

MICROBÍZ

CP/M pro ZX SPECTRUM

SPOLEČNÁ SLUŽBA
AMATÉRSKÉHO RADIA
PRO UŽIVATELE
MIKROPOČÍTAČŮ

WB
AR-602

O P E R A Č N Í S Y S T É M

CCC	PPPP	/	M	M	222	222			
C	C	P	P	M	M	2	2	2	2
C		P	P	MM	MM	2	2		
C		PPPP		M	M	2	2		
C	C	P		M	M	2	..	2	
CCC	P	/	M	M	22222	..	22222		

Implementace na ZX Spectrum

Jiří Lamac

602. ZO Svazarmu

Praha, březen 1988

OBSAH:

		str.
1.	Úvod	3
2.	Užívání operačního systému	5
2.1	Co to je operační systém	5
2.2	Vznik CP/M	6
2.3	Základní informace o CP/M	8
2.4	Interpretr příkazů CCP	14
2.5	Standardní služby BDOSu	16
2.6	Modul BIOS	26
3.	Implementace CP/M na ZX Spectru -popis BIOSu	30
3.1	Zavedení systému	30
3.2	Obsluha klávesnice	32
3.3	Obsluha obrazovky	33
3.4	Tiskárna a děrná páska	34
4.	Diskové operace	35
4.1	Formát záznamu na mechanikách Microdrive	37
4.2	BIOS 4.0	41
4.2.1	Výběr mechaniky	41
4.2.2	Operace čtení	42
4.2.3	Operace zápisu	43
4.2.4	Práce s přístupovými buffery	43
4.3	BIOS 3.3 - odlišnosti od BIOSu 4.0	48
5.	Základní programové příslušenství	50
5.1	Program FORMAT	50
5.2	Programy MLOAD a MSAVE	53
6.	Než začnete se systémem pracovat	57
6.1	Využití programu z jiného počítače	60
6.2	Instalace tiskárny uživatelem	62
6.3	Instalace systému pro 80 kB RAM	66
7.	Rozšíření paměti RAM	67
8.	Literatura	71

1. Úvod

ZX Spectrum je dnes u nás nejrozšířenějším počítačem, a to nejen mezi amatéry. Je však pravda, že valná většina jeho majitelů ponejvíce jen hraje různé hry, nebo píše jednoduché programy v BASICu. Je to zaviněno nedostatkem osvěty, úplnou absencí odborné literatury a také tím, že v záplavě profesionálních programů tyto hry mají převahu. Kvalitních systémových a uživatelských programů mnoho není. A tak ti, kteří by třeba i chtěli dělat něco "lepšího", nemají možnost. Většina profesionálních programů je určena pro práci s kazetovým magnetofonem, což je zařízení patřící již minulosti, vhodné snad jen k zálohování dat. Hlavním nedostatkem ZX Spectra je však to, že neumožnuje přímo, bez pomocných programů, pracovat se sekvenčními datovými soubory.

Popisovaný operační systém tyto nedostatky beze zbytku řeší. Je možný jednoduchý přístup k souborům, lze užívat kvalitní programy firem zvučných jmen, jako jsou překladače snad všech významných programovacích jazyků (TURBO Pascal, ADA, BASIC, FORTRAN, COBOL, C, PROLOG, FORTH, LISP), textové editory (WordStar, Word Master), databázové programy (dBASE II), programy pro tabulkové výpočty (MULTIPLAN, SuperCalc), makroassembly a dokonce i křízové assembly pro jiné mikroprocesory. Důležitý je také fakt, že Spectrum získá kompatibilitu s jinými osmibitovými počítači u nás, jako je např. TNS, SAPI 1, Robotron 1715, SHARP MZ-800, Schneider CPC 664 a 128, ale i Commodore 128 a další.

Je pravdou, že u Spectra se původně s tvorbou operačního systému vůbec nepočítalo, vždyť byl konstruován spíše jako hračka pro výuku dětí. Když asi před dvěma roky prohlásil J. Lamač v kruhu několika přátel, že pro Spectrum CP/M napíše, říkal to tak napůl v žertu a nikdo mu ani moc neveril. A to vlastně vůbec nikdo netušil, co tahle "legrace" obnáší. Když však, asi po roce úporné práce, předvedl zcela "chodivou" verzi 3.3, pochopili jsme, že tohle už není hračka, ale opravdový počítač. Asi za půl roku pak přišlo dalsí překvapení - verze 4.0.

CP/M na Spectru má své zvláštnosti a úskalí. Především mechanika na pružné disky není ve standardním příslušenství pro počítače Sinclair. Nehledě na to, že disketový řadič se poměrně jednoduše programově ovládá a není tudíž tak velký problém napsat CP/M pracující s disketou. Programy pro čtení a zápis sektoru na disk mohou mít délku okolo 500 byteů. Majitelů disketových mechanik je však poměrně málo; rozhodně nejrozšířenější je ZX Microdrive, který dokonce před časem dovezl též PZO Tuzex. Proto i první implementace CP/M na ZX Spectrum pracuje s Microdrive, přestože koncepcně toto médium není příliš na úrovni. Pochoptitelně budou v dalsí etapě zpracovány a publikovány i diskové verze pro známé řadiče Western Digital.

Brzy po začátku vývoje CP/M s Microdrive se ukázalo,

že je vyloučené použít původní rutiny v přídavné ROM, ani není možné přímo provádět požadované operace, protože při náhodném přístupu k více souborům je Microdrive velmi pomalý. Byl vytvořen zcela nový programový ovladac Microdrive o délce asi 5kB, který je založen na skutečnosti, že obsluha mechaniky se děje pomocí přerušení. Nedochází tudíž k pozastavování běhu hlavního programu, nýbrž vykonávání diskových operací běží paralelně jako samostatná úloha. Činnost Microdrive se tím v některých případech asi desetkrát až stokrát urychluje a podstatně méně se s mechanikou točí. Tím se výrazně prodlouží její životnost a zvýší spolehlivost. CP/M je asi jediný program pro Spectrum, který důsledně využívá sdílení času.

Těchto vlastností je dosaženo, kromě netradiční programové obsluhy Microdrive, též rozšířením operační paměti RAM, která slouží jednak jako pracovní prostor, ukládají se do ní systémové informace a je využita jako velmi rychlé médium k záznamu souborů, tj. RAM disk. Rozšíření paměti může být bud na 272 kB, nebo 528 kB pro rychlou verzi 4.0. Verzi 3.3 postačí k práci i 80 kB RAM. Ve spojení s některými programy však pracuje dosti pomalu. Po vzniku cen RAM 256 kB připravujeme variantu verze 4.0, schopnou práce s 2 krát 64 kB RAM, při plné rychlosti, ovšem bez RAM disku. Vše vzájemně programově slučitelné.

Pozor ! Verze CP/M pro 272 a 528 kB je schopna plnohodnotné funkce i bez Microdrive, v nouzí jí postačuje i samotný kazetový magnetofon !

Operační systém, i když je implementován na jednoduchém počítaci, jako je Spectrum, je mocným nástrojem. Počítac, určený původně převážně pro televizní hry, se rázem dostává do kategorie blízké profesionálním počítacům. Na úrovni zdrojových textů je dokonce kompatibilní s IBM PC. Nabízí se zde užitečná možnost psát programy doma na Spectru a v práci je pak přenést do IBM PC nebo jiného počítace. Jinou výhodou je, že operační systém MS DOS, použitý v IBM PC, vývojově vysel z CP/M a proto uživatelům CP/M nebude prechod na MS DOS dělat žádné potíže. Každý, kdo viděl pracovat verzi 4.0, byl překvapen neobyčejnou pružností systému, pracujícího s několika buffery. Např. zavedení programu TURBO Pascal trvá asi sekundu. Nebo smazání souboru trvá kolem 0.2 s proti původním 15 sekundám. Celý tento návod byl napsán v textovém editoru WordStar, a to jako jeden soubor. Práce na systému trvala téměř jeden rok. Současná poslední verze vychází z řady verzí předešlých, mnoha praktických zkoušek, testů a všelikého čarování s mechanikami Microdrive.

Předem děkujeme všem uživatelům tohoto systému za případné konstruktivní návrhy a připomínky.

(c) 1988 Jiří Lamač, autor BIOSu
Jakub Vaněk, autor zapojení RAM
Daniel Meca, spoluautor textu

2. Užívání operačního systému

Kapitola 2 se týká operačního systému CP/M obecně, nezávisle na popisované implementaci, protože ti, kteří CP/M již znají odjinud, mohou pokračovat kapitolou 3.

Zároveň doporučujeme těm, kteří hodlají k systému přistupovat jen jako uživatelé (a to je správný postoj), aby přeskočili kapitoly 2.5 a 2.6 popisující BDOS a BIOS. Tyto části, převzaté z [7], jsou potřebné při volání systému ze strojového kódu a v tomto textu jsou zahrnuty spíše pro úplnost.

2.1 Co to je operační systém

Malé osobní počítače, takové jako je např. Spectrum, nemají skutečný operační systém, i když se to někdy mylně uvádí. Je téměř pravidlem, že bývají vybaveny pevnou pamětí ROM o kapacitě 16 až 48 kB, ve které bývá umístěn interpret jazyka BASIC, případně ještě hexadecimální monitor a textový editor. O nějakém standardním systému ani známky. Není pak divu, že člověk, který nikdy nepracoval na větších počítačích (např. SMEP), si o operačním systému, případně o sdílení času, může leda nechat zdát.

O co tedy vlastně jde. Učená definice říká, že OPERAČNÍ SYSTÉM je souhrn určitého množství programů, které přesné definovaným způsobem zajistují komunikaci mezi hlavním programem a uživatelem. Je to jakýsi mezičlánek, který umí zpracovat informace dodané mu hlavním programem a zároveň obsluhuje periférie (to by zpravidla neměla být starost hlavního programu). U větších počítačů se systém stará též o přidělování strojního času a paměti různým současně běžícím úlohám.

Řeknete si, vždyť to není nic nového. Obsluha klávesnice a obrazovky je přece v pevné paměti každého, i toho nejménšího počítace. Ale ouha, když si na Spectrum přenesete program třeba z Robotronu 1715, těžko asi bude pracovat. Tady už to nikomu divné nepřijde, vždyť je to úplně jiný počítač. Když ale oba počítače mají podobný mikroprocesor, mělo by to přece nějak jít. A jsme u toho.

Každý počítač obsluhuje určité periférie, které se obecně ovládají různě (jen kolik je např. různých videoprocesorů). Operace prováděné s perifériemi však jsou stále stejné - "přečti znak", "zapiš znak" a u počítačů s disky ještě "přečti z disku" a "zapiš na disk". Jestliže dohodneme nějaký způsob, jakým budou programy volat obslužné podprogramy, mohou být programy bez omezení přenosné mezi naprostě odlišnými počítači. Operační systém CP/M zde predstavuje takový celosvětově dohodnutý (a na osmibitových počítačích nejvíce rozšířený) standard, definující toto propojení.

2.2 Vznik CP/M

CP/M (Control Program for Microcomputers) je v současné době pravděpodobně nejpopulárnější operační systém pro osmibitové mikropočítace s mikroprocesorem Z80, 18080 nebo 8085. Přes 300 firem vyrábějících mikropočítace dodává CP/M jako součást standardního programového vybavení a odhaduje se, že CP/M je využíván asi na 800 000 pracovistech.

CP/M je sice již starší, ne však zastaralý systém. Jeho první verze byla vytvořena již v roce 1974, jako důsledek značného rozšíření mikroprocesoru Intel 8080. Do té doby totiž byla v programovém vybavení pro tento typ mikroprocesoru značná nejednotnost. Otcem systému je Garry Kildall, původně programový konzultant u firmy Intel. Firma uvažovala o vytvoření standardního systému pro I8080. Pracovalo na něm více programátorů, avšak po rychlém vzestupu firmy byly softwarové problémy odsunuty do pozadí pozornosti (jak se to bohužel běžně dělá u nás). G. Kildall se s firmou Intel nepohodl, navíc bylo konstatováno, že jeho výtvar nemá žádné zvláště vlastnosti, především mu chybí nepřeberné množství rozličných příkazů, jako mají jiné bohatší a honosnější systémy. G. Kildall však věděl, že v jednoduchosti je síla, a pracoval na dalším vývoji CP/M na vlastní pěst. Zajímavé je, že k pozdějšímu bezkonkurenčnímu rozšíření CP/M došlo i přes nezájem firmy Intel.

V téže době přišel John Torode, zaměstnanec firmy Digital Systems, s řadicem mechaniky pro pružné disky, který umožnil perfektní spolupráci CP/M s diskovou pamětí. Předvádění prototypové verze CP/M mělo značný ohlas mezi nadšenými vyznavači mikropočítacových systémů a zájem projevily i nově vznikající firmy produkující mikropočítacové stavebnice a systémy. Vznikla tak potřeba instalovat operační systém CP/M na různé mikropočítacové systémy.

V letech 1974 až 1976 přepracoval G. Kildall CP/M tak, že ze systému vycílenil části závislé na konkrétním technickém prostředí. Systém se tak rozdělil na technicky závislou a technicky nezávislou část, s přesně definovaným rozhraním. Při přenosu systému bylo nutno přizpůsobit technicky závislou část novému prostředí, zbytek systému zůstal beze změn, včetně uživatelských příkazů. Vznikla tak verze označovaná CP/M 1.3, která byla jako první zveřejněna a otevřela CP/M brány do světa uživatelů. V roce 1976 založil G. Kildall firmu Digital Research, která distribuuje CP/M a související programy.

Další vývoj CP/M byl ovlivněn především vývojem diskových pamětí. Vzniká verze CP/M 1.4, která dovoluje ovládat standardní disk firmy IBM (formát IBM 3740 - osmipalcový jednostranný pružný disk s jednoduchou hustotou záznamu). Široké použití tohoto disku dovolilo vysokou přenosnost souborů mezi jednotlivými instalacemi CP/M a tím

nebývalou přenosnost programových produktů.

Vývoj pružných disků se však nezastavil - vznikají disky dalších rozměrů (5 1/4 a 3 1/2 '') s rozmanitými formáty záznamu. Objevily se rovněž pevné disky s velkými kapacitami, použitelné v mikropočítáčových systémech. Operační systém CP/M byl proto upraven tak, aby dovolil ovládat co nejvíce škálu diskových pamětí. Do technicky závislé části byly zabudovány tabulky, popisující vlastnosti použitých disků. Příslušná verze z roku 1979 nese označení CP/M 2.2, a je nejrozšířenější verzí systému CP/M. U nás je známý spíše pod označením Mikros. Rozšířené jsou i východoněmecké verze SCP, SCPX a odvozené.

Vývoj operačního systému pokračoval ještě dále. Byla vytvořena verze CP/M 3.1 (někdy také označovaná CP/M PLUS), která umí ovládat bankovanou paměť, má rozšířenou škálu systémových služeb, možnost nápovědy apod. Existují také varianty CP/M, umožňující spouštění dalších úloh (Concurrent CP/M), multiuzivatelský přístup z více terminálů (MP/M), či sdílení prostředků v rámci sítě (CP/NET). Tyto verze však nedosáhly výraznéjšího rozšíření, neboť nejsou zcela standardní a některé programy na nich nepracují.

Navíc se pozice systému CP/M výrazně posílila, když firma Zilog přišla s mikroprocesorem Z80, pro který byla napsána spousta skvělých programů pracujících pod CP/M (odhadováno o 5000 programů). Tento mikroprocesor díky podstatně vyššímu hodinovému kmitočtu (až 8 MHz) a bohatšímu instrukčnímu souboru umožnil výrazné zrychlení všech činností systému i uživatelských programů. V roce 1979 byl CP/M oceněn cenou Datapro Software Honor Roll.

Firma Intel kupodivu nakonec systém CP/M přes všechny jeho výhody a velké rozšíření nepřijala a vydala se cestou svého vlastního operačního systému ISIS, který se však neujal a je používán pouze ve vývojových systémech Intel.

V čem spočívá neobyčejný úspěch operačního systému CP/M ? Systém vyniká značnou jednoduchostí a nezávislostí na technických prostředcích. Přitom jde o mocný systém s možností snadného rozširování. Jistě by bylo možné navrhnout systémy, které by byly z jistých hledisek rafinovanější než CP/M. Otázkou však zůstává, zda by se podařilo zachovat pružnost při tak malých náročích na paměť. Jedná se o velmi účelný kompromis mezi mocností příkazů a minimalizací nároků na paměť. Rovněž strategie využívání diskového média je jednoduchá a přitom spolehlivá. Celková struktura systému je navržena tak chytře, že byla prevzata ve většině operačních systémů pro mikropočítáče.

2.3 Základní informace o CP/M

CP/M je monoprogramový diskově orientovaný operační systém. Je přísně modulární, přičemž návaznost na technické prostředky je soustředěna v jediném modulu, který je lehko modifikovatelný.

Služby systému umožňují tvorbu další programové nadstavby a to systémových programů i účelového aplikačního programového vybavení.

CP/M je rozdělen na tři moduly:

- vstupně/výstupní modul (BIOS, Basic Input/Output System),
- jádro systému (BDOS, Basic Disk Operating System),
- interpretér příkazů (CCP, Console Command Processor).

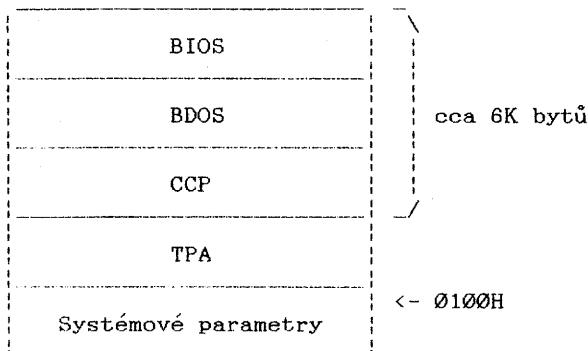
Vstupně/výstupní modul obsahuje fyzické drivery jednotlivých periférií včetně disků. Tento modul představuje základní část operačního systému a umožňuje v konfiguraci technických prostředků používat různé typy disků.

Jádro tvoří na konfiguraci technických prostředků nezávislý operační systém, který zajišťuje vykonávání všech systémových služeb.

Interpretér příkazů, který umožňuje komunikaci operátora se systémem na úrovni konzole, tvoří relativně nezávislý modul CP/M. Je-li to potřebné, může být v paměti překryt uživatelským programem nebo jeho daty.

Po zavedení systému zbyde v paměti RAM volná část, nazývaná pracovní oblast (TPA, Transient Program Area). Slouží k zavádění a běhu služebních a uživatelských programů a k uložení jejich dat. Tato oblast začíná na adresě 0100H, která je též startovací adresou všech programů pracujících pod systémem.

Umístění jednotlivých modulů v paměti RAM je na následujícím schématu:



Ze zobrazeného rozdělení paměti vyplývá, že operační systém je rezidentní na konci paměti RAM. Je to proto, aby TPA začínala vždy na stejně adrese. V případě, že velikost TPA nestací, mají programy možnost překrýt modul CCP. Tímto způsobem je uživatelským programům v rámci dané konfigurace paměti poskytnuta co největší oblast pro jejich práci.

Nejspodnější část paměti (od adresy 0000 do 0FFF) je rezervována pro systémové informace, které jsou nutné pro práci systému. Tyto informace budou popsány dále.

CP/M je schopen ovládat 1 až 16 různých diskových jednotek, přičemž každý soubor na disku může mít kapacitu až 8 MB. Každý z disků má svůj vlastní adresář, ve kterém jsou uloženy informace o souborech na disku. Počet položek adresáře může být předem určen. Položka adresáře obsahuje informace o rozsahu a umístění příslušného souboru na disku. Systém ovládání souborů umožňuje přímý přístup ke kterémukoli z až 65536 záznamů max. 8 MB souboru. Každému z diskových souborů je možno přiřadit příznaky "systémový" a "chráněný proti zápisu". Přes tuto velkou flexibilitu typů disků zabezpečuje systém ovládání souborů logicky jednotný a rychlý přístup k jednotlivým souborům.

Systém ovládání souborů používá pevnou délku logického záznamu - 128 bytů. Diskům o větší kapacitě vyhovuje (z důvodu rychlosti přístupu) větší délka fyzického záznamu. Na to je v CP/M též pamatováno a v BIOSu je možno implementovat algoritmus kumulující více logických 128 bytových záznamů do delšího fyzického záznamu s jediným omezením, že jeho délka musí být násobkem 128 bytů.

