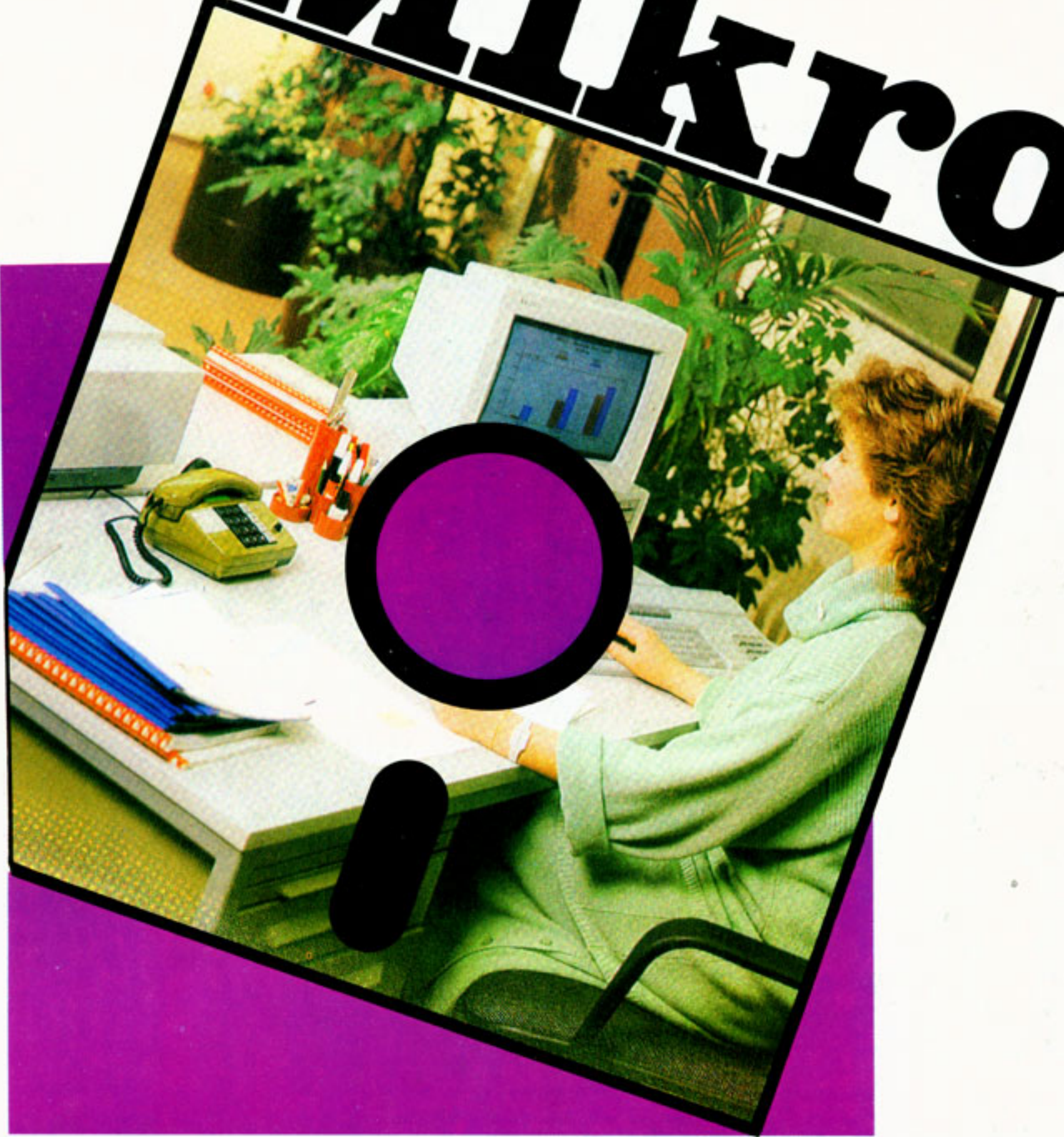


1988 / 6
cena 12Kčs

Mikro



báze

technický
zpravodaj
svazarmu
pro zájemce o
mikropočítače

DTP HEWLETT-PACKARD

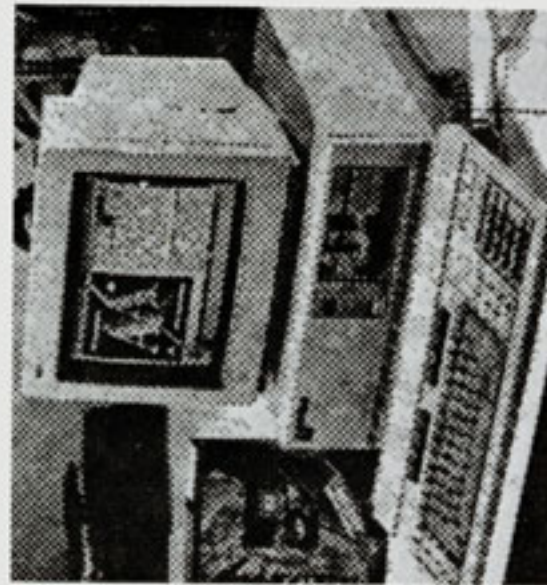
DTP je zkratka pro DeskTop Publishing, což doslova přeloženo znamená stolní publikační činnost. DTP zahrnuje hardware a software pro přípravu a zpracování textů a obrázků a jejich kombinaci při automatizovaném zlomu stránky včetně automatického vkládání mezer a dělení slov na konci řádky. Uživatel má velmi přesnou představu, jak bude tištěná strana vypadat, díky zobrazení formou WYSIWYG - What You See Is What You Get, nebo-li co vidíš, to dostaneš.

HEWLETT-PACKARD nabízí komplexní řešení DTP jak pro západoevropské jazyky, tak pro češtinu a ruštinu včetně klávesnice a zobrazení na displeji a tiskárně v několika typech a velikostech písma (TmsRmn, Helv).

Popis doporučené konfigurace DTP systému:

HARDWARE:

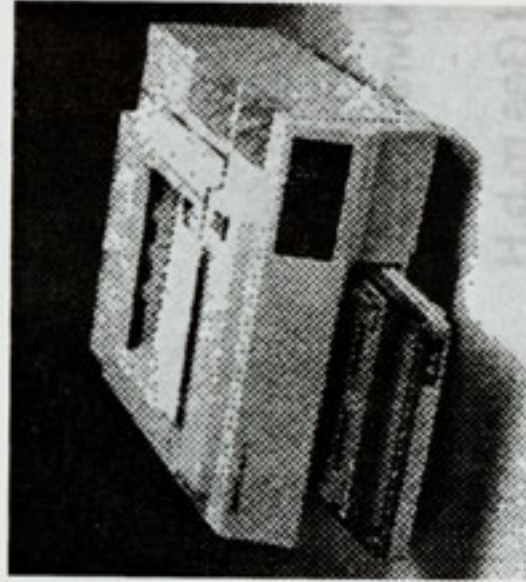
HP Vectra ES: Osobní počítač kompatibilní s IBM PC/AT, procesor 80286, systémové hodiny 8 MHz, matematický koprocessor 80287, 8 MHz, sedm volných I/O slotů, operační paměť 640 kB rozšiřitelná na 8 MB, 8 MHz LIM/EEMS na jedné desce, standardní vybavení interface RS-232C, CENTRONICS a HP-HIL, kontrolery pro všechny typy použitelných disků, Floppy disk 5,25" 1,2 MB se čtením, zapisováním a formátováním předchozích formátů, Floppy disk 3,5" s formáty 720 kB a 1,44 kB IBM PS/2 a 710 kB technických počítačů HP, Winchester 40 MB resp. 20 MB se střední dobou



přístupu 28 ms resp. 65 ms, konfigurovat lze až dva pevné disky, zobrazení VGA, které emuluje všechny předchozí grafické režimy, EGA, Hercules, CGA, a to jak monochromaticky tak barevně.

LaserJet II: Laserová tiskárna druhé generace, základní vybavení paměti 0.5 MB rozšiřitelné na 4,5 MB, jazyk PCL eventuelně Adobe PostScript, rychlost tisku až 8 stran A4/min, zásobník na 200 listů, rozlišitelnost 300 dpi,

dva sloty na font cartridge, download pro softwarové fonty, v současnosti více než 200 různých typů a velikostí písma, interface RS232C a CENTRONICS.



ScanJet: Snímač faksimilií, rozlišitelnost programově volitelná 38-600 dpi, měřítko 13% - 200%, rychlost strana A4/300 dpi za cca 20 s včetně uložení dat na disk, 16 úrovní šedi, výkonný software pro ovládání scanneru a grafický editor pro korekci nasnímaných obrázků, interface obousměrný CENTRONICS.

PageMaker.

SOFTWARE:

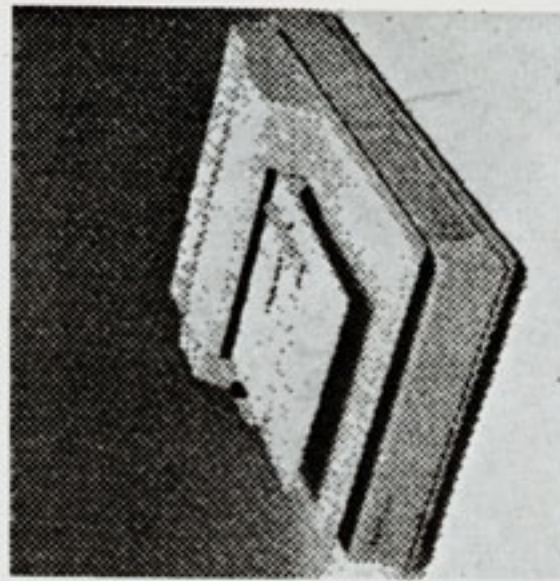
MS-DOS 3.2: Operační systém.
MS-Windows: Grafické prostředí zahrnující textový a grafický editor MS-Write a MS-Paint, oba pracující s českými fonty.

PageMaker 3.0: Program pro automatizovaný zlom strany včetně porčního vkládání mezer a dělení slov na konci řádky pracující s českými

fonty. Verze 3.0 rozšiřuje původní verzi o dalších 35 funkcí jako např. komplexní zlom celé publikace, práce v barvě s využitím tiskárny HP PainJet, definice makroinstrukcí pro rychlejší formátování textu, zalamování textu kolem nepravidelných obrázků, možnost stranově obráceného tisku aj.

Fontware: Programy pro tvorbu a instalaci různých typů písma.

Scanning Gallery: Program pro ovládání snímače faksimilií zabezpečující snímání obrázků, převod do datových souborů, editaci a tisk.



ReadRight: Program pro převod textů v OCR písma sejmutých na snímáči faksimilií jako grafická předloha do textových editorů.

Recogniter: Obdobný program jako ReadRight s možností učení, který rovněž umožní práci s českou diakritikou.

Hewlett-Packard DeskTop Publishig systém nabízí efektivní řešení jak v oblasti malonákladových publikací, jako jsou tiskoviny a dokumentace pro podniky a instituce, ale může být i cestou ke zefektivnění prací v oblasti fotosazby.



O B S A H

ZX nostalgie	1
Hovory o programování	2
Scanner	4
HP Deskjet	5
Dialog Mac - Vax	5
Trh pamětí CD ROM do r. 1990	5
Další optomagnetická paměť	5
CD ROM po dvou letech	6
Z88 Computer	7
Kopírování obrazovky	8
Stabilizátor pro ZX Spectrum	10
Moduly pro ZX Spectrum	11
Laserový tisk bez laseru	16
Profesionální klávesnice a V.24 pro µB-Pascal	17
Hrátky se zásobníkem Z80	20
Soubor pomocných programů k modulu BASIC 6 mikropočítače IQ 151 ...	23
Středisko VTI pro elektroniku	27
mc - Die Mikrocomputer-Zeitschrift	31
Programová nabídka Mikrobáze	32

Technický zpravodaj Svazarmu pro zájemce o mikropočítače. Vydává 602. ZO Svazarmu ve spolupráci s redakcí časopisu Amatérské radio. Povoleno ÚVTEI pod evidenčním číslem 87 007. Zodpovědný redaktor Ing. J. Klbal, sestavil ing. A. Myslík. Redakční rada: P. Horský, ing. J. Klbal, ing. P. Kratochvíl, J. Kroupa, ing. A. Myslík, ing. J. Truxa. Ročně vyjde 10 čísel, cena výtisku 12 Kčs podle ČCÚ a SCÚ č. 1030/202/86. Roční předplatné 120 Kčs. Objednávky přijímá a zpravodaj rozšifruje 602. ZO Svazarmu, Wintrova 8, 160 41 Praha 6.



602.ZO

&

ZX NOSTALGIE

Ťukám do svého ZX Spectra, abych si postěžoval. Protože mě přestávají znát, už se ke mně vlastně vůbec nehlásí. Kdo? Ti, co si na stůl místo Specter postavili obludy zvané IBM PC a jim podobné.

Vzpomínám, jak jsem jako malý kluk pobýval v jedné balkánské zemi. Tenkrát tam znakem vyššího společenského postavení byl deštník. Normální černý deštník. A kdo ho měl, nosil ho i ve sluneční výhni (stažený). A když přšelo, kráčel pod roztaženým deštníkem s výrazem vyzařujícím převahu nad těmi, co jim lilo za krk. Člověk s deštníkem se stýkal většinou jen s lidmi, kteří v ruce třímali deštník. Když spolu mluvili, z jejich vznešené gestikulace bylo zřejmo, že řeší velmi závažné otázky bytí a nebytí všeho tvorstva. Ti bez deštníku kolem se v porovnání s nimi chovali ... no prostě úplně nemožně.

Když jsem se pak vrátil do Prahy, byl králem naší třídy ten, kdo měl žvejkačku. Podobně jako ti muslimové od svých deštníků, také on se od své žvejkačky neodloučil ani na moment. Prožvykal celé vyučování. Když si na to vzpomenu ... jak, chudák, musel tenkrát trpět! My, co neměli k zahraničním žvejkačkám přístup, jsme se vycpávali mejdlíčkem a pendreký. Jenže stačilo, aby mezi nás přišel ten, co přístup měl, a táááááhle jí natáhl z pusy, až kam rukou dosáhl. Pak se zaklonil, žvejkačku nasoukal jazykem zpátky a beze slova odcházel. My stáli jako přimražení a ve zbožné úctě civěli za ním...

Co to bylo pak? Tranzistorák, samozřejmě zahraniční. Největším výrazem shovívavé dobrotivosti jeho majitele bylo, když někoho nechal trošku zakroutit knoflíkem. Pak těch symbolů společenské převahy a vznešenosti rychle přibývalo. Auto - sice tak akorát do šrotu, ale stálo před domem jako rytířský kůň. Černobílá talevize - kdo jí měl, nemusel zdravít první. Magnetofon - jeho majitel byl ctěným hostem radovánek a všude se zadarmo najedl a napil.

A jsem v době skoro současné. Žvejkačka už dávno nikoho neničí a auto je krám, který všude překáží a smrdí. Pokrok nelze zastavit. Ale s ním přicházejí nové symboly obecné vážnosti. Po nesehnatelné barevné televizi (ministr zrušeného FMEP: "Barevných televizorů je na našem trhu dostatek...") pole postupně ovládá ZX-81, video, ZX Spectrum a satelitní příjem televize.

Z toho všeho mám jen to Spectrum. A cítím, jak se moje spoluúčast na hrotu vážnosti rozkládá na kousky. Ještě nedávno se telefon nezastavil: "Ahoj, nevíš, jakou rutinou romky se maluje bod, nemáš program ten a ten, já právě dostal ty a ony, to je jízda, to víš, že ti je dám stočit...". Dnes telefon zarytě mlčí. Jsem mimo síť hovorů jako: "Nemáš emesdos tři? Stáh bys mi ho z disku na flopáč? Nechceš český fonty do čirajtru? Konečně jsem sehnal egu, to je hajres jako víno!" Už mě prostě neznají. Nemáme si o čem povídat. Jsem odpadlík, počítačová spodina, povalující se ve zvadlém spektřišti...

Chvilími propadám filosofování bédných o tom, jak se lidstvo žene do zkázy, jak ničí svou sociabilitu, depersonifikuje se, blbne ... A pak zas volám: "Jděte se s těma vašima deštníkama vycpat, ajbímáci nepromokaví!" A když se mi nakonec k ránu podaří usnout, v hlavě mi tancují ty nejkrásnější konfigurace péceček s laserovými tiskárnami.

A je mi jasné, že za nějakou, zatím přesněji neurčitelnou dobu budu mít tu příšeru na stole taky. A už slyším, jak mi zase drnčí telefon ... a v něm se mě někdo ptá, jak připojit joystick ke Spectru. A já odpovídám: "Prosím vás! Že se tou trapnou krabičkou ještě vůbec zaobíráte! Já mám teď IBM PC-XT-AT-E.T., víte? A teď na něm řeším loulevl formátíng hárddisku s pártyšnem a vy na mě s takovouhle prkotinkou! Zeptejte se u vás v Domě pionýrů, tam to budou všichni vědět..."

A nebo ne! Já mu to řeknu. Protože kdybychom měli být všichni takhle praštění, tak bych si radši s orientální lhostejností lehl pod deštník a dělal bubliny žvejkačkou.

-elzet-

Ing. Rudolfa Pecinovského, CSc. není nutno představovat. Pokud snad neznáte jeho práci pro Svazarm nebo pro SSM, pak jste ho jistě slyšeli a viděli v některých televizních pořadech věnovaných výpočetní technice. Uvítal jsem ho hřejivou otázkou:

Co tě dnes kolem počítačů pálí nejvíc?

Pálí mě dvě věci, a to nejen dnes, ale pořád. Je to metodika výuky programování a koncepčnost vlastního programování. Obojí spolu úzce souvisí. Jde v podstatě o to, aby tvorba softwaru nebyla bastlením. V praxi se svoje nápady, názory a zkušenosti snažím prosazovat na různých seminářích, v kroužcích, psaním článků do časopisů, sborníků... Vlastně je to taková osvětová činnost.

HOVORY O PROGRAMOVÁNÍ

K programování jsem se dostal na vysoké škole před více než deseti lety. Metodika mého profesora byla trysková. Položil přede mne učebnici Fortranu od Vogela a řekl: "Naučte se to až sem před procedury a přijďte za tři týdny." Přišel jsem, on mi dal úkol vytvořit program na převod mezi arabskými a římskými čísly, zavedl mne ke stroji s děrnými štítky, ukázal, kde se co mačká a dodal: "Až to budete mít, hodte mi štítky do kanceláře. A doučte se zbytek učebnice." Začal jsem vymýšlet svůj první program. Sem tam jsem se profesora ještě na něco zeptal. Opět se mi dostalo tryskem výkladu, že jsem sotva stačil vnímat, co říká. Když jsem měl program hotov, odevzdal jsem ho v podobě děrných štítků. Profesor byl značně překvapen, že mi to z nich chodilo napoprvé. Mě zase překvapilo jeho překvapení. Říkal jsem si: "Když napíšu program, tak přece musí fungovat, ne? Tak jaképak divení?"

Ale brzy jsem sám sebe z toho vyvedl. Profesor mi dal vyřešit problém Hanojských věží. Proháněl jsem je počítačem snad sedmnáckrát, než to začalo být ono. Je to zvláštní, ale jako by obecně fungovalo nějaké pravidlo prvního doteku. Když třeba někdo hraje minigolf poprvé v životě, může porazit i dobrého hráče. Pak už ne. Podobně jsem na tom byl při svém prvním programu. Až později mi došlo, že ne ten první, ale ten druhý případ, kdy mě týraly Hanojské věže, je ten normální.

Z jaderné fakulty jsem přešel na elektro a začal se programování věnovat hlouběji. Diplomku jsem psal ve Fortranu - program měl asi 3000 řádek, takže kvantitativně už se dalo mluvit o programování. Dál jsem měl to štěstí, že vojenská správa vyhověla mé prosbě, abych mohl na vojně učit programování. Tak jsem nevyšel ze cvíku a neodnaučil se učit. Po návratu z vojny jsem přešel na jazyk PL/1, který je vynikající, ale jen pro profesionály. Má obrovské možnosti, ale když v něm neděláš denně, do příštího programování toho z rozdělaného programu polovinu zapomeneš. Asi jako když děláš v Céčku nebo assembleru. PL/1 je hodně "domýšlivý" jazyk. Ve Fortranu nemusíš v některých detailech úplně přesně dodržovat syntaxi, a přesto překladač pozná, co jsi měl na mysli - třeba zda jde o integer nebo real. PL/1 v tom "domýšlení" jde ještě dál. Když sem tam něco v zápisu opome- neš, překladač to dožene. Někdy si ale může něco domyslet tak, jak to vůbec nechceš. Tak jsem byl poprvé donucen všechno velmi přesně deklarovat. Vlastně to, co je v Pascalu povinností, v PL/1 děláš raději dobrovolně, aby ti program nevykolejil.

Po aspirantuře jsem šel do skupiny Edy Smutného, kde jsem konečně dostal k programování s nádechem profesionality. To, co jsem dělal do té doby, bylo převádění vědeckotechnických výpočtů do

počítačového programu. A to není programování v tom pravém smyslu.

Jen jiná symbolika téhož?

Skoro tak. Na druhou stranu jsem se ale naučil něco, co se mi v praxi moc hodí. Mě totiž vůbec nezajímalo, jestli bude výpočet trvat minutu nebo dvě. Podstatné bylo mít program rychle hotov. Čili najít algoritmus a nějak ho zakódovat. Takové ty finty - třeba jak udělat, aby nějaká smyčka běžela rychleji - byly nezajímavé, v podstatě k ničemu. Nebyl jsem tísněn tím, čím si, podle mého názoru, zbytečně láme hlavu řada lidí - jak program co nejvíc optimalizovat, ještě o trochu zkrátit, ještě malinko zrychlit. Třeba zrychlovat smyčku, která čeká, až se stiskne tlačítko na klávesnici, je



slabomyslnost. A přesto se tím někteří lidé zabývají. Nebo ušetří tři bajty, přičemž jim v paměti zůstanou čtyři kila neobsazená.

PL/1 je dost marnotratný jazyk. Při jeho používání se naučíš jistě velkorysosti. U vědeckotechnických výpočtů je dost jedno, jestli ti program zabere sto nebo dvě stě kil paměti a jestli vyhodí výsledek hned nebo až za chvíli. Tam je důležité zrychlit metodu o řád nebo i o dva. Aby to, co se dříve počítalo celou noc, bylo hotovo za pár minut. Myslím, že ta určitá velkorysost marnotratností je při programování nutná. Když se optimalizace ukáže nezbytná, můžeš se jí věnovat po dokončení programu.

Jak jsi získával zkušenosti v osvětové práci?

V SSM jsem měl různé funkce. Ale začal mi tam vadit zmatek, pramenící z amatérismu v organizátorské práci. Třeba se domluví, že ve čtvrtek v osm hodin mám někde být se dvaceti dětmi. A večer ve středu ti zavolají, že máš přivést třicet dětí, ale ne v osm, ale v sedm, a to úplně někam jinam. A když se jich zeptáš, jak to máš takhle nahonem přeorganizovat, řeknou ti: "Jste pionýrský pracovník, tak buďte operativní!" Chtěl jsem s dětmi dělat dál, ale ne takovýmhle hurá-stylem. Po vojně jsem se dozvěděl o "hášovcích" (Městská stanice mladých techniků). Zrovna tam dostali dnes už historickou krabičku ZX-80 a neměli nikoho, kdo by na tom učil děti programovat. Už na vysoké škole jsem učil BASIC a PL/1, takže jsem o metodice měl určité představy, které jsem chtěl dál rozvíjet. Se ZX-80 jsem děti učil BASIC, ale brzy jsem pochopil, že je to slepá ulička. Tudy se děti nemůžou dostat na žádnou slušnější úroveň.

Důvod?

BASIC je na první pohled jednoduchý, dá se rychle vstřebat. Děti jdou rychle nahoru, dostanou se na nějakou úroveň, ale na ní se zaseknou. A ten BASIC je už nepustí dál, protože je koncepčně nešikovně postavený. Mluvím samozřejmě o BASICu v té klasické podobě, jako je třeba ve Spectru, PMD a podobně. Třeba Beta BASIC je už v trochu jiné kategorii. Ten klasický nemá procedury volané jménem, procedury s parametry, lokální proměnné. Když je jazyk vybaven prostředky strukturovaného programování, pak se v něm dá přemýšlet a dělat rozumně něco rozumného. Bez toho je všechno jen takové bastlení, hraní si na programování.

Po těch dvou basicových letech jsem to zkusil s vývojovými diagramy. Ale ani to nebylo optimální.

Až se objevil Petr Novák, který přišel s tím, že na elektro se učí zavírat "vývojáky" do takových chlívčeků, které mají jeden vstup a jeden výstup. Uvnitř nich se to pak zase "navývojuje". Když jsme nad tím bloumali, řekli jsme si, že ty spojovací "vývojákové" čáry jsou tam vlastně dost zbytečné, protože to jinudy stejně nemůže jít. Tak nám z toho vypadly jednotlivé bloky struktury. Nato se objevil MUDr. Jiří Kofránek, který řekl, že mu to připomíná jeho pokusy, jak lékařům srozumitelně popsat funkce programů, které byly součástí jeho kandidátské práce. Dali jsme to všechno dohromady a nakonec z toho vzešlo to, co jsme nazvali kopenogramy...

-gramy s monogramy Kofránek, Pěcinovský, Novák.

Nejdřív jsme tomu říkali struktogramy, jenže tenhle název už pro něco obdobného existoval. Tak nás napadlo to pojmenovat takhle a nakonec se to vžilo. Při vlastním programování se mi postupně prokázalo, že to, co kopenogramy hlásáme, je větší pravda, než jsme si zpočátku uvědomovali. Třeba i pro zpětnou dokumentaci hotového programu, což je z principu otravná činnost, nikdo ji nedělá rád.

Největší síla kopenogramů je v první fázi tvorby programu. Z předem zpracovaných kopenogramů budoucího programu pak všechno už jen pohodlně přepisují do zdrojového textu. Věř-nevěř, takhle mi produktivita práce vzroste o sto procent.

Já jsem dost zmatkář a nedokážu v hlavě udržet představu o celém programu. Jsou jedinci - třeba šachisté hrající simultánky - kteří mají hlavu tak vycvičenou, že si pamatují všechno, co potřebují. Považují se za relativně normálního. Ale pro vzrůst produktivity se bez téhle grafické berličky neobejdou. Současně s kvantitou mi vzroste i kvalita programu, který získá na přehlednosti. Člověk pak nedělá tolik kopanců, všechno je mnohem čistší.

Názory na to, jak postupovat při tvorbě programu, se různí. Klasický je příkaz jít strukturou shora dolů. Někdo ale začíná "z prostředka", jiný přebíhá sem tam. Někdo si předem všechno do detailu rozplánuje, jiný vyzývá počítač na lýtý souboj a usedá k němu s mlhavou konturou programu v hlavě.

Jak postupuješ ty?

Ocituji výrok jednoho kolegy: "Čím víc přemýšlím o programu, tím dřív jsem s ním hotov." Plně s ním souhlasím. Podle okolností volím jeden ze dvou postupů.

První je hurá-styl. Sednu a píšu. To je ale možné jen u krátkých programů, tak do dvouset řádek, kde nemám potíže s udržením koncepce v hlavě. Takhle jsem za den napsal program pro převod datových souborů mezi "pécéčkem" a CP/M. Ale i v takovémto případě musím mít předem jasno, co chci, a alespoň základní koncepční rysy a nejdůležitější datové struktury si načrtnout dopředu. Když někdo nemá jasnou koncepci, pak buď brzy zkrachuje, nebo je tím jeho výtvar nějak poznamenán. Není myslitelné vynalézat nějaké zásadní věci uprostřed programování. Pokud to není začátek konce, pak určitě velkých potíží.

Větší programy hurá-stylem dělat nejdu. Širší předběžná analýza je nutná. Nemusí jít do úplného detailu, ale musí obsáhnout všechny podstatné funkce a jejich vztahy. O správnosti všech zásad moderního programování, počínaje důležitostí předběžné analýzy, přes zodpovědný návrh datových struktur, konče modulární strukturou celého programu jsem se přesvědčil právě minulý týden. Přinutil jsem se striktně dodržovat všechny zásady a dokázal jsem asi 1800 řádek dlouhý program v Pascalu napsat za necelých 10 dnů. V životě bych předtím nevěřil, že pouhé dodržování těchto zásad povede k tak velkému zvýšení produktivity.

Jiný příklad - dělal jsem program pro řízení robota ve Forthu. Na začátku jsem musel zhruba rozvrhnout hierarchii příkazů, jednotlivé hladiny, ve kterých se bude všechno pohybovat.

Nedávno jsem se účastnil jednoho korespondenčního kursu ČSVTS. Byl nazván Architektura programového vybavení a řízení jeho tvorby. Dostal jsem skripta, jejichž autor, ing. Jiří Voříšek, se zabývá vrstvenou architekturou. Víceméně říká věci, které člověk tuší, ale on je dokázal formulovat.

Ty vrstvy jsou velmi rozumné, protože do nich se celý problém láme.

U toho robota je vrstva komunikace s uživatelem, vrstva převodu na přímé příkazy robotu a nejnižší vrstva, která příkazy převádí do sledů pulsů pro motor. Vedle toho se ještě udělala boční analýza pro nalezení nejlepší metody rozběhu a brzdění krokového motoru. To byla spíš matematicko fyzikální záležitost, ale nicméně z toho vzešel algoritmus. S programem jsem začal na jeho dně, napřed jsem si vyzkoušel řízení motoru.

U Forthu se vůbec špatně dělá shora dolů. Je to jazyk, který tě dost nutí dělat zdola nahoru. Ale přitom ti poskytuje všechny prostředky k tomu, aby se ti to dělalo dobře.

Obecně můžu říct, že komunikaci s hardwarem v té nejnižší vrstvě dělám vždycky napřed. Nepovažuji se za hardwaráře. Tak si nejdříve ověřuji, jestli jsem z dokumentace dobře pochopil funkce hardwaru a možnosti jeho ovládní softwarem. Tahle vrstva se obvykle promítá do těch vyšších. Pokud v tom nemáš jasno, dostaneš se do problémů. Když se mi vyjasní, udělám třeba jednu celou větev shora, spustím se po ní až dolů. A pak už nabalují další kousky programu. Takhle získám přehled o tom, co vlastně dělám a jak se věci mají.

Proč jsi robota oživoval Forthem?

Tenhle jazyk používám pro regulační software, protože se tam potřebuji pořádně ptát, nějak konverzovat. Pro práci se soubory volím Pascal (i proto, že jsem trochu líný je do Forthu dodělávat). V assembleru dělám pouze ze zoufalství, když okolnosti nedovolují použít vyšší jazyk. Nejradši mám Céčko, protože zahrnuje nejširší oblast aplikací. To je zatím poslední jazyk, do kterého jsem se pustil. Při programování v Céčku na "pécéčku" používám Turbo C. To je perfektní program, programátorovi poskytuje velký komfort. Překlad chvilku trvá, ale když děláš menší kousky, jde to svižně. Céčko je typický jazyk pro profíky. Každému, kdo neprogramuje denně, ho rozmlouvám. Je to jazyk stejně nebezpečný jako assembler. Když pořádně nevíš, co si smíš a nesmíš dovolit, a k počítači se dostaneš jednou týdně, určitě tě nachytá na švestkách, provede ti nějakou nepřičetnost, kterou pak několik dní hledáš.

