

ZO SVAZARM KAROLINKA



S E M I N Á Ř - K V Ě T E N 1 9 8 7

Obsah

Textový editor	Tasword	1
Textový editor	Writer	7
Mega Basic		10
Hi-T		21
VU-CALC		27
COLECTOR PACK		30
Plošné spoje		32
ARTIC FORTH		40
DEBUGGER		43
MONITOR		48

SEZNAM FUNKCÍ-- T A S W O R D T W O T U T O R

BOTTOM	(sym+U)	-Skok na konec textu
TOP	(sym+I)	-Skok na začátek textu
ROLL DOWN	(sym+F)	-Roluje dolů - po řádcích
ROLL UP	(sym+G)	-Roluje nahoru
HELP 1	(caps+I)	-Seznam řídicích funkcí - návrat ENTER
HELP 2	(caps+I+EM)	-"- _"- _"- _"-
ZEILE <>	(sym+W)	-Symetrie textu dle vodorovné osy
ZEILE >=	(sym+E)	-Posun textu vpravo od kurzoru
ZEILE <=	(sym+Q)	-Posun textu vlevo od kurzoru
WORT >=	(caps+4)	-Posun kurzoru vpravo o 1 slovo
WORT <=	(caps+3)	-Posun kurzoru vlevo o 1 slovo
GRAPHICS		-Grafické znakyřícení tiskárny
BLOCK 3	(sym+D)	-Srovnání odstavce
DEL LINE	(sym+S)	-Vymaže řádek
INSERT	(sym+Y)	-Vkládání znaků do napsaného textu
I/O MENUE	(sym+A)	-Menu (into Basic)

ROZŠÍŘENÝ MÓDUS E X T E N D E D M O D E

PASE UP	(EM+G)	-Roluje po stránkách (nahoru)
PASE DOWN	(EM+F)	-Roluje po stránkách (dolů)
ZEILE >!	(EM+J)	-Zarovnání pravého okraje řádky
ZEILE ><	(EM+H)	-Zruší zarovnání pravého okraje řádky
>!	(EM+E)	-AUTOMATICKÉ zarovnání p.o.řádky zap. ON,vypíná OFF (R.Justify ON) (případ, dát kurzor na zač.odstavce a BLOCK >!) (sym+D)
WORD WRAP	(EM+W)	-Zap.a vyp.dělení slov na konci řádky (ON-OFF)
NEME ZEILE	(EM+I)	-Vložení nových řádek -(ON-OFF)
32-64	(EM+C)	-(ON-OFF) - LUPA -pohyb kurzorem
RAND LINILS	(EM+A)	-Levý okraj - kurzorem
RAND RECHTS	(EM+D)	-Pravý okraj - kurzorem
DEL RAND	(EM+S)	-Zruší "-"
REPLACE	(EM+R)	-Hledat text za kurzorem a nahrazovat jiným - další hledání + ENTER ,další změnu-před ENTER zadat nová slova
START BLOCK	(EM+B)	-Označení začátku bloku
END BLOCK	(EM+V)	-Označení konce bloku

POZOR po definici bloku nastavit kurzor na správné místo pro kopírování

MOVE BLOCK	(EM+M)	-Přesun bloku
COPY BLOCK	(EM+N)	-Kopírování bloku
CLEAR TEXT	(EM+X)	-Vymazání celého textu
LIST TEXT	(EM+P)	-Listuje po řádcích automaticky - zastavit klávesa Q

T A S W O R D T W O T U T O R
(Taswan Software - ZX Spectrum 48k)

Položte prosím přiloženou šablonu na klávesnici.

Pro normální módus jsou funkce řídicích kláves nadepsány nad příslušným písmenem, zde U.	ZEILE <>
Pro módus rozšířený (Extended), který bude pojednán později, jsou funkce zapsány pod písmenem.	<pre> ----- U ----- WORD WRAP </pre>

Aby klávesa fungovala jako řídicí, je třeba stisknout některé z tlačítek SHIFT. V následujícím textu je klávesa SYMBOL SHIFT zkráceně označena jako SYM a klávesa CAPS SHIFT jako CAPS.

První klávesa, se kterou se nyní seznámíme, je klávesa BOTTOM. Pomocí této klávesy se lze dostat na konec textového souboru. Vyzkoušejte si tedy ,BOTTOM tak, že stisknete klávesu SYM současně s klávesou U.

Stiskněte nyní prosím současně klávesu SYM a F a podržte tak dlouho, dokud se na poslední řádce neobjeví zpráva : >> UVOLNIT KLÁVESU F <<

Klávesa F - ROLL DOWN posunuje "okénko" obrazovky o jednu řádku textu dolů. Opačnou funkci, kterou lze okénko obrazovky v textu posouvat o jednu řádku vzhůru je SYM & klávesa G - ROLL UP.

Klávesami ROLL DOWN a ROLL UP tedy lze procházet textovým souborem dopředu a nazpět. Vyzkoušejte si nyní pomocí těchto kláves prohlížení textu!

Užitečnou klávesou je HELP (anglicky "pomoc"), díky které není třeba příliš často nahlížet do manuálu Tasword 2. Stisknutím CAPS & klávesy I se na obrazovku dostává seznam řídicích funkcí v normálním módu. Návrat zpět do textu se provede klávesou ENTER. Vyzkoušejte si nyní HELP! Existuje také seznam řídicích funkcí v módu rozšířeném (extended). Ten lze získat jako pokračování stránky HELP současným stisknutím SYM & CAPS. Pokračujme však dále pomocí ROLL DOWN.

>> UVOLNIT KLÁVESU F <<

Klávesa U-ZEILE <> (SYM & U) posouvá obsah řádku osmě symetricky podle svislé osy obrazovky. Vyzkoušejte si funkci ZEILE <> na podtržené zprávě o 4 řádky výše. K tomu je třeba posunout blikající kurzor na začátek řádku (pomocí kláves pro pohyb kurzoru - CAPS & 5,6,7,8) a stisknout SYM & U. Kurzor automaticky skočí na novou řádku.

Klávesou ZEILE >= lze posunout text vpravo od kurzoru o jednu posíci doprava. Klávesou ZEILE <= lze text nalevo od kurzoru posunout o jednu posíci vlevo.

Použijte prosím podtrženou zprávu výše pro vyzkoušení právě popsáných funkcí. A nyní dále ROLL DOWN !

Klávesami CAPS & 2 získáme funkci CAPS LOCK. Stiskněte prosím tuto kombinaci. Každé písmeno, které nyní zadáte, bude velké. Že je funkce CAPS LOCK sepnuta, Vám ukazuje zpráva vlevo nad stavovou řádkou. Opakovaným stisknutím klávesové kombinace se funkce CAPS LOCK vypíná.

Jednatlivá velká písmena lze samozřejmě také zadávat samostatně stisknutím CAPS & klávesy příslušného písmene. Všechny znaky, jako je čárka, tečka atd. se zadávají pomocí klávesy SYM, např. SYM & c = ?.

Klávesami WORT >= a WORT => můžete jedním stisknutím posunout kurzor vždy o jedno celé slovo vpravo či vlevo: CAPS & 3 nebo 4. Těmito funkcemi lze pohybovat kurzorem rychleji, než klávesami pro pohyb kurzoru!

Klávesa GRAPHICS se používá pro psaní grafických znaků na číslicových klávesách. CAPS & 9 spíná grafický módus, indikací tohoto módu je zpráva nad stavovou řádkou vpravo. Opakovaným stisknutím klávesové kombinace se grafický módus vypíná.

Grafické symboly mohou být také použity ke styku s tiskárnou. Inicializaci symbolů lze nalézt na stránce HELP. Tyto symboly slouží k řízení tiskárny Epson FX-80. Jak se řídí tiskárny jiných typů je třeba najít v manuálu.

Nejjednodušší způsob, jak opravit takto špatně napsané písmeno, je nastavit kurzor pomocí kláves pro jeho pohyb na příslušnou chybu a tuto pohybem přepsáním opravit. Zkuste opravit chybu vludivší se do tohoto odstavce.

Pro vymazání řádky se používá klávesa DELETE ; CAPS & 0. Kurzor se nastaví na mazaný znak a provede se DELETE. Část řádky vpravo od kurzoru a kurzor sám jsou posunuty o jednu pozici doleva. XYZXYZ. Použijte funkci DELETE a vymažte z tohoto odstavce znaky, které sem nepatří !

Jistě jste shledali, že po odstranění znaků XYZXYZ se stavba odstavce poněkud rozházela. Při změnách a opravách v textu se Vám bude toto stávat často, avšak to není žádný důvod k obavám :

Toto je odstavec, ve kterém bylo provedeno několik změn. Nastavte prosím kurzor na začátek odstavce a stiskněte BLOCK>= (SYM & D), budete překvapeni! Funkce BLOCK>= odstavec zase pěkně srovná!

Toto je řádka, která na toto místo nepatří !

Funkce DEL LINE (SYM & S) - na německé šablone DEL ZEILE, umí vymazat celou řádku, na které je kurzor nastaven. Vymažte prosím přebytečnou řádku !