Systém ovládání souborů CP/M rozděluje disk na tzv. alokační bloky. Všechny alokační bloky na jednom disku mají

stejnou délku. Jsou tvořeny celistvým počtem sektorů, který je vždy mocninou 2 (např. často používaná délka alokačního bloku je $2^3 = 8$ sektorů = 1 kB, příp. $2^4 = 16$ sektorů). Je-li souboru přidělované místo na disku, děje se tak po diskrétních jednotkách - alokačních blocích. Alokační bloky jsou číslovány 0, 1, 2 atd. Informace o uložení daného souboru na disku je jednoznačně daná číslami alokačních bloků tohoto souboru. Tento způsob má proti tradičním technikám ukládání souboru na disku formou zřetězeného seznamu dvě výhody:

- Nemůže dojít z důvodu přerušení seznamu k ztrátě informace o umístění zbytku souboru,
- je zaručené, že přinejmenším ta část souboru, která vyplňuje alokační blok, je na disku uložena kontinuálně (v těch sektorech, které tvoří alokační blok) a tedy v případě sekvenčního přístupu k tomuto souboru je přístupová doba minimalizovaná. Navíc systém ovládání souborů v případě potřeby přiřazuje souboru další volný alokační blok fyzicky nejbližší k poslednímu alokovanému bloku souboru. Pohyby čtecí hlavy jsou tím omezené na minimum.

Na jednotlivé soubory na disku se CP/M odvolává pomocí jména. Jméno souboru je tvoreno 1 až 8 ASCII znaky; případně je možné za ním udat ještě verzi, která je tvorena 1 až 3 ASCII znaky a je od vlastního jména souboru oddělena tečkou. Příklad: Soubory TURBO a TURBO.COM jsou dva různé soubory, ze kterých první je specifikovaný jen jménem, druhý má jméno stejně jako 1. soubor, ale odlišuje se od něho verzí COM.

Jak už bylo uvedeno, soubory umístěné na disku jsou katalogizované v adresáři. Adresář daného disku je umístěný vždy v počátečních alokačních blocích (0 a výše), v každém případě zabírá alespoň jeden alokační blok, a to nultý. Systém ovládání souborů považuje alokační bloky adresáře za obsazené (alokované).

Položka adresáře je tvořena popisem jednoho souboru na daném disku. To znamená, že kolik položek má adresář, tolik souborů může disk obsahovat. Počet položek se definuje při implementaci systému. Položka adresáře zabírá 32 bytů. To znamená, že do jednoho sektoru adresáře se vejde 4 položky. Položka obsahuje kromě jiného příznak obsazení položky, jméno a verzi souboru, počet sektorů souboru a čísla alokačních bloků, které byly souboru přidělené. V položce je vždy 16 míst pro čísla alokačních bloků. Jestliže seznam alokačních bloků zabírá více než 16 míst, musí být pro něj v adresáři rezervovaná další položka, která popisuje toto další rozšíření (extension) souboru.

Zrušíme-li soubor, projeví se to v adresáři tak, že do příznaku obsazenosti položky daného souboru se zapíše E5H. Místo v adresáři, které položka zabírala je možno využít pro položku popisující jiný soubor. Příznak zrušení položky (a

tedy i souboru) - konstanta E5H - byla zvolena proto, že distributori pružných disků popisují disky právě touto konstantou a systému ovládání souborů se takový disk jeví jako prázdný - všechny položky v adresáři mají příznak "zrušený".

Programátor, který pracuje s diskovým souborem, se na tento soubor odvolává pomocí řídícího bloku souboru (FCB - File Control Block): FCB je vlastní položkou adresáře (má v podstatě stejnou strukturu) a při práci se souborem jsou do něho přepisované některé údaje z položky adresáře (např. čísla alokačních bloků souboru), které jsou po dobu práce se souborem aktualizované a při uzavření souboru jsou zapsané zpět do adresáře.

Struktura FCB je následující:

DN	N1	...	N8	V1	V2	V3	EX	S1	S2	RC	ABØ	...	AB15	CR	RØ	R1	R2
----	----	-----	----	----	----	----	----	----	----	----	-----	-----	------	----	----	----	----

Ø	1	8	9	10	11	12	13	14	15	16	31	32	33	34	35
---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----

FCB zabírá 36 bytů s následujícím významem:

DN - číslo diskové jednotky, na které chceme vykonávat diskovou operaci
1 znamená jednotku A,
2 B,

16 znamená jednotku P (max. 16 jednotek označujeme A až P),

Ø znamená už předtím (implicitně) vybranou jednotku. Vybraná jednotka (nebo častěji disk), je jednotka na které budou probíhat diskové operace.

N1 az N8 - jméno souboru, pricemž v případě, jestlize je kratsí než 8 znaku, je do tohoto počtu doplneno mezerami,

V1 až V3 - verze souboru, v případě potřeby doplněná mezery. Protože verze je tvorena ASCII znaky, může být 7. bit kódu využit pro jiné účely:
Je-li 7. bit V1 nastaven, je daný soubor chráněný proti zápisu (značí se R/O, Read Only), je-li zrušen, soubor není chráněný (R/W, Read/Write),
Je-li 7. bit V2 nastaven, soubor je systémový (značíme SYS) a v seznamu souborů (DIR) se nevypíše, při zrušeném bitu není soubor systémový (značíme DIR).

EX - číslo "části" souboru (extension) popsané jednou položkou adresáře. Části (EX) číslujeme Ø, 1, 2

atd.

- S1, S2 - rezervované pro operační systém,
- RC - počet záznamů (sektorů), které obsahuje daná část (EX) souboru,
- AB0 až AB15 - čísla alokačních bloků daného souboru, které obhospodařuje operační systém,
- CR - číslo záznamu (sektorů), se kterým se bude pracovat (v rámci dané části - extension souboru) při sekvenčních operacích (číst/zapisovat). Chce-li uživatel zpracovávat soubor od začátku, nuluje CR.

- R0 až R2 - tvorí absolutní číslo záznamu (bez ohledu na část EX souboru), které se využívá při přímém přístupu k danému záznamu.

Každý soubor, se kterým programátor pracuje, musí být popsán položkou FCB, přičemž místo pro daný FCB může programátor rezervovat kdekoli ve svém programu. Když přistupuje k souboru, musí vyplnit prvních 16 bytů FCB (obvykle vyplní DN, jméno, verzi a zbytek šestnáctice nuluje) a musí inicializovat CR. V případě přímého přístupu musí namísto CR inicializovat R0 až R2.

V případě prvního přístupu na disk zjišťuje operační systém o tomto disku některé důležité informace, které potom udržuje v paměti. Jednou z takovýchto velmi důležitých informací je např. obsazenost disku. Operační systém musí zjistit, které alokační bloky jsou již přiřazeny a které jsou ještě volné, aby v případě zápisu nového souboru nepřepsal (použitím už přiřazeného alokačního bloku) některý z existujících souborů. Operační systém přečte celý adresář, kde v jednotlivých položkách jsou čísla už přiřazených alokačních bloků a vytvoří alokační vektor disku. Tento vektor je tvořen posloupností bitů. Kolik alokačních bloků má disk, tolik bitů má alokační vektor. Každý bit odpovídá jednomu alokačnímu bloku. Má-li hodnotu 0, je daný alokační blok volný, v opačném případě je již přiřazen. Operační systém potom během své práce aktualizuje tento alokační vektor (při rušení souboru nuluje, při zápisu nastavuje ty bity, které odpovídají příslušným alokačním blokům).

Jak bylo uvedeno, operační systém vytvoří z adresáře alokační vektor disku jen jednou, v případě prvního přístupu na disk. Potom ho už jen aktualizuje. Pro disk, s kterým systém po svém zavedení ještě nepracoval, nemá vytvořený alokační vektor. Disk, pro který už existuje alokační vektor, nazýváme aktivním diskem. Aktivních disků může být několik. Jestliže systém pracoval s prvním diskem, pak s druhým a naposled se třetím, všechny tyto tři disky jsou již aktivní. O tom, které disky jsou aktivní, má systém

informaci v tzv. vektoru aktivních disků.

Představme si, že v době práce systému někdo v jednotce, se kterou se již pracovalo (je aktivní) vymění disk. Je jisté, že alokační vektor tohoto nového disku nebude totožný s alokačním vektorem předcházejícího disku a při zápisu na tento nový disk (kdy systém využívá alokační mapu předcházejícího disku - neví, že disky byly vyměněny) dojde k přepisování jeho už existujících souborů. Aby se to nestalo, vytváří si operační systém při prvním přístupu na disk kontrolní součty jednotlivých sektorů adresáre a ty potom při své práci kontroluje a aktualizuje. Když nenastane shoda kontrolních součtů disku, se kterým pracuje, se součty, které si vytvořil, opravněně předpokládá, že se jedná o jiný disk a označí ho jako R/O - chráněný proti zápisu a tak zabrání přepsání informace na tomto disku.

Při výměně disku se proto doporučuje stlačit CTRL/C, což způsobí kromě jiného, že všechny disky v systému se označí jako neaktivní, t. j. že se pro všechny budou vytvářet alokační vektory znova, a ty už budou odpovídat vyměněným diskům.

Na závěr si všimneme oblasti systémových parametrů. Jak už bylo řečeno, tato oblast je na adresách 0000 až 00FFH.

Na adrese 0000 je skok do BIOSu, který způsobí znovuzavedení modulů CCP a BDOS ze systémového disku do paměti, nastaví adresu DMA na 0080H a všechny disky označí jako neaktivní. (Tuto funkci označujeme jako WBOOT, Warm Boot.).

Na adrese 0005 je skok do BDOSu a pomocí této adresy jsou volány všechny systémové služby.

Skoky na adresách 0000 a 0005 t. j. skoky do BIOSu a do BDOSu jsou při generování systému nasměrovány na skutečné adresy odpovídající uložení operačního systému pro dany rozsah paměti.

Od adresy 005CH vytváří CCP řídící blok souboru (FCB) pro soubor daný jako parametr příkazu interpretovaného CCP.

Oblast od 0080H do 00FFH, t. j. 128 bytů, tvoří buffer pro jeden sektor disku. DMA se automaticky nastavuje na tuto adresu při funkci WBOOT. Všechny diskové operace se vykonávají po sektorech (128 bytů) a operační systém nemá vlastní I/O buffer pro tyto operace. Programátor musí tento buffer nastavovat sám. Oblast od adresy 0080H využívá též CCP a odevzdává v ní programátorovi text příkazového rádku, který zadal z konzole.

2.4 Interpretér příkazů CCP

Uživatel komunikuje s CP/M pomocí interpreteru příkazů, který čte příkazy zadávané z konzole.

Jednotlivé diskové jednotky označuje CCP písmeny A, B, C, až P (max. počet jednotek, se kterými je CP/M schopen pracovat, je 16), za kterými se píše dvojtečka ":". Říkáme, že disk je vybraný, když CCP adresuje právě tento disk. Aby bylo vždy jasné, který disk je vybraný, CCP informuje operátora kódem tohoto disku (A, B, ..P), který je následován symbolem ">", kterým CCP oznamuje, že je schopen přijmout příkaz. Například, v případě, že je vybraný disk B, CCP se na obrazovce zahláší výpisem:

B>

a čeká na příkaz operátora. Když chceme vybrat pro následující operace disk A, zadáme jeho kód:

B>A:

a CCP se ohlásí

A>

Kódem disku spolu s dvojtečkou udaným před jménem souboru udáváme, na kterém disku je daný soubor. Jestliže kód disku neudáme, CCP předpokládá, že soubor je na disku, který je právě vybraný.

Příkazy CP/M jsou implementované ve dvou úrovních:

- příkazy zabudované v CCP,
- příkazy realizované pomocí programů rezidentních na discích.

Každý příkaz je aktivován zápisem jeho jména, případně parametrů a stlačením klávesy CR. Jde-li o příkazy DIR, ERA, REN, SAVE, TYPE, USER a d:, zabudované v CCP, jsou ihned vykonány. Ostatní, tzv. tranzientní příkazy, hledá CCP na disku. V případě nalezení je zavede do TPA a odevzdá jim řízení.

Funkce zabudovaných příkazů je následující:

- DIR vypíše seznam souborů aktivní nebo zadané jednotky
 - např. DIR B:*.COM nebo jenom DIR,
- ERA zrusí specifikované soubory - ERA B:A*.*,
- REN provede přejmenování souboru
 - REN NEW.MAC=OLD.MAC,
- SAVE uloží požadovaný počet stránek oblasti TPA (od zacátku) na disk pod zadaným jménem
 - SAVE 16 HELP.TXT,
- TYPE vypíše obsah souboru (týká se pouze textových sou-

borů), přičemž výpis lze ukončit stiskem libovolné klávesy,
USER nastaví číslo uživatele - USER Ø (až 15).

Při psaní parametrů příkazů je povoleno užívání hvězdičkové konvence, tj. lze napsat otazník místo libovolného znaku a hvězdička nahrazuje skupinu znaků.

Interpreter příkazů využívá jako řídící znaky některé nezobrazitelné znaky ASCII kódu. Tyto znaky jsou generované konzolí displeje při současném stlačení klávesy CTRL a příslušného písmene. Zmíněné znaky se označují jako CTRL/<písmeno>. Např. CTRL/C generuje kód Ø3H apod. Následuje seznam těchto řídících znaků i s jejich funkcemi:

CTRL/C	(Ø3)	znovuzavedení operačního systému,
CTRL/E	(Ø5)	fyzické ukončení řádku,
CTRL/P	(10H)	logické zapnutí/vypnutí tiskárny,
CTRL/S	(13H)	přerušení výstupu na konzoli a vykonávání programu až do vstupu jakéhokoli dalšího znaku,
CTRL/R	(12H)	výpis obsahu vstupujícího řádku,
CTRL/U	(15H)	zrušení vstupujícího řádku,
CTRL/X	(18H)	vymazání vstupujícího řádku z obrazovky i z operační paměti,
CTRL/Z	(1AH)	označení konce souboru (u textových souborů).

Pro programy rezidentní na disku konstruuje CCP ze zbytku příkazového řádku, který následuje bezprostředně za jménem programu, dva FCB. První od adresy ØØ5CH, druhý od adresy ØØ6CH. Je-li parametrem příkazu jméno souboru, FCB tohoto souboru připraví CCP od adresy ØØ5CH. V případě, že je uvedený druhý parametr, potom pro tento je vytvořen FCB na místě ABØ až AB15 prvního FCB a musí být presunut do jiné oblasti paměti, aby nebyl aktivací prvního FCB přepsán.

Dále je pro programy rezidentní na disku oblast od adresy ØØ8ØH inicializovaná ASCII řetězec, který tvoří zbytek příkazového řádku. Na adrese ØØ8ØH je délka tohoto řetězce a od adresy ØØ81H následují jednotlivé ASCII znaky. Tímto způsobem může volaný program získat potřebné parametry.

2.5 Standardní služby BDOSu

Každá služba volaná do systému je realizovaná modulem BDOS, který dle potřeby volá drivery BIOSu. Každá služba má přiřazeno číslo, které se při volání předává v registru C. V případě, kdy služba vyžaduje vstupní parametr, např. adresu FCB, je systému poskytnut v registrech DE. Je-li zapotřebí vstupní parametr jednobytový (např. znak), odevzdává se v registru E, obsah D je pak nepodstatný.

Jestliže služba vrací volajícímu programu dvoubytový parametr, je tento vregistrech HL. Jednobytový výstupní parametr vrací v registru A. Všeobecně pro výstupní parametr platí B=H a A=L, t.j. obsahy registrů B, H a A, L jsou stejné.

Volání služby se realizuje odevzdáním řízení pomocí instrukce CALL 0005H. Na této adrese je instrukce JP BDOS, která zabezpečí správný vstup do BDOSu nezávisle na tom, kde je BDOS v paměti umístěn. Každá služba se tedy volá následující sekvencí instrukcí:

```
LD   C,číslo služby  
LD   DE,parametr  
CALL 5
```

přičemž návrat ze služby je vykonán na adresu následující za instrukcí CALL 5.

Je potřebné si uvědomit, že služby BDOSu nezachovávají obsahy registrů !

služba: Reset systému

číslo služby: 0

vstupní parametr: -

výstupní parametr: -

funkce: Odevzdává řízení do BIOSu, který vykoná znovuzavedení modulů CCP a BDOS z disku do paměti, nastaví adresu DMA na hodnotu 0080H, zachová předtím vybraný disk a číslo uživatele. Potom odevzdá řízení na CCP. Tato služba má stejný efekt jako instrukce JP 0.

služba: Vstup znaku z konzole

číslo služby: 1

vstupní parametr: -

výstupní parametr: ASCII znak

funkce: Přečte znak z konzole (čeká na jeho vstup). Když je kód přečteného znaku <20H (SPACE) a není to znak CR, LF, TAB nebo BS, nevypíše ho na konzoli. Ostatní znaky zobrazí na konzoli.

služba: Výstup znaku na konzoli

číslo služby: 2

vstupní parametr: ASCII znak

výstupní parametr: -

funkce: Zobrazí vystupující znak na konzoli, případně i na tiskarně, je-li zapnuta pomocí CTRL/P. Znak TAB zobrazí jako příslušný počet mezer.

služba: Vstup znaku ze snímače

číslo služby: 3

vstupní parametr: -

výstupní parametr: ASCII znak

funkce: Přečte znak ze snímače děrné pásky (čeká na jeho vstup).

služba: Výstup znaku na děrovač

číslo služby: 4

vstupní parametr: ASCII znak

výstupní parametr: -

funkce: Vydeřuje znak na děrovači děrné pásky.

služba: Výstup znaku na tiskárnu

číslo služby: 5

vstupní parametr: ASCII znak

výstupní parametr: -

funkce: Vytiskne znak na tiskárně.

služba: Přímý vstup/výstup na konzoli

číslo služby: 6

vstupní parametr: ØFFH (pro vstup znaku)

nebo ASCII znak (pro výstup)

výstupní parametr: ASCII znak (byl-li požadován vstup)

nebo nedefinovaná hodnota.

funkce: Je-li vstupní parametr ØFFH , čte status konzole (pta se, zda nebyla stlačena klávesa). Jestliže nebyla žádná klávesa stlačena, vrací jako výstupní parametr ØØ , v opačném případě vrací ASCII reprezentaci vloženého znaku. Z uvedeného vyplývá, že se nečeká na vstup znaku z konzole! Na rozdíl od služby č. 1 nezobrazuje vstupní znak na konzoli. Je-li vstupní parametr různý od ØFFH , považuje tento parametr za ASCII kód vypisovaného znaku a vypíše ho na konzoli.

služba: Zjisti hodnotu I/O bytu

číslo služby: 7

vstupní parametr: -

výstupní parametr: hodnota I/O bytu

funkce: Poskytne obsah adresy ØØØØ . Na této adrese je u systému Intel I/O byte.

služba: Nastav I/O byte

číslo služby: 8

vstupní parametr: hodnota I/O bytu

výstupní parametr: -

funkce: Na adresu ØØØØ uloží vstupní parametr, který představuje I/O byte.

služba: Vypíš ASCII řetězec
číslo služby: 9
vstupní parametr: adresa počátku řetězce
výstupní parametr: -
funkce: Vypíše na konzoli textový řetězec od zadané adresy.
Řetězec musí být ukončen znakem "\$" (24H), před kterým výpis skončí. Výpis jednotlivých znaků vykonává stejně jako služba č. 2.

služba: Čti řetězec znaků z konzole
číslo služby: 10
vstupní parametr: adresa paměťového bufferu
výstupní parametr: -
funkce: Vstupní parametr je adresou počátku bufferu, který má následující strukturu:
MAX - 1 byte
SKUT - 1 byte
ZNAKY - SKUT bytů, kde

MAX je maximální počet znaků (bytů), které je možné uložit do bufferu,

SKUT bude skutečný počet znaků načtených do bufferu. Za tímto bytem jsou ZNAKY, t. j. ASCII reprezentace znaků tak, jak přišly z konzole. Vstup znaků z konzole je ukončen znakem CR nebo LF, které se ale do bufferu již neukládají. Buffer, který je třeba v paměti rezervovat pro načítání řádku z konzole, musí mít délku MAX + 2 byty. Jestliže byl buffer naplněn MAX počtem znaků, je vstup ukončen stejně, jako kdyby vstoupil znak CR.

