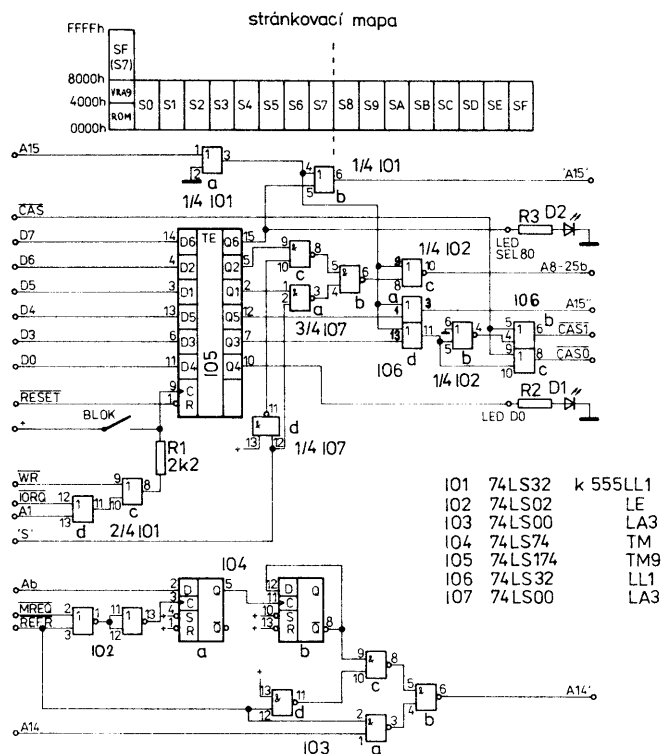
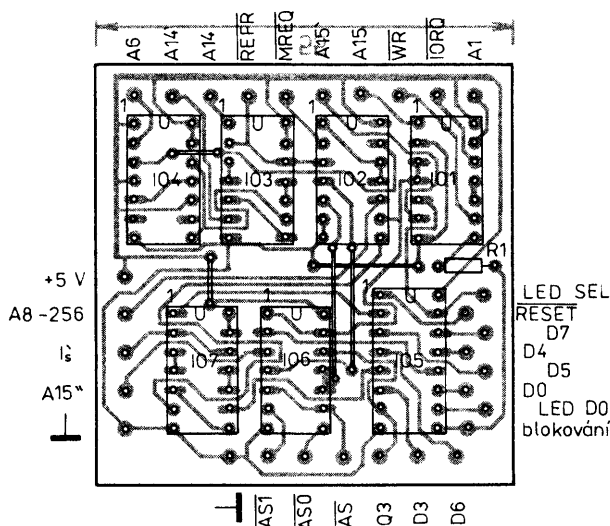


mikroelektronika

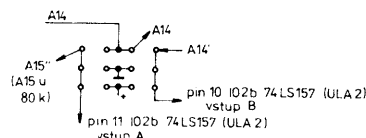


Obr. 1. Schéma zapojení řídicích obvodů



Obr. 2. Zapojení propojek verzí 3, 4, 5 a 6

Obr. 4. Rozložení součástek na desce s plošnými spoji



CP/M na ZX Spectrum

Jak jsme již dříve slíbili, přinášíme dnes kompletní návod na rozšíření paměti pro mikropočítač ZX Spectrum na 80 až 528 kB a podrobný popis implementace operačního systému CP/M 60 kB v. 2.2 na tento počítač. Autory jsou Jiří Lamač — software a Jakub Vaněk — hardware, text článku napsal Daniel Meca.

Různých návodů na úpravu paměti tohoto u nás vůbec nejrozšířenějšího počítače vyšlo již i na stránkách AR velké množství. Přesto si právě tato úprava zaslouží zvláštní pozornost. Na rozdíl od ostatních se totiž autoři neomezili jen na pouhé rozšíření paměti, ale na maximální rozšíření možností počítače, při zachování jednoduchosti a dostupné pořizovací ceny. V tom jde popisovaná úprava důsledně „ve stopách sira Sinclaira“.

Pro takto upravený počítač bylo již napsáno mnoho zajímavých a užitečných programů, které plně využívají nové konfigurace paměti a jejích možností. Mnoho uživatelů jistě zaujme komfortní kopírovací program s volným pracovním prostorem plných 64 kB, s průběžným zobrazováním velikosti paměti během nahrávání. Program umožňuje libovolně kopírovat z kazety i Microdrive a to též prostřednictvím sítě NET.

Zajímavý je nesporně i monitor a disassembler s pronikavě rozšířeným pracovním prostorem. Pro další rozšíření

možností Spectra navrhl J. Lamač novou EPROM na místo původní ROM. Ta zachovává všechny původní možnosti, navíc však opravuje chyby původního systému a obsahuje vestavěný monitor paměti. Dále umožňuje zadávat klíčová slova BASICu vypsáním, zkratkou, nebo jedinou klávesou. Pomocí této EPROM lze též přímo z BASICu ovládat RAM-DISK. Korunou tohoto snažení je však úspěšná implementace operačního systému CP/M v. 2.2, která povýšila rázem počítač, původně určený převážně pro televizní hry, do kategorie blízké profesionálním počítačům.