Chodíš na koně do Chuchle?

Ne. Proč?

Vedení závodiště se v rámci boje proti černým sázkařům rozhodlo zavést sázkové kursy řízené počítačem. Program je pro osm pokladen, teď jich mají instalováno pět. Když pokladna přijme sázku, pošle po drátě informaci, že na koně toho a toho bylo vsazeno tolik a tolik. Centrální počítač to zaregistruje, a jak se přijímají další a další sázky, v jejich průběhu mění kursy jednotlivých koňů. Podstata je taková, že kůň, na kterého se sází nejvíc, musí mít nejnižší kurs, abys výhru favorita mohl pokrýt ze sázek na ostatní koně. Ti pak mohou mít vyšší kurs, protože sázek na ně bylo méně. Kdyby vyhrál nějaký outsider, máš předem zajištěno, že výplaty výher pokryješ penězi z velkého počtu sázek na favority. Proměnlivost kursů v průběhu příjmu sázek lidí chytla, protože nezáleží jen na tom, že vůbec vsadíš, ale i kdy a s čím k pokladně přijdeš.

Aby vše probíhalo plynule, projekce informace na monitorech nesmí zdržovat příjem sázek. Proto se přepisují jen políčka, ve kterých se něco změnilo, nikoli celá obrazovka. Do políček se zapisuje přímo, bez nějakých zvláštních kontrol. Pro název závodu je vyhrazena jedna řádka. Jednou probíhal závod s příliš dlouhým názvem, který pronikl do druhé řádky. Tak se soupis koňů posunul o řádku níž, ale kursy se vypisovaly na svých pozicích. A nastal průšvih. Favorit měl u sebe kurs 1:17 a lidé málem povalili pokladnu. Pokladní však měla nezakreslenou informaci a říkala lidem, že ten kůň má kurs 1:2. Když se to lidem vysvětlilo, přijali to, a nic zvlášť nepřijemného z toho nebylo. Tady vidíš, že vždycky může nastat překvapivá situace, kterou v programu nemáš ošetřenou, protože tě možnost jejího vzniku nenapadla. Když jsem program dělal, napřed jsem ladil algoritmy pro řízení kursů. První dva mi vyhořely. Až třetí, úplně nejed-

nodušší, se chytil. Pak jsem pokračoval na komunikaci s uživatelem. S ním se musíš nějak dohodnout, jak to chce, protože od komunikace, coby vršku programu, se odvíjí celý jeho spodek.

Můžeš říct, kolik jsi za ten program dostal?

Zprostředkovatelem objednávky byl MON. Vývoj šel ve dvou etapách. Celý program má 80K, takže si aktuální část musí načítat z hard disku. Odměna byla dvakrát 7000 korun. Třeba se to někomu může zdát hodně. Ale není to tak. Takové objednávce mohou věnovat dvě hodinky volna denně. Je to asi jako kdybych psal články, kterými se tady zbohatnout nedá. Jiná situace by mohla nastat, kdybych byl na volné noze. Jsou taky lidé, kteří nedělají programy přes velkého zprostředkovatele, ale prodávají je přímo. A výsledek prodávají na několika místech. Zvláště u databázových aplikací mohou dosáhnout dost slušných výdělků.

(Dokončení příště)

Rozmlouval -elzet-

SCANNER

Po klávesnici, světelném peru, grafických tabletech, myši a dalších prostředcích pro digitalizaci obrazu se více rozšiřuje další druh periferního zařízení pro osobní počítače: numerický obrazový převodník neboli scanner. Scanner umožňuje digitalizovat grafickou předlohu plošného charakteru, např. list papíru, stránku knihy, fotografií atd.

Základem scanneru je lineární prvek CCD s minimálně 2 000 fotocitlivými buňkami. Protože plošné prvky CCD ještě nejsou běžně k dispozici, současné scannery používají přesné mechanické zařízení, které pohybuje vozíkem s lineárním snímačem CCD s přesnou optikou. Optika se skládá minimálně z parabolického zrcadla a soustavy čoček s filtrem. Ceny i těch nejjednodušších scannerů se pohybují v blízkosti cen osobních počítačů - 2 až 3 tisíce dolarů. Citlivým místem scannerů je počet buněk, vytvářejících jeden řádek. Rozlišovací schopnost musí být alespoň 12 bodů na milimetr, tj. asi 300 bodů na palec (dot per inch - dpi). Pro současné kvalitní laserové tiskárny a programové vybavení pro zpracování obrazu a textu a rozpoznávání písma je však tato rozlišovací schopnost na hranici použitelnosti. Výstup ze scanneru nelze použít k větší reprografii, i když jsou už ohlášeny modely s 500 dpi.

Další Achillovou patou scannerů je fakt, že jejich výstup nelze přímo přivést do počítače. Výstupní signál z prvku CCD se musí "vyhladit" (vyrovnat kontrast nebo extrapolovat úrovně šedi, obvykle 64). Snímaná data se musí demultiplexovat, převést do sériové podoby a zkomprimovat pro přenos velkou rychlostí v asynchronních blocích. Nakonec je nutné data reorganizovat do matice bodů pro další zpracování počítačem. Komunikace scanneru s osobním počítačem probíhá prostřednictvím specializované desky (karty). Scanner tedy nelze připojit tak snadno jako např. modem nebo tiskárnu (standardní periférie).

Jedním z hlavních kritérií použitelnosti je rychlost snímání a přenosu a kvalita získaného obrazu. Doba zpracování jednoho obrázku formátu A4 se pohybuje okolo 1 minuty. Tuto dobu lze u některých scannerů zkrátit asi na jednu třetinu zvýšením rychlosti řádkování, ovšem na úkor kvality obrazu.

Nejprodávanějšími modely v Evropě jsou ScanJet firmy Hewlett-Packard, IS 30 M2 firmy Ricoh, IX 12 firmy Canon a S200 firmy Agfa. Nejlevnějším typem je tzv. Handy Scanner, u kterého je zcela vypuštěna řádkovací mechanika a kvalita obrazu závisí z valné části na neochvějnosti ruky uživatele, který přejíždí tímto zařízením podobně jako např. čtečkou čárkového kódu nad předlohou ručně.

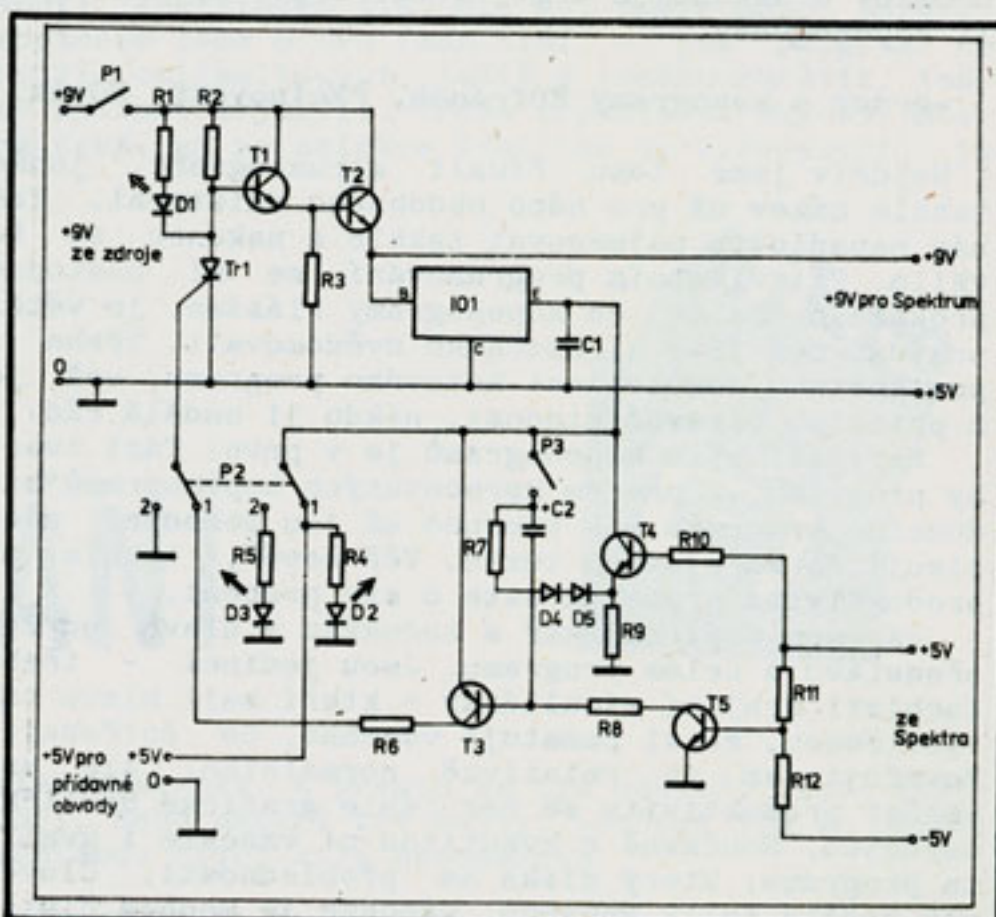
Literatura:

Mariate, A.: Six scanneurs. Ordinateur individuel, No 100, 1988.

Ochrana napájení pro mikropočítač ZX-Spectrum

U prvních mikropočítačů Sinclair ZX-Spectrum docházelo k poruchám zdroje napětí -5 V. Při výpadku tohoto zdroje obvykle dojde ke zničení dynamických pamětí. Při připojování přídatných obvodů (např. MHB 8255) na sběrnici mikropočítače je nežadoucí přítomnost napětí na sběrnici mikropočítače v době, kdy počítač není napájen.

Obvod, který slouží k napájení jednoduchých přídatných obvodů a k ochraně mikropočítače ve výše uvedených případech, je uveden na obr. 1.



Obr. 1. Ochrana napájení mikropočítače ZX-Spectrum

Logiku, vyhodnocující výpadek napětí -5 V nebo přítomnost napětí +5 V pro napájení přídatných obvodů bez napájení mikropočítače, tvoří tranzistory T3, T4, T5. K vypnutí napájení dojde při nárůstu napětí z -5 V na přibližně -4 V a při poklesu napájecího napětí ve Spectru z +5 V na asi +4,5 V (rozdíl určují diody D4, D5).

Při vyhodnocení poruchy sepne tyristor Tr1 a přes tranzistory T1 a T2 vypne napětí pro mikropočítač a přídatné obvody. Tento stav signalizuje červená LED D1. Normální stav signalizuje zelená LED D2.

Ochranu je možné vypnout přepnutím přepínače P2 do polohy 2, což při zapnutém napájení žlutá LED D3 signalizuje. Napájení se zapíná spínačem P1.

Protože napětí nenabíhají ve správném sledu (což je dáno konstrukcí zdrojů v mikropočítači), je nutné při zapnutí napájení po krátký čas blokovat funkci ochrany. Pro jednoduchost jsem použil mikrospínač P3, který připojuje kondenzátor C2 mezi bázi a emitor tranzistoru T3. Kondenzátor přibližně 0,2 s blokuje funkci ochrany. Mikrospínač P3 je třeba stisknout při zapínání zdroje.

Jestliže je mikrospínač sepnut i dále, ochrana sice pracuje, ale reaguje až po 0,2 s. Přídatné obvody jsou napájeny z monolitického stabilizátoru MA7805, který též limituje jejich největší proudový odběr (při větších odběrech by navíc došlo k nežádoucímu poklesu napětí transformátoru pod 9 V).

Seznam použitých součástek

Rezistory	Kondenzátory
R6 100	C1 0,1 uF
R4, R5 220	C2 50 uF/6V
R1 560	
R3 1,2 k	Polovodiče
R9 2,4 k	T1, T4, T5 KC508
R8 2,7 k	T2 KU611
R10 12 k	T3 KF517
R11, R12 36 k	Tr1 KT505
R7 100 k	D1, D2, D3 LED
R2 50 /2W	D4, D5 KA501
	IO1 MA7805

HP Deskjet

Laserové tiskárny dosud porážejí všechny své konkurenty ve všech podstatných parametrech. Firma Hewlett-Packard přichází na trh s inkjetovou tiskárnou, která se laserovým vyrovná bodovou hustotou tisku. Tisková hlava HP Deskjetu, která není větší než devítijehličková hlava úderníkových tiskáren, obsahuje 56 trysek. Maticovou síť tvoří 30 x 50 bodů, což je přibližně "laserových" 300 bodů/palec. Kvalita tisku HP Deskjetu je od laserového nerozeznatelná. Oproti toneru laserových tiskáren, které jej na papíru tepelně utvrzují a papír prakticky vůbec nedeformují, bývá inkoust inkjetových tiskáren příčinou kroucení některých druhů papíru. Nevýhoda nemožnosti současného tisku kopíí přes uhlový papír je společná oběma typům tiskáren.

HP Deskjet má lehce výměnnou hlavu se zásobníkem inkoustu. Celý komplet stojí 16 Lstg. Inkoust zásobníku vystačí na tisk cca 1,2 miliónu znaků v draft módu nebo 0,5 miliónu znaků NLQ. Tisk je pochopitelně stále stejně kontrastní. Přímou součástí tiskárny je zásobník papíru pro 100 listů, které jsou odebírány automaticky. Přímou nad ním je plošina, pro ukládání potisknutých listů. Tak HP Deskjet zabírá minimum místa oproti tiskárnám, kterým zásobníky nehezky trčí do stran jako křídla rackovi před vzletnutím a ubírají užitek plochu desky stolu. HP Deskjet nemá mechaniku pro tisk na perforovaný papír, což nelze považovat za nedostatek. Podobné tiskárny se používají především pro firemní korespondenci apod., kde se perforovaný papír stejně nepoužívá. HP Deskjet může při ručním vkládání tisknout i na menší formáty papíru než A4.

Výrobce udává maximální rychlost tisku 240 zn/s v draft a 120 zn/s v NLQ módu. Tiskárna obsahuje ROM s bohatým výběrem různých typů znaků. Jejich rejstřík lze podstatně rozšířit přídatnými paměťmi, které jsou na trhu v ceně 66 Lstg za kus.

Řídicí kódy tiskárny jsou o dost komplikovanější, a tak se liší od těch, jimiž se řídí chod většiny jehličkových tiskáren. To s sebou nese určitou nevýhodu pro využití HP Deskjetu při práci s některými programy, s nimiž si bez provedení úprav nebude rozumět buď vůbec nebo jen částečně. Firma pro takovou eventualitu nabízí další paměť ROM, která převádí řídicí kódy standardu Epson na HP Deskjet. To ale omezuje využití celého rejstříku schopností tiskárny. Na trhu je i disketa (5,25"), která obsahuje drivery přízpusobené komunikaci se standardem IBM PC. Tak může prakticky všechen nejpopulárnější software tohoto standardu s tiskárnou plně spolupracovat. Vlastní řízení HP Deskjetu je velmi blízké ovládání laserové tiskárny HP LaserJet téhož výrobce.

Kromě již uvedených předností je tisk tichý. Jiné tiskárny s podobným podáním tisku a jeho rychlostí jsou výrazně dražší. HP Deskjet stojí 850 Lstg.

Podle What Micro? 6/88

-elzet-

Dialog Mac - Vax

U příležitosti výstavy Mac-World EXPO v San Francisku uzavřely firmy Apple a DEC dohodu o společném vývoji výrobků. Byl tak započat dialog jednoho z největších výrobců mikropočítačů s jedním z největších výrobců minipočítačů v USA. Tato strategická aliance možná rozvíří v nejbližších letech všechny sféry informatiky.

President firmy Apple John Sculley prohlásil, že spojení technologií Apple a Digital Equipment je jednou z nejsilnějších zbraní pro dobytí počítačového trhu. Z této dohody budou těžit hlavně poloprofesionální a profesionální sféry, jako jsou např. univerzity a velké společnosti. Podle různých analýz používá velké procento společností zároveň systémy Macintosh a VAX. Více než 36 % minipočítačů VAX používá totiž Macintoshe jako terminály a jejich uživatelé hledají způsoby, jak by tyto dva mocné elektronické světy mohly komuniko-

vat co nejlépe. Jejich problémy by měla vyřešit právě dohoda mezi Apple a DEC. V srpnu 1988 budou publikovány technické specifikace budoucích společných výrobků. John Sculley prohlásil, že ačkoliv jsou filozofie Apple a DEC velmi blízké, bude trvat několik let, než obě společnosti nabydou plné vzájemné důvěry. Uživatelé Macintoshe budou mít přístup k souborům VAX prostřednictvím protokolu AppleTalk Filling (AFP). Dále budou mít možnost prostřednictvím počítače VAX se účastnit elektronických konferencí. Zároveň bude mít každý uživatel transparentní přístup k sítím AppleTalk nebo DecNet.

I když John Sculley tvrdí, že nový osobní systém PS firmy IBM je zajímavý, není pochyb o tom, že boj o nový standardní počítač teprve začíná.

Ji-

Trh pamětí CD ROM do r. 1990

Podle odhadu marketinkové fy Input v Mountain View (Kal.) má být rozvoj trhu pevných pamětí CD ROM obdobný jako rozvoj trhu osobních počítačů na počátku 80.let. I když v roce 1986 nebo 87 nepřekročil prodej těchto pamětí hodnotu 30 mil.dol., v roce 1991 má dosáhnout hodnoty 937 mil.dol.

Podle mínění této firmy popularita pamětí CD ROM bude záviset na tempu, s jakým budou pronikat na trh osobních počítačů. Optické paměti CD ROM jsou již levnější než papírové kartotéky nebo mikrofilmové záznamy a na konci našeho desetiletí budou měrné náklady na uchování v těchto pamětech stokrát menší než právě uvedenými způsoby.

Přestože jde o pevné paměti, z nichž lze pouze vybavovat, mají velké možnosti a zcela jistě nahradí daleko vžitější typy trvalých nebo polotrvalých pamětí v řadě aplikací. Za polotrvalou paměť se považuje např. archiv, kde dochází v periodických obdobích k aktualizaci obsahu.

Na druhé straně jsou stále ještě bariéry bránící širokému prosazení těchto pamětí. Pořizovací náklady záznamů jsou stále ještě vysoké. Nejde jen o formátovou úpravu velkých objemů dat a informací, což je časově náročné a drahé, ale informace, které se mají ukládat, musí být dokonale zredigovány. Přehrávací jednotky pro CD ROM jsou také zatím drahé a dosud neexistují normy na technické ani na programové vybavení k řízení a integraci těchto pamětí do paměťových nebo počítačových systémů.

Jiří Kaplan

Další japonská optomagnetická disková paměť

Japonská firma Kobusai Denshin Denwa Co zahájila výrobu velkokapacitní optomagnetické diskové paměti s možností opětového zápisu, která je mimo jiné určena též pro mezinárodní veřejnou službu přenosu dat pod označením VENUS - P. Tato disková paměť je výsledkem společného vývoje s firmou Sony Corporation.

Základem je amorfní magnetická membrána v záznamové vrstvě. Průměr disku je 30 cm, kapacita zhruba 1 Gbyte, což je ekvivalentní obsahu 40 tis. stran formátu A4, tedy padesátinásobek kapacity magnetické pásky, a 5 500 až 1 000 násobek obsahu pružné diskety. Tuto kapacitu lze také vyjádřit jako obsah 93 svazků encyklopedie.

Předběžné zkoušky prokázaly, že optomagnetický disk udrží data při normální teplotě alespoň po deset let. Japonská firma bude využívat těchto disků k ukládání dat a zpráv v telekomunikačním provozu, dálnopisném, faksimilním apod.. Vzhledem k časovému rozdílu mezi Japonskem, Evropou a USA má toto meziuložení dat a zpráv svůj význam a využije se tak lépe telekomunikačních zařízení.

Optomagnetic disk stores 40 000 pages.
JEI 34, 1987, č.2, s.64

Jiří Kaplan 5

CD ROM

Přibližně před třemi lety, tj. na začátku roku 1986, se objevily první komerčně dostupné přehrávače CD ROM. Jejich nástup byl poněkud váhavý, zvláště v Evropě. Teprve během roku 1988 se očekávalo opravdu masové nasazení tohoto velkokapacitního média.

V USA dosáhl na konci roku 1987 počet prodaných přehrávačů CD ROM 1 miliónu. Celkový roční obrat na trhu CD ROM dosáhl jedné miliardy dolarů. Převážná většina uživatelů je z řad profesionálních pracovníků.

Poněkud méně příznivá je situace v Evropě, kde na konci roku 1987 dosáhl počet přehrávačů CD ROM pouze asi 6 700 kusů. Z toho připadá 4 000 kusů na Itálii, Velká Británie a NSR se podílejí každá 700 kusy, Francie 400, Skandinávie 200 a na ostatní západoevropské země připadá 700 kusů. V Itálii se sdružily společnosti Microsoft, Olivetti a Seat do organizace nazvané Eikon, která se zabývá prodejem a aplikacemi CD ROM. Představitelé této společnosti optimisticky tvrdí, že evropský trh je stejně perspektivní jako americký. Do konce roku 1988 předpokládají, že bude v Evropě prodáno 25 000 a do roku 1990 300 000 kusů přehrávačů CD ROM.

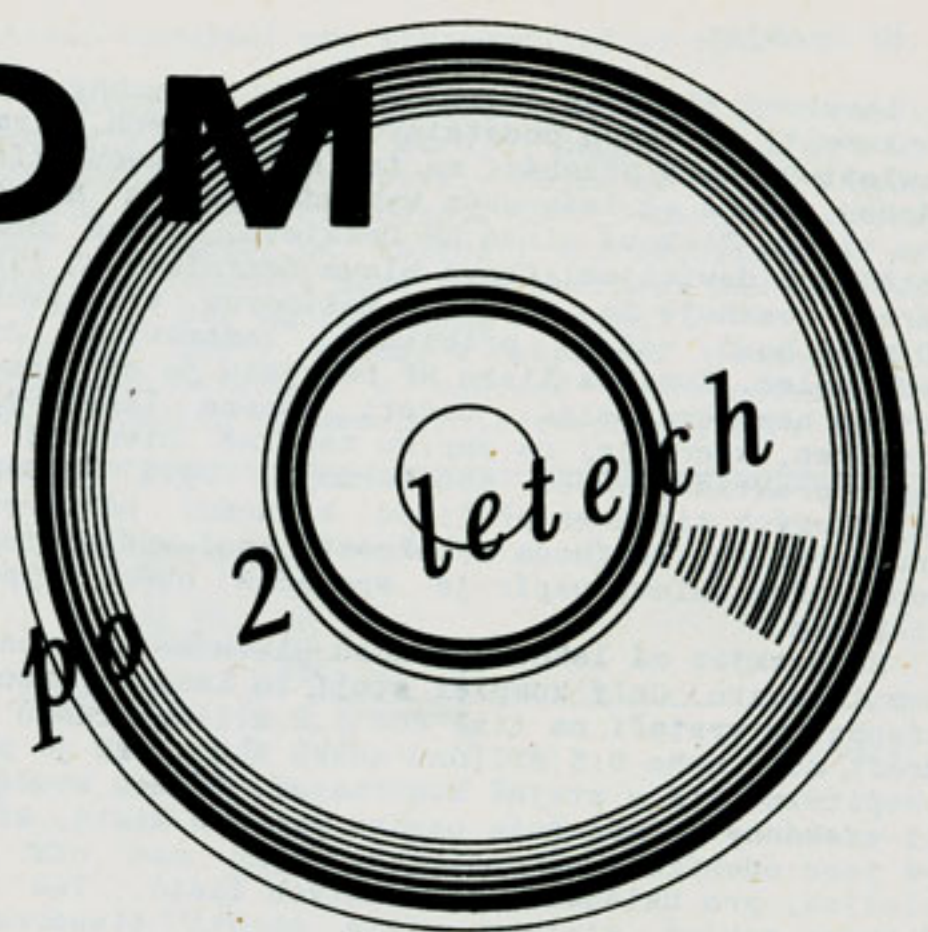
Největším evropským výrobcem přehrávačů je Philips se svojí řadou CM. Základním typem této řady je CM 100, o kterém firma tvrdí, že v oblasti CD ROM přehrávačů znamená totéž, co Apple II v oblasti mikropočítačů. CM 100 podobně jako všechny následující typy přenáší data rychlostí 150 kilobajtů za sekundu. Střední doba přístupu k datům je 1 sekunda. Připojuje se prostřednictvím interfejsu CM 153. Cena se pohybuje okolo 3 000 dolarů. Dalšími typy jsou CM 110, CM 121 a CM 131, které se od CM 100 výrazně neliší. Typ CM 200 je možné prostřednictvím interfejsu SCSI připojit k IBM PC. U typu CM 201 je navíc zvukový výstup, takže jej lze použít i k přehrávání zvukových kompaktních desek. U řady 200 se podařilo rovněž snížit střední přístupovou dobu na 0,4 s.

Na evropském trhu se dále uplatňuje firma Hitachi s přehrávači CDR 1502, CDR 1503 a CDR 3500. Tyto přehrávače rovněž poskytují zvukový výstup. Navíc je možné je po čtyřech sdružit do sítě (tzv. "daisy chain").

Kromě firmy Toshiba bude dalším vážným konkurentem firem Philips a Hitachi pravděpodobně firma Atari, která nabízí přehrávač CD 001, určený rovněž pro řadu ST. Tento přehrávač se bude prodávat za přibližně polovinu ceny přehrávačů Philips. Maximální rychlost přenosu dat bude 500 kilobajtů za sekundu.

Mnoho nejasností panovalo okolo přehrávače CD ROM pro počítače firmy Apple. Byl ohlášen až začátkem března tohoto roku pod názvem CD SC. Jeho rozměry by měly odpovídat rozměrům základní jednotky Macintosh, takže je bude možné stavět do věže. Kompaktní deska se vkládá do přehrávače zepředu v průhledné pružné obálce (systém "caddy"). Apple tvrdí, že se takto s deskou lépe manipuluje v kancelářských podmínkách a navíc je deska chráněna proti mechanickému poškození. Střední přístupová doba je 0,6 s, rychlost přenosu 150 kilobajtů za sekundu, ale bude možné použít i zvýšenou přenosovou rychlost 800 kilobajtů za sekundu díky vyrovnávací paměti 64 kilobajtů, která je přímo zabudována v interfejsu SCSI. Přehrávač obsahuje rovněž sekci D/A, zvukový výstup je ale možný pouze na sluchátka. Nevýhodou tohoto přehrávače je, že ho nebude možné připojit k IBM PC.

Aplikace CD ROM se soustřeďují zatím převážně do profesionálních sfér. Kompaktní desky slouží jako rozsáhlé katalogy velkých knihoven (například databáze Lise obsahuje na 400 000 titulů). Na kompaktních deskách se objevují katalogy průmyslového zboží, umožňující rychlé vyhledávání zboží a ve spojení s databází cen na pevném disku i jeho



účtování ve velkých obchodních domech. Širokého uplatnění dosáhnou CD ROM jako dokumentační materiál. Firma Renault poskytuje prostřednictvím jediné kompaktní desky kompletní dokumentaci svých výrobků. Databáze obsahuje 12 000 obrázků, 80 000 stran textu - katalog součástí a speciálního nářadí, manuály pro opravy. Databáze je přístupná prostřednictvím IBM PC s podporou programů Hypertext a Hyperdoc. Podobně firma Hewlett-Packard dodává na kompaktní desce dokumentaci CD ROM pod názvem Laser ROM.

Firma Apple nabízí ke svému přehrávači pět kompaktních desek, které jsou vlastně zvukovými encyklopediemi. První z nich vychází z Essential Whole Earth Catalogue, což je zeměpisná a přírodovědná encyklopedie. Whole Earth Learning Disc umožňuje studium např. určitého ptačího druhu nejen prostřednictvím textu a obrazu, ale je možné si také poslechnout záznam jeho zpěvu. Druhá deska se nazývá Groelien American Series a zahrnuje historii USA v období 1800 až 1850. Třetí se jmenuje Jefferson project a zabývá se ústavou a zákonodárstvím v USA. Čtvrtá deska obsahuje přes 100 megabajtů textu a 10 000 obrázků z antického Řecka. Poslední kompaktní desku ocení hlavně studenti medicíny, neboť se jedná o komfortní anatomický atlas s poněkud morbidním názvem Electric Cadaver.

Zajímavým způsobem využívá možností kombinace CD ROM, videodesky a počítače firma Aeroinformation v projektu CVAE (Computer and Video Assisted Engineering). Systém pro výcvik pilotů airbusů A310 a A320 doprovází prostřednictvím CD ROM obrazový materiál na videodesce 25 hodinami mluvené řeči. Počítač Zenith 2248 komunikuje s přehrávačem CD ROM, z něhož čte data s informačním tokem 32 bajtů za sekundu a předává je na výstup prostřednictvím karty pro syntézu řeči. Tímto způsobem lze na jednu kompaktní desku o kapacitě 550 megabajtů zaznamenat s použitím metody ADPCM až 36 hodin kontinuílní řeči.

Evropa dnes stojí teprve na začátku využití možností CD ROM. Naše země jako producent přehrávačů kompaktních desek a od letošního jara i samotných kompaktních desek má velkou šanci uplatnit výrobky tohoto druhu. Budeme však schopni pružně reagovat, nebo zůstaneme u tradice vlaku, který nám ujel?

Ing. Jiří JIRÁČEK

Literatura:

- [1] Charreyron, V.: 1988, l'année de tous les espoirs. Ordinateur individuel No 101, mars 1988.
- [2] Garret, Y.: CD ROM pour Macintosh. Science et vie micro No 49, avril 1988.

Z88

R47
might even prove actively dangerous, and would certainly be
unwise. To understand our position, it may be worth looking
at the costs in the following spreadsheet extract:

	1987	1988	1989
Site Purchase	45,500	5,500	5,500
Site Prep	23,750	2,600	

SINGLE KEY FUNCTIONS

To abort press ESC	End of row	End of column	Next column	Next word	Previous word	Previous column
Clear marks	Delete row	Format paragraph	Next column	Next word	Previous word	Previous column
Delete character	Delete word	Insert row	Next column	Next word	Previous word	Previous column
Delete end of doc	Edit expression	Insert row	Next column	Next word	Previous word	Previous column
BLOCK MENU	CURSOR MENU	EDIT MENU	FORMAT MENU	LAYOUT MENU	OPTIONS MENU	PRINT MENU



Z88 COMPUTER

Počítačová tvořivost známého sira Clive Sinclaira nevydržela být dlouho nečinná ani po odprodání práv včetně jména firmě Amstrad. Clive Sinclair vymyslel a vyvinul další počítač, opět s oblíbeným mikroprocesorem Z80 (tentokrát v provedení C-MOS). Ve stručnosti uvádíme některé základní parametry tohoto malého přenosného kompaktního počítače o rozměrech 292 x 210 x 22 mm a váze 1 kg.

