kanál - GPIO např. skupina Ø (konektor K3)



RAK-
- B2Ø -
- A2Ø -
- A22 -
- A19 -
na + (A23, 24, 25)
(B23 24, 25)

Naprogramování kanálu 4 : CONTROL 4, 3; 168, 13

DZM 180 požaduje negovaná vstupní data !

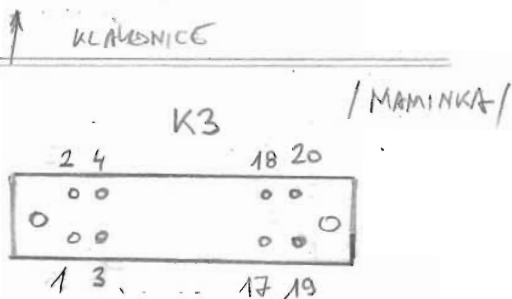
Úprava interpreteru pro výstup dat na DZM.

A\$ = '060C21117F11CD207E12231305C2087FC9DD4EE608C6CD20792FD34CC9" CODE A\$

Použití příkazů : LIST #403; výpis programu
OUTPUT 403; výraz (výstup dat)

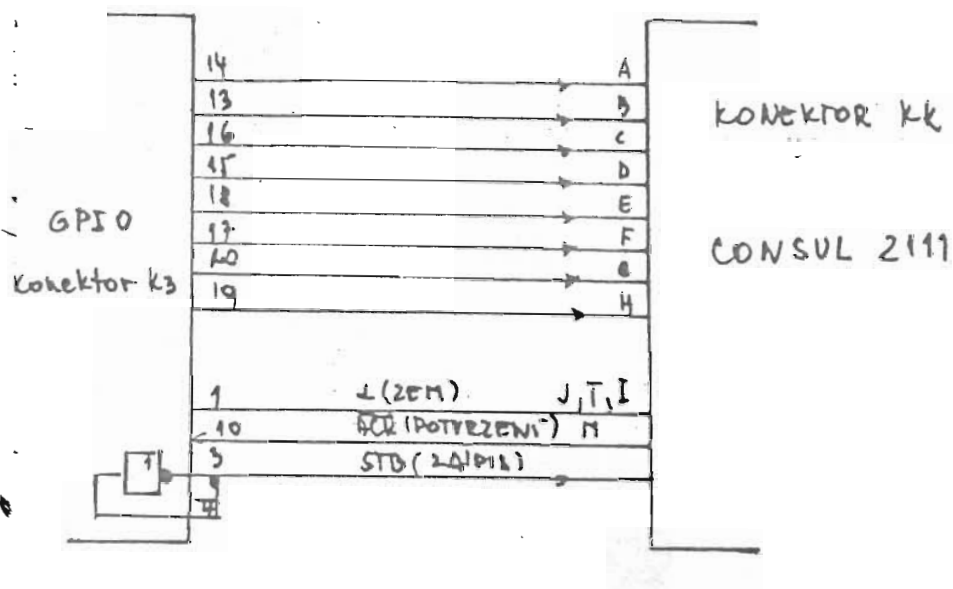
Kontrola připravenosti tiskárny :

S = BIT STATUS 4, 2, 5



Připojení tiskárny CONSUL 2111 - 3

Využitý kanál GPIO např. skupina Ø (konektor K3)



Neprogramování kanálu 4 - skupina Ø

```
CONTROL 4, 3; 160, 13
```

Výstup znaků :

- výpis programu :

