

JEDNODESKOVÝ MIKROPOČÍTAČ BOB-85

Ing. Josef Kratochvíl

V AR, ST i na stránkách zahraničních časopisů se v posledních letech objevuje řada publikací o mikroelektronice. Také tento článek by měl přinést některé zajímavé poznatky z tohoto proudu se rozvíjejícího oboru. Námětem článku je popis a stavební návod nejen pro jednodeskový mikropočítač BOB-85 s mikroprocesorem 8085, ale i pro celou sestavu potřebnou pro komunikaci s mikropočítačem – klávesnici, displej a napojení na páskový nebo cívkový magnetofon. Na závěr jsou uvedeny příklady použití mikropočítače.

1. Úvod

Stavbou mikropočítače jsem chtěl získat praktické zkušenosti s technicky novým řešením v době, kdy čs. obvody pro stavbu mikropočítače nebyly ještě na trhu. Mikropočítač slouží jako univerzální řídicí jednotka pro různá technologická zařízení. Celá sestava je tvořena vlastním mikropočítačem, deskou klávesnice a displeje (obr. 1 na 2. str. obálky), která umožňuje komunikaci uživatele s počítačem, a magnetofonem s příslušným obvodem, který slouží jako vnější trvalá paměť. Různá zařízení, která může počítač řídit, jsou připojitelná přes konektor, který tvoří tzv. multibus. Kromě zařízení je možno připojit přes multibus vnější paměť až do kapacity 60 kB.

Vlastní mikropočítač (obr. 2 na 2. str. obálky) obsahuje paměť PROM 1,5 kB s řídicím programem MONITOR v 6 obvodech 74S287 (0,75 kB, centrální jednotka s mikroprocesorem 8085, paměť RAM (1 kB) pro programy uživatele tvořenou 8 obvody K565RU2 (čs. ekvivalent je MHB 2102 A), 6 vstupně výstupních portů a skupinu TTL obvodů pro posílení a negaci sběrnice, dekódování adres pamětí a portů, oddělení vývodů mikroprocesoru apod.

Deska klávesnice a displeje je na obr. 3. (na 2. str. obálky). Připojuje se k mikropočítači přes konektor se 46 vývody. Na desce je umístěna klávesnice zhotovená z mikrosplínů. Signály z mikrosplínů jsou zpracovány dekodérem. Displej tvoří 6 sedmisedimentových zobrazovacích prvků LQ410. Provoz displeje je statický, kódování čísel pro displej je řešeno softwarově a není tedy třeba používat převodníky.

Interfejs pro magnetofon je tvořen dvěma operačními zesilovači a jedním monostabilním klopným obvodem TTL. Obsluha je řešena opět softwarově. Na typu použitého magnetofonu nezáleží, pouze je důležitá konstantní úroveň nahrávacího a přehrávacího signálu.

Dále popsané obvody a části článku předpokládají určité znalosti alespoň základních pojmů z mikropočítačové techniky a základů programování v jazyku symbolických adres Asembleru 8080/8085. Tyto základní pojmy je možno studovat v literatuře [2], [3], [7], [9], [10].

2. Volba mikroprocesoru

Původním záměrem byla stavba mikropočítače s mikroprocesorem 8080 a s ostatními obvody, které jsou v ČSSR vyráběny. Jedná se zejména o hodinový obvod 8224, stavový registr s budičem datové sběrnice 8228, rychlý dekodér 3205, budiče sběrnice s třístavovými výstupy 3212, 3216, 3226, obvod prioritního přerušování 8214, interfejsové obvody 8251, 8255 a další.

V době, kdy jsem mikropočítač stavěl, bylo obtížné uveste integrované obvody získat. Zejména z tohoto důvodu jsem sledoval možnost nákupu jiného typu mikroprocesoru a dalších obvodů. Jako nejvýhodnější se ukázal mikroprocesor 8085; má stoprocentně slučitelný software s mikroprocesorem 8080, tzn. že programy pro mikroprocesor 8080 se dají bez úprav použít i pro mikroprocesor 8085. Další předností je, že mikroprocesor 8085 v sobě zahrnuje většinu výše uvedených obvodů (obr. 4). Tzn. že celá centrální jednotka je řešena jediným IO. Tím se

podstatně zjednodušil celý návrh mikropočítače. Z tabulky 1. je zřejmé, že stavba mikropočítače s mikroprocesorem 8085 je podstatně jednodušší a vyžaduje velmi málo součástek (v [8] se uvádí 3 IO s pamětmi ROM 1 kB a RAM 256 bajtů).

Velmi výhodné, zejména pro jednodeskový mikropočítač, je i napájení 5 V. Není vyžadován žádný složitější nebo speciální zdroj (možnost napájení z baterie). Sériový vstup a výstup se velmi osvědčil při připojení magnetofonu.

3. Výběr dalších obvodů mikropočítače

Centrální jednotka mikropočítače (CPU) tvoří pouze jeden mikroprocesor 8085 s několika TTL obvody pro posílení a oddělení sběrnice a vývodů mikroprocesoru. Další velmi důležitou částí mikropočítače je paměť. Pevná paměť ROM byla navržena s kapacitou 1,5 kB. Zatím zpracovaný řídicí program Monitor využívá tuto kapacitu pouze z poloviny. Monitor je pevně naprogramován v 6 IO typu 74S287 s formátem 256×4 bity. Kromě pevné paměti má mikropočítač statickou paměť RAM 1 kB. Paměť slouží pro uložení uživatelského programu. Je tvořena 8 IO ze SSSR K565RU2, které jsou ekvivalentní čs. obvodům MHB 2102A. Jejich formát je 1024×1 bit. Uvedená kapacita obou pamětí se ukázala jako dostačující pro všechny použité aplikace. V případě potřeby je možno připojit vnější paměť o kapacitě až 60 kB.

Přímou na desce mikropočítače bylo realizováno 6 vstupně výstupních portů, které umožní napojení 6 periférních zařízení. Porty jsou realizovány IO typu MH 3216. Kromě těchto obvodů jsou na desce obvody pro adresování, převážně dekodéry MH7442.

Důležitou částí mikropočítače je sběrnice multibus. Na tuto sběrnici jsou vyvedeny všechny důležité signály. Jsou to zejména negovaná adresová, datová a řídicí sběrnice, napájení, vstupy pro přerušování (interrupt), odpojení sběrnice HOLD, činnost mikroprocesoru READY (WAIT), počáteční nulování RESET, sériový vstup a výstup a další. Multibus má signály napojeny pomocí třístavových oddělovacích budičů nebo výstupů s otevřeným kolektorem, nebo se jedná o vstupy. To umožňuje připojení téměř neomezeného počtu dalších desek (pamětí, periférií).

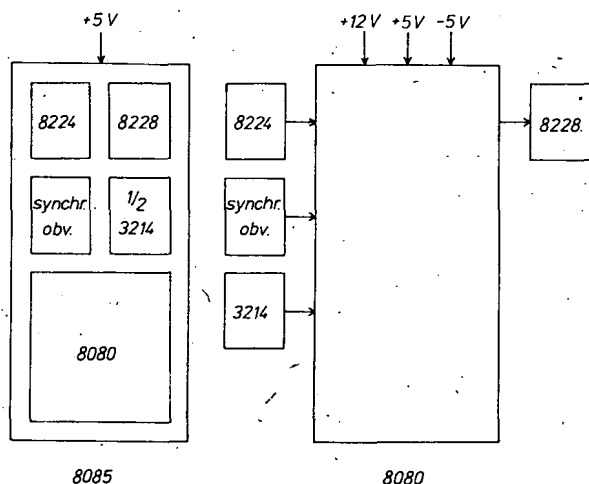
4. Popis činnosti mikropočítače

4.1. Mikroprocesor 8085

4.1.1 Funkční schéma

Základní vnitřní uspořádání mikroprocesoru 8085 je na obr. 5. Jednotlivé části budou stručně popsány v souladu s vnitřním uspořádáním. Jsou to:

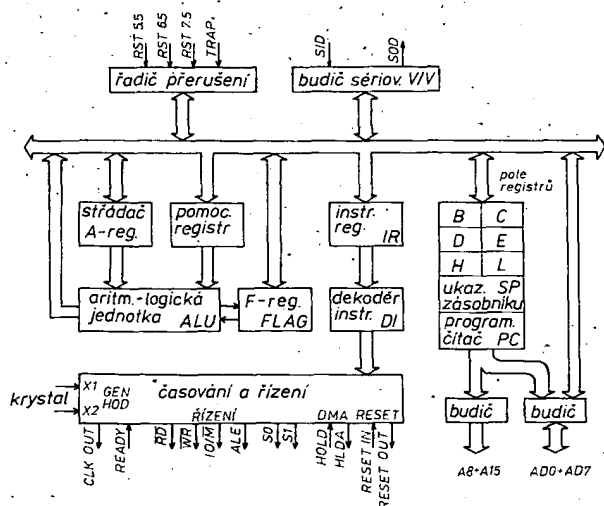
Střádač (A-registr, akumulátor) – je to základní registr mikroprocesoru,



Obr. 4. Porovnání mikroprocesoru 8085 a 8080

Tab. 1. Srovnání vlastností mikroprocesorů 8080 a 8085

VLASTNOST	8080	8085
hodinový obvod	8224 nutno připojit	uvnitř mikroprocesoru
stavový registr	8228 nutno připojit	
řadič přerušování	8214 nutno připojit	
synchronizace RESET	řešit vně pomocí 8224	
READY, HOLD	řešit vně 8251	+5 V
sériový vstup a výstup	+5 V, +12 V, -5 V	
napájení		
délka instrukčního cyklu	2 μs	1,3 μs



Tab. 2. Příznaky F – registru [2], [7], [9]

bit	ozn.	název přízn.	význam
7	S	znaménka	S = 1, je-li 7. bit A – registru roven 1
6	Z	nuly	Z = 1, je-li výsledek operace roven 0
5	–	–	–
4	AC	pomocný přenos	využívá se pouze při instrukci DAA
3	–	–	–
2	P	parity	P = 1, je-li po operaci v A – registru sudá parita
1	–	–	–
0	CY	přenos	CY = 1, nastane-li při operaci přenos.

Tab. 4. Přerušeni [4], [8]

ozn. vývodu mikroprocesoru	priorita	skok na adresu
RST 5.5	nejnižší	002CH
RST 6.5	↓	0034H
RST 7.5	↓	003CH
TRAP	nejvyšší	0024H

Obr. 5. Vnitřní uspořádání mikroprocesoru 8085

Tab. 3. Hexadecimální kód

dek.	H2	H1	hex.	dek.	H2	H1	hex.	dek.	H3	H2	H1	hex.
0		0	0	10		1010	A	32		10	0000	20
1		1	1	11		1011	B			10	1111	2F
2		10	2	12		1100	C	47		11	0000	30
3		11	3	13		1101	D	48				
4		100	4	14		1110	E					
5		101	5	15		1111	F	255		1111	1111	FF
6		110	6	16	1	0000	10	256	1	0000	0000	100
7		111	7	17	1	0001	11					
8		1000	8					4095	1111	1111	1111	FFF
9		1001	9	31	1	1111	1F					

Tab. 5. Význam vývodů S1, S2

S1	S2	význam
0	0	stav HALT (zastavení po instrukci HLT)
0	1	probíhá operační cyklus zápis (WRITE)
1	0	probíhá operační cyklus čtení (READ)
1	1	probíhá první operační cyklus vyvednutí instrukce

přes něj se provádějí aritmetické a logické operace a do něho se také ukládají výsledky operací.

Registry B, C, D, E, H, L – jsou registry pro všeobecné použití, mohou pracovat i ve dvojicích B C, D E, H L.

Pomocný registr – pomáhá při výpočtech, z uživatelského hlediska není důležitý.

Aritmeticko-logická jednotka (ALU) – provádí výpočty podle typu aritmetické nebo logické instrukce.

Registr F – registr příznaků, které se nastavují po výpočtené aritmetické nebo logické operaci, příznaky se nastavují podle tabulky 2. Nastavené příznaky dále využívají podmíněné instrukce (podmíněné skoky, podprogramy a návraty z podprogramů).

Instrukční registr (IR) – přijme instrukci z vnějšku (mimo mikroprocesor).

Dekodér instrukcí (DI) – obsahuje paměť ROM, která dekoduje instrukci a dekodéry, které řídí provádění mikroinstrukcí (například podle [10], [9]).

Programový čítač (PC) – čítač, který vede program po jednotlivých adresách. Není-li proveden skok, vede směrem k vyšší adrese. Pracuje v tzv. hexadecimálním kódu (viz tabulka 3).

Ukazatel zásobníku (SP) – speciální registr, který adresuje část uživatelské paměti tzv. zásobník (zde se ukládají instrukce, data nebo adresy při určitých instrukcích – blíže [2], [3], [7], [9]). Změna adresy v SP je při některých instrukcích prováděna automaticky, nebo ji může provést uživatel.

Budiče – jednosměrné nebo obousměrné brány pro adresové a datové sběrnice (skupina vodičů po nichž se vedou adresy a instrukce nebo data, jsou řešeny jako třístavové [7], [9], [10]).

Radič přerušeni – určuje prioritu (přednost) přerušeni a toto přerušeni také

uskuteční a to tak, že se provede skok na adresu podle tabulky 4, po ukončení přerušeni se vrátí na místo v programu, které při přerušeni opustil (při přerušeni si uschoval adresu z PC v zásobníku pomocí SP).

4.1.2 Základní pojmy

Instrukční krok (cykl) – doba potřebná k vyvednutí a provedení jedné instrukce. Instrukce může být tvořena 1, 2, 3 bajty (1 bajt = 8 bitů).

Operační krok – jedna nebo více period hodinového cyklu. Počet závisí na typu instrukce.

Hodinový cyklus – čas mezi dvěma hodinovými impulsy.

Vyvednutí instrukce – vždy první operační krok z jednoho instrukčního cyklu. Činnost probíhá tak, že mikroprocesor vyšle po adresové sběrnici adresu paměti z níž si přečte instrukci do IR.

Čtení – (read) z paměti nebo periférie (MEM/R, IO/R) – probíhá stejná činnost jako při vyvednutí instrukce, ale bajt z paměti (M) nebo z periférie (IO*) není předán do IR, ale do registru A, případně do jiného registru (podle typu instrukce).

Zápis (write) do M nebo IO – (MEM/W, IO/W) – probíhá stejná činnost jako při čtení, ale směr toku dat je opačný.

Čekání (WAIT) – je-li M nebo IO pomalá, lze mikroprocesor zastavit signálem WAIT = READY. Po uvolnění WAIT pokračuje mikroprocesor v činnosti.

Přerušeni (interrupt) – činnost je popsána v čl. 4.1.1. K původní činnosti se mikroprocesor vrací instrukcí návrat (RETURN).

Stav HOLD – pomocí signálu HOLD se sběrnice (BUS) adresová, datová i řidiči uvedou do stavu velké impedance (třístavová logika [9], [10]) – chovají se jako kdyby byly

odpojené). Tím je umožněn přímý přístup do M (DMA), přičemž mikroprocesor může uvnitř pracovat dál.

***/IO** – zkratka periférie nebo také integrovaného obvodu (je nutno rozlišovat).

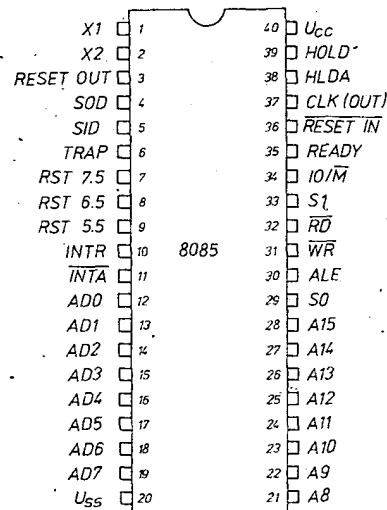
4.1.3 Vývody mikroprocesoru

Uspořádání a označení vývodů mikroprocesoru je na obr. 6. Nyní si blíže vysvětlíme jejich význam.

A8 až A15 – 8 horních bitů adresové sběrnice, třístavové výstupy.

AD0 až AD7 – dolních 8 bitů adresové sběrnice v době prvního hodinového impulsu každé instrukce, jinak 8 bitů datové sběrnice, vývody jsou obousměrné (vstupy, třístavové výstupy).

ALE – signál, který v době prvního hodinového cyklu slouží k oddělení dolních 8 bitů adresové sběrnice do



Obr. 6. Vývody mikroprocesoru 8085

vnějšího adresového registru mikropočítače.

S0, S1 – signály, podle nichž lze usuzovat na chování mikroprocesoru podle tabulky 5 (u BOB-85 nevyužito).

RD – třístavový výstup čtení (aktivní při úrovni L).

WR – třístavový výstup zápis (aktivní při úrovni L).

READY – mikroprocesor pracuje při úrovni H, při L je ve stavu WAIT.

HOLD – při úrovni H je mikroprocesor ve stavu HOLD.

INTR – přerušení (jako signál INT u mikroprocesoru 8080 [7], [9]) – u BOB-85 není využit.

HLDA – výstup potvrzující stav HOLD.

INTA – výstup potvrzující přerušení.

TRAP, RST 5.5, 6.5, 7.5 – vstupy řadiče prioritního přerušení čí. 4.1.1.

RESET IN – vstup pro počáteční nulování

RESET OUT – výstup potvrzující příjem signálu RESET.

X1, X2 – vývody pro připojení krystalu generátoru hodin.

CLK – výstup hodinových impulsů ($f = f_{\text{krystalu}}/2$).

IO/M – třístavový výstup určující aktivaci IO nebo M.

SID – sériový výstup do sedmého bitu A – registru při instrukci RIM.

SOD – sériový vstup ze sedmého bitu A – reg. při instrukci SIM.

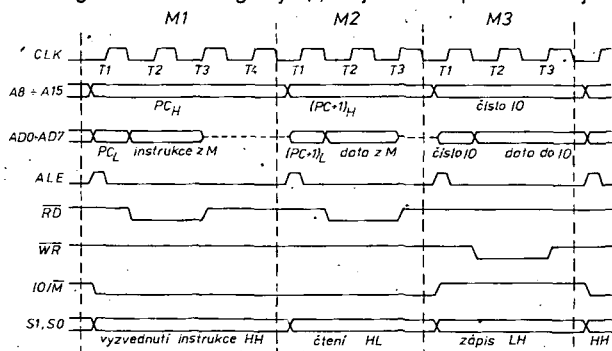
V_{cc} – napájení +5 V.

V_{ss} – zem.

4.1.4 Činnost mikroprocesoru

Činnost mikroprocesoru si vysvětlíme na příkladu časového průběhu signálů. Průběhy hlavních signálů jsou na obr. 7. M1, M2, M3 jsou příklady operačních cyklů, z nichž jsou složeny instrukční cykly. T1, T2, ... jsou jednotlivé hodinové impulsy.

Na horní části adresové sběrnice A8 až A15 je adresa po celou dobu operačního cyklu (při práci s periferií její číslo). Na vývodech AD0 až AD7 je v době T1 dolní adres adresové sběrnice A0 až A7 (při práci s periferií její číslo), později instrukce nebo data D0 až D7. Signál ALE je aktivní také v době T1. Jeho sestupnou hranou zapisujeme dolní část adresy A0 až A7 do vnějšího tzv. adresového registru. Tam je tato adresa držena po celou zbývající dobu operačního cyklu, podobně jako horní část adresy na vývodech A8 až A15. Jako adresový registr může sloužit obvod 3212 nebo dvojice 7475 apod. Signál RD je aktivní v době operačního cyklu čtení, kdy vzestupnou hranou zapisuje data nebo instrukci do vnitřního registru mikroprocesoru. Podobně signál WR zapisuje v době operačního cyklu zápis data nebo instrukci z vnitřního registru mikroprocesoru do M nebo IO. Má-li se uskutečnit zápis nebo čtení z M nebo do M (do IO nebo z IO), je dáno signálem IO/M. Signály S0, S1 jsou



Obr. 7. Příklad časového průběhu signálů mikroprocesoru 8085

vysvětleny v tabulce 5 a mohou být využity jako signály informační, nebo řídící. V době T2 je mikroprocesorem testován vstup READY. Je-li na něm úroveň H, mikroprocesor pracuje dál, při úrovni L přejde do stavu WAIT (činnost mikroprocesoru je zastavena), kde setrvá, dokud se na vstupu READY neobjeví H.

Časové průběhy při stavu HOLD a INTERRUPT nejsou pro činnost mikropočítače BOB-85 podstatné a proto se jimi nebudeme podrobněji zabývat. Blíže je možno se s nimi seznámit např. v literatuře [4], [9]. Signály READY, HOLD a vstupy pro přerušení RST 5.5, RST 6.5, RST 7.5, TRAP jsou luvnitř mikroprocesoru synchronizovány s hodinovými impulsy. Tzn., že tyto signály mohou být přivedeny na vstupy kdykoliv a mikroprocesor si je sám přečte v pro něj výhodné době, zpravidla po dokončení započatého operačního nebo instrukčního cyklu.

Činnost mikroprocesoru lze popsat následovně. Po připojení napájecího napětí +5 V (předpokládáme zapojení mikroprocesoru v počítači např. podle obr. 12 s ošetřenými nevyužitými vstupy) by mikroprocesor pracoval v náhodné nekontrolovatelné činnosti. Proto musí nejdříve přijít signál RESET, který nuluje PC ($PC = 0000H$). Potom probíhá operační cyklus vyvednutí instrukce z paměti ROM. Instrukce je vyvednuta z adresy 0000H (první instrukce řídícího programu Monitor) a předána do IR. Prostřednictvím DI je dekodována a provedena. Současně se inkrementuje PC ($PC + 1$). Další činnost je již řízena podobně další instrukcí.

4.1.5 Stručný přehled instrukcí

Celá problematika programování se většinou učí v několikadenních kurzech programování. Zde v krátkém článku se pouze zmíníme o tom, co je to instrukce, jak se zapisuje do programu, stručně si probereme jejich přehled a ukážeme si krátký program v jazyku symbolických adres Assembleru 8080/8085. Pro studium uvedených programů, zejména programu Monitor, bude zřejmě potřeba hlubší studium např. z literatury [2], [3].