Něco o CP/M jako takovém

První použitelná verze operačního systému CP/M se rozšířila do světa již v roce 1976, kdy jeho autor, Gary Kildall, založil firmu Digital Research. Tehdy se jednalo o verzi 1.3. Následovaly verze 2.2 a 3.1, které okamžitě po svém zveřejnění opanovaly sféru osmibitových počítačů a své prvenství si udržely dodnes. Ostatní operační systémy, ať už to byl multiprogramový Concurrent CP/M, multiuživatelský MP/M, případně CP/NET, který mohl sdílet prostředky v rámci počítačové sítě, přestože z CP/M bezprostředně vycházely, se nikdy tak masově nerozšířily. Z CP/M v podstatě vycházel i operační systém ISIS II firmy Intel, který je určený pro vývojové systémy s procesorem 8080. S nástupem šestnucibitových procesorů rozhodně sláva CP/M nepohasla ale promítla se do operačního systému CP/M-86.

Zde je dobře si uvědomit, že podstatou velkého úspěchu CP/M byla možnost přizpůsobení pro nejrůznější hardwarové konfigurace počítačů různých výrobců. Jeho jádro je totiž systémově nezávislé a přizpůsobení ke konkrétnímu počítači zprostředkovává pouze modul BIOS. Programy pracující nad CP/M jsou pak plně přenositelné z jednoho počítače na druhý a vzhledem ke shodným základním instrukčním kódům, mohou většinou pracovat s procesory 8080 i Z80.

Masová produkce IBM PC naopak donutila ostatní výrobce přizpůsobit své šestnáctibitové počítače nejen operačnímu systému MS-DOS, ale i hardwarovému uspořádání vzoru.

Přestože se v ČSSR začínají šířit i počítače kompatibilní s IBM PC, není jich zatím tolik, aby se dalo mluvit o masovém rozšíření. Ostatně, na řadu aplikací by byl IBM PC zbytečně velký kalibr. Osmibitové počítače tedy ještě dost dlouho budou u nás patřit mezi nejrozšířenější v profesionálním nasazení. A zde se právě používá CP/M v. 2.2, buď pod označením MIKROS u našich počítačů, nebo jako SCP u dovážených počítačů Robotron 1715 z NDR.

Při prudkém rozvoji elektronizace našeho hospodářství je zapotřebí mnoha zkušených programátorů i operátorů k počítačům. Nejlepším školením je jistě praxe a tu umožní mimo jiné i zmíněná implementace CP/M na nejrozšířenější mikropočítač Sinclair ZX Spectrum. Úpravy schopný by byl i kompatibilní Didaktik Gama, výrobek družstva Služba Skalica, který je na našem trhu. Na úrovni tohoto operačního systému se však mohou sejít uživatelé Spectra s uživateli řady dalších u nás dostupných osobních mikropočítačů a vyměňovat si programy. Berte to jako výzvu k implementaci tohoto systému i na další počítače, pokud jím nejsou vybaveny již výrobci.

Daniel Meca

ZX Spectrum 528 kB Úprava vhodná k implementaci CP/M

Následující zapojení umožňuje rozšířit paměť počítače ZX Spectrum na 80 až 528 kB tak, aby mohl mimo jiné pracovat též s operačním systémem CP/M. Byla ověřena výborná implementace CP/M v. 2.2, jejímž autorem je J. Lamač. Při použití paměti RAM 4164, je dosažena kapacita paměti 80 kB, při jedné sadě 41256 (4256) vzroste na 272 kB, v případě použití dvou sad, až na 528 kB. Pro celé zapojení je navržena deska plošných spojů, která umožňuje všechny popsané varianty osazení. Její rozměry jsou zvoleny tak, aby ji bylo možné vestavět přímo do počítače (i do tzv. „gumáku“) (vlastní paměti se osazují přímo do základní desky počítače).

S výjimkou [1] a [2] používala všechna dosud známá zapojení rozšíření paměti stránkováním horních 32k, takže se nehodila k implementaci operačního systému CP/M. V literatuře [2] je sice použito stránkování spodní části adresového prostoru, ale zapojení je příliš složité a nepočítá zřejmě s dalším rozšiřováním paměti. Navíc se zapojení nevejde do počítače. Podstatným nedostatkem zapojení podle [2] je však to, že při použití ZX Interface 3, Beta-disk, Opus Discovery, ZX Interface 3, Kempston E, Lprint 4 a dalších standardních periférií, stránkujících ROM, dochází ke kolizím na sběrnici. To je u našeho zapojení uveřejněného v [1] i u zde předkládaného, softwarově s ním zcela kompatibilního zapojení, zcela vyloučeno.

Popis zapojení

Především něco o tom, jak je v ZX-Spectrum adresována paměť. Od adresy 0000h do 3FFFh je normálně paměť ROM. V oblasti od 4000h do 7FFFh je fyzicky oddělená RAM, v jejíž části je uložena

obrazová paměť a která je přednostně obsluhována zobrazovací částí obvodu ULA. V dalším textu bude označována jako VRAM. Od 8000h do FFFFh je zcela nezávislá RAM.

Výběrové obvody pro ROM a VRAM jsou umístěny v obvodu ULA a pracují následovně:

A15	A14	ROM	VRAM
0	0	; aktivní	odpojená
0	1	; odpojená	aktivní
1	x	; odpojená	odpojená

Výběrový obvod paměti RAM je u verzi Spectra 2 až 4 tvořen obvody 74LS32 a 74LS00, u verze 6 (Spectrum +) slouží k tomu účelu obvod ULA2.