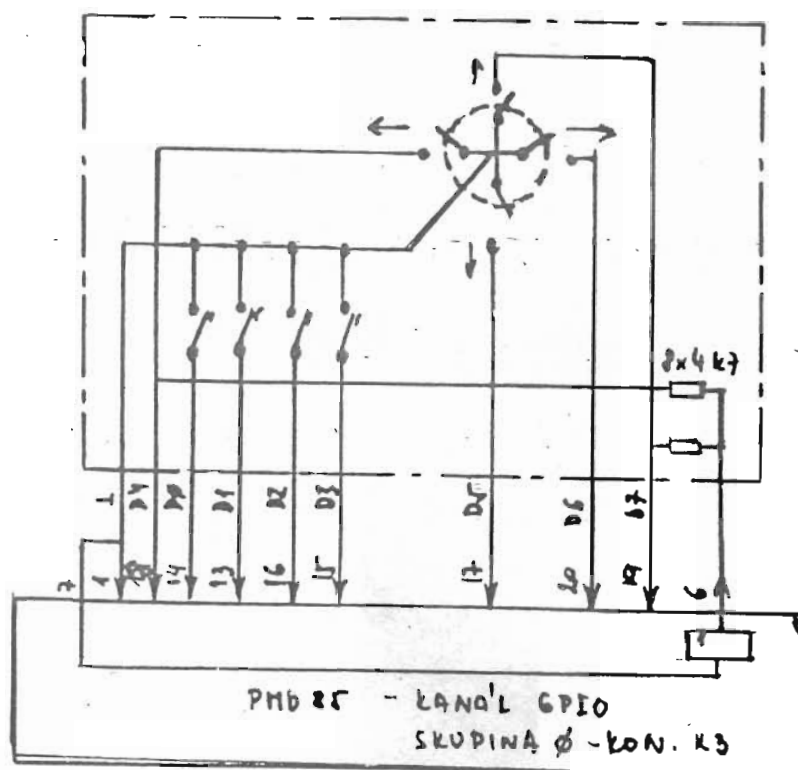
```
LIST # 403;
```
 - výstup znaků :

```
OUTPUT 403; výraz
```
- např.:

```
OUTPUT 403; "TISKÁRNA CONSUL"
```
- ```
OUTPUT 403; 5 * 2/2
```
- ```
OUTPUT 403; TAB(10); A$; TAB (40); B$
```

Poznámka : Počet znaků pro výpis na tiskárnu (display) `OUTPUT(PRINT)` můžeme upravit příkazem

```
POKE 1927, počet znaků
```


Připojení JOYSTICKu

Naprogramování kanálu 4 :

CONTROL 4, 3; 144

Zesnímání stavu JOYSTICKu :

A = STATUS 4, Ø

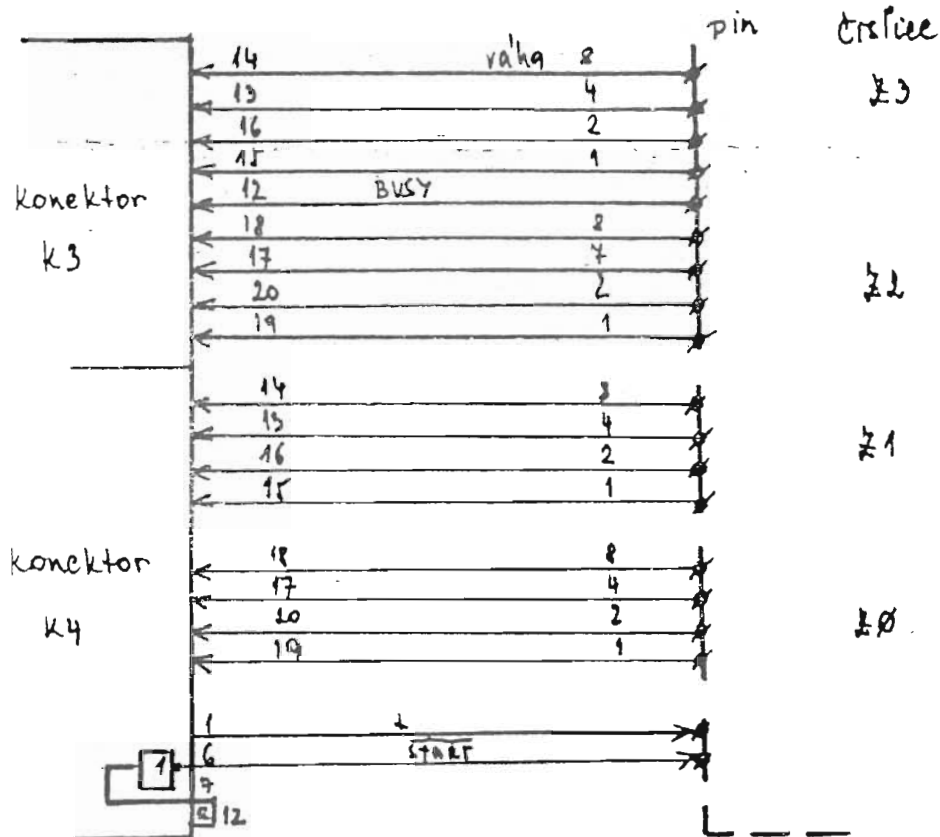
V nezatlačeném stavu proměnná A nabývá hodnoty 255.