Instrukce je příkaz, který nařizuje mikropočítači, aby vykonal určitou činnost. Jednotlivé instrukce jsou pro každý mikroprocesor pevně stanoveny již při jeho výrobě. Každá instrukce má svůj mnemonický tvar a operační kód. Mnemonický tvar je anglická zkratka, vyjadřující funkční význam povelu, který má mikroprocesor po přijmutí instrukce vykonat. Pomocí mnemonického tvaru zapisujeme program v jazyku symbolických adres. Operační kód je dvojčíslí hexadecimální číslo, které je uloženo do paměti jako program. Mikroprocesor si tedy vlastně čte operační kódy instrukcí a tyto si zapisuje do IR. Ty jsou dekodovány DI a podle nich jsou prováděny jednotlivé

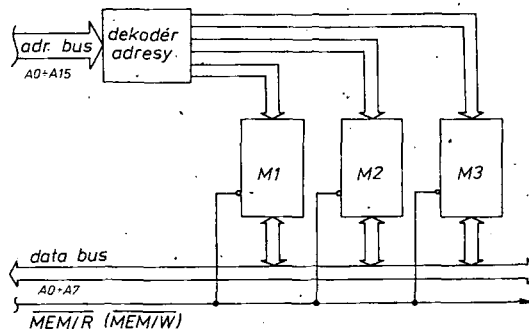
Tab. 6. Konvence pro tabulku 7

označení	význam
reg	registru A, B, C, D, E, F, H, L nebo paměť M
data 8	8-bitové hexadecimální číslo
adr 16	16-bitové hexadecimální adresa
dreg	dvojice registrů BC, DE, HL
zař (kód)	8-bitová adresa zařízení číslo zakódované v operačním kódu instrukce
→	přesun
M (adr 16)	obsah M na adrese 16
M (HL)	obsah M na adrese uložené ve dvojici registrů
F	registru příznaků
CY, Z, S, P, AC	příznaky dle tabulky 2
↔	záměna
—	negace
→	rotace vpravo o 1 bit
←	rotace vlevo o 1 bit
A, B, C, D, E, H, L	střadač a ostatní registry
PC	programový čítač
SP	ukazatel zásobníku
V/V	periferie
zásob	zásobník
^	logický součin AND
✓	logický součet OR

mikroinstrukce. Při zápisu programu do paměti musíme zapisovat jednotlivé operační kódy. Překlad z mnemonického tvaru do operačního kódu lze provést vhodným překladačem nebo ručně viz tab. 8. Pro stručný přehled významu jednotlivých instrukcí (tab. 7) si zavedeme určité konvence (tab. 6).

Program v jazyku Assembler 8080/8085 se zapisuje do vhodného formuláře. Jako příklad si uvedeme krátký program pro generování zpoždění (tab. 9).

Instrukce mohou být jedno, dvoj, nebo tříslabičné, tzn. že jejich délka je 2, 4, nebo 6 hexadecimálních číslic (8, 16, 24 bitů). Do formuláře se zapisuje vždy jedna instrukce na jeden řádek. Adresou programu rozumíme adresu paměti, na které je uložen program (2000H – 200DH). Operační kód se stanovuje tak, že první slabiku instrukce určíme podle tabulky 8. U dvojslabičných instrukcí se píše druhá slabika (zpravidla operand instrukce) do druhého sloupce pro operační kód. U tříslabičných instrukcí se píše vyšší část operandu nebo adresy (8 významnějších bitů) do třetího sloupce, nižší do druhého sloupce. V programu jsou použita 2 návěští – CEKEJ a CEK 1. Návěští slouží k označení určitých míst v programu, na něž se chceme nějakou skokovou instrukcí dostat. Hodnota návěští je dána adresou, na které je umístěno (CEKEJ = 2000H, CEK 1 = 2005H). Chceme-li provést skok na určité návěští, je třeba uvést jeho adresu (v tab. 9 instrukce



Obr. 8. Zapojení paměti M v mikropočítači

Tab. 7. Význam instrukcí mikroprocesorů 8080/8085 (zleva instrukce, význam)

Přesuny:	
MOV reg, reg	reg ← reg
MVI reg, data8	reg ← data8
LDA adr16	M(adr16) → A
LDXA dreg	M(B, C(D, E)) → A
LHLD adr16	M(adr16) → M, (adr16 + 1) → H
LXI dreg, data16	data16 → dreg
STA adr16	A → M(adr16)
STAX dreg	A → M(B, C(D, E))
SHLD adr16	L → M(adr16), H → M(adr16 + 1)
XCHG	H, L ↔ D, E
Součet:	
ADD reg	A + reg → A, mění F
ADI data8	A + data8 → A, mění F
ADC reg	A + CY + reg → A, mění F
ACI data8	A + CY + data8 → A, mění F
Rozdíl:	
SUB reg	A - reg → A, mění F
SUI data8	A - data8 → A, mění F
SBB reg	A - reg - CY → A, mění F
SBI data8	A - data8 - CY → A, mění F
Inkrement. a dekrement.:	
INR reg	reg + 1, mění F kromě CY
DCR reg	reg - 1, mění F kromě CY
INX dreg	dreg + 1
DCX dreg	dreg - 1
Převod na dekadický tvar a součet dreg	
DAA	hex. tvar → dekadický tvar
DAD dreg	dreg + H, L → H, L
Logické operace:	
ANA reg	reg ∧ A → A, mění F CY, AC = 0
ANI data8	data8 ∧ A → A, mění F, CY, AC = 0
ORA reg	reg ∨ A → A, mění F, CY, AC = 0
ORI data8	data8 ∨ A → A, mění F, CY, AC = 0
XRA reg	reg ⊕ A → A, mění F, CY, AC = 0
XRI data8	data8 ⊕ A → A, mění F, CY, AC = 0
Komparace:	
CMP reg	A = reg, nemění A, Z = 1, CY = 0 je-li A ≥ reg
CPI data8	A = data8, nemění A, Z = 1, CY = 0 je-li A ≥ data8
Negace a změny CY:	
CMA	A → \bar{A}
CMC	CY → \bar{CY}
STC	CY = 1
Rotace:	
RLC	A ← 1 bit, mění CY
RRC	A → 1 bit, mění CY
RAL	A ← 1 bit s CY, mění CY
RAR	A → 1 bit s CY, mění CY

Skok:	
JMP adr16	adr16 → PC, skok
JC adr16	skok je-li CY = 1
JNC adr16	CY = 0
JZ adr16	Z = 1
JNZ adr16	Z = 0
JM adr16	S = 1
JP adr16	S = 0
JPE adr16	P = 1
JPO adr16	P = 0
Podprogram:	
CALL adr16	SP-2 PC → zásob., adr16 → PC, skok
CC adr16	skok je-li CY = 1
CNC adr16	CY = 0
CZ adr16	Z = 1
CNZ adr16	Z = 0
CM adr16	S = 1
CP adr16	S = 0
CPE adr16	P = 1
CPO adr16	P = 0
Návrat z podprogramu:	
RET	SP + 2, PC ← zásob.,
RC	návrat je-li CY = 1
RNC	CY = 0
RZ	Z = 1
RNZ	Z = 0
RM	S = 1
RP	S = 0
RPE	P = 1
RPO	P = 0
V/V instrukce:	
IN zač	V/V → A
OUT zač	A → V/V
Řídící instrukce:	
EI	povoleno přerušení
DI	zákaz přerušení
HLT	stop
NOP	prázdná instrukce
Přerušení a přímý skok:	
RST (kód)	SP-2, PC → zásob., kód → PC, skok
PCHL	PC, skok
Instrukce pro práci se zásobníkem:	
PUSH dreg	SP-2, dreg → zásob.
POP dreg	SP + 2, zásob. → dreg
SPHL	H, L → SP
XTHL	H, L ↔ zásob.
Sériový vstup/výstup: (pouze 8085)	
RIM	SID → 7. bit A reg.
SIM	7. bit A reg → SOD

C2 05 20 JNZ CEK 1). Ve sloupci instrukce je uveden mnemonický tvar instrukcí. Ve sloupci komentář může být uveden bližší význam jednotlivých instrukcí. Při tvorbě složitějších programů je vhodné kreslit vývojový diagram.

4.2 Paměti

Další velmi důležitou součástí mikropočítače jsou paměti. Rozlišujeme 2 základní typy. Pevná paměť ROM slouží k uložení řídicího programu Monitor (u BOB-85 na adresách 000H – 02FFH), který je trvale zapsán i po vypnutí počítače. Do paměti ROM může být program zapsán již ve výrobě podle požadavků uživatele, nebo si jej může uživatel zapsat do paměti sám pomocí programátoru pevných pamětí (např. čl. 12.5) Paměti ROM, které si může programovat sám uživatel se označují PROM nebo EPROM. Program jednou zapsaný do paměti PROM nelze již vymazat, paměti EPROM vymazat jdou. Bližší informace o těchto pamětech jsou např. v literatuře [9], [11].

Dalším typem paměti v mikropočítači je paměť typu čti – piš, které se v literatuře

uvádějí jako paměti RAM. Tato paměť slouží k uložení uživatelského programu a k uložení zásobníku uživatelského a řídicího programu.

Způsob činnosti paměti v mikropočítači si vysvětlíme podle obr. 8. Podle typu M je v souhlase s obr. 7 nejdříve přivedena adresa (přes dekodér nebo bez podle způsobu adresování příslušného typu M). Ta adresuje příslušný bajt v M. Při čtení je aktivován signál MEM/R, tím se na datové sběrnici objeví obsah M z naadresovaného místa v M. Vzestupnou hranou signálu MEM/R si mikropočítač uloží obsah M do některého svého registru (podle typu instrukce). Při zápisu (pouze paměti RAM) jsou přivedena data, která chceme zapsat do M na datovou sběrnici. Dále je aktivován signál MEM/W, jehož vzestupnou hranou jsou data zapasána do M na adresované místo (původní obsah je přepsán).

4.3 Periferie

Periférii (IO) rozumíme zařízení, které je řízené mikropočítačem. Může to být například snímač děrné pásky, tester inte-

grovaných obvodů, programátor pevných pamětí, různá signalizační a technologická zařízení apod. Do IO lze data psát nebo lze z IO data číst podobně jako u M. Rozdíl je pouze v adresování. IO je adresována osmibitovou adresou (číslo zařízení) uloženou přímo ve dvojslabičné instrukci. Jedná se o instrukce IN zač. (pro čtení z IO) a OUT zač. (pro zápis do IO).

V mikropočítači BOB-85 jsou jako periférie napojeny klávesnice a displej. IO jsou napojeny přes t. zv. porty. IO se aktivují signály IO/R, IO/W stejně, jako aktivace M signály MEM/R, MEM/W. Těmto signálům říkáme řídicí a mluvíme o řídicí sběrnici. Signály se získávají jednoduchou logickou sítí ze signálů mikroprocesoru RD, WR, IO/M (viz podrobné schéma mikropočítače BOB-85 obr. 14).

Na základě dosud řečeného si již můžeme nakreslit jednoduché blokové schéma mikropočítače (obr. 9). Mikropočítač je tvořen centrální jednotkou CPU, pamětmi ROM a RAM a perifériemi IO. Další doplňující obvody jsou vlastně součástí těchto větších skupin. Tyto skupiny jsou propojeny sběrnicemi (adresová, datová a řídicí).

Tab. 8. Abecední seznam instrukcí a jejich operační kódy (zleva instrukce, operační kód, počet slabik) a ... A16, d ... D8, w ... D16

ACI	d	CE	2	CNZ	a	C4	3	JMP	a	C3	3	MOV	D, C	51	1	MVI	E, d	1E	2	RST	4	E7	1
ADC	A	8F	1	CP	a	F4	3	JNC	a	D2	3		D, D	52	1		H, d	26	2		5	EF	1
	B	88	1	CPE	a	EC	3	JNZ	a	C2	3		D, E	53	1		L, d	2E	2		6	F7	1
	C	89	1	CPI	d	FE	2	JP	a	F2	3		D, H	54	1		M, d	36	2		7	FF	1
	D	8A	1	CPO	a	E4	3	JPE	a	EA	3		D, L	55	1	NOP		00	1	RZ		C8	1
	E	8B	1	CZ	a	CC	3	JPO	a	E2	3		D, M	56	1	ORA	A	B7	1	SBB	A	9F	1
	H	8C	1	DAA		27	1	JZ	a	CA	3		E, A	5F	1		B	B0	1		B	98	1
	L	8D	1	DAD	B	09	1	LDA	a	3A	3		E, B	58	1		C	B1	1		C	99	1
	M	8E	1		D	19	1	LDAX	B	0A	1		E, C	59	1		D	B2	1		D	9A	1
ADD	A	87	1		H	29	1		D	1A	1		E, D	5A	1		E	B3	1		E	9B	1
	B	80	1		SP	39	1	LHLD	a	2A	3		E, E	5B	1		H	B4	1		H	9C	1
	C	81	1	DCR	A	3D	1	LXI	B, w	01	3		E, H	5C	1		L	B5	1		L	9D	1
	D	82	1		B	05	1		D, w	11	3		E, L	5D	1		M	B6	1		M	9E	1
	E	83	1		C	OD	1		H, w	21	3		E, M	5E	1	ORI	d	F6	2	SBI	d	DE	2
	H	84	1		D	15	1		SP, w	31	3		H, A	67	1	OUT	d	D3	2	SHLD	a	22	3
	L	85	1		E	1D	1	MOV	A, A	7F	1		H, B	60	1	PCHL		E9	1	SIM		30	1
	M	86	1		H	25	1		A, B	78	1		H, C	61	1	POP	B	C1	1	SPHL		F9	1
ADI	a	C6	2		L	2D	1		A, C	79	1		H, D	62	1		D	D1	1	STA	a	32	3
ANA	A	A7	1		M	35	1		A, D	7A	1		H, E	63	1		H	E1	1	STAX	B	02	1
	B	A0	1	DCX	B	0B	1		A, E	7B	1		H, H	64	1	PSW		F1	1		D	12	1
	C	A1	1		D	1B	1		A, H	7C	1		H, L	65	1	PUSH	B	C5	1	STC		*37	1
	D	A2	1		H	2B	1		A, L	7D	1		H, M	66	1		D	D5	1	SUB	A	97	1
	E	A3	1		SP	3B	1		A, M	7E	1		L, A	6F	1		H	E5	1		B	90	1
	H	A4	1	DI		F3	1		B, A	47	1		L, B	68	1		PSW	F5	1		C	91	1
	L	A5	1	EI		FB	1		B, B	40	1		L, C	69	1	RAL		17	1		D	92	1
	M	A6	1	HLT		76	1		B, C	41	1		L, D	6A	1	RAR		1F	1		E	93	1
ANI	d	E6	2	IN	d	DB	2		B, D	42	1		L, E	6B	1	RC		D8	1		H	94	1
	a																						
CALLa		CD	3	IHR	A	3C	1	BE		43	1		H, L	CC	1	RET		C9	1		L	95	1
CC	a	DC	3		B	04	1		B, H	44	1		L, L	6D	1	RIM		20	1		M	96	1
CM	a	FC	3		C	0C	1		B, L	45	1		E, M	6E	1	RLC		07	1		d	D6	2
CMA		2F	1		D	14	1		B, M	46	1		M, A	77	1	RM		F8	1	XCHG		EB	1
CMC		3F	1		E	1C	1		C, A	4F	1		M, B	70	1	RNC		DO	1	XRA	A	AF	1
CMP	A	BF	1		H	24	1		C, B	48	1		M, C	71	1	RNZ		CO	1		B	A8	1
	B	B8	1		L	2C	1		C, C	49	1		M, D	72	1	RP		FO	1		C	A9	1
	C	B9	1		M	34	1		C, D	4A	1		M, E	73	1	RPE		E8	1		D	AA	1
	D	BA	1	INX	B	03	1		C, E	4B	1		M, H	74	1	RPO		EO	1		E	AB	1
	E	BB	1		D	13	1		C, H	4C	1		M, L	75	1	RRC		OF	1		H	AC	1
	H	BC	1		H	23	1		C, L	4D	1	MVI	A, d	3E	2	RST	0	C7	1		L	AD	1
	L	BD	1		SP	33	1		C, M	4E	1		B, d	06	2		1	CF	1		M	AE	1
	M	BE	1	JC	a	DA	3		D, A	57	1		C, d	0E	2		2	D7	1	XRI	d	EE	2
CNC	a	D4	3	JM	a	FA	3		D, B	50	1		D, d	16	2		3	DF	1	XTHL		E3	1

Tab. 9. Ukázka programu pro zpědnění

adresa	operační kód	návěsti	instrukce	komentář
2000	D5	CEKEJ:	PUSH D	uložení DE do zásobníku
2001	F5		PUSH PSW	uložení AF do zásobníku
2002	1100F0		LXI D, F000H	naplnění DE číslem F000H
2005	1B	CEK1:	DCX D	odečtení 1 od DE
2006	7A		MOV A, D	přesun obsahu D do A
2007	C600		ADI 0H	přičtení 0H k A
2009	C20502		JZN CEK1	SKOK na CEK1 je-li Z = 0
200C	F1		POP PSW	obnovení původního obsahu AF
200D	D1		POP D	obnovení původního obsahu DE

Tab. 10. Napájení IO BOB-85

IO	+5 V	zem
3216	16	8
3226	16	8
7475	5	12
7442	16	8
K565 RU2	10	9
(1902)	10	9
8085	40	20
74S287	16	8
7400	14	7
7404	14	7
7405	14	7
7420	14	7
7430	14	7
7438	14	7

4.4 Multibus

➔ Pro připojení mikropočítače na další M, IO, případně na další mikropočítače je vhodné vyvést vývody adresové, datové i řídicí sběrnice, napájení a další signály na společný konektor. Protože jsou tyto signály převážně řešeny jako třístavové, nebo mají výstupy s otevřeným kolektorem, je možné napojit přes tento konektor téměř libovolný počet M nebo IO (počet je dán schopnostmi použitého mikroprocesoru – 8085 může obsluhovat M o kapacitě 64 kB a 256 IO). Způsob propojení, kterému říkáme multibus, je znázorněn na obr. 10. Kromě vnějších M a IO jsou přes multibus připojeny 2 mikropočítače. Oba mohou obsluhovat vnější M i IO. Takovýmto sestavám říkáme multiprocesorové [9].

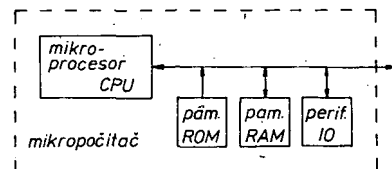
5. Základní sestava mikropočítače BOB-85

Čelá sestava je tvořena mikropočítačem BOB-85, klávesnicí a displejem pro komunikaci uživatele s mikropočítačem

a magnetofonem, který slouží pro trvalé uložení programů a dat. Propojení jednotlivých částí sestavy je na obr. 11. Programy se vkládají do paměti mikropočítače klávesnicí a kontrolují se na displeji. Po odhlášení nebo při přerušení práce se nahrají na magnetofon, odkud je možno je kdykoliv vyvolat.

Mikropočítač se osvědčil například při testování integrovaných obvodů, kdy se do paměti RAM nahraje z magnetofonu testovací program pro test příslušného IO. Přes multibus (K1) se připojí periferie (deska s 13 IO TTL), na níž se do objímky zasune testovaný obvod a program se spustí z klávesnice. Výsledek testu čteme na displeji.

Dále se sestava osvědčila při programování pevných pamětí. Programátor se k mikropočítači připojí přes multibus, z magnetofonu se do paměti RAM nahraje řídicí program pro programátor. Program, který chceme nahrát do pevné paměti, nahrajeme do paměti RAM ze snímáče děrné pásky nebo ručně klávesnicí. Program můžeme ještě klávesnicí a displejem prohlédnout. Řídicí program

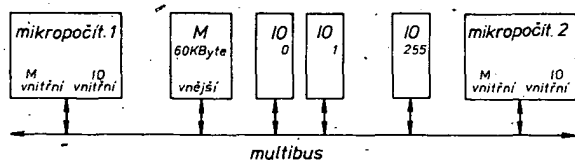


Obr. 9. Jednoduché blokové schéma mikropočítače

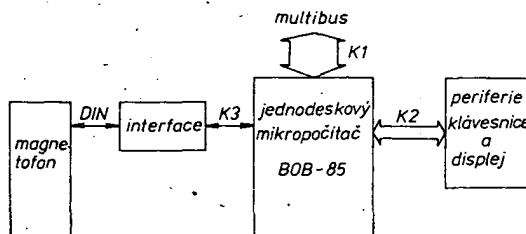
pro vlastní programování spustíme opět z klávesnice.

Zajímavé byly rovněž pokusy s generováním melodie apod. Blíže je o těchto aplikacích pojednáno v kapitole 12.