V navrženém zapojení je využit stav, kdy A15 na 74LS00 (ULA2 pin 37) je na úrovni logické jedničky a paměti ROM i VRAM jsou odpojeny. Zapojení je na **obr. 1**.

K řízení a stránkování paměti RAM je použit port na adrese 253 (0FDh). Tento port byl zvolen z následujících důvodů:

- v žádné standardní periférii ZX Spectra není využit;
- možnost použití lineární adresace pomocí A1;

- přímá návaznost na rozšíření pro 80 kB z [1];

- tento port je použit též na stránkování ZX Spectrum 128k, +2 a +3, kde je však navíc, poněkud nešťastně, připojen ještě zvukový generátor a tiskárna.

Výběrový obvod rozšířené paměti je tvořen dvěma hradly OR (IO1c a IO1d). Ten při aktivních signálech IORQ=0 a A1=0 zapíše vzestupnou hranou signálu WR data z datové sběrnice do šestinasobného registru D (IO5).

Rezistor, zařazený mezi výstupem výběrového obvodu (IO1c) a hodinovým vstupem registru, spolu se spínačem slouží k zablokování zápisu do registru, takže nelze manipulovat s pamětí a počítač se chová jako bez úpravy (nutno použít u her firmy ULTIMATE a u her pro ZX 48/128).

Po připojení napájení, nebo po resetu, je registr nastaven do výchozí polohy, tj. mód 48k.

K významu jednotlivých bitů:

bit 7 — odpojuje ROM a VRAM a připojuje RAM od adresy 0000h — jedná se vlastně o mód 80k, jako v [1];

bity 6, 5, 4 a (3) — určují číslo stránky (stránka = 32k);

bit 0 — softwarově ovládaná LED.

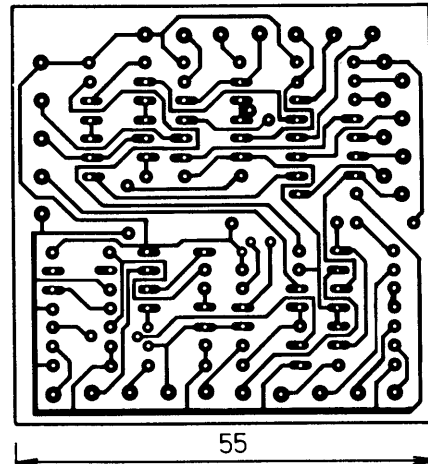
Z KO (klopný obvod) **bitu 7** je řízeno hradlo OR (IO1b), které posílenu A15 upraví na A15'. Pokud je bit 7 na log. 0, pak A15' = A15. Při bitu 7 na log. 1, je A15' trvale na log. 1. Protože A15' je vedena na obvod ULA a na 74LS00 (ULA2), je zajištěno odpojení ROM a VRAM. RAM může být tedy připojena od adresy 0000h. A protože je A15' vyvedena i na vnější sběrnici, nemůže být při takovém přestránkování na této sběrnici nižší adresa než 8000h, což automaticky vylučuje připojení vnější ROM a z toho pramenící kolize.

Bity 6 a 5 jsou vedeny z KO na IO7, který multiplexuje devět adresových vodičů A8-256 pro obvody 41256 (pin 1). Přepínacím signálem je signál 'S' původního 74LS157 (ULA2 pin 35). Za IO7b je hradlo NOR (IO2c), které zajišťuje spolu s hradly OR (IO6a,d) adresování posledních stránek (stránky '7', případně 'F') od adresy 8000h.

Bit 4 je zaveden přes hradlo OR (IO6a) na výstup A15', který se připojí na vstup 'A' multiplexeru 74LS157 (ULA2 pin 25/24) místo propojky T1 (viz **obr. 2**).

Bit 3 se používá pouze při zapojení pro 528K a hradla IO6b, c, d a IO2b slouží pro přepínání signálu CAS do příslušné sady.

Bit 0 je rezervní a v současné době je využíván pro softwarově ovládanou LED,



Obr. 3. Obrázek desky s plošnými spoji W309

která může například indikovat použití RAM atd.

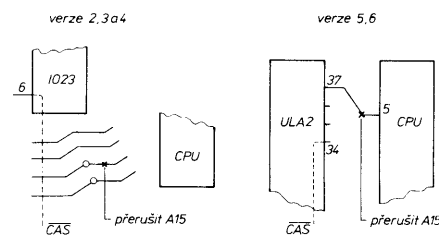
Bity 1 a 2 nejsou použity.

Všechny paměti 41256 a některé 4164 vyžadují osmibitový občerstvovací cyklus (refresh) a Z80 generuje pouze sedmibitový. Proto je součástí zapojení též obvod vytvářející osmý bit refreshu. Do klopného obvodu D (IO4a) je vzestupnou hranou signálu MREQ v době, kdy je aktivní signál RFSH, přepsán stav A6. V té době je na adresové sběrnici platná občerstvovací adresa. Druhý KO D (IO4b) je zapojen jako dělička dvěma. IO3 tvoří multiplex pro A14 a osmý bit refreshu. Signál A14' je zaveden na vstup 'B' multiplexeru 74LS157 (ULA2 pin 24/25) místo propojky OK1 (viz **obr. 2**).