Mikropočítač se dodává s 32 kilobajty RAM. Jeho operační systém však je schopen pracovat s pamětí až 4 MB. Paměť lze rozšiřovat zásuvnými moduly o kapacitách 32 kB, 128 kB nebo 1 MB. Do počítače lze současně vsunout tři takovéto moduly. Počítač se nikdy zcela nevypíná a všechna data a programy v pamětech RAM jsou trvale uchovávána. Umožňuje to použití obvodů C-MOS s minimální spotřebou.

Z88 nemá ani kazetový magnetofon, ani disketovou jednotku. Ekvivalentní záznam programů a dat se provádí do paměti EPROM. Nejsou zde tedy žádné pohyblivé součásti, poškoditelná páska nebo diska. Paměť EPROM se po naplnění vymění za jinou, lze ji samozřejmě i obvyklým způsobem smazat. Je to řešení velmi originální, i když nám se bude zdát asi vzhledem k našim cenám paměti EPROM poněkud drahé a luxusní.

Počítač má sériový port RS232. Lze jej tak propojit s většinou používaných tiskáren. Lze jej ta-

ké propojit s počítačem typu IBM PC a přesunout do něj všechny texty a data. Z88 tak poslouží jako malý přenosný terminál většího počítače. Dodává se i modem k přenosu dat po telefonu.

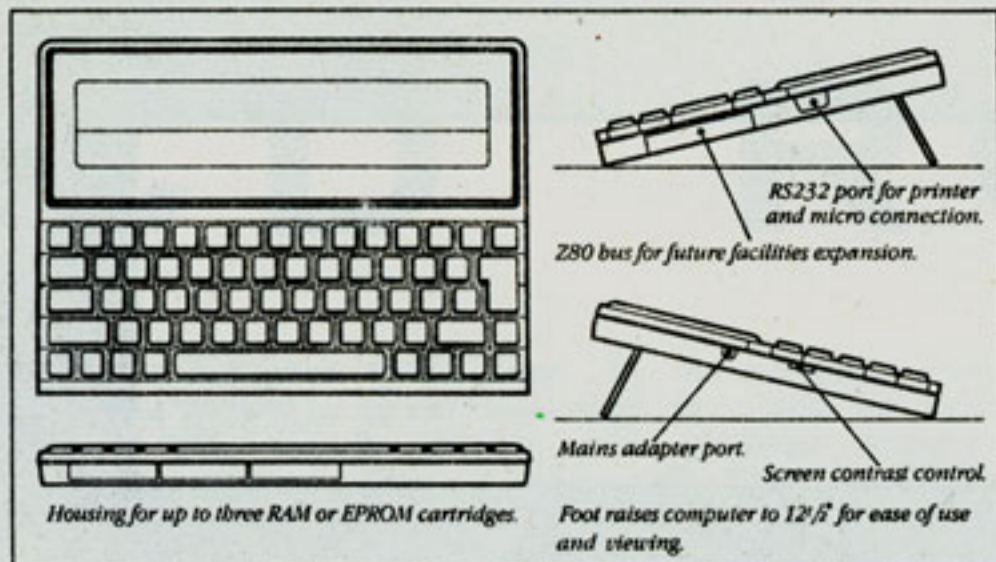
Mikropočítač Z88 se napájí ze čtyř tužkových baterií nebo ze síťového napáječe. Baterie vydrží na 20 hodin aktivního používání a téměř 1 rok udrží obsah paměti RAM (ve "sleepy" režimu). Displej upozorní uživatele na blížící se nutnost výměny baterií.

Klávesnice je tlačítková, typu "short travel", naprosto tichá a vhodná tedy k uvažovanému využití počítače jako zápisníku při nejrůznějších příležitostech.

Supertwist LCD displej je velmi kontrastní, dává klidný čistý tmavěmodrý obraz na světlešedém pozadí. Je rozdělen funkčně do čtyř sekcí. Největší sekce (uprostřed) je pracovní prostor, zobrazující 8 řádků po 80 znacích, vertikálně i horizontálně posunovatelných (scroll). Vpravo od pracovního prostoru je dynamická "mapa" celé stránky, zobrazující v každém okamžiku stávající grafické uspořádání celé stránky. Vlevo od pracovní oblasti je "topic bar" - seznam sedmi základních režimů počítače (cursor, edit, layout, files, blocks, print, options). Lze z něj vybrat a tlačítkem "me-

4-section LCD display

CURSOR EDIT LAYOUT FILES BLOCKS PRINT OPTIONS	R47 might even prove actively dangerous, and would certainly be unwise. To understand our position, it may be worth looking at the costs in the following spreadsheet extract:			
	Site Purchase	1987 45,500	1988 5,500	
	Site Prep	23,750	2,600	



Tvar počítače 288

nu" zvolit příslušnou činnost. Systém menu umožňuje zacházení s počítačem i úplným začátečníkům. Čtvrtá část displeje zcela vpravo trvale sleduje a zobrazuje status 288.

Počítač 288 má vlastní operační systém, který podporuje dodávané programové vybavení, umožňuje využít všech vlastností mikroprocesoru 280 a umí pracovat s pamětí až 4 MB. Rezidentní programovací jazyk je BBC BASIC. Integrovaný programový balík obsahuje textový procesor, spreadsheet, databázi, kalkulátor, diář, kalendář a hodiny reálného času. Lze plynule přecházet z jednoho bloku do druhého i mezi různými úlohami, aniž by bylo nutné zajišťovat ukládání dat nebo jakkoli ošetřovat přerušení práce. Displej zobrazuje vše věrně, včetně tučného písma, kursivy, podtržení a rozdělení stránek. Bez problémů lze i přesouvat data mezi jednotlivými bloky (např. z textového procesoru do tabulek nebo z databáze do textu apod.).

Jak se rozběhla výroba počítače 288 nevíme. Jeho cena v době uvedení na trh byla okolo 200 anglických liber. Vyrábí a dodává jej, pouze na přímou objednávku, Cambridge Computer Ltd., FREEPOST, Cambridge, CB41BR, England.

-ra

KOPIROVÁNÍ OBRAZOVKY

Přestože je téměř u všech osmibitových počítačů implementována grafika, nenalezneme většinou mezi příkazy jazyka BASIC příkaz pro kopírování grafické obrazovky na tiskárně. Dále popsany program umožňuje kopírování grafické obrazovky u počítačů Commodore C 16, C 116, plus/4.

Program můžeme umístit do hlavního programu jako podprogram a volat ho příkazem GOSUB 61000. Podprogram při prvním vyvolání zavede do paměti rutinu ve strojovém kódu a při dalším volání pouze přepisuje část rutiny určující měřítko zobrazení. Podprogramu můžeme pomocí čtyř parametrů předat požadavek na měřítko na ose x,y (1:1;2:1) a na rozměr kopírovaného okna OX (1 až 40), OY (1 až 25). Nežadáme-li žádné vstupní parametry, podprogram se automaticky nastaví na zkopírování celé obrazovky v měřítku 1:1. Některé důležité programové řádky:

- 61000-61140 - úvodní popis,
- 61160-61200 - nastavení vstupních parametrů,
- 61210-61310 - zavedení rutiny ve strojovém kódu,
- 61320-61350 - přepis rutiny pro zvolené měřítko,
- 61370-61400 - počáteční adresy kopírovaných řádků,
- 61410 - nastavení řádkování tiskárny,

- 61460 - nastavení tiskárny do grafického režimu,
- 61490 - posun tiskárny o řádek,
- 61530-61680 - přepis rutiny pro zvolené měřítko,
- 61690 - nastavení normálního řádkování tiskárny.

Existuje pouze jedno omezení, a to je počet kopírovaných znaků (znak=8x8 bodů) na ose x při měřítku 2:1, tj. $OX \leq 35$ (šířka papíru 200 mm). Nebude-li vám vadit menší vzdálenost bodů osy X než u osy Y, můžeme vypustit řádek č.61200 a přepsat řádek č.61460 `PRINT#4,CHR$(27);"*";CHR$(4);CHR$(M1);CHR$(M2);`. Po této úpravě je možno kopírovat celou obrazovku i při měřítku osy X 2:1.

Program je odladěn na počítači Commodore plus/4 s tiskárnou CITIZEN 120-D, ale bude fungovat beze změn i na počítačích Commodore C 16, C 116. Místo tiskárny CITIZEN můžeme použít jakoukoliv tiskárnu se sériovým přenosem dat, ale musíme upravit programové řádky 61410, 61460, 61490, 61690. Při použití tiskárny s paralelním přenosem by bylo nutné upravit rutinu ve strojovém kódu.

Ing. M. NUTIL

```

61000 REM *****
61010 REM          KOPIROVANI OBRAZOVKY
61020 REM *****
61030 REM
61040 REM (C) ING. NUTIL MIROSLAV 1988
61050 REM
61060 REM **** POUZITE PROMENNE ****
61070 REM MX-MERITKO NA OSE X , STABILNE MX=1
61080 REM MY-MERITKO NA OSE Y , STABILNE MY=1
61090 REM OX-POCET ZNAKU NA OSE X , STABILNE OX=40
61100 REM OY-POCET RADEK , STABILNE OY=25
61110 REM SK-URCUJE ZDA JE PROGRAM VE STROJ. KODU V PAMETI SK=0 NE
        SK=1 AND
61120 REM CT-CITAC DVOJITEHO PRUCHODU CYKLU PRI MY=2
61130 REM D1-UKLADANI PROGRAMU VE STROJ. KODU DO PAMETI
61140 REM M1,M2-NASTAVENI TISKARNY NA POCET KOPIROVANÝCH ZNAKU

```



```

61150 OPEN 4,4
61160 IF MX<>1 AND MX<>2 THEN MX=1
61170 IF MY<>1 AND MY<>2 THEN MY=1
61180 IF OX=0 OR OX>40 THEN OX=40
61190 IF OY=0 OR OY>25 THEN OY=25
61200 IF MX=2 AND OX>35 THEN OX=35
61210 IF SK=1 THEN 61320
61220 RESTORE 61230
61230 DATA 169,4,32,177,255,169,4,73,96,32,147,255,160,0,162,0,177,216
,149,208
61240 DATA 200,232,234,234,234,234,224,8,208,241,162,8,134,219,162
,0,138,168
61250 DATA 181,208,41,128,192,0,240,5,74,136,76,137,6,149,224,232,224,
8,208,234
61260 DATA 169,0,21,223,202,208,251,32,168,255,234,234,234,234,234,162
,8,22,207
61270 DATA 202,208,251,198,219,208,206,24,165,216,105,8,133,216,144,2,
230,217
61280 DATA 198,218,208,168,32,174,255,96
61290 FOR K=1630 TO 1733
61300 READ D1: POKE K,D1
61310 NEXT K: SK=1
61320 IF MX=1 THEN 61540
61330 IF MY=1 THEN 61590
61340 IF MX=2 THEN 61640
61350 IF MY=2 THEN 61670
61360 M2=INT((8*MX*OX)/256): M1=(8*MX*OX)-(M2*256)
61370 RESTORE 61380
61380 DATA 32,00,33,64,34,128,35,192,37,00,38,64,39,128,40,192,42,00,4
3,64,44,128
61390 DATA 45,192,47,00,48,64,49,128,50,192,52,00,53,64,54,128,55,192,
57,00
61400 DATA 58,64,59,128,60,192,62,00
61410 PRINT#4,CHR$(27);"A";CHR$(8);
61420 FOR K=1 TO OY
61430 CT=1
61440 READ VA,NA
61450 POKE 218,OX
61460 PRINT#4,CHR$(27);"*";CHR$(5);CHR$(M1);CHR$(M2);
61470 POKE 216,NA: POKE 217,VA
61480 SYS 1630
61490 PRINT#4,CHR$(27);CHR$(10);
61500 IF CT=1 AND MY=2 THEN CT=2: NA=NA+4: GOTO 61450
61510 NEXT K
61520 GOTO 61690
61530 REM ***** MX = 1 *****
61540 FOR K=1699 TO 1703
61550 POKE K,234
61560 NEXT K
61570 GOTO 61330
61580 REM ***** MY = 1 *****
61590 FOR K=1652 TO 1656
61600 POKE K,234
61610 NEXT K
61620 GOTO 61340
61630 REM ***** MX = 2 *****
61640 POKE 1699,169: POKE 1700,0: POKE 1701,32: POKE 1702,168: POKE 17
03,255
61650 GOTO 61350
61660 REM ***** MY = 2 *****
61670 POKE 1652,169: POKE 1653,00: POKE 1654,149: POKE 1655,208: POKE
1656,232
61680 GOTO 61360
61690 PRINT#4,CHR$(27);"A";CHR$(14): CLOSE4

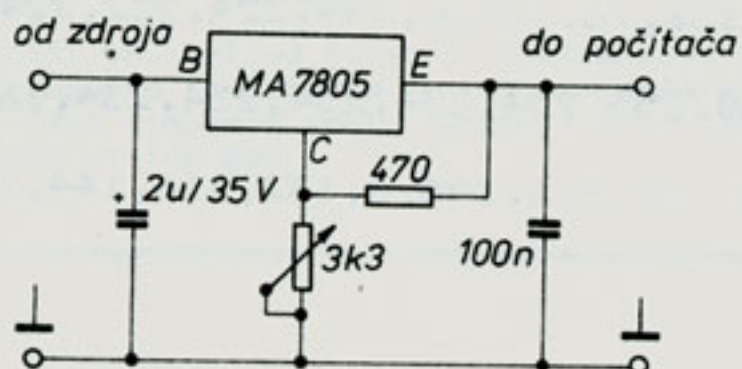
```


STABILIZÁTOR

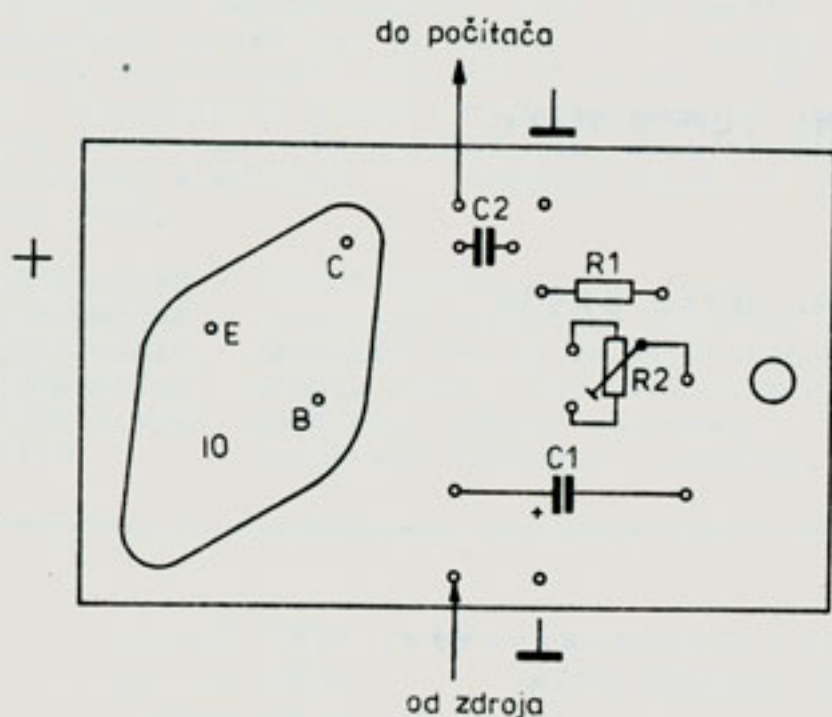
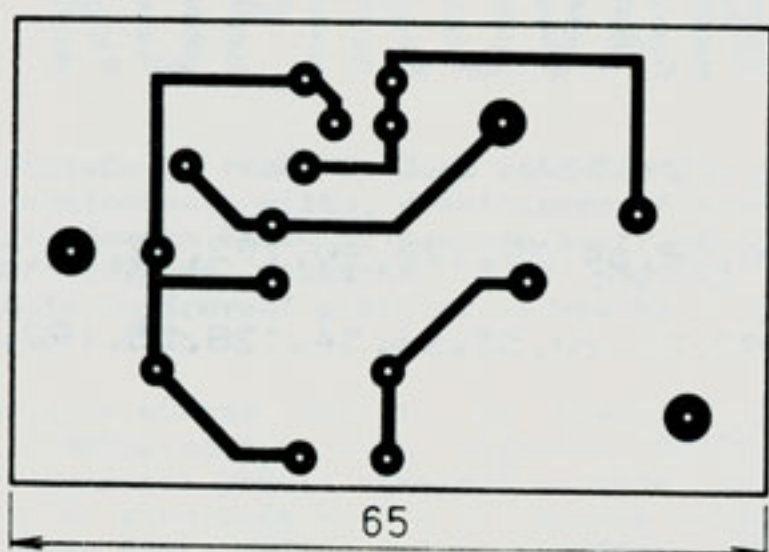
pre mikropočítač ZX-Spectrum

Ak používame počítač bez prídavných zariadení napájaných priamo z neho, dochádza k jeho značnému prehrievaniu, čo môže spôsobiť vážnu poruchu. Je to dané príliš vysokým napätím z nestabilizovaného továrenského zdroja, ktoré sa musí zrážať priamo v počítači.

Opísaný stabilizátor zráža napätie na 8V a tvorí kompaktný celok s pôvodným napájačom. Je s ním spojený nedeštruktívnym spôsobom a kedykoľvek sa možno vrátiť k pôvodnému stavu.



Obr. 1. Schéma zapojenia stabilizátora



Obr. 2. Plošné spoje a rozmiestnenie súčiastok

Jednoduchý obvod s využitím stabilizátora 7805 nie je potrebné popisovať. Doštička plošného spoja sa zoskrutkuje do jedného celku s chladiacim plechom (hliník, 2 mm) a stabilizátorom skrutkami M4 x 12.

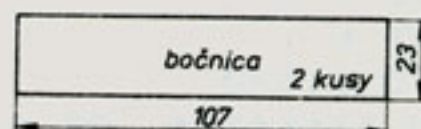
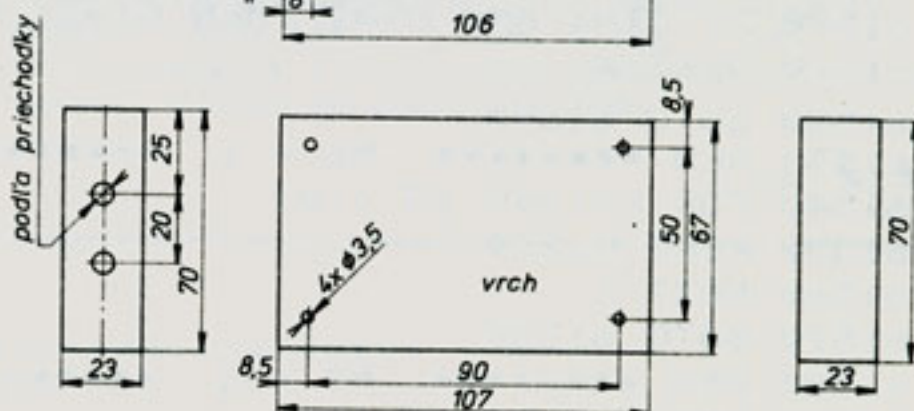
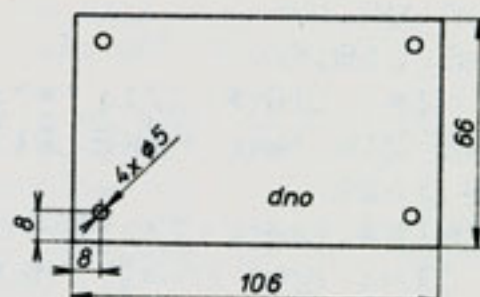
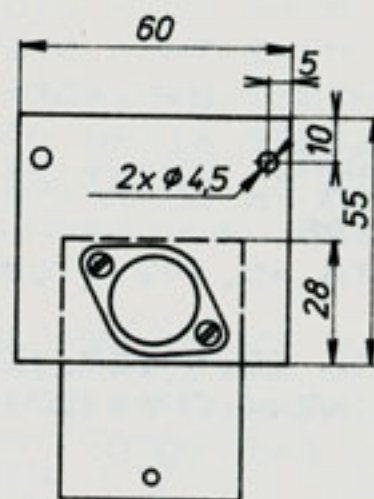
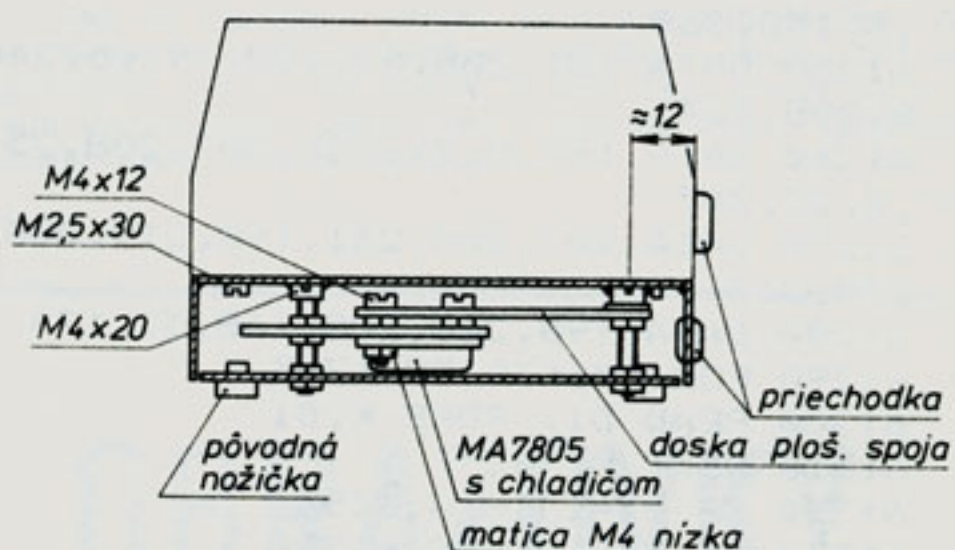
Skrinka je z Cuprexitu 1,5 mm hrubého. Diely zhotovené podľa obrázkov spájkujeme k sebe. Na vrchný diel skrinky prispájkujeme hlavami 3 skrutky M4 x 20 do polôh podľa náčrtku a podľa otvorov v doštičke s chladičom. Skrinku po zdrsnení všetkých plôch brúsnym papierom nastriekame čiernym lakom.

Pôvodný zdroj demontujeme uvoľnením 4 skrutiek po vybratí gumových nožičiek a káblik s konektorom nahradíme kúskom iného, ktorý bude spájať zdroj so stabilizátorom. Pôvodnú navádzaciu ohybnú priechodku možno nahradiť obyčajnou. Skrutky do plas-

tu nahradíme skrutkami M2,5 x 30, ktoré idú bez problémov zaskrutkovať do pôvodných otvorov. Ak ich nemáme, závitníkom M3 číslo 2 upravíme otvory a skrinku pripevníme skrutkami M3x30. Po pripevnení doštičky s chladičom (výšku doštičky je treba vymedziť podložkami, aby se nedotýkala Cuprexitu, prípadne možno podlepiť lepenkou) pripojíme cez priechodky v bočnici vstupný a výstupný káblik. Výstupné napätie nastavíme na 8 V. Pozor - kladný pól je na obrube súosého konektora, zem uprostred.

Do dna podľa polohy troch upevňovacích skrutiek M4 x 20 vyvrtáme otvory priemeru 4,5 mm. Potom vtlačíme do otvorov vo dnu pôvodné gumové nožičky a dno priskrutkujeme. Ešte raz skontrolujeme správnosť funkcie stabilizátora a matice zakvapneme lakom.

Ing. Vladimír ODEHNAL



Obr. 3. Mechanické časti a celkové usporiadanie

Rozpiska súčiastok

R1	470	TR 151
R2	3,3 k	TP 012
C1	2 uF/35V	TE 986
C2	100 nF	TK 782
IO	MA 7805	
Skrutky:	M2,5 x 30 (M3 a 30)	4 ks
	M4 x 20	3 ks
	M4 x 12	2 ks
Matica M4 nízka		15 ks
Priechodka		3 ks

MODULY *pro ZX-Spectrum*

KVT Hradec Králové, sekce Sinclair

Pod tímto názvem předkládají členové Klubu výpočetní techniky z Hradce Králové svým kolegům několik úprav či přídatných zařízení pro Spectrum. Určitě u znalců vzbudí mnoho kritiky, ale i zájmu, a je nakonec věcí každého amatéra, co si ke svému Spectru či Deltě postaví.

Mnoho nápadů je převzatých, ne však s úmyslem chlubit se cizím peřím, ale vytvořit řadu či stavebnici úprav ke Spectru. Za vzornou spolupráci děkují Janu Ruppertovi a Ing. Františku Kottnerovi a členům našeho Klubu výpočetní techniky.

1.0 ÚVOD

Koncepce stavebnice spočívá v jednoduchém modulovém řešení. Každý modul má pouze jeden účel. Prvním modulem je TURBO úprava, druhým interfejs CENTRONICS, třetím interfejs pro Kompston joystick. V plánu máme další moduly.

Základním modulem je TURBO. Ke Spectru bude připojen přímým konektorem na sběrnici. Sběrnice bude pokračovat jako průchozí konektorem FRB, do kterého můžeme zasunout buď modul CENTRONICS pro připojení tiskárny nebo modul KEMPSTON pro připojení ovládače her. Nejsou použity zesilovače sběrnice, proto nedoporučujeme zařazovat více než dva moduly najednou. Modul CENTRONICS nabízí výstupní signály pro tiskárnu CONSUL 2111 třicetivývodovým konektorem FRB, který je zapojen shodně jako u systému SAPI 1. Z toho plyne, že kterákoliv tiskárna připojitelná k SAPI 1 bude připojitelná i k tomuto modulu přímo kabelem od SAPI 1.

Rozhodne-li se někdo pro stavbu jen jednoho modulu, osadí ho přímým konektorem (upravený WK 46580 dle AR 4/87) pro připojení na sběrnici ZX Spectra. Další možností je zhotovit redukci z přímého konektoru WK 46580 a FRB a moduly osazovat vždy konektorem FRB.

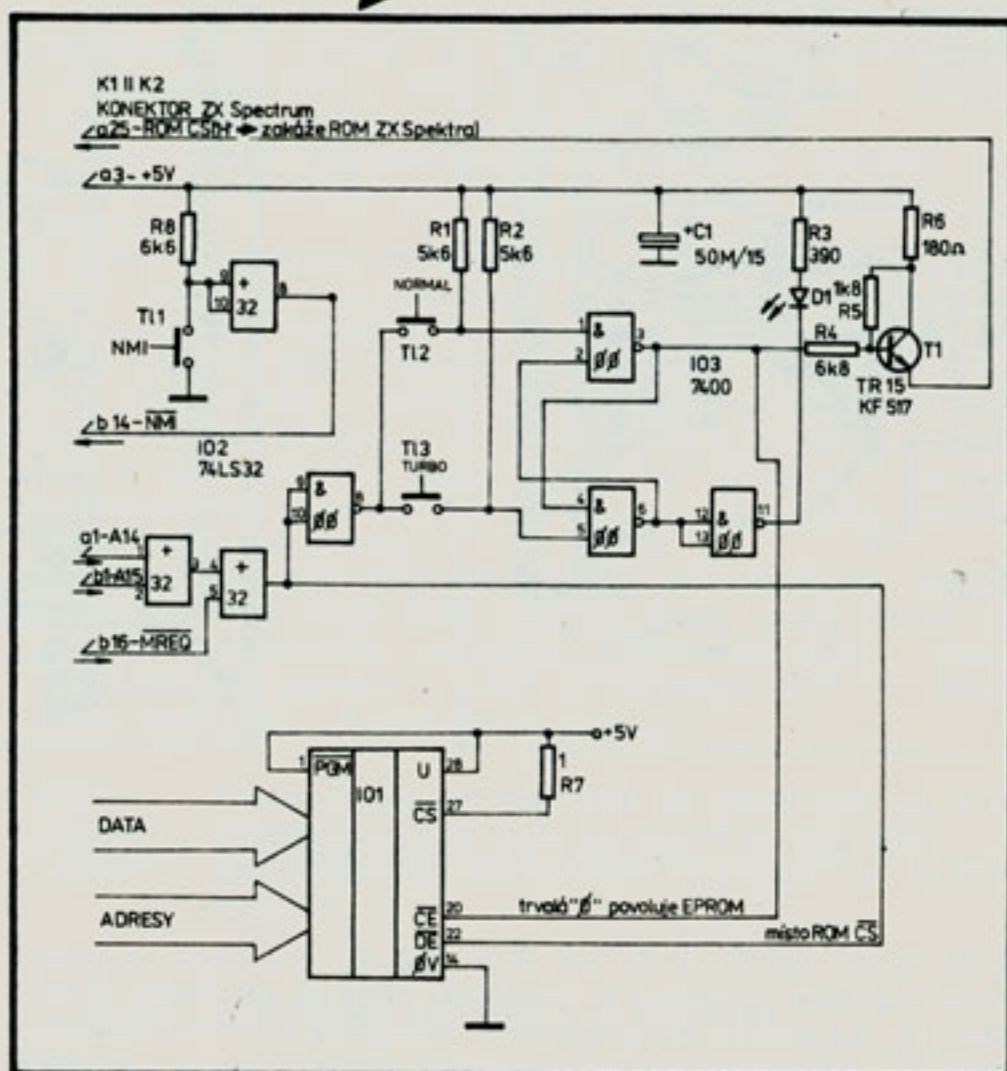
2.0 Modul TURBO

2.1 Popis zapojení

Obvodové schéma modulu TURBO je na obr. 1. Modul umožňuje připojit ke Spectru 16 kB paměť EPROM 27128 tak, že si tlačítka TURBO či NORMAL vybíráme buď naši přidanou nebo původní ROM Spectra. Obě jsou adresovány od nulté adresy, zabírají tedy prvních 16384 adres. I když jsme fandové TURBA, které bude popsáno dále, s radostí konstatujeme, že do patice pro EPROM 27128 lze samozřejmě vložit i YU-ROM či ISO-ROM. Všechny úpravy programu těchto pamětí vycházejí z původního obsahu ROM Spectra, každá má své výhody či nevýhody. Většina původních programů bude fungovat i s uvedenými alternativními ROM.

Tlačítka TURBO / NORMAL je ovládán klopný obvod RS, který jednak spíná diodu LED signalizující použití přidané paměti EPROM, jednak zabezpečuje povolení výběru zvolené paměti. Přes tranzistor T₁ úrovní "H" zakazuje výběr ROM Spectra a úrovní "L" na vývodu 20/EPROM povoluje výběr přidané paměti EPROM. K vlastnímu výběru paměti dochází signálem na vývodu 22/EPROM. Ten je odvozen pomocí hradel obvodu 74LS32 ze signálů A14, A15 a MREQ. Protože tento signál je aktivní v úrovni "L", odvodíme si, že za součtovými hradly bude úroveň "L" tehdy, budou-li signály A14, A15 i MREQ ve stejný okamžik na úrovni "L". To je pouze v případě, že procesor si přeje komunikovat s pamětí, adresovanou ve spodních 16 kB.