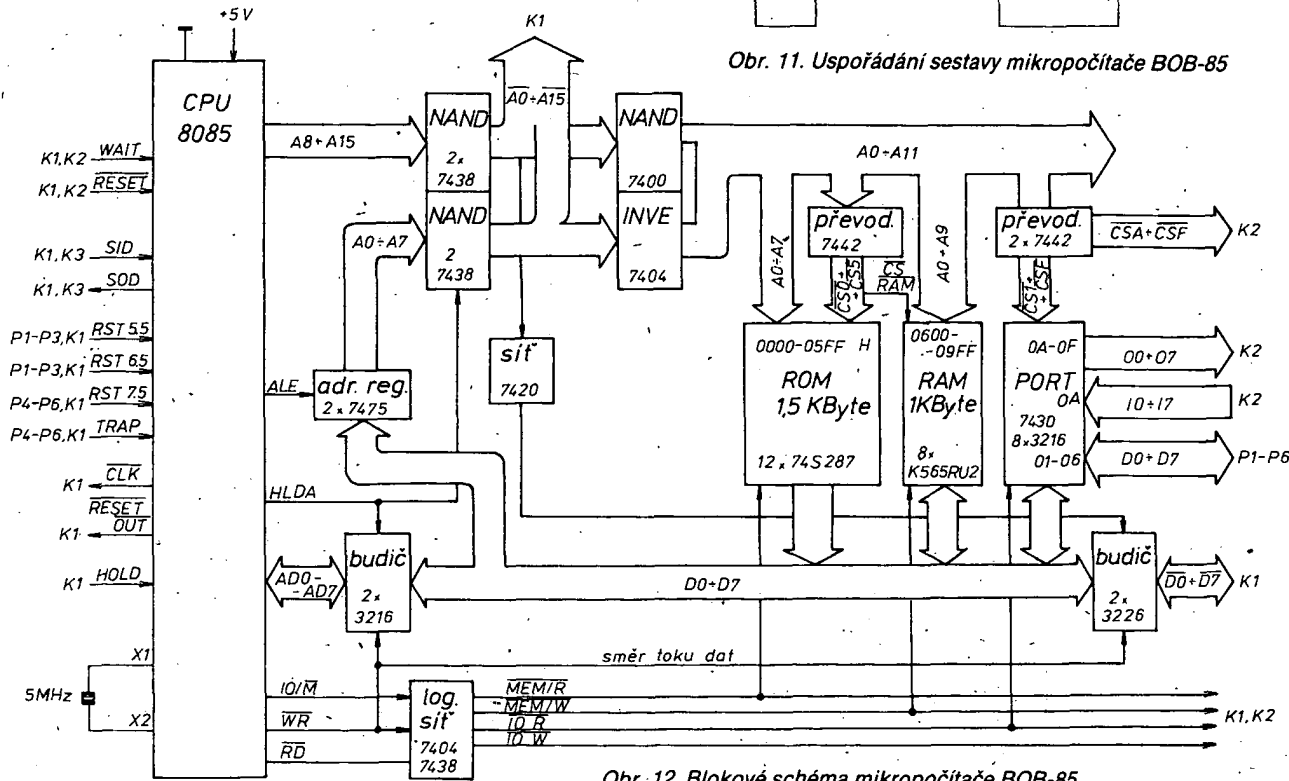
Po prvních zkušenostech se ukazují obrovské možnosti této mikropočítačové sestavy. Lze říci, že pouhou záměnou programu (nahrání z magnetofonu trvá asi 20 sekund – 256 bajtů) může počítač pomocí jednoduché desky s interfejsem řídit různá zařízení. Zřejmě není



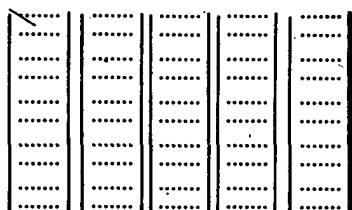
Obr. 10. Možnosti zapojení multibusu



Obr. 11. Uspořádání sestavy mikropočítače BOB-85



Obr. 12. Blokové schéma mikropočítače BOB-85



Obr. 13. Příklad části univerzálního plošného spoje pro IO

daleko doba, kdy se budou i v ČSSR používat jednotné automaty řízené mikropočítači (jedna deska s malým počtem IO). Tyto automaty budou řídit různá zařízení pouhou záměnou jednoho integrovaného obvodu (paměť ROM), nebo záměnou obsahu paměti. Přitom tyto desky nahradí složitá hardwarová řešení s několika sty i tisíci IO. Tím se uspoří nejen na ceně, pracnosti při výrobě a elektrické energii, ale především se zmnohonásobí rychlost vývoje. Práce vývojových pracovníků se přesune z oblasti hardwarové do oblasti softwarové.

Mikropočítač BOB-85 je klasickou ukázkou, jak lze spojit na počtu součástek (tím i spotřebě elektrické energie, váze zařízení, objemu, ceně apod.). Veškerá zařízení jsou napojena co nejjednodušším způsobem a veškeré řízení je řešeno programově. Příkladem je třeba vypuštění dekodéru displeje (dekodování se provádí softwarově – v Monitoru – podprogramem KOD), napojení magnetofonu pouze třemi aktivními součástkami (celá činnost je řízena programem Monitoru MGF), napojení snímače pásky pomocí 4 integrova-

ných obvodů (program SP nahraný na magnetofonu) a další.

Po zkušenostech s užíváním mikropočítače budou ke konci článku uvedena ještě další podstatná zjednodušení, která budou zajímavá zejména pro amatéry, neboť se projeví na dalším snížení počtu součástek a tedy i ceně. Že se tak nestalo již v době návrhu mikropočítače, je způsobeno tím, že počítač měl být na multibusu ekvivalentní mikropočítači s mikropočítačem 8080, který je jeho předchůdcem. Proto jsou např. na výstupu adresové a datové sběrnice negovány, je použit 80-ti vývodový konektor pro multibus apod. Zkušenosti ukazují, že prakticky celá sestava (mikropočítač, klávesnice a displej, interface magnetofonu) by se vešla na jednu desku plošného spoje (viz kapitola 13).

6: Mikropočítač BOB-85

Blokové schéma mikropočítače je na obr. 12. K1 je konektor multibusu, K2 konektor pro napojení klávesnice a displeje, K3 konektor pro napojení interface magnetofonu. Vývody mikropočítače jsou vyvedeny tak, jak byly popsány v čl. 4.1.3. Vyšší bity (A8 až A15) adresové sběrnice jsou negovány a posílány obvody NAND (2x7438). Odpojujány mohou být signálem HLDA ve stavu HOLD. Nižší bity (A0 až A7) adresové sběrnice jsou drženy adresovým registrem (2x7438), který je nahrazen signálem ALE. Dále jsou zapojeny stejně jako vyšší část adresové sběrnice přes NAND (2x7438). Takto upravená adresová sběrnice je vyvedena na multibus K1. Dále je upravována inver-

tory (7404), obvody NAND (7400) a převodníky (7442) pro adresování M a IO.

Obousměrná datová sběrnice pracuje uvnitř mikropočítače jako pozitivní, mimo jako negativní (2x3226). Směr pohybu dat je určen signálem WR, který řídí budiče 2x3226, 2x3226. Také datová sběrnice může být odpojena (přivedena do stavu velké impedance) signálem HLDA. Budič pro výstup negované datové sběrnice (2x3226) je uzavřen při adrese menší než 1000H. Uzavření zajišťuje (1/2 7420).

Řídící sběrnice se signály MEM/R, MEM/W, IO/R, IO/W je získána hradlovou sítí (7404, 7438) ze signálů mikropočítače IO/M, WR, RD. Také tato sběrnice je ve stavu HOLD odpojena (výstupy s otevřeným kolektorem jsou ve stavu H), to je zajištěno signály WR, RD ve stavu HOLD. Řídící sběrnice řídí směr toku dat mezi mikropočítačem a M nebo IO.

Jako pevná paměť je použito 12 obvodů PROM (74S287). Paměť je přístupná na adresách 0000H až 05FFH. Kapacita této paměti je 1,5 kB. V této paměti je na adresách 0000H až 02FFH uložen řídicí program Monitor (6x74S287). Adresy 0300H až 05FFH nejsou zatím využity, i když objímky pro IO jsou zapojeny. Zde by měl být později uložen řídicí program pro různá technologická zařízení. Jako paměť čti – piš je použito 8 IO typu K565RU2 (čs. ekvivalent je MHB 2102A). Jsou to statické paměti RAM, které dohromady tvoří uživatelskou paměť o kapacitě 1 kB na adresách 0600H až 09FFH. Obě tyto paměti ROM a RAM jsou zapojeny tak, jak bylo popsáno v čl. 4.2.

Periferie je možno připojit přes multibus (je rovněž možno připojit paměť o ka-

pacitě 60 kB od adresy 1000H až po adresu FFFFH, nebo přes 6 portů o číslech 01H až 06H přes konektory P1 až P6. Napojení klávesnice je řešeno jako vstupní port 0AH přes K2 a displeje jako 6 výstupních portů 0AH až 0FH opět přes K2.

Zdrojem hodinových impulsů je v mikroprocesoru zabudovaný krystalem řízený oscilátor. Krystal se připojuje vně. Byl použit krystal 5 MHz, tzn. že mikroprocesor pracuje na kmitočtu 2,5 MHz (viz čl. 4.1.3).

Pohled na mikropočítač BOB-85 je na 2. straně obálky. Funkční vzorek byl realizován na univerzálním plošném spoji (obr. 13). Diody a pasivní součástky jsou pájeny zespolu. Spoje jsou provedeny vodičem U 0,3. Jednotlivé IO jsou umístěny v objímkách, mikroprocesor se 40. vývody má objímku vyrobenou rozřezáním tří objímek se 14 vývody. Před osazením a oživováním doporučuji důkladně prozkoušet všechny spoje na desce bez IO. Protože se jedná o dynamický provoz, je vyhledání závady velmi obtížné zejména v amatérských podmínkách. Jako příklad závady mohou uvést, že při oživování mikropočítače byla zjištěna záměna dvou vodičů adresové sběrnice. Ta se projevila tím, že mikroprocesor pracoval v naprosto nekontrolovatelných adresách a přitom zjištění této závady trvalo několik dnů. IO do objímek doporučuji vložit postupně a současně vždy zkontrolovat jejich činnost. Nejdříve je vhodné vložit IO TTL. Po důkladném zkontrolování činnosti se kontrolují vývody objímek pro paměti a mikroprocesor logickou sondou a voltmetrem. Zde je nutné si vždy uvědomit, jak se chovají nepřipojené vstupy IO, vstupy připojené přes odpor na +5V, vstupy připojené na zem, výstupy ve stavu H a L. Teprve potom se vkládají paměti a mikroprocesor. Při vkládání mikroprocesoru je třeba dodržovat všechny zásady nutné pro práci s IO typu N-MOS, jinak hrozí poškození IO statickou elektřinou. Veškerá tato činnost vyžaduje určité zkušenosti s oživováním desek s číslicovými IO. Tolik připomínka k oživování mikropočítače bez zvláštních přístrojů.

Protože je použita univerzální deska s plošnými spoji s otvory pro šestnácti-vývodové IO, byly kromě mikroprocesoru použity pouze čtrnácti a šestnácti vývodové IO, přestože by bylo v některých případech cenově i energeticky výhodnější použít IO s více vývody.

Například adresový registr 2x7475 lze nahradit jedním IO 3212, budič datové sběrnice 2x3216, podobně porty 01H až 06H, port klávesnice a displeje, dekodér portů 2x7442 obvodem 74154 apod.

Logický zisk IO TTL N = 10, u výkonových členů N = 30, mikroprocesor má N = 2, jak plyne z katalogových hodnot. Z tohoto důvodu je vhodné posílit vstupy mikroprocesoru ve stavu H napětím +5V přes odpor 3,3 kΩ. Nevyužité vstupy mikroprocesoru se ošetří stejně jako u obvodů TTL.

Celkové podrobné schéma mikropočítače je na obr. 14. Signál READY je při činnosti mikropočítače nutno udržovat v úrovni H. Jestliže některý ze vstupů WAIT přejde do stavu H, přejde signál READY do L a činnost mikroprocesoru se zastaví. Signál RESET je aktivní ve stavu L, tím se nuluje PC mikroprocesoru a po jeho přechodu do H začne mikroprocesor pracovat od adresy 0000H. Vstup RESET je zapojen tak, aby po připojení napájení začal mikroprocesor pracovat od adresy 0000H. Vývody SID a SOD jsou sériový vstup a výstup. O nich bude blíže pojednáno při vysvětlování činnosti interfejsu magnetofonu. Vstup HOLD je při

normální činnosti ve stavu L. Je-li převeden do stavu H, odpojí se přes výstup HLDA adresová, datová a řídicí sběrnice a tím je umožněn přímý přístup do vnitřní paměti mikropočítače přes konektor K1. Vstupy RST 5.5, RST 6.5, RST 7.5, TRAP a INTR jsou při normální činnosti mikroprocesoru (není stav HOLD, WAIT, INTERRUPT) ve stavu L. Do stavu H se přivedou, chceme-li provést přerušování. Výstupy RESET OUT, CLK nejsou k činnosti mikropočítače využity, jsou pouze vyvedeny na K1, K3. Krystal je připojen na vývody X1, X2. Signály S0, S1 nejsou využity. Vývody A8 až A15, AD0 až AD7, WR, RD, IO/M slouží jako adresová, datová a řídicí sběrnice. Jejich činnost je popsána v čl. 4. Signál ALE přechází do stavu H pouze v T1 (obr. 7). Jeho sestupná hrana slouží k oddělení nižších bitů adresové sběrnice, které po celou zbývající dobu operačního cyklu drží adresový registr (2x7475) IO 6C, 6D. Signály MEM/R, MEM/W, IO/R, IO/W jsou získány hradlovou sítí z IO 3G, 2G. Jako obousměrný budič datové sběrnice slouží IO 6A, 6B (2x3216). Budiče 7A, 7B negují datovou sběrnici mimo mikropočítač. Při adresách menších než 1000H jsou obvody 7A, 7B (2x3226) uzavřeny signálem na jejich vstupu CS, který je aktivován z adresové sběrnice přes hradlo IO 3G. Směr toku dat přes obvody 6A, 6B, 7A, 7B je dán úrovní vstupů DC, ty jsou získány z řídicí sběrnice. Obvody NAND 7C, 7D, 7E, 7F (4x7438) slouží jako posilovače adresové sběrnice. Ta může být přes tyto obvody také odpojována signálem HLDA ve stavu HOLD.

Pro napojení paměti a periférií na desce mikropočítače stačí signály MEM/R, MEM/W, IO/R, IO/W řídicí sběrnice, D0 až D7 datová sběrnice a A0 až A11 část adresové sběrnice. Paměť ROM IO 4F, 4E, 5E, 4D, 5D, 4C, 5C, 4B, 5B, 4A, 5A, (12x74S287) je adresována přímo z adresové sběrnice A0 až A7 a signály CS0 až CS5 dekodovanými z adresové sběrnice dekodérem 4H (7442). Čtení z paměti ROM je zajištěno signálem řídicí sběrnice MEM/R. Paměť RAM IO 3C, 3D, 3E, 3F, 2C, 2E, 2F (8xK 565 RU2) je adresována z adresové sběrnice A0 až A9 a signálem CS RAM, který je získán opět dekodérem 4H. Jedná-li se o čtení nebo zápis rozhoduje úroveň signálu MEM/W. Vstupy paměti RAM a výstupy obou typů paměti jsou připojeny na datovou sběrnici po které probíhá přenos dat.

Na desce mikropočítače jsou zapojeny porty 01H až 06H, IO, 1A, 1B, 2A, 2B, 3A, 3B, 1H, 1K, 2H, 2K, 3H, 3K (12x3216). Porty jsou adresovány z adresové sběrnice přes signály CS1 až CS6 z dekodéru 1C (7442). Dále jsou na desce porty pro klávesnici a displej 1E, 1F (2x3216). Také tyto porty jsou adresovány přes dekodér 1D (7442), obvod 1G a hradlo 1H. Adresa těchto portů 0AH až 0FH je získána z výstupů dekodéru 1D jako signály CSA až CSF. Tyto signály adresují klávesnici přes K2. Směr toku dat z portů do mikroprocesoru a naopak je určen úrovní signálu IO/R. Je-li adresováno vnější zařízení port 10H až FFH, nebo vnější paměť, je vstupem CS, IO 1E, 1F (2x3226) přes hradlo z diod blokován vstup dat z portů 00H až 0FH na datovou sběrnici. Tím jsme v podstatě probrali stručně činnost jednotlivých obvodů mikropočítače BOB-85.

Napájení jednotlivých IO uvádí tabulka 10. Uspořádání blokovacích kondenzátorů a diody, která zabráňuje zničení mikropočítače při přepólování napájecího napětí, je na obr. 15. Rozmístění IO a systém značení pozic IO je na obr. 16. Na tomto obrázku jsou také znázorněny jednotlivé

Tab. 11. Signály na vývodech konektoru K1 (multibus) BOB-85

1	0V	29		57	A0
2	0V	30		58	AT
3	+5V	31	CLK	59	SID
4	+5V	32	RESET OUT	60	SOD
5	+5V	33		61	RST 5.5
6	+5V	34		62	RST 6.5
7		35		63	RST 7.5
8		36		64	TRAP
9		37		65	
10	0V	38		66	HOLD
11	0V	39		67	D6
12		40		68	D7
13		41		69	D4
14	RESET	42		70	D5
15		43	A14	71	D2
16		44	A15	72	D3
17		45	A12	73	D0
18		46	A13	74	D1
19	IO/W	47	A10	75	0V
20	IO/R	48	A11	76	0V
21	MEM/W	49	A8	77	
22	MEM/R	50	A9	78	
23		51	A6	79	0V
24		52	A7	80	0V
25	WAIT 8	53	A4		
26		54	A5		
27		55	A2		
28		56	A3		

Tab. 12. Signály na vývodech konektoru K2 (klávesnice a displej) BOB-85

1	0V	24	O7
2	0V	25	CSA
3	+5V	26	CSB
4	+5V	27	CSC
5	RESET	28	CSD
6		29	CSE
7		30	CSF
8	WAIT 7	31	
9	IO	32	
10	O1	33	
11	I1	34	
12	O1	35	
13	I2	36	
14	O2	37	
15	I3	38	
16	O3	39	
17	I4	40	
18	O4	41	
19	I5	42	
20	O5	43	IO/W
21	I6	44	IO/R
22	O6	45	MEM/W
23	I7	46	MEM/R

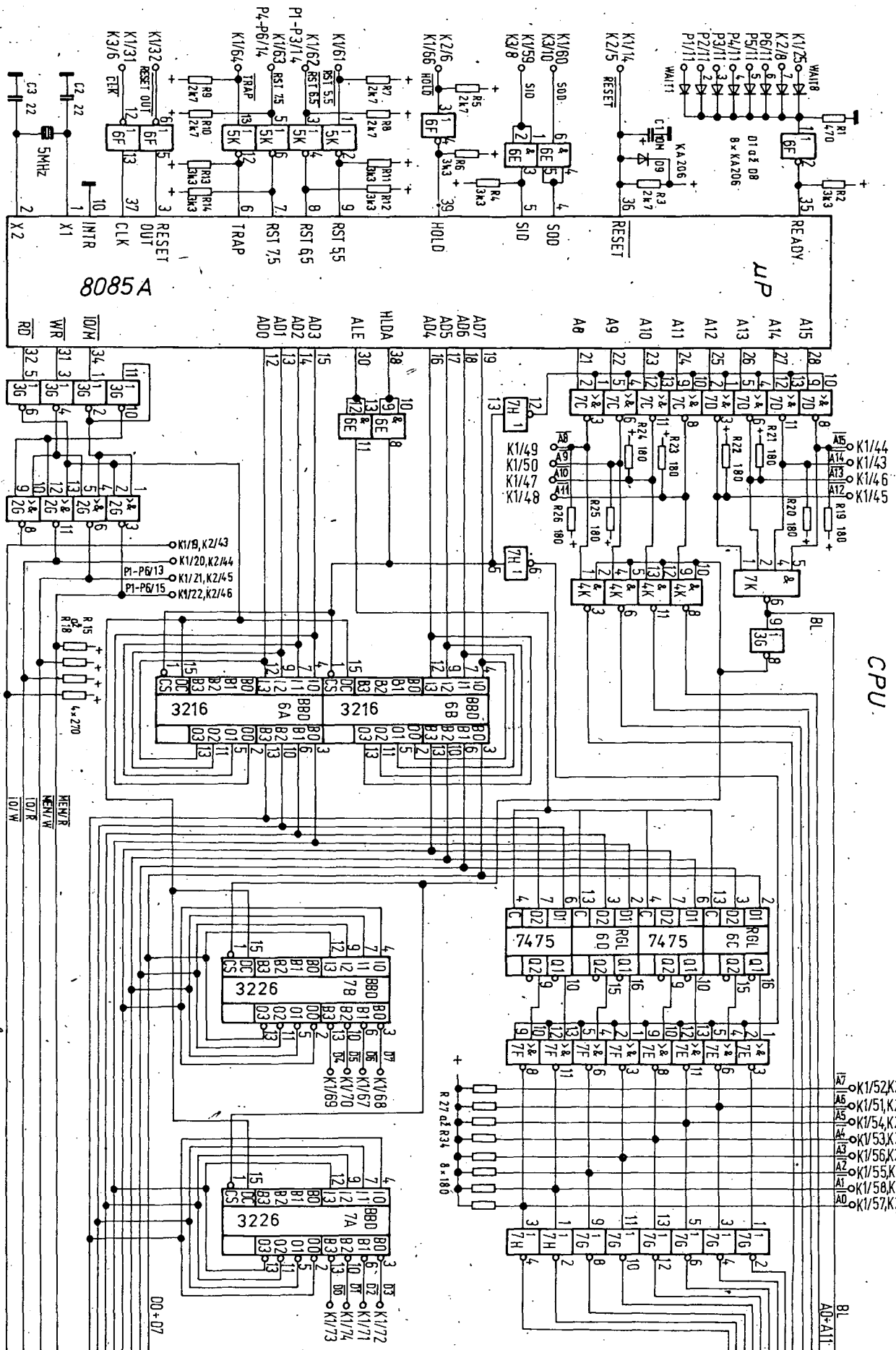
Tab. 13. Signály konektoru P1-P6 BOB-85

1	0V
2	+5V
3	D0
4	D1
5	D2
6	D3
7	D4
8	D5
9	D6
10	D7
11	WAIT i
12	CST
13	IO/R
14	RST 6.5
15	IO/W

Tab. 14. Signály konektoru K3 BOB-85

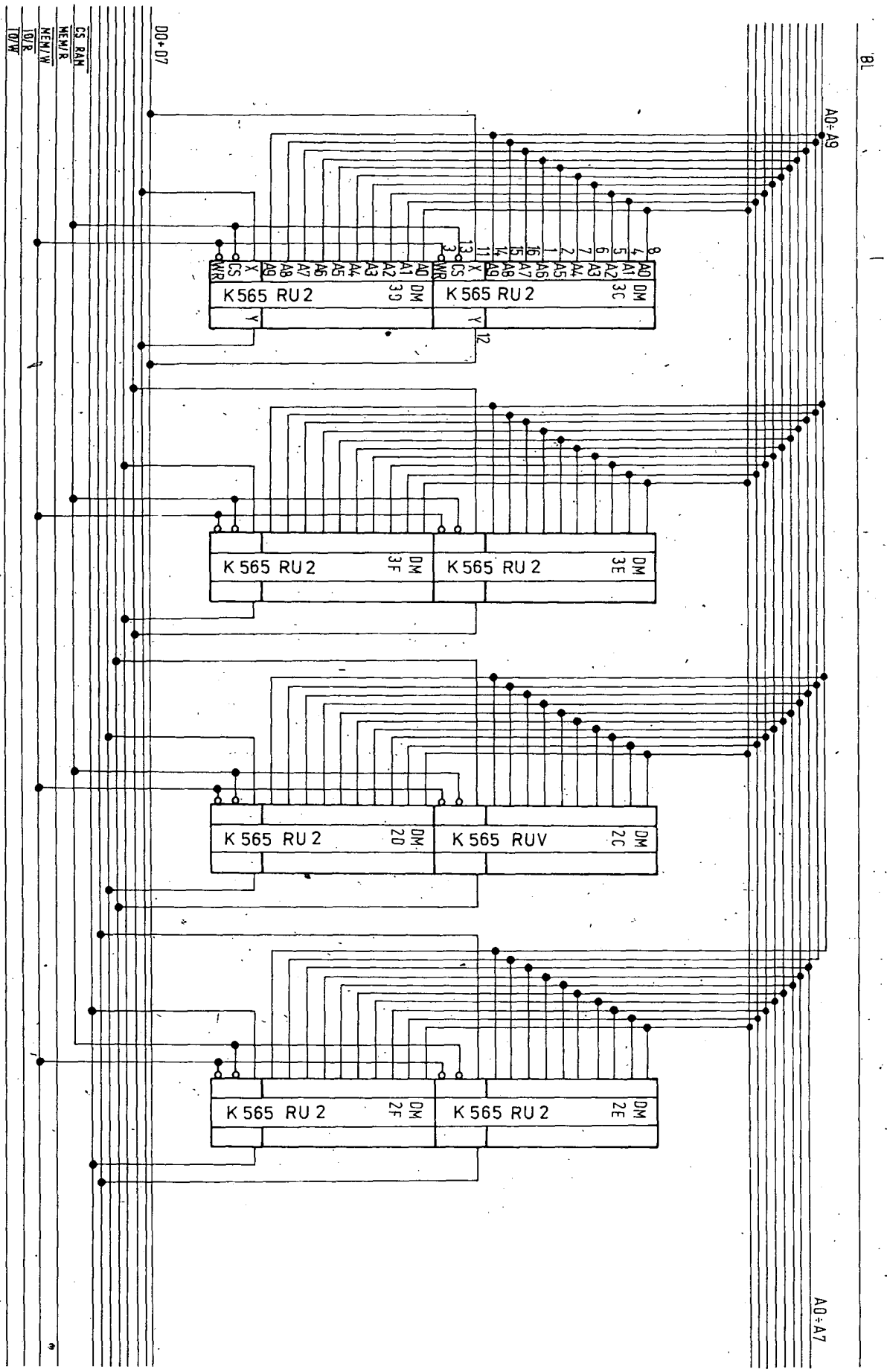
1	0V
2	+5V
3	—
4	—
5	0V
6	CLK
7	0V
8	SID
9	0V
10	SOD