U verzi Spectra 2, 3 a 4 je obvod 74LS00 označen jako IC24, 74LS32 jako IC23 a 74LS157 jako IC26.

Celé zapojení je umístěno na jednostranném plošném spoji podle **obr. 3**; rozmístění součástek je na **obr. 4**. Tato destička se vejde u verzi Spectra 2, 3 a 4 přesně mezi procesor Z80A a obvod ULA, u verze 6 je ji možno umístit prakticky kamkoliv, protože v pouzdře ZX Spectrum + je místa dostatek.

Místa nejvhodnějšího přeškrábnutí A15



Obr. 5. Úpravy na desce počítače u jednotlivých verzí

a CAS jsou vyznačena na **obr. 5**. Při osazování 41256 je nutno pin 1 ohnout stranou a u všech osmi (šestnácti) obvodů spojit s A8-256. Při použití šestnácti obvodů 41256 se musí vždy dva obvody umístit na sebe s tím, že horní sada má piny 15 vyhnuty nahoru a spojeny vodičem se signálem CAS1.

Pro modifikaci 80k se osazují IO1 a IO5, případně pro refresh IO2, IO3 a IO4. Z vodičů se zapojí A15, A15', D7, WR, IORQ, A1, RESET a místo A15' se ve Spectru zapojí A15. V případě použití obvodu pro generování osmého bitu občerstvení se zapojí A6, A14, A14', MREQ a REFR. Pro variantu 272k se zapojí vše, kromě D3, CAS, CAS0, CAS1.

Při použití variant s pamětí 272k a 528k je vždy poslední stránka RAM přístupná dvakrát. Je to třeba mít na zřeteli při psaní programů.

Důležité vývody v ZX Spectru:

A15 ... pin 5 CPU Z80;
A15' ... pin 37 ULA, konektor 1B, nebo pin 37 ULA2 (**obr. 5**);
A15'' ... místo propojky T1 **obr. 2**;
CAS ... pin 6 IC23 (74LS32), nebo pin 34 ULA2;
A14 ... střed propojek T1 a OKI (**obr. 2**);
A14' ... místo propojky OKI (**obr. 2**);
'S' ... pin 1 IO26 (74LS157), nebo pin 35 ULA2.

Jakub Vaněk (autor zapojení), **Daniel Meca**

Literatura

- [1] Lamač, J.: 80k RAM pro ZX Spectrum. Mikrobáze 06, str. 141.
- [2] Troller, P.; Císař, P.: Úprava adresování a zvětšení rozsahu paměti počítače ZX Spectrum. ST 11/87, str. 417.
- [3] Nepřehledné množství článků v AR i v zahraniční literatuře.

Implementace CP/M na ZX Spectrum

ZX Spectrum je dnes u nás jednoznačně 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 různých firem tyto hry naprosto převažují. Kvalitních systémových a uživatelských programů není zas tak mnoho. Navíc se šíří pokoutním způsobem, mnohdy v poškozené verzi a hlavně, málokdy s návodem. A tak ti, kteří by třeba i chtěli dělat něco „lepšího“, nemají možnosti. Existují sice překladače jazyka PASCAL, assembleru a C, nedosahují však úrovně obvyklé u programů pro „profesionální“ počítače. Především však jsou většinou určeny pro práci s kazetovým magnetofonem, což je zařízení již patřící minulosti. Hlavní nedostatek Spectra však vidíme v tom, že na něm není možno přímo bez pomocných programů pracovat se sekvencemi souborů.

Popisovaný operační systém tyto nedostatky beze zbytku řeší. Je možný jednoduchý přístup k souborům, lze užívat kvalitní programy firem zvučných jmen, jako je WordStar, TURBO-Pascal, MACRO 80, dBASE II, Fortran a další. Důležitý je také fakt, že Spectrum získá plnou kompatibilitu s jinými osmibitovými počítači u nás, jako je např. TNS, SAPI 1, ROBOTRON 1715, ale i Commodore 128, Schneider/Amstrad 664 a 6128. Sharp 821 a další.

Základní údaje o systému

Popisovaný operační systém pro Spectrum je vytvořen na základě u nás používaného operačního systému MIKROS, což je v podstatě klasický CP/M verze 2.2. Vzhledem k větší délce BIOSu je systém vygenerován pro 60k RAM, aby se do paměti vůbec vešel (BIOS V3.2 je dlouhý asi 5 kB, u verze 4.0 se počítá asi s 6,5 kB). Může využívat jednu až tři mechaniky Microdrive, případně ještě RAM disk.

Veškeré programy pracující pod OS MIKROS jsou pak bez omezení na Spectrum

přenositelné. Výjimkou jsou snad jen systémové programy, které již předem předkládají určité dané parametry disku, jako je velikost adresáře, počet systémových stop apod. a nezískávají je čtením systémových diskových tabulek. Zdá se, že se to týká jen programů MOVCPM, SYSGEN a DOCTOR, jejichž verze pro Spectrum dokončujeme. Popis uspořádání a funkce operačního systému MIKROS byl před časem publikován na zelených stránkách AR.