Uvedené logiky výběru paměti EPROM je využito i k povolení okamžiku přepnutí paměti ROM / EPROM. Tlačítka TURBO / NORMAL jsou společně napájena z



Obr.1. Schéma zapojení modulu Turbo

invertovaného signálu pro výběr paměti EPROM, což způsobuje jejich funkci jen v případě, že tento signál je na úrovni "L". V opačném případě nedojde k překlopení klopného obvodu RS. Lze přepínat paměti za běhu programu, což je velká výhoda tohoto zapojení.

Původní ROM Spectra má chybnou rutinu pro obsluhu nemaskovaného přerušení NMI. U dalších typů ROM je tato chyba odstraněna, proto máme k dispozici tlačítka pro vyvolání NMI. Majitelé "gumových" Specter si mohou zapojit tlačítka RESET obdobným způsobem. Mechanické uchycení tlačítek je ponecháno na Vás. Můžete tedy počítat i s tlačítkem pro RESET a jeho spínací kontakt zapojit vodiči (mezi zem a vývod 43/FRB, resp. a20 dle označení na schématu Spectra).

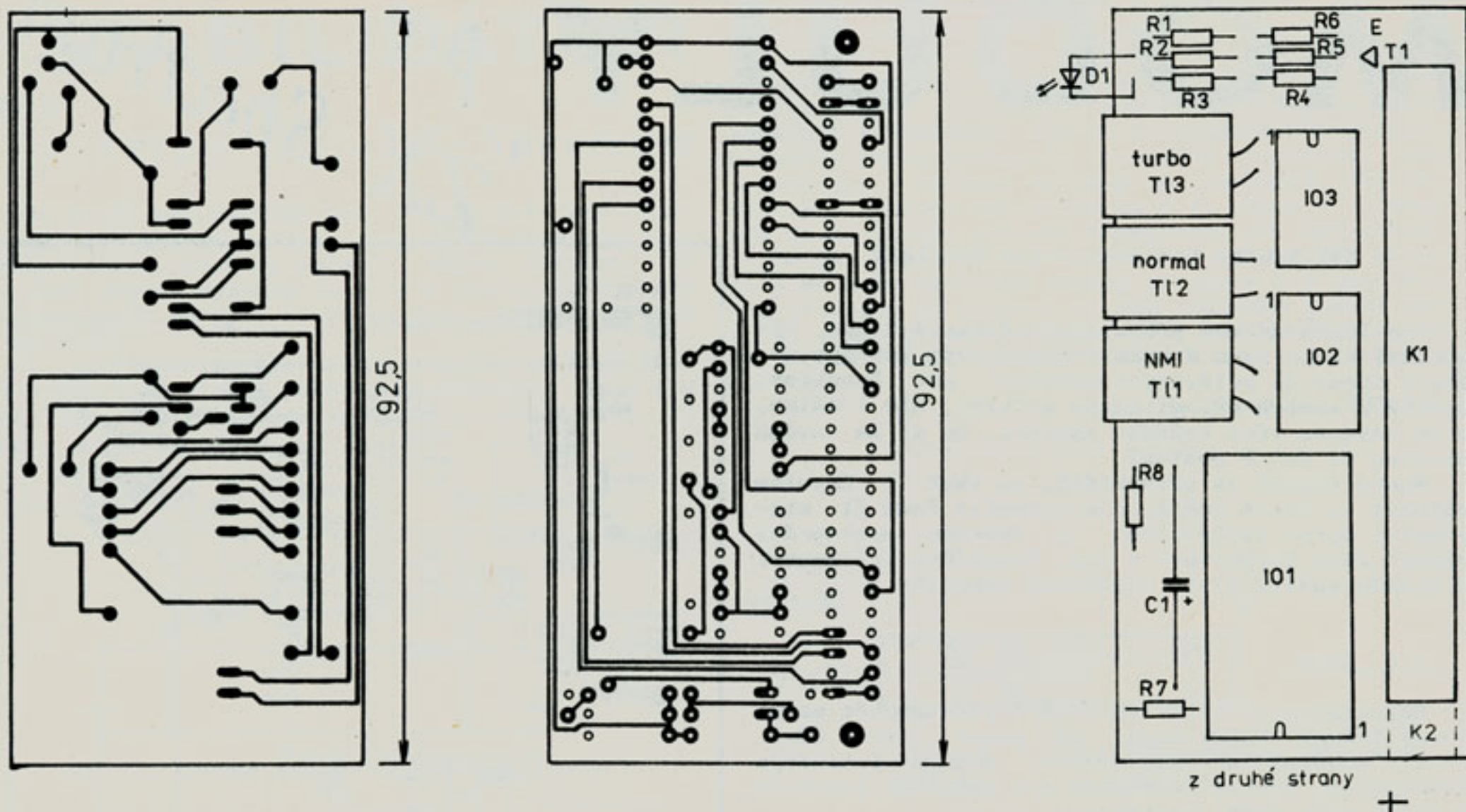
2.2 Stavba a oživení

Na obr. 2 je obrazec plošných spojů obou stran desky a rozmístění součástek na ní.

Při osazování desky s plošnými spoji o rozměrech 43,75 x 92,5 mm s neprokořenými otvory zkontrolujte nejprve plošné spoje a proveďte propojení, nahrazující prokoření v místech, kde toto propojení nebudou zabezpečovat součástky. Nezapomeňte zejména na propojení pod 74LS32. Potom pečlivě připájejte patici pro paměť EPROM. Výhodnější než klasická je "stříhací" z dovozu; lze pájet i v místech, která budou pod pamětí. Potom zasuňte do otvoru pokračovací konektor FRB ze strany spojů, vodiče konektoru nezkracujte! Připájejte FRB z obou stran desky. Na přečnívající vodiče konektoru ze strany součástek připájejte přímý konektor, u kterého opět nezkracujte jeho vývody! Po zasunutí přímého konektoru do Spectra musí zůstat dostatek místa pro součástky na desce mezi modulem a Spectrem.

Pozor! V případě, že součástku nelze připájet ze strany spojů, je třeba protáhnout současně s vývodem součástky tenký pocínovaný vodič, který pak lze pájet z obou stran spoje.

Po připájení všech součástek zkontrolujte, zda nedochází ke zkratům mezi sousedními vývody konektoru FRB. Zkontrolujte také, zda není zkrat mezi vývody, na které je přiváděno ze Spectra napájení



Obr. 2. Obrazce plošných spojů a rozložení součástek na desce modulu Turbo

5 V, 9 V, 12 V, 0 V, -5 V a -12 V, a všemi ostatními vývody. Jednou z častých příčin zničení Spectra je lajdácké připojování různých (i firemních) modulů. Zkontrolujte propojení napájecí a nulové sběrnice na desce a počítači (nejlépe ohmmetrem).

Při ožiování nejdříve změřte odběr modulu ze zdroje +5 V. Měl by být asi 25 mA (dioda nesvítí). Potom připojováním úrovně "L" na vývody 1 a 5 obvodu 7400 vyzkoušejte funkci klopného obvodu RS. Jeho správnou funkci signalizuje dioda LED. Současně tímto způsobem zkontrolujeme správnou funkci tranzistoru T1.

Prozatím jsme vše dělali bez Spectra. Nyní (a vždy když budete něco připojovat ke Spectru) vypněte počítač a zasuňte konektor modulu na sběrnici Spectra. Televizi mějte již zapnutou a zapněte počítač. Musí proběhnout inicializace obvyklým způsobem. V opačném případě zkontrolujte signál na a25 z kolektoru T1. Očekáváme úroveň "L" při funkci původní ROM Spectra.

Po stlačení tlačítka TURBO se rozsvítí dioda LED, odepne se ROM Spectra a povolí se paměť EPROM. Zda je funkční, poznáme u TURBO-EPROM podle změny typu písma, či po RESET podle inverzní obrazovky a nového výpisu v editační zóně.

Je-li funkce Spectra v "NORMÁL" obvyklá a v "TURBO" ne, proměřte připojení všech signálů k paměti EPROM.

Po zapnutí počítače není jednoznačně definováno, zda se připojí původní ROM Spectra či přidaná EPROM na modulu. Jestliže chcete mít o tuto jednoznačnost postaráno, zapojte kondenzátor o kapacitě 10 nF jedním vývodem na vývod 2 nebo 4 a druhým na vývod 7 obvodu MH7400.

Seznam součástek modulu TURBO

I01	paměť EPROM typu 27128-150 (fy NEC, HITACHI ap.) vybavovací doba paměti max. 250 ns
I02	74LS32 (sovětský ekvivalent K555LL1)
I03	MH7400 (možná náhrada libovolným ekvivalentem)
T1	TR15 nebo KF517 (v podstatě možno nahradit jakýmkoliv typem p-n-p)
D1	dioda LED (libovolná)
T11,2,3	tlačítka se spínacím kontaktem (typ omezen pouze velikostí); tlačítka přilepíme na desku modulu

C 1	20 až 50 uF / 15 V
R 1,2,8	4,7 až 10 nF
R 3	390 (omezuje proud diodou LED)
R 4	6,8 k
R 5	1,8 k
R 6	180 k
R 7	1 k
K 1	WK 46580 (úprava dle AR 4/87)
K 2	TX 5183215 (či obdobný FRB s 62 dutinkami) pro průběžnou (pokračovací) sběrnici Spectra
patice 28 vývodů	TX 7875281 či obdobná pro paměť EPROM
ostatní patice	na IO dle vlastní úvahy

2.3 Stručný popis softwarových změn

2.31 Úvod

Každý počítač má podle firemních dokladů jen výhody a většinou se pravda o daném zařízení objeví teprve v okamžiku, kdy pomine počáteční nadšení a měla by začít vážná práce.

Ani při práci s velmi rozšířeným a oblíbeným mikropočítačem ZX SPECTRUM se praxe neobejde bez zklamání. Většina nás pracuje bez INTERFACE1 a microdrovů. Proto je jedna z nepříjemných stránek provozu doba, promarněná až pětiminutovým čekáním při práci s kazetovým magnetofonem. Toto čekání rádi podstoupíme v případě zajímavé hry, kterou pak provozujeme celý večer. Používáme-li však počítač k práci, jsou minuty vzácné. Tato skutečnost a několik jiných nedostatků vedla k úpravě "TURBO-EPROM".

Tato úprava umožňuje provádět příkazy LOAD, SAVE, VERIFY a MERGE rychlostí více než dvojnásobnou. Po dvouletém zkušebním provozu a několika verzích zůstává v platnosti verze "MON 6 A".

Kvalita moderních kazet a kazetových magnetofonů dovoluje používání vyšších rychlostí pro přenos dat při zachování spolehlivosti přenosu. Používání různých způsobů rychlého nahrávání (např.: SPEEDY-LOAD, Q-SAVE aj.) se příliš neosvědčilo. Překáží totiž v paměti RAM, nedají se použít ve spojení s dlouhými programy a komplikují obsluhu. Program, který má vlastní zrychlené nahrávání (viz SIGMA BASIC), se dá používat bez obtíží, zrychlení je ale vázáno na tento program.

Z používaných kazetových magnetofonů se osvědčily typy Panasonic RQ-8100, Fair Mate CS-660,

Tesla K-10. Poslední uvedený typ však nemá počítadlo. U prvního typu se osvědčila úprava, která spočívá v přidání kondenzátorů 10 nF paralelně k původním, které jsou označeny jako C14 a C15.

Z používaných levnějších kazet se osvědčily především kazety C 60 s označením TYPE I, BIAS NORMAL / EQ 120 μs, a to výrobky FUJI DR 60, DYNAMIC C 60 CASSETTE od ELECTRONIC INDUSTRIES CO. BAGHDAD - IRAQ, které se objevily i v naší obchodní síti.

2.32 Popis změn

ZÁZNAM NA MGF rychlost asi 3x500 bitů/s, úprava spočívá ve změně časových konstant rutin pro SAVE/LOAD, úvodní tón - 1250 Hz, původní pomalé rutiny: LOAD na adrese #3B39, SAVE na adrese #3AA5. Touto úpravou je vyřešeno zrychlené SAVE/LOAD pro 99% programů. Pouze některé hry používají vlastní LOADER. Tyto provozujeme v režimu NORMAL.

GENERÁTOR ZNAKŮ dle CPC 464 fy. Amstrad (výraznější znaky),

NMI se vrací zpět do BASICu. Tím je opravena chyba v ROM, která znemožnila používání NMI.

Oblast paměti #5B00 - #5BFF se při inicializaci nemaže (ani po příkazech NEW nebo RAND USR 0). Lze ji tak využít pro vlastní rutiny (obsluha tiskárny apod.).

VEKTOR pro printer je nastaven na #3A88 (tam je rutina pro výstup ASCII na port 223 (pro paralelní výstup např. připojení tiskárny CONSUL). Původní hodnota byla #09F4. READY tiskárny se čte na portu 191 (0). Navržený modul CENTRONICS tak dokáže bez dalších úprav komunikovat s tiskárnami, které jsou určeny pro SAPI 1.

Drobné změny:

Autorepeat	#1271	:	12	02
Varovný tón, zvuk tlačítka	#1213	:	05	32
Inverzní obraz (INK 7, PAPER 0)	#1266	:	07	
Nemaže print buffer	#0EDF	:	18	0A
Výmaz pouze od #5C00	#11D8	:	3E	5B
černý okraj (border)	#11CD	:	00	
Kratší interval mezi header a prg.	#0990	:	08	

Tabulka časových konstant:

pro S A V E			pro L O A D		
ADRESA	TURBO	NORMAL	ADRESA	TURBO	NORMAL
#04C7	E6 12	80 1F	#0572	0A 02	15 04
#04DF	69	A4	#0581	8E	9C
#04E9	17	2F	#0588	9A	C6
#04F7	0E 1E	0E 3B	#0590	BE	C9
#0519	1A	42	#0598	CC	D4
#051F	15	3E	#05A6	E2	B0
#052E	08	31	#05C7	E4	B2
			#05CF	EB	CB
			#05D4	E2	B0
			#05E8	07	16

Nový povel tlačítkem CAT

Je to pomůcka hlavně pro zvědavé. Nahrazuje jakýsi mikromonitor. Usnadňuje pracné přepočítávání a ulehčuje přístup k systému.

Po stlačení tlačítka dostaneme menu:

H_EX pro přepočet HEX ↔ DEC
D_EC pro přepočet DEC ↔ HEX

M_EM pro výpis a změnu paměti
SPACE = adresa + 1
ENTER = zápis do paměti
Q_UIT konec

2.33 Zjištěné nedostatky

Pro uvedené změny bylo nutno využít oblast #386E-#3C1F, která v orig. ROM nebyla využita (obsahovala #FF). Několik programů zřejmě využívá tuto oblast nepřímo (pracují v IM2).

Dále mají některé programy vlastní obsluhu nahrávání; v těchto případech používáme původní ROM Spectra.

Při používání INTERFACE1 se nepředpokládá nutnost použití zrychleného nahrávání pro kazetový magnetofon. Pokud přesto budete TURBO užívat, musíte paměti ROM / EPROM přepínat ve vhodnou chvíli. INTERFACE1 a MICRODRIVE můžeme provozovat pouze v režimu NORMAL. Při přepínání z TURBO na NORMAL v případě, že byly použity příkazy, určené původně pro microdrive (CAT, ...), může nastat zhroucení programu.

Prozatím nevyužijeme povely pro obsluhu vnější CMOS RAM 48k (RAM DISC). Tyto jsou použitelné po doplnění Spectra modulem TURBO-RAM-DISC, který opět ve spolupráci s námi připravuje KVT v Uherském Hradišti. Jedná se o povely:

MOVE Přesun obsahu 48k RAM do CMOS-RAM za 0,5s
FORMAT Přesun z CMOS RAM do vnitřní 48k RAM
ERASE Vzájemná výměna obsahu vnitřní a vnější paměti

2.34 Speciální software pro TURBO

a) použitelné pouze s TURBO-ROM:

turbo/key převodní program KEY V2.0 upravený pro obě rychlosti. Přepíná se tlačítkem "T" a zvolená rychlost je indikována v levé dolní části pod označením "speed:"
N = norm. rychlost 1500 bitů/s,
T = turbo rychlost 3500 bitů/s.
Provádí LOAD,SAVE,VERIFY v obou rychlostech.

FRE(c)TURBO převodní program Free Copy pro TURBO.

STX/v1.6T program typu HEADER-READER pro rychlost TURBO. Kontroluje správnost nahrávek, vypisuje údaje z hlaviček a vytvoří seznam, který je možno ještě editovat.

T-TURBO(c) převodní program typu 2speedcopy. Dovoluje přepisovat programy v délce až 48825 bajtů. Přepíná se pomocí tlačítek "T" a "N".

b) Použitelné pro ROM i TURBO-EPROM:

FREE(c)SPEED konverze TURBO/NORM, přepíná se pomocí tlačítka "T". Může posloužit i ke zjištění vhodnosti daného kazetového magnetofonu pro rychlost TURBO.

2.4 Závěrem

Modul TURBO je určen především pro ty amatéry, kteří mají možnost naprogramování paměti EPROM 27128. Podrobný popis stavby programátoru paměti EPROM 2716 až 27256 přinesla maďarská RADIOTECHNIKA č. 7 a 8/1986, a to včetně návrhu tištěných spojů a výpisu programu s podporou strojového kódu. Náš klub má v plánu letošní rok toto zařízení postavit a dát tak možnost nejen členům našeho klubu paměti EPROM naprogramovat.

Chtěli bychom umožnit i všem čtenářům Mikrobáze získání naprogramované Turbo EPROM, protože nebylo reálné vytisknout výpis programu (zabere příliš mnoho místa v časopisu a času při "přečukávání" do počítače, obvykle s těžko odstranitelnými překlepy a chybami).

3.1 Popis zapojení

Obvodové schéma modulu CENTRONICS je na obr.3. Modul umožňuje připojit ke Spectru pomocí tohoto paralelního interfejsu tiskárnu CONSUL 2111, resp. CONSUL 2113, a všechny další tiskárny, které jsou připojitelné přímo k počítači SAPI-1. Uvedené tiskárny jsou připojitelné přímo kabelem ze SAPI-1, protože výstup modulu je na třicetivývodovém konektoru FRB, který je zapojen shodně jako konektor v SAPI-1 na desce JPR-1. Lze samozřejmě připojit i ostatní tiskárny, které mají paralelní vstup typu CENTRONICS. V tomto případě si musíte podle manuálu tiskárny zapojit propojovací kabel s patřičným konektorem, resp. zvolit správnou polaritu řídicích signálů AC a SC.

Paralelní výstup je tvořen obvodem MH3212, který byl vybrán zejména pro jeho výkonové výstupy (proti obvodu MH8255, jehož zatížitelnost je max. 4mA, což pro tiskárny CONSUL nestačí).

Po zapnutí je obvod nulován díky časové konstantě R4.C2, která způsobí pomalý náběh z nuly do úrovně "H" (proti rychlejšímu náběhu napájecího napětí). Obvod R4.C2 je připojen do vstupu RESET portu MH3212, a protože je tento signál aktivní v úrovni "L", způsobí jeho pomalejší náběh vynulování 8-bitového registru. Do tiskárny jsou pak přiváděny na datových vodičích úrovně "L".

Počítač dodává obvodu data instrukcí OUT 223. Výběr obvodu I01 určují hradla I02 typu 74LS32, a to pomocí signálů \overline{IORQ} a \overline{WR} , resp. adresovacího signálu A5. Protože I02 obsahuje pouze součtová hradla, jedná se o součet uvedených signálů. Tento součet je veden do vstupu S1 obvodu I01, který je aktivní v úrovni "L". Protože za součtem bude aktivní "L" jenom tehdy, budou-li uvedené signály na úrovni "L" všechny současně, logicky si zdůvodníme dobu výběru obvodu I01. Negované signály \overline{IORQ} a \overline{WR} přijdou z mikroprocesoru vždy tehdy, chce-li procesor zapisovat data na port. A to je u I01 povoleno dále jen tehdy, nebude-li v úrovni "H" adresový vodič AS. Tímto způsobem je vybrán I01 a bajt z datové sběrnice ZX-Spectra zapsán do registru I01. Z popsaného adresování je zřejmé, že adresa 223 nebude jediná, na kterou bude popsán obvod reagovat. Měl by být funkční na všechny adresy, ve kterých je A5 na úrovni "L".

V okamžiku zápisu bajtu do I01 se na pinu 23 (signál IT) objeví na okamžik úroveň "L". Tento impuls je přiveden jako spouštěcí signál do monostabilního klopného obvodu I04 typu UCY 74121. Zde je vygenerován impuls definované délky (podle časové konstanty R3.C4). Tento impuls je veden do tiskárny jako žádost o převzetí nabízeného znaku (SC).

Manuál tiskárny by měl definovat jeho polaritu a dobu trvání. Pro tiskárnu CONSUL 2111 je aktivní

v úrovni "H" a časová konstanta R3.C4 ho definuje na délku asi 50 μ s. Pro tiskárnu EPSON je aktivní v úrovni "L" a musí být delší než 5 μ s.

Polaritu signálu SC volíme na I04, který má výstup jak přímý, tak i invertovaný.

Akce s tiskárnou může začít, je-li tiskárna připravena. To dává najevo signálem AC, který si počítač přečte, aby věděl, zda může tiskárně dodat další znak.

Tento opačný tok informace je zajištěn přes vstupní čtyřbitový port, který je tvořen I03 typu MH3216. Obvod je vybírán signálem CS (vývod 1), který je aktivní v úrovni "L". Logika výběru je obdobná jako u I01. Opět jsou využita součtová hradla I02 typu 74LS32. Tentokrát musí být současně v úrovni "L" signály \overline{IORQ} , \overline{RD} a A6. Procesor chce číst port z adresy, která nemá A6 na úrovni "H". Tomu odpovídá adresa 191. Můžeme tedy číst informaci o připravenosti tiskárny instrukcí IN 191. Informace je přiváděna na datový vodič D0, proto v přečtené informaci je rozhodující pouze bit 0.

Některé tiskárny mají opět signál AC aktivní v opačné úrovni ("L"). Na obr. 7 je tranzistorem KC507 tento signál invertován, na plošném spoji je s touto možností počítáno.

3.2 Stavba a oživení

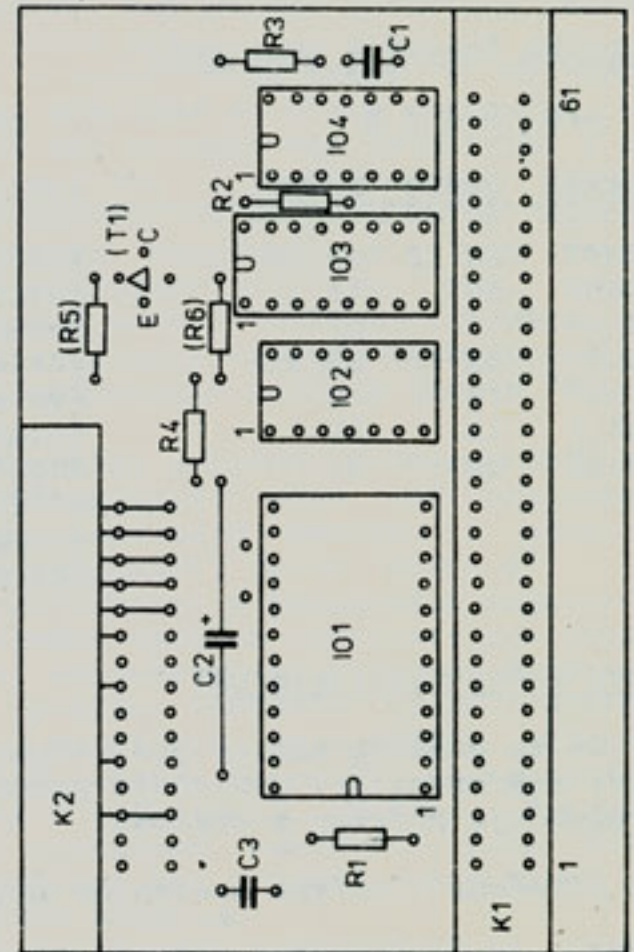
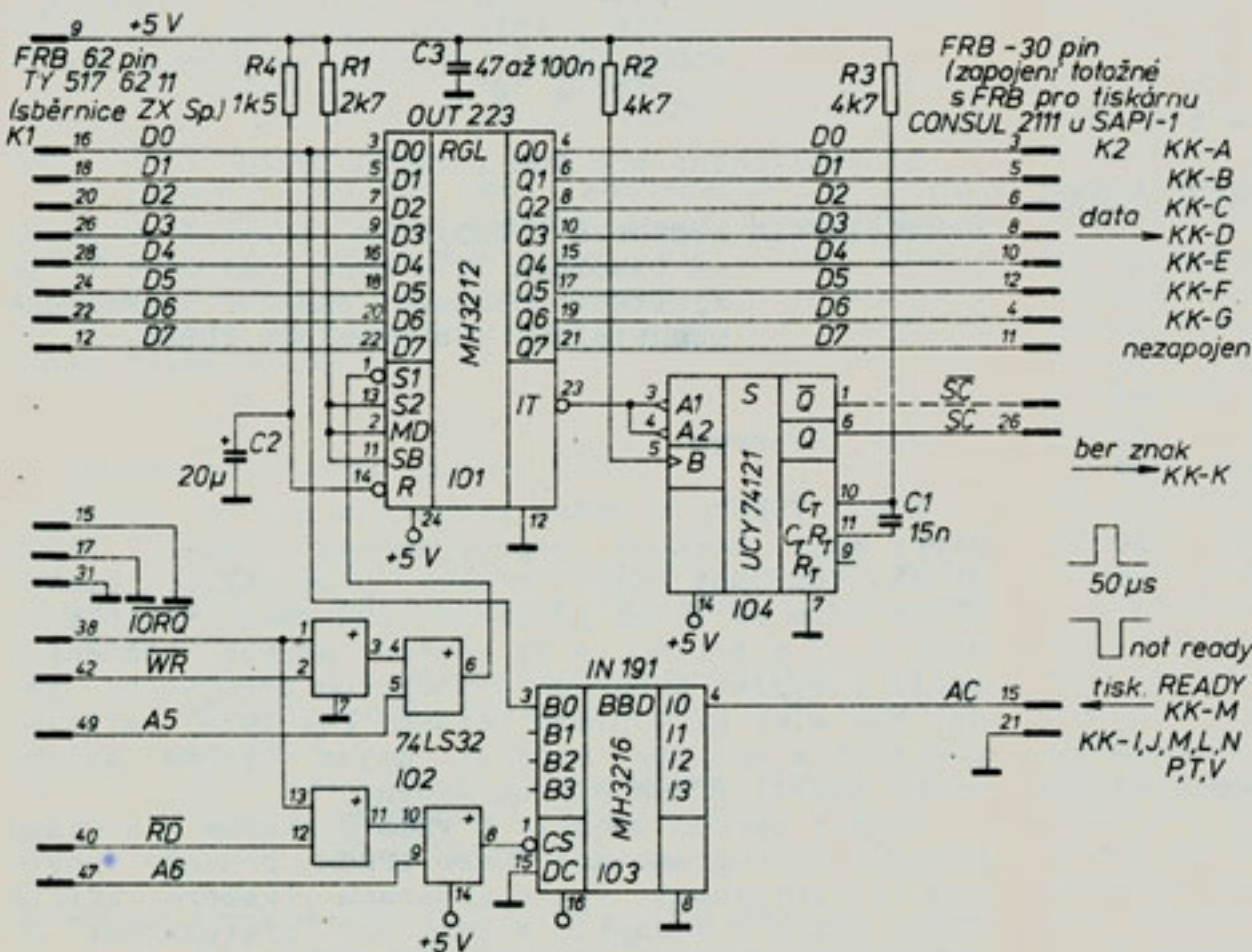
Obrazce plošných spojů obou stran desky modulu jsou na obr. 4. Rozmístění součástek je zřejmé z obr.3. Rozměr desky je 60 x 92,5 mm. Strana o délce 60 mm bude zmenšena podle čárkované čáry na spoji v případě, že budeme modul používat "na stojato" (nebude ležet plochou na stole).

Před osazením desky se musíme rozhodnout, zda budeme používat modul samostatně (bez TURBA) a zda bude provozován "nastojato" nebo "naležato". Podle toho pak osadíme modul vhodným typem konektoru ze strany součástek, případně i pokračovacím konektorem, který bude sběrnici Spectra poskytovat případným dalším modulům. Tuto úvahu učiňte obdobně jako u modulu TURBO i s ohledem na výšku zástavby součástkami a návaznosti na okolí modulu.

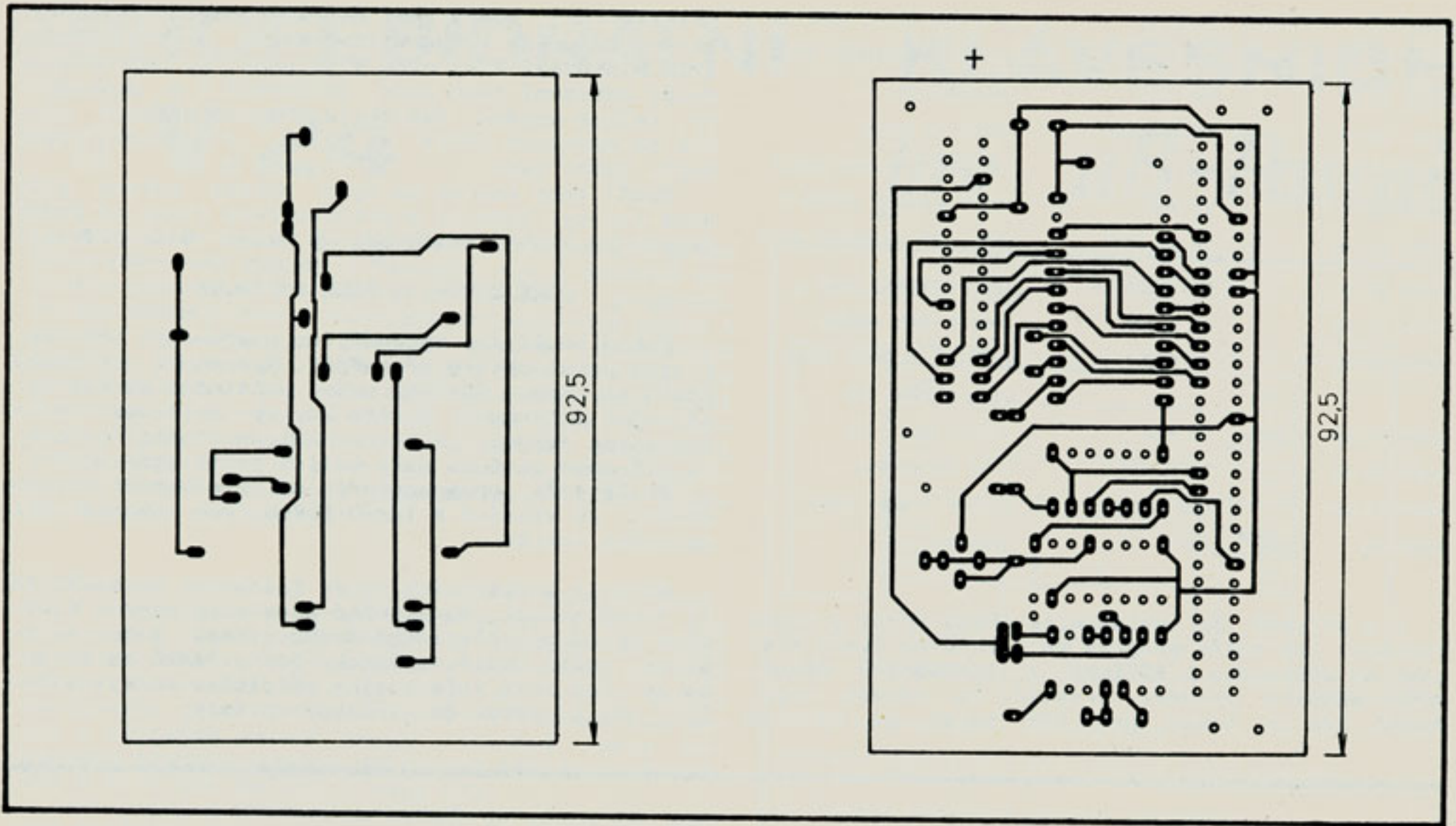
V případě řazení více než dvou modulů si alespoň propočtem zjistěte, zda nepřetěžujete některý ze signálů sběrnice Spectra.