Při zadávání řetězce znaků je možné vykonávat následující editační funkce:

- DEL vymaže naposled napsaný znak z bufferu a zobrazí tento rušený znak na konzoli (DEL se do bufferu neuloží).
- BS vymaže naposled zapsaný znak z bufferu i z obrazovky (BS se do bufferu neuloží).
- CTRL/E vykoná fyzický přechod kurzoru na nový řádek obrazovky (obsah bufferu se nezmění).
- CTRL/X způsobí vymazání všech vložených znaků z bufferu i z obrazovky.
- CTRL/U způsobí vymazání všech vložených znaků z bufferu a nastaví kurzor na novém řádku pod pozici, odkud začíná načítávaný řádek.
- CTRL/R nastaví kurzor na nový řádek pod pozici, odkud začíná načítávaný řádek a vypíše aktuální obsah bufferu.
- CTRL/C - Jestliže byl tento znak stlačený jako první, způsobí znovuzavedení operačního systému, jinak se uloží do bufferu.
- CTRL/P zapne/vypne tiskárnu, t. j. další vložené znaky se budou zobrazovat i na tiskárně (když tomu dosud tak nebylo), až do následujícího stlačení CTRL/P. Hodnota CTRL/P (10H) se do bufferu neuloží.

Ostatní znaky, pokud jsou zobrazitelné, t. j. jejich kód je $\geq 20H$, se při vstupu zobrazují na konzole (případně i na tiskárně při CTRL/P). Rídící znaky se zobrazují jako dvojznaky a to znak "^" a příslušné písmeno. Například Ø2 se zobrazí jako ^B (B má kód 42H), 1AH se zobrazí jako ^Z (Z má

kód 5AH).

služba: Zjisti status konzole

číslo služby: 11

vstupní parametr: -

výstupní parametr: hodnota odpovídající stavu konzole

funkce: Je-li ve vstupním bufferu klávesnice nějaký znak (t.j. pokud byla stlačena klávesa konzole), vrátí hodnotu 01, v opačném případě vrátí 00.

služba: Zjisti číslo verze systému

číslo služby: 12

vstupní parametr: -

výstupní parametr: číslo verze

funkce: Z důvodu možnosti dalšího rozvoje CP/M a z toho vyplývající možné nekompatibility mezi jednotlivými verzemi (při předpokladu, že vyšší verze může mít další nové služby, které nižší verze nemá), poskytuje tato služba uživatelským programům možnost zjistit verzi operačního systému. Volání vrátí v registru jednobytové číslo predstavující číslo verze (v našem případě 22H, protože jde o systém 2.2).

služba: Reset diskového systému

číslo služby: 13

vstupní parametr: -

výstupní parametr: -

funkce: Volání zabezpečí inicializaci diskového systému, t.j. nastaví DMA adresu na 0080H, vybere disk A, všechny ostatní disky označí jako neaktivní a všechny disky včetně disku A označí jako R/W. Tato služba není ekvivalentní se službou č. 0. Nedojde k znovuzavedení systému.

služba: Vyber disk

číslo služby: 14

vstupní parametr: číslo vybíraného disku

výstupní parametr: -

funkce: Označí zadaný disk jako "vybraný", t.j. všechny následující diskové operace budou probíhat na tomto disku, pokud není explicitně v FCB uvedeno jinak. Vstupní parametr je 0 pro disk A, 1 pro B atd. až 15 pro P. Toto nastavení platí až do následujícího resetu diskového systému.

služba: Otevří soubor

číslo služby: 15

vstupní parametr: adresa FCB

výstupní parametr: kód adresáře

funkce: Před prvním čtením ze souboru, nebo prvním zápisem do existujícího souboru je potřebné získat informace o tomto souboru. K tomuto účelu používáme službu "Otevří soubor". Soubor, který chceme otevřít, udáváme adresou jeho FCB. BDOS prohlédne adresář zadaného disku pro aktuálního uživatele a hledá položku adresáře, která odpovídá bytům 1 až 14 v udaném FCB. (T.j. hledá položku, které odpovídá jménu, verze, EX, S1 a S2, přičemž S2 automaticky nuluje). Jestliže se ve jméně nebo verzi udané v FCB vyskytuje na pozici

některého znaku otazník "?", není tento znak s odpovídající pozicí v položce adresáře porovnáván.

Nastane-li shoda s některou položkou adresáře (adresář je prohledávaný od začátku), je informace z této položky adresáře překopírována do FCB (t.j. čísla alokačních bloků souboru jsou dosazeny na pozici 16 až 31 v FCB).

Služba Otevří soubor tedy zabezpečí obsazení daného FCB informacemi o rozsahu a umístění daného souboru na disku. Tyto informace v FCB jsou během následujících diskových operací s uvedeným souborem aktualizovány. Opětovné uložení aktualizovaného FCB do odpovídající položky adresáře zabezpečí služba "Zavří soubor". Výstupní parametr (kód adresáře) bude mít hodnotu ØFFH, jestliže daný soubor nebyl v adresáři nalezen, v opačném případě má hodnotu Ø až 3, což odpovídá pozici dané položky adresáře v sektoru adresáře (- pro aplikační programování nepodstatné).

služba: Zavří soubor

číslo služby: 16

vstupní parametr: adresa FCB

výstupní parametr: kód adresáře

funkce: Soubor, který byl nějakým způsobem aktualizován, je potřebné uzavřít, t.j. informace o rozsahu souboru a jeho uložení na disku, které jsou během práce s tímto souborem průběžně zaznamenávány v FCB, je potřebné uložit do příslušné položky adresáře. Soubor, z kterého se jen četlo, není nutno zavírat, protože jeho rozsah ani umístění na disku se nezměnily.

Jestliže byla operace uzavření úspěšná, výstupní parametr má hodnotu Ø až 3, v případě neúspěšného zavírání (např. soubor není na disku) vrací služba hodnotu ØFFH.

služba: Hledej první odpovídající položku v adresáři

číslo služby: 17

vstupní parametr: adresa FCB

výstupní parametr: kód adresáře

funkce: Služba hledá od začátku adresáře položku, kterou popisuje FCB. V případě, kdy najde zadanou položku, dá k dispozici od DMA adresy záznam (128 byteů) adresáře, ve kterém je daná položka, a jako výstupní parametr vrátí kód adresáře (Ø až 3). Volající program má potom tuto položku na adrese DMA + 32 * A. Jestliže hledaný soubor není v adresáři, má výstupní parametr hodnotu ØFFH. Ve jménu, nebo verzi souboru, udaného v FCB se může vyskytovat jeden, nebo více otazníků "?". V tomto případě, stejně jako při službě "Otevří soubor", není odpovídající znak v položce adresáře kontrolovaný na shodu. Otazník tedy zastupuje jakýkoli znak. V případě, že se otazník nachází na pozici DN (Ø. byte FCB), je prohledáván adresář vybraného disku a služba vrátí první nalezenou položku, která odpovídá danému FCB, i když je tato položka označena jako zrušená.

služba: Hledej následující položku adresáře
číslo služby: 18

vstupní parametr: adresa FCB

výstupní parametr: kód adresáře

funkce: Služba je ekvivalentní se službou č. 17 s tím rozdílem, že hledání je vykonáváno od předtím nalezené položky. Tuto službu je možné volat pouze bezprostředně po službě č. 17 nebo č. 18.

Výstupní parametr má hodnotu ØFFH , jestliže tato položka v adresáři neexistuje, v opačném případě nabývá hodnoty \emptyset až 3, což je pořadové číslo položky v záznamu (sektoru) adresáře.

služba: Zruš soubor

číslo služby: 19

vstupní parametr: adresa FCB

výstupní parametr: kód adresáře

funkce: Na disku se zruší soubor odpovídající zadanému FCB. Jméno a verze souboru mohou obsahovat jeden nebo více otazníků. V tomto případě jsou zrušeny všechny soubory odpovídající dané specifikaci. Služba vrací hodnotu ØFFH , jestliže zadaný soubor nebyl v adresáři nalezen, v opačném případě vrací hodnotu \emptyset až 3.

služba: Sekvenční čtení

číslo služby: 20

vstupní parametr: adresa FCB

výstupní parametr: kód adresáře

funkce: Ze souboru (udaného pomocí FCB), který byl předtím aktivován službou "Otevři soubor", přečte následující 128 bytový záznam do paměti na adresu DMA. Záznam se přečte od pozice CR dané části souboru a CR se přitom automaticky zvýší tak, že ukazuje na následující záznam souboru. (CR udává počet přečtených záznamů z dané části souboru). Jestliže se CR přeplní, automaticky se otevře další část (EX) souboru a CR se vynuluje.

Služba vrátí hodnotu $\emptyset\emptyset$, když byla operace čtení úspěšná. Nenulová hodnota indikuje, že soubor už neobsahuje žádná data (jsme na konci souboru).

služba: Sekvenční zápis

číslo služby: 21

vstupní parametr: adresa FCB

výstupní parametr: kód adresáře

funkce: Do souboru (popisovaného FCB), který byl předtím aktualizován službou "Otevři" nebo "Založ soubor", se zapíše 128 bytový záznam z adresy DMA. Záznam je uložen na pozici, na kterou ukazuje CR a tento se přitom automaticky zvýší tak, že ukazuje na následující budoucí záznam souboru. Když se CR přeplní, automaticky se otevře další část souboru a CR se nuluje. Při vykonání zápisu do existujícího souboru je příslušný starý záznam (dany CR) přepsán novým záznamem.

Služba vrátí hodnotu $\emptyset\emptyset$, byla-li operace zápisu úspěšná. Nenulová hodnota indikuje plný disk.

služba: Založ soubor
číslo služby: 22

vstupní parametr: adresa FCB

výstupní parametr: kód adresáře

funkce: Služba "Založ soubor" je podobná službě "Otevři soubor", s rozdílem, že FCB musí obsahovat jméno souboru, který se v adresáři nenachází. Operační systém vytvoří na disku prázdný soubor o nulové délce a jeho jméno zapíše do adresáře. Služba vrací hodnotu 0 až 3 při úspěšné operaci, vytvořeného souboru je možno zapisovat pomocí služeb č. 21 a č. 34.

služba: Přejmenuj soubor

číslo služby: 23

vstupní parametr: adresa FCB

výstupní parametr: kód adresáře

funkce: Služba změní jméno souboru, specifikovaného prvními šestnácti byty FCB, na jméno, které je zadáno v druhé šestnáctici FCB. Kód disku na pozici 0 v FCB se využije pro výběr disku. Při návratu ze služby má výstupní parametr hodnotu 0 až 3 při úspěšném přejmenování, nebo 0FFH, když přejmenovávaný soubor na disku není.

služba: Vrať vektor aktivních disků

číslo služby: 24

vstupní parametr: -

výstupní parametr: vektor aktivních disků (2 byty)

funkce: Služba poskytne v registrech HL informaci o discích, které jsou aktívni, t. j. pro které systém ovládání souboru vytvořil příslušné tabulky v operační paměti. Bit 0 reg. L odpovídá jednotce A, bit 7 reg. H odpovídá jednotce P. Má-li bit hodnotu 1, znamená to, že odpovídající jednotka je aktívni.

služba: Vrať číslo vybraného disku

číslo služby: 25

vstupní parametr: -

výstupní parametr: číslo vybraného disku

funkce: Služba vrátí číslo vybraného disku, t. j. číslo 0 pro disk A, atd. až č. 15 pro disk P.

služba: nastav DMA adresu

číslo služby: 26

vstupní parametr: požadovaná DMA adresa

výstupní parametr: -

funkce: DMA adresa (DMA = Direct Memory Access) je adresa v operační paměti, kam se čte (odkud se zapisuje) sektor diskového souboru.

Poznámka: V případě většiny řadičů pružných disků nejde o přímý přístup do paměti, jak název DMA napovídá, ale DMA je třeba chápat jako ukládací adresu v operační paměti.

Adresa DMA zůstává pro všechny následující diskové operace nezměněna až do dalšího volání služby č. 26. Vyjímkou tvoří volání služeb č. 0 a 13, které nastavují DMA adresu na

0080H.

služba: Dodej adresu alokačního vektoru
číslo služby: 27

vstupní parametr: -

výstupní parametr: adresa alokačního vektoru

funkce: Služba poskytne v registrech HL adresu počátku alokačního vektoru vybraného disku. Alokační vektor je tvořen posloupností bitů, přičemž každý bit této posloupnosti odpovídá příslušnému alokačnímu bloku. Je-li alokační blok volný, má příslušný bit hodnotu 0, v opačném případě má hodnotu 1. Délka alokačního vektoru, t. j. počet jeho bitů odpovídá max. počtu alokačních bloků a tedy závisí na kapacitě disku.

služba: Označ disk jako chráněný proti zápisu
číslo služby: 28

vstupní parametr: -

výstupní parametr: -

funkce: Služba označí vybraný disk jako chráněný proti zápisu (R/O). Na tento disk není možné zapsat soubor, mazat nebo přejmenovávat soubor. Tato ochrana trvá až do doby volání služby 0 nebo 13, které všechny disky v systému označí jako R/W, t. j. zruší jejich ochranu.

služba: Dodej hodnotu R/O vektoru
číslo služby: 29

vstupní parametr: -

výstupní parametr: hodnota R/O vektoru

funkce: Služba poskytne v registru HL hodnotu R/O vektoru tvořeného posloupností bitů, z kterých každý odpovídá jednomu disku v systému. Má-li příslušný bit hodnotu 1, je odpovídající disk chráněný proti zápisu. Nejméně významný bit odpovídá disku A, atd.

služba: Nastav atributy souboru
číslo služby: 30

vstupní parametr: adresa FCB

výstupní parametr: kód adresáře

funkce: Služba změní atributy souboru "chráněný proti zápisu" a "systémový" podle hodnoty příslušných bitů verze v FCB, tj. nastavený 7. bit prvního znaku verze označuje R/O, 7. bit druhého znaku verze označuje soubor jako systémový.

služba: Dodej adresu bloku diskových parametrů
číslo služby: 31

vstupní parametr: -

výstupní parametr: adresa bloku diskových parametrů

funkce: Volání této služby poskytne v registru HL adresu v BIOSu, od které začíná blok diskových parametrů vybraného disku. Tento blok udává fyzické charakteristiky příslušného disku.

služba: Poskytni/změň číslo uživatele
číslo služby: 32

vstupní parametr: - \Omega FFH pro poskytnutí čísla uživatele,
- hodnota různá od \Omega FFH pro nastavení
nového čísla uživatele

výstupní parametr: číslo uživatele (bylo-li požadováno)

funkce: Je-li vstupní parametr \Omega FFH , poskytne služba číslo
právě nastaveného uživatele (0 až 15). V případě, že je
vstupní parametr různý od \Omega FFH , předpokládá se, že je to
číslo uživatele, které je třeba nastavit.

služba: Přímé čtení ze souboru
číslo služby: 33

vstupní parametr: adresa FCB

výstupní parametr: kód chyby

funkce: Volání služby poskytne na nastavené DMA adresu 128
bytový záznam ze souboru (předtím aktivovaného službou
"Otevři soubor"), popsaného v FCB, přičemž absolutní
pořadové číslo tohoto záznamu v souboru bylo udáno v FCB+33
a FCB+34. Při volání služby musí být hodnota bytu na adresě
FCB+33 a FCB+34 nulová! Tato služba, na rozdíl od sekvenčního čtení,
automaticky nezvýší udanou hodnotu čísla záznamu.

Číslo požadovaného záznamu může nabývat hodnoty 0 až
65535 a může proto specifikovat část až 8 megabytů dlouhého
souboru.

Služba navíc automaticky nastaví v FCB odpovídající
část souboru (EX) i číslo přečteného záznamu (CR).

Vrácený výstupní parametr reprezentuje následující
chybové kódy:

- 0 - žádná chyba, operace se vykonalá,
- 1 - byla čtena nezapsaná data,
- 2 - tento kód není využit,
- 3 - nemůže uzavřít danou část souboru,
- 4 - chce číst z nezapsané části souboru,
- 5 - plný adresár,
- 6 - pokus o čtení mimo fyzický konec disku.

služba: Přímý zápis do souboru
číslo služby: 34

vstupní parametr: adresa FCB

výstupní parametr: kód chyby

funkce: Služba je podobná službě předešlé s tím rozdílem, že
vykonává přímý zápis do souboru. I chybové kódy, které
služba vraci, jsou stejné.

služba: Vypočti délku souboru
číslo služby: 35

vstupní parametr: adresa FCB

výstupní parametr: vrácená délka souboru na adresách FCB+33,
FCB+34, FCB+35

funkce: Služba vrátí počet záznamů souboru udaného v FCB na
adresy FCB+33, +34, +35. Soubor může mít délku 0 až 65536
záznamů (sektorů) a na uložení tohoto čísla jsou nutné 3
byty. Délka souboru podle nejvyššího pořadového čísla
záznamu v nejvyšší přiřazené části souboru udává tedy

pořadové číslo prvního volného záznamu na konci souboru. Byl-li ale soubor zapsán pomocí služby č. 34, nemusí délka odpovídat počtu uložených záznamů.

služba: Vrať číslo záznamu pro přímý zápis

číslo služby: 36

vstupní parametr: adresa FCB

výstupní parametr: číslo záznamu na adresách FCB+33 a FCB+34
funkce: Při sekvenčním čtení nebo zápisu je udané v FCB číslo části (EX) souboru a pořadové číslo záznamu v této části (CR). Volání služby 36 poskytne na adresách FCB+33 a FCB+34 absolutní pořadové číslo záznamu (od začátku souboru) nehledě na právě otevřenou část souboru.

služba: Reset diskové jednotky

číslo služby: 37

vstupní parametr: vektor diskových jednotek (reg. DE)

výstupní parametr: Ø

funkce: Na rozdíl od služby č. 13, která vykonávala reset celého diskového systému (t.j. všech diskových jednotek), zabezpečuje tato služba reset jen zadaných diskových jednotek. Jako vstupní parametr se odevzdává 16 bitový vektor (délka odpovídá max. možnému počtu jednotek v systému), přičemž nejnižší bit odpovídá jednotce A, nejvyšší jednotce P. Jednotkám, na které chceme aplikovat reset (t.j. nastavit jako neaktivní a nechráněné proti zápisu) nastavíme v odevzdávaném vektoru odpovídající bit.

číslo služby: 38

Tato služba není implementovaná

číslo služby: 39

Tato služba není implementovaná

služba: Popiš alokační blok nulami a potom do něho ulož záznam

číslo služby: 40

vstupní parametr: adresa FCB

výstupní parametr: kód chyby

funkce: Služba má stejný efekt jako služba č. 34 s tím rozdílem, že v případě prvního zápisu do nově alokovaného bloku se tento nejdříve popíše binárními nulami a až pak se do něho uloží záznam. Vykonává vlastně před zápisem "Inicializaci" obsahu alokovaného bloku.

Chybový kód, který služba vrací, je stejný, jako v případě služby č. 34.

2.6 Modul BIOS

Vstupně/výstupní modul (BIOS) je modul, ve kterém jsou soustředěny všechny části operačního systému, závislé na technických prostředcích, na kterých je systém CP/M implementován. Tento modul má jednoznačně definovaný vstupní vektor, přes který jsou volány drivery (obslužné podprogramy) jednotlivých zařízení. Implementovat CP/M na konkrétní typ počítače znamená v konečném důsledku napsat BIOS pro tento počítač.

Protože je BIOS pro ZX Spectrum napsán poněkud jiným způsobem, než jak je to obvyklé, budeme se dále zabývat jen společnými znaky, t.j. vstupním vektorem, přes který může uživatel programovat přímo vstupně/výstupní operace, aniž by volal jádro operačního systému. Popis BIOSu pro Spectrum je součástí tohoto textu a je o něm pojednáno později. Případného zájemce o podrobnější popis "klasického" BIOSu odkazují na [6], popř. [3].

Vstupní vektor se nachází na začátku BIOSu. Má 17 položek, z kterých každá představuje 3 bytovou instrukci skoku na příslušný podprogram BIOSu. Struktura vektoru je následující:

JP BOOT	;vstupní bod do BIOSu při počátečním zavedení operačního systému z disku do paměti
JP WBOOT	;zajistí znovuzavedení modulů CCP a BDOS do paměti
JP CONST	;zjistí stav konzole
JP CONIN	;čti znak z konzole
JP CONOUT	;vypíš znak na konzoli
JP LIST	;vypíš znak na tiskárně
JP PUNCH	;vydeřuj znak do děrné pásky
JP READER	;přečti znak ze snímače děrné pásky
JP HOME	;nastav hlavu na stopu Ø vybraného disku
JP SELDSK	;vyber diskovou jednotku
JP SETTRK	;nastav číslo stopy
JP SETSEC	;nastav číslo sektoru
JP SETDMA	;nastav DMA adresu
JP READ	;čti nastavený sektor
JP WRITE	;zapiš nastavený sektor
JP LISTST	;vrať status tiskárny (její připravenost)
JP SECTRН	;převod logické číslo sektoru na fyzické

Vstupní vektor BIOSu je možno rozdělit na tři logické části: inicializaci systému (BOOT, WBOOT), jednoznačkové I/O operace (CONST, CONIN, CONOUT, LIST, PUNCH, READER, LISTST) a diskové operace (HOME, SELDSK, SETTRK, SETSEC, SETDMA, READ, WRITE, SECTRН).