(i = 1-6)

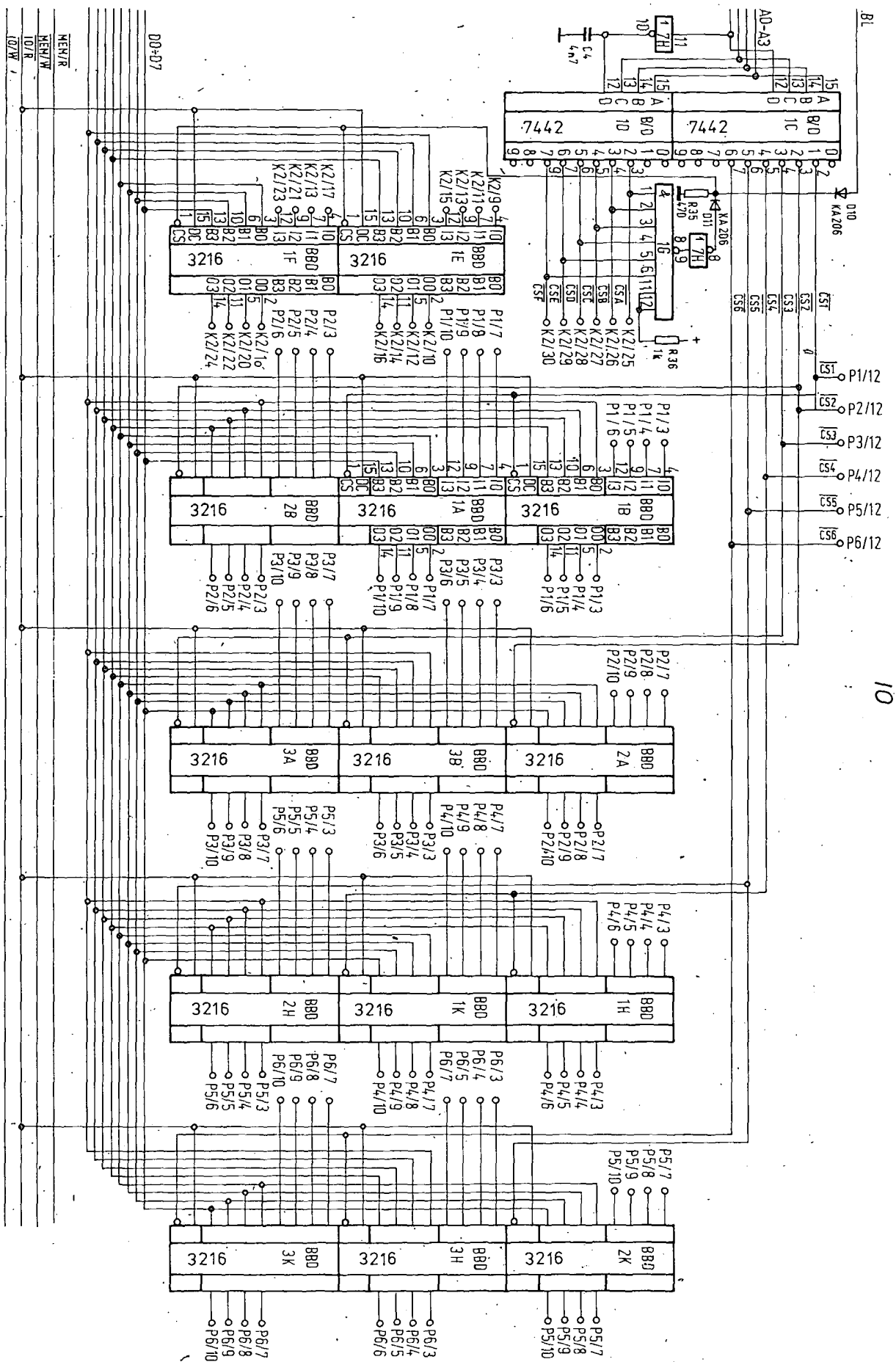


Obt. 14. Schéma mikropočítače BOB-85

RAM



Obr. 14. Schéma mikropočítače BOB-85 – pokračování



Obr. 14. Schéma mikropočítače BOB-85 – pokračování

Tab. 15. Operační kódy programu Monitor a jejich uspořádání v MH 74S287

pozice 10:4F		(adresa 00H–FFH)	
3	E 0 3 0 3 F 2 8 8 C 9 0 C 4 0 E F 7 C 9 0 C 6 0 F 6 C 1 0 C F		
0	F E C 0 C 0 0 0 C		
F	0 C E 0 C 6 0 F 1 D 5 0 C 6 0 F 2 C 0 0 F 4 C 0 0 F 5 C 0 0 C		
F	0 C 7 0 2 F 0 F 3 C 1 F 0 1 C F 0 F C F 0 C 9 0 C 8 0 C 6 0 F		
6	C 5 0 C F C D 0 C 6 0 F 7 C C 0 6 F 8 6 F C 9 0 F C D 0 F 7 C		
6	0 C C 0 C 6 0 6 F 8 6 F C 9 0 F C D 0 C 6 0 F 7 C C 0 6 F 8 6		
F	C 9 0 F C D 0 F 7 C 6 0 C 0 0 C 6 0 6 F 8 6 F C 9 0 C 8 0 C 6		
0	F 1 D 3 3 3 3 3 3 C 7 0 2 0 0 F 3 0 D 0 D 0 F C F A D 0 D 0 D		
pozice 10:5F		(adresa 00H–FFH)	
1	6 9 E 8 0 B 1 5 0 D 5 1 3 2 0 3 5 E D 5 1 D 7 1 E 0 2 6 0 D 9		
0	1 3 9 0 9 0 0 0 D		
9	0 D D 0 D 7 1 E 0 A 8 1 A 2 0 E 0 A 9 1 E 0 A 0 0 E 0 A 0 0 3		
E	1 D 6 0 2 E 9 5 E 3 1 D 9 2 D 9 0 1 3 D 9 D 5 1 D 5 0 D 7 1 E		
0	2 8 1 9 5 D E 0 D 1 1 5 C D E 1 7 1 4 7 1 D 5 1 5 D E 0 5 C D		
1	1 D E 1 D 1 1 7 1 4 7 1 D 5 1 5 D E 0 D 1 1 5 D D E 1 F 1 5 F		
1	D 5 1 5 D E 0 5 D D 1 1 D E 1 D 1 1 F 1 5 F 1 D 5 1 3 5 0 D 7		
1	E 0 8 3 3 3 3 3 3 F 0 1 0 0 5 E 4 3 E 3 F 1 9 5 F 3 A 3 B 3		
pozice 10:4E		(adresa 00H–FFH)	
0	D 0 D 0 D 0 F C C 7 0 7 C 9 0 C 6 0 F 1 D 2 0 F 6 C 2 0 2 C 0		
0	F 3 C 5 0 2 C 0 0 C 6 0 F 7 C C 0 7 F 8 C 9 0 F 4 7 C 6 0 F 1		
D	4 0 C C 1 0 7 7 C 6 0 C C 0 C 6 0 8 7 C C 0 0 2 E E C 9 0 C 4		
0	0 0 0 0 0 C 0 C C 7 0 4 C 8 0 D 0 1 3 3 1 B C C C 4 C 6 0 D 0 1		
D	7 0 3 1 C D F 1 9 0 1 7 C 0 C 8 0 F D C F D E C D 0 D 0 7 C 6		
0	C D 0 D 0 E E 7 C D 0 D 0 7 C 6 0 C D 0 D 0 E E 7 C D 0 D 0 7		
C	6 0 C D 0 D 0 E D F C 0 0 5 0 0 3 3 1 0 C D 0 0 0 0 0 C C C 0		
C	E 0 C E 2 E 0 8 6 7 E C F 6 B F 6 D D 7 F 7 7 C 9 E 9 1 0 F 3		
pozice 10:5E		(adresa 00H–FFH)	
C	3 D 3 E 3 F 1 9 D 6 0 E D 5 1 D 7 1 E 0 A A 1 E 0 2 1 1 3 3 C		
1	E 0 2 8 1 B 3 C 1 D 1 1 5 E D E 1 7 1 6 D 5 1 5 7 7 D 7 1 E 0		
A	7 1 1 3 8 1 7 8 D 1 1 D E 1 D 1 1 6 7 1 3 C 1 1 E E D 5 1 3 2		
0	7 7 7 7 9 0 5 D D 1 F D 6 1 B A 7 7 F F 9 1 8 5 F 3 F 1 B A 7		
2	D 1 F F 9 5 5 1 D 1 B A 6 0 2 B 1 1 1 9 5 5 5 D D 1 3 F A D 1		
1	D D 1 3 E 1 5 C D D 1 3 B A D 1 1 D D 1 3 A 1 5 D D D 1 3 D A		
D	1 1 D D 1 3 C 1 1 1 9 0 0 7 E 4 7 F 7 D 2 1 1 F F F F 9 D E 1		
D	4 1 9 5 1 D 1 5 F E 1 9 3 0 5 4 6 6 7 0 7 6 7 7 3 5 7 7 0 E 0		
pozice 10:4D		(adresa 00H–FFH)	
C	5 0 C E 0 3 0 D 0 C 6 0 F 5 C 6 0 F 4 C 0 0 C C 0 E C F 0 0 7		
E	0 0 F 3 C C 5 0 0 C 2 0 A C 5 0 4 0 3 0 2 0 0 3 0 C F D C 0 0		
A	3 C C 5 0 7 1 4 3 0 1 1 C 5 0 A C 5 0 0 C 4 0 C D F C 1 1 3 1		
1	1 C 6 0 E 8 1 C 5 0 C C C 0 0 E C 0 F C B 0 D 7 0 0 C 7 0 C C		
9	0 7 2 C 0 C 7 0 C E 0 C F 0 B C 5 0 C 0 0 1 1 1 C B 0 D 9 0 C		
B	0 D 9 0 1 C B 0 D A 0 C B 0 D A 0 7 1 7 1 4 0 C 9 0 C 1 1 1 C		
B	0 2 1 C C 7 0 A C D 0 C F 0 C C D 0 C 6 0 4 C D 0 8 4 C D 0 C		
9	0 C 6 0 F 6 D C D F 1 D 5 0 C C A 8 8 8 8 2 0 C F 0 2 3 C C 0		
pozice 10:5D		(adresa 00H–FFH)	
2	8 1 D D 0 E 5 3 F D 7 1 E 0 A C 2 E 0 2 A 2 D 5 2 5 D 0 2 0 7		
1	4 E A E 0 D C 2 D 2 6 2 F D C 2 E D B 2 3 5 2 1 2 7 3 5 5 6 9		
F	E 0 D C 2 9 F F E 1 F F D C 2 F D C 2 5 2 0 2 1 1 B 9 6 0 0 E		
E	D 2 1 2 E 0 5 2 E 2 9 D 5 2 4 5 5 E A D C 2 2 2 2 D 2 4 2 5 D		
4	2 1 3 1 5 2 E 2 1 1 5 D 0 2 E 2 8 1 7 6 9 6 6 5 D C 2 A 8 2 D		
C	2 A 8 2 4 D C 2 2 5 2 D C 2 2 5 2 A 7 9 F F 5 2 6 2 9 E 6 D 2		
E	2 0 7 9 D 6 0 F D 0 2 D 9 0 9 D F 2 D 1 1 7 D F 2 0 7 3 0 2 D		
5	1 D 7 1 E 0 1 8 5 E 0 2 8 1 9 5 F 4 5 0 6 3 5 2 5 2 F C 1 9 0		

Integrované obvody

IO μ P	1 8085 A	1 ks
IO 5F, 4F, 5E, 4E, 5D, 4D (5C, 4C, 5B, 4B, 5A, 4A)	MH 74S287	6 ks (12 ks)
IO 2C, 2D, 2E, 2F, 3C, 3D 3E, 3F	K 565 RU2 (MHB 2102 A)	8 ks
IO 6A, 6B, 1E, 1F (1A, 1B, 2A, 2B, 3A, 3B, 1H, 1K 2H, 2K, 3H, 3K)	MH 3216	4 ks (16 ks)
IO (7A, 7B)	MH 3226	(2 ks)
IO 4H, 1C, 1D	MH 7442	3 ks
IO 6C, 6D	MH 7475	2 ks
IO 7C, 7D, 7E, 7F, 2G	MH 7438	5 ks
IO 1G	MH 7430	1 ks
IO 7K	MH 7420	1 ks
IO 6E	UCY 7408	1 ks
IO 7G, 7H, 6F, 5K, 3G	MH 7404	5 ks
IO 4K	MH 7400	1 ks

Diody

D1 až D11	KA 206	11 ks
D12	KZ 260/6V2	1 ks

Krystal
KR 5 MHz (možno nahradit viz text) 1 ks

Odporý (miniaturní např. TR191, 151, 112a)

R1, R35	470 Ω	2 ks
R2, R4, R6, R11–R14	3,3 k Ω	7 ks
R3, R5, R7–R10	2,7 k Ω	6 ks
R15–R18	270 Ω	4 ks
R19–R34	180 Ω	16 ks
R36	1 k Ω	1 ks

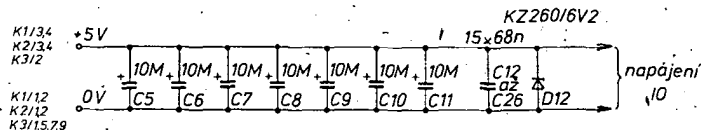
Kondenzátory keramické (miniaturní např. TK754-774, 794, 724, 744, 783, 755, 795, 725, 745)

C2, C3	22 pF	2 ks
C12 – C26	68 nF	15 ks
C4	4,7 nF	1 ks

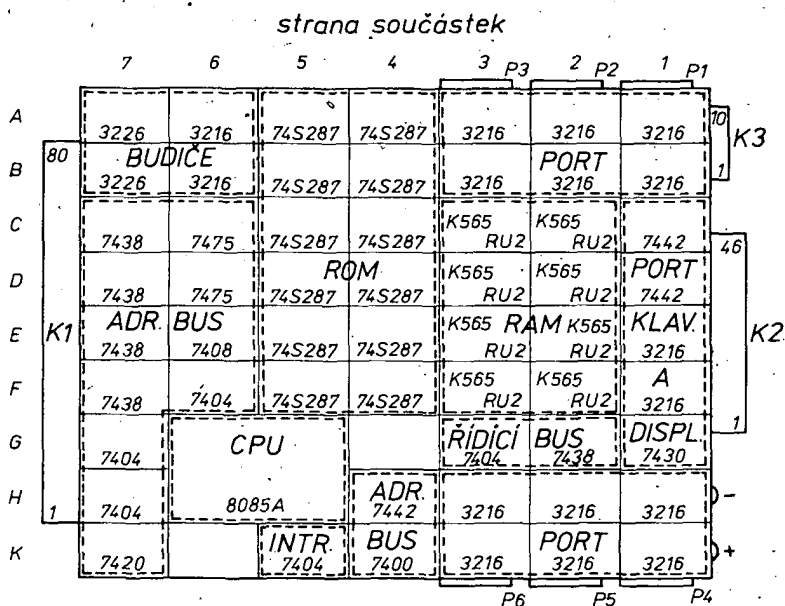
Kondenzátory elektrolytické (miniaturní minimálně na 6 V např. TE 181–993, TC 972–979)

C1, C5 – C11	10 μ F	8 ks
--------------	------------	------

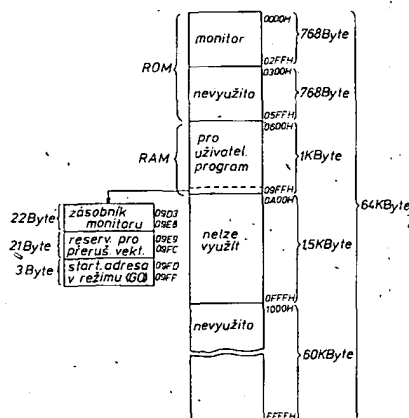
V tomto článku, kde se popisuje konstrukce a zapojení funkčního vzoru mikropočítače BÖB-85, se nebudeme detailně seznamovat s řídicím programem Monitor, ale budou uvedeny pouze operační kódy programu tak, jak jsou zapsány do IO 745287. Operační kódy pro jednotlivé IO jsou uvedeny v **tabulce 15**. Jejich pořadí je uvedeno vzestupně od adresy 00H až FFH. Jeden operační kód programu Monitor je vždy tvořen dvojicí znaků, přičemž každý znak dvojice je naprogramován v jiném IO. Například první operační



Pozn.: C5 až C11 jsou umístěny na desce tak, aby každý blokoval jednu řadu IO. C12 až C26 jsou přímo n
vývodech napájení IO 7B, 6A, 6D, 3A, 1B, 3K, 1H, 1E, 5B, 4E, 3F, 2D, 3E, 2F, 5D.



Obr. 16. Uspořádání IO na desce mikropočítače BOB-85



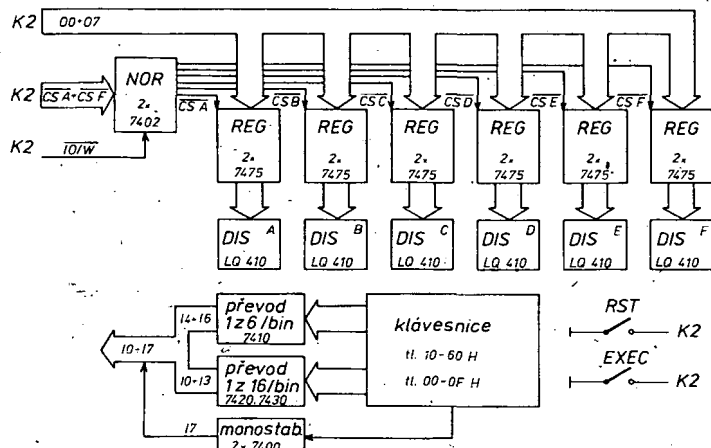
Obr. 17. Využití paměti BOB-85

Seznam součástí mikropočítače je uveden v **tabulce 16.**

7. Klávesnice a displej

Klávesnice obsahuje 24 tlačítek. Z toho 16 tlačítek generuje 16 čísel v kódu 1 z 16 a převodník je převádí do binárního kódu

Displej je připojen přes mezipaměti k datové sběrnici tak, jak bylo vysvětleno na blokovém schématu. Bajt z A-registru



Obr. 18. Blokové schéma desky klávesnice a displeje.