Funkce INSERT (SYM & Y) se používá třemi způsoby:

- (1) K vložení nové řádky do textu se kurzor nastaví na začátek řádky, která má po vkládané řádce následovat; pak se už jenom stiskne INSERT.
- (2) Ke vsunutí slova do mezery mezi dvěma již napsanými slovy se kurzor nastaví do mezery mezi slovy a stiskne se INSERT.

(3) Vložení znaků do napsaného slova se provede nastavením kursoru na znak, který má se vsunovaným znakem sousedit zprava, a opět stisknutím INSERT.

Funkcí INSERT se stavba odstavce vždy poruší, avšak funkcí BLOCK>= ji lze zase opravit. Pokuste se nyní do tohoto odstavce vsunout pár řádek, slov a znaků a dejte narušený text funkcí BLOCK>= zase do pořádku!

Funkce I/O MENUE se používá hlavně tehdy, chceme-li nahrát textové soubory na kazetu anebo z kazety natáhnout, či chceme-li je vytisknout na větší tiskárně. Kromě toho lze pro řízení tisku definovat funkci grafických znaků a přizpůsobit tak program TASWORD použitému typu tiskárny.

EXTENDED MODE

(rozšířený módus)

Tasword pracuje také v tzv. rozšířeném módu neboli "Extended Mode" (v dalším textu EM), do kterého se lze dostat současným stisknutím klávesy CAPS SHIFT (CAPS) a klávesy SYMBOL SHIFT (SYM). Nastavení rozšířeného módu je indikováno blikáním spodní stavové řádky na obrazovce. Opakovaným současným stisknutím obou kláves se lze opět vrátit do normálního módu.

V rozšířeném módu lze funkci EXT HELP (CAPS & 1) dostat na obrazovku stránku HELP pro rozšířený módus přímo. Zkuste tedy nyní prosím vypsat HELP-stránku pro rozšířený (extended) módus a pak se vraťte do módu normálního, abychom mohli pokračovat.

Také v rozšířeném módu lze pohybovat kurorem libovolným směrem na obrazovce pomocí kláves pro pohyb kursoru.

Pro vyvolání ostatních funkcí v rozšířeném módu musíte stisknout příslušnou klávesu (bez shiftu)!

Jako vždy, existuje výjimka. Jsou jí přehlásky ({} \[] ~), Tato verze programu Tasword je uzpůsobena pro německé texty. Přehlásky jsou dosažitelné v rozšířeném módu a jsou na šabloně vyznačeny (vyzkoušejte si prosím klávesy písmen Y, O, U a T). Stisknutí samotné klávesy dává písmeno malé, spolu s CAPS dostaneme písmeno velké.

Funkce PAGE UP a PAGE DOWN získáme v rozšířeném módu klávesami G a F. Těmito klávesami lze prolístovat napsaný text po stránkách, to znamená, že jediným stisknutím klávesy posunujete okénko obrazovky o plných 22 řádků směrem dolů nebo nahoru. Těmito klávesami můžete rychle vyhledávat určitá místa v textu. Vyzkoušejte si prosím nyní tyto funkce, avšak pamatujte si toto místo v textu, abyste se na ně mohli vrátit a pokračovat dále ve studiu programu Tasword.

Z X P R I N T E R (tiskárna ZX)

Pomocí celkem tří kláves v rozšířeném módu lze ovlivňovat tisk na tiskárně ZX. Používejte tyto funkce však jen tehdy, je-li tiskárna skutečně připojena.

Pro výtisk textu se užívá klávesy ZX-PRINT (EM & P).

Funkcemi RAND LINKS (EM & A) a RAND RECHTS (EM & D) můžete pomocí kursoru stanovovat pravý a levý okraj textového souboru. Nový text pak bude psán jen v takto stanovených mezích. Kursor však je pomocí kláves pro jeho pohyb stále volně pohyblivý, což umožňuje uvedené meze popsáním způsobem znovu předefinovat. Vyzkoušejte si tyto klávesy a až budete hotovi, zrušte nastavené meze funkcí DEL RAND (EM & S).

Funkce REPLACE (EM & R) umožňuje v textu následujícím po momentální pozici kursoru hledat daný text anebo jej nahrazovat textem jiným. Proto je nutné, před zamýšlenou akcí kursor správně nastavit, například funkcí TOP (SYM & I) na začátek souboru.

Pokuste se nyní pomocí funkce REPLACE najít v textu slovo

Profisoft

a nezapomeňte si předem nastavit kursor na začátek textu. Pak stisknete REPLACE a dvakrát za sebou ENTER. A nemějte starost, ze to chvílku potrvá. Mnoho štěstí!

Hledáte-li text, který je v souboru obsazen vícekrát, zadáváte jej po vyvolání funkce REPLACE jen poprvé. Při dalších užitích funkce už jen stačí stisknout ENTER.

Pokud při druhém zadání jen stisknete ENTER, předpokládá Tasword, že jen hledáte nějaké slovo. Chcete-li však nějaké slovo v souboru nahradit slovem jiným, zadejte před druhým stisknutím klávesy ENTER nové slovo. Nahradte nyní v textu od tohoto místa dále slovo

Tasword

slovy "textový editor Tasword" a po ukončení náhrady se vraťte na toto místo!

Tasword také umožňuje přesuny textových odstavců (bloků). Klávesou START BLOCK (EM & B) si označíte speciálním znakem pozici kursoru na začátku bloku. Stejně se používá klávesa END BLOCK (EM & V) na označení konce bloku. Přitom je možné, že nastane na chvíli prodleva, neboť Tasword testuje, není-li již někde v textu definován jiný blok.

Poté, co byl rozsah bloku definován, nastavíte kursor na řádku, od které má být blok vložen, a stisknete MOVE BLOCK (EM & M). Má-li být blok kopírován, stisknete COPY BLOCK (EM & N). Pokuste se prosím tento odstavec zkopírovat!

Pro vymazání celého textového souboru se používá CLEAT TEXT. Tasword se Vás pro jistotu ještě jednou zeptá, máte-li opravdu v úmyslu soubor zrušit. Toto je také konec textu "tréning" a chcete-li zkusit i tuto poslední funkci, tedy prosím (EM & X).

Firmy Taswan Software a Profisofofa Vám přejí příjemnou a pohodlnou práci s textovým editorem Tasword !

Pokud jste se na toto místo dostali stisknutím klávesy BOTTOM, bude Vás asi zajímat, jak se zase dostanete zpět na začátek textového souboru. Nuže, stiskněte jednoduše klávesu TOP (SYM & I)

POPIS TEXTOVÉHO EDITORU
SPECTRAL WRITER

22.2.1986

JRN + R&K HK 21213

VOLBA REŽIMU => ss + "A" (METOP)

1. Kopie bloku z textu na jiné místo v textovém souboru
2. LOAD / MERGE textového souboru (dále jen textu) z kazety
3. SAVE textu na kazetu
4. LOAD textu z wafadriby
5. SAVE textu na wafadriby
6. Výmaz textu z wafadriby
7. Přímě z wsiery
8. Tisk textu
9. Redefinice řídících znaků tiskárny
- A. Redefinice speciálních parametrů
- B. SAVE programu SPECTRAL WRITER
- C. Zpět k textu
- D. Zpět k systému BASIC

cs + ss MODE NORMAL / EXTEND

cs + "1" (EDIT) help page = listing příkazů NORMAL MODE
nebo EXTEND MODE (dle právě zvoleného mode)

PŘÍKAZY NORMAL MODE

EDIT	help page (listing těchto příkazů)
CAPS LOCK	písmena velká / malá
TRUE VIDEO	insert a krok (posuv textu od kurzoru)
INV. VIDEO	delete a krok (posuv textu ke kurzoru)
arrows	posuv kurzoru dle šipek (cs + 5.6.7.8)
GRAPHICS	G mode vyp/zap
DELETE	výmaz znaku
<=	posun znaků doleva
<>	vystředění textu na řádku
>=	posun znaků doprava
AND	Reform paragraf
OR	posun textu na obrazovce o řádek dál
AT	posun textu na obrazovce o řádek zpět
STOP	listing VOLBA REŽIMU
NOT	výmaz řádku
STEP	vložení řádku
TO	posuv kurzoru na začátek textu
THEN	posuv kurzoru na konec textu
ENTER	nový řádek
BREAK	skok na další tabulátor

P R Í K A Z Y E X T E N D M O D E

```

EDIT      listing těchto příkazů
CAPS LOCK písmena velká / malé
arrows    posun kurzoru podle šipek
1         nastavení tabulátoru
2         zrušení tabulátoru
3         posun textu na obrazovce o stránku dle
4         posun textu na obrazovce o stránku zpět
5         posun kurzoru na nejbližší tečku (koniec věty)
6         posun textu na řádcích k levému okraji
7         vystředění textu na řádcích
8         posun textu na řádcích k pravému okraji
E         UNJUSTIFY LINE - srazí slova k prvnímu levému
R         JUSTIFY LINE - první levé slovo ponechá na svém
T         místě a pak rovnoměrně doplní mezi slova mezery
          tak, že je pravidelně zaplněn celý řádek
Y         DIVIDE LINE - vloží řádek, zbytek textu posune
P         tisk textu na IX PRINTER
S         hledání zadaného řetězce znaků
D         DELETE (vymaže) textu
F         zápis znaků FF - nový formát papíru na tiskárně
G         zápis znaků LF - nový řádek na tiskárně
H         UNJUSTIFY THE TEXT - jako "R" ale pro celý text
J         JUSTIFY THE TEXT - jako "T" ale pro celý text
K         AUTO-JUSTIFY vyp/zap - při tvorbě textu automa-
          ticky doplňuje řádek mezerami, při použití ENTER
          na příslušném řádku nefunguje
L         AUTO-WORDWRAP vyp/zap - při tvorbě textu nenechá-
          vá zbytky slov na konci řádku, ale přesouvá je
          na nový řádek
C         CHANGE (změna) barev PAPER, INK, BORDER + BRIGHT
ENTER     nový řádek
BREAK     další tabulátor

```

REDEFINICE ŘÍDÍCÍCH ZNAKŮ TISKÁRNY

Redefinici navolíme jako OPTION 9 při VOLBĚ REŽIMU. Nejprve dostaneme možnost předefinovat znaky 1 až 8 GRAPHICS MÓDU. Každému grafickému znaku jsou přiřazeny pouze 4 bity. Narazí-li se při tisku na grafický znak, jsou na tiskárnu vyslány zadané 4 bity (0 se nevysílá). Těto skutečnosti lze využít např. k přepínání barvicí pásky apod.