O všech jednoznačkových operacích se předpokládá, že jsou vykonávány se znaky v kódu ASCII, s nulovým sedmým bitem. Na periferní zařízení CP/M hledí jako na "logická", konkrétní fyzická zařízení definuje BIOS.

Logická periferní zařízení, která CP/M ovládá, jsou:

- konzole: Základní I/O periférie, přes kterou operátor komunikuje se systémem. Nejčastěji je to displej s klávesnicí (terminál), ale obecně to může být i psací stroj, dálnopis apod.
- tiskárna: Zařízení pro výpisy textů (mozaiková tiskárna, rychlá rádková tiskárna apod.).
- děrovač: Zařízení pro děrování děrné pásky.
- snímač: Zařízení pro čtení děrné pásky.

V BIOSu napiše implementátor drivery pro konkrétní zařízení, která mají funkce odpovídající uvedeným logickým zařízením. Je zřejmé, že daný systém nemusí všechny tyto periférie obsahovat. Pro práci CP/M je nezbytná pouze konzole.

Nutno poznamenat, že poslední dvě jmenovaná zařízení (pro práci s děrnou páskou) jsou dnes uváděna jaksi pro úplnost, neboť jejich využití mělo význam spíše v době vzniku CP/M. V dnešní době je výhodnější místo těchto zařízení zabudovat např. přídavná sériová rozhraní.

Diskové operace se provádějí jako posloupnost volání jednotlivých podprogramů, které nastavují číslo diskové jednotky, stopy, sektoru a adresu v paměti pro danou operaci. Příslušné parametry musejí být nastaveny ještě před samotnou operací čtení nebo zápisu a zůstávají nezměněné až do okamžiku, kdy nastavíme jejich novou hodnotu.

V následujícím textu jsou detailně popsány jednotlivé rutiny obsažené v BIOSu.

- BOOT - Vstupní bod do BIOSu, na který odevzdává řízení zavaděč operačního systému (Cold Start Loader) po zavedení celého systému do paměti. Inicializuje některé systémové parametry.
- WBOOT - Na tuto adresu je odevzdáno řízení při volání služby č. 0 (reset systému). Adresa 0 v oblasti systémových parametrů obsahuje instrukci skoku na tuto položku vstupního vektoru BIOSu. WBOOT zabezpečí znovuzavedení modulů CCP a BDOS do paměti a odevzdává řízení na CCP. Skok na adresu 0 (JP 0) je obvyklý způsob ukončení práce uživatelských programů a odevzdání řízení operačnímu systému.
- CONST - Přes tento bod se získá stav vstupu konzole. V registru A poskytne tento podprogram hodnotu 0FFH, pokud je na konzoli připravený znak pro vstup, jinak vrací nulu.
- CONIN - Podprogram přečte ze vstupu konzole znak do registru A a nuluje sedmý (paritní) bit. Podprogram čeká, dokud na konzoli není znak k dispozici.
- CONOUT - Podprogram zabezpečí výpis znaku z registru C na konzoli.
- LIST - Znak z registru C se vypíše na tiskárně.

- PUNCH** - Znak z registru C se vyděruje do děrné pásky.
READER - Ze snímače děrné pásky se přečte znak do registru A. Podprogram na znak čeká.
HOME - Na vybraném disku nastaví hlavu na nultou stopu.
SELDISK - Vybere disk, kterého číslo je v registru C (0 pro disk A, 1 pro B atd.) pro následující diskové operace. Podprogram vrátí v registru HL adresu 16 bytové oblasti (tzv. hlavičky bloku diskových parametrů) patřící vybranému disku. Při pokusu vybrat neimplementovaný disk vraci hodnotu 0000.
 Struktura hlavičky bloku diskových parametrů a vlastního bloku diskových parametrů přesahuje rámec tohoto přehledu a je popsána v [6].
SETTRK - V registru BC se BIOSu předává číslo stopy, na které se má vykonat následující disková operace. Podprogram nastaví tuto stopu.
SETSEC - V registru BC se zadává číslo sektoru, na kterém se má vykonat následující disková operace. Podprogram nastaví tento sektor.
SETDMA - V registru BC se zadává počáteční adresa oblasti paměti (DMA adresa) pro následující čtení nebo zápis na disk. Například jestliže při BC = 1000H nastavíme DMA adresu, budou následující diskové operace čtení i zápisu vykonávané na adresách 1000H až 107FH tak dlouho, dokud tuto adresu nezměníme.
READ - Podprogram čte jeden záznam o délce 128 bytů z vybraného disku, nastavené stopy a sektoru na nastavenou adresu DMA.
 V registru A vrátí 0 v případě úspěšné operace a 1 v případě chyby. Chybový kód 1 je jednotný pro všechny druhy chyb, ať už jde o chybu adresace na disku, ztrátu dat, chybu kontrolního součtu atd.
WRITE - Podprogram zapíše jeden záznam o délce 128 bytů na nastavený sektor na nastavené stopě na vybraném disku z nastavené DMA adresy.
 V registru A je vrácena 0 při úspěšné operaci, nebo 1 v případě chyby.
LISTST - V registru A vrátí hodnotu 0, jestliže tiskárna není schopna převzít vypisovaný znak. V případě její připravenosti se vrací hodnota 0FFH.
SECTRН - Podprogram zabezpečuje převod logického čísla sektoru na fyzické. Vstupními parametry jsou logické číslo sektoru v registru BC a adresa převodní tabulky v registru DE. Logické číslo sektoru je vlastně indexem v této tabulce. Podprogram poskytne v registru HL vysledné fyzické číslo sektoru.
 CP/M nemusí zapisovat jednotlivé sektory na disk v plynule stoupajícím poradí, ale z důvodu minimalizace přístupové doby mohou být sektory zaznamenány na přeskáčku. Tento způsob záznamu potom v konečném důsledku znamená, že čísla sektorů, ve kterých je sekvenčně zapsán soubor, se neshodují s čísly odpovídajících sektorů na disku. Říkáme, že sektory

každé stopy jsou přečíslované. Je-li např. parametr tohoto přečíslení rovný 6, vypadá posloupnost po sobě následujících fyzických sektorů takto: 1, 7, 13, 19, 25, 5, 11, 17 atd. Logickému číslu sektoru 0 pak odpovídá fyzický (skutečný) sektor č. 1, logickému číslu 1 odpovídá sektor 7, logickému číslu 2 odpovídá sektor 13 atd.

Ze všech funkcí realizovaných přes vstupní vektor, kromě BOOT a WBOOT vykonává BIOS návrat pomocí instrukce RET.

Na závěr této kapitoly je znázorněn způsob, jak může uživatel programovat diskové operace přes vstupní vektor nezávisle na umístění BIOSu v paměti. Toto je potřebné v případě, kdy vykonávame konverzi diskových souborů mezi různými operačními systémy apod.

Víme, že na adrese 0 v oblasti systémových parametrů je instrukce skoku na 2. položku, t.j. na instrukci JP WBOOT vstupního vektoru BIOSu. Adresová část této instrukce tedy obsahuje adresu instrukce JP WBOOT. Protože struktura vstupního vektoru je pevně daná, znamená to, že instrukce JP SELDSK, JP SETTRK, JP SETSEC, JP SETDMA, JP READ, JP WRITE je o 24, 27, 30, 33, 36, 39 bytů dále od instrukce JP WBOOT (její adresu můžeme zjistit) a tak můžeme napsat následující podprogramy:

DISK:	LD	HL, (1)	;vyber disk
	LD	DE, 24	
	ADD	HL, DE	
	JP	(HL)	
TRACK:	LD	HL, (1)	;nastav stopu
	LD	DE, 27	
	ADD	HL, DE	
	JP	(HL)	
WRT:	LD	HL, (1)	;zapiš sektor
	LD	DE, 39	
	ADD	HL, DE	
	JP	(HL)	

Funkci vyber disk E a nastav stopu 37H pak můžeme programovat následujícím způsobem:

LD	C, 4	;disk E
CALL	DISK	
LD	A, L	;je zadáný disk v systému ?
OR	H	
JR	Z, ERROR	;jednotka není v systému
LD	BC, 37H	;číslo stopy
CALL	TRACK	;nastav zadanou stopu

3. Implementace CP/M na Spectru - popis BIOSu

Modulu BIOS jsou věnovány celé dvě následující obsáhlé kapitoly. Podrobný a úplný popis programové obsluhy Microdrive by ovšem byl zbytečně obsáhlý a není ani nutný pro účely pochopení funkce BIOSu jako celku. Naopak část o změnách systému a instalaci tiskárny je napsána velice podrobně, neboť se předpokládá, že tyto zásahy budou dělat i osoby programování méně znalé.

Vzhledem k netradičnímu způsobu implementace systému CP/M (použití Microdrive místo pružného disku) je BIOS pomérne složitý program. Svedcí o tom i jeho délka (kolem 6 kilobytu), která ve srovnání s klasickým BIOSem je asi pětkrát větší. Rozsah paměti, kterou zabírá celý systém, se tím zdvojnásobuje. Proto je systém implementován tak, jako by pracoval na počítači s 60 kB RAM. CCP začíná na adresě D400H, BDOS na DC00H a BIOS na adresě EA00H, přičemž zabírá místo až do konce paměti RAM (FFFFH).

Hlavním programem BIOSu je emulátor pružných disků na ZX Microdrive. Kdyby se Microdrive obsluhovaly stejně jako disk, celý systém by pracoval neuvěřitelně pomalu (vyzkoušeno). Bylo nutno se uchýlit k takovým prostředkům, jako je sdílení času, priorita diskových operací, nepřetržitá činnost klávesnice i při běhu Microdrive a přístup k souborům přes velké vyrovnávací paměti. Tyto věci známe spíše od počítačů podstatně vyšších tříd. Díky tomu popisovaný BIOS pracuje s Microdrive několikrát rychleji, než podobné systémy s disketovými mechanikami. BIOS umí ovládat nejvíce tři mechaniky Microdrive a RAM disk o kapacitě 128 nebo 384 kB.

3.1 Zavedení systému

Studený start systému je vyřešen asi jediným možným způsobem. Do mechaniky 1 se vloží kartridž se systémem a zadá se příkaz RUN. Zavaděč zjistí velikost paměti RAM, otestuje ji, zavede systém a spustí jej.

Z mechaniky se zavádějí celkem čtyři moduly. Je to modul se jménem "run", což je basicový program zajišťující nastavení několika systémových proměnných a zavedení dalších modulů. Následuje blok "init", což je důkladný RAM test. Kromě vlastního otestování paměti ve dvou průchodech provádí i test správného občerstvování paměti (Refresh test) a RAM (uloží ji na adresu 23681). V případě nějaké závady v paměti se program zastavuje (DI + HALT). Jako další je načten z mechaniky modul s názvem "disk", který zajišťuje inicializaci RAM disku a způsobí, že ihned po zavedení systému jsou již v RAM disku připraveny některé základní programy. O tom, jak tyto programy do modulu "disk" uložit, je pojednáno v kapitole 6. Nakonec se zavede modul "cp/m",

představující vlastní operační systém. Z celé této zaváděcí sekvence není možno žádný modul vyněchat, neboť jednotlivé moduly si mezi sebou předávají parametry.

V případě, že používáme pouze 80 kB RAM, zobrazí se po spuštění systému kurzor a čeká se na vložení kartridže s formátem CP/M do první mechaniky. Po této operaci se stiskne libovolná klávesa a systém se rozbehne. Jestliže systém uvedený kartridž nenalezne, činnost se opakuje až do správného načtení adresáře (přítomnost CP/M kartridže je nezbytná).

CP/M je v případě nutnosti možno zavést i z kazety, a to obvyklým příkazem LOAD "".

Teplý start ovšem funguje poněkud jinak. Na jiných počítačích se systém znova zavádí z disku. Zde ale systém na kartridži vůbec není, aby tu nezabíral drahocenné místo. Proto se při prvním natažení systém přemístí do nevyužité části paměti VRAM, odkud se při teplém startu instrukcí LDDR přesune zpět nahoru na místo funkce.

3.2 Obsluha klávesnice

Také činnost klávesnice byla pozměněna. Především nestačil počet kláves. Nová obsluha klávesnice tudíž zpracovává tři současně stisknuté klávesy. Je tak možno vkládat všechny ASCII znaky s kódy od 00H až po 7FH. Některé klávesy mají odlišný význam a jsou uspořádány v následující tabulce:

Klávesa	Funkce	Klávesa	Funkce
EXTEND MODE	CTRL	EDIT	ESCAPE
GRAPH	TAB	BREAK	CTRL/S
TRUE VIDEO	CTRL/C	INV VIDEO	CTRL/R
CURSOR LEFT	BS	CURSOR RIGHT	CTRL/D
CURSOR UP	CTRL/E	CURSOR DOWN	CTRL/X

Symbol shift + SPACE + B je přerušení programu se skokem do systému. Je to výhodná funkce např. v případě, že program uvízne v nekonečném cyklu. Ostatní klávesy fungují tak jako na jiných počítačích, tzn. CTRL + písmeno dává odpovídající řídící znak.

Klávesnice je čtena neustále nezávisle na ostatních činnostech počítače. V případě, že běžící program nestačí znaky z klávesnice zpracovávat, klávesnice pracuje do vyrovnávací paměti o velikosti několika příkazových rádek. Je proto možné např. při překladu souboru asemblerem již zadat příkaz pro spuštění překládaného programu, který v době zadání tohoto příkazu ještě vlastně neexistuje.

Podle potřeby je možno využít dalších funkcí klávesnice, a to zrušení činnosti rutiny CONST (když chceme psát další příkazy v době, kdy program testuje klávesnici) pomocí kláves Symbol Shift + SPACE. Obnovení činnosti provádí ESCAPE, CTRL/C nebo CTRL/S.

Jinou funkcí je opakování již zadaného příkazu. Text několika naposled napsaných rádek je totiž i po odeslání klávesou ENTER uložen v bufferu a jednotlivé příkazy je možno vyvolat klávesami Symbol Shift + ENTER. Chceme-li např. opakovat poslední zapsaný příkaz, stiskneme Symbol Shift + ENTER a ENTER. O více příkazů se dá vracet vícenásobným stiskem Symbol Shift + ENTER. Pro vymazání nahromaděného textu z bufferu na obrazovce lze s výhodou použít CTRL/X. Lze využít i samočinného odeslání dříve zadaného příkazu, což lze docílit odesláním klávesou LF (CTRL/ENTER) místo obvyklého ENTER. Zpětné vyvolání Symbol Shift + ENTER pak způsobí překopirování naposled napsaného příkazu z bufferu do příkazového rádku a jeho okamžité odeslání. Klávesnicový buffer je cyklicky uzavřen do kruhu, takže v případě "přejetí" hledaného příkazu se lze k tomuto příkazu zpět vrátit.

3.3 Obsluha obrazovky

Jak je známo, CP/M vyžaduje obrazovku s 80 znaky na řádek. Jak se však ukázalo, s určitými omezeními pracuje i na systémech s 64 znaky. U Spectra je to vyřešeno mikroprintem s formátem 64 znaků na 24 řádkách. Součástí výstupního programu jsou i podprogramy pro rychlé rolování obrazovky nahoru a dolů a pro mazání řádky, což výrazně zvyšuje komfort při editování textu. ASCII znaky o kódech 20H až 7EH, BEL, BS, LF a CR jsou zobrazovány obvyklým způsobem, seznam ostatních řídících kódů je v následující tabulce:

Znak	Funkce	Znak	Funkce
ESC = y x	CURSOR LEAD	ESC T	ERASE TO EOLN
ESC R	DELETE LINE	ESC E	INSERT LINE
ESC (LOWLIGHT	ESC)	HIGHLIGHT
CTRL Z	CLS+HOME		

(souřadnice y a x u sekvence CURSOR LEAD mají offset 20H)

BIOS neumí zpracovat znaky TAB, FF a DEL. Řídící znaky byly zvoleny ve snaze po co největší standardizaci BIOSu a jsou shodné s terminálem Televideo 925, případně i 912 nebo 950.

Součástí výstupní rutiny je i podprogram pro tisk kurzoru, který běží přes přerušení a kurzor je tudíž stále viditelný (některé programy vyžadují viditelný kurzor i v době, kdy se nečeká na klávesnici). Kurzor bliká i v době běhu mechanik Microdrive.

3.4 Tiskárna a děrná páska

To jsou zařízení, která má každý uživatel jiná. Proto v BIOSu nejsou obslužné programy pro tisk a pásku zahrnuty. Protože případné uživatelské podprogramy pro obsluhu těchto periférií nelze volat z místa vstupního vektoru BIOSu (jiná stránka RAM a registr SP na nevhodném místě), jsou za vstupním vektorem BIOSu ještě čtyři vektory další, které jsou v pořadí:

Adresa	Podprogram
BIOS+33H	Výstup znaku na tiskárnu
BIOS+36H	Status tiskárny
BIOS+39H	Vyděrování znaku na pásku
BIOS+3CH	Přečtení znaku z pásky

Vstupní parametr je v registru C, výstupní v A. Uživatel na příslušné místo BIOSu (přidané vektory) umístí instrukci skoku na svůj podprogram, který napiše, přeloží od adresy 6200H a zařadí do systému na adresu CCP - 800H, odkud jej systém přemístí na místo určení a zavolá. Pro uživatelské rutiny je zde volných 300H bytů. Při volání je nastránkováný mód 48K, povolené přerušení v módu IM 2 a SP je nastaven na adresu, pod níž je volno nejméně 20H bytů. Uživatel smí používat jen základní registry AF, BC, DE a HL. Nedoporučuje se také volat rutiny do ROM ani do přídavné ROM, kterou má např. ZX Interface 1 nebo jiné interfejsy.

4. Diskové operace

Diskové operace na úrovni Microdrive jsou zdaleka nejsložitějšími operacemi BIOSu. Jejich základem je sdílení času procesoru mezi činností Microdrive a běžícím programem, které se uskutečňuje přes přerušení a probíhá ve dvou různých prioritách. Vyšší prioritu mají okamžité požadavky běžících programů, nižší prioritu má přístup BIOSu na disk při vykonávání dodatečných činností, vyvolaných dřívějšími požadavky na sekvenční čtení a zápis. Komunikace programu se soubory se uskutečňuje přes přístupové vyrovnávací paměti, které mají velikost až 62 kB.

V souvislosti s Microdrive si nemohu odpustit jednu poznámku. Často je slyšet názory, jak je Microdrive nespolehlivý, že se z kartridžů ztrácejí data, že soubory nejdou přečíst, že se kartridže zadírají, pásek se zamotává atd. Nechci Microdrive nikterak vychvalovat, ale sám dvě mechaniky vlastním již více než dva roky, používám je téměř denně a chyb vzniklé za tuto dobu bych spočítal na prstech svých rukou. Při vytváření CP/M se uvažovalo o tom, že by každý zápis na kartridž byl automaticky verifikován, zkoušky však ukázaly, že to není nutné, protože chyb prostě nevznikají. Jestliže ale přijdu k nějakému jinému vlastníku Microdrive, stávám se sám svědkem znacných potíží.

Jak si vysvětlit tento rozpor? Zdá se, že má asi tyto příčiny:

- Špatné zacházení s mechanikami a kartridži (zádna ochrana před prachem, nečistotami, magnetickým polem, vysší teplotou, vibracemi a otresy, nedůsledné uzavírání kartridžů do krabiček ihned po vyjmnutí z mechanik atd.).
- Neudržování mechanik (špatné nastavení pracovních poloh prvků páskové dráhy, síly přitlačných per, kolmosti hlavy a rychlosti posuvu pásku).
- Některá várka kartridžů, prodávaných v PZO Tuzex, byla zřejmě vadná, protože vykazovaly tolik chyb, že uživatelé byli nutni někdy až třetinu kartridžů vyrádit. K tomu je ovšem nutno poznamenat, že i některé kartridže zakoupene v zahraničí nemají požadované parametry. Obvykle však celé série fungují bezchybně.