Pro testování příslušného kontaktu můžeme využít funkce :

A = BIT STATUS 4, Ø, (číslo bitu)

Připojení příznakového analyzátoru BM 578

Využít celý kanál GPIO.

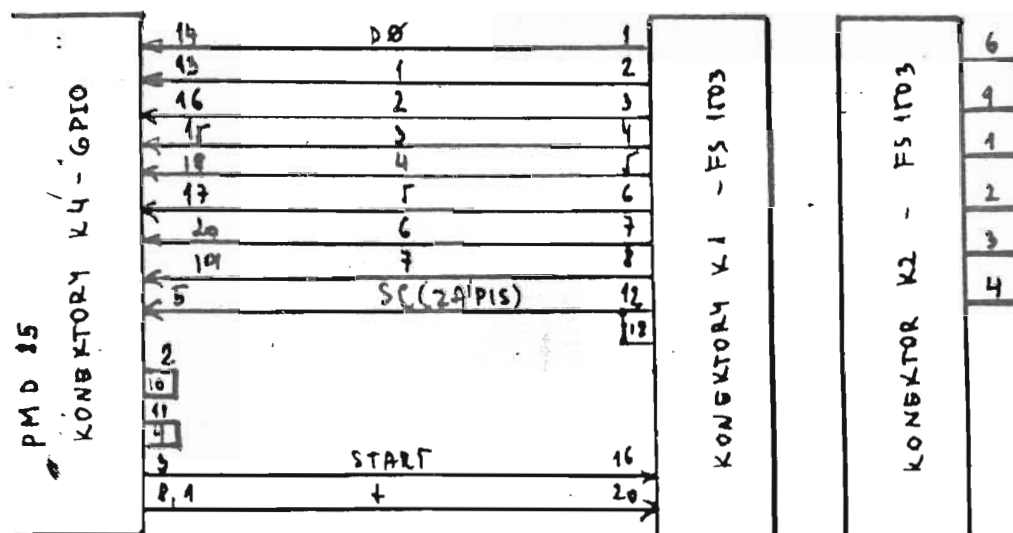


Naprogramování kanálu : CONTROL 4, 3; 154

Je třeba vytvořit snímací rutinu pro příjem čtyř znaků (Z0 - Z3) z přirozených čísel 0 - 9 a šest znaků (A, H, U, F, C, P) ke standardnímu kódu ASCII. Tuto rutinu umístit pod uživatelské volání.

např. ENTER 406; AŽ.

Využit kanál 4 - GPIO - skupina I (konektor K4)



CONTROL 4, 3; 134

START-DYNAMICKÉ
OVLÁDÁNÍ
REGIM - JEDNOVSTUPOVÝ

ENTER 404; AS

Vstupní data generovaná snímačem pásky FS 1503 musí být ukončena znaky CRLF (ØDØA) a jejich počet nesmí překročit 250 znaků. Pro delší bloky je třeba změnit ukazatel vstupního bufferu v místech :