Po připájení spojů, nahrazujících prokovení, po osazení desky součástkami a po volbě polaritu řídicích signálů se řiďte radami v popisu modulu TURBO. Nejprve zkontrolujte možné zkratky mezi sousedními vývody konektoru, zejména mezi vývody sousedícími s těmi, na které je přiváděno napájecí napětí.

První připojení proveďte pouze na zdroj +5 V (mimo Spectrum). Odběr by měl být asi 200 mA. Logickou sondou můžeme zkontrolovat výstupy I01 až na konektoru K 2 - vývody 3,5,6,8,10,2,4,11. Jsou na úrovni "L". Některé vstupy I01 a I04 jsou přes



Obr. 3. Schéma zapojení a rozmístění součástek na desce modulu Centronics



Obr.4. Obrazce plošných spojů obou stran desky modulu Centronics

rezistory R1, R2 a R4 připojeny na úroveň "H". Vše zkontrolujte přímo na uvedených IO podle schématu na obr. 6.

Ohmmetrem lze zkontrolovat cestu výběrových signálů obvodů I01 a I03.

Při vypnutém Spectru připojíme modul ke sběrnici. Po zapnutí musí proběhnout inicializace Spectra.

Další oživení probíhá ještě bez tiskárny. Napíšeme krátký program v BASICu, kterým ověříme správnou funkci I01. Postupně zapisujeme instrukcí OUT 223 data 1,2,4,8,16,32,64,128. S každou touto instrukcí musí být obvod vybrán (S1 - vývod 1 - úroveň "L"). S každým zápisem bajtu do obvodu musí vyjít signál IT - vývod 23 - úroveň "L", který startuje monostabilní klopný obvod. Zkontrolujeme až jeho výstup na konektoru K 2 - vývod 15. Všimneme si správně navolené polaroty tohoto pulsu.

Na výstupu I01 zkontrolujeme, zda je příslušný bajt dat nahrán beze ztráty bitu.

Nyní přistoupíme k připojení tiskárny. Ta by se měla zapínat do sítě vždy při vypnutém Spectru nebo alespoň při stisknutí tlačítka RESET. Rázy při zapnutí způsobují často kolizi programu.

Předpokládáme správně zapojený propojovací kabel. Je-li tiskárna READY (připravena), je na vývod 15 konektoru K 2 přivedena úroveň "H".

Instrukcí PRINT IN 191 počítač vypíše liché číslo (např. 255) v případě, že je tiskárna READY. V opačném případě vypíše sudé číslo (např. 254).

Instrukcí OUT 223,65 vypíše tiskárna znak A - viz tabulku znaků v manuálu Spectra.

Neprovádí-li tiskárna očekávané funkce, zkontrolujte nejprve řádné propojení kabelem včetně propojení 0 V. Také polarita řídicích signálů mohla být zvolena opačně!

Vyskytnou-li se při ožívování větší potíže, doporučujeme napsat vhodný program v BASICu a po jeho spuštění proměřit cestu dat i signálů logickou sondou nebo osciloskopem.

Příklad programu:

```
5 INPUT "zadej data: ";DATA
10 IF (IN 191 AND 1) THEN OUT 223,DATA
20 GOTO 10
```

Hodnotu proměnné DATA volíme od 32 do 122 (podle manuálu tiskárny či Spectra).

Seznam součástek modulu CENTRONICS

I01	MH3212
I02	74LS32
I03	MH3216
I04	UCY74121N
T1	KC507 (použít pouze tehdy, je-li třeba invertovat signál AC z tiskárny - viz text), (možno nahradit obdobným typem n-p-n KC 147,148 ap.)
C1	15 nF (může být i keramický polštářek)
C2	20 uF / 15 V
C3	47 až 100 nF
R1	2,7 k
R2,3,6	4,4 k
R4	1,5 k
R5	479 až 680

Poznámka: R5 a 6 jsou zapojeny jen při použití T1.

3.3 Stručný popis základního software

Původní hodnota vektoru pro tiskárnu je #09F4. Na této adrese se nalézá rutina pro ZX Printer, resp. pro tiskárnu SEIKOSHA GP 50 S.

Adresu rutiny pro příkazy LPRINT a LLIST můžeme změnit opravou kanálových informací. Rutina pro COPY není adresována přes kanálové informace, bez úpravy ROM Spectra jí nelze měnit. Dokážeme tedy po našich úpravách využívat přímo pouze příkazy pro běžný tisk, ne pro tisk obrazovky. Máme-li k dispozici tiskárnu, která umí tisknout i grafiku, lze rozšířit vlastní rutinu pro tisk o test, který by při zjištění např. kódu 30 (neznamená žádný znak ASCII) způsobil přechod na vlastní rutinu pro COPY (přes PRINT CHR\$ 30).

V TURBO-EPROM je vektor pro tiskárnu změněn na hodnotu #3A88, což je adresa, na níž je uložena rutina:

#3A88 FE A5	PRTOK CP #A5	:TOKENY NA PORT
#3A8A 38 07	JR C,PRNT	:ODSKOK NENÍ-LI TOKEN
#3A8C D6 A5	SUB #A5	:A-REG MINUS #A5
#3A8E CD 10 0C	CALL #0C10	:000A ZA TOKENE SERII
		:ASCII ZNAKŮ
#3A91 18 03	JR PREND	:SKOK NA KONEC
#3A93 CD 4C 3A PRNT	CALL PRDUT	:VLASTNÍ TISK ZNAKU
#3A96 C9	PREND RET	:ZPĚT PO TISKU ZNAKU

#3A4C F5	PROUT	PUSH AF	:V A-REG ASCII ZNAK
#3A4D DB BF	READY	IN A,(191)	:ČTE PŘIPRAVENOST TISK.
#3A4F CB 47		BIT 0,A	:NA DATU DO
#3A51 28 FA		JR Z,READY	:ČEKA NA READY
#3A53 F1		POP AF	:JE READY - ZPĚT ASCII
#3A54 FE 0D		CP #0D	:NOVÝ ŘÁDEK ? ENTER ?
#3A56 20 06		JR NZ,DATA	:NE - JDI TISKOUT DATO
#3A58 D3 DF		OUT (223),A	:AND = NAVRAT VOZU
#3A5A 3E 0A		LD A,#0A	:#0A = NOVÝ ŘÁDEK
#3A5C 18 EE		JR PROUT	:ZPĚT A AŽ BUDE READY
			:TAK PROVEĎ #0A
#3A5E FE 0A	DATA	CP #0A	:JE-LI KÓD (10
#3A60 38 02		JR C,NAV	:JDI NA NAVRAT !
#3A62 D3 DF		OUT (223),A	:JINAK TISKNI ZNAK
#3A64 C9	NAV	RET	:NAVRAT

I v případě, že nevládníte TURBO-EPROM s touto rutinou, lze tento program ve zdrojovém tvaru přepsat do některého z EDITORŮ / ASSEMBLERŮ (např. MONS, GENS) a provést překlad na adresu např. #5B00, kde je místo pro 256 bajtů po bývalém

printbufferu, který nyní nepotřebujeme. Printvektor se dá opravit buď univerzálně, ale složitěji (viz MIKROBÁZE č.4) nebo v případě, že nepoužíváte další zařízení jako např. INTERFACE 1 a mikrodrive, celkem snadno. Vektor je v tomto případě uložen na adresách 23749 a 23750, v pořadí nižší bajt, vyšší bajt.

Stačí tedy zapsat na tyto adresy adresu naší nové rutiny. Bude-li v printbufferu (tedy od #5B00, dekadicky 23296) stačí nám k tomu dva příkazy:

POKE 23749,0: POKE 23750,91.

Pokud používáme microdrive, musíme být při definici printvektoru přesnější. Systémová proměnná CHANS na adrese 23631+2 udává počáteční adresu kanálových informací. K této adrese přičteme 15 a dostaneme adresu, na kterou můžeme uložit adresu, na níž máme uloženu naši rutinu pro tiskárnu.

Zkušenější programátoři si uvedenou rutinu zkrátí. Je vypsána z TURBO-EPROM, kde nemohla být umístěna vcelku.

Máte-li k dispozici jinou tiskárnu, prostudujte si řádně manuál. Bude možná obsahovat mnoho řídicích znaků pro přepínání druhů písma, jeho velikosti, počtu znaků na řádek, počtu řádků na stránku ap. Pak musí Vaše rutina příslušné znaky povolit, aby se vůbec do tiskárny dostaly.

LASEROVÝ TISK BEZ LASERU

Vypadají a chovají se jako laserové tiskárny, ale nemají v sobě žádný laser. Na trh s nimi přišly firmy Taxan (Crystal Jet) a Data Technology (CrystalPrint VIII). Místo laseru a mechaniky pro rozmítání paprsku mají pevný řetězec světlo emitujících diod LCS (Liquid Crystal Shutter). Tyto diody, lidově řečeno, "sedí" těsně vedle sebe na tyči jako slepice na hřadě. Jde však pořád o tentýž xerografický princip, kdy osvětlená část světlocitlivého bubnu tiskárny mění svůj náboj, čímž buď přitáhne jemný prášek barvicího toneru nebo ne.

Každá z řetězce 2400 diod je ovládána individuálně a je schopna "vystřelit" světelný svazek vysoké intenzity. Teoreticky má takováto konstrukce tiskárny některé přednosti. V laserové tiskárně je na přesnost velmi náročný a proto i snadno zranitelný mechanismus přenosu paprsku laseru pomocí rotujících zrcátek a optiky. To u LCS tiskáren odpadá - jejich mechanika je jednodušší, odolnější a lehčí. Proto i výroba celého zařízení je levnější. Tolik teorie. Ale ani Crystal Jet (3495 \$), ani CrystalPrint VIII (2495 \$) ji nijak nedemonstrují. Jejich ceny i tiskárny samy jsou stejně "masivní" jako laserové. Firma Taxan prohlásila, že její LCS tiskárna bude podstatně levnější, až vzroste sériová výroba.

Obě uvedené tiskárny tisknou 8 stran A4 za minutu. Jsou vybaveny mechanikou Casio LCS se samostatným zásobníkem toneru (není spojen s bubnem). Hustota tisku (300 bodů/palec) je vysoká, ale kvalita je o trochu horší než u laserové tiskárny HP LaserJet. Pod lupou je vidět, že podání šedé není jednolitě, na bílém poli se objevují drobné, i když normálně neregistrovatelné body a ani kontury obrysů nejsou srovnatelně ostré. Výkyvy šedé plochy jsou patrné i zkušenějšímu oku. Je však možno říci, že podání tisku je pro běžné užití velmi vyhovující. Uvedené drobné nedostatky mají svůj původ v tom, že přímo na buben zaměřené diody LCS nevystřelují světlo tak okamžitě a skokově, jak bychom si přáli. To se týká i jejich zhasínání.

Z obou tiskáren vychází papír potisknutou plochou nahoru, což je nepraktické, protože natisknu-

tou složku s navazujícím textem je pak nutno přerovnávat ručně stránku po stránce. Automatické zásobníky papíru umožňují i odběr obálek a transparentního materiálu. Celkem dobře provedená je mechanika, která brání prokluzování papíru, i když při testech k němu občas docházelo.

Na panelu tiskáren je dvouznakový LCD displej, který informuje o stavu a chybách tisku. Takto kódovaná informace však uživatele nutí k věčnému prohlížení tabulek s informačními kódy.

Crystal Jet se dodává se 2 MB paměti RAM. Jeho příjemnou, nekonvenční zvláštností je, že tvary znaků, s nimiž pak pracuje, se do něj předem softwarově "napouštějí" (download). Tento způsob práce se znakovými sadami je o dost pružnější, než vyměňování drahých rodkových kazet. I řada dalších funkcí tiskárny může být stanovena či ovládána softwarově - tiskárna spolupracuje s CPU hostitelského počítače, což jí dodává vysoký stupeň flexibility.

Separátní zásobník toneru ulehčuje uživateli obsluhu v porovnání s LaserJetem, kde zásobník se světlocitlivým bubnem tvoří jeden komplet. Tak také nemusíte zbytečně zahazovat ještě dobrý buben jen proto, že vám došel toner. Buben Crystal Jetu (75 \$) vydrží tisk asi 10 000, toner (60 \$) asi 5 000 stran. Životnost mechaniky je podobná jako u LaserJetu - asi 300 000 stran.

CrystalPrint používá rodkové kazety se sadami znaků (cena jedné 195 \$). Dodává se s pamětí 1,5M RAM. Zásobník toneru (119 \$) vystačí na 5 000 stran, ale buben (159 \$) jen na 8 000 stran. Provoz je tedy o dost dražší než u konkurenta.

Lze předpokládat, že dokud nebude tento typ tiskáren podstatně levnější a vybavenější pro spolupráci s řadou programů, nemůže počítat s příliš rychlým rozšířením. Firma Data Technology již ohlásila přípravu výroby zdokonaleného typu CrystalPrintu, který má stát 1995 \$.

Podle BYTE 4/88

-elzet-

PROFESIONÁLNÍ KLÁVESNICE

a V.24

pro $\mu\text{B-Pascal}$

Mikrobáze-Pascal je výbornou pomůckou pro první programátorské pokusy v Pascalu, ale i poměrně solidním nástrojem pro vývoj aplikačních programů. Velkým nedostatkem z hlediska produktivity práce i pohodlí obsluhy je původní gumová klávesnice Spectra. Při vývoji software pro jiné počítače je rovněž žádoucí, aby bylo možno přenášet data mezi Spectrem a cílovým počítačem. Proto jsem se rozhodl připojit ke Spectru profesionální klávesnici CONSUL (typ 259.11) a sériové rozhraní V.24 (smyčka 20/40 mA) a začlenit tato zařízení jako standardní pod $\mu\text{B-Pascal}$.

Popsané rozšíření je součástí $\mu\text{B-Pascalu}$. Umožňuje po zavedení programu jednorázově zvolit typ klávesnice (Spectrum/CONSUL), se kterou se bude nadále pracovat, a přes kanál K podporuje sériový přenos. Formát přenosu lze měnit na úrovni edice zdrojového textu driveru, rychlost přenosu lze měnit i průběžně pomocí "pouků". Uvedená verze sériového přenosu pracuje rychlostí 600 Bd s formátem 8 bitů bez parity, 1 stop bit.

Při práci s klávesnicí CONSUL je zachován způsob ovládání ISP a editoru téměř ve všech případech (viz popis funkčních kláves). Místo klávesy BREAK se používá klávesa CTRL C, pouze při sériové komunikaci a při práci s magnetofonem je nutno použít klávesu BREAK na Spectru, protože klávesnice je ošetřována pouze při povoleném přerušení. Navíc je přidána jedna funkce: klávesa CTRL B způsobí záměnu barev PAPER, BORDER a INK.

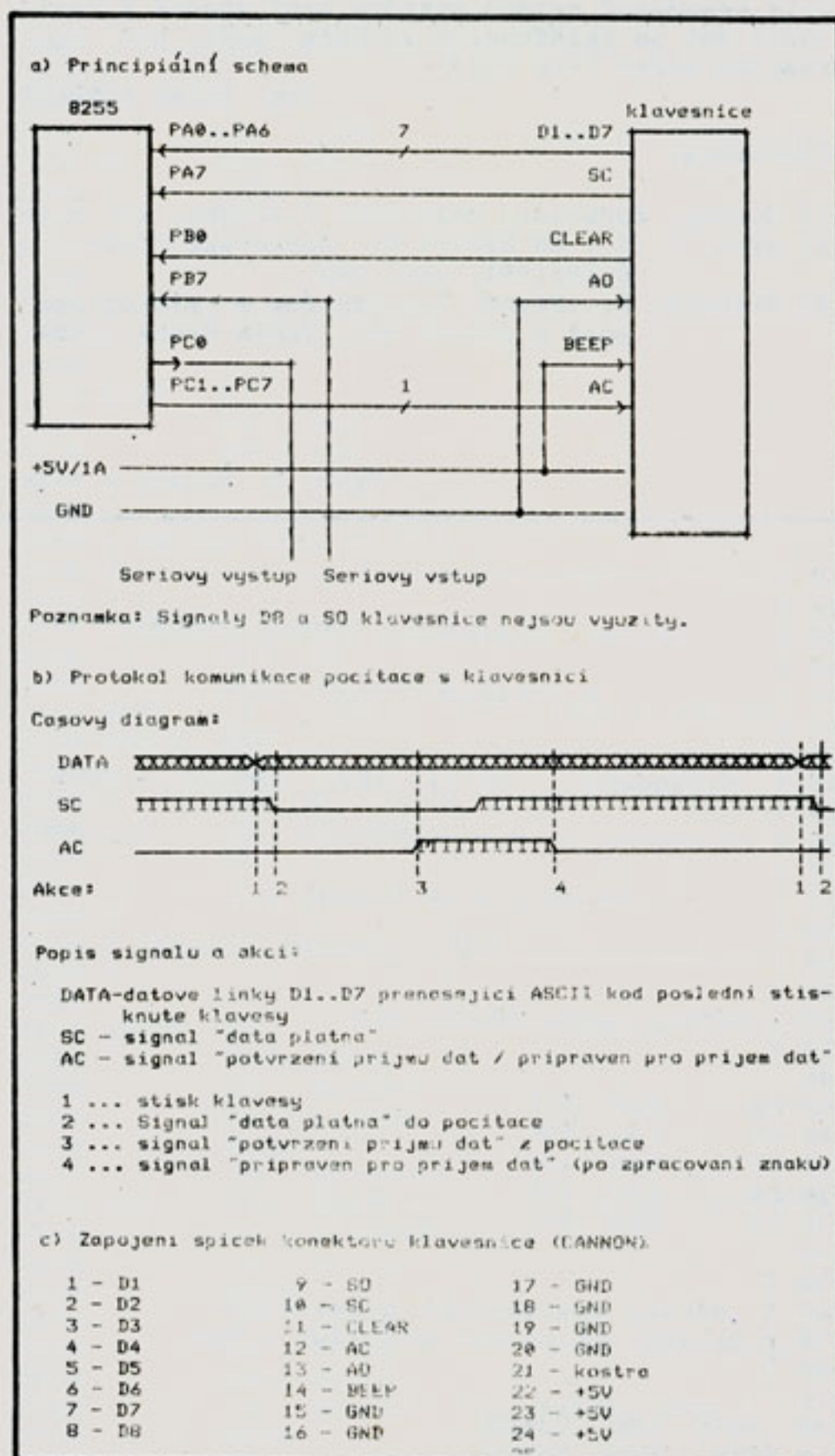
Pro připojení ke Spectru je využit interfejs s obvodem 8255, který je zřejmě mezi uživateli Spectra nejrozšířenější. Zapojení špiček konektoru klávesnice, schéma připojení klávesnice k interfejsu a protokol komunikace mezi klávesnicí a počítačem jsou na obr. 1. Výstupy PC1 až PC7 obvodu 8255 jsou spojeny a budí společně vstup AC klávesnice (I = 12 mA). Klávesnice je napájena přímo z interfejsu, všechny špičky +5 V a všechny špičky GND musí být propojeny na konektoru, protože uvnitř klávesnice nejsou spojeny.

Je-li sériový styk realizován přímo na úrovni TTL (např. mezi dvěma Spectry), není třeba připojovat žádné převodníky signálu pro vstup ani výstup. Je-li použita proudová smyčka V.24 resp. styk RS 232C, je třeba vybudovat příslušné převodníky pro vstup a výstup. Převodníky lze napájet přímo z interfejsu. Zapojení převodníku závisí na konkrétních požadavcích a bylo již několikrát publikováno na stránkách Amatérského rádia i Zpravodaje Mikrobáze, viz např. [1], [2].

Programové vybavení je vytvořeno v BASICu a Assembleru. Basicová část v zaváděči $\mu\text{B-Pascalu}$ (viz Výpis 1) zajišťuje volbu typu klávesnice a instalaci klávesnice CONSUL (byla-li zvolena) přepsáním adres vstupních bodů driveru klávesnice a úpravou instrukcí IM 1 na IM 2 v těle Pascalu.

V Assembleru jsou napsány dvě sekce (viz Výpis 2). První v délce 9 bajtů modifikuje inicializační sekci Pascalu a ve druhé o délce asi 240 bajtů je uložen driver klávesnice, driver sériového styku a podprogram pro inicializaci interfejsu. Tato sekce je uložena ve stránce vektoru přerušení, kterou autoři Pascalu ponechali nevyužitou. (Pro správnou činnost Spectra stačí naplnit pouze vektor na adrese #FEFF.) Toto řešení zajistí kompatibilitu se všemi pascalovskými programy, žádným způsobem neomezuje používání příkazů T, Y, U ISP a nezabírá použitelnou část paměti.

Při použití klávesnice CONSUL reaguje ISP a editor na funkční klávesy uvedené ve výpisu driveru (Výpis 2). Jde vesměs o klávesy generující kódy menší než #20, překódování do kódu Spectra provádí driver podle tabulky. Úpravou této tabulky je možné přizpůsobit ovládání Pascalu jinému typu klávesnice, případně svým specifickým požadavkům.



Obr. 1. Připojení klávesnice k interfejsu

Modifikaci $\mu\text{B-Pascalu}$ verze 2.4 provedete takto:

- 1) Zaveďte do paměti Assembler (např. Dr.MG), vložte editorem zdrojový text assemblerovské části. Program přeložte a uložte na pásku jako dva soubory, např. PRVNI.COM (adresa 62805, délka 9 bajtů) a DRUHY.COM (adresa 65024, délka 240 bajtů).
- 2) Inicializujte počítač, zadejte CLEAR 30000 a nahrajte příkazem MERGE "pascal" zavaděč $\mu\text{B-Pascalu}$. Zrušte řádek 25 a vložte sekvenci řádků podle Výpisu 1. Takto upravený zavaděč uložte na pásku příkazem SAVE "pascal" LINE 20.
- 3) Nahrajte z pásky příkazem LOAD "paskod" CODE tělo Pascalu. Modifikaci provedete nahrazením původního obsahu vybraných částí novým: LOAD "PRVNI.COM" CODE: LOAD "DRUHY.COM" CODE, kde PRVNI.COM a DRUHY.COM jsou soubory úprav vytvořené podle bodu 1). Takto upravený program nahrajte za zavaděč příkazem SAVE "paskod" CODE 40653,24882.

Používáte-li dosud $\mu\text{B-Pascal}$ verzi 2.3, je vhodné upravit jej na verzi 2.4 postupem, který zaslala Mikrobáze všem uživatelům $\mu\text{B-Pascalu}$.

Popsanou úpravu používám ke své naprosté spokojenosti již tři měsíce bez jakýchkoli problémů. Zkušenosti ukazují, že na rozdíl v ovládání Pascalu si lze zvyknout během několika minut a pohodlí práce s profesionální klávesnicí mnohonásobně vyváží námahu vynaloženou na úpravu. Možnost sériové komunikace ocení zřejmě především uživatelé, kteří využívají svůj počítač přímo na pracovišti, neboť časté přenášení celého systému není vhodné a 'sdělování dat po telefonních linkách není pro soukromníka možné (viz [3]).

Ing. Milan FINDURA

Literatura:

- [1] Modemy. Zpravodaj Mikrobáze č.4/1986, str.9-16.
- [2] Hyan, J.,T.: RS 232C-V.24. Amatérské rádio č.10/1984, str. 382.
- [3] Dlabola, F., Starý, J.: Systémy s mikroprocesory a přenos dat. NADAS Praha, 1984.

Výpis 1. Úpravy v zaváděči ISP (soubor "pascal")

```

22 CLS : PRINT AT 15,0: PRINT #1:"Instalovat klaves.CONSUL "
  [A/N]"
23 IF INKEY#="" THEN GO TO 23
25 IF INKEY#("<" AND INKEY#(">") THEN GO TO 45
30 POKE VAL "64565",VAL "21"
31 POKE VAL "64566",VAL "254"
32 POKE VAL "65300",VAL "25"
33 POKE VAL "65301",VAL "254"
34 POKE VAL "65303",VAL "68"
35 POKE VAL "65304",VAL "254"
36 POKE VAL "62869",VAL "94"
37 POKE VAL "64496",VAL "94"
38 POKE VAL "64510",VAL "94"
40 PRINT "Klavesnice CONSUL instalovana""
45 INPUT "": PRINT "Styk V.24 instalovan na kanale K":
  BEEP 2,-6
50 CLS : RANDOMIZE USR 62753

```

(Řádky 22 až 50 se vloží za řádku 20 v souboru "pascal" V2.4)

Výpis 2. Úpravy v těle ISP (soubor "paskod")

```

10 ;-----
20 ;
30 ; Driver klavesnice CONSUL 259.11
40 ;
50 ; a serioveho styku V.24
60 ;
70 ;
80 ; M.Findura 29.11.1987 rev.21.2.1988
90 ;
100 ;-----
110 ;
120 ; Verze pro uB-Pascal 2.4
130 ;
140 ;-----
150 ;
160 ; Adresy portu I8255, ULA
170 ;
180
190 PA EQU #1F
200 PB EQU #3F
210 PC EQU #5F
220 CW EQU #7F
230 PS EQU #FE
240
250 ;
260 ; Adresy systemovych promennych,
270 ; oblasti atributu, procedur v ROM
280 ;
290
300 LASTK EQU 23560
310 ATTR EQU 22528
320 ATTRP EQU #5C8D
330 BORDCR EQU #5C48
340 BREAK EQU #1F54
350
360 ;
370 ; Adresy v tabulce driveru - kanal K
380 ;
390
400 CHKOUT EQU #5CB6
410 CHKIN EQU #5CB8
420
430 ;-----
440
450 ORG #F555
460
470 ;
480 ; Zmena uvodni sekvence v uB-Pascalu
490 ; ( naplneni int.vektoru, inicializace )
500 ;
510
520 LD HL,#FDFD
530 LD (#FEFF),HL
540 CALL V24INI
550
560 ;-----
570
580 ORG #FE00
590
600 ;
610 ; Inicializace interface a driveru V.24

```

```

620 ;
630
640 V24INI LD A,#92
650 OUT (CW),A
660 LD A,1
670 OUT (PC),A
680 LD HL,OUTPUT
690 LD (CHKOUT),HL
700 LD HL,INPUT
710 LD (CHKIN),HL
720 RET
730
740 ;-----
750
760 ORG #
770
780 ;
790 ; Vstupni bod do driveru
800 ;
810 ; KBDL pro vstup z kanalu P ( po znacich )
820 ; KBD pro vstup z kanalu S ( po radcich )
830 ;
840
850 KBDL RES 5,(IY+1) ;minuly znak zrus
860 KBD IN A,(PA) ;vstup znaku
870 CPL
880 SUB #80
890 RET M ;zadny znak
900 LD B,A
910 LD A,#FF ;potvrzeni prijmu
920 OUT (PC),A
930 LD A,1
940 OUT (PC),A
950 LD A,B ;hledani v tabul.
960 LD HL,KTAB
970 LD BC,KTBL
980 PUSH BC
990 CPIR
1000 POP BC
1010 JR NZ,KB0 ;nenalezen
1020 ADD HL,BC ;prekodovani
1030 DEC HL
1040 LD A,(HL)
1050 KB0 LD (LASTK),A ;uchovani znaku
1060 SET 5,(IY+1)
1070 CP 2 ;Barvy (CTRL B)
1080 JR Z,BARVY
1090 RET
1100
1110 ;
1120 ; Test BREAK = (CTRL C)
1130 ;
1140
1150 KBDL LD A,(LASTK)
1160 CP #3
1170 RET Z
1180 SCF
1190 RET
1200
1210 ;
1220 ; Tabulka pro prekodovani znaku (CTRL x) na

```



```

1230 ; znaky akceptovane uB-Pascalem
1240 ;
1250 ; Obsahuje:
1260 ;
1270 ; CONSUL      uB+Pascal      Poznamka
1280 ;
1290 ;* BS          kurzor vzhuru
1300 ; LF          kurzor dolu      Nekoduje se
1310 ;* ESC        kurzor vlevo
1320 ;* FF        kurzor vpravo
1330 ; CR          ENTER          Nekoduje se
1340 ;* DEL        DELETE
1350 ;* DC2 (^R)   GRAPHICS
1360 ;* ^E         EDIT
1370 ;* ^A         "A (STOP)
1380 ;* ^D         "D (STEP)
1390 ;* ^F         "F (TO)
1400 ;* ^G         "G (THEN)
1410 ;* TAB (^I)  "I (AT)
1420 ;* ^S         "S (NOT)
1430 ;* DC4 (^T)  "Q (<=)
1440 ;* ^U         "U (OR)
1450 ;* ^W         "W (<>)
1460 ;* ^Z         "Y (AND)
1470 ;
1480 ; Dalsi funkcní klavesy zpracovava driver:
1490 ;
1500 ; ^C          BREAK
1510 ; ^B          inverze barev INK,PAP.,BOR.
1520 ;
1530 ; Poznamky:
1540 ; 1.Tabulka obsahuje pouze znaky oznacene *
1550 ; 2.Znak ^A znamena CTRL A na klav.CONSUL
1560 ; 3.Znak "A znamena Sym._shift A na Spectru
1570 ; 4.Znak "Y byl nahrazen ^Z z duvodu stejne
1580 ; polohy na klavesnici (QWERTY - QWERTZ)
1590 ;
1600 ; Znaky klavesnice CONSUL:
1610 ;
1620
1630 KTAB      DEFB #08,#1B,#0C
1640          DEFB #7F,#11,#05
1650          DEFB #01,#04,#06
1660          DEFB #07,#09,#13
1670          DEFB #12,#15,#17
1680          DEFB #1A
1690 KTBL      EQU #-KTAB'
1700
1710 ;
1720 ; Odpovidajici znaky uB-Pascalu
1730 ;
1740
1750          DEFB #0B,#08,#09
1760          DEFB #0C,#0F,#07
1770          DEFB #E2,#CD,#CC
1780          DEFB #CB,#AC,#C3
1790          DEFB #C7,#C5,#C9
1800          DEFB #C6
1810
1820 ;
1830 ; Zmena barev PAPER a INK po (CTRL B)
1840 ;
1850
1860 BARVY     LD   A,(ATTRP)
1870          OUT  (PS),A
1880          XOR  #3F
1890          LD   HL,ATTR
1900          LD   DE,ATTR+1
1910          LD   BC,768
1920          LD   (ATTRP),A
1930          LD   (HL),A
1940          LDIR
1950          LD   (BORDCR),A
1960          JR   KBD
1970
1980 ;-----
1990 ;
2000 ;          Driver styku V.24
2010 ;
2020 ; -format 8 bitu bez parity, rychlost 600 Bd
2030 ; -zmenu parametru lze provest pouze prepsa-
2040 ; nim zdrojového textu ( zbyva jen 16 byte)
2050 ;
2060 ; Vystup znaku na V24
2070 ;
2080
2090 OUTPUT    DI
2100          LD   C,A
2110          XOR  A          ;start bit
2120          CALL OUTBIT+1

```

```

2130          LD   B,B
2140 01       CALL OUTBIT      ;bity 0..7
2150          RRC  C
2160          DJNZ 01
2170          LD   C,1
2180          CALL OUTBIT      ;stop bit c.1
2190          EI
2200
2210 ;
2220 ;          Vlozeni cekaci smycky
2230 ; ( manipulacni, vzdy' )= 1 bit )
2240 ;
2250
2260          LD   B,#4          ;dalsi stop bity
2270 02       CALL WAIT
2280          DJNZ 02
2290          RET
2300
2310 ;
2320 ;          Chybove ukonceni (BREAK)
2330 ;
2340
2350 EXIT      EI
2360          RST  #0B
2370          DEFB #0C
2380
2390 ;
2400 ;          Vystup bitu na PC0
2410 ;
2420
2430 OUTBIT    LD   A,C
2440          AND  #1
2450          OUT  (PC),A
2460 WAIT      LD   D,237      ;237 = 600 Bd
2470 WAIT1     NOP
2480          NOP
2490          DEC  D
2500          JR   NZ,WAIT1
2510          RET
2520
2530 ;
2540 ;          Vstupni procedura V24
2550 ;
2560
2570 INPUT     DI          ;cekani na stav H
2580          CALL BREAK
2590          JR   NC,EXIT
2600          IN   A,(PB)
2610          RLCA
2620          JR   NC,INPUT
2630 IC0       CALL BREAK      ;cekaj start bit
2640          JR   NC,EXIT
2650          IN   A,(PB)
2660          RLCA
2670          JR   C,IC0
2680          LD   D,118        ;1/2 start bitu
2690          CALL WAIT1
2700          IN   A,(PB)      ;stale stav L ?
2710          RLCA
2720          JR   C,IC0
2730          LD   B,#80        ;cteni bitu 0..7
2740 IC1       CALL WAIT
2750          IN   A,(PB)
2760          RLCA
2770          RR   B
2780          JR   NC,IC1
2790          CALL WAIT      ;test stop bitu
2800          IN   A,(PB)
2810          RLCA
2820          JR   NC,INPUT    ;chyba=vse znovu
2830          EI
2840          LD   AB
2850          CP   26          ;test (EOF)
2860          RET  Z
2870          SCF
2880          RET
2890 ;-----
2900 ;
2910 ;          Velikosti konstant pro ruzne rychlosti
2920 ;
2930 ;          prvni konstanta: adresa #FEAD
2940 ;          druha konstanta: adresa #FECA
2950 ;
2960 ;          600 Bd : 237      118
2970 ;          1200 Bd : 117     58
2980 ;          2400 Bd : 57      28
2990 ;          4800 Bd : 27      13
3000 ;          9600 Bd : 12      6
3010 ;
3020 ;-----

```


HRÁTKY SE ZÁSOBNÍKEM

Z 80

K tomuto příspěvku mne podnítil článek Podprogram pro úklid registrů (Mikrobáze 2/88). Je v něm uvedena milá hříčka pro úschovu obsahu registrů do zásobníku a jejich opětné obnovení. Ale jen hříčka. S tvrzením redakční poznámky, že uvedený způsob není vzdálen skutečné praxi, nejsem nikterak zajedno.