Při potížích s Microdrive proto radím to, co jsem provedl sám: špatné kartridže prostě vyřadit (i ty, co již někdy selhaly a od té doby fungují).

Důležitá informace se týká ošetření chyb, které případně mohou vzniknout při diskových operacích. CP/M je sam o sobě velmi málo zabezpečen proti chybám. Při detekování chyby BIOSem je sice volajícímu programu vrácen příznak chyby, avšak následná reakce systému se nejčastěji redukuje pouze na výpis hlášení

BDOS Err On d: Bad Sector

a čekání na klávesu. Je-li stisknuta klávesa CTRL/C, provede se teplý start systému. Při stisku kterékoli jiné klávesy program pokračuje ve své činnosti tak, jako kdyby k chybě vůbec nedošlo. POZOR ! Při troše nepozornosti se tak lehce může systém zhroutit (při zavedení chybného programu) nebo dojde ke vzniku dalších chyb. Účinná ochrana před následky vzniklých chyb je pouze stisk CTRL/C při vypsání chybového hlášení.

V této souvislosti je nutno se zmínit ještě o jedné věci. Jde o případ, kdy nějaký soubor na disku, do kterého píšeme, nebude zapsán celý (pozastavíme-li činnost programu, vyskočíme-li z programu pomocí Symbol Shift + SPACE + B nebo když se nám počítac zakousne). Tehdy nedojde k správnému uzavření souboru, který pak má menší délku a jeho poslední část na disku chybí. Jindy je v adresáři uvedeno pouze jméno souboru, který ale na disku není. Takové zbytky souborů je vhodné včas z disku vymazat (příkaz ERA). Neuzavřený soubor lze v některých případech identifikovat podle přípony .\$\$\$, kterou používají některé programy. Toto tvrzení však neplatí všeobecně.

4.1 Formát záznamu na mechanikách Microdrive

CP/M na Microdrive využívá jiný formát záznamu než originální ROM, a to ze dvou důvodů. Jednak je zvýšena jmenovitá kapacita kartridže na 96 kB, jednak původní záznam není možný, protože u něj není počítáno s existencí adresáře. Kartridž je implementován jako disk s jednou stopou, na kterém je 192 sektorů o velikosti 0.5 kB. Každý sektor má krátkou hlavičku, ve které je nulový flag, číslo kartridže (2 byty), číslo sektoru a kontrolní součet. Za hlavičkou následuje 512 bytů dat, která jsou uvedena flagem o hodnotě FFH a končí kontrolním součtem.

Pro větší názornost jsou oba formáty znázorněny následujícími tabulkami:

a) Sinclair ROM formát

Pásek v kartridži

Sektor 5 Sektor 4 Sektor 3 Sektor 2 Sektor 1
--

Sektor

Header Gap1 Descriptor Record Gap2
--

Header

Header preamble Header flag Data Checksum

Descriptor

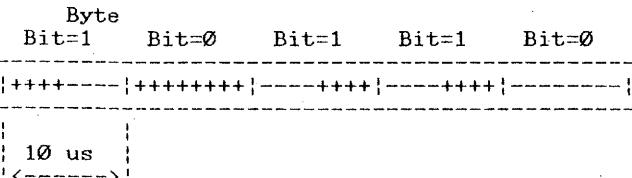
Record preamble Record flag Data Checksum

Record

Data Checksum

Data

Byte 1 Byte 3 Byte 5 Byte 7 Byte 2 Byte 4 Byte 6 Byte 8
--



-----| ++++++ | ++++++ | -----+ + + + | -----| - kladná magnetizace | -----| - záporná magnetizace -----

Záznam jednotlivých bitů je totožný se záznamem Single Density na pružných discích.

Popis	Počet bytů
Header preamble	12
Header flag	1
Header data	13
Header checksum	1
Gap1	-
Record preamble	12
Record flag	1
Descriptor data	13
Descriptor checksum	1
Record data	512
Record checksum	1
Gap2	-

b) CP/M formát

Pásek v kartridži

-----| Sektor 0 | Sektor 96 | Sektor 1 | Sektor 97 | Sektor 2 | -----

Sektor

-----| Header | Gap1 | Record | Gap2 | -----

Header

-----| Header preamble | Header flag | Data | Checksum | -----

Record

-----| Record preamble | Record flag | Data | Checksum | -----

Popis	Počet bytů
Header preamble	12
Header flag	1
Header data	3
Header checksum	1
Gap1	-
Record preamble	12
Record flag	1
Record data	512
Record checksum	1
Gap2	-

Význam jednotlivých položek záznamu je následující:
 Gap - mezera na pásku mezi místy, kde jsou zaznamenané byty (pásek zde není využit pro záznam). Slouží k vyhledávání začátku zapsaných dat.

Preamble - úsek uvádějící zapsaná data. Je potřebný pro správné zasynchronizování obvodů řadiče při čtení. Aby byla data správne nactena, je třeba na každé stopě zaznamenat alespoň 5 nulových bytů a jeden byte o hodnotě FFH.

Flag - byte udávající druh přečtených dat (zda se jedná o header či record).

Data - zaznamenané byty nesoucí vlastní záznam. Data se zapisují současně na obě stopy střídavě po jednom byte.

Checksum - kontrolní součet zapsaných dat. Používá se k detekci chyb vzniklých při zápisu nebo čtení. Jeho hodnota se určí součtem s kruhovým přenosem, který je proveden se všemi kontrolovanými byty.

Header - značka na pásku sloužící k identifikaci následujícího sektoru.

Descriptor - datový blok předcházející vlastnímu záznamu. Specifikuje data zapsaná v sektoru (příznak obsazenosti sektoru, příslušnost k určitému souboru a definice umístění tohoto sektoru v daném souboru).

Z uvedeného vyplývá, proč má kartridž v systému CP/M větší kapacitu. Je zkrácena hlavička sektoru, z které bylo vypuštěno jméno kartridže. Dále byl zcela vypuštěn nepotřebný descriptor, jehož funkce byla úsporně nahrazena existencí adresáře. Oproti původnímu záznamu byla také zvýšena spolehlivost spočtením časových prodlev při záznamu a čtení, které jsou v důležitých místech určeny s přesností jednoho taktu hodin procesoru. Proto není vhodné do rutin obsluhujících Microdrive sebeméně zasahovat.

Existují také programy, které zvyšují kapacitu kartridže zkrácením nevyužitých mezér mezi daty (Gaps). Tato cesta byla také zkoušena, avšak zkrácení Gapu má za následek snížení spolehlivosti záznamu. Jako vhodný kompromis byla zvolena délka Gapu (mezi headerem a recordem) 10920 taktů hodin procesoru, délka druhého Gapu není přesně stanovena vzhledem k jiné délce sektorů použitých při formátování (bude vysvětleno v popisu programu FORMAT). Tento Gap může být dlouhý zhruba 7 ms.

Číslo kartridže obsažené v každé hlavičce sektoru je náhodné číslo, které je generováno při formátování a slouží BIOSu k rozpoznání výměny kartridže bez počítání kontrolních součtů adresáře BDOSem.

Z důvodu maximální optimalizace přístupových doby při sekvenčních operacích není pevně stanoven parametr přecíslení sektorů (nutnost přecíslení byla vysvětlena v kapitole 2.6 - služba SECTRН). BIOS přecíslení neprovádí vůbec, sekvence po sobě následujících čísel sektorů je generována při formátování kartridže a parametr přecíslení je volitelný. Uživatel tak může přístupové doby optimalizovat na minimum, když přihlédne k druhu prováděné práce a konkrétním časovým konstantám svých mechanik, které se bohužel u jednotlivých kusů liší. Jako univerzálně nejvýhodnější se ukázalo přecíslení po dvou, které při použití BIOSu 4.0 vyhovuje všem mechanikám. U BIOSu 3.3 je nejlepší přecíslení 2 pro programy a 12 až 24 pro data.

Kartridž je médium, na kterém se po formátování téměř vždy vytvoří nějaké vadné sektory, způsobené přinejmenším slepkou páskové smyčky. Proto zápis do těchto vadných sektorů je znemožněn tak, že po několikerém testu kartridže je do adresáře zapsán systémový, proti zápisu chráněný soubor, který obsahuje všechny vadné sektory a je zaznamenán pod číslem neexistujícího uživatele 31. Systém proto na tento soubor nijak nereaguje.

Na kartridž je možno v ideálním případě zaznamenat až 95 kB dat, adresář má velikost 1 kB (2 sektory). Velikost alokačního bloku je rovněž 1 kB a tudíž maximální počet souborů zaznamenaných na jednom kartridži je dvaatřicet. Kartridž nemá žádné systémové stopy.

4.2 BIOS 4.0

V průběhu vývoje systému došlo nutně k rozdvojení způsobu činnosti BIOSu v závislosti na použité konfiguraci RAM. Je pochopitelné, že důmyslné strategie využívání přístupových bufferů pro soubory není možno použít, není-li pro ně paměť. Ještě před dokončením verze 3.3, určené pro 80 kB RAM, vznikla úprava pro 272 a 528 kB, což autorovi umožnilo rozpracovat zcela novou strategii bufferování a dalo impuls k vytvoření mnohem rychleji pracující verze 4.0. Nasledující odstavce popisují tuto verzi.

BIOS 4.0 má zabudovanou obsluhu RAM disku, jehož kapacita je buď 128 kB nebo 384 kB podle dané konfigurace paměti RAM. RAM disk má adresář o kapacitě 2 kB, délka alokačního bloku je taktéž 2 kB. Disk nemá systémové stopy, obsahuje jen jednu stopu pro data. Maximální počet souborů, které se ještě na disk vejdu, je 64.

Obsazení rozšířené paměti RAM v systému 4.0 znázorňuje následující tabulka:

Číslo stránky	Využití
0	Dolní část TPA a systémové parametry.
1	Dolní část bufferů pro Microdrive a tabulky popisující stav těchto bufferů.
2	Horní část bufferů pro Microdrive.
3 až předposlední	RAM disk.
Poslední (7 resp. 15)	Horní část TPA a vlastní operační systém.
VRAM	Kopie CCP a BDOS a uživatelské ovladače.

4.2.1 Výběr mechaniky

Používáme-li více mechanik, je do BIOSu volaná služba výběru mechaniky, se kterou se bude dále pracovat. Tato služba spočívá především ve zjištění, zda je požadovaná mechanika připojena a je v ní kartridž. BIOS v průběhu své činnosti udržuje v paměti vektor připojených mechanik, tj. vede si informace o tom, s kterými mechanikami se od spuštění systému úspěšně pracovalo (pozor, není totožné s vektorem aktivních disků BDOSu, neboť tento vektor se vytváří znova po teplém zavedení systému). Jestliže se s mechanikou ještě nemanipulovalo, číslo kartridže uložené v paměti je nula a program zkouší z mechaniky přečíst některou hlavičku sektoru. Přečte-li, zjištěné číslo kartridže uloží do paměti a vrací se, jinak hlásí chybu.

Zároveň se testuje, zda nedošlo k výměně kartridže. Byl-li kartridž vyměněn za jiný, tento nový kartridž má jiné náhodné číslo. BIOS v tomto případě je nucen zrušit data v

přístupových bufferech příslušných dané mechanice a vytvořit buffery nové. Celá operace zjištění náhodného čísla kartridže z hlavičky sektoru včetně zapnutí motoru netrvá déle než 150 ms. Test ekvivalence kartridže se neprovádí v případě, že poslední operace provedená s danou mechanikou byla ukončena během posledních dvou sekund. Za tuto dobu totiž není možné kartridž vyměnit (prakticky vyzkoušeno).

Při výměně kartridžů může dojít k závažné situaci, kdy např. vyjmeme kartridž z mechaniky C, vložíme jej do mechaniky B, kde na něj zapíšeme nějaká data a pak kartridž vrátíme zpět do mechaniky C. Při následujících operacích s mechanikou C by mohlo dojít k načtení chybných dat, neboť data uložená v bufferu nesouhlasí s daty na kartridži a protože číslo kartridže souhlasí s číslem v paměti, BIOS nedetectuje výměnu kartridže. Tato skutečnost byla zjištěna náhodou, a zajímavé je, že ani profesionální operační systém QDOS, který je přímo určen pro práci s Microdrive, tento stav programově neosetruje. V BIOSu pro Spectrum je této situaci zabráněno tím, že při zjištění výměny kartridže v libovolné mechanice jsou testována čísla kartridžů v ostatních mechanikách a je-li nalezeno shodné číslo, jsou zrušeny buffery této mechaniky.

Microdrive čísluje BIOS zprava doleva postupně B, C a D, disk A je RAM disk. Pokud není RAM disk přítomen vůbec, mechaniky se značí A, B a C.

4.2.2 Operace čtení

Sekvenční čtení sektoru je nejčastěji prováděna operace. Musí proto pracovat co nejrychleji. Provádí se tak, že při požadavku na čtení se nejdříve testuje, zda daný sektor se už nevyskytuje v přístupovém bufferu (ve většině případu tomu tak je). Když ano, sektor se předá volajícímu programu a pochopitelně s mechanikou Microdrive není nijak manipulováno. Pouze při operaci otvírání a zavírání souboru (přístup do adresáře) se hlídá, zda mezikádřidž někdo nevyměnil. Tato kontrola spočívá v prečtení první hlavičky sektoru a porovnání náhodného čísla kartridže s číslem uloženým v paměti. Jestliže se číslo neshoduje, jsou vyprázdněny příslušné přístupové buffery dané mechaniky a čtení probíhá přímo z kartridže. Celý test výměny netrvá déle než asi 150 ms. Aby však nedocházelo k zbytečnému zdržování, tento test se také neprovádí vždy. Jeho činnost je podobná jako pri testu výměny kartridže při výběru mechaniky. Kontrola je vypuštěna v případě, že poslední operace čtení nebo zápisu na dané mechanice proběhla během posledních dvou sekund.

Jiná situace nastává, když daný sektor není v bufferu nalezen. Pak se uvolní ta část bufferu, která je v dané chvíli relativně nejméně potřebná a do takto vzniklého místa se načte sektor z kartridže, aby se mohl posléze předat

volajícímu programu.

U systému s rozšířenou pamětí je navíc každá operace čtení následovaná logickými procedurami, jejichž cílem je co nejvíce minimalizovat doby spojené se čtením souboru. I když není požadovaná žádná další operace čtení a běží hlavní program, BIOS si přes přerušení zaplňuje přístupový buffer dalšími sektory z otevřeného souboru, dokud se buffer zcela nezaplní nebo pokud počet sektorů nepřesáhne hodnotu 10. Tím je zajištěno, že čtení souboru z mechaniky se o 5 kilobytů "předchází" pred vlastními požadavky na čtení a odezva BIOSu na požadavek čtení je velmi rychlá (jedná se vlastně jen o přemístění dat v paměti).

4.2.3 Operace zápisu

Zápis je podobně přizpůsoben velkým časovým nárokům, jako čtení. V naprosté většině případů je uskutečněn tak, že je uvolněna nejméně potřebná část bufferu a do této části je zapisován sektor uložen. V příslušných tabulkách se zapisí informace o tom, že daný sektor má být zapsán a řízení se předá hlavnímu programu, jako když zápis již byl hotov. Vlastní zapsání sektoru provede BIOS později, nejčastěji přes přerušení nebo často je také operace zápisu provedena jako vedlejší účinek operace čtení (když se při čtení nalezne sektor, který má být zapsán). Není nútne se obávat, že by například při výpadku sítě nedošlo k záznamu posledních dat, neboť systém všechny požadované zápisy provede nejdříve ihned a nejpozději do padesáti sekund od vlastního požadavku.

4.2.4 Práce s přístupovými buffery

Jak již bylo vysvětleno, čtení a zápis sektoru neprobíhá přímo, ale přes přístupové buffery, které mají obecně různé délky, které si systém mění podle potřeby. Po zavedení systému a několika minutách práce jsou pochopitelně všechny buffery zcela zaplněny a systém musí začít šetrně hospodařit s místem v bufferu. Tato strategie byla nekolikrát měněna a zkoušena, než byl nalezen její definitivní tvar.

Základem přidělování místa v bufferech čteným a zapisovaným souborům je vyhodnocení, který buffer je v současné chvíli relativně nejméně potřebný a je ho tudíž možno uvolnit pro sektor z jiného souboru. Buffery jsou rozděleny na 124 půlkilobytových úseků, kde každý úsek přísluší jednomu sektoru na kartridži. Výběr nejméně důležitého sektoru se provádí na základě tabulky priorit bufferů. Tato tabulka je dlouhá 124 bytů (124 uložených sektorů) a začíná na adresě **0600H** v první stránce RAM. Každý byte této tabulky obsahuje číslo od 0 do 6 včetně, a to podle následujícího kódu:

Hodnota	Význam
Ø	prázdný, nevyužitý sektor
1	sektor, z něhož se pouze četlo a byl přečten CELÝ (tzn. že byly přečteny všechny 4 jeho 128 bytové záznamy)
2	sektor, z něhož se právě čte
3	sektor, který byl načten s předstihem (ještě před požadavkem na jeho přečtení) a předpokládá se, že jeho čtení bude v blízké době požadováno
4	sektor, do kterého se zapisovalo a není ještě uložen na kartridži
5	sektor patřící adresáři, ze kterého se pouze čte
6	adresářový sektor, do něhož se psalo

Jak je vidět, nejvyšší prioritu má adresář, a to z toho důvodu, že ze všech sektorů na disku je do adresáře přistupováno nejčastěji. Hned potom následují sektory, do kterých se psalo, a protože ještě nejsou fyzicky zaznamenány, je jejich zápis na kartridž "životně" důležitý (tento buffer se nesmí vyprázdnit dříve, než po vykonání fyzického zápisu).

Velikou výhodou tohoto systému práce s Microdrive je to, že požadované operace čtení a zápisu nejsou prováděny v tom pořadí, jak byly do BIOSu volány, ale pořadí prováděných činností si určuje BIOS sám. Například jestliže požadujeme od BIOSu zápis více sektorů, tato data se uloží do bufferu a řízení se vrátí do volajícího programu. Přerušovací rutina, když zjistí podle tabulek popisujících stav bufferů, že byl vykonán nějaký zápis, spustí motor dané mechaniky a hledá nějaký sektor, do kterého má být proveden zápis. Jestliže takový sektor najde, vyhledá jeho umístění v bufferu a zapíše jej. Poněkud komplikovanější situace nastává, když do sektoru sice bylo psáno, ale nebyly provedeny zápisu do všech jeho čtvrtin (128 bytových záznamů, které používá CP/M). V tom případě je nutné při prvním nalezení sektoru doplnit v bufferu ty části, které nebyly zapsány a při druhém nalezení se teprve sektor zapíše. Protože pořadí zápisových operací není určeno pořadím, v jakém byly zápisu volány, ale rozmištěním sektorů na kartridži, celý zápis libovolného počtu sektorů z bufferu nemůže nikdy trvat déle než na dvě otočení páskovou smyčkou.

BIOS se musí také starat o operace prováděné na ostatních mechanikách. Bylo by neefektivní při každém čteném nebo zapisovaném sektoru střídavě na jedné a druhé mechanice s těmito mechanikami střídavě manipulovat. Proto jsou nejdříve provedeny všechny požadované operace s jednou mechanikou a teprve po jejich skončení se provádějí operace s mechanikou druhou. To je dobré patrné např. při kopírování více souborů z jedné mechaniky na druhou, kdy BIOS neprovádí kopírování po jednom souboru, ale běh mechanik se střídá vždy po několika zkopiovaných souborech.

Informace o datech uložených v bufferech jsou zapsány ve dvou dalších tabulkách. První z nich začíná na adrese 0000H v první stránce RAM a má organizaci 3 krát 192 bytů (3 mechaniky a na každé z nich 192 sektorů). V každém bytu této tabulky je hodnota přiřazující určitému sektoru (jehož číslo je indexem v této tabulce) určitý buffer. Pokud je hodnota bytu FFH, pak danému sektoru není přiřazen žádný buffer (sektor není v paměti). Hodnota FEH říká, že daný sektor je vadný (není z něj možno číst). Toto je velmi výhodné opatření, neboť v případě vícenásobného pokusu o přečtení vadného sektoru by mechanika dluho běžela. Tomu je zamezeno tím, že při první zjištěné chybě v sektoru se do odpovídajícího bytu tabulky přiřazení bufferů dosadí FEH a při každém dalším přístupu do tohoto sektoru je automaticky hlášena chyba, anž by se BIOS pokoušel sektor znova přečíst. Hodnota v tabulce přiřazení bufferů různá od FFH a FEH udává pořadové číslo bufferu, ve kterém je daný sektor. Buffery začínají na adrese 0800H a zabírají zbytek první stránky RAM a celou stránku druhou (systém je ve stránce nulté a poslední).