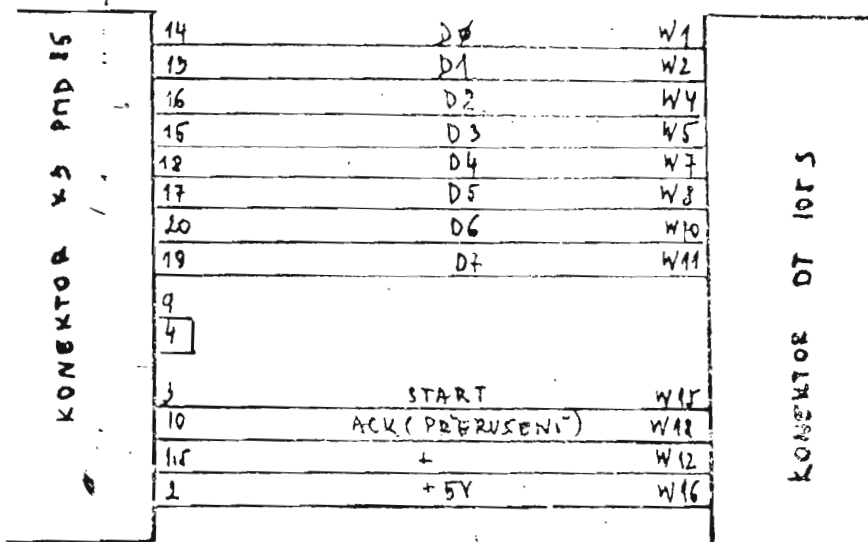
POKE 8723 , nová adresa

POKE 8723, 95

se přemodifikuje vstupní buffer do oblasti 5F00 pro příkazy
INPUT, ENTER.

Připojení děrovače pásky DT 105 S

Využít kanál 4 - GPIO - skupina 0 (konektor K3)



Nastavení kanálu :

CONTROL 4, 3; 160, 13

Výstup znaků :

LIST# 403;

OUTPUT 403; výraz

např.:

OUTPUT 403; "DĚROVAČ"

OUTPUT 403; 5 * SIN(12), "TESLA"

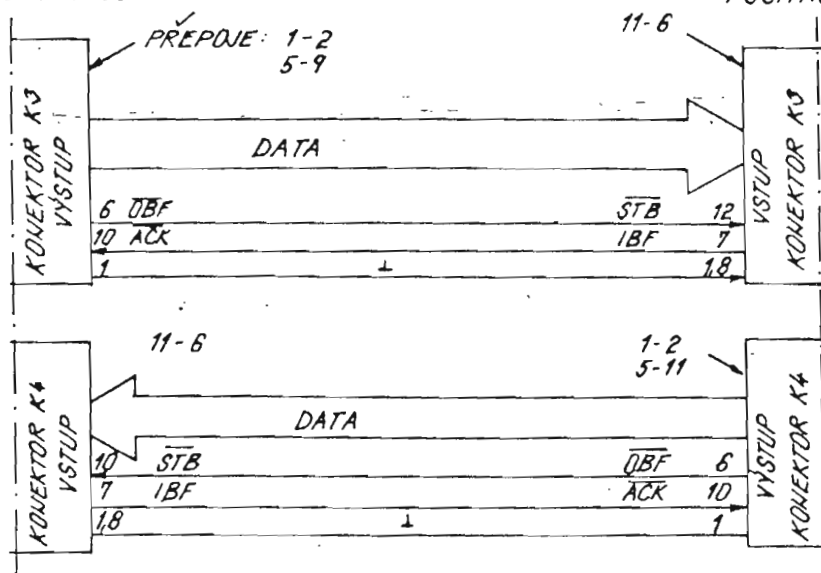
Poznámka : V případě většího bloku dat vysílaných na DT 105 je po každém 64. znaku vsunut znak CR a znak LF. Zvýšení počtu znaků v daném bloku se uskuteční příkazem :

POKE 1927, počet znaků

VYUŽITÝ KANÁL 4 GPIO, REŽIM 1

POČÍTAČ PMD 85

POČÍTAČ PMD 85



HLAVNÍ POČÍTAČ (MASTER)

PODPŮRNÝ POČÍTAČ (SLAVE)

NASTAVENÍ : CONTROL 4,3, 166, 13

CONTROL 4,3; 180,5

VYSÍLÁNÍ : OUTPUT 403, A\$PŘIJÍMÁNÍ : ENTER 403, A\$PŘIJÍMÁNÍ : ENTER 404, B\$VYSÍLÁNÍ : OUTPUT 404, B\$

- POZNÁMKY
- PODPŮRNÝ POČÍTAČ JE V ZÁKLADNÍM STAVU ENTER
 - NEJÍ DOVOLENO POUŽÍT ZAVEDENÍ CRLF (STŘEDNÍK) VE VÝSTUPNÍM PŘÍKAZU OUTPUT
 - STAV POČÍTAČŮ MUŽEME TESTOVAT FUNKCÍ STATUS