Nejčastější případ, kdy je nutné uklidit obsahy všech registrů do bezpečí zásobníku, nastává při vstupu přerušeni do děje programu. Mnohdy je vhodné uschovat (a ev. znovu nastavit) i registry IX a IY, kterých si článek vůbec nevšimá. A vyžaduje-li to situace, je nutno uložit i obsahy čárkovaných registrů. Mimo přerušeni se v drtivé většině případů uschovávají průměrně dva až tři páry v jednom sledu. A dále:

* Pořadí uložení registrů bývá různé. Někdy rozhoduje, zda uloží registry ve sledu DE, BC, nebo BC, DE. To když s tím druhým párem ještě nějak někde v rutíně nakládám jako s prvním v pořadí.

* Velmi často při výstupu z rutiny vůbec nepotřebují, aby byl obsah některých registrů obnoven. Tento obvyklý případ přenosu parametrů je sice v závěru článku cudně zmíněn, ale autor už neuvádí, co z toho pro proklamovanou konstrukci plyne.

* Obsah některých uložených registrů může být adresou skoku z jakéhokoli místa rutiny (instrukcemi RET nebo POP HL, JP (HL) apod.).

Nemluvte už o takové specialitě, jakou je třeba změna adresy zásobníku. Autor článku i redakční poznámky mohou namítnout, že samozřejmě pro takové případy nelze jejich způsob použít. Jenže ony vyhovující případy se vyskytují tak řídce, že skoro nemá cenu o nich uvažovat. Stačí zahledět se do assemblerového výpisu jakéhokoli většího programu. Způsob prezentace obsahu článku může začínající programátory přechodně zmást. Absence instrukce (nebo automatiky) pro uchování obsahu všech registrů mikroprocesoru Z80 je jeho neřest - nezbyvá než se s ní smířit.

Při psaní zdrojového textu lze obejít nepříjemně se opakující zapisování instrukcí PUSH a POP definicemi makroinstrukcí. Třeba PUADX mi po překladu do paměti uloží strojový kód instrukcí PUSH AF, PUSH DE, PUSH IX. Pro "vzobnutí" obsahu těchto registrů ze zásobníku mohou definovat makro POXDA. A tak podobně. Kdo nemá možnost použít makra, tomu nezbyvá než instrukce vypsát celé. Tato prastará metoda je účelná, přehledná a hlavně nezanášá do zásobníku další "meziskoky", které mohou vyvolat větší zmatek než ev. překlep při zápisu. Ušetřených pár bajtů paměti na úkor přehlednosti programu (a nakonec i jeho rychlosti) staví celou věc do nepříliš příznivého světla. Chápu, že autor by rád našel způsob, jak se vyhnout zátěži hlídání správného zápisu instrukcí PUSH a POP. Já taky. Jenže...

Na doložení onoho "jenže..." je v závěru příspěvku rutina z rodu těch, kterým říkám kukaččí vejce. Znáte to - člověk je na posedu u počítače a úporně vyhlíží řešení ... až nakonec vysedí něco, co se ani jinak nazvat nedá. Připojená rutina je z rodu těžkotonážnějších. U nich je však zajímavá jedna věc. Přestože jsou pro praxi skoro nepoužitelné, bývají skvělými demonstračními pomůckami pro vysvětlování různých záhad. Jsou totiž nabitý tolikerým krasoumem, tak ornamentálně vykrouženy a rubensovsky vykynuty... Prostě čím absurdnější je výsledek, tím lépe na něm lze demonstrovat některé detaily, které u uhlazeně funkčních produktů nebývají tak patrné. I to potvrzuje, že absurdita má v životě své místo.

Pro orientaci začínajících experimentátorů napřed pár drobných ukázek a informací o práci se zásobníkem, resp. registrem SP. Jako k zásobníku vůbec, i k některým z uvedených ukázek je třeba přistupovat obezřetně a používat je jen v případech nouze nebo nezlomné jistoty.

1 Přenos reg.SP do reg.HL	LD HL,0 ADD HL,SP
2 Přenos reg.F do reg.A	PUSH AF POP HL LD A,L
3 Přenos reg.F do reg.H (obsah reg.L nepředpokládatelný)	PUSH AF DEC SP POP HL
4 Výměna reg.BC s reg.HL	PUSH BC EX HL,(SP) POP BC
5 Výměna reg.HL s reg.SP	EX DE,HL LD HL,0 ADD HL,SP EX DE,HL LD SP,HL EX DE,HL
6 Přechná změna adresy zásobníku (např.pro odběr dat od adr.ZAS2 instrukcemi POP; zde uvedeno ukládání přenášeného parametru přes reg.A a reg.HL instrukcí PUSH HL do paměti od adr.ZAS2 dolů)	LD (ZAS1),SP LD SP,(ZAS2) PRENOS CALL VYPOCET LD L,A CALL VYPOCET LD H,A PUSH HL ... JR NZ,PRENOS LD SP,(ZAS1)
7 Snížení spodku zásobníku o N adres (pro zvýšení +N)	LD HL,-N ADD HL,SP LD SP,HL
8 Odběr dat za instrukcí CALL (v HL bude 1.adresa za instr. CALL; zde na ní je 1.bajt slova AHOJ)	CALL SUBRUT DEFM "AHOJ" SUBRUT POP HL
9 Porovnání reg.SP s číslem NNNN	LD HL,0 ADD HL,SP LD DE,-NNNN ADD HL,DE

U všech operací se zásobníkem je nutno mít na paměti, že pod jeho spodní adresou bývá velmi čílo. Při vstupu do obslužných rutin přerušeni jsou adresy pod spodkem zásobníku přepisovány obsahy ukládaných registrů. Nechceme-li, aby v příkladu 6 při odběru dat (POP) byly bajty od adresy ZAS2 přepisovány tak, jak bude spodek zásobníku "vyjíždět nahoru", musíme maskovatelné přerušeni zablokovat instrukcí DI a navrch si ještě být jisti, že se po tu dobu neobjeví přerušeni nemaskovatelné (ev. přepnutí na stínovou romku). To se týká i příkladu 7 při zvyšování adresy spodku zásobníku. Proto např. bývá nemožné použít návratovou adresu z těchže dvou adres spodku zásobníku víckrát po sobě (třeba ve sledu RET...DEC SP, DEC SP, RET). Už napodruhé na oněch dvou adresách může být něco jiného. Následující ukázka přenosu parametrů mezi rutinami ukazuje, jak lze odebírat data ze zásobníku, aniž bychom je vystavili nebezpečí přepisu:

```
LD HL,-2 ;Rezerva pro uložení 2-bajt.
;výsledku
ADD SP,HL ;operace, kterou provede
```



```

ORG 50000

ZALOZ LD (PRGZAS), SP ;Uchova spodní adr.programového zás.
LD SP, (REGZAS) ;Spodní adr.registrového zás.do reg.SP
PUSH AF ;Uložení všech reg.do registr.zásobníku
PUSH BC
PUSH DE
PUSH HL
PUSH IX
PUSH IY

ZALOZO LD (REGZAS), SP ;Uchova nové adr.spodku registr.zás.
LD SP, (PRGZAS) ;Obnovení program.zásobníku
RET ;a návrat

LIMIT EQU 49999-11 ;Horní hranice registr.zás.-11 (konst.)
REGZAS DEFW LIMIT+11 ;" " " " (akt.prom.)
REG DEFM "YYXLHEDCBFA" ;Porovnávací řetězec pro hledání pozice
PRGZAS DEFS 2 ;Spodní adr.programového zásobníku

VYBER PUSH AF ;Uložení všech reg.do progr.zásobníku
PUSH BC
PUSH DE
PUSH HL
PUSH IX
PUSH IY
LD HL, 13 ;Zjištění návratové adresy (na ní leží
ADD HL, SP ;znak.kód prvního hledaného registru)
LD D, (HL) ;Přenos této adresy ze spodku reg.zás.
DEC HL ;do reg.DE
LD E, (HL)
DEC HL ;Adr.prvního uloř.reg.v progr.zásobníku
EX DE, HL ;do DE, v HL 1. adr.za instr.volání sub-
PUSH HL ;rutiny. Její uložení do progr.zás.
LD A, " " ;Test, zda je řetězec hledaných regis-
LD BC, 11 ;trů ukončen tečkou
CPIR ;Zkus jí najít
JP NZ, CHYBZ ;Když chybí, vyvolej chybové hlášení
HLEDEJ POP HL ;Obnov adr.s kódem hledaného registru
LD A, (HL) ;Kód uloř do reg.A
CP " " ;Je to už konec řetězce hled.registrů?
JR Z, VYSTUP ;Když AND, zakonči akci
INC HL ;Jinak jdi na adr.dalšího znak.kódu
PUSH HL ;a uloř ji do progr.zásobníku
LD HL, REG ;Do HL 1. adr.porovnávacího řetězce
LD BC, 11 ;V BC pak bude číslo pozice nalezeného
CPIR ;kódu "shora" (i pro reg.zásobník)
JP NZ, CHYBZ ;Když neznámý kód, chyba
LD HL, (REGZAS) ;Do HL aktuální adr.spodku reg.zásob.
LD A, C ;Číslo pozice kódu do reg.A
LD C, 11 ;Posun na adr.1.uloženého registru
ADD HL, BC ;v registr.zásobníku (je to "reg.A")
LD C, A ;Číslo pozice kódu do reg.C
SBC HL, BC ;a nalezení jeho adr.v registr.zásob.
PUSH DE ;Adr.1.uloř.reg.v progr.zás.do pr.zás.
EX DE, HL ;a do HL pro nalezení adr."stejnomen-
SBC HL, BC ;ného reg." v progr.zásobníku
EX DE, HL ;Nalezená adr.do DE
LD A, C ;Stanovení, zda se bude odebírat 1 nebo
CP 8 ;2 bajty (IX, IY)
LD C, 1
JR C, JEDENR ;Když jeden, skok na JEDENR
INC C ;Jinak zvýšit na 2
JEDENR LDDR ;Provedení přenosu z reg.do prog.zás.
POP DE ;Obnovení adr.1.uloř.reg.v prog.zás.
JR HLEDEJ ;Pokračuj pro další hled.kód (ev." ")
VYSTUP POP IY ;Vyložení všech registrů z progr.zás.
POP IX ;Hledané obsahy jsou již v požadova-
POP HL ;ných registrech
POP DE
POP BC
POP AF
RET ;Návrat

VYLOZ LD (PRGZAS), SP ;Viz subr.ZALOZ
LD SP, (REGZAS)
LD HL, 0 ;Hlídní, aby zásobník nepřekročil svou
ADD HL, SP ;stanovenou horní hranici
LD DE, -LIMIT
ADD HL, DE
JR C, CHYBL ;Když překročil, na chybové hlášení
POP IY ;Přenos z reg.zás.do všech registrů
POP IX
POP HL
POP DE
POP BC
POP AF
JP ZALOZO ;Skok na adr.ZALOZO

CHYBL CALL HLASKA ;Volej chybovou rutínu
CHBBAJ DEFB 0 ;Chyb.kód překročení horní hranice
CHYBZ CALL HLASKA ;Volej chybovou rutínu
CHBBYT DEFB 1 ;Kód chybného zadání řetězce hled.reg.

```



```

LD HL,(DVAB) ;subr.PARAM.
PUSH HL ;Do HL 2-bajt.parametr.
LD A,(JEDENB) ;Jeho uložení do zásobníku.
PUSH AF ;Totéž s 1-bajt.parametrem.
INC SP ;Protože obsah reg.F je teď
CALL PARAM ;v zás.nejniž a nezajímá nás,
;spodek zásobníku o 1 adr.
;výš. Volání subrutiny.
LD HL,3 ;Po návratu posun spodku zá-
ADD HL,SP ;sobníku o 3 adr.výš; na je-
LD SP,HL ;ho 2 spodních adr. teď leží
.... ;výsledek operace subrutiny.

PARAM LD HL,2 ;Adresa spodku zás.zvýšená o
ADD HL,SP ;2 (nad adr.návratu) je pře-
;vedena do reg.HL.
LD A,(HL) ;Do reg.A 1-bajt.parametr.
INC HL
LD C,(HL) ;Do reg.DE 2-bajt.parametr.
INC HL
LD B,(HL)
CALL VYPOCET ;Provedení operace*s 2-bajt.
;výsl. v DE.
INC HL
LD (HL),E ;Uložení výsledku do zásob-
;níku.
INC HL
LD (HL),D
RET ;Reg.SP je "na svém", návrat

```

Povšimněte si ještě určité výhody způsobu ukládání přijatých dat v příkladu 6. Pokud bychom nevyužili zásobník jako oblast pro ukládání dat, místo instrukcí LD L,A - LD H,A - PUSH HL bychom museli použít LD (HL),A - DEC HL - LD (HL),A - DEC HL. První sled trvá 19 taktů, druhý 26. Při ukládání většího počtu dat může tento rozdíl hrát svou roli (zvláště v grafice).

Nyní již ke kukaččí rutině. Část ZALOZ ukládá obsahy všech registrů do registrového zásobníku, což je předem vyhrazená volná oblast paměti. Subrutina VYLOZ zase všechny registry naplní hodnotami bajtů v tomto zásobníku uloženými. Část VYBER slouží pro vyhledání původní hodnoty libovolného 8-bitového registru A,F,B,C,D,E nebo 16-bitových IX,IY. Hledané registry zapíšeme do řetězce za instrukcí volání subrutiny VYBER. Např.:

```

SUBR CALL ZALOZ
....
CALL VYBER
HLEDR DEFM "ACFXE." ;Hledané registry A,C,F,IX,E
....
CALL VYLOZ
RET

```

Ze zadaného řetězce HLEDR subrutiny VYBER postupně odebírá ASCII kódy názvů hledaných registrů. Porovnáním získaného kódu určí číselnou pozici názvu registru v porovnávacím řetězci REG. Poziční číslo slouží k vypočtení adresy uložení obsahu onoho registru v registrovém i programovém zásobníku. Další operací je obsah určení adresy přenesen z registrového zásobníku do adresy uložení stejnojmenného registru v programovém zásobníku. Po vykonání tohoto sledu operací pro všechny zadané registry jsou nakonec jejich uložené hodnoty (i se změnami) přeneseny z programového zásobníku zpět do "svých" registrů.

V zadávaném řetězci nezáleží na pořadí, v jakém názvy registrů zapíšeme. Místo IX a IY se píše jen X a Y. Řetězec musí být vždy zakončen tečkou, podle níž rutina pozná konec řetězce. Když tečky chybí, následuje skok do chybové rutiny. Ta zde uvedena není, protože její podoba je závislá na soft/hardwareových podmínkách. V každém případě by se po jejím vyvolání měl program zastavit s patřičným hlášením a oba zásobníky by měly být nastaveny na své výchozí hodnoty. Chybová rutina je volána i v případě, kdy je v zadaném řetězci nesmyslný znak, nebo když by měl odběr překročit horní vymezenou hranici LIMIT+11 registrového zásobníku. Na adresu této hranice je na začátku inicializována proměnná (REGZAS), která pak vždy obsahuje aktuální adresu spodku registrového zásobníku.

Nevýhoda rutiny je zřejmá na první pohled - je dlouhá, jejím voláním se programový běh zpomaluje. Další mínus tkví v potřebě zavést chybová hlášení. Lze se obejít i bez nich, ale pak je třeba počítat s kolapsovými důsledky možných omylů. Pro případ pomnutí těchto hlášení je v rutině nutno zrušit vše, co s nimi souvisí. Pokud lze vůbec hovořit o výhodách, pak mezi ně patří možnost individuálního nastavování obsahu jednotlivých registrů na jejich původní hodnoty kdykoli v průběhu programu. Instrukce POP pracují vždy s celým párem, mění obsahy obou zúčastněných registrů, což se nám občas velice nehodí. Tuto výhodu lze uplatnit i při výstupu z jakékoli rutiny, např.:

```

SUBR CALL ZALOZ
....
CALL VYBER
HLEDR DEFM "FB."
RET

```

V tomto případě zůstanou subrutinou SUBR změněné obsahy reg.A a C nedotčeny (instrukce POP AF a POP BC na výstupu by je zničily). Reg.B tak můžeme pohodlně použít třeba pro prostorově rozsáhlejší cyklování pomocí instrukce DJNZ. Též lze samozřejmě dosáhnout i "klasickými" prostředky.

Díky přepínání hodnoty reg.SP na aktuální adresu spodku programového a registrového zásobníku může rutina pracovat i s větším počtem vrstev uložení registrů při prolínání různých subrutin programu. Zadaný řetězec za CALL VYBER se pak bude vždy vztahovat k poslední uložené vrstvě. Což by zase šlo upravit, aby...atd. Jenže...proč vlastně?

Zatímco u mnoha vyšších jazyků člověk nemá o zásobníku ani potuchy, každý "assembleurník" ví, že i když programuje s předem jasnou koncepcí, skoro jistě se v průběhu tvorby párkrát do zásobníku "zřítí". Následné hledání chyby bývá dost úmorné. Proto je dobré, když si na ni předem nezaděláváme zaváděním komplikovaných manipulací.

Zcela na závěr přiznávám, že "ornamentální" rutina není mé původní dílo. Pochází ze zažloutlého sešitku s nápisy: Bavory, L.P.1884 - 0 chovu do-bajtka - Rozkošné selce von Neumann věnuje Jára da C. Velmi zajímavá je autorova prorocká poznámka pod rutinou vpravo dole: Přiznejme si, že v zásobníku máme rezervy!

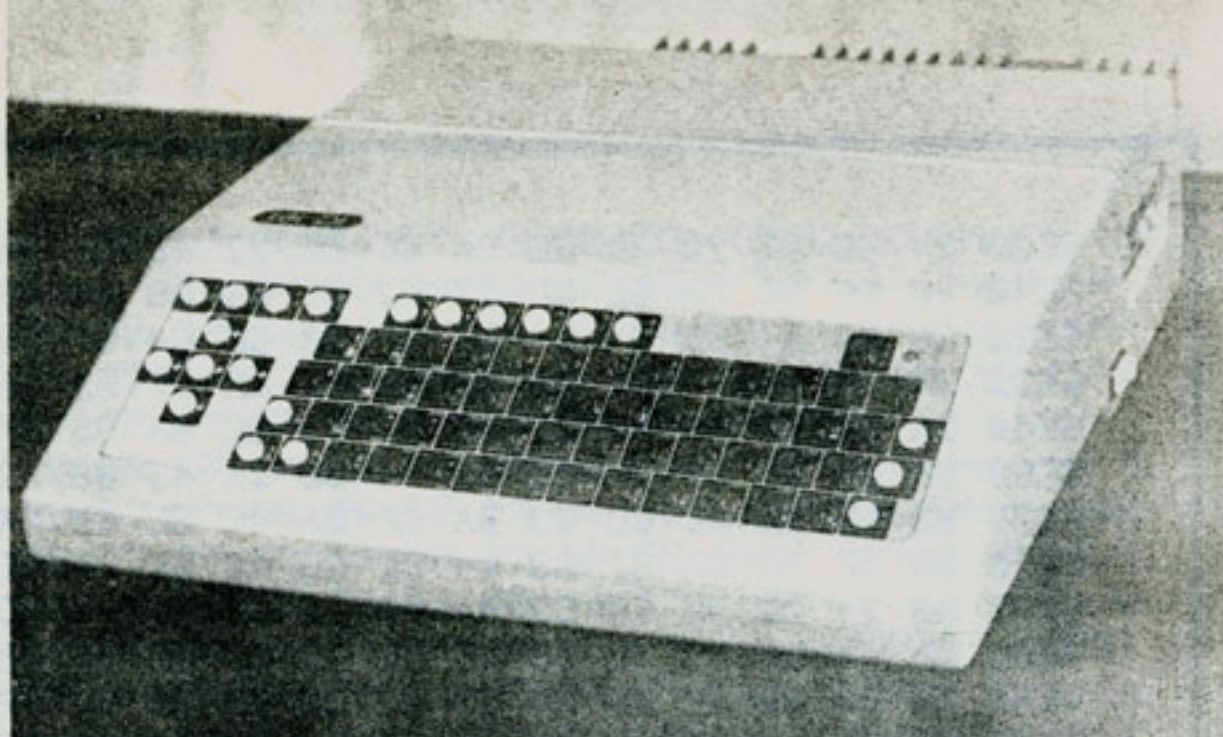
-elzet-

Opravte si, prosím,

tiskové chyby v článku Pár slov o μ B-Pascalu ve zpravodaji Mikrobáze 3/88:

1. Strana 21, levý sloupec, uprostřed vpravo, místo
C9 FB CB má být
C9 FD CB
2. Strana 21, pravý sloupec, uprostřed vlevo, místo
F500E má být
F50E
3. Strana 22, levý sloupec, uprostřed, místo
dpake (dpek má být
dpoke (dpeek
4. Strana 22, pravý sloupec, vlevo dole, místo
5B0B-4EFF-----6 má být
5B0B-EFF-----6

SOUBOR POMOCNÝCH PROGRAMŮ



k modulu *BASIC 6* mikropočítače *IQ 151*
Jiří Ježek

Soubor obsahuje tyto podprogramy:

1. zrychlené nahrávání programu v BASICu včetně podprogramů ve strojovém kódu, ,
2. rozšíření příkazů GOTO, GOSUB, RESTORE, LIST a RUN,
3. výpis programu na obrazovku od řádku do řádku,
4. mazání některých částí programu v BASICu,
5. přečíslování programových řádků BASICu,
6. výpis identifikátorů proměnných a definovaných funkcí použitých v určité části programu,
7. výpis programu ve strojovém kódu na MINIGRAF,
8. výpis hodnot proměnných v určité části programu.

Programy 2,3,6 a 7 nahráváme příkazem LOAD, ostatní pak v monitoru příkazem L.

1. Zrychlené nahrávání programu

Podprogram ve strojovém kódu přihrájeme k hotovému nebo k rozpracovanému programu v BASICu. Na obrazovce se zobrazí tečkami formát titulku, kterým svůj program označíme. Po vložení titulku se automaticky vloží programový řádek 50000 obsahující volání příslušného podprogramu pro nahrání na magnetofon. Program pak můžeme kdykoliv nahrát na kazetu příkazem

RUN 50000

Tímto způsobem se nahraje:

- a) titulek s pomocným podprogramem,
- b) podprogram ve strojovém kódu, pokud byl vložen do oblasti `USR`; oblast `USR` definujeme druhým parametrem v příkazu `CLEAR`, její začátek je na adresách `00A4-00A5 H`, konec na `00A6-00A7 H`,
- c) adresa počátku programového bufferu BASICu a vlastní program v BASICu.

Při zpětném přehrání takto nahraného programu z kazety do počítače se:

- a) zobrazí titulek programu,
- b) nahraje podprogram ve strojovém kódu, pokud byl uložen v oblasti `USR`,
- c) nahraje program v BASICu do bufferu, jehož počátek je na adrese `00CE-00CF`; počáteční adresa bufferu se mění připojením některých periférií (MINIGRAF, kreslič XY 4120); popisovaný podprogram přesouvá nahrávaný program automaticky,

- d) po nahrání celého programu se program automaticky spouští, jako bychom vložili z klávesnice `RUN`.

Kopírování podprogramu: W 5E00,5FA7,5F2D

2. Rozšíření příkazů GOTO, GOSUB, LIST, RESTORE a RUN

Následující podprogram nejdříve voláme příkazem

GOSUB 60000.

Tím se uloží na adresy 256 až 294 příslušný podprogram ve strojovém kódu a přiřadí se číselné hodnoty identifikátorům

RE, LI, GS, GT a RU.

V programu pak můžeme používat příkazy:

CALL RE, výraz ... místo RESTORE výraz,
CALL LI, výraz ... místo LIST výraz,
CALL GT, výraz ... místo GOTO výraz,
CALL GS, výraz ... místo GOSUB výraz,
CALL RU, výraz ... místo RUN výraz.

Příklad: Příkazem `CALL RE; 1980 + 50xN` nastavíme ukazatel `DATA` na řádek, jehož číslo dostaneme po vyčíslení výrazu, tedy 2030 pro `N=1`, 2080 pro `N=2` ap.

3. Výpis programu od čísla řádku do čísla řádku

Podprogram voláme příkazem `RUN 60100`.

4,5. Mazání a přečíslování řádků BASICu

RUN 60200 ... mazání
RUN 60300 ... přečíslování
Kopírování programu: W 7900, 7B98, 7906

6. Výpis identifikátorů proměnných

Lze zařadit na kterémkoliv místě programu, nebo po skončení programu voláme jako `GOSUB 60400` (!!!`RUN` ruší pole proměnných).

8. Výpis hodnot proměnných

Voláme na kterémkoliv místě programu jako `CALL HEX(7770)`. Podprogram vypíše identifikátor a hodnotu číselné i textové proměnné, pokud byla v proběhlé části použita. Výpis lze zastavit tlačítkem `CTRL`, stiskem libovolného tlačítka ve výpisu pokračujeme.