Tab. 17. Kód displeje sestavený podle obr. 19, 20.

znak	tvar	kód displeje		znak	tvar	kód displeje	
		hex.	bin.			hex.	bin.
			7 6 5 4 3 2 1 0				7 6 5 4 3 2 1 0
0		F3	1 1 1 1 0 0 1	P		37	0 0 1 1 0 1 1 1
1		60	0 1 1 0 0 0 0 0	r		05	0 0 0 0 0 0 1 0 1
2		B5	1 0 1 1 0 1 0 1	mez		00	0 0 0 0 0 0 0 0
3		F4	1 1 1 1 0 1 0 0	J		E1	1 1 1 0 0 0 0 1
4		66	0 1 1 0 0 1 1 0	U		E3	1 1 1 0 0 0 1 1
5		D6	1 1 0 1 0 1 1 0	Y		E6	1 1 1 0 0 1 1 0
6		D7	1 1 0 1 0 1 1 1	n		45	0 1 0 0 0 1 0 1
7		70	0 1 1 1 0 0 0 0	o		C5	1 1 0 0 0 1 0 1
8		F7	1 1 1 1 0 1 1 1	.		08	0 0 0 0 1 0 0 0
9		76	0 1 1 1 0 1 1 0	-		04	0 0 0 0 0 1 0 0
A		77	0 1 1 1 0 1 1 1	?		3D	0 0 1 1 1 1 0 1
B		C7	1 1 0 0 0 1 1 1	!		28	0 0 1 0 1 0 0 0
C		93	1 0 0 1 0 0 1 1	/		02	0 0 0 0 0 0 1 0
D		E5	1 1 1 0 0 1 0 1	//		63	0 1 1 0 0 0 1 1
E		97	1 0 0 1 0 1 1 1	=		94	1 0 0 1 0 1 0 0
F		17	0 0 0 1 0 1 1 1	4		46	0 1 0 0 0 1 1 0
H		67	0 1 1 0 0 1 1 1	7		25	0 0 1 0 0 1 0 1
L		83	1 0 0 0 0 0 1 1	U		26	0 0 1 0 0 1 1 0

mikroprocesoru je přímo poslán na displej instrukcí OUT XX, kde XX je číslo displeje 0AH až 0FH v pořadí zleva doprava. Pokud chceme zobrazit obsah A-registru na displeji (obsah A-registru má tvar 0XH, kde X je hexadecimální číslo), použijeme nejdříve podprogram KOD (v Monitoru). Ten překóduje číslo X v A-registru na kód displeje (viz tabulka 17). Potom se kód displeje pošle na příslušný displej instrukcí OUT. Chceme-li zobrazit jiný znak než hexadecimální, musíme jeho kód podle tabulky 17 nejdříve zapsat do A-registru a potom jej vyslat na displej instrukcí OUT. Kdybychom při řešení interfejsu displeje použili převodník kódu displeje (např. paměť ROM), nebylo by možné zobrazovat jiný znak než hexadecimální.

Seznam součástek desky klávesnice a displeje je v tabulce 18. Uspořádání IO na desce je na obr. 21. Na obr. 20 je nakreslen způsob napojení displeje na datovou sběrnici, což má význam pro stanovení kódu displeje, jehož několik kombinací je ukázáno v tabulce 17. Odpory napojené na displej jsou umístěny pod displejem, odpory mikrosopínačů jsou umístěny mezi nimi. Opět je použit univerzální plošný spoj a jednotlivé spoje jsou vedeny vodičem U 0,3. Konstruktivní řešení vlastní klávesnice je naznačeno na obr. 22. Klávesnice byla popsána bílým propisem o výšce písma 3 mm a přestříkána lakem Pragosorb. Názyvy kláves a jejich význam je na obr. 23. Celkový pohled na funkční vzorek je na 2. str. obálky.

Tab. 18. Seznam součástek klávesnice a displeje

Integrované obvody

IO 1A, 1B, 1C, 1D, 1E, 2A 2B, 2C, 2D, 2E, 3A, 4A	MH 7475	12 ks
IO 3E, 4E	UCY 7402	2 ks
IO 5A, 7A	MH 7420	2 ks
IO 6A	MH 7430	1 ks
IO 6E	MH 7400	1 ks
IO 7E	MH 7410	1 ks

Zobrazovací prvky LED

A - F	LQ 410	6 ks
-------	--------	------

Mikrosopínače

TL. 10-60, 00 - OF, RST, EXEC	mikrosopínač miniaturní	24 ks
-------------------------------	-------------------------	-------

Odpory (typ jako u BOB-85)

R1-R3, R5	180 Ω	4 ks
R4, R6, R7, R9, R10, R12, R17	270 Ω	7 ks
R8, R11, R13, R14	470 Ω	4 ks
R15, R16	5,6 kΩ	2 ks
R18-R65	150 Ω	48 ks

Kondenzátory keramické (typ jako u BOB-85)

C3	1,5 nF	1 ks
C6-C12	68 nF	7 ks

Kondenzátory elektrolytické (typ jako u BOB-85)

C1	2 μF	
C2	2 μF	
C4, C5	50 μF	

Tab. 19. Seznam součástek interfejsu magnetofonu

Integrované obvody

IO 1,2	MAA 741	2 ks
IO 3	UCY 74 123	1 ks

Diody

D1-D3	GA 201	3 ks
-------	--------	------

Odpory (typ jako u BOB-85)

R1	33 kΩ trimr	1 ks
R2	1 MΩ	1 ks
R3	3,9 MΩ	1 ks
R4	0,15 MΩ	1 ks
R5	270 Ω	1 ks
R6, R8	10 kΩ	2 ks
R7	27 kΩ	1 ks
R9	1,2 kΩ	1 ks
R10	820 kΩ	1 ks
R11	680 Ω	1 ks
R12	3,3 kΩ	1 ks
R13	68 kΩ	1 ks
R14	4,7 kΩ trimr	1 ks

Kondenzátory keramické (typ jako u BOB-85)

C3, C7	68 nF	2 ks
C4	0,1 μF	1 ks
C5	6,8 nF	1 ks

Kondenzátory elektrolytické (typ jako u BOB-85)

C1, C2	20 μF	2 ks
--------	-------	------

Tab. 20. Řídicí program MONITOR BOB-85

adresa	oper. kód	návěští	instrukce	komentář
0000	31E609	RS 0:	ORG 0000 H	Připrava
0003	3E08		LXI SP, 09EH	
0005	30		MVI A, 08H	
0006	FB		SIM	
0007	218580		EI	
000A	CD9501		LXI H, 8085H	
000D	C34200		CALL ZOBR	
0010	E3	RS 2:	JMP DAL 1	bod zastavení
0011	E5		XTHL	
0012	7E		PUSH PSW	
0013	CD9501		MOV A, M	
0016	CD6701	ZAS:	CALL ZOBR	
0019	FE60		CALL ZNAK	
001B	C21600		CPI 60H	
001E	CD9501		JNZ ZAS	
0021	F1		CALL TMA	
0022	E3		POP PSW	
0023	C9		XTHL	
0024	00	TRA:	RET	rezervováno pro přerušení
0025	00		NOP	
0026	00		NOP	
0027	00		NOP	
0028	00	RS 5:	NOP	
0029	00		NOP	
002A	00		NOP	
002B	00		NOP	
002C	00	RS 5.5	NOP	
002D	00		NOP	
002E	00		NOP	
002F	00		NOP	
0030	00	RS 6:	NOP	
0031	00		NOP	
0032	00		NOP	
0033	00		NOP	
0034	00	RS 6.5:	NOP	
0035	00		NOP	
0036	00		NOP	
0037	00		NOP	
0038	00	RS 7:	NOP	
0039	00		NOP	
003A	00		NOP	
003B	C9		RET	
003C	00	RS 7.5:	NOP	
003D	00		NOP	
003E	00		NOP	
003F	CD9501	DAL 2:	CALL TMA	pokračování přípravy
0042	CD9501	DAL 1:	CALL POMLK	
0045	CD6701	START:	CALL ZNAK	identifikace řídícího příkazu
0048	FE10		CPI 10H	
004A	DA5801		JC CHYBA	
004D	CA6200		JZ GO	
0050	FE20		CPI 20 H	
0052	CA0901		JZ SMEM	
0055	FE40		CPI 40H	
0057	CA0000		JZ VEK 1	
005A	FE50		CPI 50 H	
005C	CA0000		JZ VEK 2	
005F	C3FE01		JMP MGF	
0062	CD7600	GO:	CALL POM	skok na definovanou adresu
0065	22FE09		SHLD 09FE H	
0068	F5		PUSH PSW	
0069	3EC3		MVI A, C3 H	
006B	11FD09		LXI D, 09FD H	
006E	12		STAX D	
006F	CD9501		CALL TMA	
0072	F1		POP PSW	
0073	C3FD09		JMP 09FD H	
0076	CD9501	POM:	CALL ZOBR	pomocný podprogram pro definování adresy a ukončení definice tl. NEXT (60 H)
0079	CD8500		CALL ADR	
007C	CD6701		CALL ZNAK	
007F	FE60		CPI 60 H	
0081	C25801		JNZ CHYBA	
0084	C9		RET	
0085	F5	ADR:	PUSH PSW	podprogram umožňující definovat adresu
0086	CDDE00		CALL TEST D	
0089	CD6101		CALL ROT	
008C	F5		PUSH PSW	
008D	7C		MOV A, H	
008E	CDCE01		CALL POSUV	
0091	67		MOV H, A	
0092	F1		POP PSW	
0093	84		ADD H	
0094	67		MOV H, A	
0095	F1		POP PSW	
0096	CD9501		CALL ZOBR	
0099	F5		PUSH PSW	
009A	CDDE00		CALL TEST D	
009D	F5		PUSH PSW	
009E	7C		MOV A, H	
009F	CD6101		CALL ROT	
00A2	CDCE01		CALL POSUV	
00A5	CD6101		CALL ROT	

00A8	67		MOV H, A	
00A9	F1		POP PSW	
00AA	84		ADD H	
00AB	67		MOV H, A	
00AC	F1		POP PSW	
00AD	CD9501		CALL ZOBR	
00B0	F5		PUSH PSW	
00B1	CDDE00		CALL TEST D	
00B4	CD6101		CALL ROT	
00B7	F5		PUSH PSW	
00B8	7D		MOV A, L	
00B9	CDCE01		CALL POSUV	
00BC	6F		MOV L, A	
00BD	F1		POP PSW	
00BE	85		ADD L	
00BF	6F		MOV L, A	
00C0	F1		POP PSW	
00C1	CD9501		CALL ZOBR	
00C4	F5		PUSH PSW	
00C5	CDDE00		CALL TEST D	
00C8	F5		PUSH PSW	
00C9	7D		MOV A, L	
00CA	CD6101		CALL ROT	
00CD	CDCE01		CALL POSUV	
00D0	CD6101		CALL ROT	
00D3	6F		MOV L, A	
00D4	F1		POP PSW	
00D5	85		ADD L	
0096	6F		MOV L, A	
0097	F1		POP PSW	
0098	CD9501		CALL ZOBR	
00DB	C38500		JMP ADR	
00DE	CD6701	TEST D:	CALL ZNAK	pomocný podprogram, čeká na data, testuje je a vyrovnává ukazatel zásobníku
00D1	FE10		CPI 10 H	
0093	D8		RC	
00D4	33		INX SP	
00D5	33		INX SP	
00D6	33		INX SP	
00D7	33		INX SP	
00D8	33		INX SP	
00D9	33		INX SP	
00DA	C37F00		JMP TEST Q	
00ED	210000	POMLK:	LXI H, 0000 H	podprogram nuluje H, L registry a zobrazí pomlky v datové části displeje
00F0	F5		PUSH PSW	
00F1	3E04		MVI A, 04 H	
00F3	D30E		OUT 0E H	
00F5	D30F		OUT 0F H	
00F7	F1		POP PSW	
00F8	C9		RET	
00F9	F5	TMA:	PUSH PSW	podprogram ztmaví displej
00FA	AF		XRA A	
00FB	D30A		OUT 0AH	
00FD	D30B		OUT 0BH	
00FF	D30C		OUT 0CH	
0101	D30D		OUT 0DH	
0103	D30E		OUT 0EH	
0105	D30F		OUT 0FH	
0107	F1		POP PSW	
0108	C9		RET	
0109	CD7600	SMEM:	CALL POM	prohlížení paměti a zápis do paměti
010C	7E	PAMET:	MOV A, M	
010D	CD9501		CALL ZOBR	
0110	CD6701		CALL ZNAK	
0113	FE10		CPI 10H	
0115	DA2A01		JC DATA1	
0118	FE60	TEST SR:	CPI 60H	
011A	C22101		JNZ ZPET	
011D	23		INX H	
011E	C30C01		JMP PAMET	
0121	FE30	ZPET:	CPI 30H	
0123	C25801		JNZ CHYBA	
0126	2B		DCX H	
0127	C30C01		JMP PAMET	
012A	CD6101	DATA1:	CALL ROT	
012D	F5		PUSH PSW	
012E	7E		MOV A, M	
012F	CDCE01		CALL POSUV	
0132	77		MOV M, A	
0133	F1		POP PSW	
0134	86		ADD M	
0135	CD9501		CALL ZOBR	
0138	F5		PUSH B	
0139	47		MOV B, A	
013A	77		MOV M, A	
013B	CD6701		CALL ZNAK	
013E	FE10		CPI 10H	
0140	DA4701		JC DATA2	
0143	C1		POP B	
0144	C31801		JM TEST SR	
0147	77	DATA2:	MOV M, A	
0148	78		MOV A, B	
0149	CD6101		CALL ROT	
014C	CDCE01		CALL POSUV	
014F	CD6101		CALL ROT	
0152	86		ADD M	
0153	77		MOV M, A	
0154	C1		POP B	
0155	C30C01		JMP PAMET	

0158	21EEEE	CHYBA:	LXI H, EEEH	podprogram zobrazí	01EF	B5	DB B5H	2
015B	CD9501		CALL ZOBR	EEEE — a předá řízení	01FO	F4	DB F4H	3
015E	C34200		JMP DAL1	na identifikaci příkazů	01F1	66	DB 66H	4
0161	07	ROT:	RLC	zamění 4 významnější a 4 méně	01F2	D6	DB D6H	5
0162	07		RLC	významné bity v A registru	01F3	D7	DB D7H	6
0163	07		RLC		01F4	70	DB 70H	7
0164	07		RLC		01F5	F7	DB F7H	8
0165	C9		RET		01F6	76	DB 76H	9
0166	00		NOP		01F7	77	DB 77H	A
0167	C5	ZNAK:	PUSH B	čeká na 1 znak z klávesnice	01F8	C7	DB C7H	B
0168	CD7D01		CALL JEDEN	(PORT 0AH)	01F9	93	DB 93H	C
016B	4F		MOV C, A		01FA	E5	DB E5H	D
016C	CD8601		CALL MS3		01FB	97	DB 97H	E
016F	DB0A	VSTUP:	IN 0AH		01FC	17	DB 17H	F
0171	17		RAL		01FD	00	NOP	program magnetofon,
0172	37		STC		01FE	FE30	MGF: CPI 30H	příprava, identifikace,
0173	3F		CMC		0200	C25801	JNZ CHYBA	nahrávání nebo přehrávání
0174	1F		RAR		0203	CD0D00	CALL POMLK	
0175	B9		CMP C		0206	3E05	MVI A, KOD	
0176	C1		POP B		0208	D30F	OUT 0FH	
0177	C8		RZ		020A	CD6701	CALL ZNAK	
0178	C5		PUSH B		020D	FE50	CPI 50 H	
0179	4F		MOV C, A		020F	CA6C02	JZ PM	
017A	C36F01		JMP VSTUP		0212	FE40	CPI 40H	
017D	DB0A	JEDEN:	IN 0AH		0214	C20A02	JNZ MGF1	
017F	17		RAL		0217	CDC502	MP: CALL UVOD	příprava nahrávání z paměti
0180	D27D01		JNC JEDEN		021A	E5	PUSH H	na magnetofon, výpočet kontrol.
0183	3F		CMC		021B	CD0F02	CALL KS	součtu, generování dlouhého
0184	1F		RAR		021E	00	NOP	tónu, řízení nahrávání
0185	C9		RET		021F	77	MOV M, A	definovaného počtu BYTE
0186	D5	MS3:	PUSH D	zpoždění asi 3 ms	0210	E1	POP H	
0187	F5		PUSH PSW		0211	04	INR B	
0188	119D01		LXI D, 019D H		0222	0EFA	MP1: MVI C, LEADER	LEADER = délka dlouhého tónu
018B	1B	DC:	DCX D		0224	3EC0	MVI A, C0H	
018C	7A		MOV A, D		0226	CD5C02	MP2: CALL OBDEL	
018D	C600		ADI 0 H		0229	0D	DCR C	
018F	C28B01		JNZ DC		022A	C22602	JNZ MP2	
0192	F1		POP PSW		022D	AF	XRA A	
0193	D1		POP D		022E	CD5C02	CALL OBDEL	
0194	C9		RET		0231	4E	MP3: MOV C, M	
0195	F5	ZOBR:	PUSH PSW	zobrazí obsah H, L, A registrů	0232	CD3B02	CALL TAPEO	
0196	D5		PUSH D		0235	23	INX H	
0197	E5		PUSH H		0236	05	DCR B	
0198	CDDD01		CALL KODNUL		0237	C23102	JNZ MP3	
019B	D30F		OUT 0FH		023A	C7	RST 0	
019D	7A		MOV A, D		023B	F3	TAPEO: DI	nahrávání 1 BYTE z paměti do
019E	CD6101		CALL ROT		023C	D5	PUSH D	magnetofonu
01A1	CDDD01		CALL KODNUL		023D	C5	PUSH B	
01A4	D30E		OUT 0EH		023E	0609	MVI B, 9H	
01A6	E1		POP H		0240	AF	XRA A	
01A7	E5		PUSH H		0241	3EC0	MVI A, C0H	
01A8	7C		MOV A, H		0243	CD5C02	CALL OBDEL	
01A9	CDDD01		CALL KODNUL		0246	79	MOV A, C	
01AC	D30B		OUT 0BH		0247	1F	RAR	
01AE	7A		MOV A, D		0248	4F	MOV C, A	
01AF	CD6101		CALL ROT		0249	3E01	MVI A, 01H	
01B2	CDDD01		CALL KODNUL		024B	1F	RAR	
01B5	D30A		OUT 0AH		024C	1F	RAR	
01B7	E1		POP H		024D	CD5C02	CALL OBDEL	
01B8	E5		PUSH H		0250	AF	XRA A	
01B9	7D		MOV A, L		0251	CD5C02	CALL OBDEL	
01BA	CDDD01		CALL KODNUL		0254	05	DCR B	
01BD	D30D		OUT 0DH		0255	C24002	JNZ TO1	
01BF	7A		MOV A, D		0258	C1	POP B	
01C0	CD6101		CALL ROT		0259	D1	POP D	
01C3	CDDD01		CALL KODNUL		025A	FB	EI	
01C6	D30C		OUT 0CH		025B	C9	RET	
01C8	E1		POP H		025C	1610	OBDEL: MVI D, CYCNO	vysílání obdělíků přes
01C9	D1		POP D		025E	30	OB 1: SIM	výstup SOD podle příkazů
01CA	F1		POP PSW		025F	E11E	OB 2: MVI F, HALFCYC	podprogramů MP, TAPEO
01CB	C9		RET		0261	1D	DCR E	
01CC	00		NOP		0262	C2602	JNZ OB2	CYCNO = počet obdělíků
01CD	00		NOP		0265	EE80	XRI 80H	HALFCYC = délka obdělíků
01CE	57	POSUV:	MOV D, A	významnější 4 bity A registru	0267	15	DCR D	
01CF	0E04		MVI C, 04H	nuluje	0268	C25E02	JNZ OB1	
01D1	37	CYKL:	STC		026B	C9	RET	
01D2	3F		CMC		026C	CDC502	PM: CALL UVOD	příprava přehrávání z magneto-
01D3	17		RAL		026F	04	INR B	fonu do paměti, výpočet
01D4	0D		DCR C		0270	E5	PUSH H	kontrol, součtu a jeho porovná-
01D5	C2D101		JNZ CYKL		0271	C5	PUSH B	ni s původním čeká na dlouhý
01D8	0F		RRC		0272	0EFA	PM1: MVI C, LDRCHK	tón a řízení přehrávání
01D9	0F		RRC		0274	CDBC02	PM2: CALL VSTUP	definovaného počtu BYTE
01DA	0F		RRC		0277	D27202	JNC PM1	
01DB	0F		RRC		027A	0D	DCR C	LDRCHK = doba testování
01DC	C9		RET		027B	C27402	JNZ PM2	dlouhého tónu
01DD	CDCE01	KOD NUL	CALL POSUV	nuluje 4 významnější bity A re-	027E	C5	PM3: PUSH B	
01E0	CDE401		CALL KOD	gistru a 4 méně významné bity	027F	CD9402	CALL TAPEIN	
01E3	C9		RET	překóduje na kód displeje	0282	71	MOV M, C	
01E4	E5	KOD:	PUSH H		0283	23	INX H	
01E5	21ED01		LXI H, TAB K	překóduje 4 méně významné bity	0284	C1	POP B	
01E8	85		ADD L	A registru na kód displeje	0285	05	DCR B	
01E9	6F		MOV L, A		0286	C27E02	JNZ PM3	
01EA	7E		MOV A, M		0289	C1	POP B	
01EB	E1		POP H		028A	E1	POP H	
01EC	C9		RET		028B	05	DCR B	
01ED	F3	TAB K:	DB F3H	znak 0, tabulka kódu displeje	028C	CD0F02	CALL KS	
01EE	60		DB 60H	1	028F	BE	CMP M	

0290	C25801		JNZ CHYBA	
0293	C7		RST 0	
0294	0609	TAPEIN:	MVI B, 9H	přehrávání 1 BYTE z magneto-
0296	1616	TI 1:	MVI D, 16 H	fonu do paměti
0298	15	TI 2:	DCR D	
0299	CDBC02		CALL VSTUP	
029C	DA9802		JC TI 2	
029F	CDBC02		CALL VSTUP	
02A2	DA9802	TI 3:	JC TI 2	
02A5	14		INR D	
02A6	CDBC02		CALL VSTUP	
02A9	D2A502		JNC TI 3	
02AC	CDBC02		CALL VSTUP	
02AF	D2A502		JNC TI 3	
02B2	7A		MOV A, D	
02B3	17		RAL	
02B4	79		MOV, A, C	
02B5	1F		RAR	
02B6	4F		MOV C, A	
02B7	05		DCR B	
02B8	C29602		JNZ TI 1	
02BB	C9		RET	
02BC	1E16	VSTUP:	MVI E, CKRATE	testování úrovně na vstupu SID,
02BE	1D		DCR E	podle příkazů podprogramů PM,
02BF	C2BÉ02		JNZ VST 1	TAPEIN
02C2	20		RIM	
02C3	17		RAL	CKRATE = doba testování
02C4	C9		RET	
02C5	C97600	UVOD:	CALL POM	podprogram pro definování
02C8	AF		XRA A	výchozí adresy a počtu nahrá-
02C9	CDD002		CALL POCETB	vaných nebo přehrávaných BYTE
02CC	CDF900		CALL TMA	
02CF	C9		RET	

02D0	CDDF02	POCETB:	CALL TEST ZN	podprogram pro definování
02D3	CD6101		CALL ROT	počtu BYTE
02D6	47		MOV B, A	
02D7	CDDF02		CALL TESTZN	
02DA	80		ADD B	
02DB	47		MOV B, A	
02DC	C3D002		JMP POCETB	
02DF	CD9501	TESTZN:	CALL ZOBR	podprogram pro příjem znaku
02E2	CD6701		CALL ZNAK	a současně testuje, není-li
02E5	FE60		CPI 60H	spuštěno tlačítko NEXT (60)
02E7	D1		POP D	
02E8	C8		RZ	
02E9	D5		PUSH D	
02EA	PE10		CPI 10H	
02EC	D25801		JNC CHYBA	
02EF	C9		RET	
02F0	C5	KS:	PUSH B	podprogram pro výpočet
02F1	AF		XRA A	kontrolního součtu
02F2	84		ADD H	
02F3	85		ADD L	
02F4	80		ADD B	
02F5	86	KS1:	ADD M	
02F6	23		INX H	
02F7	05		DCR B	
02F8	C2F502		JNT KS1	
02FB	2F		CMA	
02FC	3C		INR A	
02FC	C1		POP B	
02FE	C9		RET	
			END	

Poznámka: Seznam pseudoinstrukcí, ani seznam návěstí není uveden.