Technické předpoklady

Používání CP/M na Spectrum se bohužel neobejde bez zásahu do obvodů počítače. Je totiž nutné, aby paměť RAM začínala na adrese 0000h. Byla proto navržena velmi jednoduchá úprava, kterou lze snadno zvládnout i amatérsky, spočívající ve výměně osmi původních pamětí RAM 4532, za obvody 4164, nebo 41256. Přepnutí do módu CP/M se provede tak, že se znemožní výběr ROM a VRAM (VideoRAM), které stejně k ničemu nepotřebujeme, a na jejich místo se nastrákuje nově připojená paměť RAM (adresa 0000h až 7FFFh).

Pomocí stránkovacího portu je možno volit, která stránka přidané paměti bude aktivní. Při použití paměti 4164 vznikne známé Spectrum 80K, které má oproti Spectrum 48K navíc jednu stránku paměti (stránka = 32kB), paměti 41256 umožní získat navíc 7 stránek (Spectrum 272kB). Lze zapojit i dvě sady pamětí stejného typu, počítač pak má paměť o kapacitě 528kB. Výběr paměti obstarává port na adrese FDh, jenž má následující funkci:

bit 7 ... 0 — mód Spectrum 48K,
1 — mód CP/M,

bit 6, 5, 4 a případně 3 určují aktivní stránku RAM.

Jak je vidět, i po zásahu zůstane Spectrum plně kompatibilní se všemi programy pro něj určenými, které lze i nadále beze změn používat.

Zde je však nutné se zmínit o dosud publikovaných úpravách paměti Spectra. Mohlo by se totiž zdát, že námi navrhovaná úprava je další variací na staré téma. Není tomu tak. Starší úpravy totiž spočívaly vesměs v tom, že umožňovaly stránkovat paměť v horní polovině adresovaného prostoru, tj. na adresách 8000h až FFFFh, což pro náš účel nevyhovuje. Úprava popsaná ve Sdělovací technice č. 11/1987 sice stránkuje paměť dole, ale způsobuje kolize dat na sběrnici při připojení standardních periférií pro Spectrum, ať už je to ZX Interface 1, Opus Discovery, Beta-disk, ZX Interface 3, Kempston E, Lprint 4 a další. To pak přináší uživateli mnoho nepříjemných překvapení.

Při úpravě je nutno zaručit správné občerstvování nově připojených pamětí. Tyto paměti vyžadují v občerstvovacím cyklu osmibitovou adresu. Použitý mikroprocesor Z80A však poskytuje jen sedmibitové občerstvení. Součástí přidané destičky s několika IO je proto také obvod zajišťující generování osmého bitu občerstvovací adresy.

Programové vybavení

Základním požadavkem při vzniku CP/M pro ZX Spectrum bylo využití standardních periférií pro záznam dat, což je ZX Interface 1 a Microdrive. Máme pochopitelně napsanou CP/M na Spectrum i pro disketové mechaniky, což však při velmi jednoduché programové obsluze disketového řadiče nebyl žádný zvláštní problém. Základem této verze je poměrně rozšířený systém Beta-disk, který používá řadič FD 1793 a který je možno snadno amatérsky vyrobit. Podmínkou je však získání potřebných IO a disketové mechaniky. Daleko dostupnější

a naprosto nejrozšířenější jsou však Microdrive. Dovážel je před časem PZO Tuzex.

Proto tedy ZX Microdrive. Je to, pravda, poněkud specifické zařízení, které nebylo jednoduché přizpůsobit požadavkům CP/M, který vyžaduje možnost přímého přístupu na disk. Tím se dostáváme k vlastnímu systémově závislému bloku operačního systému, k BIOSu. Hlavním programem je zde totiž simulátor 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, prioritizace 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, jako je např. IBM PC. Nicméně výsledkem je funkční, použitelný systém. Verze BIOSu č. 3.2, použitelná pro celkový rozsah paměti 80K i 272K, ovládá Microdrive zřetelně rychleji než originální Sinclair ROM v ZX Interface 1. Nejnovější verze BIOSu V4.0, určená pro RAM 272K a více, pracuje s Microdrive dokonce několikrát rychleji, než podobné systémy s disketovými mechanikami.

Při použití celkového rozsahu RAM nejméně 272K, lze pracovat pod operačním systémem CP/M na Spectrum i s použitím kazetového magnetofonu. Po zdlouhavém nahrání všech programů i dat z kazety, lze totiž dále pracovat již jen s RAM diskem a nakonec data opět nahrát na kazetu. I to bylo úspěšně vyzkoušeno. Pro vážnější práci by však bylo zálohování zdroje asi podmínkou. Přesto je jisté tento způsob zajímavý, zvláště pro výukové účely.

Rutiny BIOSu Studený a teplý start

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

Teplý start ovšem funguje poněkud jinak. Na jiných počítačích se systém znovu zavádí z disku. Zde ale systém na běžných kartidžích vůbec není, aby tu nezabíral drahocenné místo. Proto se při prvním natažení systém přemístí do nevyužitých pamětí VRAM, odkud se při teplem startu instrukcí LDDR přesune na místo funkce.

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 od CHR(0) až po CHR(127). Některé klávesy mají odlišný význam — 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 řídicí 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 128 znaků. Je proto možné např. při překladu souboru assemblerem již psát další příkazy, které budou provedeny později po skončení překladu.