Nechceme-li zadávat znaky chr# 128 - chr# 143 grafického módu, odpovíme na otázku "DO YOU WISH TO CHANGE THESE ?" stiskem klávesy "N".

Dále je nabídka řídicích znaků tiskárny s možností jejich změny. V závorce je uveden kód právě platný, tlačítkem ENTER ho odsouhlasíme.

- CARRIAGE RETURN (13) = návrat vozu
 - LINEFEED (10) = nový řádek
 - FORMFEED (12) = nový list (formát)
- Číslo v závorce udává posun papíru o zadaný počet řádků.
Je-li rovno 0, pokračuje tisk po stisku tlačítka ENTER

Místo schválení tlačítkem ENTER lze psát nový kód pro danou funkci.

REDEFINICE SPECIALNICH PARAMETRŮ
=====

Nastavíme jako OPTION "A" při VOLBĚ REŽIMU. Nabídka:

COLUMN AT WHICH THE BELL RINGS (50)
sloupec, od kterého dinkne z tonaček
LENGTH OF KEBOARD CLICK (4)
délka pípnutí po stlačení klávesy
DELAY BEFORE AUTO-REPEAT (30)
zpoždění před opakováním znaku po stlačení kláve-
sy
SPEED OF AUTO-REPEAT (1)
rychlost opakování znaku při trvalém stisku klá-
vesy
BAUD RATE (2400)
rychlost přenosu v bodech
INTERFACE CODE (0)
druh připojení (interface) tiskárny (0-255)
PAGE NUMBERING MODE (0)
0 - nečísluje stránky, bez hlavičky
1 - čísluje stránky a hlavičkou, je-li definována
 (hlavička vlevo, číslo stránky vpravo)
2 - střídá pozici čísla stránky a hlavičky
NEXT PAGE NUMBER (1)
číslo příští stránky - chceme-li vždy od 1 nutno
po předchozím tisku nastavit 1
LINES PER PAGE (50)
řádků tisku na stránku
LEFT MARGIN (8)
zanechané rozpětí prázdných znaků před tiskem
každého řádku - levý okraj
PAGE HEADER
hlavička stránky lze v tomto režimu nedefinovat
např. datum, firma ap. Pak každá stránka tisku
začíná vlevo hlavičkou a vpravo číslem stránky.

Textový editor SPECTRAL WRITER pojme 352 řádků po 64 znacích
textu.

M E G A B A S I C + S P R I T E D E S I G N E R

Jak začít:

Nahrání YS MegaBasicu je jednoduchost sama. Vložte:

LOAD ""

YS MegaBasic je dodáván na pásku spolu s programem nazvaným Sprite Designer, který vám umožňuje vytvářet spraity, které můžete potom používat ve svých programech.

KLÁVESNICE

Jakmile se MegaBasic nahraje, uvidíte krátkou úvodní zprávu a inverzní mezeru v levém dolním rohu obrazovky. Toto je nový kurzor a udává, kde se budou objevovat vámi zadávané příkazy.

YS MegaBasic využívá pro input celou obrazovku, oproti normálnímu Spectru, které má obrazovku rozdělenou na dvě části. Zkuste stisknout několik tlačítek na klávesnici a všimnete si, že se neobjevují obvyklá klíčová slova... Místo toho vidíte samostatná písmena. Všechny příkazy musíte zadávat písmeno po písmenu - vlastnost, která změnila klávesnici vašeho Spectra v něco, co se blíží klávesnici normálního počítače. Toto má však nevýhodu v tom, že např. příkaz PRINT, který jste na Spectru získal stisknutím jediného tlačítka, musíte nyní získat stisknutím pěti tlačítek: P,R,I,N,T. Proto vám YS MegaBasic umožňuje mnoho příkazů psát zkratkami.

Nyní následuje kompletní seznam klíčových slov spolu s jejich zkratkami. Slova, která zde nenajdete, nemají zkratky, a tudíž je musíte vypisovat celá. Také si všimněte, že zkratka příkazu musí končit tečkou; např. zkratka z 'CONTINUE' je 'CON.'.

A.TTR	ER.ASE	ME.RGE	RES.TORE
BE.EP	E.XP	M.OVE	RET.URN
B.IN	FL.ASH	NE.XT	R.NO
BO.RDER	F.ORMAT	N.OT	SA.VE
BR.IGHT	GO S.UB	OP.EN #	S.GREEN\$
CH.R\$	G.O TO	OV.ER	ST.R\$
CI.RCLE	I.NKEY\$	PA.PER	T.AB
CLE.AR	INP.UT	PAU.SE	TH.EN
CL.OSE #	INV.ERSE	PE.EK	U.SR
C.ODE	L.EN	PL.OT	V.AL\$
CON.TINUE	LI.NE	P.OINT	VE.RIFY
DA.TA	LL.IST	PR.INT	
D.EF FN	LP.RINT	RA.NDOMIZE	
DR.AW	LO.AD	RE.AD	

Uvědomte si také, že chcete-li vkládat vcelku příkazy jako 'GO TO' nebo 'OPEN #', nesmíte vypustit mezery, které se v nich vyskytují; počítač rozumí příkazu 'GO TO', ale 'GOTO' dává zprávu 'Syntax error'.

Spodní řádek obrazovky je použit k zobrazování módu kurzoru. V následující tabulce jsou uvedeny normální módy Spectra spolu s ekvivalentními zprávami MegaBasicu:

Mód kurzoru	Zpráva v dolním řádku
'L'	CAPS OFF
'C'	CAPS ON
'E'	CAPS OFF EXTENDED
'G'	CAPS OFF GRAPHICS

Mód kurzoru je měněn obvyklým způsobem - použitím tlačítek Caps Lock, Graphics, Caps Shift a Symbol Shift.

EDITOR

MegaBasic má výborné schopnosti editace a proto některá tlačítka fungují jinak než u standardního Spectra:

EDIT	Zkopíruje programový řádek do editačního řádku
TRUE VIDEO	Smaže celý text v editačním řádku
INVERSE VIDEO	Vymaže text napravo od kurzoru
'<='	Přesune kurzor na začátek editačního řádku
'>'	Vymaže text od kurzoru do konce řádku
'>='	Přesune kurzor na konec editačního řádku
SCREEN\$	Zobrazí automatický listing (do okna 1) Řádek s programovým kurzorem bude první na obrazovce.
OR	Posune určený řádek nahoru a zobrazí automatický listing
AND	Posune určený řádek dolů a zobrazí automatický listing
STOP	Posune kopírovací kurzor o jeden znak doleva
NOT	Posune kopírovací kurzor o jeden znak nahoru
STEP	Posune kopírovací kurzor o jeden znak dolů
TO	Posune kopírovací kurzor o jeden znak doprava
AT	Zkopíruje znak pod kopírovacím kurzorem pod editační kurzor. Kopíruje pouze znaky standardní velikosti
OVER	Přesune kopírovací kurzor do dalšího okna
INVERSE	Přesune kopírovací kurzor do levého horního rohu určeného okna

Všimněte si, že obrazovka je rozdělena na čtyři okna, z nichž každé má své specifické určení: Okno nula zobrazuje uživatelský input a chybová hlášení, okno jedna zobrazuje automatický listing, produkovaný řádkovým editorem, okno dvě zobrazuje výsledky programu a okno tři je využito pro zobrazení čelního panelu (viz dále). Kdýž nahrajete YS MegaBasic, můžete si myslet, že na obrazovce je pouze jedno okno, ale to není pravda. Na obrazovce jsou čtyři okna, která se překrývají.