9.1.2 Skok do programu

Je-li stisknuto tlačítko GO, pracuje program od návěští GO (0062H). Nejdříve se na displeji objeví 0000 10 (příkaz číslo 10). Potom program čeká na další znaky. Může to být adresa, která se vypíše na displej (CALL POM). Nebo příkaz NEXT, který vyvolá ztmavnutí displeje (CALL TMA) a skok na adresu, která byla zapsána na displeji.

Program GO používá podprogram POM (0076H). Ten zobrazí obsah H, L, A registrů prostřednictvím podprogramu ADR (0085H). Dále provádí testování je-li znak, který přišel z klávesnice 60H tj. příkaz NEXT oddělovač. Teprve tímto znakem se dostaneme z podprogramu POM.

Podprogram POM používá podprogram ADR. Ten umožňuje vložení adresy do dvojice registrů HL a současně ji zapiše na displej. Podprogram ADR využívá podprogramy TEST D (00DEH – ten čeká na znak a testuje, jedná-li se o data), ROT (0161H – ten zaměňuje 4 nižší a 4 vyšší bity A-registru), POSUV (01CEH – ten nuluje 4 vyšší bity A-registru) a ZOBR (0195H ten zobrazuje na displej obsah H, L, A registrů). O podprogramu ZOBR pojednáme později v čl. 9.1.3. Použitý podprogram ZNAK (0167H) ještě využívá podprogram JEDEN (017DH). Vyskytne-li se v programu podprogram ZNAK, začne mikroprocesor číst z klávesnice (port 0AH). Vždy kontroluje 7. bit (I₇) přečtený z klávesnice. Je-li I₇ = H, je znak přečten, je-li I₇ = L čte a testuje dále. Je-li znak přečten, je 7. bit vynulován, znovu se čte a opět je 7. bit nulován. Jsou-li oba výsledky čtení shodné, je znak přijmut. Tak se kontroluje podprogramem JEDEN správnost přečteného znaku.

9.1.3 Prohlížení paměti a zápis programu

Je-li při identifikaci řídicího programu stisknuto tlačítko S MEM, následuje skok na návěští S MEM (0109H). Díky podprogramu POM se na displeji objeví 0000 20 (příkaz číslo 20) a stejně jako v programu GO se může na displej napsat výchozí adresa. Dále podprogram POM čeká na tlačítko oddělovače NEXT, teprve po něm je POM opuštěn. Následuje zobrazení

obsahu paměti, jejíž místo adresují HL registry (MOV A, M, CALL ZOBR). Dále čeká program na skupinu znaků, které identifikuje. Nyní mohou nastat 4 případy. Přijmuté znaky jsou data (znak <10H), oddělovač (znak = 60H), tlačítko REC (znak = 30H), nebo jiný znak. Jestliže je stisknut jiný znak, objeví se na displeji chyba a program je třeba začít znovu tlačítkem S MEM. Je-li stisknut oddělovač NEXT, je připočtena 1 k registrům H, L a následuje skok na návěští PAMET (010CH), podobně tlačítko REC snižuje obsah registrů H, L o 1. T. zn. že tlačítko NEXT, respektive REC, umožňuje prohlížení obsahu paměti po krocích (jednotlivých adresách od zvolené adresy v podprogramu POM) vpřed (adresa roste), respektive vzad (adresa klesá).

Jedná-li se o data (znak <10H), následuje skok na návěští DATA 1. Program DATA 1 zapiše a zobrazí obsah 4 předešlých horních bitů paměti (1 znak) a čeká na další znak. Jedná-li se o NEXT nebo REC, vrací se na návěští TEST SR (posun H, L registru o 1 vpřed nebo vzad) a dále na návěští PAMET (tím se celý cyklus prohlížení paměti nebo zápis do paměti může opakovat). Je-li znak přijatý programem DATA 1 (menší než 10H), přechází se na program DATA 2. Tam pracuje stejně jako DATA 1, pouze s tím rozdílem, že zapiše a zobrazí obsah 4 přepsaných dolních bitů paměti (1 znak).

Program S MEM tvoří tedy uzavřenou smyčku, z které se lze dostat jediné tlačítkem EXEC (nulování – funkce RESET).

Program S MEM používá zajímavý podprogram ZOBR. Tento podprogram zobrazí obsah H, L, A registrů na displeji. Nyní se s ním blíže seznámíme. Podprogram začíná návěstí ZOBR (0195H). Nejdříve se uloží obsahy registrů, které potřebuje pro svou činnost, do zásobníku, aby nedošlo k jejich zničení. Dále zapiše kód displeje dolní části (4 nižší bity) A-registru do A-registru (CALL KOD NUL) a zobrazí jej na displeji 0F, přičemž původní obsah A registru uložil podprogram POSUV do D – registru. Jestliže chceme zobrazit horní 4 bity A registru, musíme nejdříve naplnit A-registr původním obsahem (MOV A, D) a zaměnit dolní

a horní 4 bity A-registru (CALL ROZ). Po překódování podprogramem CALL KOD NUL je kód horních 4 bitů A-registru poslán na displej 0E (OUT 0EH).

Podobně zobrazíme obsahy registrů H, L přes A-registr na displej 0A; 0B, 0C, 0D. Nejdříve je však musíme umístit do A-registru ze zásobníku (POP H, PUSH H).

Nakonec podprogram ZOBR obnoví původní obsah použitých registrů A, D, E, H, L.

Podprogram ZOBR používá podprogram KOD NUL (01DDH), který obsahuje podprogram POSUV (vysvětlen v 9.1.2) a podprogram KOD (01E4H). KOD přepíše obsah dolních 4 bitů A-registru (vyšší 4 bity A registru rovný 0H) podle tabulky displeje TABK (01EDH) na kód displeje a tento umístí do A-registru.

V programu Monitor jsou použity ještě podprogramy MS 3 (0186H – zpoždění asi 3 ms), CHYBA (0158H) zobrazí na displeji EEEE – a předá řízení Monitoru. POMLK (00ED) nuluje H, L registry a zobrazí pomlky na displejích OE a OF.

9.2 Řídicí program magnetofonu

Program začíná sice až na adrese 0200H a končí na 02FFH, ale jestliže se na program dostáváme z klávesnice tlačítkem REC (recorder), nikoliv prostřednictvím programu GO (skok na adresu 0200H), začíná návěstí MGF na adrese 01FEH. Byl-li při identifikaci řídicího příkazu (čl. 9.1.1) zjištěn příkaz 30 (tlačítko REC) dostává se program na návěští MGF. Určitá nelogičnost v uspořádání testu příkazů je způsobena tím, že magnetofon byl připojen až později a snahou bylo, aby program MGF nepřesáhl délku 256 Byte (celý Monitor v 6 IO typu 74S287).

V programu MGF je nejdříve zaznamenán jeho příjem zobrazením r na displeji. Dále čeká na znak VEK 1 nebo VEK 2 z klávesnice (jiný nepřijme). Podle toho jaké tlačítko bylo stisknuto je zvolen režim PM (026CH) zápis programu z magnetofonu do paměti nebo MP (0217H), záznam programu z paměti do magnetofonu.

9.2.1 Záznam programu na magnetofon

Program začíná návěstí MP. Podprogram UVOD (02C5H) umožní definovat

pomocí podprogramu POM výchozí adresu a po stisknutí tlačítka NEXT i počet zaznamenaných bajtů (max. FF). Podprogram KS (02F0H) provede výpočet kontrolního součtu (čl. 11.4) a výsledek ponechá v A-registru. Program MP uloží kontrolní součet za program, který se bude nahrávat na magnetofon.

Na návěští MP 1 (0222H) je definována délka úvodního dlouhého tónu (MVI C, LEADER), dále následuje instrukce určující chování následujícího podprogramu OBDEL (025CH – ten vytvoří 8 obdélníkových impulsů ozn. (8 × H, L) na výstupu mikroprocesoru SOD (obr. 25) je-li obsah A-registru = 00H, je-li obsah A-registru = 00H, vytvoří na SOD 8 × LL – délka úrovně H nebo L je dána konstantou HALFCYC a počet impulsů je dán konstantou CYCNO:2

$(CYCNO:2 = \frac{10H}{2} = \frac{16}{2} = 8)$. Celkový počet impulsů dlouhého tónu je tedy LEADER × CYCNO.

Po dlouhém tónu následuje 8 × LL (XRA A, CALL OBDEL). Potom již začíná zápis programu z paměti od adresy uložené ve dvojici registrů HL.

Obsah paměti je zapsán do C – registru a podprogram TAPEO (023BH) jej zapíše na magnetické médium. Potom je obsah HL zvýšen o 1 a následuje zápis dalšího bajtu. To pokračuje až do vyčerpání posledního bajtu (počet bajtů v B-registru). Potom se program vrací do Monitoru (RST0).

Nejdůležitější částí programu MP je podprogram TAPEO, který vysílá obsah C-registru na výstup mikroprocesoru SOD. Nejdříve se uloží registry, které TAPEO používá. Každý bajt se vysílá jako 9 bitů (1 bit startovací a 8 bitů nesoucích informaci – bajt). Pro každý bit je volán 3 × podprogram OBDEL a pomocí A6 (SOD enable bit) se klíčuje vysílání obdélníkových impulsů (viz obr. 25a). Před návratem z podprogramu TAPEO jsou obnoveny použité registry.

9.2.2 Zápis programu z magnetofonu do paměti

Program začíná návěští PM. Podprogram UVOD má stejný význam jako v čl. 9.2.1. Po návěští PM 1 již program čeká na dlouhý tón z magnetofonu. To se uskutečňuje prostřednictvím podprogramu VSTUP (02BCH), který čte úroveň signálu na sériovém vstupu mikroprocesoru SID do bitu CY. Teprve jestliže dlouhý tón přijde a skončí, následuje zápis programu z magnetofonu do paměti od adresy a s počtem bajtů definovaným podprogramem UVOD. Příjem jednotlivých bajtů do C-registru zajišťuje podprogram TAPEIN (0294H). Po nahrání celého programu včetně kontrolního součtu je vypočten kontrolní součet (CALL KS) nahraného programu a ten je porovnán s původním. Nejsou-li hodnoty stejné je signalizována chyba, jsou-li stejné předá se řízení Monitoru.

Nejdůležitější částí programu PM je podprogram TAPEIN. Ten vždy přečte 1 bajt (9 bitů jako v čl. 9.2.1) a uloží jej do C-registru. Čtení 1 bitu je založeno na principu odečítání 1 od obsahu D-registru v případě, že na vstupu mikroprocesoru SID je úroveň H a přičítání 1 do D-registru je-li na SID úroveň L. Byla-li čtena hodnota bitu obr. 25d log. 0, je výsledek odečítání v D-registru číslo větší než 0. Tím se nastaví hodnota sedmého bitu D-registru. D-registr se přehrává do A-registru, kde je posunut o 1 bit vpravo a dále je přehrán do C-registru. Po příchodu všech 9 bitů je v C-registru správná informace a následuje návrat z podprogramu.

9.3 Využití podprogramů z Monitoru

Podprogramy z programu Monitor je vhodné používat i v uživatelských programech. Pro jejich použití je nutné znát jejich název, adresu, činnost. Některé podprogramy ničí obsah některých registrů mikroprocesoru, proto je nutné před jejich použitím obsah těchto registrů (pokud jsou v uživatelském programu použity) uschovat např. do zásobníku nebo jiného nepoužívaného registru a po užití podprogramu zase obsah těchto registrů obnovit. Přehled nepoužívaných podprogramů Monitoru uvádí tabulka 21.

10. Obsluha BOB-85

Z hlediska programování je důležitá pouze znalost programovacího jazyku, bloková stavba mikropočítače (adresy umístění paměti, registry mikroprocesoru, adresy periférií, zejména klávesnice a displeje), možnosti využití podprogramů Monitoru a obsluha. Pokud by tedy někdo mikropočítač vyrobil, včetně uložení řídicího programu do ROM, mohl by s ním uživatel komunikovat a stačí mu minimální znalosti o jeho stavbě. Protože však hlavním cílem při použití mikropočítače BOB-85 je využití jeho řídicí schopnosti, je třeba, aby uživatel byl sám schopen vyvinout a ověřit jednoduché obvody (interfejs), které mohou pomoci řídicím programům řídit i poměrně složitá zařízení. O některých způsobech řešení interfejsu bude pojednáno v čl. 12. Tři jednoduché obvody (řízení klávesnice, displeje, magnetofonu) jsme již probrali.

Abychom mohli tato zařízení řídit, musí být pro ně vypracován nejen řídicí program, ale musí být popsána i obsluha. Pro jednodušší popis obsluhy minimální sestavy mikropočítače BOB-85 si nejdříve zavedeme určité konvence (viz tab. 22). Obsluhu popisují články 10.1–10.10.

10.1 Uspořádání sestavy a uvedení mikropočítače do provozu

Mikropočítač propojíme s deskou klávesnice a displeje přes konektor K2. Přes konektor K3 připojíme desku interfejsu magnetofonu, k níž je připojen přes konektor DIN magnetofon. Na zdířky mikropočítače, připojíme stabilizovaný zdroj 5V/4A. Po připojení napájecího napětí je mikropočítač automaticky nulován (obvod na vstupu RESET) a začne probíhat řídicí program Monitor. Na displeji se mikropočítač představí [8085] a dále čeká na řídicí příkaz <GO>, <SMEM>, <REC> apod. Po stisknutí tlačítka řídicího příkazu se objeví číslo příkazu na displeji [0000 10], [0000 20], [0000 30] apod.

10.2 Prohlížení paměti

1. <S MEM> → [0000 20]
2. adresa → [adr. 20]
3. <NEXT> → [adr., data]
4. opakujeme bod 3
5. prohlížení ukončíme
<EXEC> → [8085]
FORMÁT prohlížení paměti
<S MEM> adresa <NEXT>, <NEXT> ...
<NEXT> <EXEC>

10.3 Zápis programu do paměti

1. <S MEM> → [0000 20]
2. adresa → [adr. 20]

Tab. 21 Některé podprogramy Monitoru mohou být použity v uživatelských programech

Název	adresa	činnost
POMLK	00ED	nuluje H, L registry a zobrazí pomlky v datové části displeje
TMA	00F9	ztmaví displej
CHYBA	0158	zobrazí EEEE — a předá řízení Monitoru
ROT	0161	zamění 4 významnější a 4 méně významné bity v A registru
ZNAK	0167	čeká na 1 znak z klávesnice a ten zapíše do A registru, ničí C, D, F registry
MS 3	0186	čeká asi 3 ms
ZOBR	0195	zobrazí na displeji obsah H, L, A registrů, ničí C-registr
POSUV	01CE	nuluje 4 významnější bity A registru, ničí C, D registr
KOD	014E	zapíše do A registru kód displeje 4 méně významných bitů A registru

Pozn.: ničí registry rozuměj přepíše jejich původní obsah

Tab. 22 Konvence pro popis obsluhy mikropočítače

označení	význam
[]	záznam na displeji (obsah vypsán uvnitř rámečku)
< >	stisknutí tlačítka (typ tlačítka vypsán uvnitř závorek)
adresa	4 hexadecimální znaky
BYTE	2 hexadecimální znaky
→	z toho plyne (provede se následující apod.)

3. <NEXT> → [adr., data]
4. nová data → [adr., nová data]
5. <NEXT> → [adr.+1, data]
... (vlastní zápis nových dat při <NEXT>)
6. opakujeme body 4, 5
7. zapisování ukončíme tlačítkem <EXEC> → [8085]
FORMÁT vkládání programu do paměti
<SMEM> adresa <NEXT> data <NEXT> data <NEXT> ... data <NEXT> <NEXT>

10.4 Oprava programu

Zjistíme-li při vkládání programu nebo prohlížení paměti chybu, můžeme ji opravit tak, že tlačítkem <REC> se vrátíme o jednu adresu zpět, program opravíme a na původní adresu se vrátíme tlačítkem <NEXT>. Tlačítkem <REC> lze program také prohlížet jako v čl. 10.2, ale pozpátku. FORMÁT opravy programu při režimu podle čl. 10.2 a 10.3
... <REC> data <NEXT> ...

10.5 Start programu od určité adresy

Používá se při vstupu do uživatelského programu.

1. <GO> → [0000 10]
2. adresa → [adr. 10]
3. <NEXT> → program je odstartován a současně zhasne displej

FORMÁT spuštění programu
<GO> adresa <NEXT>

Tab. 23. Program HODINY

adresa	kód	návěšť	instrukce	komentář
0700	00	SEK:	ORG 0700H	
0701	00	MIN:	NOP	nastavení sekund
0702	00	HOD:	NOP	minut
0703	00		NOP	hodin
0704	00		NOP	
0705	318007		LXI SP, ZASOB	
0708	210007	SKOK:	LXI H, 0700H	
070B	7E		MOV A, M	
070C	23		INX H	
070D	5E		MOV E, M	
070E	23		INX H	
070F	56		MOV D, M	
0710	EB		XCHG	
0711	CD9501		CALL ZOBR	podprogram Monitoru
0714	EB		XCHG	
0715	CD5307		CALL ZPOŽ.	
0718	210007		LXI H, 0700H	
071B	7E		MOV A, M	
071C	C601		ADI 1	
071E	27		DAA	
071F	320007		STA 0700H	
0722	FE60		CPI 60H	
0724	C20807		JNZ SKOK	
0727	00		NOP	rezervováno pro CALL BUDIČ
0728	00		NOP	rezervováno pro CALL BUDIČ
0729	00		NOP	rezervováno pro CALL BUDIČ
072A	AF		XRA A	
072B	23		INX H	
072C	320007		STA 0700H	
072F	7E		MOV A, M	
0730	C601		ADI 1H	
0732	27		DAA	
0733	320107		STA 0701H	
0736	FE60		CPI 60H	
0738	C20807		JNZ SKOK	
073B	AF		XRA A	
073C	320107		STA 0701H	
073F	23		INX H	
0740	7E		MOV A, M	
0741	C601		ADI 1H	
0743	27		DAA	
0744	320207		STA 0702 H	
0747	FE24		CPI 24H	
0749	C20807		JNZ SKOK	
074C	AF		XRA A	
074D	320207		STA 0702 H	
0750	C30807		JMP SKOK	
0753	018FB2	ZPOZ:	LXI B, konst	podprogram zpoždění 1 sek.
0756	F5	1:	PUSH PSW	
0757	0B	2:	DCX B	
0758	CD6207		CALL ZP 3	
075B	78		MOV A, B	
075C	C600		ADI 0H	
075E	C25707		JNZ ZP 2	
0761	F1		POP.PSW	
0762	C9	3:	RET	
			END	

Tab. 24. Program DLOUHÝ TEXT

adresa	kód	návěšť	instrukce	komentář
0700	31FF07	ZAC:	LXI SP, ZASOB	
0703	213B07		LXI H, TEXT	
0706	7E		MOV A, M	
0707	FEFF		CPI FFH	
0709	CA0007		JZ ZAC	
070C	CD1807		CALL DISPL 6	
070F	CD3007		CALL ZPOZ	
0712	CD3007		CALL ZPOZ	
0715	C30607		JMP CYKL	
0718	D30A	DISPL6:	OUT 0AH	
071A	23		INX H	
071B	7E		MOV A, M	
071C	D30B		OUT 0BH	
071E	23		INX H	
071F	7E		MOV A, M	
0720	D30C		OUT 0CH	
0722	23		INX H	
0723	7E		MOV A, M	
0724	D30D		OUT 0DH	
0726	23		INX H	
0727	7E		MOV A, M	
0728	D30E		OUT 0EH	
072A	23		INX H	
072B	7E		MOV A, M	
072C	D30F		OUT 0FH	
072E	23		INX H	
072F	C9		RET	
0730	1100CF	ZPOZ:	LXI D, KONST	zpoždění
0733	1B	DC:	DCX D	
0734	7A		MOV A, D	
0735	C600		ADI 0H	
0737	C23307		JNZ DC	
073A	C9		RET	
073B	7767F3E10000	TEXT:	DW kód	00000000
41	B5E597000000		DW kód	00000000
47	00F7F3F7F7D6		DW kód	00000000
4D	E1776700D697		DW kód	00000000
53	7377D6000000		DW kód	00000000
59	000000000000		DW kód	00000000
5F	1777E1450000		DW kód	00000000
65	9377E3000000		DW kód	00000000
6B	E1E5E3000000		DW kód	00000000
71	E5F373E30000		DW kód	00000000
77	000000000000		DW kód	XXXXXX
7D	000000000000		DW kód	XXXXXX
83	000000000000		DW kód	XXXXXX
89	FF		DB FFH	XXXXXX; konec
			END	

7. <NEXT> → [] , po nahrání programu

FORMÁT pro zápis programu na magnetofon
 <REC> <VEK1> adresa₀ <NEXT> BYTE
 <záznam magnetofonu> <NEXT>

10.6 Přerušovací vektory

<VEK 1>, <VEK 2>, <RST> nejsou zatím využity. Tyto vektory umožní po stisknutí tlačítka přímý nájezd programu na předem definovanou adresu. Příkladem přerušovacího vektoru je i tlačítko <EXEC>, které vyvolá vynulování PC a tím uskuteční skok na adresu 0000H (začátek Monitoru).