Výstup na terminál

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 řádků. Součástí výstupního programu jsou i podprogramy pro rolování obrazovky nahoru a dolů a pro mazání řádky, což výrazně zvyšuje komfort při editování textu. ASCII znaky CHR(20) až CHR(126), BEL, BS, LF a CR jsou zobrazovány obvyklým způsobem, seznam ostatních řídicích kódů je v následující tabulce:

Znak	Funkce	Znak	Funkce
ESC = Y x	CURSORS LEAD	ESC T	ERASE TO EOLN
ESC R	DELETE LINE	ESC E	INSERT LINE
ESC (END HIGH-LIGHT	ESC)	START HIGH-LIGHT
CTRL Z	CLS+HOME		

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

Součástí výstupní rutiny je i podprogram pro tisk kursoru, který běží přes přerušení a kursor je tudíž stále viditelný. Některé programy totiž vyžadují viditelný kursor i v době, kdy se nečeká na klávesnici. Kursor bliká i v době běhu mechanik Microdrive.

Tiskárna a děrná páska

To jsou zařízení, která má každý uživatel jiná. Proto v BIOSu nejsou obsluhované programy pro tisk a pásku zahrnuty. Za jeho vstupním vektorem jsou 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 napíše, přeloží od adresy BASIC RAMTOP + 1 a zařadí do systému na adresu CCP-800h, odkud jej systém přemístí na místo určené a zavolá. Pro uživatelské rutiny je zde volných 300h bajtů. Při volání je nastránkovaný mód 48K, povolené přerušení a SP je nastaven na adresu, pod níž je volných nejméně 20h bajtů. Uživatel smí používat registry AF, BC, DE a HL a nesmí zakázat přerušení (pozor na časovací smyčky!). Není možno také volat rutiny do ROM ani do přídavné ROM, kterou má např. ZX Interface 1 nebo jiné interfejsy.

Diskové operace

Diskové operace na úrovni Microdrive jsou zdaleka nejsložitějšími operacemi BIOSu. Jejich základem je sdílení času procesoru mezi činnostmi Microdrive a běžícím programem. To se uskutečňuje přes pře-

rušení a probíhá ve dvou různých prioritách. Vyšší prioritu mají okamžité požadavky běžících programů. Nižší prioritu má přístup BIOSu na disk při vykonávání dřívějších požadavků na sekvenci čtení a zápis. Komunikace programu se soubory se uskutečňuje přes přístupové vyrovnávací paměti. Jejich velikost je 2K u BIOSu V3.2, nebo 32 až 64K u systému 4.0.

Formát záznamu

CP/M na Microdrive využívá jiný formát záznamu než originální ROM, a to ze dvou důvodů. Jednak je zvýšená jmenovitá kapacita kartridže na 96K, jednak původní formát záznamu není možný, protože u něj není počítáno s existencí adresáře. Kartridž je implementován jako disk s jednou stopou, na které je 192 sektorů o velikosti 0,5K. Každý sektor má hlavičku o délce 4 bajty, ve které je nulový flag, číslo kartridže (2 bajty), číslo sektoru a kontrolní součet. Za hlavičkou následuje 512 bajtů dat, která jsou uvedena flagem o hodnotě FFh a zakončena kontrolním součtem. Číslo kartridže je náhodné číslo, které je generováno při formátování a slouží BIOSu k rozpoznání výměny kartridže bez počítání kontrolních součtů adresáře BDOsem. Z důvodu co největší optimalizace přístupových dob při sekvenci operacích není pevně stanoven parametr přečtení sektorů. BIOS přečtení neprovádí vůbec, sekvence po sobě následujících čísel sektorů je generována při formátování kartridže a parametr přečtení je volitelný. Uživatel tak může přístupové doby optimalizovat na minimum, s ohledem na druh prováděné práce a konkrétní časové konstanty svých mechanik Microdrive. Ty se však, bohužel, u jednotlivých kusů liší. Jako nejvýhodnější se ukázalo přečtení po dvou u kartridže určených pro záznam programu a po dvanácti pro sekvenci souborů, které vyhovuje všem mechanikám.

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

Na kartridž je možno v ideálním případě zaznamenat až 95kB dat, z čehož adresář má velikost 1kB (2 sektory). Tím je omezen maximální počet souborů zaznamenaných na jedné kartridži na dvaatřicet, což ale běžně postačuje.

Operace čtení

Sekvenci čtení sektoru je nejčastěji prováděná 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 valné většině případů tomu tak je). Když ano, sektor se předá volajícím programu a pochopitelně s mechanikou Microdrive není nijak manipulováno. Pouze při operaci otírání a zavírání souborů (přístup do adresáře) se hledá, zda mezitím kartridž někdo nevyměnil. Pokud by se prováděl kontrolní součet adresáře, jako u standardní CP/M, trvalo by to nejméně celý jeden oběh pásku. Proto tato kontrola spočívá v přečtení 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

neprovádí vždy. Jeho činnost je vypuštěna v případě, že poslední operace čtení nebo zápisu na dané mechanice neproběhla před více než dvěma sekundami. Je totiž zjištěno, že dříve než za zmíněné dvě sekundy není možno kartridž vyměnit.

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 posleze předat volajícím programu.