Všimněte si také, že druhý kurzor může být používán ke kopírování textu z jiné části obrazovky pod editační kurzor (input cursor). Tento kurzor (kopírovací) vypadá jako blikající čtverec na obrazovce a může být přemístován pomocí řídicích tlačítek. Kopírovací kurzor pracuje pouze v oknech nula, jedna a dva. Jákýkoliv řádek může být editován buď klasickým způsobem (CAPS SHIFT + 'i') nebo pomocí příkazu 'EDIT_'. Za příkazem 'EDIT_' musí následovat numerický výraz, který udává, který řádek má být editován. Jestliže řádek s udaným číslem neexistuje, je použit řádek s nejbližším vyšším číslem. Jestliže takový řádek také neexistuje, zobrazí se chybové hlášení 'Line not found'.

KLÁVESY DEFINOVATELNÉ UŽIVATELEM

Je možné naprogramovat klávesy horního řádku klávesnice, aby produkovaly řetězec až 255 znaků dlouhý. K naprogramování klávesy použijte příkaz 'KEY_' spolu s numerickým výrazem a řetězcem (odděleným čárkou). Numerický výraz udává, které tlačítko má být nadefinováno, řetězec udává, co má být na tlačítku nadefinováno. Umístění znaku ENTER (CHR\$ 13) za řetězec vám umožní, aby se nadefinované příkazy ihned automaticky provedly. Abyste se nemuseli zdržovat definováním tlačítek s příkazy 'RUN' a 'LOAD', YS MegaBasic je už obsahuje:

VERIFY	Obsahuje řetězec 'RUN + CHR\$ 13'. Toto tlačítko okamžitě spustí program uložený v paměti.
VAL\$	Obsahuje řetězec 'LOAD":RUN + CHR\$ 13'. Nahraje program z magnetofonu a spustí ho.

ŘÍDÍCÍ KLÁVESY

V době, kdy běží program, je mezera (SPACE) použita jako nové tlačítko SHIFT, takže abyste získal mezery, musíte stisknout dohromady SYMBOL SHIFT a SPACE. O mezeře, stisknuté spolu s jiným tlačítkem, budeme hovořit jako o řídicí klávese:

CONTROL F	Zobrazí čelní panel (viz dále)
CONTROL E	Zastaví provádění programu - Spectrum vytiskne zprávu 'Escape' a vrátí se do editoru.
CONTROL R	Zastaví provádění programu, sepne BORDER, PAPER a INK na jejich počáteční hodnoty, vytiskne úvodní zprávu a aktivuje editor. Nevymaže program !

VÝSTUP NA OBRAZOVKU

OKNA

Velikost a vzhled znaku a způsob, jakým mohou být tisknuty, jsou faktory, které byly v MS MegaBasicem velice rozšířeny. Nyní je například možné směřovat výstupní hodnoty programu do určité oblasti obrazovky. Tyto oblasti se nazývají 'OKNA' a mohou mít jakoukoliv velikost. Na MegaSpectrum má 10 oken - číslovaných od nuly do devíti a všechna můžete používat ve svých programech (i když, jak bylo řečeno, okna nula až tři jsou již používána systémem).

Když běží váš program, příkaz PRINT automaticky používá okno dvě. Je ale možné přepnout výstup do jiného okna příkazem 'CURRENT'. Příkazové slovo je následováno číslem okna, které má být použito.

Velikost a poloha okna se definuje příkazem 'WINDOW_', který má formu:

WINDOW_y,x,d,w

kde 'y' (číslo řádku, ve kterém se nachází levý horní roh okna) může být nula až 23, 'x' (číslo sloupce, ve kterém se nachází levý horní roh okna) může být nula až 63, 'd' je výška okna a 'w' je šířka okna. Protože MS MegaBasic zavedl 64 znaků na řádek, obrazovka má vlastní rozměry 24x64.

Standartní funkce Spectra ATTR a SCREEN# stále používají starý souřadnicový systém, ale PRINT AT používá nový systém. Také si uvědomte, že souřadnice příkazu PRINT AT jsou relativní vzhledem k levému hornímu rohu určeného okna, zatímco ATTR a SCREEN# používají absolutní souřadnice.

Po provedení příkazu 'WINDOW_' se pozice tisku přesune do levého horního rohu okna.

CLS a CLW

Standartní příkaz CLS je používán pro mazání celé obrazovky. Nový příkaz CLW je určen k vymazání jednoho okna. Slovo CLW je následováno jedním nebo dvěma čísly. Jestliže jsou čísla dvě, pak první udává číslo okna, které má být vymazáno a druhé udává o jaký typ 'vymazání' jde. Jestliže je vloženo pouze jedno číslo, je použito okno, s kterým se právě pracuje. V následující tabulce 'n' je číslo okna:

CLW_n,0	Vymaže okno (zaplní ho barvou permanentního papíru)
CLW_n,1	Zaplní okno barvou permanentního inkoustu
CLW_n,2	Změní INK v okně na PAPER a naopak
CLW_n,3	Vymaže pouze atributy

Ve všech případech se pozice pro tisk přesune do levého horního rohu.

Příkaz CLW vždy používá permanentní atributy z právě aktivního okna. Takže například je-li proveden příkaz CLW_0,0 a aktivní je okno 3, pak okno nula bude vymazáno s použitím atributu okna 3.

PAN a SCROLL

Pomocí těchto příkazů můžete pohybovat okna po jednom pixlu do všech směrů. PAN posunuje okno do stran, SCROLL posunuje okno nahoru nebo dolů. Za oběma příkazy musí být dvě čísla. První číslo udává, zda má být okraj okna vyplněn barvou inkoustu nebo papíru. Je-li toto číslo jedna, okraj je vyplněn inkoustem. Je-li nula, okraj je vyplněn barvou papíru. Další číslo udává, jak daleko a do jakého směru má být okno posunuto. Je-li u příkazu 'PAN_' druhé číslo kladné, okno se posune doprava. Je-li záporné, okno se posune doleva. U příkazu 'SCROLL_' - je-li druhé číslo kladné, okno se posunuje nahoru, je-li záporné, okno se posunuje dolů.

Také je možné, umožnit oknům, aby se po vyjetí z obrazovky objevila na stejném místě na protější straně (wraparound) použitím příkazu 'PANW_' a 'SCROLLW_'.

Pomocí příkazu FX můžete určovat, které výstupy se mají tisknout do kterých oken.

FX_0.n Určí okno, ve kterém se má zobrazovat zadávání příkazů a chybová hlášení
 FX_1.n Určí okno pro automatický listina
 FX_2.n Určí okno pro uživatelský výstup
 FX_3.n Určí okno, ve kterém se má zobrazovat číselní panel
 Takže např. 'FX_0.5' zajistí, že okno 5 bude zobrazovat chybová hlášení.

MODE

Znaky mohou být nyní tisknuty na obrazovce ve čtyřech velikostech - pomocí příkazu 'MODE_':

MODE_n,1 Vyvolá mód 64x24 znaků - maximální rozlišení
 MODE_n,2 Vyvolá standardní velikost znaků
 MODE_n,3 Vyvolá znaky s dvojitou výškou
 MODE_n,4 Vyvolá znaky s dvojitou výškou i šířkou. Při této velikosti můžete použít stínování (viz dále)

Za slovem 'MODE_' může být jedno nebo dvě čísla (jako u příkazu CLW). V jednom okně se může vyskytovat několik velikostí najednou.

Při práci v módu 4 můžete používat 'STIPLOVÁNÍ', při kterém znaky vypadají stínovaně. Uvědomte si, že v módu 4 jsou znaky 4x větší a tudíž každý pixel zabere ve skutečnosti pixly čtyři. Stiplování umožní určit sestavu těchto 4 pixelů, čímž vzniká efekt stínování. Sestava pixelů se zadává příkazem 'STIPPLE_' a číslem z intervalu nula až 15. Čím je číslo vyšší, tím tmavší je stínování.

FONT

MegaSpectrum má v paměti tři druhy písma. Druh písma, se kterým má Spectrum pracovat, se určuje příkazem FONT:

FONT_0 Vyvolá standardní znak
 FONT_1 Vyvolá 1. set (stejná písma jako na BBC Micro nebo na Acorn Electron)-uložen od adresy 49000
 FONT_2 Vyvolá 2. set (jako Amstrad CPC 464)-od adr. 45000

Příkaz FONT změní charaktery pro VŠECHNY tisky, nikoli: pouze pro jedno okno.

CHR\$,VDU a DOWN

MegaSpectrum má několik nových řídicích znaků:

CHR\$ 1-4 Jako MODE 1-4
 CHR\$ 7 Zinvertuje znak pod kurzorem
 CHR\$ 24-31 Určí okno, které má být použito pro výstup - např.
 CHR\$ 24 = okno 0 ; CHR\$ 31 = okno 7
 Nový příkaz VDU provede totéž jako příkaz PRINT CHR\$. Například:
 VDU_2 Přepne na standardní velikost znaků
 VDU_65.66 Vytiskne 'AB'

Řetězec znaků může být vytisknut použitím příkazu DOWN ve formě:

DOWN_y,x,a\$

Kde y je řádek, x je sloupec a a\$ je řetězec, který má být vytisknut. Jestliže řetězec dosáhne spodního okraje okna, tisk dále pokračuje opět od vrchu okna.