1

```

5E00 21 56 02 22 1B 00 2A CE 00 22 4E 5E 21 00 5E 11 !V * *N *N*! *
5E10 2C 5F 01 78 5E CD 67 F2 3E 04 CD A5 F5 2A A6 00 ,_ x^Mgr> MXu#&
5E20 EB 2A A4 00 01 81 5E CD B8 CE C4 67 F2 CD 88 5E k*$ ^M8NDgrM ^
5E30 21 A0 00 11 D1 00 01 68 5E CD 67 F2 2A D0 00 EB ! 0 h^Mgr*P k
5E40 2A CE 00 01 50 5E CD 67 F2 3E 8A D3 87 C9 00 00 *N P^Mgr> S I
5E50 3E 8A D3 87 3E 02 32 14 00 2A 42 00 22 CE 00 CD > S > 2 *B *N M
5E60 88 5E CD 7C CD C3 DA CF 2A 4E 5E EB 2A 42 00 7D ^MIMCZO*N*k*B )
5E70 93 6F 7C 9A 67 C3 84 5E CD AB 5E 2A CE 00 22 42 ol gC *M+*N *B
5E80 00 21 00 00 E5 C3 BA F3 2A CE 00 54 5D 7E 23 B6 ! eC:s*N T]~#6
5E90 CA A4 5E 23 23 23 AF BE 23 C2 97 5E EB 73 23 72 J$###/>#B ^ks#r
5EA0 EB C3 8B 5E EB 23 23 22 D0 00 C9 CD 47 F6 1F 21 kC ^k##^P IMGv !
5EB0 03 0A 22 0E 00 3E 01 32 14 00 21 DD 5E CD 88 F4 ' > 2 !]M t
5EC0 CD D2 5E 21 E0 5E CD 88 F4 CD D2 5E 21 08 5F C3 MR^!'M tMR^! _C
5ED0 88 F4 06 16 CD 47 F6 51 05 C2 D4 5E C9 0F 4D D0 t MGv0 BT^I MP
5EE0 5F 0D 20 20 20 20 0F 54 0E 2E 2E 2E 2E 2E 2E - T .....
5EF0 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E .....
5F00 54 0D 20 20 20 20 0F CB 4A 0D 0D 0D 2A 2E 2E 2E T KJ *...
5F10 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E .....
5F20 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2A 8D CD AB 5E .....* M+^
5F30 21 05 0B 22 0E 00 CD B4 F5 21 E9 5E 06 16 CD 86 ! ^ M4u!i^ M
5F40 5F 21 01 0F 22 0E 00 CD B4 F5 21 0D 5F 06 1E CD _! ^ M4u! _ M
5F50 86 5F 3E 02 32 14 00 CD 47 F6 0D CD 47 F6 0D CD _> 2 MGv MGv M
5F60 88 5E 21 7C 5F 11 85 5F 01 50 00 CD 47 F2 11 50 ^!I_ _ P MGr P
5F70 C3 01 0E 00 21 4F 00 79 A7 C3 ED CC A0 DB 35 45 C !0 y^CmL [5E
5F80 30 30 29 3A 80 00 CD AA F8 FE 08 CA 96 5F FE 20 00): M*x~ J _~
5F90 DA 86 5F C3 9E 5F 2B 04 CD 03 F0 C3 86 5F 77 23 Z _C _+ M pC _w#
5FA0 CD 03 F0 05 C2 86 5F C9 FF FF FF FF FF FF FF M P B _I

```

2

```

60000 REM PRIKAZ, VYRAZ
60002 DATA225, 195, 129, 209, 235, 205, 128, 205, 235, 205, 144, 209, 195, 218
60004 DATA207, 205, 41, 204, 3, 193, 193, 229, 42, 194, 0, 227, 229, 42
60006 DATA198, 0, 227, 62, 140, 245, 51, 197, 195, 129, 209
60008 RESTORE60002:FORI=256T0294
60010 READA:POKEI,A:NEXT
60012 RE=53307:LI=53045:GT=256:RU=260:CS=271
60014 RETURN

```

3

```

60100 REM LIST 00-00
60102 CLEAR40,82
60104 RESTORE60124
60106 FORI=32592T032669
60108 READA:POKEI,A:NEXT
60110 CLS:PRINT&14,0^Vypis programu od radku cislo^:INPUTCR
60112 PO=WORD(32592,CR)
60114 INPUT^do radku^,PR
60116 PK=WORD(32592,PR):CLS
60118 IFPO>=PKTHENEND
60120 PO=WORD(32598,PO)
60122 GOTO60118
60124 DATA205,82,205,96,105,201,235,205,200,213,193,120,177,202,144,204,205
60126 DATA85,200,197,205,176,210,205,200,213,227,205,23,226,62,32,225,205
60128 DATA191,206,126,183,35,202,156,127,242,113,127,214,127,79,229,17,215
60130 DATA200,213,26,19,183,242,133,127,13,225,194,132,127,126,183,250,112
60132 DATA127,205,191,206,35,195,144,127,225,201

```


4 5

7900	C3	25	79	C3	00	7B	C3	4E	7B	CD	88	F4	21	19	7B	CD	CXyC (CN(M t! (M
7910	88	F4	CD	96	7A	22	99	7B	21	23	7B	CD	88	F4	CD	96	tM z' (!*(M tM
7920	7A	22	9B	7B	C9	21	50	00	22	D6	7A	21	3C	7B	CD	09	z' (I!P "Vz!<(M
7930	79	21	2C	7B	CD	88	F4	CD	96	7A	22	9D	7B	21	37	7B	y!, (M tM z' (!7(
7940	CD	88	F4	CD	96	7A	22	9F	7B	11	00	02	2A	D0	00	19	M tM z' (*P
7950	22	A3	7B	22	A5	7B	2A	9B	7B	EB	2A	99	7B	CD	A0	F4	"#('X(* (k* (M t
7960	DA	C3	79	EB	CD	B6	79	CD	52	CD	D5	E5	7E	23	B6	C2	ZCykM6yMRMUe~#6B
7970	78	79	21	FF	FF	C3	7D	79	23	5E	23	56	EB	22	99	7B	xy! C)y#'#Vk' (
7980	E1	50	59	13	13	13	13	01	04	00	E5	21	50	00	1A	77	aPY e!P w
7990	23	13	0C	E3	CD	B8	CE	E3	C2	8E	79	E1	D1	C5	CD	A1	# cM8NcB yaQEM!
79A0	7A	2A	9D	7B	EB	CD	B6	79	2A	9F	7B	19	22	9D	7B	E1	z* (kM6y* (" (a
79B0	CD	BB	7A	C3	56	79	E5	2A	A5	7B	73	23	72	23	22	A5	M; zCVye*X(s#r#*X
79C0	7B	E1	C9	2A	CE	00	7E	23	B6	CA	B5	7A	23	23	23	7E	(aI*N ~#6J5z###~
79D0	23	FE	87	D4	DD	79	A7	C2	CF	79	C3	C6	79	FE	98	CA	#~ Tly'BOyCFy~ J
79E0	F7	79	FE	99	CA	F7	79	FE	91	CA	79	7A	FE	B6	CA	72	wy~ Jwy~ Jyz~6Jr
79F0	7A	FE	8D	D0	FE	8A	C8	CD	1B	D0	D0	E5	CD	F2	D0	42	z~ P~ HM PPeMrPB
7A00	4B	22	A7	7B	2A	A5	7B	EB	2A	A3	7B	CD	B8	CE	CA	6B	K' '(X(k*#(M8NJK
7A10	7A	D5	5E	23	56	23	E5	60	69	CD	B8	CE	E1	5E	23	56	zU'#V#e' iM8Na'#V
7A20	23	EB	22	A9	7B	EB	D1	C2	0B	7A	E1	E5	44	4D	2A	A7	#k')(kQB zaeDM*
7A30	7B	EB	CD	A5	7A	3E	01	32	A0	00	2A	A9	7B	CD	17	E2	(kMxz> 2 *) (M b
7A40	21	DE	00	01	FF	00	7E	0C	23	FE	00	C2	46	7A	D1	D5	!~ ~ #~ BFzQU
7A50	2A	D0	00	E5	09	C1	22	D0	00	CD	18	CC	11	DE	00	E1	*P e A'P M L ' a
7A60	1A	A7	CA	6F	7A	77	23	13	C3	60	7A	E1	2A	A7	7B	2B	'Jozw# C'zaz*(+
7A70	7E	C9	CD	1B	D0	D0	C3	F7	79	7E	23	FE	88	CA	85	7A	~IM PPCwy~#~ J z
7A80	FE	8C	C2	79	7A	CD	F7	79	23	7E	FE	2C	23	CA	85	7A	~ ByzMwy#~~, #J z
7A90	FE	20	CA	89	7A	C9	CD	5D	CE	CD	1A	D0	CD	F2	D0	EB	~ J zIMJNM PMrPk
7AA0	C9	CD	52	CD	EB	2A	D0	00	1A	02	03	13	CD	B8	CE	D2	IMRMk*P M8NR
7AB0	A8	7A	C3	E0	7A	CD	E0	7A	C3	D6	CA	D5	E5	CD	52	CD	(zC'zM'zCVJUeMRM
7AC0	50	59	2A	D0	00	44	4D	E1	09	CD	18	CC	EB	74	23	23	PY*P DMa M Lkt##
7AD0	D1	73	23	72	23	11	7B	7B	1A	77	23	13	B7	C2	D8	7A	Qs#r# ((w# 7BXz
7AE0	2A	CE	00	54	5D	7E	23	B6	23	C2	F0	7A	22	D0	00	C9	*N Tj~#8#Bpz'P I
7AF0	23	23	AF	BE	23	C2	F3	7A	EB	73	23	72	EB	C3	E3	7A	##/>#Bszks#rkCcz
7B00	21	48	7B	CD	09	79	EB	CD	52	CD	E5	2A	99	7B	EB	CD	!H(M ykMRMe* (kM
7B10	52	CD	D1	CD	A5	7A	C3	D6	CA	20	6F	64	20	72	61	64	RMQMxzCVJ od rad
7B20	6B	75	A0	64	6F	20	72	61	64	6B	75	A0	4E	6F	76	65	ku do radku Nove
7B30	20	63	69	73	6C	6F	A0	4B	72	6F	6B	A0	50	72	65	63	cislo Krok Prec
7B40	69	73	6C	6F	76	61	6E	E9	4D	61	7A	61	6E	E9	3E	8A	islovaniMazani>
7B50	D3	87	21	8A	7B	22	D6	7A	11	28	EB	CD	70	7B	21	7B	S ! ('Vz (kMp(! (
7B60	7B	22	D6	7A	11	8C	EB	CD	70	7B	11	28	EB	C3	35	CF	('Vz kMp((kC50
7B70	D5	CD	A1	7A	D1	21	13	00	C3	BB	7A	A0	DB	37	39	30	UM!z0! C;z [790
7B80	30	29	3A	80	3A	13	50	52	45	00	A0	DB	37	39	30	33	0): : PRE [7903
7B90	29	3A	80	3A	13	4D	41	5A	00	FF	FF	FF	FF	FF	FF	FF): : MAZ

6

```

60400 REM VYPIS IDENTIFIKATORU PROMENNYCH
60402 RESTORE60412
60404 FORQ9=HEX(5CF0)TOHEX(5DBC)
60406 READQA:POKEQ9,QA:NEXT
60408 CALLHEX(5CF0)
60410 RETURN
60412 DATA42,210,0,43,235,42,208,0,205,160,244,218,11,93,205,105
60414 DATA93,1,5,0,9,205,151,93,195,248,92,205,176,245,42,212
60416 DATA0,43,235,35,205,160,244,218,176,245,205,105,93,205,71,248
60418 DATA40,35,78,35,70,35,197,213,229,78,6,0,9,9,86,43
60420 DATA94,43,229,197,27,58,189,93,187,202,73,93,235,43,205,105
60422 DATA93,62,1,50,189,93,195,76,93,205,24,226,193,225,13,202
60424 DATA91,93,197,205,71,246,44,193,195,46,93,205,71,246,41,225
60426 DATA209,193,9,205,173,93,195,20,93,70,35,78,121,230,128,50
60428 DATA189,93,202,131,93,197,205,71,246,70,205,71,246,78,193,121
60430 DATA230,127,79,205,7,240,120,167,200,230,127,205,3,240,120,230
60432 DATA128,200,205,71,246,36,201,50,14,0,254,28,210,176,245,230
60434 DATA252,198,4,79,62,31,161,50,14,0,195,180,245,58,14,0
60436 DATA254,16,210,176,245,62,16,50,14,0,195,180,245

```


7

```

60600 REM DISP NA MINIGRAF
60602 IFPEEK(HEX(C000))<>62THENPRINT"NENI PRIPOJEN MINIGRAF" : END
60604 CLEAR160,145
60606 RESTORE60670
60608 FORI=HEX(7F10)TOHEX(7F93)
60610 READA:POKEI,A:NEXT
60612 POKE20,2:POKE19,30:CLS
60614 PRINT&0,0"ORG          Q" :PRINT"Text          I"
60616 PRINT"Odradkovani Q" :PRINT"Vypis          V"
60618 GOSUB60662:A$=INKEY$:IFA$=""THEN60618
60620 WAIT(4):IFA$="O"THENGOSUB60660:GOTO60618
60622 IFA$<>"D"THEN60626
60624 CALLHEX(7F3A):GOTO60618
60626 IFA$="T"THENINPUTA$:WRITEA$:PRINT&0,0;SPC(80):GOTO60624
60628 IFA$<>"V"THEN60618
60630 CLS:PRINT&14,0"Vypis str.kodu od radku cislo":PA=WORD(HEX(F4C3))
60632 PRINT"Do radku cislo ";KA=WORD(HEX(F4C3))
60634 POKE20,1:IFPA>KATHEN60612
60636 GOSUB60662
60638 PA=WORD(HEX(7F56),PA):PRINT
60640 CALLHEX(7F10)
60642 IFPEEK(0)<67THEN60634
60644 PRINT&0,0"ZALOZ NOVY PAPIR !"
60646 FORJ=1TO10:PRINTCHR$(7);:NEXT:CALLHEX(F8AA)
60648 GOSUB60660
60650 PRINT&0,0;SPC(28)
60652 GOTO60634
60660 ORG50,2130:POKE389,20:POKE0,1:RETURN
60662 POKE19,30:PRINT&26,0;PEEK(0)&0,0;:RETURN
60670 DATA42,106,1,236,33,0,236,62,1,50,115,1,205,90,208,126
60672 DATA254,13,202,58,127,230,96,202,43,127,126,205,63,194,126,183
60674 DATA1,253,255,252,105,196,35,195,28,127,229,213,1,224,255,17
60676 DATA50,0,42,112,1,9,68,77,205,0,192,33,0,0,52,209
60678 DATA225,175,50,115,1,201,235,6,16,205,250,245,205,167,246,32
60680 DATA126,205,255,245,35,5,194,92,127,229,125,214,16,111,124,38
60682 DATA0,156,103,205,71,246,32,205,71,246,32,6,16,126,230,127
60684 DATA254,32,210,135,127,62,32,79,205,9,248,35,5,194,125,127
60686 DATA225,195,176,245

```

8

```

7770 2A D0 00 22 AF 78 2A D2 00 2B EB 2A AF 78 CD A0 *P "/x*R +k*/xM
7780 F4 DA 93 77 CD 70 78 22 AF 78 CD A6 78 CD 4E 78 tZ wMpx"/xM&xMNx
7790 C3 76 77 2A D2 00 22 AD 78 21 B7 78 22 B1 78 2A Cvw*R "-x!7x"1x*
77A0 D4 00 2B EB 2A AD 78 CD A0 F4 D8 CD 70 78 4E 23 T +k*-xM tXMpxN#
77B0 46 23 7E 32 B6 78 E5 09 22 AD 78 E1 4E 06 00 09 F#~26xe "-xaN
77C0 09 23 22 AF 78 2B 2B 5E 2B E5 2A B1 78 73 23 73 #"/x++"+e*1xs#s
77D0 23 36 00 23 22 B1 78 E1 3D C2 C6 77 21 B7 78 22 #6 #"1xa=BFw!7x"
77E0 B3 78 3A B6 78 32 B5 78 CD A6 78 CD 47 F6 08 CD 3x:6x25xM&xMGv M
77F0 47 F6 28 2A B3 78 23 23 5E 23 22 B3 78 16 00 CD Gv(*3x##"#"3x M
7800 18 E2 21 B5 78 35 CA 10 78 CD 47 F6 2C C3 F3 77 b!5x5J xMGv,Csv
7810 CD 47 F6 29 CD 47 F6 3D CD 4E 78 2A B1 78 2B 34 MGv)MGv=MNx*1x+4
7820 2B 35 C2 DC 77 2B 2B 11 B7 78 CD B8 CE DA 99 77 +5B\w++ 7xM8NZ w
7830 34 2B 35 CA 25 78 23 23 7E 23 77 23 36 00 23 E5 4+5JXx##~#w#6 #e
7840 2A B1 78 EB E1 CD B8 CE C2 38 78 C3 DC 77 2A AF *1xkaM8NB8xC\w#/
7850 78 EB 3A AC 78 32 A2 00 CD 76 D5 21 60 78 CD 7F xk:',x2" MvU!"xM
7860 D2 00 CD 55 D0 2A AF 78 01 04 00 09 22 AF 78 C9 R MUP*/x "/xI
7870 AF 32 AC 78 11 A2 78 D5 11 C8 78 46 23 4E 23 79 /2,x "xU FxF#N#y
7880 E6 80 CA 8A 78 E1 E1 C3 99 77 79 12 13 78 A7 C8 f J xaaC wy x"H
7890 E6 7F 12 13 78 E6 80 C8 3E 24 12 13 3E 01 32 AC f xf H>$ > 2,
78A0 78 C9 3E BD 12 C9 21 C6 78 C3 88 F4 FF FF FF FF xI>= I!FxC t

```


v BASICu (Atari) [10] Bez možnosti výběru - popis jazyka Pascal pro Spectrum [11] Commodore a RS-232C - popis vlastního programování 1.část [13] Porucha disketové jednotky 1541 - jak opravit buffer 74LS14 (Commodore) [13] 3*Video - jak získat mód 80 znaků (Commodore) [15] Hardcopy pro C-16/116 a Plus/4 - program [15] Spectrum+3 - test [20] Amstrad PC1640ECD - test [20] Jak budou vypadat za několik let auta [21] Interpret, kompilátor, assembler (2) - překlad programů na assembler [22] Továrna v roce 2087 [31]

BAJTEK (PL)

10/87

Rozhovor s Markem Bilinskim [3] Jazyk Forth - základy programování [4] BASIC XL - popis [6] Tajemství Atari - 3 část [7] PIA - Atari [7] Data báze - popis programů pro Atari [8] Popis jazyka Pascal 2 část (Spectrum) [9] Commodore 16 a 116 - popis [10] Char a Rdot - program ukazující jak používat tyto příkazy (Commodore) [11] Příkazy Sshape a Gshape - popis (C-128/16/116/+4) [12] DOS+ pro C-64 - program zavádějící 2 nové příkazy [12] Firma - hra pro Amstrada [14] Interpret, kompilátor, assembler - 3 část [20] Nové počítače firmy IBM - Personal System/2 [21] O rozvoji kosmonautiky [31]

BAJTEK (PL)

11/87

Rozhovor s Boguslawem Radziszewskim a Krzysztofem Gajewskim [3] Tajemství Atari - 4 část [4] BASIC XE - popis [5] Disk Protector - program na ochranu diskety pro Commodore [7] Regulace hlavy v Data-sette - program pro Commodore [8] Ještě jednou o Turbu pro C-16 - úprava programu z Bajtku č.5/87 [9] Bez výběru - 3 část [10] Ukrytý assembler - jak pracovat s assemblerem na Spectru [12] Sledování práce programu - program pro Spectrum [12] Učíme mluvit CPC - program pro Amstrad [13] Star NX-15 - test [14] Jak získat majetek - s jakými problémy se můžeme setkat při programování [21] Různé názory na kosmický prostor [31]

BAJTEK (PL)

12/87

Rozhovor s Andrzejem Pagowskim [3] Využití počítačů v medicíně [4] Program v jazyce Logo, který předvádí rekurzi [5] Triton - test [6] Jehlou na papíře - co umí tiskárna [7] Program pro ZX Spectrum, který umožňuje tvorbu vlastního písma [8] 9888 - program pro Spectrum, který ulehčí edici textů [9] Microsoft BASIC - popis jazyka BASIC pro Atari [10] Katalog - program pro Amstrada [12] Super Expander Plus - popis jazyka pro Commodore [14] Drago Basic - grafický program pro Commodore [15] Počítač v kanceláři konstruktéra - popis systému CAD [20] Co je operační systém [22] Jsem skeptikem - rozhovor s Ivanem Malcem [23] Mars po "Fobosu" - Poletí člověk na Mars? [31]

BAJTEK (PL)

1/88

Rozhovor s Ryszardem Tateusiewiczem [3] Využití mikropočítačů ve včelařství [4] Jak vylistovat zablokované programy [6] TOS - popis disketové jednotky pro ZX Spectrum [6] Jak kreslit - (ZX Spectrum) [8] Weritest - program pro kontrolní součty (Commodore) [9] Turbo 64 Board - jak urychlit práci Commodore 64 [9] Zavádění funkcí pro C-64 - použití příkaz INPUT [10] Jak přenášet obrazovky mezi různými typy Commodore [11] Editor BASICu - program pro kontrolní součty (Atari) [12] Tajemství Atari - 5 část [12] Porovnání BASICů pro Atari [13] Program pro práci v několika grafických módech najednou (Amstrad) [14] Program pro kontrolní součty (Amstrad) [14] Bitimage - program umožňující grafické zobrazení strojového kódu (Amstrad) [15] Operační systém - 2. část [20] Cartridge - popis [21] Obsah BAJTKU 1986/1987 [29] O družicové televizi [32]

BAJTEK (PL)

2/88

Rozhovor s Krzysztofem Sadowskim [3] Připojení joysticku - Atari [7] Tajemství Atari - 6 část [7] Blikající kurzor - programy pro Atari [8] Útok - hra pro Atari [9] Polské písmena - Commodore [10] Jak rozšířit interpret - Warsaw Basic (Commodore) [11] Popis nové verze C-64 [12] Jak zachraňovat programy (ZX Spectrum) [13] Počítač a telefon - počítače v dopravě [13] Od vevnitř - 2 část [14] Disketa 5 1/4 pro CPC 6128 - návod montáže [21] Operační systém - 3 část [22] Buffer - popis [23] Tiskárna Amstrad DMP 3160 - test [24] Invaze magnetoskopů [32]

Computing Reviews (US)

1/87

Abecední rejstřík autorů [2] Popis dodatků k třídícímu systému časopisu Computing Reviews z let 1983-1987 [5] Třídící systém časopisu Computing Reviews, verze 1987 [7] Souhrn dodatků k třídícímu systému časopisu Computing Reviews z let 1983-1987 [9] Dvě nejvyšší úrovně třídícího systému časopisu Computing Reviews [10] Úplné schéma třídícího systému časopisu Computing Reviews, verze 1987 [11] Rejstřík klasifikačních názvů třídícího systému, verze 1987 [21] Formulář pro recenzenty literatury [47] Knihy a přednášky - recenze [49] Vstupní a výstupní obvody a datová komunikace [49] Navrhování programu [49] Programovací jazyky [52] Různé problematiky v oboru programování [52] Datové struktury [52] Zobrazení dat v paměti [56] Analýzy algoritmů a problému složitosti [56] Modely a principy informačních systémů [56] Spravování databází [56] Aplikace informačních systémů [56] Algebraické manipulace [57] Aplikace počítačů v umění a literatuře [59] Průmysl výpočetní techniky [59] Počítače a společnost [59] Právní aspekty výpočetní techniky [59] Řízení výpočetních postupů a informačních systémů [59] Neknižní literatura - recenze [60] Návrh logických obvodů [60] Integrované obvody [60] Architektury procesorů [60] Počítačové - sdělovací sítě [60] Výkonnost systémů [61] Implementace počítačových systémů [62] Navrhování programu [62] Operační systémy [63] Datové struktury [63] Zobrazení dat v paměti [64] Výpočetní postupy na abstraktních zařízeních [64] Analýzy algoritmů a problému složitosti [64] Matematická logika a formální jazyky [65] Diskrétní matematika [65] Pravděpodobnost a statistika [65] Modely a principy informačních systémů [65] Spravování databází [66] Záznam a čtení informací [66] Aplikace informačních systémů [68] Umělá inteligence [68] Počítačová grafika [70] Zpracování obrazu [70] Rozpoznání tvaru, obrazu [71] Simulace a modelování [71] Zpracování hromadných dat [71] Aplikace počítačů v medicíně [71] Průmysl výpočetní techniky [72] Počítače a vzdělání [72] Řízení výpočetních postupů a informačních systémů [72] Rejstřík podle termínů [72]

Computing Reviews (US)

2/87

Abecední rejstřík autorů [74] Knihy a přednášky - recenze [75] Obecná tematika v oboru výpočetní techniky [75] Vstupní a výstupní obvody a datová komunikace [75] Návrh logických obvodů [76] Integrované obvody [76] Různá témata o hardware [76] Počítačové - sdělovací sítě [76] Implementace počítačových systémů [78] Různé problematiky počítačových systémů [79] Navrhování programu [79] Programovací jazyky [80] Operační systémy [81] Různé problematiky v oboru programování [81] Datové struktury [81] Obecné teorie výpočetních postupů [81] Matematická logika a formální jazyky [82] Numerické analýzy [83] Pravděpodobnost a statistika [84] Modely a principy informačních systémů [84] Spravování databází [84] Aplikace informačních systémů [84] Umělá inteligence [85] Počítačová grafika [86] Zpracování obrazu [88] Rozpoznání tvaru, obrazu [88] Obecné problematiky v aplikacích počítačů [88] Aplikace počítačů ve fyzice a konstruování [88] Využití počítačů ve výrobě [88] Obecná problematika výpočtů [89] Počítače a společnost [90] Řízení výpočetních postupů a informačních systémů [91] Vlastní

výpočet [91] Neknižní literatura - recenze [92] Úvodní stať a přehled [92] Vstupní a výstupní obvody a datová komunikace [92] Architektury procesorů [92] Počítačové - sdělovací sítě [93] Systémy pro speciální účely a aplikace [95] Výkonnost systémů [95] Implementace počítačových systémů [96] Programovací techniky [96] Navrhování programu [96] Programovací jazyky [98] Operační systémy [100] Různé problematiky v oboru programování [100] Datové struktury [101] Teorie kódování a informace [101] Datové soubory [101] Výpočetní postupy na abstraktních zařízeních [101] Analýzy algoritmů a problému složitosti [102] Logická stavba a smysl programů [103] Matematická logika a formální jazyky [103] Numerické analýzy [104] Diskrétní matematika [107] Pravděpodobnost a statistika [107] Software pro matematiku [108] Modely a principy informačních systémů [108] Spravování databází [109] Záznam a čtení informací [110] Aplikace informačních systémů [112] Umělá inteligence [113] Počítačová grafika [117] Zpracování obrazu [117] Rozpoznání tvaru, obrazu [118] Simulace a modelování [118] Zpracování textu [118] Zpracování hromadných dat [119] Aplikace počítačů ve fyzice a konstruování [119] Aplikace počítačů v medicíně [120] Aplikace počítačů v sociálních a společenských vědách [120] Aplikace počítačů v umění a literatuře [121] Různé problematiky týkající se aplikací počítačů [121] Průmysl výpočetní techniky [121] Historie výpočetních postupů [122] Počítače a vzdělání [122] Počítače a společnost [125] Řízení výpočetních postupů a informačních systémů [125] Povolání v oboru výpočetní techniky [125] Rejstřík podle termínů [126]

Computing Reviews (US)

3/87

Abecední rejstřík autorů [130] Recenze knih, přednášek [131] Úvodní stať a přehled [131] Odkazy na slovníky, encyklopedie apod. [131] Vstupní a výstupní obvody a datová komunikace [131] Počítačové - sdělovací sítě [131] Implementace počítačových systémů [131] Programovací techniky [131] Navrhování programu [131] Programovací jazyky [133] Operační systémy [136] Různé problematiky v oboru programování [137] Datové soubory [137] Výpočetní postupy na abstraktních zařízeních [138] Logická stavba a smysl programů [139] Matematická logika a formální jazyky [139] Numerické analýzy [140] Diskrétní matematika [140] Pravděpodobnost a statistika [140] Modely a principy informačních systémů [140] Spravování databází [140] Záznam a čtení informací [141] Aplikace informačních systémů [141] Umělá inteligence [141] Počítačová grafika [143] Zpracování textu [144] Zpracování hromadných dat [144] Aplikace počítačů ve fyzice a konstruování [145] Historie výpočetních postupů [145] Počítače a vzdělání [145] Počítače a společnost [145] Řízení výpočetních postupů a informačních systémů [145] Vlastní výpočet [145] Neknižní literatura - recenze [147] Úvodní stať a přehled [147] Návrh logických obvodů [147] Integrované obvody [147] Architektury procesorů [147] Počítačové - sdělovací sítě [147] Systémy pro speciální účely a aplikace [149] Výkonnost systémů [149] Implementace počítačových systémů [150] Programovací techniky [150] Navrhování programu [151] Programovací jazyky [152] Operační systémy [153] Datové struktury [155] Kódování dat [155] Datové soubory [156] Obecné teorie výpočetních postupů [156] Výpočetní postupy na abstraktních zařízeních [156] Analýzy algoritmů a problému složitosti [156] Logická stavba a smysl programů [157] Matematická logika a formální jazyky [158] Numerické analýzy [159] Diskrétní matematika [161] Pravděpodobnost a statistika [161] Software pro matematiku [161] Různé problematiky matematických postupů [161] Modely a principy informačních systémů [161] Spravování databází [162] Záznam a čtení informací [163] Aplikace informačních systémů [163] Různé problematiky v informačních systémech [164] Algebraické manipulace [164] Umělá inteligence [164] Počítačová grafika [169] Zpracování obrazu [169] Rozpoznání tvaru, obrazu [170] Simulace a modelování [170] Zpracování textu [170] Aplikace počítačů ve fyzice a konstruování [172] Aplikace počítačů v medicíně [172] Aplikace počítačů v umění a literatuře [173] Využití počítačů ve

výrobě [173] Využití počítače v různých systémech [173] Průmysl výpočetní techniky [174] Historie výpočetních postupů [174] Počítače a vzdělání [175] Počítače a společnost [176] Řízení výpočetních postupů a informačních systémů [176] Povolání v oboru výpočetní techniky [176] Vlastní výpočet [176]

Computing Reviews (US)

4/87

Abecední rejstřík autorů [178] Knihy a přednášky - recenze [180] Úvodní stať a přehled [180] Aritmetické a logické struktury [180] Paměťové struktury [180] Návrh logických obvodů [180] Integrované obvody [180] Počítačové - sdělovací sítě [180] Systémy pro speciální účely a aplikace [181] Výkonnost systémů [181] Implementace počítačových systémů [181] Navrhování programu [181] Programovací jazyky [182] Operační systémy [182] Datové struktury [185] Teorie kódování a informace [185] Datové soubory [185] Výpočetní postupy na abstraktních zařízeních [186] Analýzy algoritmů a problému složitosti [186] Numerické analýzy [186] Diskrétní matematika [186] Pravděpodobnost a statistika [187] Obecná problematika informačních systémů [187] Modely a principy informačních systémů [187] Spravování databází [187] Záznam a čtení informací [188] Aplikace informačních systémů [189] Umělá inteligence [189] Počítačová grafika [191] Zpracování obrazu [191] Rozpoznání tvaru, obrazu [192] Simulace a modelování [192] Zpracování hromadných dat [192] Aplikace počítačů ve fyzice a konstruování [193] Aplikace počítačů v sociálních a společenských vědách [194] Aplikace počítačů v umění a literatuře [194] Využití počítačů ve výrobě [194] Využití počítače v různých systémech [195] Průmysl výpočetní techniky [196] Historie výpočetních postupů [197] Počítače a vzdělání [198] Počítače a společnost [198] Řízení výpočetních postupů a informačních systémů [200] Povolání v oboru výpočetní techniky [202] Neknižní literatura - recenze [203] Architektury procesorů [203] Počítačové - sdělovací sítě [203] Systémy pro speciální účely a aplikace [203] Výkonnost systémů [203] Obecné tématiky z programování [204] Navrhování programu [204] Programovací jazyky [208] Operační systémy [209] Datové struktury [211] Kódování dat [212] Výpočetní postupy na abstraktních zařízeních [212] Analýzy algoritmů a problému složitosti [212] Logická stavba a smysl programů [213] Matematická logika a formální jazyky [214] Numerické analýzy [215] Diskrétní matematika [217] Pravděpodobnost a statistika [217] Modely a principy informačních systémů [218] Spravování databází [218] Záznam a čtení informací [220] Aplikace informačních systémů [221] Algebraické manipulace [221] Umělá inteligence [222] Počítačová grafika [224] Zpracování obrazu [226] Simulace a modelování [226] Zpracování hromadných dat [226] Aplikace počítačů ve fyzice a konstruování [228] Aplikace počítačů v medicíně [228] Aplikace počítačů v sociálních a společenských vědách [228] Aplikace počítačů v umění a literatuře [228] Využití počítačů ve výrobě [228] Využití počítače v různých systémech [228] Průmysl výpočetní techniky [228] Počítače a vzdělání [228] Počítače a společnost [229] Řízení výpočetních postupů a informačních systémů [230] Povolání v oboru výpočetní techniky [232] Vlastní výpočet [232] Rejstřík podle termínů [232]