U systému 4.0 s rozšířenou pamětí je navíc každá operace čtení následována logickými procedurami, jejichž cílem je co nejvíce minimalizovat doby spojené se čtením souborů. I když není požadována žádná další operace čtení a běží hlavní program, BIOS si přes přerušení zapisuje přístupový buffer dalšími sektory z otevřeného souboru, dokud se buffer zcela nezaplní, nebo dokud počet sektorů nepřesáhne hodnotu 16. Tím je zajištěno, že čtení souboru z mechaniky se o 8 kilobajtů „předchází“ před vlastními požadavky na čtení a odezva BIOSu na požadavek čtení je velmi rychlá. Jedná se jen o přemístění dat v paměti RAM.

Operace zápisu

Zápis je podobně uzpůsoben velkým časovým nárokům, jako čtení. Většinou je uskutečněn tak, že je uvolněna nejméně potřebná část bufferu a do této části je zapisovaný sektor uložen. V příslušných tabulkách se zapisí informace o tom, že daný sektor má být zapsán a řízení se předá hlavnímu programu, jako kdyby zápis již byl hotov. Vlastní zápis sektoru provede BIOS později, nejčastěji přes přerušení. Někdy však je také operace zápisu provedena jako vedlejší účinek operace čtení v okamžiku, kdy je při čtení nalezen sektor, který má být zapsán. Nikdo se však nemusí obávat, že by např. při výpadku sítě neměl poslední data zaznamenaná, protože systém všechny požadované zápisy provede většinou ihned, nejpozději však do třiceti sekund od vlastního požadavku.

Práce s přístupovými buffery

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

Je-li požadována operace čtení nebo zápisu a není-li daný sektor v bufferu nalezen, začne BIOS hledat, který sektor v bufferu relativně nejméně potřebuje. Celý postup se liší podle toho, zda sektor, pro nějž uvolňujeme místo, náleží adresáři či nikoli. Má-li sektor číslo 0 nebo 1, tedy patří adresáři a pro získání potřebného místa bude vypuštěn sektor:

1. Neadresářový sektor, který byl ale už volajícím programem přečten nebo zapsán celý (to znamená, že byly přečteny nebo zapsány všechny čtyři subsektory tohoto sektoru, neboť fyzický sektor má délku 512 bajtů 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 znamená zastavení práce hlavního programu, je třeba nějakou část bufferu co nejrychleji uvolnit. Protože všechny sektory v bufferu byly hlavním programem zapsány,

ale na kartridži ešte 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 přebytečný sektor odklídí, má samozřejmě přednost mechanika již běžící.

Pokud je zapotřebí získat prostor pro sektor, který je součástí nějakého souboru, jsou zvažována kritéria v tomto pořadí:

1. Předcházející sektor, z něhož se pouze četlo a je celý přečtený volajícím programem. Je to nejčastější sektor, který v souboru novému sektoru předcházela 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.

Protože BIOS musí obsluhovat mechaniky Microdrive na nejnižší možné úrovni, stará se i o zapínání a vypínání pohonných motorků. Jelikož tato operace trvá zhruba 18 ms, je vyloučeno s motorkem manipulovat po každé operaci čtení nebo zápisu. Motorky se proto zapínají okamžitě, je-li to nutné, avšak jejich vypínání je zpožděné, tak jak to dělají diskové řadiče. Proto Microdrive už tak často neblíkají a neprovádějí záznam a čtení s opětovným zapínáním a vypínáním mechanik jako při práci s originálním ROM, ale většinou vypadá celé čtení nebo zápis tak, že se mechanika rozběhne a po chvíli zastaví. Kromě jiného to také prodlužuje nepříliš velkou životnost mechanik.

Jiří Lamač, autor BIOSu, Daniel Meca

Doplnok prerušovacieho systému mikropočítačov

Ing. Ján Tuška

Článok „prerušovací systém a časovač mikropočítačov“ ing. Šadláka a ing. Chovana, ktorý bol uverejnený v prílohe AR 1988 Mikroelektronika, riešil prerušovací systém u mikropočítačov, ktoré majú jeden prerušovací vstup. Nasledujúci príspevok rozširuje možnosti tohto riešenia z hľadiska programového zabezpečenia navrhovaného prerušovacieho systému.

Rozlišovací program prerušenia, tak ako bol publikovaný v AR, má nasledovné vlastnosti:

— ak je v obslužnom programe prerušenia povolená žiadosť o prerušenie inštrukciou EI, obslužný program sa preruší každou došlou žiadosťou a to bez ohľadu na jej prioritu voči žiadosti, ktorej obslužný program sa práve vykonáva,

— ak sa v obslužnom programe prerušenie nepovolí, ani žiadosť s vyššou prioritou, ako je obsluhovaná, nespôsobí prerušenie.

Za týchto okolností má zmysel hovoriť o prioritnom prerušovacom systéme len v prípade súčasného príchodu viacerých žiadostí o prerušenie, čo užívateľom navrhovaného systému nemusí vyhovovať. Z tohto dôvodu bol navrhnutý rozlišovací program ROZINT s nasledovnými vlastnosťami:

— pri príchode viacerých žiadostí o prerušenie naraz sa začne vykonávať obslužný program žiadosti s najvyššou prioritou a po jeho skončení sa postupne vybavujú ostatné požiadavky podľa ich priority,

— ak je systém v stave obsluhy žiadosti o prerušenie a prerušenie je povolené, pri príchode ďalšej žiadosti o prerušenie sa vykoná nasledovná činnosť:

— v prípade, že priorita príslej žiadosti je vyššia ako priorita žiadosti, ktorá sa obsluhuje, preruší sa obslužný program, vykoná sa obsluha príslej žiadosti a potom sa dokončí obsluha žiadosti, ktorá bola prerušená;

— ak je priorita príslej žiadosti menšia alebo rovná priorite žiadosti, ktorá sa obsluhuje, obsluha sa dokončí a až potom sa vykoná obslužný program príslej žiadosti.