SPRINT a PRINTER

Příkaz SPRINT umožňuje tisk znaků v jakékoliv velikosti. Jeho forma je: SPRINT_x,y,a,b,a\$

Kde x a y udávají pozici (v pixlech) odkud má začít tisk, 'a' a 'b' udávají zvětšení ve směru x a y, a a\$ je řetězec, který má být vytisknut. (U příkazu SPRINT je pozice 0,0 v levém HORNÍM rohu)

Příkaz PRINTER umožňuje výstup na periferní zařízení (např. na tiskárnu). Za příkazem následuje číslo. Jestliže toto číslo je jiné než nula, všechny další výstupy půjdou místo na obrazovku na periferní zařízení. Uživatel musí do Spectra vložit podprogram ve strojovém kódu pro výstup znaků na zvolené periferní zařízení. Adresa tohoto podprogramu musí být uložena na adrese 59934/59935. Znak, který má být vytištěn, je při volání podprogramu uložen v akumulátoru. Podprogram musí být ukončen instrukcí RET.

***** POZOR *****

Při pokusu o použití 'CLEAR #', 'OPEN #2' nebo 'CLOSE' se systém zhroutí !

G R A F I K A
cccccccccccc

CHANGE a SWAP

Příkaz CHANGE umožňuje změnu určitých částí atributů. Za příkazem následují dvě čísla - 'maska' a data. Maska určuje bity, které mají být v každém atributu měněny. Data určují, k jakým změnám má dojít.

CHANGE_1 Maska je negována (??) - všechny jedničky se změní na nuly a naopak
 CHANGE_2 Každý byte atributu je ANDován s negovanou maskou
 CHANGE_3 Každý byte atributu je ORován s data bytem

Příkaz SWAP je následován dvěma čísly v rozmezí 0-255. Všechny atributy, které mají hodnotu prvního čísla, jsou nahrazeny druhou hodnotou.

FADE

Příkaz FADE je určen k produkci efektů. Následuje za ním číslo *x* v rozsahu 0-255. Počítač projíždí všechny atributy a ty, které nemají hodnotu *x*, dekrementuje. Toto opakuje tak dlouho, dokud všechny atributy nemají hodnotu *x*.

INVERT

U atributů celé obrazovky zamění PAPER za INK a naopak.

DEFG

Definuje UDG znaky. Má formát:

DEFG_a#,b1,b2,b3,b4,b5,b6,b7,b8

Předefinuje definovatelný znak obsazený v a# podle bytu b1 až b8. Např. DEFG "H",255,255,255,255,255,255,255,255 nadefinuje grafické "H" jako plný čtverec.

GET a PUT

Je možné uchovávat část obrazovky v paměti a potom ji vrátit zpět na obrazovku, do jiné pozice. Toto umožňují dva příkazy: GET a PUT. 'GET' uloží obrazovku do paměti a 'PUT' uloží obsah paměti na obrazovku. Příkaz 'GET' má tvar:

GET_0,a,y,x,d,w

Kde 'a' je adresa v paměti, od které se má obrazovka ukládat, 'y' je číslo řádky levého horního rohu oblasti, která má být uložena, 'x' je číslo sloupce levého horního rohu oblasti, která má být uložena, 'd' je výška oblasti, která má být uložena a 'w' je šířka oblasti.

Příkazy PUT a GET používají stejné souřadnice, jako např. standardní SCREEN# nebo ATTR. 'GET' ukládá do paměti nejdříve display file a potom atributy. Použijte příkaz CLEAR k vyhrazení části paměti pro uchování obrazovky. Můžete vypočítat počet bytů, které zabere uchovaná část obrazovky podle výrazu:

9 * w * d

Příkaz 'PUT' má tvar

PUT_f,a,y,x,d,w

Kde f udává způsob, jakým má být obsah paměti uložen zpět na obrazovku.

f=0 Obrazovka je přepsána obsahem paměti

f=1 Obrazovka je ORována s obsahem paměti

f=2 Obrazovka je XORována s paměti

f=4 Jako f=0, ale jsou zachovány původní atributy

f=5 Jako f=1, ale jsou zachovány původní atributy

f=6 Jako f=2, ale jsou zachovány původní atributy

Proměnné 'a', 'y', 'x', 'd' a 'w' mají tentýž význam jako u příkazu GET.

SPUT

SPUT je variace příkazu PUT. Příkazem 'SPUT' můžete obrazovku uloženou v paměti zvětšovat. SPUT má formu

SPUT_a,x,y,b,c,w,d

Kde 'a' je adresa začátku bloku, 'x' a 'y' jsou souřadnice levého horního rohu (SPUT používá tentýž souřadnicový systém jako SPRINT), 'b' a 'c' udávají, kolikrát má být uchovaný blok zvětšen ve směru 'x' a 'y', 'w' je šířka bloku (jako u GET) a 'd' je výška bloku (v pixlech! - t.j. 8 krát více, než u GET) SPUT vždy zachová původní atributy.

ŘÍZENÍ BĚHU PROGRAMU

PROCEDURY

YS MegaBasic vybavil Spectrum procedurami s parametry. Nevýhodou těchto procedur je, že neumožňují používat lokální proměnné. Začátek procedury je označen symbolem '@' (zavináč) a jménem procedury. Tento příkaz musí být prvním na řádku. Jestliže procedura používá parametry, musí za jménem procedury následovat podtržení '_' a názvy proměnných oddělené čárkami. Konec procedury je označen příkazem 'ENDPROC'. Pro přehlednost je možné přidat za ENDPROC jméno ukončované procedury (např. ENDPROC_DISP).

Příklad procedury:

```
9000 @DISPLAY_a,a$
9010 PAPER a:INK 9
9020 MODE_4:STIPPLE_6
9030 PRINT a$
9040 ENDPROC_DISPLAY
```

Řádek 9000 definuje proceduru DISPLAY a určuje, že je potřebná jednak numerická proměnná (a), jednak řetězcová (a\$). Řádek 9010 nastaví potřebné barvy a řádek 9020 nastaví velikost znaků a druh šrafování. Řádek 9030 vytiskne dodaný řetězec a řádek 9040 proceduru ukončí. K aktivování procedury použijte kdekoliv v programu např. 'DISPLAY_2."MEGABASIC"'. Procedury mohou být volány pouze v rámci programu, takže nemohou být použity jako přímé příkazy.

REPEAT-UNTIL

YS MegaBasic umožňuje používat smyčky typu 'REPEAT-UNTIL'. Příkaz REPEAT označuje začátek smyčky a za příkazem 'UNTIL_' následuje numerický výraz. Jestliže je hodnota tohoto výrazu nula, program se vrátí za poslední příkaz REPEAT. Jestliže je výraz nenulový, program pokračuje na dalším řádku. Těchto smyček může být vnořeno až 10 do sebe.

STACK

Stack je používán procedurami a smyčkami 'REPEAT-UNTIL' k uložení čísla řádku a čísla příkazu na řádku. Při volání procedury, nebo při provádění příkazu 'REPEAT', je na stack uloženo číslo řádku a číslo následujícího příkazu. Při provádění příkazu ENDPROC, nebo je-li parametr příkazu UNTIL pravdivý, program skočí na příkaz, jehož číslo a číslo jeho řádku je uloženo na stacku a tato hodnota je ze stacku odebrána. Jestliže na stacku žádná hodnota není, Spectrum ohlásí chybu. Na stack se vejde celkem 10 hodnot...pokusíte-li se uložit jedenáctou, Spectrum ohlásí chybu 'PROC stack overflow'.

POP a PUSH

Příkaz POP vybere hodnotu ze stacku, PUSH uloží hodnotu na stack. Za příkazem PUSH následují dva numerické výrazy: číslo příkazu na řádku, číslo řádku. Každý, kdo používá procedury nebo smyčky 'REPEAT-UNTIL', by měl na začátku svých programů používat příkaz PCLEAR k vynulování stacku.

BRANCH

Příkaz BRANCH vám umožní po provedení každého programového řádku provést určený podprogram. Za příkazem 'BRANCH_' následuje numerický výraz, který udává číslo prvního řádku podprogramu. Jestliže je tento výraz nulový, Spectrum bude provádět program normálním způsobem. Konec podprogramu je označen příkazem 'ENDPROC_'.

MTASK

Příkaz MTASK umožňuje provádění programu ze dvou míst najednou. Představte si, že se program skládá ze dvou částí: První začíná za příkazem MTASK a druhá na řádku, jehož číslo následuje za příkazem MTASK. Jakmile dojde program k řádku s příkazem MTASK, je dále program prováděn tak, že se provádí střídavě po jednom řádku z obou částí programu. Jestliže je výraz za příkazem MTASK nulový, program se dále provádí normálním způsobem. Nemůžete současně používat příkazů MTASK a BRANCH. Když používáte MTASK, za všemi příkazy, které pracují se ZX Interface 1 musí následovat '::'.

EDITOVÁNÍ a SLEDOVÁNÍ BĚHU PROGRAMU

TRON, TROFF a SPEED

Použijete-li příkaz TRON, pak při běhu programu bude v levém dolním rohu obrazovky zobrazováno číslo právě prováděného řádku. Příkaz TROFF toto zobrazování zruší.

Příkaz SPEED má tvar

SPEED_x

Kde 'x' je rychlost provádění programu (@=plná rychlost, 254=nejpomalejší). Jestliže je 'x' rovno 255, Spectrum se po provedení každého řádku zastaví a pokračuje až po stisknutí tlačítka.