Pozn.: U BOB—85 se nejedná o klasické přerušovací vektory, neboť ty umožňují přerušování probíhajícího programu. Jedná se pouze o přímý skok z monitoru na předem definovanou adresu.

10.7 Přerušování výpočtu

<EXEC> → [8085 — —]
 FORMÁT přerušování výpočtu
 <EXEC>

10.8 Využití bodů znovuspouštění

Při odlaďování programu je výhodné na místa, o kterých předpokládáme, že jimi musí program projít, vložit jednoduchou instrukci RST 2. Po odstartování programu se tento zastaví na adrese za RST 2 a na displeji se objeví adresa a obsah paměti. Dále program pokračuje v činnosti po stisknutí tlačítka <NEXT>.

10.9 Zápis programu z paměti do magnetofonu

1. <REC> → [8085 — —]
 2. <VEK 1> → [0000 40]
 3. adresa → [adr₀ 40] ... výchozí adresa
 4. <NEXT> → [adr₀ 00]
 5. BYTE → [adr₀ BYTE] ... počet bajtů
 6. nahrávání magnetofonu

10.10 Zápis programu z magnetofonu do paměti

1. <REC> → [8085 — —]
 2. <VEK 2> → [0000 50]
 3. adresa₀ → [adr₀ 50] ... výchozí adresa
 4. <NEXT> → [adr₀ 00]
 5. BYTE → [adr₀ BYTE] ... počet byte
 6. <NEXT> → [] mikropočítač čeká na dlouhý tón

7. přehrávání magnetofonu → [8085 — —] vše v pořádku
 [EEEE — —] chyba

FORMÁT pro zápis programu z magnetofonu do paměti
 <REC> <VEK 2> adresa₀ <NEXT> BYTE
 <NEXT> <přehrávání magnetofonu>
 Poznámka: Mezi jednotlivé programy na magnetickém médiu je

Tab. 25. Program ZOBRAZOVÁNÍ ORNAMENTŮ

adresakód	návěst	instrukce	komentář
0600	315006	ZAC:	ORG 0600H
0603	213006		LXI SP, 0650H
0606	7E	CYKL:	LXI H, ORNAM
0607	FEFF		MOV A, M
0609	CA0006		CPI FFH
060C	CD1506		JZ ZAC
060F	CD2506		CALL DISPL
0612	C30606		CALL ZPOZ
0615	D30A	DISPL:	JMP CYKL
0617	D30C		OUT 0AH
0619	D30E		OUT 0CH
			OUT 0EH

možno nahrát mikrofonom informace o typu programu. Osvědčilo se namluvit NÁZEV programu, výchozí adresa ADR₀ a počet BAJTŮ programu, mikropočítač na mluvené slovo nereaguje.

11. Příklady jednoduchých uživatelských programů

U jednotlivých odladěných uživatelských programů bude uvedeno pouze jejich použití, výpis programu v jazyku symbolických adres Assembler 8085 bez komentáře a jejich obsluha.

11.1 Program HODINY

Program zobrazuje hodiny, minuty a sekundy na displeji. Aby byl zajištěn správný čas, musíme přednastavit čas na adresách 0700–0702H a spustit program např. podle rozhlasu. Přesnost programu je díky použití krystalu 5 MHz velmi dobrá (asi 10⁻⁵ s). Pokud by byl použit jiný krystal, je třeba stanovit konstantu v instrukci LXI B, konstanta na adrese 0753H experimentálně. Program je v tab. 23. Jeho obsluha je následující:

1. Nahrajeme program do RAM od adr. 0700 (z mgf., nebo ručně čl. 10.3)
2. Na adresy 0700–0702H přednastavíme čas (s, min, h) ručně čl. 10.3
3. Startujeme od adresy 0705H dle čl. 10.5 při šestém pipnutí přesného času z rozhlasu.

11.2 Program DLOUHÝ TEXT

Program zobrazuje na displeji text podle kódu v tabulce TEXT. Program je zapsán v tab. 24 a startuje se od adresy 0700H podle čl. 10.5.

11.3 Program zobrazování ornamentů

Program byl upraven podle [3]. Zobrazuje na displeji ornamenty a je zapsán v tab. 25. Startuje se od adresy 0600H.

11.4 Program Kontrolní součet

Při zápisu programu do paměti z děrné pásky se na konec programu děruje tzv. kontrolní součet. Tento kontrolní součet slouží k testování, jestli se nahrávaná děrná páska dobře nahrála. DP je psána v tzv. formátu HEX a jednotlivé znaky jsou v kódu ISO. S kódem ISO se zde sice seznamovat nebudeme, ale formát HEX DP je uveden v tab. 26.

Tab. 26. Formát HEX pro N BYTE

pořadí znaků	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
znak (v kódu ISO):	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
význam	úvod	počet BAJTŮ	N	výchozí adresa	oddělovač	BAJT 1	BAJT 2	BAJT 3							BAJT N	kontrolní součet

Pozn.: Úvod – dvojtečka podle níž se pozná, že jde o formát HEX
počet BAJTŮ – nahrávaného programu
výchozí adresa – místo v paměti od kterého se program nahrává
kontrolní součet – viz čl. 11.4

Z tabulky je zřejmé, že DP obsahuje výchozí adresu, počet bajtů programu, oddělovač, a vlastní program (1 bajt = dvojice znaků). Na konci je uveden kontrolní součet, který se počítá ze všech znaků DP (kromě kontrolního součtu a úvodní dvojtečky) podle následujícího vzorce.

$$KS = \sum_{i=2}^N B_i + 1 - \text{oddělovač}$$

Slovně lze říci, že kontrolní součet je dvojkovým doplňkem součtu všech byte v hexadecimálním kódu zmenšeným o oddělovač. Oddělovač nabývá hodnot 00 nebo 01 (jedná-li se o konec programu). Maximální počet byte v jednom formátu HEX je FFH tj. 256. Je-li program delší, musí se rozdělit a právě zde má význam oddělovač.

Program pro výpočet kontrolního součtu je v tabulce 27.

Obsluha programu je následující. Program se startuje od adresy 0900H na displeji se zobrazí 0000 60. Dále určíme výchozí adresu ADR₀ (NEXT) počet byte a (NEXT). Na displeji se objeví konečná adresa a kontrolní součet. Od tohoto výsledku je třeba z paměti odečíst oddělovač.

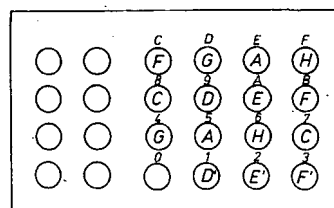
Stejným způsobem jak bylo uvedeno v tomto článku probíhá i výpočet kontrolního součtu v programu MGF v Monitoru, ten však probíhá automaticky a uživatel se s ním nemusí seznamovat. Má vliv pouze na výsledek testu je-li program nahrán správně nebo špatně.

11.5 Další jednoduché programy

K dispozici je celá řada jednoduchých ověřených programů pro různé hry publikované v AR. Např. hra NIM, HLEDÁNÍ BOMBY, POSTŘEH apod.

Dále byly vypracovány programy pro dekadické aritmetické operace s libovol-

061B	23		INX H	
0610	7E		MOV A, M	
061D	D30B		OUT 0B H	
061F	D30D		OUT 0D H	
0621	D30F		OUT 0F H	
0623	23		INX H	
0624	C9		RET	
0625	1100F0	ZPOZ:	LXI D, F000H	zpoždění
0628	1B	DC:	DCX D	
0629	7A		MOV A, D	
062A	C600		ADI 0H	
062C	C22806		JNZ DC	
062F	C9		RET	
0630	6394		DW ORNAM1	
0632	9463		DW ORNAM2	
0634	FF		DB FFH	konec
			END	



Obr. 26. Hexadecimální klávesnice jako varhany

nou délkou operandů v pevně řádové čáře apod. S těmito programy i s jinými si však uživatel jistě poradí sám.

12. Použití mikropočítače pro řízení periférií

Jako další periférie byly k mikropočítači napojeny snímač DP FS 330, rozhlasový přijímač (generování melodie), programátor PROM pro programování paměti 74S287 a 74188, deska pro testování IO a další. Některé případy si alespoň částečně probereme.

12.1 Generování melodie

Jako periférie je použit nf zesilovač např. z rozhlasového přijímače. Na jeho vstup je přiveden přes odpor (zmenší úroveň napětí) a kondenzátor (oddělí ss složku) signál A 10 z adresové sběrnice. Pozn.: Při pokusech bylo zjištěno, že není třeba počítač s přijímačem propojovat, ale stačí přijímač pouze zapnout, nastavit dlouhé vlny a naladit na stupnici místo, kde není žádný vysílač. Přijímač musí být umístěn co nejbližší u počítače.

Program pro generování melodie je v tabulce 28. Tento program hraje melodii uloženou od adresy 0700H v pořadí výška a délka tónu. Výška tónu je dána tabul-

Tab. 27. Program výpočet kontrolního součtu

adresa	kód	návěš	instrukce	komentář
0900	318009		ORG 0900H	
0903	210000		LXI SP, 0980 H	
0906	CD7600		CALL POM	podprogram Monitoru
0909	AF		XRA A	
090A	CD9501		CALL ZOBR	podprogram Monitoru
090D	CD3309		CALL DZN	
0910	5F		MOV E, A	
0911	CD6701	Q:	CALL ZNAK	podprogram Monitoru, po příjmutí oddělovače Q probíhá výpočet
0914	FE60		CPI 60H	
0916	C21109		JNZ Q	
0919	7B		MOV A, E	
091A	85		ADD L	
091B	84		ADD H	
091C	4F		MOV C, A	
091D	AF		XRA A	
091E	BB		CMP E	
091F	79		MOV A, C	
0920	CA2D09		JZ KON	
0923	0600		MVI B, 0H	
0925	46	DB 1:	MOV B, M	
0926	80		ADD B	
0927	23		INX H	
0928	1D		DCR E	
0929	C22509		JNZ DB1	
092C	2B		DCX H	
092D	2F	KON:	CMA	konec
092E	3C		INR A	
092F	CD9501		CALL ZOBR	podprogram Monitoru
0932	76		HLT	
0933	CD4509	DZN:	CALL TZN	dva znaky (BYTE)
0936	CD6101		CALL ROT	podprogram Monitoru
0939	CD9501		CALL ZOBR	podprogram Monitoru
093C	47		MOV B, A	
093D	CD4509		CALL TZN	
0940	80		ADD B	
0941	CD9501		CALL ZOBR	podprogram Monitoru
0944	C9		RET	
0945	D5	TZN:	PUSH D	testovaný znak
0946	C5		PUSH B	
0947	CD6701		CALL ZNAK	podprogram Monitoru
094A	C1		POP B	
094B	D1		POP D	
094C	FE10		CPI 10H	
094E	D8		RC	
094F	33		INX SP	
0950	33		INX SP	
0951	C35801		JMP CHYBA	podprogram Monitoru
0954	C9		RET	
			END	

➔ **kou 29.** Příklad písničky je v **tabulce 30**. Nutno podotknout, že tabulky 29, 30 platí pro použitý krystal 5 MHz. Pro jiný krystal je třeba počítač naladit podle hudebního nástroje. Program se startuje od adresy 0800H.

12.2 Program Varhany

Přijímač připojíme stejně jako v čl. 12.1. Po spuštění programu **tab. 31** od adresy 0800H, slouží klávesnice jako klávesnice varhan. Ladění kláves je zřejmé z **obr. 26**.

12.3 Napojení snímače děrné pásky

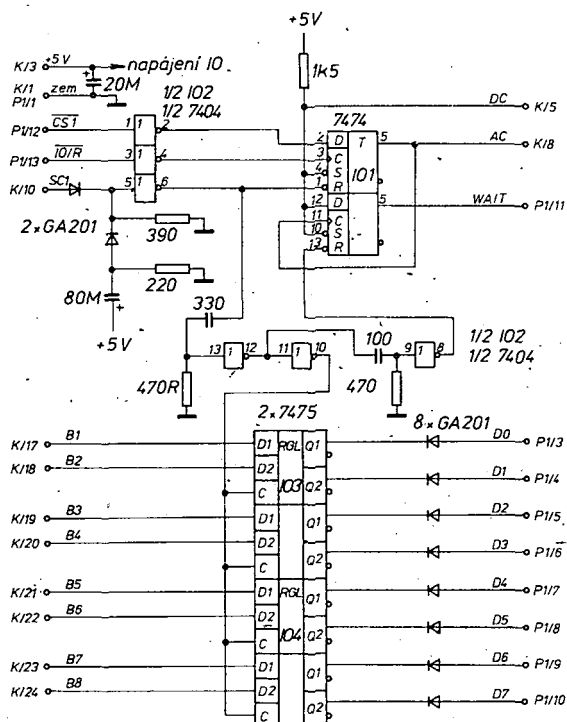
Snímač DP (byl použit typ FS 330) se připojí k mikropočítači přes interfejs (**obr. 27**), přes konektor P1 (PORT 01). Mezi interfejsem a snímačem DP je přes konektor K napojen třináctizolový kabel, který končí na konektoru snímače pásky. Mezi interfejsem a snímačem DP jsou vedeny signály B1 až B8 (8 stop DP), zem, +5 V, SC1 (znak přítomen na výstupu ze snímače DP), DC (spuštění motoru snímače DP), AC (start snímače DP). Protože je snímač DP pomalejší než počítač, generuje interfejs signál WAIT. Je-li WAIT = H, počítač čeká na snímač DP. Činnost obvodu je

Obr. 27. Schéma zapojení interfejsu snímače pásky FS330. Uspořádání konektoru K pro snímač pásky FS330 může být libovolné, P1 je dáno zapojením portu 1 na BOB-85.

velmi jednoduchá. Přímoý příkaz ke čtení ze snímače vznikne požadavkem IN 01. Tím počítač uvede dekodovaný signál z adresové sběrnice CS 1 do L a později i signál řídicí sběrnice IO/R do L. Tím se přes IO1 generuje signál WAIT (počítač se zastaví) a AC (rozběhne se snímač DP). Po příchodu znaku (SC1 = H) na B1 až B8 je díky SC1 tento zapsán do paměti IO3, IO4 (2×7475), je zrušen signál AC (snímač DP se zastaví) a WAIT (počítač se rozběhne). Dokud nepřijde další instrukce IN 01 není čten další znak. Program pro čtení DP ve tvaru HEX je v **tab. 32**. Ten přečte celou DP a kontrolu správnosti nahrávání. Startuje se od adresy 0600H. Po správném nahrání svítí na displeji . Při chybě je na displeji adresa, kde došlo k chybě a E1 (chyba parity), E2 (nehexadecimální znak), E3 (špatný kontrolní součet). Po chybě lze pokračovat tlačítkem NEXT. Program ignoruje prázdný a plně děrovaný znak.

Tab. 28. Program Generování melodie

adresa	kód	návěš	instrukce	komentář
07F0	41	OTON:	ORG 07FOH	
07F1	1B	OKMIT:	MOV B, C	výška tónu do B
07F2	7A		DCX D	
07F3	B3		MOV A, D	
07F4	CA0308		ORA E	
07F7	05		JZ ZAC	
07F8	C2F107		DCR B	
07FB	C30908		JNZ OKMIT	
07FE	00		JMP TON	
07FF	00		NOP	
0800	310007	START:	NOP	
0803	C1		LXI SP, SOLO	SOLO – místo uložení písničky
0804	0D		POP B	výška C, délka B
0805	CA0008	KONEC:	DCR C	
0808	50		JNZ START	opakování písničky
0809	41	TON:	MOV D, B	délka do D
080A	1B		MOV B, C	výška do B
080B	7A	KMIT:	DCX D	DE – 1 do DE
080C	B3		MOV A, D	
080D	CA0308		ORA E	
0810	05		JZ ZAC	
0811	C20A08		DCR B	
0814	C3F007		JNZ KMIT	
			JMP OTON	
			END	



12.4 Testování integrovaných obvodů

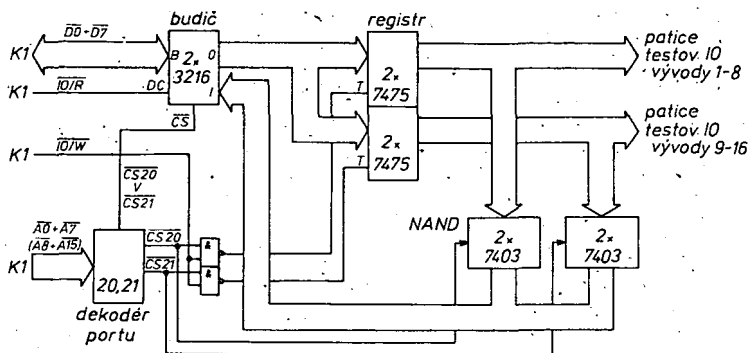
Při testování obvodů se k mikropočítači napojí jednoduchý interfejs přes K1. Blokové schéma je na **obr. 28**. Na desce je použito 13 IO.

Princip je velice jednoduchý. Postupně se instrukcemi OUT 20H a OUT 21H (Port 20, 21) přes budič 2×3216 přesunou testovací signály do registrů 4×7475. Ty se objeví jako úroveň H, L na vývodech patice pro testovaný integrovaný obvod. Přes obvody NAND 4×7403 a budič 2×3216 čteme instrukcemi IN20H, IN21H odezvu testovaného IO na vstupní signály a tu programem testujeme.

Zapojení vývodů patice testovaného IO je na **obr. 29**. Přes zdičku je možno připojit napájení (+5 V, ZEM), přičemž omezovací odpor 120 Ω a dioda GA 201 chrání výstup 7475 a vstup 7403. Zapojení je navrženo tak, že na výstupy testovaných IO, nepoužité vstupy a napájení +5 V přivádíme

Tab. 29. Tabulka tónů pro program „Generování melodie“ při použití krystalu 5 MHz

tón	kód	tón	kód
Fis	C1	fis	60
G	B6	g	5B
Gis	AA	gis	55
A	A1	a	51
B	98	b	4D
H	90	h	48
C	87	c	44
Cis	7E	cis	40
D	78	d	3D
Dis	71	dis	3A
E	6C	e	36
F	66	f	33
pomlka	02	konec	01



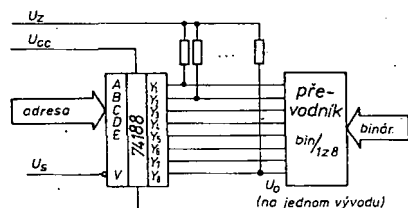
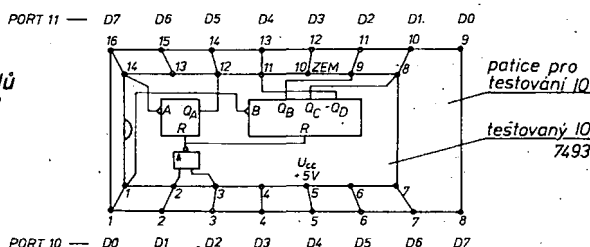
Obr. 28. Blokové schéma desky pro testování IO

Tab. 30. Tabulka písničky Zpívající cikán (amer. lidová) pro program GENEROVÁNÍ MELODIE

adresa	kód	komentář
0 7 0 0	7 8 8 0 5 B 8 0 5 1 8 0 4 8 4 0 5 1 4 0	Poznámka:
0 7 0 A	5 B 4 0 7 8 C 0 0 2 0 2 7 8 F 0 0 2 0 2	1. BYTE = konst. výšky 1. tónu
0 7 1 4	7 8 8 0 5 B 8 0 5 1 8 1 4 8 4 0 4 4 4 0	2. BYTE = konst. délky 1. tónu
0 7 1 E	4 8 F 0 5 1 F 0 7 8 8 0 5 B 8 0 5 1 8 0	3. BYTE = konst. výšky 2. tónu
0 7 2 8	4 8 4 0 4 4 4 0 3 D 4 0 4 8 C 0 5 B C 0	4. BYTE = konst. délky 2. tónu
0 7 3 1	5 1 4 0 4 8 8 0 0 2 0 2 4 8 8 0 5 1 4 0	(2N-1). BYTE = konst. výšky N. tónu
0 7 3 C	5 B 4 0 6 C 8 0 7 8 8 0 5 B F 0 6 C 8 0	2N. BYTE = délky N. tónu
0 7 4 6	7 8 F 0 0 2 F 0 0 1 0 1	01 = konec písne
		konstanty výšky tónů
		a pomlka viz tab. 29
		konstanty délky tónů jsou
		libovolné, podle tempa;
		F0 – celá nota
		C0 – tříčtvrtová
		80 – půlová
		40 – čtvrtová
		20 – osminová
		02 – krátké přerušení tónu
		01 – konec písničky

Obr. 29. Zapojení vývodu patice testovacího IO

Obr. 30. Uspořádání vývodů při testování IO MH7493



Obr. 31. Princip programování paměti 74188

vstupní úroveň H, na zem L. Na využité vstupy přivádíme vhodné kombinace H a L tak, abychom IO co nejlépe otestovali. Současně sledujeme odezvu všech vývodů IO na vstupní kombinace H, L. Každá odezva se porovnává v A-registru mikroprocesoru s hexadecimálním číslem, které má odpovídat příslušné odezvě. Dojde-li ke shodě, pokračuje činnost testování dále. Nenastane-li shoda, je hlášena displejem chyba.

Příklad testování si ukážeme na IO 7493. Nejdříve nakreslíme podle katalogu IO uspořádání vývodů a jejich označení (obr. 30). Dále sestavíme tabulku vstupních a výstupních signálů (tab. 33) podle

Obr. 32. Časový průběh U_{CC} , U_S při programování MH74188 (přesné časové údaje viz [11])

pravidel z předchozího odstavce. Vlastní program (tab. 34) je sestaven tak, že na porty 20 a 21 přivádíme instrukcemi OUT vhodnou kombinaci čísel předem zapsanou do A registru a příslušné odezvy čteme instrukcemi IN.