Computing Reviews (US)

5/87

Abecední rejstřík autorů [234] Knihy a přednášky - recenze, slovníky, encyklopedie apod. [236] Vstupní a výstupní obvody a datová komunikace [236] Počítačové systémy obecně [236] Architektury procesorů [236] Počítačové - sdělovací sítě [236] Implementace počítačových systémů [237] Navrhování programu [237] Programovací jazyky [239] Operační systémy [242] Datové struktury [244] Zobrazení dat v paměti [245] Kódování dat [245] Teorie kódování a informace [245] Datové soubory [245] Výpočetní postupy na abstraktních zařízeních [246] Analýzy algoritmů a problému složitosti [246] Logická stavba a smysl programů [246] Matematická logika a formální jazyky [246] Numerické analýzy [246]

Diskrétní matematika [24] Modely a principy informačních systémů [247] Spravování databází [247] Záznam a čtení informací [250] Aplikace informačních systémů [250] Umělá inteligence [251] Zpracování hromadných dat [253] Využití počítačů ve výrobě [253] Využití počítače v různých systémech [253] Průmysl výpočetní techniky [254] Historie výpočetních postupů [254] Počítače a vzdělání [254] Počítače a společnost [254] Řízení výpočetních postupů a informačních systémů [255] Povolání v oboru výpočetní techniky [257] Neknižní literatura - recenze [258] Řídicí obvody a mikroprogramování [258] Integrované obvody [258] Architektury procesorů [258] Počítačové - sdělovací sítě [258] Systémy pro speciální účely a aplikace [259] Výkonnost systémů [259] Implementace počítačových systémů [259] Programovací techniky [259] Navrhování programu [260] Programovací jazyky [261] Operační systémy [262] Různé problematiky v oboru programování [263] Zobrazení dat v paměti [263] Kódování dat [263] Datové soubory [263] Výpočetní postupy na abstraktních zařízeních [263] Analýzy algoritmů a problému složitosti [265] Logická stavba a smysl programů [266] Matematická logika a formální jazyky [267] Numerické analýzy [267] Diskrétní matematika [267] Pravděpodobnost a statistika [268] Software pro matematiku [270] Modely a principy informačních systémů [270] Spravování databází [272] Záznam a čtení informací [273] Aplikace informačních systémů [275] Algebraické manipulace [276] Umělá inteligence [277] Počítačová grafika [281] Zpracování obrazu [282] Rozpoznání tvaru, obrazu [282] Simulace a modelování [283] Zpracování textu [283] Aplikace počítačů ve fyzice a konstruování [283] Aplikace počítačů v medicíně [284] Aplikace počítačů v sociálních a společenských vědách [284] Aplikace počítačů v umění a literatuře [284] Využití počítačů ve výrobě [285] Využití počítače v různých systémech [285] Průmysl výpočetní techniky [285] Historie výpočetních postupů [285] Počítače a vzdělání [286] Počítače a společnost [287] Řízení výpočetních postupů a informačních systémů [288] Povolání v oboru výpočetní techniky [288] Rejstřík podle termínů [obálka 3]

Computing Reviews (US)

6/87

Abecední rejstřík autorů [290] Knihy a přednášky - recenze [291] Odkazy na slovníky, encyklopedie apod. [291] Řídicí obvody a mikroprogramování [291] Návrh logických obvodů [291] Integrované obvody [291] Počítačové systémy obecně [291] Architektury procesorů [292] Počítačové - sdělovací sítě [293] Systémy pro speciální účely a aplikace [293] Výkonnost systémů [293] Implementace počítačových systémů [293] Programovací techniky [294] Navrhování programu [294] Programovací jazyky [296] Datové struktury [302] Datové soubory [302] Analýzy algoritmů a problému složitosti [302] Logická stavba a smysl programů [302] Matematická logika a formální jazyky [302] Matematika ve výpočtech [302] Numerické analýzy [303] Diskrétní matematika [303] Pravděpodobnost a statistika [304] Modely a principy informačních systémů [304] Spravování databází [304] Záznam a čtení informací [306] Aplikace informačních systémů [306] Algebraické manipulace [306] Umělá inteligence [306] Počítačová grafika [308] Zpracování obrazu [308] Rozpoznání tvaru, obrazu [308] Simulace a modelování [309] Obecné problematiky v aplikacích počítačů [309] Zpracování hromadných dat [309] Aplikace počítačů ve fyzice a konstruování [309] Aplikace počítačů v medicíně [310] Aplikace počítačů v sociálních a společenských vědách [310] Aplikace počítačů v umění a literatuře [310] Využití počítačů ve výrobě [310] Využití počítače v různých systémech [312] Historie výpočetních postupů [312] Počítače a vzdělání [312] Počítače a společnost [312] Řízení výpočetních postupů a informačních systémů [312] Rejstřík podle termínů [314] Neknižní literatura - recenze [315] Paměťové struktury [315] Vstupní a výstupní obvody a datová komunikace [315] Návrh logických obvodů [316] Integrované obvody [316] Počítačové systémy obecně [316] Architektury procesorů [317] Počítačové - sdělovací sítě [317] Systémy pro speciální účely a aplikace [318] Výkonnost systémů

[318] Implementace počítačových systémů [318] Programovací techniky [319] Navrhování programu [319] Programovací jazyky [321] Operační systémy [321] Datové struktury [322] Teorie kódování a informace [322] Obecné teorie výpočetních postupů [322] Výpočetní postupy na abstraktních zařízeních [322] Analýzy algoritmů a problému složitosti [323] Logická stavba a smysl programů [323] Matematická logika a formální jazyky [324] Numerické analýzy [326] Diskrétní matematika [327] Pravděpodobnost a statistika [328] Modely a principy informačních systémů [328] Spravování databází [329] Záznam a čtení informací [330] Aplikace informačních systémů [330] Algebraické manipulace [331] Umělá inteligence [332] Počítačová grafika [333] Zpracování obrazu [333] Rozpoznání tvaru, obrazu [333] Simulace a modelování [334] Obecné problematiky v aplikacích počítačů [334] Aplikace počítačů ve fyzice a konstruování [334] Aplikace počítačů v medicíně [335] Využití počítačů ve výrobě [335] Využití počítače v různých systémech [335] Průmysl výpočetní techniky [335] Historie výpočetních postupů [335] Počítače a vzdělání [335] Počítače a společnost [336] Řízení výpočetních postupů a informačních systémů [336]

Computing Reviews (US)

7/87

Abecední rejstřík autorů [338] Knihy a přednášky - recenze [339] Odkazy na slovníky, encyklopedie apod. [339] Technické vybavení obecně [339] Vstupní a výstupní obvody a datová komunikace [339] Registry, přenos, úrovně, implementace [340] Návrh logických obvodů [341] Integrované obvody [341] Počítačové systémy obecně [343] Architektury procesorů [343] Počítačové - sdělovací sítě [345] Systémy pro speciální účely a aplikace [345] Výkonnost systémů [346] Implementace počítačových systémů [346] Obecné tematiky z programování [346] Navrhování programu [346] Programovací jazyky [348] Operační systémy [350] Logická stavba a smysl programů [352] Matematická logika a formální jazyky [352] Pravděpodobnost a statistika [352] Modely a principy informačních systémů [352] Spravování databází [352] Aplikace informačních systémů [353] Algebraické manipulace [353] Umělá inteligence [353] Počítačová grafika [358] Simulace a modelování [359] Obecné problematiky v aplikacích počítačů [360] Zpracování hromadných dat [360] Aplikace počítačů ve fyzice a konstruování [360] Využití počítačů ve výrobě [361] Využití počítače v různých systémech [361] Historie výpočetních postupů [361] Počítače a vzdělání [362] Řízení výpočetních postupů a informačních systémů [363] Povolání v oboru výpočetní techniky [364] Neknižní literatura - recenze [365] Technické vybavení obecně [365] Aritmetické a logické struktury [365] Paměťové struktury [365] Vstupní a výstupní obvody a datová komunikace [365] Návrh logických obvodů [365] Integrované obvody [366] Architektury procesorů [368] Počítačové - sdělovací sítě [368] Systémy pro speciální účely a aplikace [368] Výkonnost systémů [369] Implementace počítačových systémů [369] Programovací techniky [369] Navrhování programu [369] Programovací jazyky [372] Operační systémy [372] Datové struktury [374] Zobrazení dat v paměti [374] Datové soubory [374] Výpočetní postupy na abstraktních zařízeních [374] Analýzy algoritmů a problému složitosti [376] Logická stavba a smysl programů [377] Matematická logika a formální jazyky [377] Numerické analýzy [378] Diskrétní matematika [379] Pravděpodobnost a statistika [379] Modely a principy informačních systémů [380] Spravování databází [380] Záznam a čtení informací [382] Aplikace informačních systémů [382] Algebraické manipulace [383] Umělá inteligence [383] Počítačová grafika [386] Zpracování obrazu [387] Rozpoznání tvaru, obrazu [387] Aplikace počítačů ve fyzice a konstruování [388] Aplikace počítačů v medicíně [388] Využití počítačů ve výrobě [388] Využití počítače v různých systémech [389] Historie výpočetních postupů [390] Počítače a vzdělání [390] Počítače a společnost [391] Řízení výpočetních postupů a informačních systémů [391] Rejstřík podle termínů [391]

Erich Esders: Arbeiten mit dem 80287 (Práce s 80287), část 2., str. 50

Článek pokračuje popisem instrukčního souboru:

Přesuny dat: NPX provádí přesuny dat typu REAL, INTEGER a PACKED DECIMAL. Jedná se o instrukce typu LOAD, STORE, STORE AND POP a XCHG (výměna obsahu registrů).

Aritmetické operace: jako zdrojové operandy lze užít obsah explicitně udaných registrů; nejsou-li udány, provádějí se operace nad obsahem registru TOS (Top of Stack - vrchol zásobníku), případně registru po něm následujícího; výsledek se ukládá do registru TOS. Kromě čtyř základních početních operací je k dispozici odmocnina, násobení mocninou dvou, funkce modulo, zaokrouhlování na celé číslo, rozklad čísla na mantisu a exponent, absolutní hodnota, změna znaménka.

Operace porovnávání: porovnává se obsah registru TOS s následujícím registrem zásobníku, resp. explicitně udaným registrem, resp. s nulou. Výsledek se ukládá ve formě podmínkového kódu (Condition Code) do stavového registru.

Transcendentní funkce: operace se provádějí nad dvěma nejvyššími registry zásobníku, výsledek se ukládá do registru TOS. Instrukce předpokládají, že operandy leží v definičním oboru funkcí (to je nutno případně softwarově ověřit). Jedná se o funkce tangens, arcustangens a obecnou mocninu.

Operace s konstantou: tyto operace ukládají do registru TOS nulu, jedničku, Ludolfovo číslo, resp. další čtyři logaritmické konstanty ($\log(2)$, $\ln(2)$, dvojkové logaritmy $\log(10)$ a $\log(e)$).

Řadicí instrukce: tyto instrukce slouží k ovládání NPX na systémové úrovni, podporují např. inicializaci systému a ošetření výjimek. Některé z nich mají tvar WAIT (před provedením instrukce se testuje signál ERROR) i NO-WAIT (numerické instrukce testují vždy signál BUSY pro synchronizaci s CPU). Jde o instrukce přenosu obsahu řídicího, stavového a příznakového registru do externích registrů, resp. celé "okolí" NPX do externí paměti RAM, synchronizace stavu WAIT u CPU a NPX a některé další instrukce.

Nasazení NPX v uživatelských programech: v uváděných ukázkách se předpokládá počítač IBM PC-AT pod operačním systémem MS-DOS s makroassemblerem MASM 4.00, C-kompilátorem MSC 4.00, spojovačem LINK 3.51, debuggerem CV 1.00. Náročnější programy testují dostupnost NPX a v kladném případě užívají příslušné instrukce. V záporném případě (NPX nebyl vložen do objímky) jsou instrukce emulovány pomocí podprogramů. Kód generovaný např. Microsoft C pro prostředí MS-DOS automaticky obsahuje test dostupnosti NPX. Dostupný NPX lze zablokovat nastavením proměnné "N087" v prostředí (environment) MS-DOSu.

V článku jsou dále uvedeny ukázky programů pro vyčíslení časového trvání některých instrukcí, pro předávání parametrů, skalární součin vektorů, výpočet komplexních funkcí aj., jež se svým rozsahem vymykají rámci této informace.

Werner Hilf: Der MC68030 (Mikroprocesor 68030 - software a cache, str. 68)

Typová řada 68000 dosud umožňovala dva režimy - uživatelský (User) a privilegovaný (Supervisor). V neprivilegovaném režimu je možnost zásahů do systému zvenčí omezena. Protože využití tohoto režimu se příliš nerozšířilo, bylo od něj u 68030

upuštěno. Tím zanikly dvě instrukce - CALLM, RTM. Je-li nutná plná kompatibilita se 68020, lze tyto příkazy emulovat.

Paměť typu cache

Pro cache neexistuje odpovídající překlad; cache znamená skrytě uchovávat; tento termín je třeba odlišit od cash, jež má stejnou výslovnost, ale naprosto odlišný význam. Cache se užívají od konce šedesátých let. "On-chip-cache" (256 bajtů) byl poprvé užít u 68020 (zvětšil výkon o 10 až 20 procent). Cache leží paralelně k hlavní paměti (sdílí tytéž adresy); do cache má přístup pouze CPU, nikoli uživatel. Při provádění instrukce procesor zjišťuje, zda obsah dotyčné adresy už je v paměti cache. Zjišťuje-li se shoda celé adresy, cache se nazývá plně asociativní. U datového a příkazového cache 68030 se srovnává jen část adresy (částečně asociativní cache). Při kladném výsledku komparace jde o zásah (Hit) a CPU pracuje s cachem. V opačném případě (Miss) musí CPU pracovat s pomalejší hlavní pamětí. Při vyzvednutí instrukce z hlavní paměti zapíše procesor tuto instrukci do cache. Vyskytne-li se tato instrukce znovu, bude už obsažena v paměti cache. Užití cache má smysl u opakujících se sekvencí, které nejsou rozsáhlejší než cache. Situaci lze zlepšit pomocí řídicího (CACR - Cache Control Register) a adresového (CAAR - Cache Address Register) registru cache, které příslušnými příkazy umožňují zablokování cache, zmrazení obsahu, příp. smazání cache nebo jednoho zápisu. Zmrazení obsahu znamená, že dochází ke komparaci obsahu a k využití cache v případě zásahu, ale nedochází k přepisu.

Uwe Hildebrand: Die mc-Transputerkarte (Transputerová deska z redakce mc; část 2., str. 134)

K základnímu vybavení desky patří monitor v pamětech EPROM, který umožňuje napojení na hostitelský počítač (Download) přes sériové rozhraní.

S dodávaným crossassemblerem lze software pro transputerovou desku vyvíjet na hostitelském počítači. Vývody na konektoru desky jsou signálově kompatibilní s Inmos, takže desku lze užít v rozsáhlejším transputerovém systému.

Procesor rozděluje svůj pracovní čas mezi paralelně probíhající procesy (procesy ve stavu čekání nevyžadují žádný čas). Čtyři obousměrné komunikační kanály (Links) umožňují přímý přístup do paměti, aniž by byla přerušena práce procesoru. Komunikace se uskutečňuje po bajtech (dva start-bity - log1, osm datových bitů, jeden koncový bit - log0); příjemce potvrzuje každý bajt (dva bity - log1 a log0). Integrovaná paměť RAM odpovídá dobou přístupu cyklu procesoru. 32-bitový paměťový interfejs používá multiplexované datové a adresové vodiče. Lineárně lze adresovat 4 GB paměti, interní RAM přitom leží na začátku prostoru. Transputer lze zastavit signálem log1 na vývodu "Analyse-pin" na dobu tří cyklů při zachování aktuálního stavu procesoru. Vývod "Event Req" přenáší informaci o tom, zda externí obvody chtějí komunikovat s některým procesem. Článek pokračuje podrobným popisem systémových signálů transputeru T-414, vstupně-výstupního obvodu MC 2681 a schématem desky.

Guenther Sternberg: Overlays unter Turbo 4.0 (Překrývání v Turbo-Pascalu 4.0; str. 78)

S Turbo-Pascallem 4.0 padla hranice 64 kB pro kód programu. Podle názoru firmy Borland tím vymizela nutnost užívat techniku překrývání, která už dříve umožňovala spouštění programů s rozsahem větším než 640 KB. Důvodem k používání této techniky byla ovšem také snaha zachovat co největší prostor paměti pro data. Proto americká firma Turbo-Power-Software k Turbo-Pascalu 4.0 dodává Overlay-Manager 4.0 (na 5,25-palcové disketě s třicetistránkovým manuálem za 100 DM) pro počítače kompatibilní s IBM PC. Kdo by chtěl upravovat zdrojový program na disketě, potřebuje ještě makroassembler MASM 4.0.

(Pokračování příště) 31



PROGRAMOVÁ NABÍDKA



Pokyny k objednávání programů

Nabízené programy si zájemci objednávají výhradně na korespondenčních listcích adresovaných na 602. ZO Svazarmu, Wintrova 8, 160 41 Praha 6. Programy zasíláme na dobírku, je ale možný přímý nákup ve středisku VTI v Martinské 5, Praha 1.

Programy základní nabídky pro ZX Spectrum

Dr.MG 135 Kčs

Upravená verze spojených programů GENS 3 a MONS 3.

Datalog 186 Kčs

Databázový program, který svým komfortem v mnoha směrech převyšuje obdobné programové produkty. Pracuje s českou a slovenskou abecedou.

uB-PASCAL 205 Kčs

Prostředek pro editaci, překlad a běh programů, vhodný i pro výuku programování.

CP/M 191 Kčs

Vstupenka do světa profesionálních osmibitových počítačů; možnost využívání množství programů, které jsou tímto systémem řízeny. Instalace vyžaduje hardwarovou úpravu počítače (AR/A 10/1988).

Assembler 80 198 Kčs

Původní program, výkonný pomocník při programování ve strojovém kódu.

BASIC S 119 Kčs

Výukový program určený hlavně začátečnickům. Seznamuje s hlavními zásadami programování.

PROFESOR II 120 Kčs + kazeta za maloobchodní cenu

Univerzální výukový program, základ pro instalaci dodávaných znalostníchází typu STUDENT z nejrůznějších oborů.

STUDENT 1 96 Kčs + kazeta za maloobchodní cenu

Pět znalostníchází pro program PROFESOR (Města v ČSSR, Evropská pohoří, Světová moře a oceány, Slovní druhy, Souhvězdí).

STUDENT 2 96 Kčs + kazeta za maloobchodní cenu

Pět znalostníchází pro program PROFESOR: Naše pohoří, Významné vrcholy, Města světa, Křižovatky (dopravní výchova), Malá násobilka (pro děti).

TESTEDITOR 418 Kčs + kazeta za maloobchodní cenu

Program pro vlastní tvorbu znalostníchází typu STUDENT. Práce nevyžaduje znalost vnitřní struktury programu.

* * NOVINKY PRO ZX SPECTRUM * *

MULTITASKING 99 Kčs + kazeta za maloobchodní cenu

Operační systém pro ZX Spectrum, který umožňuje současný běh více programů na jednom počítači. Jeden z programů je v popředí, druhý v pozadí; je možné je libovolně prohodit. S programem dostanete ukázkovou pomůcku "Kalkulátor a zápisník".

GROS 79 Kčs + kazeta za maloobchodní cenu

Grafický rozhodovací systém je účinný prostředek pro podporu rozhodovacích procesů. Umožňuje volit nejvýhodnější z několika možných variant řešení daného problému.

ODA 110 Kčs + kazeta za maloobchodní cenu

Osobní databázový systém s jednoduchým a názorným ovládáním. Kromě obvyklých editačních a vyhledávacích možností dovoluje aritmetické výpočty, volbu formátu zobrazení a tisku.

PROGRAF 89 Kčs + kazeta za maloobchodní cenu

Prostorové grafy jsou programem pro názorné zobrazení prostorových funkcí s možností odstraňování zakrytých částí a s bohatou volbou parametrů (úhly natočení a nadhledu, počty řezů apod.).

TEMPERAMENT 49 Kčs + kazeta za maloobchodní cenu

Zábavný psychologický test k určení typu temperamentu. Můžete se dozvědět, "kdo" vlastně jste, kam můžete zařadit své příbuzné a známé.

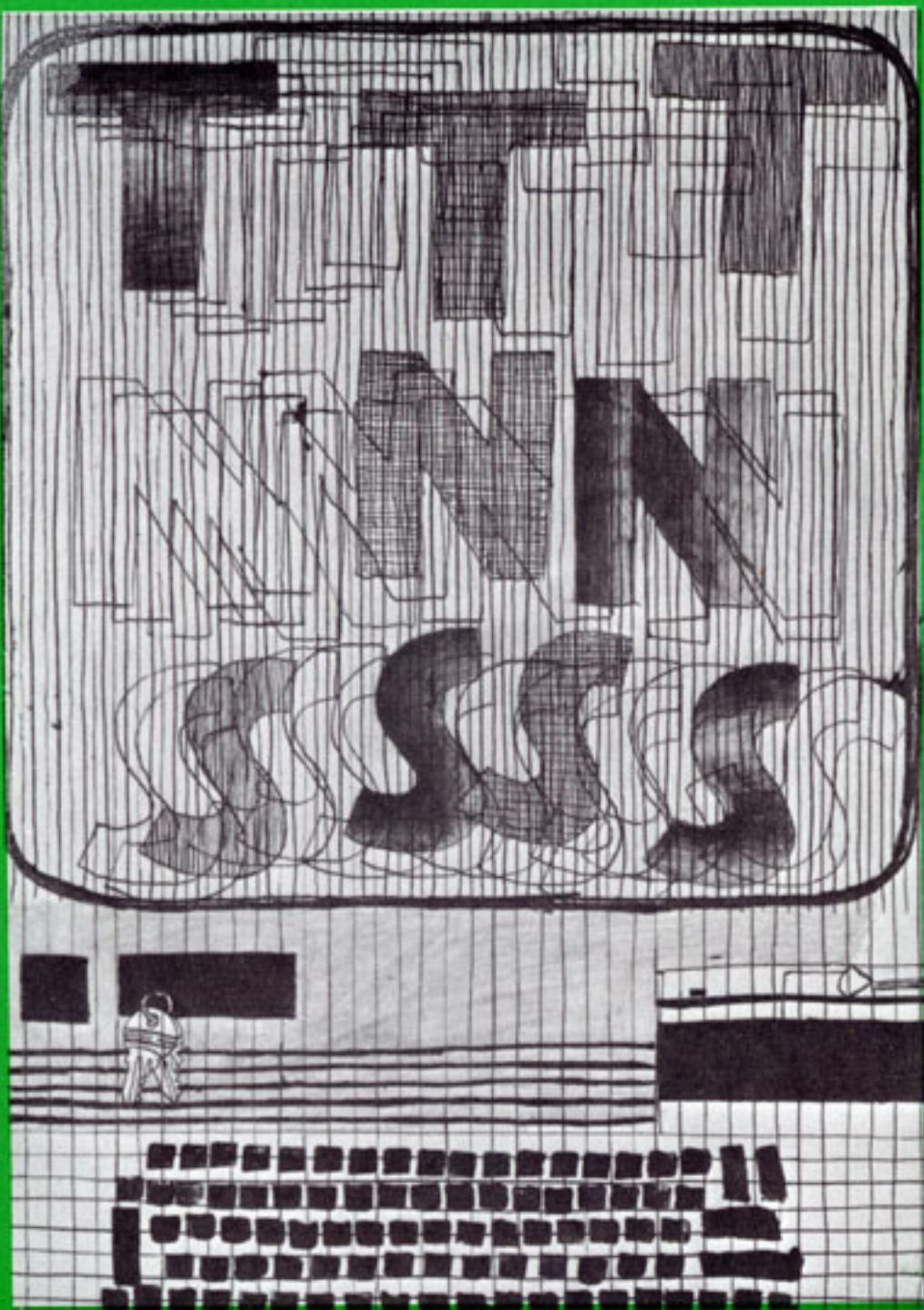
STOPKY 110 Kčs + kazeta za maloobchodní cenu

Časoměrný program s editorem pracujícím v paralelním režimu. Tento multitasking umožňuje zároveň odečítat časy a editovat záznamy o jednotlivých závodnicích nebo jiných měřených jevech. Vhodný pro různé sportovní soutěže nebo pro měření, když je zapotřebí sbírat velké množství časových údajů doplněných komentáři.

SONDA 4D 49 Kčs + kazeta za maloobchodní cenu

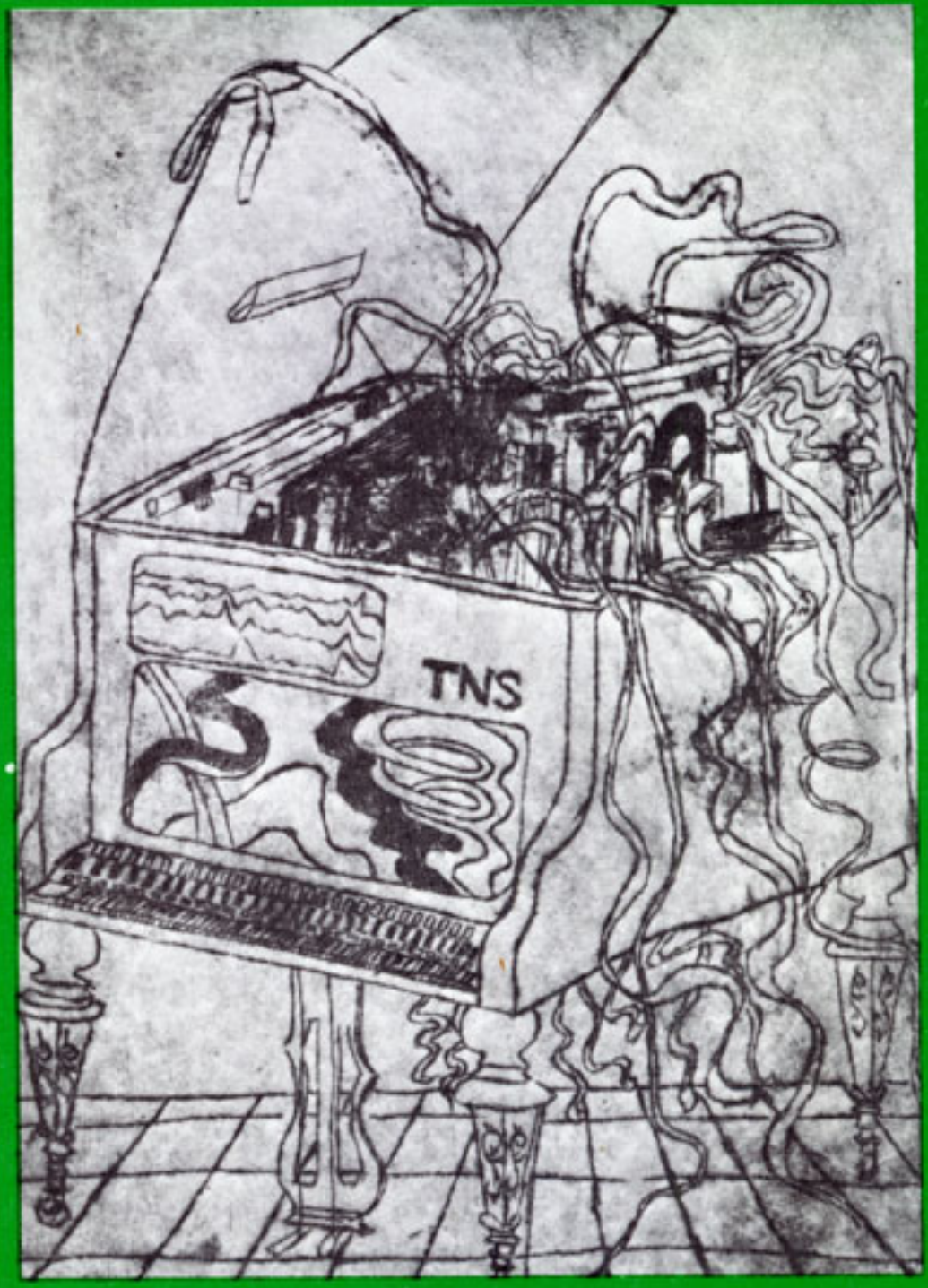
Zábavný program, který vám umožní cestu do fantastického světa čtvrtého rozměru. K programu lze přistupovat různým způsobem - od pohledu průzkumníka, který se pouze seznamuje se zajímavým prostředím, až po úroveň hlavolamu.

Nabídky z minulých čísel zůstávají v platnosti



DĚTI A POČÍTAČE

Úspěšné obrázky z výtvarné soutěže, uspořádané JZD AK Slušovice





Desktop Publishing — tiskový systém firmy HP založený na osobním počítači

Jedná se o kombinaci osobního počítače HP Vectra, kompatibilního s IBM-PC AT s odpovídajícím programovým vybavením, laserovou tiskárnou HP LaserJet, případně s dalšími doplňky pro produkci vysoce kvalitních dokumentů. Ve spojení s novým modelem rastrového snímáčiho zařízení HP ScanJet můžete kombinovat ve Vašich dokumentech i fotografie s grafikou a textem.



HP LaserJet II

Nejnovější model laserové tiskárny firmy Hewlett-Packard, model HP 33440AB, umožňuje používat 32 různých typů písma (včetně české a slovenské abecedy a sad písma IBM) a může být vybaven až 4.5 MBytů vnitřní paměti pro grafické aplikace. Rychlost tisku: 8 stran za min., formát A4, rozlišení: volitelné až do 300 bodů/inch.



HP ScanJet

Tento nový rastrový snímač grafických předloh, model HP 9190A, má volitelné rozlišení až 300 bodů/inch a je kompatibilní s laserovou tiskárnou LaserJet.

Zpracovává libovolné předlohy formátu A4 (knihy, fotografie). Umožňuje sejmutý dokument dále zpracovat na počítači, doplnit o další informace a vytisknout na laserové tiskárně.

ScanJet umožňuje zvětšení/zmenšení v krocích o velikosti 1% a snímání v 16-ti stupních šedi.