AUTO a DELETE

AUTO_x,y provede automatické číslování řádků od x s přírůstkem y. Automatické číslování zrušíte stisknutím Extended Symbol Shift 'L'.

DELETE_x,y vymaže programové řádky od x do y.

BRON, BROFF a RESTART

Použijete-li na začátku programu příkaz BROFF, program nemůže být zastaven tlačítkem BREAK. Zastavení programu umožníte příkazem BRON.

Příkaz RESTART je ekvivalentem příkazu ON ERROR GOTO u jiných počítačů. Číslo řádku, na který má program skočit při výskytu chyby, následuje za příkazem 'RESTART_'. Po provedení příkazu 'RESTART_OFF' bude Spectrum fungovat normálním způsobem. Příkaz RESTART nezachytí chyby Interface 1 ani nové chyby MegaBasicu. Je-li chyba zachycena, některé užitečné informace jsou uloženy na těchto adresách v paměti:

59873/4 Řádek, na kterém došlo k chybě
59875 Číslo příkazu (v řádku), ve kterém došlo k chybě
59862 Kód chyby ke které došlo

YS MegaBasic umožňuje dva nové způsoby ovládání zvuku: příkaz PLAY a ISG (interrupt sound generator). PLAY má tvar:

PLAY_n,l,s,d,f

Kde 'n' určuje čisté tóny (n=0) nebo sumy (n=1), 'l' je délka každého kroku, 's' je počáteční frekvence, 'd' je počet kroků a 'f' je změna frekvence po každém kroku. Hodnoty 'd' a 'f' mohou být několikrát opakovány v jednom příkazu.

INTERRUPT SOUND GENERATOR

Zatímco PLAY je pouze vylepšený příkaz BEEP, YS MegaBasic umožňuje, aby zvuk zněl, zatímco je prováděn program. ISG je ovládán těmito příkazy:

SOFF Vypne ISG

SON ISG pokračuje v činnosti tam, kde přestal při SOFF

SREP_n Vyžaduje numerický výraz (n). Jestliže je výraz nulový, data ve zvukovém bufferu jsou hrána jen jednou. Jestliže je výsledek jedna, obsah bufferu se opakuje stále dokola.

SOUND_ Tento poměrně složitý příkaz je používán k ovládání zvukového bufferu. Má tvar:

SOUND_n,a,b,c,d

Kde: 'n' vymaže zvukový buffer před tím, než se do něj uloží něco nového (n=0), nebo přidá zvuk do bufferu (n=1). 'a' určuje čisté tóny (a=0), nebo sumy (a=1), 'b' je hodnota, která je přidávána k frekvenci po každém kroku, 'c' je počet kroků v sekvenci a 'd' udává, kolikrát se má sekvence opakovat.

Když je prováděn příkaz PLAY, ISG je automaticky vyřazen z činnosti. Takže, než může být produkován jakýkoliv zvuk, musíte použít 'SON' k aktivování ISG (všimněte si také, že ISG nefunguje při editaci). Čím je zvuk složitější, tím pomaleji se program provádí. Zde je příklad práce ISG:

```
10 SOUND_0,0,1,20,255
20 SREP_1
30 SON
40 MODE_4:STIPPLE_6:FONT_2
50 VDU_(128+RND*15)
60 PAPER RND*7:INK 9
70 GO TO 50
```

Řádky 10 až 30 aktivují ISG a řádky 40 až 70 provádějí vedlejší činnost.

STROJOVÝ KÓD A YS MEGABASIC

DUKE a CALL

Při práci se strojovým kódem v MegaBasicu je důležité uvědomit si dvě fakta:

Oblast paměti nad adresou 45000 je využita YS MegaBasicem, takže do ní nemůžete ukládat programy ve strojovém kódu.

YS MegaBasic využívá Interrupt mode 2, takže programy, které ho využívají, budou potřebovat jisté změny.

Když pracujete s procesorem Z80, často je nutné ukládat do paměti dvoubajtová čísla. V MS MegsBasicu je možné toto provádět příkazem DOKE. Čísla jsou ukládána ve standardní notaci Z80, tj. nejprve dolní, potom horní bajt.

Příkaz CALL je používán k volání podprogramů ve strojovém kódu. Za příkazem 'CALL_' následuje startovní adresa strojového kódu. Za adresou může následovat libovolný počet numerických výrazů, které budou při volání uloženy na strojový stack.

ČELNÍ PANEĽ

Čelní panel umožňuje měnit obsah paměti a registrů za použití hexadecimálních čísel. Čelní panel je aktivován příkazem MDH nebo stisknutím SPACE a 'F' při běhu programu. Čelní panel využívá okno 3, které musí mít velikost alespoň 40 sloupců krát 20 řádků.

Čelní panel je řízen několika jednopísmennými příkazy:

- SPACE Návrat do Basicu
 - R nn Uloží do určeného registru šestnáctibitové číslo
 - P Změní 'určený' registr
 - L nn nn nn Přemístí blok paměti. První číslo určuje začátek bloku, druhé určuje adresu, kam má být blok přemístěn a třetí určuje délku bloku.
 - M nn Zobrazí jinou část paměti
 - S Nastaví bod přerušení
 - K Pokračuje v provádění programu za bodem přerušení
 - U Zruší bod přerušení
 - I nn nn n Zaplní oblast paměti. První číslo udává začátek oblasti, druhé délku oblasti, třetí udává bajt, kterým má být oblast zaplněna.
 - J nn Zavolá podprogram ve strojovém kódu na adrese nn
 - ENTER Posune zobrazení úseku paměti o jeden bajt
 - '-' Posune zobrazení úseku paměti o jeden bajt zpět
- Jestliže vložíte dvě hexadecimální číselice, bude tato hodnota vložena do paměti a zobrazení se posune o jeden bajt.

SPIAJTY +++++

MegaSpectrum má 8 sprajtů číslovaných 0 až 7.

Příkaz SPRON je používán k jejich aktivování. Je následován dvěma numerickými výrazy - první udává, který sprajt má být aktivován, druhý udává, jakým způsobem má být sprajt kreslen na obrazovce. Jednička udává, že sprajt bude QRován, dvojka udává, že bude XORován. Příkaz SPRUFF_n vypne sprajt n.

Sprajty v MegaBasicu mají velikost 16x16 pixelů a každá skupina 8x8 pixelů může mít svůj vlastní atribut. Pro definování sprajtů jsou využity dvě oblasti paměti. První oblast (56750-56823) obsahuje všechny informace o sprajtu, kromě jeho tvaru a barvy. Každý sprajt používá 18 bajtů. Počáteční adresu informace o sprajtu 'n' zjistíte podle vzorce:

adr=56750+18*n

Nůd sprajtu je určen obsahem adresy 's+w'. Sprajt je buď QRován s obrazovkou (s=1), XORován s obrazovkou (s=2), nebo je sprajt vypnutý (s=0) tj. není kreslen na obrazovku. Konec oblasti informací o sprajtech je označen číslem 255. Dobrý způsob, jak vypnout všechny sprajty najednou, je POKE 56750,255

HI - T

=====
Copyright Timedata Ltd. 1984

Program HI-T umožní Vašemu 16 nebo 48 kB Spectru zobrazovat až 32 řádek po 64 znacích. Využívá se při tom nový 133-znakový soubor zmenšených písmen a číslic (každý znak v matici 4 * 6 bodů), který však můžete používat zároveň s původním znakovým souborem Spectra.

Používáte-li znakový soubor HI-T, můžete také definovat na obrazovce různá "okénka" do kterých se bude tisknout, používat horních i dolních indexů, předefinovat počet mezer mezi začátky tiskových pozic u položek, oddělovaných čárkou v příkazu PRINT, a konečně, pro vstup dat na kterémkoliv místě obrazovky využívat příkaz INPUT AT.

Nahrávání HI-T

Nahrávání HI-T z pásky je zcela standardní pomocí příkazu LOAD "HI-T" nebo LOAD "...". Pokud jej máte uložen na mikrodrajvu, použijte nejprve CLEAR #.

Po nahrání programu HI-T do paměti počítače se automaticky čistí oblast, kam se ukládá BASICový program a proměnné, takže před vkládáním Vašeho programu není nutno použít příkaz NEW.

Ba co více, používat povel NEW je nutno velice obezřetně. Tento povel sice program HI-T z paměti počítače nevymazává, avšak rozpojí jeho vazbu s podprogramy pro tisk uloženými v ROM. Pokud se tak stalo, dá se to však snadno napravit následujícím povelom: RANDOMIZEUSR (1 + PEEK 23730 + 256 * PEEK 23731).

"Srdcem" HI-T je 1.5 kB relokativní strojový program, který se po nahrání automaticky umístí těsně pod RAMTOP a RAMTOP se pak přesune pod něj (připomínáme, že RAMTOP je hodnota adresy, kam až smí zasahovat oblast BASICOVÉHO programu - tato hodnota je uložena ve stejnojmenné systémové proměnné a můžete ji obdržet pomocí PEEK 23730 + 256 * PEEK 23731).