Druhá tabulka, organizovaná stejně jako předchozí, začíná od adresy 0300H a nese informaci o obsazení jednotlivých čtvrtin (logických záznamů) každého bufferu (fyzického sektoru). Bity 0 až 3 každé položky této tabulky udávají správnost příslušné čtvrtiny bufferu (bit 0 přísluší první čtvrtině, bit 3 poslední, čtvrté čtvrtině). Bit 4 udává zápis do sektoru. Při požadovaném zápisu do sektoru je tento bit nastaven a po provedení zápisu je zrušen. Bity 5 až 7 slouží jako čítač jednotlivých přístupů do sektoru. Jestliže dosáhne hodnoty 4, znamená to, že sektor byl celý přečten a tudíž je možno snížit příslušnou hodnotu v tabulce priorit. Pochopitelně podobné snížení hodnoty v tabulce priorit vyvolá i úspěšné dokončení zápisu do sektoru, který byl dříve požadován.

Protože BIOS musí obsluhovat mechaniky Microdrive na nejnižší možné úrovni, stará se i o zapínání a vypínání pohonného motoru. Tato operace však trvá zhruba 18 ms a proto je vyloučeno s motorem manipulovat po každé operaci čtení nebo zápisu, neboť by to byla velká časová ztráta. Motory se proto zapínají okamžitě, je-li to nutné, avšak jejich vypínání je zpožděné asi o 0.8 s, tak jak to dělají hardwarové diskové rádiče. Proto Microdrive tak často nepublikují a neprovádějí záznam a čtení s opětovným zapínáním a vypínáním mechanik jak to dělá ROM, ale většinou vypadá celé čtení nebo zápis tak, že se mechanika rozběhne a po chvíli zastaví. Kromě jiného to také šetří pásek v kartridžích.

Rutiny obsluhující transport dat mezi Microdrive a počítačem jsou napsány s ohledem na co největší efektivnost. Původní ROM např. při čtení sektoru nejdříve pomocí instrukce INIR načte celý sektor do paměti RAM, v které pak

vypočítává kontrolní součet a poté se teprve rozhoduje, co se sektorem má vlastně dělat. Autor ROM se patrně domníval, že se během vlastního čtení časově nestihá již nic dalšího dělat.

To je ovšem veliký omyl. Rutiny v ROM pracují tak, že v době mezi posláním dvou po sobě následujících bytů je procesor ve stavu WAIT. Jeden průběh instrukce INIR trvá 20 taktů hodin procesoru, zatímco na jeden přečtený byte připadá asi 140 taktů. Je sice třeba ponechat jistou dobu jako rezervu pro případ různé rychlosti posuvu pásku a zpracování informací rádičem Microdrive, stejně nám ale asi 80 taktů hodin na jeden byte zůstává k dispozici. Kromě toho obvod ULA v Interface 1 má vyrovnávací registr pro čtení a zápis do Microdrive, takže v případě opoždění převzetí jednoho byte se stále ještě nic vážného neděje. Nově vytvořené rutiny proto komunikují s Microdrive pomocí instrukcí OUT (C),r a IN r,(C), mezi kterými využívají volného času k výpočtu kontrolních součtů, kontroly čísla kartridže a správnosti flagu. Čtecí rutiny se také v průběhu své činnosti rozhodují, zda určitou 128 bytovou část půlkilobytového záznamu mají číst do paměti RAM (tj. do příslušného bufferu), nebo zda se data mají načíst jen "do vzduchu" a testovat jen kontrolní součet. Tato metoda umožňuje provádět skládání fyzických záznamů ze zapsaných a nezapsaných částí (čtvrtin) přímo v průběhu čtení sektoru a v daném bufferu, není nutné mít na to v paměti zvláštní pracovní prostor a zdlouhavě pak data přesouvat sem tam mezi přístupovými buffery a tímto pracovním prostorem.

Další využitelný čas (nejméně dvě milisekundy !) je mezi hlavičkou sektoru a záznamem, ve kterém BIOS vyhledává příslušný buffer a vyhodnocuje operace, které se mají se sektorem provádět. Negativní stránkou využití této doby je, že příslušné programy musejí kromě vlastní činnosti také určovat velikost časového intervalu, který potřebovaly pro vykonání prováděných operací (jedna rutina BIOSu vrací jako výstupní parametr dobu, jak dlouho běžela).

Poslední využitelnou dobou je doba mezi dvěma sousedními sektory, která je použita pro čtení klávesnice, zobrazování kurzoru a případně v této době ještě několik milisekund běží hlavní program.

Také přesuny bloků dat v paměti jsou o čtvrtinu rychlejší než obvykle. Přesun bloku paměti se běžně řeší instrukcí LDIR. Tato instrukce trvá 21 taktů hodin procesoru na každý přenesený byte. Málokdo však využívá instrukce LDI, prestože ta trvá pouhých 16 taktů. BIOS pro blokové přesuny používá sekvenci dvaatřiceti za sebou bezprostředně následujících instrukcí LDI, přičemž se s výhodou využívá toho, že délka přenášeného bloku je vždy násobkem třiceti dvou. Stejný způsob je aplikován i na rychlé rolování obrazovky dolů i nahoru. Uvedený přenos je vůbec nejrychlejší metodou přenosu dat v paměti, které lze s

mikroprocesorem Z80 dosáhnout.

4.3 BIOS 3.3 - odlišnosti od BIOSu 4.0

Jak již bylo uvedeno, je napsán i BIOS 3.3 pro 80 kB Spectrum. Tento systém nepoužívá velké přístupové buffery jako BIOS 4.0, má jen malou pracovní oblast pro čtyři sektory o velikosti 2 kB. Samozřejmě výkonnost tohoto systému je podstatně nižší než u jeho většího kolegy. Praktické použití lze doporučit jen v případech, kdy nebude požadován častější přístup na disk, což se týká například jazyků MBasic, TURBO Pascal apod. Překladače jazyků Ada, MT Plus, nebo databázové programy pracují velice pomalu.

Rozdělení paměti v systému 3.3:

Stránka	Využití
Ø	Dolní část TPA a systémové parametry.
1 až 6 (existují-li)	RAM disk.
RAM 8000H až FFFFH (resp. stránka 7)	Horní část TPA a vlastní operační systém.
VRAM	Kopie CCP a BDOSu a uživatelské ovladače.

Zavedení systému, čtení klávesnice, tisk na obrazovku, způsob záznamu na Microdrive, výběr mechaniky a další operace jsou v podstatě shodné s BIOSem 4.0, odlišnosti budou popsány v následujícím textu.

Klávesnice nemá možnost opakování příkazu a blokování výkonu rutiny CONST, takže bufferování klávesnice se omezuje jen na programy, které neustále klávesnici netestují.

Systém má implementovaný RAM disk o kapacitě 192 kB pro případ, že by snad někdo vlastnil tuto starsí verzi CP/M a dodatečně rozšířil paměť na 272 kB. Použití této varianty však nedoporučuji, protože je daleko rozumnější oželet těch 64 kB RAM disku a mít mnohonásobně rychlejší Microdrive. RAM disk má adresář o velikosti 2 kB, délka alokačního bloku je 1 kB.

BIOS 3.3 využívá buffery pro čtyři sektory, přičemž jeden z nich je většinou obsazen jedním sektorem adresáře. Účinná funkce BIOSu je tedy možna jen při práci s nejvýs třemi soubory současně. Strategie bufferování je asi následující:

Je-li požadovaná operace čtení nebo zápisu a není-li daný sektor v bufferech nalezen, začne BIOS hledat, který sektor v bufferu relativně nejméně potřebuje. Celý další postup závisí na tom, zda sektor, pro nějž uvolňujeme místo, nalezi adresáři či nikoli. Má-li sektor číslo Ø nebo patří adresáři a pro získání místa bude vypuštěn sektor:

1. Neadresárový sektor, který byl ale už volajícím programem prečten nebo zapsán CELÝ (to znamená, že byly prečteny nebo zapsány všechny čtyři čtvrtiny

- (logické záznamy) tohoto sektoru, neboť fyzický sektor má délku 512 bytů a logický sektor jen 128).
2. Neadresářový sektor, z něhož se pouze četlo.
 3. Jakýkoli sektor, z něhož se pouze četlo.
 4. Nebyla-li žádná z předchozích činností úspěšná, nastane operace honosně nazvaná "aktivní časový výběr". O co se vlastně jedná. V této situaci, která v danou chvíli známená zastavení práce hlavního programu, je třeba nějakou část bufferu co nejrychleji uklidit. Protože všechny sektory v bufferech byly hlavním programem zapsány, ale na kartridži ještě nejsou, je nutno nějaký sektor rychle zaznamenat. Proto se spustí motor jedné mechaniky Microdrive (pokud ovšem nějaká již neběží) a hledá se nějaký sektor, který má být zapsán. Je-li nalezen, zápis se provede a takto uvolněné místo je využito pro jiný sektor. Při výběru mechaniky, na kterou se prebytčný sektor odklidí, má samozřejmě přednost mechanika již běžící.

Výběr se provádí podle mírně jiného algoritmu, pokud chceme získat prostor pro sektor z nějakého souboru. Pak se nejméně potřebný sektor určí takto:

1. Předcházející sektor, z něhož se pouze četlo a je celý přečtený volajícím programem. Je to nejčastěji sektor, který v souboru novému sektoru předcházel a tudíž již s největší pravděpodobností nebude čten.
2. Sektor, z něhož se pouze četlo a je přečtený celý.
3. Jakýkoli sektor, z něhož se pouze četlo.
4. Aktivní časový výběr.

Volba nejméně potřebného sektoru se navíc vždy řídí pravidlem, že v bufferu musí být stále přítomny sektory patřící adresáři, neboť s těmi se pracuje velmi často.

Obsluha pohonného motoru mechanik je obdobná jako u BIOSu 4.0. Motory se zapínají ihned při nároku na kartridž, jejich vypnutí je zpožděné asi o 1.5 s.

5. Základní programové příslušenství

K systému CP/M pro Spectrum byly napsány tři programy, tvorící jeho nezbytný doplněk. Je to program FORMAT pro kvalitní formátování kartridžů a programy MLOAD a MSAVE pro zálohování programů na magnetofonových kazetách.

Bohužel popis dalších uživatelských programů, které pracují pod systémem, jako jsou překladače různých programovacích jazyků, editory a databáze, není do tohoto textu zahrnut. Znamenalo by to totiž napsat knihu o několika stech stránkách. Zájemci se proto musí obrátit na jinou dostupnou literaturu popisující uživatelské programy CP/M. Poměrně rozsáhlý sortiment takové literatury např. nabízí JZD Agrokombinát Slušovice jako příslušenství svých počítačů TNS, nebo podnik Kancelářské stroje k počítačům Robotron.

5.1 Program FORMAT

Tento program slouží k formátování kartridžů použitelných pak v CP/M. Spuštění programu se provede prostřednictvím CCP napsáním jeho názvu (FORMAT) a odesláním. Program se zeptá:

Drive number to format or ENTER to reboot:

na což lze odpovědět buď zadáním čísla mechaniky 1, 2, nebo 3, nebo stlačením klávesy ENTER se vrátit zpět do CCP. V případě číselné odpovědi se program dotáže:

Sectors renumber parameter (2, 3, 4, 6, 8, 12, 16, 24, 32, 48):

Číselná hodnota zadaná po tomto dotazu udává, s jakým přecíslením se má kartridž naformátovat. Parametr přecíslení určuje počet sektorů, následujících těsně za sebou na kartridži, které připadají na jeden sektor přečtený (zapsaný). Například zadáme-li hodnotu 6, bude systém čist při zavádění programu vždy každý šestý sektor, tzn. jeden sektor přečte a následujících pět přeskočí.

Osvědčilo se používat přecíslení po dvou, které je ve většině případů nejrychlejší. Pouze v případě 80 kB verze systému je nutno kartridže, které nejsou určeny pro záznam programů, nýbrž pro sekvenční soubory (zdrojové texty, manuály apod.), formátovat s přecíslením 12, aby se mechanika stáčila mezi dvěma sousedními čtenými sektory rozjet a zastavit. Při nejasnostech je nejlepší v konkrétním případě optimální parametr přecíslení najít zkusmo.

Po zadání parametru přecíslení se vypíše:

Insert disk in drive X and press ENTER

poté se stiskem klávesy ENTER odstartuje vlastní

formátování. Celá činnost má několik fází:

- Test, neběží-li nějaká jiná mechanika (když např. systém dokončuje předešlé požadované operace).
- Test, není-li kartridž chráněný proti zápisu (Write protect).
- Mazání starého záznamu na kartridži (trvá asi 8 sekund). Cílem je odstranit starý záznam i v oblastech mezi sektory (Gap) a kartridž usadit v mechanice.
- Formátování. Na kartridž jsou zaznamenány hlavičky sektorů a za nimi následují formátovací záznamy, jimž je zbytek sektorů celý zaplněn. Tyto záznamy mají větší délku, než běžné sektory, aby překrývaly i oblast Gap. Formátovací záznamy obsahují asi 600 byteů o hodnotě FCH (hodnota FCH byla zvolena s ohledem na to, že se na ní dobře projeví chyby). Činnost trvá asi 8 sekund.
- Předběžný test. Tento test se pokouší přečíst formátovací záznamy a podle souhlasu či nesouhlasu přečtených dat si vytváří mapu dobrých a špatných sektorů. Výsledek testu je průběžně zobrazován na obrazovce. Test má dva průchody, z nichž každý trvá 7 s.
- Test na zápis a čtení. Spočívá v zápisu všech sektorů záznamem o délce 512 byteů o hodnotě E5H a následném čtení. Výsledek čtení každého sektoru je zaznamenán v mapě. Tento test nemá pevně stanovený počet průchodů. Provádí se tak dlouho, než je výsledek všech průchodů jednoznačný a shodný. Činnost končí, když pět po sobě jdoucích testů dává stejný výsledek. Doba trvání jednoho průchodu je 14 s.
- Další činností je vytvoření a zápis adresáře. Program z výsledku formátování podle vytvořené mapy sektorů vyrobí adresář, který obsahuje jeden soubor s názvem "-----". Tento soubor má příznaky "systémový" a "chráněný proti zápisu" a je zaznamenán pod číslem fiktivního uživatele 31. Tento uživatelským nemůže existovat a tudíž soubor ----- není vlastně vůbec přístupný. Obsahem souboru jsou všechny sektory, které se v průběhu formátování projevily jako špatné a tímto je do nich znemožněn zápis. Uložení adresáře na kartridž trvá 7 s.

V případě, že je chybný sektor, který obsahuje adresář, je celá formátovací procedura opakována.

Rutiny, které provádějí čtení a kontrolu zapsaných formátovacích záznamů, nejsou shodné s rutinami BIOSu. Liší se přísnějšími tolerancemi při vyhodnocování mezí sektory (Gap) a synchronizačních signálů. Na jakosti naformátování kartridžů závisí spolehlivost funkce celého zařízení. Formátovací program v paměti ROM takové jakostní naformátování vůbec neumožňuje, neboť provádí pouze jediný test.

Po úspěšném formátování se vypíše:

Operation completed, XX K available.

s následným

Drive number to format or ENTER to reboot:

a celá činnost se opakuje.

Při špatném zadání některého parametru se program na tento dotazuje znova. Nedáří-li se kartridž naformátovat vůbec (vadný kartridž nebo špatně fungující mechanika), vypíše program zprávu

Sorry, permanent error occurred.

Činnost programu lze v kterémkoli okamžiku přerušit stiskem kláves Symbol Shift + SPACE + B. Program pak vypíše hlášení

Operation aborted.

a spustí se od začátku.

Poslední vyvinutá verze programu FORMAT nese označení FORMAT 4.0.

5.2 Programy MLOAD a MSAVE

Tyto programy jsou určeny především pro zálohování (backup) programů a dat zaznamenaných na kartridžích. Nikdo si nemůže dovolit takový přepych mít tolik kartridžů, aby měl na nich všechny své programy. Je také možné použít programy MLOAD a MSAVE k přenosu souborů ze systému CP/M do obyčejného Spectra či naopak. Není problém rovněž přenášet programy mezi Spectrem a jinými počítači, neboť kompatibilní program existuje např. na SHARP MZ-800, IBM PC a další.

Základem je struktura záznamu velmi příbuzná se záznamem obyčejného Spectra. Záznam vytvořený programem MSAVE a posléze čitelný pomocí programu MLOAD má následující strukturu:

Magnetofonový pásek

| Blok Ø | Blok 1 | Blok 2 | ... | Blok X (poslední) |

Blok

| Header | Record |

Header

| Leader | Header flag | Data | Header Checksum |

Record

| Leader | Record flag | Data | Record Checksum |

Leader

| ++++++ | ++++++ | ++++++ | ... | ++++++ | ----- |

| 1.24 ms | | 191 us | 210 us |
<-----> | | <----> | <----> |

5 s pro header, 2 s pro record

Data
Bit=Ø Bit=Ø Bit=1 Bit=Ø

| ++++++ | ++++++ | ++++++ | ----- | ++++++ |

| 489 us | | 977 us |
<-----> | | <-----> |

-----| +++++ - kladná magnetizace | ----| - záporná magnetizace -----

Formát dat v hlavičce každého bloku je podobný hlavičce, kterou vytváří Spectrum ROM při záznamu CODE. Rozdíly jsou dány nutností použít v CP/M jedenáct znaků pro jméno (8 znaků pro název a 3 pro verzi) a potřebou počítat jednotlivé bloky, neboť programy delší než operační paměť se musí na kazetu ukládat po několika blocích. Tyto bloky mají délku 32 kB a každý z nich má samostatnou hlavičku:

Byte číslo	Význam	Hodnota
Ø	Typ CODE	Ø3H
1 až 10	Název souboru bez osmého znaku a verze znak (znaky mají zrušený 7. bit)	
11 a 12	Délka bloku v bytech	0000-8000H
13 a 14	Startovací adresa	Ø100H
15	Poslední znak názvu souboru	znak
16	Pořadové číslo bloku počínaje nulou (poslední blok má nastaven 7. bit)	ØØ-FFH

Kontrolní součet bloku (checksum) se určí XORováním všech bytů bloku včetně úvodního (Flag).

Programy mají jednoduchou obsluhu. Program MLOAD se zavolá z CCP příkazem

MLOAD <specifikace souboru>

přičemž specifikace souboru může být předznamenaná kódem požadované diskové jednotky a je v ní povoleno používat znaků hvězdička "*" a otazník "?". Specifikace souboru musí být uvedena. Program hledá na pásmu soubor vyhovující dané specifikaci a když ho najde (presněji když najde nultý blok tohoto souboru), soubor překopíruje na příslušnou diskovou jednotku. Pokud zde soubor daného jména již existuje, je novým souborem přepsán. Byl-li požadován přenos jen jednoho souboru, program se po vykonání akce vrátí do CCP. V opačném případě budou přeneseny všechny nalezené programy vyhovující specifikaci. Návrat z programu je zprostředkován klávesou SPACE nebo BREAK, stisknutou v okamžiku hledání dalšího souboru na pásmu.

Hlášení programu MLOAD:

- | | |
|-------------|---|
| Searching | - program hledá na pásmu soubor daného jména. |
| Ignoring | - byl nalezen program jiného jména nebo nalezený blok nemá číslo nula (jsme uprostřed souboru). |
| Loading | - soubor se nahrává z pásku do oblasti TPA. |
| Writing | - soubor se kopíruje z oblasti TPA na disk. |
| Transferred | - soubor byl úspěšně překopirován. |

Operation completed - program uspesne vykonal pozadovanou funkci a vrací se.

- Operation aborted** - činnost programu byla předčasně zastavena (klávesa SPACE nebo BREAK).
Tape error - byla detekována chyba při čtení z kazety (ztráta dat nebo chybý kontrolní součet).
Disk write error - chyba při zápisu na disk (není místo na disku nebo v adresáři).
Too long - délka bloku je příliš velká, takže není možno jej nahrát do paměti.