Takto navrhnutý rozlišovací program zabezpečuje podobné vlastnosti prerušovacieho systému ako pri obvodovej realizácii radičom prerušení typu 3214.

(Do programu si prosím doplňte za deviaty riadok (IN PIO4) ďalšie dva riadky: CMA; inverzia logických úrovní a ANI 0000111B; maska využitých prerušení. Pred záverečnou inštrukciou RET pripíšte inštrukciu EI. — omlúva sa autor.)

```

;38H: JMP NAV1 ;SKOK DO ZASOBNIKOVEJ OBLASTI RAM
;NAV1: JMP ROZINT ;SKOK NA ROZLIŠOVACÍ PROGRAM

CSEG

ROZINT: DI ;ZAKAZ PRERUSENI
        PUSH PSW ;ODLOZ REGISTRE
        PUSH B
        PUSH D
        PUSH H
        IN PIO4 ;KOD PRERUSENIA Z IO4. PIO4 = PORT IO4
        MOV C,A

; ZRUSENIE ŽIADOSTI O PRERUSENIE
ZRUINT: MOV B,A
        ANI 00001000B ;TEST CI BOLA ŽIADOST OD INT4
        JZ ZRUIN3 ;SKOK AK NEBOLA
        OUT PIO4 ;ZRUS ŽIADOST OD INT4
ZRUIN0: MOV A,B
        ANI 00001000B ;TEST CI BOLA ŽIADOST OD INT2
        JZ ZRUIN2 ;SKOK AK NEBOLA
        XRA A ;NULUJ AKUMULATOR
        OUT 107/B ;ZRUS ŽIADOST OD INT3
ZRUIN2: MOV A,B
        ANI 00000010B ;TEST CI BOLA ŽIADOST OD INT2
        JZ ZRUIN1 ;SKOK AK NEBOLA
        OUT 101/B ;ZRUS ŽIADOST OD INT2
ZRUIN1: MOV A,B
        ANI 00000001B ;TEST CI BOLA ŽIADOST OD INT1
        JZ POKRAC ;SKOK AK NEBOLA
        OUT 101/A ;ZRUS ŽIADOST OD INT1

; AKTUALIZUJ PAMAT ŽIADOSTI
POKRAC: LDA PAM2A ;PAMAT ŽIADOSTI
        ORA C
        STA PAM2A
        LXI H,PAM2A
        LDA PAM0Z ;STAV OBSLUHY ŽIADOSTI
        CMP M ;POROVNAJ PAM2A A PAM0Z
        JNC DOINT ;DOKONCI OBSLUHU AK PAM0Z>=PAM2A

; ROZLIŠENIE PRÍSLEJ ŽIADOSTI
        MOV A,M ;PAM2A DO AKUMULATORA
        CPI 8 ;ŽIADOST OD INT4
        JNC INT4
        CPI 4 ;ŽIADOST OD INT3
        JNC INT3
        CPI 2 ;ŽIADOST OD INT2
        JNC INT2

; OBSLUHA ŽIADOSTI OD INT1
INT1: ANI 00001110B ;NULUJ BIT 8 V PAM2A
        MOV M,A
        EI
        CALL OINT1 ;OBSLUŽNY PROGRAM PRE INT1
        JMP DOINT ;NAVRAZ DO PRERUSENEHO PROGRAMU

; OBSLUHA PRERUSENIA OD INT4
INT4: ANI 0000111B ;NULUJ BIT 3 V PAM2A
        MOV M,A
        EI
        CALL OINT4 ;OBSLUŽNY PROGRAM PRE INT4
        JMP DOINT ;NAVRAZ DO PRERUSENEHO PROGRAMU

; OBSLUHA PRERUSENIA OD INT3
INT3: ANI 00001011B ;NULUJ BIT 2 V PAM2A
        MOV M,A
        EI
        CALL OINT3 ;OBSLUŽNY PROGRAM PRE INT3
        JMP DOINT ;NAVRAZ DO PRERUSENEHO PROGRAMU

; OBSLUHA PRERUSENIA OD INT2
INT2: ANI 00001101B ;NULUJ BIT 1 V PAM2A
        MOV M,A
        EI
        CALL OINT2 ;OBSLUŽNY PROGRAM PRE INT2
        JMP DOINT ;NAVRAZ DO PRERUSENEHO PROGRAMU

; OBSLUHA PRERUSENIA OD INT1
INT1: ANI 00001100B ;NULUJ BIT 1 V PAM2A
        MOV M,A
        EI
        CALL OINT1 ;OBSLUŽNY PROGRAM PRE INT1
        JMP DOINT ;NAVRAZ DO PRERUSENEHO PROGRAMU

; KONIEC PRERUSENIA
DOINT: POP H
        POP D
        POP B
        POP PSW
        RET

; *****
; DSEG
PAM2A DS 1 ;PAMAT PRÍSLEJCH ŽIADOSTI
PAM0Z DS 1 ;PAMAT OBSLUHOVANÝCH ŽIADOSTI
; *****
END

```