Nejlépe je proto nahrávat HI-T do počítače v následující posloupnosti:

1. resetovat počítač (nebo odeslat RANDUSR 0);
2. do počítače nahrát cokoliv, co má být umístěno na vrcholu paměti (uživatелеm definovaný soubor znaků, programy ve strojovém kódu);
3. nahrát do paměti počítače program HI-T
4. do počítače nahrát nebo "natukat" BASICový program.

Uložit kopii programu HI-T na magnetofon lze následujícím jednoduším způsobem:

1. resetovat počítač (RANDUSR 0);
2. do počítače nahrát HI-T povelom: MERGE "HI-T"
3. pomocí povelu SAVE "HI-T" LINE 1 uložit HI-T na pásku, nebo povelom SAVE "*"m";1;"HI-T" LINE 1 jej uložit na mikrodrajvu.

Použití HI-T

Poznamenejme, že pro dobré zobrazení malých znaků vytvářených programem HI-T je třeba pečlivě naladit Váš televizor.

HI-T ve svých příkazech PRINT, INPUT a LIST otevírá proud (stream) #4. Pro výstup dat na obrazovku tedy používá PRINT #4, pro vstup INPUT #4 a pro vypisování programu LIST #4. Tyto příkazy se používají zcela shodně jako standardní příkazy PRINT, INPUT a LIST s tím rozdílem, že zobrazované znaky jsou menší (o dalších rozdílech si povíme níže), např.:

```
PRINT #4; "to je HI-T"
INPUT #4; "zadejte cislo"; a
LIST #4; 100 : REM vypisuje program od radky 100
```

Nastavení pozice tisku na obrazovce

Při použití HI-T si počítač "pamatuje" dvě různé pozice pro tisk znaků na obrazovku: jednak tu pozici, kam je nastaven kurzor pro standardní příkaz PRINT, a jednak pozici, kam bude tisknout PRINT #4.

V příkazu PRINT #4 proto můžete použít "AT řádka, sloupec" s tím, že pořadová čísla řádků jsou v rozmezí 0 až 31, a čísla sloupců od 0 do 63. Na začátku programu je tisková pozice nastavena do levého horního rohu - na řádek 0 a sloupec 0.

Povel CLS, který vymazává obrazovku a nastavuje tiskovou pozici pro normální 32-sloupcový výstup na řádek 0 a sloupec 0, nemá vliv na změnu pozice kurzoru pro malý 64-sloupcový tisk. Chceme-li nastavit kurzor do levého horního rohu obrazovky, můžeme to udělat dvěma způsoby, buď pomocí AT nebo s využitím řídicího znaku CHR\$ 0 :

```
PRINT #4; AT 0,0; "Zde!"
PRINT #4; CHR$ 0; "Zde!"
```

Vstup dat

Příkaz INPUT #4 umožňuje umístit vstup dat na kterékoliv místo na obrazovce (tedy ne pouze do dvou dolních "komunikačních" řádků). Krom toho, po provedení tohoto příkazu to, co jste odeslali do počítače z obrazovky nezmizí. Tím je mimo jiné umožněna snadná implementace tzv. "formulářového" vstupu dat, např.:

```
100 REM tisk formulare pro vstup dat
110 PRINT #4; AT 10,6; "Jmeno" "
120 PRINT #4; AT 11,6; "Prijmeni" "
130 PRINT #4; AT 12,6; "Bydliste-mesto" "
140 PRINT #4; AT 13,6; "Bydliste-ulice" "
.....
.....
.....
200 DIM v$(pocetdat,20): REM nulovani pole pro vstup dat
210 FOR i=1 TO pocetdat: REM postupny vstup dat
220 radek=9+i
230 INPUT #4; AT 10+i,21; v$: IF k$<" THEN j$=v$(i,1)
240 NEXT i
250 PRINT #4 AT 0,6;"Jsou data v poradku ? "
260 v$=INKEY$ ; IF v$="" THEN GOTO 260
270 IF v$="a" OR v$="A" THEN GOTO 300
280 PRINT #4; AT 0,6;"Zadna zmena = N/L " : GOTO 140
300 REM pokracovani programu ....
```

V příkazech PRINT #4 a INPUT #4 můžete použít povel INVERSE. Tak např. ve výše uvedeném příkladu můžeme formulář tisknout inverzně - stačí do příkazu na řádcích 110 - 140 vložit INVERSE :

```
110 PRINT #4; AT 10,6; INVERSE 1; "Jmeno" "
120 PRINT #4; AT 11,6; INVERSE 1; "Prijmeni" "
130 PRINT #4; AT 12,6; INVERSE 1; "Bydliste-mesto" "
140 PRINT #4; AT 13,6; INVERSE 1; "Bydliste-ulice" "
```

Na rozdíl od INVERSE však povely INK, PAPER, BRIGHT a FLASH nejsou v příkazech PRINT #4 a INPUT #4 rozpoznávány. Pokud ale přesto chcete změnit barvu podkladu nebo inkoustu, anebo nastavit blikání, nic vám nebrání změnit příslušné atributy pomocí standardních Basicových povelů. Před jejich použitím si nastavíme "přetisk" pomocí PRINT OVER 1. Tak např., chceme-li ve výše uvedeném příkladě, aby dotaz na potvrzení správnosti vložených dat žlutě blikal, vložíme řádky 245 (nastavuje blikání) a 290 (ruší blikání). Musíme přitom pamatovat na to, že 2 sloupce HI-T odpovídají 1 sloupci standardního Sinclair BASICu. Takže např. 6. sloupec HI-T odpovídá 3. sloupci ve standardním BASICu.

```
245 PRINT AT 0,3; OVER 1; PAPER 6; FLASH 1; " ";
290 PRINT AT 0,3; OVER 1; PAPER 7; FLASH 0; " ";
```

Tisk horních a dolních indexů

Tisknete-li malé znaky HI-T na obrazovce, můžete ovládat nejen tiskovou pozici podle řádku a sloupce, ale rovněž tisknout i "trochu nad" či "pod" zvolenou řádku - to Vám umožní psát horní a dolní indexy. Sekvence řídicích znaků CHR\$ 3; CHR\$ n v příkazech PRINT #4 či INPUT #4 umožní posunout tiskovou pozici o "n" bodů nad daný řádek (n je číslo v rozsahu od 0 do 5 - připomínáme, že znaky v HI-T jsou tištěny v matici 4 x 6 bodů). Zkuste např.:

```
PRINT #4; "TISK"; CHR$ 3; CHR$ 4; "horni-index"
```

Tisknout "pod" řádku můžeme tak, že tiskneme "nad" následující řádku:

```
10 CLS
20 PRINT #4; AT 10,10; "TISK"; CHR$ 3; CHR$ 4; "horni index"
30 PRINT #4; AT 11,14; CHR$ 3; CHR$ 2; "dolni index"
40 PRINT #4; AT 13 10; "Siranovy aniont ;"
40 PRINT #4; AT 15,10; "SO"; CHR$ 3; CHR$ 4; "2-";
50 PRINT #4; AT 16,12; CHR$ 3; CHR$ 2;"4"
```

Tabulátor

Použijeme-li ve standardním Sinclair BASICu mezi položkami v příkazu PRINT čárku (např. PRINT a,b,c,d), budou začátky zobrazovaných položek ve sloupcích 0 a 16. Interval mezi začátky zobrazovaných položek bude 16 sloupců.

Obdobně, použijeme-li čárku v příkazu PRINT #4, např.

```
PRINT #4 a,b,c,d
```

budou se jednotlivé položky tisknout do sloupců 0, 16, 32 a 48. Chceme-li však změnit tento šestnáctisloupcový interval, můžeme v příkazech PRINT #4 a INPUT #4 použít sekvence řídicích znaků :

```
CHR$ 5; CHR$ n
```

- kde n je požadovaný interval.

Tak například :

```
PRINT #4; CHR$ 5; CHR$ 10
```

nastaví hodnoty "tabulátoru" na sloupce 0, 10, 20, 30, 40, 50 a 60. A potom tedy hodnoty a, b, c, d se příkazem

```
PRINT #4 a,b,c,d
```

budou tisknout od nultého, desátého, dvacátého a třicátého sloupce.

Použijeme-li v příkazech PRINT #4 nebo INPUT #4 řídicí znak CHR\$0, nastaví se kurzor pro HI-T na pozici 0,0 a zároveň se interval mezi položkami nastaví na implicitní hodnotu 16 sloupců. Takže příkaz

```
PRINT #4; CHR$0; a,b,c,d
```

bude tisknout položky od nultého, šestnáctého, dvaatřicátého a osmačtyřicátého sloupce.

Okénka

Zkuste následující program :

```
10 FOR i=1 TO 100
20 PRINT "radka ";i;" -Sinclair BASIC"
30 NEXT i
50 PAUSE 100
60 CLS : PRINT #4; CHR$ 0.
70 FOR i=1 TO 100
80 PRINT "radka ";i;" - HI-T"
90 NEXT i
```

Potiskne-li se standardním příkazem PRINT celá obrazovka (resp. 22 řádek), výstup dat se zastaví s hlášením "scroll ?". Po stisknutí kteréhokoliv tlačítka (kromě N nebo SPACE) bude výstup dat pokračovat. Tiskne-li se na poslední nejspodnější řádku, provede se současně cyklicky posun obrazovky (scroll) o jednu řádku nahoru. To bude pokračovat tak dlouho, dokud se opět nezaplní celá obrazovka novými daty. Pak se opět počítač zeptá, zda má pokračovat v tisku.