Používání programu MSAVE je analogické programu MLOAD.
Po spuštění programu příkazem

MSAVE <specifikace souboru>

program zaznamená na kazetu všechny soubory z vybraného nebo požadovaného disku, které vyhovují uvedené specifikaci. Před začátkem záznamu prvního souboru program napiše

Start tape, then press any key.

a čeká na stisk klávesy, aby obsluha měla čas na spuštění magnetofonu. Mezi jednotlivými soubory jsou asi desetisekundové mezery s akustickým signálem, aby bylo možno soubory na páse později snadno vyhledávat. Po zaznamenání všech požadovaných souborů se program vrací zpět do CCP. V případě potřeby lze činnost programu předčasně ukončit klávesou SPACE nebo BREAK.

Programy MLOAD a MSAVE lze použít i v režimu rychlého záznamu. Tento režim je výhodný pro uživatele používající výhradně magnetofon a slouží ke krátkodobému zaznamenání pracovních souborů, u kterých nezáleží na spolehlivosti a trvanlivosti záznamu. Příkaz pro záznam na kazetu má tvar

MSAVE <specifikace souboru> <rychllost>

kde parametr "rychlost" může nabývat hodnot 1, 2 nebo 3 a určuje, kolikrát je pořízený záznam rychlejší než klasický ROMkový. Zrychlený záznam není samozřejmě čitelný pomocí ROM. Při zpětném nahrávání příkazem MLOAD se rychlosť neudává, program si ji nastavuje sám.

Hlášení programu MSAVE:

- Reading** - soubor se kopíruje z disku do oblasti TPA.
Saving - soubor se přehrává z TPA na kazetu.
Transferred - soubor byl úspěšně zaznamenán.
Start tape, then press any key - program vyzývá obsluhu, aby spustila magnetofon.
Operation completed - program úspěšně provedl záznam všech souborů a vrací se do systému.
File(s) not found - disk neobsahuje žádny soubor, který by vyhovoval zadáné specifikaci.
Operation aborted - činnost programu byla předčasně ukončena klávesou SPACE nebo BREAK.
Disk read error - chyba při čtení souboru z disku.

Too long - soubor nelze překopírovat z disku do paměti
RAM (oblast TPA je menší než 32 kB).

Nejnovější verze programu MLOAD a MSAVE nesou označení
4.ØF.

6. Než začnete se systémem pracovat

Předtím, než začnete systém běžně používat, je možno v něm provést drobné změny, týkající se nastavení některých systémových proměnných (konstanty čtení klávesnice, pipání klávesnice), dále výběru programů, kterými se při zavádění systému inicializuje RAM disk, a v neposlední řadě je to instalace tiskárny a jiných uživatelských periférií.

Především je nutno systém zkopirovat z distribuční kazety na kartridž, z kterého jej budete zavádět. Nejjednodušší je to lze provést pomocí programu copier (případně jeho české verze copier2), který dodává firma Sinclair jako standardní příslušenství Microdrive. Postup je následující:

1. Najít na distribuční kazetě soubor s názvem "run" příslušející dané verzi (na jedné straně kazety je verze pro 272 a 528 kB, na opačné straně verze pro 80 kB RAM).
2. Nahrát program copier - LOAD *"m",drive,"copier"
3. Napsat následující program v jazyku Basic:

```
10 * MOVE "c",,"run" TO "m",drive  
20 * MOVE "c",,"init" TO "m",drive  
30 * MOVE "c",,"disk" TO "m",drive  
40 * MOVE "c",,"cp/m" TO "m",drive
```
4. Zadat povel RUN a spustit magnetofon.

V případě, že nemáte program copier, lze překopírování systému provést následovně:

1. Najít na distribuční kazetě soubor "run" náležející požadované verzi (272 a 528 kB nebo 80 kB RAM).
2. Vyresetovat počítač (RANDOMIZE USR Ø).
3. Zadat následující sekvenci povelů:

```
- MERGE "run"  
- SAVE *"m",drive,"run" LINE 1  
- CLEAR 25087  
- LOAD "init"CODE :SAVE *"m",drive,"init"CODE 25088,5120  
- LOAD "disk"CODE :SAVE *"m",drive,"disk"CODE 32768,32768  
- LOAD "cp/m"CODE :SAVE *"m",drive,"cp/m"CODE 52224,13312
```

Tímto je systém CP/M překopírován na kartridž, odkud můžete zkoušet jej spustit (RANDOMIZE USR Ø a RUN).

Upozornění: není možno zaměnit jednotlivé moduly ze systému pro 272/528 kB a pro 80 kB RAM. Přestože se stejně jmenují, jejich obsah je jiný.

Pokud si chcete změnit některé drobnosti v systémových proměnných, máte možnost to provést v zaváděcím basicovém programu (modul "run") pomocí příkazů POKE. CP/M využívá tyto standardní systémové proměnné:

Název	Adresa	Význam
REPDEL	23561	Casový interval od stisknutí klávesy do

REPPER	23562	spuštění opakování funkce (autorepeat)
PIP	23609	Rychlosť opakování stisknuté klávesy
BORDCR	23624	Délka pípnutí klávesnice (při hodnotě 00 klávesnice nepípe vůbec)
FLAGS2	23658	Barva okraje obrazovky (shodné s ATTR_P)
bez jména	23681	Mód klávesnice (norm/caps)
ATTR_P	23693	Informace o rozsahu přítomné paměti (uloženou hodnotu neměnit !)
IOBORD	23750	Barva papíru a inkoustu pro tisk Tato proměnná slouží k indikaci zápisu na Microdrive (je použita pouze tehdy, je-li Microdrive připojen). Zápis je indikován blikajícím atributem na adresu 23295 (dolní pravý roh obrazovky). Jestliže je obsahem proměnné IOBORD hodnota standardního papíru, indikace je vypnuta. Jinak atribut bliká uloženou barvou.
D_STR1	23766	Číslo mechaniky, ze které se zavádí systém (jen je-li Microdrive zapojen).

Následující odstavec se týká automatické inicializace RAM disku po zavedení systému. Distribuční kazeta obsahuje modul "disk", který zajišťuje tuto inicializaci. Jak se lze snadno přesvědčit spuštěním systému, standardní modul "disk" obsahuje programy POWER, FORMAT, MLOAD a MSAVE. Pokud by chtěl někdo toto počáteční obsazení RAM disku pozměnit, má možnost to provést s jediným omezením, že souhrnná délka všech programů na disku nesmí překročit 28 kB (včetně prázdného místa v neúplně využitých alokačních blocích). Postup změny obsahu modulu "disk" je tento:

1. Zavést systém do počítače.
2. Smazat celý RAM disk (příkaz ERA *.* s odpovědí "Y" na dotaz, zda opravdu chcete vše smazat).
3. Z některé mechaniky nebo z kazety nahrát do RAM disku programy, kterými chceme disk při zavádění inicializovat. Provedeme to programem POWER příkazem COPY (kopírujeme-li z mechaniky) nebo programem MLOAD (nahráváme-li z kazety), přičemž program POWER resp. MLOAD nahrajeme z mechaniky.
4. Přesvědčit se, zda velikost obsazené části RAM disku není větší než 28 kB. Lze snadno provést programem POWER, zadáním příkazu SIZE ***. Příkaz vypíše tabulku souborů, přičemž poslední vyspaná hodnota (vpravo dole) je hledané obsazení.
5. V případě, že rozsah části RAM disku obsazené programy je větší než 28 kB, je nutno celou proceduru opakovat od začátku (pozor, obsazení RAM disku může přesáhnout uvedených 28 kB i v případě, když součet délek jednotlivých programů je menší).
6. Vyresetovat počítač.
7. Zadat příkaz CLEAR 32767: LOAD *"*m*",drive,"disk"CODE
8. Zadat RANDOMIZE USR 65002: ERASE "*m*",drive,"disk"
9. Zadat SAVE *"*m*",drive,"disk"CODE 32768,32768

10. Systém vyzkoušet. Tzn. Spectrum vyresetovat, zadat RUN, zavést systém a presvědčit se o správnosti programů v RAM disku (programem POWER pomocí příkazu CHECK ***). Zobrazené hodnoty kontrolních součtů programů musí souhlasit s kontrolními součty programů z mechaniky. To je důležité ! Při resetování počítače totiž nejsou občerstvovány dynamické paměti a tudíž obsah RAM disku může být resetem poškozen. V případě nesouhlasu kontrolního součtu některého programu provedeme celou proceduru znovu od začátku.

6.1 Využití programu z jiného počítače

Nespornou výhodou implementace CP/M na Spectru je možnost využívat programy a datové soubory vytvořené na jiných počítačích. Je pouze třeba mít tyto programy a data zaznamenaný na kartridžích. Bohužel zatím není v silách a možnostech Mikrobáze distribuovat programové produkty určené pro práci pod systémem. Tyto programy si už musí sehnat každý sám. Překopírování těchto programů na odpovídající médium (kartridže, popř. kazety) lze provést několikerym způsobem:

1. Nejjednodušší je zřejmě zaznamenat program na zdrojovém počítači na kazetu ve formátu programu MSAVE (varianty programu MSAVE již existují např. pro IBM 7C, SHARP MZ-800 apod.). Spectrem se pak program z kazety překopíruje na kartridž pomocí programu MLOAD.
2. Rychlejší způsob přenosu je po sériové lince, popř. pomocí paralelního styku (Centronics). Sériové rozhraní mají dnes všechny počítače, toto rozhraní je vůbec nejběžnější. Přenos sériovým rozhraním je sice obvykle pomalejsí, než paralelním, ale propojení nejrůznějších počítačů lze tak snáze realizovat. Přímý přenos po lince má jedinou nevýhodu - musíme oba počítače k sobě fyzicky přinést. Po přenosu dat ze zdrojového počítače do Spectra je nejjednodušší způsob záznamu přeneseného programu na kartridž pomocí LECROMky. Uvedu příklad pro zaznamenání programu PRG.COM o délce 4 kB umístěného od adresy 26000:
 - CLEAR 25000
 - MOVE "r 26000,4096", "a 256" (tento příkaz platí pro LECROM 2/3 pro 528 kB, u starší verze 1.6 je třeba psát MOVE "r 26000,4096", "d 256")
 - LOAD *"m",drive,"cp/m"CODE
 - RANDOMIZE USR 59904
 - SAVE 16 PRG.COM
3. Nemáme-li LECROMku a chceme-li se obejít bez různých machinací ve strojovém kódu, je nejrozumnější uložit pomocí SAVE "PRG" CODE 26000,4096 program na kazetu a pak pomocí MLOAD PRG.COM jej překopírovat na kartridž (program MLOAD umí číst data zaznamenaná basicovým příkazem SAVE CODE a naopak).

Přestože minulé rádky pokládáme za samozřejmé a známé, zařadili jsme je do textu, protože autor byl již mnohokrát dotazovaný, jakým záhadným způsobem se mu vlastně podařilo programy z jiného počítače přemístit do Spectra, zda je pretilkával v hexadecimálním vyjádření apod.

Po přenesení programu do Spectra je nutno vzhledem k netypickému formátu obrazovky některé programy přeinstalovat. Proto nezbývá než přenést i instalacní program a tímto vlastní program instalovat. Pokud původní program nesezeneme v nainstalovatelné podobě, je více než pravděpodobné, že nebude fungovat (bud vůbec nebo nedostatečně). Naopak po správném nainstalování budeme

překvapení, že na Spectru obvykle program funguje lépe než na původním počítači.

V praxi jsme se přesvědčili, že i tak běžná věc, jako je instalace např. textového editoru, může působit uživateli značné těžkosti, i když instalační program se na všechny potřebné informace sám dotazuje a úkolem obsluhy je jen na otázky správně odpovědět. Proto jsme ochotni v případě problémů osobně pomoci nebo program nainstalovat.

Co se týče přenosnosti programů, je situace příznivá, neboť lze přenést téměř vše. Je nutno se pouze vyhnout programům vyžadujícím systémové stopy na disku (např. programy MOVCPM, SYSGEN nebo DOCTOR). Podobná situace nastává tam, kde program předem předpokládá určité pevně dané parametry disku.

Bohužel je však třeba se raději vyhnout všem programům, které mají něco společného s JZD Agrokombinát Slusovice. Programy z této organizace totiž využívají algoritmus, které ve svém důsledku způsobují závislost programu na konkrétní hardwarové konfiguraci. Takové programy se v některých situacích nečekaně hroutí. I když se toto zdáleka netýká všech programů, nikdy dopředu nevíme, zda je určitý program systémově závislý či nikoli. Je to škoda, protože ze Slusovic jinak pochází mnoho pěkných programů.

Při přenosu nebo kopírování každého programu je nezbytné spočítat (např. programem POWER) kontrolní součty originálních a překopírovaných programů a porovnat je na shodu. Je až neuvěřitelné, kolik vadných verzí programů koluje mezi lidmi. Chyba v programu může vzniknout jak při přenosu z jiného počítače, tak i při prostém kopírování z disku na disk. Proto radíme mít sadu kartridžů se "zaručenými" programy, a pro práci pořizovat pracovní kopie na jiné médium. Nejlépe je zálohovat originální programy na kazetách.

6.2 Instalace tiskárny uživatelem

Program BIOS již předem počítá s tím, že případný uživatel bude chtít používat tiskárnu. Problém je v tom, že není nějaký všeobecný standard připojení tiskárny ke Spectru, neboť většina amatérů používá vlastnoručně zhotovená rozhraní apod. Ovladač takového zařízení by bylo nejjednodušší prostě zabudovat do BIOSu, to ale bohužel není možné vzhledem k nedostatku místa v horní části paměti (BIOS je na adresách EA00H až FFFFH - paměť je tedy zcela obsazena). Celá situace je vyřešena jinak. Po spuštění CP/M zůstává volná část videopaměti VRAM, a to na adresách 6200H až 64FFH (25088 až 25855 dekadicky). Do této oblasti lze umístit případné uživatelské ovladače. Protože však rutinu umístěnou v této lokaci není možné volat jednoduše přepsáním adresy ve vstupním vektoru BIOSu (paměť VRAM není nastránkována do adresového prostoru procesoru a registr SP ukazuje na nevhodné místo), je třeba obslužnou rutinu volat až po přestránkování a nastavení SP někam do horní části paměti (nad počáteční adresu BIOSu). Za tímto účelem je rozšířen vstupní vektor BIOSu o další čtyři položky (skoky), které následují za sedmnácti původními (standardními) položkami a mají tento význam:

Adresa	Služba
EA33H	Výpis znaku na tiskárně
EA36H	Zjištění stavu (připravenosti) tiskárny
EA39H	Vyslání znaku na děrovač (sériové rozhraní)
EA3CH	Přečtení znaku ze snímače (sériového rozhraní)

V originálním BIOSu, tak jak jej máte na kazetě, jsou tyto body ošetřeny následovně:

EA00	BIOS	EQU ØEA00H
EA33		ORG BIOS+33H
EA33 Ø0	LIST:	NOP
EA34 Ø0		NOP
EA35 C9		RET
EA36 3EFF	LISTST:	LD A,ØFFH ;tiskárna vždy připravena
EA38 C9		RET
EA39 Ø0	PUNCH:	NOP
EA3A Ø0		NOP
EA3B C9		RET
EA3C 3EØØ	READER:	LD A,1AH
EA3E C9		RET

Uživatel má možnost na místo některých tří bytů v tomto přídavném vektoru umístit instrukci skoku (JP) na svůj podprogram, který je uložen někde v oblasti nad adresou 6200H. Zde je pro uživatele vyhrazen prostor o délce 300H

bytů (tj. 768 dekadicky), což by mělo postačovat i pro ty nejnáročnější ovladače. Jedna poznámka se týká ještě způsobu zavedení uživatelské rutiny společně se systémem. Aby nebylo nutné nahrávat z Microdrive další modul (beztak už jsou čtyři), uživatelská rutina se umístí do modulu "cp/m" od adresy CCP-800H (tj. 52224 dekadicky) a takto modifikovaný modul se nahraje místo původního. Systém při svém zavádění sám přemístí uživatelské rutiny na správné místo.

Uživatelské rutiny musejí být napsány tak, aby používaly jen základní registry procesoru, tj. AF, BC, DE a HL. Není povoleno používat AF', BC', DE', HL', IX a IY, ani SP v jiné funkci než jako ukazatel zásobníku. Registr IY navíc neobsahuje hodnotu 5C3AH (23610 dekadicky), jak jsme na to u Spectra zvyklí, nelze tudíž pracovat s indexovým adresováním podle IY. Rutina během svého běhu NESMÍ ZAKÁZAT PŘERUŠENÍ, přes které běží obsluha Microdrive a klávesnice. Pokud používáte verzi BIOSu 4.0 pro 272 a 528 kB, můžete přerušení v krajním případě na chvíli zakázat (max. 50 ms) – ale pak musí zůstat alespoň 50 ms povolen. V programu je s tím počítáno. U verze 3.3 pro 80 kB však nelze přerušení zakázat vůbec. Mohlo by to mít za následek zhroucení celého systému. BIOS zajistuje, že uživatelský podprogram je volán při nastránkovém módu 48 kB (obyčejné Spectrum) a registr SP ukazuje na místo do BIOSu, pod kterým je volno nejméně 20H bytů pro používání zásobníku. Je povoleno přerušení v módu IM 2. Rutina obslužní periférie a vrátí se zpět instrukcí RET. Po návratu se prestránkuje do módu s RAM od nuly a do SP se uloží původní hodnota, nacež se řízení vrátí volajícímu programu.

Rutina obsluhující periférii tedy může být během své činnosti přerušena. Za normálních okolností (když nedojde k chybě při obsluze Microdrive) bude návrat z přerušení proveden nejdříve ihned a nejpozději asi do 60 ms. V případě závažné chyby (např. není prítomen kartridž) může dojít i k tomu, že návrat z přerušení se uskuteční na jinou adresu než kde k přerušení došlo (instrukcí JP namísto RET). Je nanejvýš vhodné napsat obslužný podprogram tak, aby mu uvedené přerušení nevadilo. V opacném případě se bude systém zdržovat a bude nedokonale pracovat klávesnice (občasné ignorování stisku klávesy).

V některých situacích se bez zákazu přerušení stejně neobejdeme. Typickým příkladem je obsluha kazetového magnetofonu. Přerušení lze ale zakázat jen tehdy, jestliže neběží žádná mechanika Microdrive. O chodu nebo nečinnosti mechanik se lze přesvědčit jednoduchým podprogramem na základě čtení status registru řadice Microdrive.

Každopádně je však třeba počítat s tím, že pro CP/M nelze napsat obslužný podprogram pracující v reálném čase, a to ani v případě zákazu přerušení. Běh programu v lokaci VRAM je výrazně a v podstatě nepravidelně brzděn obvodem ULA.

Pokud by vadilo, že nemůžete vytvořit časové smyčky pro obsluhu některých periferií, obraťte se se svým problémem na autora BIOSu, který na základě Vašich písemných připomínek zpracuje samostatné pojednání o tomto (a nejen o tomto) problému.

Obecně vůbec nelze doporučit ani volání rutin do ROM, protože tyto rutiny používají registry IX, IY, indexovou adresaci přes IY, instrukce EXX či EX AF,AF' a někdy také zakazují přerušení (např. ZX Printer). Není možné volat rutiny do přídavných ROMek (ZX Interface 1 apod.), BIOS by pak nemohl zjistit, která stránka paměti je právě přístupná a v případě odstránkování ROM by došlo ke zhroucení (kolize na sběrnici). Je zbytečné také používat systémové proměnné (kromě výše uvedených), protože systém s nimi nepracuje. Navíc rutiny v ROM obsluhující periférie testují stisk klávesy BREAK, která má v CP/M jinou funkci, a v případě jejího stisku by došlo k opuštění operačního systému a spuštění BASICu. Následně by se počítac zakousl, protože by nebyla čtena klávesnice.

Rutiny si musí předávat parametry (vysílaný nebo přijímaný znak). Vstupní parametr se ukládá před voláním rutiny do registru C (rutiny LIST a PUNCH). Pokud rutina znak vrací (týká se LISTST a READER), ukládá ho před návratem do registru A (akumulátoru). Návratový parametr ze služby LISTST (připravenost tiskárny) má hodnotu FFH, když je tiskárna připravena, nebo 00H v případě její nepřipravenosti.

Je samozřejmé, že rutiny neprovádějí žádné znakové konverze (CP/M nepoužívá TOKENY, PRINT AT, PRINT COMMA apod. ani nic podobného jako Sinclair ROM), prostě jaký znak přijde, takový se zpracuje. Rutiny také nehlídají stisk klávesy BREAK. Pochopitelně se nelze ani vracet do BASICu pomocí RST 8 a dělat podobné zbytečnosti.