Celá činnost při testování libovolného IO spočívá v připojení desky (obr. 28) přes konektor K1 k mikropočítači, zasunutí testovaného IO do objímky a připojení napájecího napětí k testovanému IO. Program pro testování se nahraje z magnetofonu do paměti RAM a odstartuje se podle čl. 10.5. Na displeji se zobrazí výsledek testu. Tímto způsobem lze testovat prakticky jakýkoliv IO, pro sestavení testovací-

Tab. 31. Program VARHANY

adresa	kód	návěšt	instrukce
07F0	41	OTON:	ORG 07F0H
07F1	1B	OKMIT:	MOV B, C
07F2	7A		DCX D
07F3	B3		MOV A, D
07F4	CA0008		ORA E
07F7	05		JZ ZAC
07F8	C2F107		DCR B
07FB	C30908		JNZ OKMIT
07FE	00		JMP TON
07FF	00		NOP
0800	CD1708	ZAC:	NOP
0803	1620		CALL JED
0805	6F		MVI D, 20H
0806	2608		MOV L, A
0808	4E		MVI H, 08H
0809	41	TON:	MOV C, M
080A	1B	KMIT:	DCX D
080B	7A		MOV A, D
080C	BB		ORA E
080D	CA0008		JZ ZAC
0810	05		DCR B
0811	C20A08		JNZ KMIT
0814	C3F007		JMP OTON
0817	DB0A	JED:	IN 0AH
0819	C600		ADI 0H
081B	CA1708		JNZ JED
081E	17		RAL
081F	37		STC
0820	1F		RAR
0820	C9		RET
			END
0880	013D3633	TABUL:	
0884	5B514844		
0888	87786C66		
0883	CDB6A190		
			END

tóny
–, D, E, E
G, A, H, C
C, D, E, F
F, G, A, H

ho programu je však nutná dokonalá znalost testovaného IO.

Poznámka: Na podobném principu je možno testovat např. desky s číslicovými IO, kabeláže, IO s větším počtem vývodů než 16 apod.

12.5 Programátor pevných pamětí PROM

Při programování pevných pamětí např. typu 74188 je třeba v zapojení podle obr. 31, zajistit správný postup. Nejdříve se naadresuje příslušná buňka v paměti, dále se připojí U_0 k výstupu toho bitu, který má být programován, ostatní vývody jsou přitom zapojeny přes odpor na U_z . Potom následuje vlastní programování tj. na vývodech V, U_{CC} jsou generovány impulsy podle obr. 32. Nakonec se ještě

Tab. 32. Program Řízení snímače děrné pásky FS 330

adresa	kód	návěšť	instrukce	komentář
0600	31C006	SP:	ORG 0600H	
0603	CD8106	DVT:	LXI SP, ZAS	
0606	D63A		CALL CTI	
0608	C20306		SUI 3A H	
060B	AF		JNZ DVT	
060C	47		XRA A	
060D	CD4A06		MOV B, A	
0610	57		CALL BYTE	
0611	C600		MOV D, A	
0613	CA0006		ADI 0H	
0616	CD4A06		JZ SP	
0619	67		CALL BYTE	
061A	CD4A06		MOV H, A	
061D	6F		CALL BYTE	
061E	CD4A06		MOV L, A	
0621	5F		CALL BYTE	
0622	CD4A06	ZAPIS:	MOV E, A	
0625	77		CALL BYTE	
0626	23		MOV M, A	
0627	15		INX H	
0628	C22206		DCR D	
062B	78		JNZ ZAPIS	
062C	2F		MOV A, B	
062D	3C		CMA	
062E	47		INR A	
062F	CD3E06		MOV B, A	
0632	B8		CALL BYTE1	
0633	C47C06		CMP B	
0636	AF		CNZ CH3	
0637	BB		XRA A	
0638	CA0006		CMP E	
063B	C34200		JZ SP	skok do Monitoru
063E	CD5206		JMP MONIT	
0641	CD6101		CALL HEX	podprogram Monitoru
0644	4F		CALL ROT	
0645	CD5206		MOV C, A	
0648	81		CALL HEX	
0649	C9		ADD C	
			RET	

064A	CD3E06	BYTE	CALL BYTE 1	
064D	4F		MOV C, A	
064E	80		ADD B	
064F	47		MOV B, A	
0650	79		MOV A, C	
0651	C9		RET	
0652	CD8106	HEX:	CALL CTI	
0655	E67F		ANI 7F H	
0657	D630		SUI 30H	
0659	DC6906		CC CH2	
065C	FE0A		CPI 0AH	
065E	D8		RC	
065F	D607		SUI 07H	
0661	FE07		CPI 07H	
0663	DC6906		CC CH2	
0666	FE10		CPI 10H	
0668	D8		RC	
0669	3EE2	CH2:	MVI A, E2	
066B	CD9501	CH:	CALL ZOBR	podprogram Monitoru
066E	CD9006	CZ1:	CALL ZN	
0671	FE60		CPI 60H	
0673	C26E06		JNZ CZ1	
0676	3E0A		MVI A, 0AH	
0678	CDF900		CALL TMA	podprogram Monitoru
067B	C9		RET	
067C	3EE3	CH3:	MVI A, E3 H	
067E	C36B06		JMP CH	
0681	DB01	CTI:	IN 01H	
0683	FEFF		CPI FFH	
0685	CA8106		JZ CTI	
0688	FE00		CPI 0H	
068A	CA8106		JZ CTI	
068D	C600		ADI 0H	
068F	E8		RPE	
0690	3EE1	CH1:	MVI A, E1H	
0692	C36B06		JMP CH	
0695	C5	ZN:	PUSH B	
0696	D5		PUSH D	
0697	CD6701		CALL ZNAK	podprogram Monitoru
069A	D1		POP D	
069B	C1		POP B	
069C	C9		RET	
			END	

Tab. 33. Vstupní a výstupní signály testu IO MH 7493

FUNKCE	Vstupní signály												Výstupní signály												POZN. pro program												
	PORT 10								HEX	PORT 11								HEX	PORT 10								HEX	PORT 11								HEX	
	7	6	5	4	3	2	1	0		7	6	5	4	3	2	1	0		7	6	5	4	3	2		1	0		7	6	5	4	3	2	1	0	
napájení	1									0									1									0									příprava
výstupy																																					
nevyuž. vývody	1	1	1	1						1	1	1	1	1	1				1	1	1	1					1										
nulování																																					
zrušení nulov.																																					
T1 pro vstupy A, B																																					
T2 pro vstup B																																					
T3 pro vstup B																																					
T4 pro vstup B																																					
T5 pro vstup B																																					
T6 pro vstup B																																					
T7 pro vstup B																																					
nulování																																					
zrušení nulování																																					

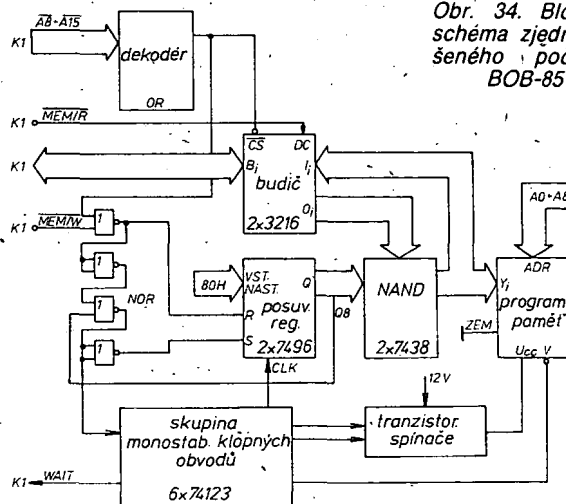
ozn. bitů ☐ vstupy
☐ výstupy

význam poznámek: OUT 10 vyšli vstup data na PORT 10
 IN 11, 6ti výstupní data z PORT 11

Tab. 34. Program TESTOVÁNÍ MH 7493

adresa	kód	návěš	instrukce	poznámka
0600	3EFF	START:	ORG 0600H	nulování
0602	D310		MVI A, FFH	
0604	3EF7		OUT 10H	
0606	D311		MVI A, F7H	
0608	3EFD		OUT 11H	
060A	D310		MVI A, FDH	
060C	D311		OUT 10H	
060E	DB11		IN 11H	
0610	FEC1		CPI C1H	
0612	C25801		JNZ CHYBY	
0614	3EFC	CYKL:	MVI A, FCH	T1
0616	D310		OUT 10H	
0618	3E77		MVI A, 77H	
061A	D311		OUT 11H	
061C	3EFD		MVI A, FDH	
061E	D310		OUT 10H	
0620	3EF7		MVI A, F7H	
0622	D311		OUT 11H	
0624	0606		MVI B, 06H	
0626	3EFC		MVI A, FCH	
0628	D310		OUT 10H	T2, T7
062A	3EFD		MVI A, FDH	
062C	D310		OUT 10H	
062E	05		DCR B	
0630	C22506		JNZ CYKL	
0632	DB11		IN 11H	
0634	FEF7		CPI F7H	
0636	C25801		JNZ CHYBA	
0638	3EFF		MVI A, FFH	
063A	D310		OUT 10H	
063C	3EF7		MVI A, F7H	nulování
063E	D311		OUT 11H	
0640	3EFB		MVI A, FBH	
0642	D310		OUT 10H	
0644	DB11		IN 11H	
0646	FEC1		CPI C1H	
0648	C25801		JNZ CHYBA	
064A	C7		RST 0	
064C			END	

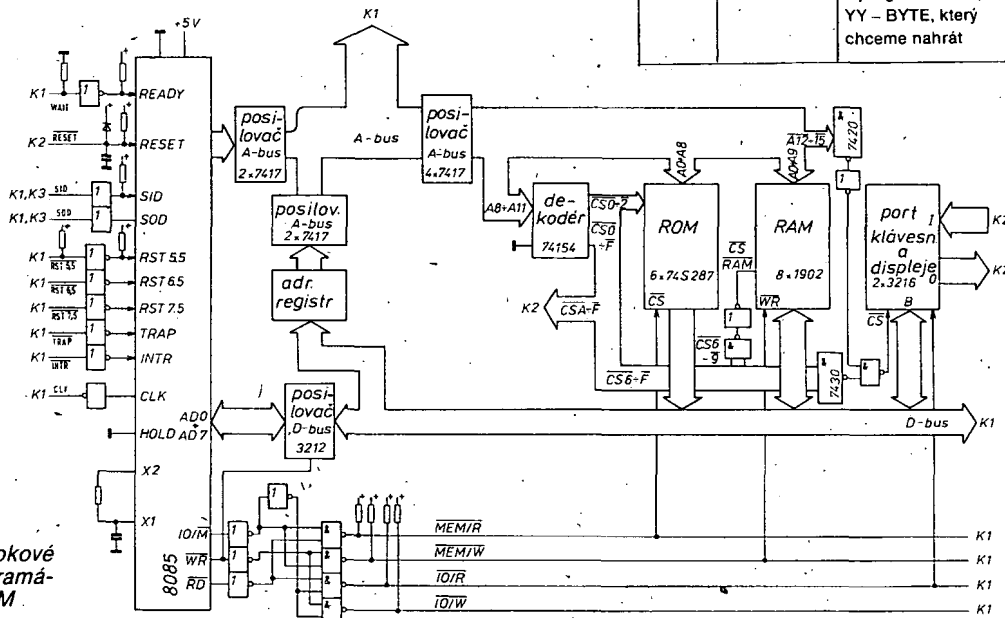
Obr. 34. Blokové schéma zjednodušeného počítače BOB-85



Tab. 35. Ukázka části programu PROM (pro nahrání 1 BYTE)

návěš	instrukce	komentář
PROM:	LXI H, FFXH	HL reg. naplněn FFXH
	MVI A, YY H	A reg. naplněn YY H
	MOV B, A	uložení YYH do B registru
	MOV M, A	YYH do PROM, programování ve stavu WAIT
	MOV A, M	nahrání YYH do A reg.
	CMP B	porovnání nahaného, a zadaného BYTE
	JNZ CHYBA	signalizace chyby při špatném nahrání
		pokračování programu po správném nahrání
		XX – adresa v programované paměti
		YY – BYTE, který chceme nahrát

Obr. 33. Blokové schéma programátoru PROM



kontroluje správnost programování stejným způsobem jako při čtení z paměti [11].

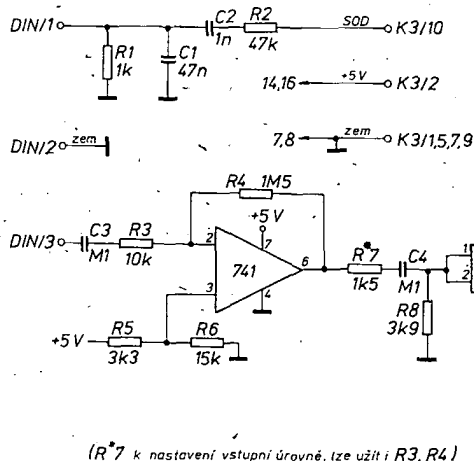
Blokové schéma zařízení, které umožňuje programování paměti 74188, 74S287, je na obr. 33. Zařízení je připojeno k počítači přes konektor K1 jako paměť na adresách FF00H až FFFFH. Řídící signály jsou MEM/W a MEM/R. Po dobu programování 1 bajtu čeká počítač na programátor (je zastaven signálem WAIT). Adresování se provádí přes adresovou sběrnici A₀ až A₁₀. Výběr právě programovaného bajtu provádí programátor přepí-

náním datové sběrnice při stavu WAIT. K desce je navíc nutno připojit napájení +12 V pro zajištění průběhu napětí U_{cc}.

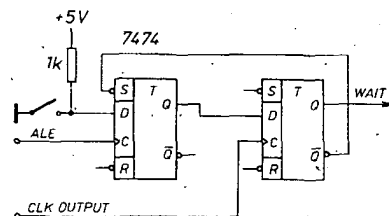
Zařízení je řízeno mikropočítačem. Pro nahrání jednoho bajtu je program velice jednoduchý (tab. 35). Při instrukci MOV A, M je naadresován bajt v programové paměti z H, L registrů a uvolněn budič CS = L. Tím se programovaný bajt dostane přes budič až na vstup obvodu NAND. Dále vysílá mikropočítač řídicí signál MEM/W. Tím je nulován a přednastaven posuvný registr číslem 80H = 1000 0000. První hranou signálu MEM/W se rozbíhá

činnost monostabilních klopných obvodů. Ty plní následující úkoly:

1. po celou dobu programování vysílají signál WAIT (zastavení 8085)
2. v průběhu programování každého bajtu generují řídicí signály pro tranzistorové spínače, které generují signály U_{cc}, U_s podle obr. 32.
3. přes signál CLK je postupně vysláno 8 impulsů, které postupně posouvají přednastavenou log. 1 posuvným registrem a tím se při programování každého bajtu přes obvody NAND aktivuje výstup programované paměti Y_i.



Obr. 35. Interfejs magnetofonu.



ALE, CLK OUTPUT, WAIT - signály 8085
nepoužité vstupy 7474 S, R nutno ošetit

Obr. 36. Obvod pro krokování 8085. (ALE, CLK OUTPUT a WAIT jsou signály 8085. Nepoužité vstupy 7474 S, R je nutno ošetit)

Upravené zapojení je na obr. 35. Typy pasivních součástek jsou stejné jako v celém mikropočítači tj. miniaturní.

Kromě návodu ke stavbě mikropočítače je poukázáno i na jeho použití. Jsou předvedeny ukázky jednoduchých programů a řízení jednoduchých, ale účelných periferních zařízení.

Součástí mikropočítače není ještě jednoduchý obvod pro krokování programem (nebyl potřebný), proto alespoň v závěrečné části je uveden podle [4] viz obr. 36.

Nakonec mi ještě zbývá poděkovat ing. Stanislavu Benešovi a ing. Jiřímu Konečnému za cenné rady a připomínky i za poskytnutou literaturu.

Po ukončení programování jednoho bajtu je zrušen stav WAIT. Následující instrukce MOV A, M (aktivován MEM/R) si mikroprocesor přečte obsah právě naprogramovaného bajtu. Podrobné schéma programátoru a jeho popis nebude uveden, neboť by si zasloužil samostatný článek.

13. Zjednodušení pro amatérský mikropočítač

Po zkušenostech se sestavou mikropočítače mohou říci, že v amatérských podmínkách zcela vyhovuje jak svým rozsahem paměti, tak napojenými perifériemi (klávesnice, displej, magnetofon). Nyní ještě zbývá maximálně snížit náklady na stavbu mikropočítače, tzn. provést taková zjednodušení, která neuberou na funkci mikropočítače ani periférii a přitom mají na cenu velký vliv. Výhodná jsou také energetická zjednodušení, např. IO 2x3216 nahradit 3212. Po níže uvedených zjednodušeních bude možné, aby se celý mikropočítač včetně klávesnice, displeje a interfejsu vešel na jednu stejně velkou desku PS.

Zjednodušení BOB-85:

1. Ukázalo se, že není nezbytné používat na desce mikropočítače porty 01 až 06H. Tím odpadne 12 x 3216 a dekodér 7442.
2. Budič datové sběrnice 2 x 3216 je energeticky i cenově výhodnější nahradit obvodem 3212.
3. Adresové a datové sběrnice není třeba používat jako negativní, tím se zjednoduší napojení adresové sběrnice a odpadnou IO 2 x 3226.
4. Dekodér pro paměť a porty klávesnice a displeje 2 x 7442 lze nahradit jedním obvodem 74154.
5. Signál HOLD, umožňující přímý přístup do paměti je možno zrušit, neboť v amatérských podmínkách se nepředpokládá velká rychlost přenosu dat přímo do paměti. Proto se může také zrušit blokování adresové a datové sběrnice signálem HLDA, obvody 4 x 7438 lze nahradit 3 x 7406.
6. Stačí jeden vstup signálu WAIT (pouze přes K1).
7. Pokud nevyžadujeme velkou přesnost hodinového kmitočtu, můžeme krystal 5 MHz nahradit odporem 10 kΩ (nelze použít např. program HODINY).
8. Pokud by se podařilo získat jinou statickou paměť RAM, než 8 x MHB 2102 A o formátu 1024 x 8 bitů s men-

ším počtem čipů (např. K541RU2, 2114, 2142 2 čipy, 4118 1 čip apod.), byly by nároky na cenu, odběr proudu ze zdroje i prostorové uspořádání podstatně nižší.

Z těchto zjednodušení si jistě každý amatér vybere to, které nejvíce vyhovuje jeho potřebám. Příklad blokového zapojení zjednodušeného mikropočítače BOB-85 je na obr. 34. Nakreslené výstupy mikroprocesoru nejsou zapojeny.

Sečteme-li počet IO zjednodušeného mikropočítače (35), klávesnice a displeje (19 - nepočítají se prvky LQ 410) a interfejsu magnetofonu (3), dospějeme k číslu 57. To je již počet, který se celý vejde na stejnou desku plošného spoje, na jaké je nyní pouze mikropočítač (9 x 7 = 63 IO).

14. Závěr

Účelem tohoto článku bylo seznámit čtenáře se zajímavým typem mikroprocesoru. Mikroprocesor byl nejen popsán, ale byla stručně, pokud možno srozumitelnou formou, vysvětlena i jeho činnost. Srovnáme-li vlastnosti mikroprocesorů 8085 a 8080, jeví se 8085 jako podstatně výhodnější a přitom programově ekvivalentní.

Dále je vysvětlen princip mikropočítače a jeho napojení na řadu užitečných periférií.

Součástí článku je návod ke stavbě jednoduškového mikropočítače s mikroprocesorem 8085, klávesnicí s displejem a interfejsu magnetofonu. Podrobně je probrán řídicí program Monitor. V závěrečné části jsou uvedena určitá zjednodušení, která umožní postavit jednoduškový mikropočítač s klávesnicí, displejem a interfejsem magnetofonu na pouhé jedné desce o rozměrech asi 250 x 180 mm.

Na začátku roku 1983 byl v AR B1 popsán mikropočítač Intelka se zajímavým připojením magnetofonu přes vstup „mikrofon“ a výstup „sluchátka“ podle mikropočítače ZX-80. Zapojení jsem odzkoušel a po menších úpravách jsem byl s výsledkem velmi spokojen. Kvalita přenosu velmi málo závisí na úrovni zesílení nf signálu. Připojení k magnetofonu je provedeno pouze jednou šňůrou přes vstup „radio“ a výstup „magnetofon“. V obvodu pro vstup do magnetofonu byly změněny hodnoty součástek přímé větve (jiná úroveň vstupního napětí pro vstup „mikrofon“ a „radio“ při zachování stejné časové konstanty). V obvodu pro výstup je signál nejdříve zesílen (MAA 741 v zapojení s nesymetrickým napájením), střední část je ponechána bez úprav a dále je přidán monostabilní klopný obvod ze stejného důvodu jako je popsáno v čl. 8.

15. Literatura

- [1] Jugel, A.: Mikroprozessorsysteme, Berlin 1978.
- [2] Partyk, P., Machačka, I.: Základní instrukce mikroprocesoru 8080. Tesla Promes 1980.
- [3] Mikuláš, J.: Sběrka cvičných programů v jazyku Assembler 8080. Tesla Promes 1980.
- [4] Katalog Intel MCS 80/85. Family users manual 10/1979.
- [5] Katalog polovodičových součástek. Tesla Rožnov 1981.
- [6] Zásady použití mikroprocesoru 8080. ST 5/1980.
- [7] Mikroprocesorový systém 8080. Tesla Piešťany P-ČSVTS 1979.
- [8] Mikrorechner mit 8085 A-Neu von Rohde-Schwarz 88.
- [9] Dědina B., Valášek P.: Mikroprocesory a mikropočítače. SNTL-Praha 1981.
- [10] Mercl, J.: Mikropočítač PIP-2. AR B3/1980.
- [11] Bipolární programovatelná paměť MH 74188. Tesla Rožnov 1978.

Všem čtenářům

Desky s plošnými spoji

konstrukcí, zveřejněných v časopise AR (řada A, B i konstrukční příloha), zasílá výhradně na dobírku

Radiotechnika ÚV Svazarmu expedice plošných spojů Žižkovo nám. 32 500 21 Hradec Králové

Na této adrese si můžete objednat všechny desky plošných spojů sérií L až S (s výjimkou desek dvoustranných). Objeďte pište na korespondenčním listku. Vaše objednávka bude podle možnosti vyřízena co nejdříve (v případě desek, o něž je mimořádný zájem do 3 měsíců).

Osobně si můžete desky plošných spojů zakoupit v prodejně podniku Radiotechnika:

Radioamatérská prodejna Budečská 7 120 00 Praha 2; tel. 25 07 33