Následuje podrobný návod, jak do BIOSu zainstalovat uživatelské rutiny pro tiskárnu a jiné periférie:

1. Napsat obslužné podprogramy v jazyce symbolických adres (asembleru) a přeložit je od adresy 6200H (25088 dekadicky). Celková délka všech rutin nesmí být větší než 300H bytů a rutiny musí být napsány dle výše uvedených pravidel.
2. Provést SAVE *"*m*",drive,"rutiny"CODE 25088,délka
3. Zadat CLEAR 52223: LOAD *"*m*",drive,"cp/m"CODE
4. Zadat LOAD *"*m*",drive,"rutiny"CODE 52224
5. Na místa přidaných vektorů BIOSu umístit instrukce skoku na příslušné obslužné rutiny, a to na místo, kde budou VE SKUTEČNOSTI PRACOVAT (BIOS rutiny přemístí). Např. jedná-li se o rutinu pro tiskárnu, byla přeložena od adresy 6200H a umístěna je na adrese CC00H. Její volání bude umožněno umístěním instrukce JP 6200H na adresu EA33H.
6. Provést ERASE "*m*",drive,"cp/m"

7. Provést SAVE *"m",drive,"cp/m"CODE 52224,13312

Je-li potřeba před používáním interface jej naprogramovat vysláním řídícího slova, lze toto naprogramování provést v zaváděcím basicovém modulu "run" (např. OUT 127,152 pro interface s 8255).

6.3 Instalace systému pro 80 kB RAM

Spuštění CP/M na osmdesátkilobytovém Spectru není tak jednoduchou záležitostí, jak by se mohlo na první pohled zdát. Překopírování systému z distribuční kazety na kartridž lze sice provést postupem popsaným na začátku šesté kapitoly, avšak takto překopírovaný systém není možno spustit. Správný start totiž vyžaduje, aby v první mechanice byl zasunut kartridž se záznamem ve formátu CP/M, což nelze na obyčejném 48 kB Spectru zajistit.

Za tímto účelem byl vytvořen zvláštní program. Je pod názvem "inst" zaznamenán na distribuční kazetě, a sice na straně určené pro 80 kB Spectrum. Obsahem programu je mírně upravený systém, dále program FORMAT a MLOAD přeložené od vhodných adres. Program inst slouží k prvnímu spuštění systému (bez jinak nutného kartridže), naformátování alespoň jednoho nejnuttnejšího kartridže a záznamu několika základních programů na tento kartridž. Postup je takový:

1. Vyresetování počítače (RANDOMIZE USR Ø)
2. LOAD "inst"
3. Po nahrání programu se spustí program FORMAT. Vložíme připravený kartridž a naformátujeme jej (viz obsluha programu FORMAT). Po úspěšném formátování vystoupíme z programu klávesou ENTER.
4. Na kartridž zaznamenáme program MLOAD příkazem
SAVE 5 MLOAD.COM
5. Můžeme přistoupit k záznamu programů PW.COM, FORMAT.COM a MSAVE.COM. Provedeme to pomocí
MLOAD *.*
a překopírování uvedených programů z distribuční kazety.
6. Po opuštění programu MLOAD je systém připraven k používání.

7. Rozšíření operační paměti počítače

Operační systém CP/M vyžaduje pro svou činnost paměť RAM začínající od adresy 0000H. Proto je nutné Spectrum přestavět. Jsou využity tři varianty rozšíření, a to na 80, 272 a 528 kB, v první z nich však z již uvedených důvodů pracuje CP/M poměrně pomalu. Připravuje se však cenově přístupnější verze s pamětí RAM 2 x 64 kB, která může pracovat s BIOSem 4.0. Použití RAM disku v této variantě samozřejmě není možné. Verze 80 kB byla popsána v [4], proto zde popíšeme pouze verzi 272/528 kB.

Pro rozšíření paměti je použito osm, resp. šestnáct paměťových obvodů 41256 nebo 4256 s dobou výběru max. 200 nanosekund a osmibitovým občerstvovacím cyklem (to jsou dnes všechny). Destička s plošnými spoji je navržena tak, aby ji bylo možno umístit do všech vyráběných verzí (ISSUE) Spectra, tzn. od verze 2 do poslední 6. Úprava nebyla doposud realizovaná na originálním Spectru 128. Zapojení nevyužívá samoobčerstvovací schopnosti uvedených obvodů, neboť by to neznamenalo úsporu v počtu dalších obvodů.

Nejdříve je nutno se zmínit o tom, jak se v obyčejném Spectru adresuje paměť. Výberové obvody pro ROM a VRAM jsou umístěny v obvodu ULA a pracují následovně: jsou-li A15 a A14 ve stavu L, je vybrána ROM; při A15=L a A14=H se pracuje s VRAM; při A15=H není ROM ani VRAM vybrána a pracuje se s pamětí RAM 32K v horní polovině adresového prostoru. K získání řídících signálů pro tyto paměti RAM je třeba ještě vyhodnocovat signály MREQ, RD a WR. Tyto funkce výběru zajišťují u verzí 2, 3 a 4 obvody 74LS00 a 74LS32, u verze 5 a 6 jediný čtyřicetipinový obvod ULA2. Paměť RAM je aktivní, je-li A15 na 74LS00 (ULA2 pin 37) v úrovni H.

Této skutečnosti je využito v popisovaném zapojení - viz obr. 1. K řízení a stránkování paměti RAM slouží ovládací port na adresě 253 (0FDH). Tento port byl zvolen z nasledujících důvodů:

- U normálního Spectra není k ničemu využit.
- Možnost jednoduché lineární adresace od vodiče A1.
- Přímá návaznost na rozšíření 80 kB z [4].
- Tentýž port je obsazen u Spectra 128, +2 a +3, kde se jím ovládá zvukový generátor, tiskárna a stránkování paměti.

Uvedené řešení oproti jiným má dvě významné výhody:

- Stránkováním se majitel Spectra nezbaví možnosti dalšího rozšiřování paměti.
- Nedochází ke kolizím na sběrnici po připojení sériově vyráběných interfejsů se stránkovanou ROM (ZX Interface 1, Opus Discovery, Beta-Disk, Wafadrive, Disciple a další). Tuto nepříjemnou vlastnost mají všechna doposud uverejněná řešení stránkování. Zmíněné interfejsy stránkovují svou ROM při výskytu určité adresy na adresové sběrnici při běhu programu. Konstruktérů jiných variant stránkování paměti se mylně domnívají, že ke stránkování dochází např.

detekováním operačního znaku instrukce RST Ø8H apod. Stránkování je, jak jsme již řekli, aktivováno určitou adresou na adresové sběrnici. Při výskytu těchto stránkovacích adres v programu, který běží v aktivní paměti RAM od adresy 0000H (a tento stav nelze v CP/M vyloučit), dojde k současnému připojení přistránkovane paměti RAM a vnější ROM na datovou sběrnici. Vzniklá situace má pochopitelně za následek zhroucení Spectra. Stránkování popsané v tomto textu znemožňuje aktivaci vnější ROM tím, že při odstránkování vnitřní ROM je vodič A15' na konektoru počítače v úrovni H.

Výběrový obvod portu je tvořen dvěma hradly OR IO1c,d, který při aktivních signálech IORQ, WR a A1=L zapíše při vzestupné hraně signálu WR data z datové sběrnice do šestinásobného D registru, IO5. Rezistor mezi IO1c a registrem spolu se spínačem slouží k zablokování zápisu do registru (nutno užít u některých her). Po připojení napájecího napětí nebo po resetu se registr nastaví do výchozí polohy, tj. mód 48 kB.

K významu jednotlivých bitů portu:

Bit 7 - při úrovni H odpojuje ROM a VRAM a připojuje RAM od adresy 0000H - jedná se vlastně o mód 80 kB podle [4].

Bity 6, 5, 4 (a případně 3 u verze 528 kB) - určují číslo aktivní stránky dolní části paměti (každá stránka má velikost 32 kB).

Bity 2 a 1 - nejsou využity.

Bit Ø - softwarově ovládaná LED.

Z klopného obvodu D, bitu 7 je řízeno hradlo OR IO1b, které posílenou A15 upraví na A15'. Je-li výstup klopného obvodu D na úrovni L, je A15'=A15 (Spectrum se chová jako původní 48 kB). Je-li tento výstup na úrovni H, je A15'=H. Tím, že A15' je vedena na obvod ULA, na přímý konektor Spectra a na 74LS00 (ULA2), je zajištěno odpojení ROM a VRAM a připojení RAM od adresy 0000H.

Bity 6 a 5 jsou vedeny z registrů na IO7, který multiplexuje devátý adresový vodič A8-256 pro 41256 (pin 1). Přepínacím signálem je signál "S" u 74LS157 (ULA2 pin 35). Za IO7b je hradlo NOR IO2c, které zajistuje spolu s hradly OR IO6a,d připojení nejvyšší stránky RAM od adresy 8000H.

Bit 4 je veden přes hradlo OR IO6a na vstup "A" multiplexeru 74LS157 (ULA2 pin 25/24) místo propojky TI (viz obr. 2).

Bit 3 se zapojuje pouze u verze 528 kB a hradla IO6b,c,d a IO2b slouží k přepínání signálu CAS mezi jednou a druhou sadou paměti.

Bit Ø je rezervní a v současné době zajišťuje pouze programové ovládání diody LED. Zapojení tohoto bitu lze

vypustit nebo jej využít pro jiné účely (např. přepínání dvou ROM).

Protože všechny paměti 41256, ale i některé 4164 mají osmibitový občerstvovací cyklus, je nutno osmý bit občerstvovací adresy generovat zvláštním obvodem (procesor Z80 poskytuje pouze sedm bitů). Do registru IO4a je při občerstvovaní vzestupnou hranou signálu MREQ (přitom je RFSH ve stavu L) přepsán stav A6, na kterém je v tomto okamžiku platná refresh adresa. Druhý klopný obvod D, IO4b, je zapojen jako dělič kmitočtu dvěma. Obvod IO3 tvorí multiplexer pro A14 a osmý bit refresh adresy. Signál A14' je veden na vstup "B" multiplexera 74LS157 (ULA2 pin 24/25) místo propojky OKI, viz obr. 2.

U verzí Spectra 2, 3 a 4 má podle firemního schématu obvod 74LS00 označení IO24, 74LS32 IO23 a 74LS157 IO26.

Celý doplněk je navržen tak, aby jeho realizace byla co možná nejsnadnější. Je použito pouze jednostranné desky s plošnými spoji, bez prokovených otvorů. Obrazec spojů je na obr. 3, osazení součástkami znázorňuje obr. 4. Destička se součástkami se ve Spectru 2, 3 a 4 přesně vejde mezi procesor Z80 a obvod ULA, u verze 5 a 6 je ji možno umístit do pravého dolního rohu - v krabici je místa více než dost.

Místa nejvhodnějšího přerušení A15 a CAS na původní spojové desce jsou vyznačena na obr. 5. Při osazování 41256 je nutno pin 1 vyhnout do strany a u všech pamětí spojit izolovaný vodičem s A8-256. Při použití šestnácti kusů paměti se tyto musí pájet po dvou na sebe s tím, že horní sada má piny 15 vyhnuté nahoru a spojené se signálem CAS1.

Pro případnou modifikaci 80 kB se osazují pouze IO1 a IO5, pro refresh ještě IO2, IO3 a IO4. Z vodičů vedoucích do základní desky se zapojí A15, A15', D7, WR, IORQ, A1, RESET a A15'' se nahradí původní A15. Pro refresh je potřeba ještě A6, A14, A14', MREQ a RFSH.

Pro 272 kB se zapojí vše, kromě LED, CAS, CAS0 a CAS1.

U 272/528 kB verze je poslední stránka přístupná dvakrát, takže při psaní programu je třeba dát pozor, aby program nepřepisoval své vlastní instrukce.

Následuje seznam signálů Spectra, které jsou využívány popisovaným doplňkem, spolu s jejich nejvhodnějším místem pro připojení na původní desce:

- A15 - CPU Z80 pin 5,
- A15' - ULA pin 37, přímý konektor 1B, (ULA2 pin 37),
prokovený otvor - viz obr. 5,
- A15'' - místo propojky TI - viz obr. 2,
- CAS - IO23 74LS32 pin 6 (ULA2 pin 34),
- A14 - střed propojek TI a OKI - obr. 2,
- A14' - místo propojky OKI - obr. 2,

"S" - IO26 74LS157 pin 1 (ULA2 pin 35),
A6 - CPU pin 31, IO25 pin 13, IO3, (ULA2 pin 21),
MREQ - CPU pin 19, IO23 piny 1 a 2, ULA pin 38, (ULA2 pin
36),
RFSH - CPU pin 28,
WR - CPU pin 22, ULA pin 2, IO23 piny 12 a 13, (IO28 pin
5),
IORQ - CPU pin 20,
RESET - CPU pin 26,
D0 - CPU pin 14,
D3 - CPU pin 8,
D4 - CPU pin 7,
D5 - CPU pin 9,
D6 - CPU pin 11,
D7 - CPU pin 13.

8. Literatura

- [1] Dickens, A.: The QL Advanced User Guide, Adder Publishing, 1984.
- [2] Gianluca, C.: Spectrum Shadow ROM Disassembly, Melbourne House Publishers, 1985.
- [3] Krejča, T.: ZX floppy, připojení floppy disků k mikropočítači, 666. ZO Svazarmu, 1987.
- [4] Lamac, J.: 80 K RAM pro ZX Spectrum, Zpravodaj Mikrobáze č. 06/1987, str. 41.
- [5] Logan, Y.: ZX Spectrum ROM Disassembly, Melbourne House Publishers, 1983.
- [6] Navrátil, Z.: OS MIKROS, Príručka pre implementáciu, Kancelárske stroje, 1985.
- [7] Navrátil, Z.: OS MIKROS, Príručka programátora, Kancelárske stroje, 1985.
- [8] QL Service Manual, Sinclair Research Ltd, 1985.
- [9] Richta, K., Zajíc, J.: Operační systém CP/M pro mikropočítače, Český výbor elektrotechnické společnosti ČSVTS, 1986.
- [10] Service Manual for ZX Microdrive, Interface I and Interface II, Sinclair Research Ltd, 1984.
- [11] Service Manual for ZX Spectrum, ZX Spectrum+, Sinclair Research Ltd, 1983.
- [12] Troller, P., Císař, P.: Úprava adresování a zvětšení rozsahu paměti počítače ZX Spectrum, Sdělovací technika č. 11/1987, str. 417.

S E Z N A M P Ř Í L O H

Obr. 1 - schema zapojení

Obr. 2 - vedení drah a propojek na základní desce

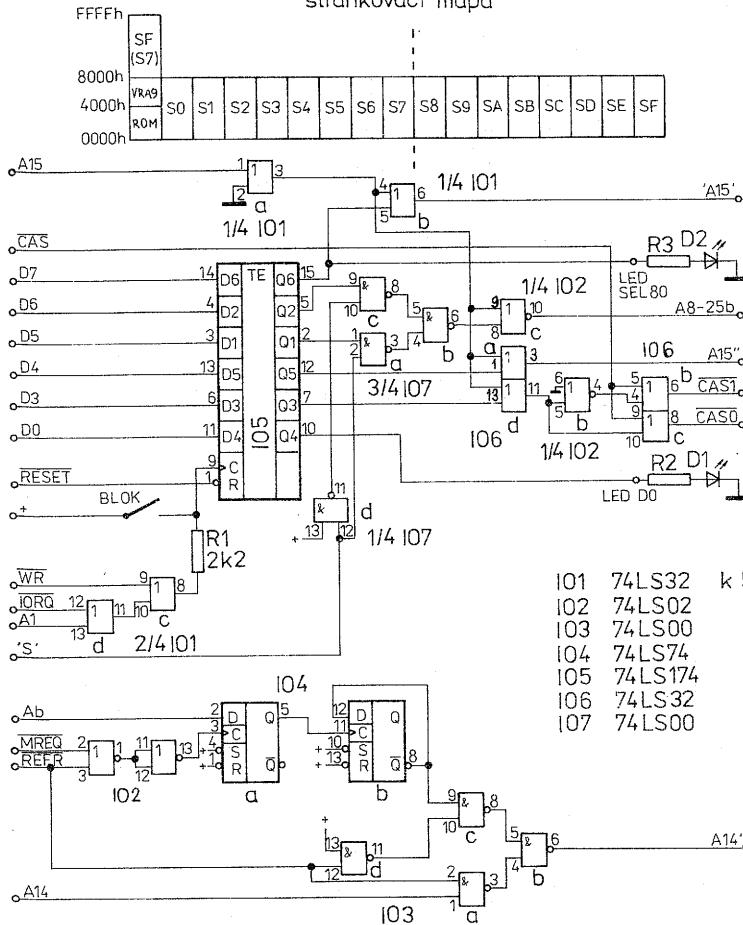
Obr. 3 - deska s plošnými spoji

Obr. 4 - osazení desky součástkami

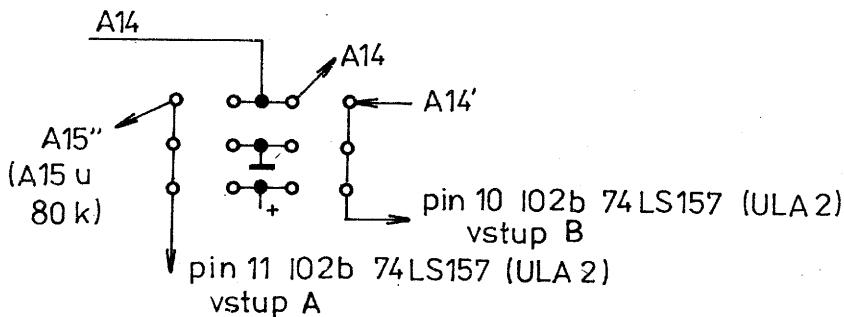
Obr. 5 - přerušení A15

(3 listy obrázků)

stránkovací mapa

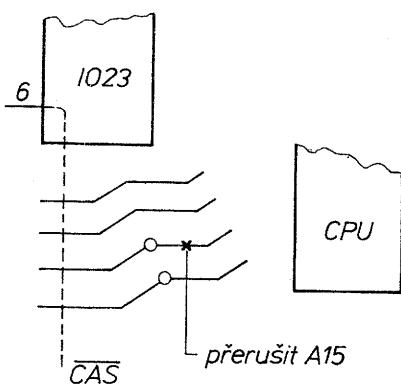


obr. 1 - schema zapojení

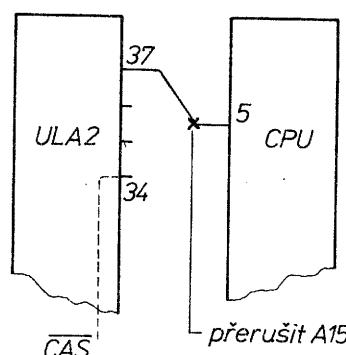


obr. 2 – vedení drah a propojek na základní desce

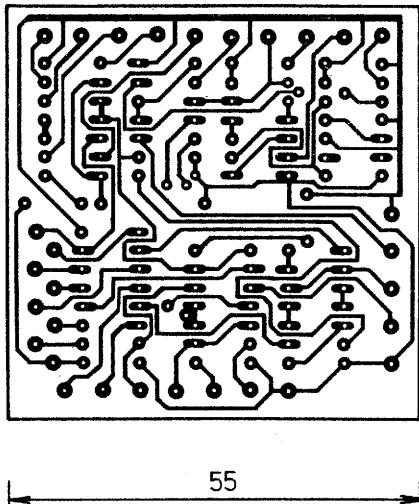
verze 2,3 a 4



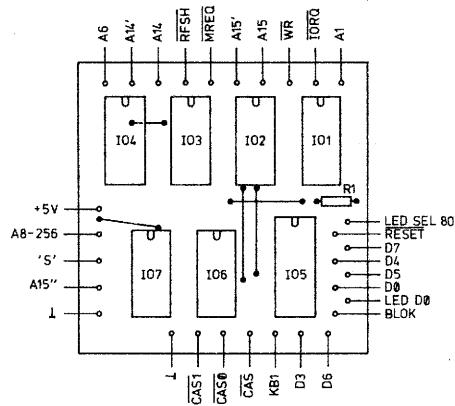
verze 5,6



obr. 5 – přerušení A15



obr. 3 - deska s plošnými spoji



obr. 4 - osazení desky součástkami

Vydala 602. ZO Svazarmu jako součást dodávek programového vybavení v rámci služeb Mikrobáze. Samostatně neprodejné!
Adresa vydavatele a výrobce: 602. ZO Svazarmu, Wintrova 8,
160 41 Praha 6.