Tiskneme-li však malé znaky příkazem PRINT #4 a potiskneme-li celou obrazovku, "odroluje" obsah obrazovky nahoru automaticky, bez čekání na stisk tlačítka. Automatický cyklický posun se dá zastavit stisknutím BREAK.

Cyklický posun se však nemusí týkat celé obrazovky. HI-T umožňuje definovat určitou část obrazovky - tzv. okénko, kam se bude nadále tisknout, a který také bude "rolovat"

Okénko se definuje pomocí sekvence řídicích znaků v příkazech PRINT #4 a INPUT #4 :

```
CHR$ 1; CHR$ h - nastavuje horní okraj okénka na řádku h;
CHR$ 2; CHR$ d - nastavuje dolní okraj okénka na řádku d;
CHR$ 21; CHR$ l - nastavuje levý okraj okénka na sloupec l;
CHR$ 12; CHR$ p - nastavuje pravý okraj okénka na sloupec p;
```

Aby "rolování" textu v okénku probíhalo správně, je třeba definovat levý okraj okénka na sudém sloupci, a pravý okraj na lichém sloupci.

Tak např. chceme-li definovat okénko :

```

horní okraj - řádka 10
levý okraj - pravý okraj
sloupec 20 - sloupec 45
dolní okraj - řádka 17

```

vložíme do předcházejícího programu řádky 65 a 66 :

```

65 LET w$=CHR$ 1 + CHR$ 10 + CHR$ 2 + CHR$ 17 + CHR$ 21
   CHR$ 20 + CHR$ 12 + CHR$ 45
66 PRINT #4; w$

```

Zrušení okénka (resp. jeho rozdělení na celou 32-řádkovou a 64-sloupcovou obrazovku provádí řídicí znak CHR\$ 0 v příkazech PRINT #4 nebo INPUT #4.

Výpis programu

HI-T lze vhodně využít pro výpis BASICových programů na obrazovku. Přehlednost programu se značně zvýší. Vidět na obrazovce najednou 22 řádek po 32 znacích nebo 32 řádek po 64 znacích je přeci jenom značný rozdíl. A co to znamená pro ladění "vzpurného" programu, který Vás nechce a nechce poslouchat, se přesvědčíte sami. Ale POZOR! Protože HI-T využívá některé znaky pro řídicí funkce (otevírání okének atd.), můžete systém zhroutit, budete-li příkazem LIST #4 vypisovat řádku, obsahující v příkazu REM instrukce strojového kódu! V takovém případě použijte raději standardní příkaz LIST.

Pro "rolování" vypisovaného programu na obrazovce platí stejná poznámka jako pro tisk pomocí PRINT #4: je-li obrazovka (nebo příslušné okénko) zaplněna, "roluje" se vesele dál, aniž by se čekalo na odpověď uživatele na otázku "scroll?". Přerušit "rolování" je možno pomocí BREAK.

Uložení a předefinování znakového souboru HI-T

Znakový soubor HI-T odpovídá znakovému souboru Spectra, včetně 21 uživatelem definovaných znaků (ASCII-kódy 144 až 164). Každý znak představuje matici 4 x 6 bodů, zakódovanou ve 3 bajtech :

```

- - - - - - první bajt, horní 4 bity
- - - - - - první bajt, dolní 4 bity
- - - - - - druhý bajt, horní 4 bity
- - - - - - druhý bajt, dolní 4 bity
- - - - - - třetí bajt, horní 4 bity
- - - - - - třetí bajt, dolní 4 bity

```

První bajt každého znaku je uložen na adrese :

RAMTOP + 1142 + 3 * ASCII-kód znaku

Takže, chceme-li např. předefinovat uživatelem definovaný grafický znak B (s ASCII-kódem 145) na ě, vložíme na příslušnou adresu 3 bajty, ve kterých je zakódován znak připomínající zmenšené háčkované ě. Nejprve si vypočteme hodnoty třech bajtů, do kterých se transformuje podoba definovaného znaku :

- X	- X	binární 0101 = 5	
- -	X -	binární 0010 = 2	1. bajt: 5*16+2 = 82
- X	- X	binární 0101 = 5	
- X	X X	binární 0111 = 7	2. bajt: 5*16+7 = 87
- X	- -	binární 0100 = 4	
- -	X X	binární 0011 = 3	3. bajt: 4*16+3 = 67

Vypočtené hodnoty pak vložíme na adresy, odpovídající uložení znaku s ASCII-kódem 145 ve znakovém souboru HI-T :

```
LET ramtop = PEEK 23730 + 256 * PEEK 23731
LET udg = ramtop + 1142 + 3*145
POKE udg,82: POKE udg+1,87: POKE udg+2,67
```

Obdobným způsobem je možno předefinovat nejen uživatelem definované znaky, ale i celý znakový soubor HI-T.

Nahrání se provede: LOAD "VU-CALC" nebo LOAD ""

VU-CALC je program pro počítání a zobrazení tabulek s uspořádanými čísly a jmény. Začíná se s prázdnou tabulkou uspořádanou do řádků a sloupců. Pomocí jednoduchých příkazů lze vložit vzorce, které udávají vztah mezi jednotlivými položkami, které mají být počítány, takže může být kalkulace provedena během několika sekund. Do tabulky lze vkládat data a jména do jednotlivých položek pro různé situace a změnou parametru lze získat upravené výpočty. Toto poskytuje účinný nástroj pro finanční, obchodní a jiné kalkulace jako pro vědecké tabulky, statistiku a pod.

Uspořádání-Tabulka, kursor a políčka.

Po vložení VU-CALC se objeví v horní části obrazovky dvoupříkazový panel a v dolní části je vstupní řádek. Prostřední část je okénko do tabulky, která je uspořádána do řádků a sloupců. Řádky jsou uspořádány abecedně a sloupce jsou číslovány nad uvedeným sloupcem. Každá položka je jednoznačně určena písmenem řádku a číslem sloupce. Tedy A01 nebo A1 označuje levý horní záznam tabulky.

V libovolné fázi použití VU-CALC je pozornost zaměřena k velkému červenému obdélníku. To je kursor. Kursorem je možno volně pohybovat po celé tabulce za účelem čtení nebo vkládání dat, textu nebo vzorců. Kursorem pohybujeme pomocí tlačítek se šipkami (5,6,7 a 8) při stisknutí CAPS SHIFT.

Dosáhne-li kursor krajní polohy, pak při stisknutí příslušného pohybového tlačítka se pohybuje celé okénko po tabulce. Toto lze pozorovat na změně čísel sloupců nebo písmen řádků. Takto lze obsáhnout velkou tabulku zejména při 48 kB.

Vkládání dat a textu.

VU-CALC lze považovat za tabulku, kde je možno libovolně umístit text nebo data. Jako prvořadý příkaz lze použít čtvř typů vstupu: text, data, vzorec nebo příkaz.

Pro vložení textu se umístí kursor do příslušného políčka, kde má začít text. Po stisknutí " lze jednoduše psát požadovaný text. Vkládaný text se postupně objevuje na vstupním řádku dole s malým červeným kursorem. Lze naplnit celý řádek. Změny lze provádět pomocí DELETE. Je-li text celý, stiskne se ENTER. Pro vkládání dat se umístí kursor na příslušné políčko a napíše se číselný údaj následovaný ENTER. Číslo se ihned zobrazí v označeném políčku.

K vypočítání výsledku a uložení do políčka se umístí kursor na požadované pole a vloží odpovídající vzorec. Je-li vzorec úplný, stiskne se ENTER. Tím je vzorec aplikován na dané políčko a okamžitě se provede kalkulace. Vzorec může být aplikován na více políček pomocí příkazu #Repeat viz dále.

Je-li vstupní řádek prázdný, lze přejít na příkazy pomocí #. tím se zobrazí v horní části obrazovky příkazy, kterých lze použít. Vložením prvního písmene příkazu se příslušný příkaz provede.

VZORCE

Hlavní výhodou VU-CALC je použití vzorců pro položky, řady a sloupce, které mohou generovat data požadovaná pro tabulku. Syntaxe vzorců zahrnuje použití číselných konstant, referencí k číslům v ostatních položkách a jednoduché aritmetické operátory (+ - / *).

Indexování čísel v jiných polích se vždy provádí pomocí písmene řádku následované číslem sloupce. Při tvorbě vzorců lze považovat indexy políček za proměnné a vzorce za jednoduché algebraické výrazy používající tyto proměnné, konstanty a aritmetické operátory.

Některé příklady takových vzorců jsou:

B1*1.03 D12*(B2+1.5)/C1
D7 C7 => pro D7=123 a C7=456 => 123456

Vzorec se může vztahovat k jedné určité položce, nebo může být aplikován na celý sloupec nebo i v rámci bloku. Abychom tohoto dosáhli, používá se příkaz #R. Vzorce jsou vždy aplikovány relativně: při příkazu #Repeat a použití vzorce na více položek podél řádku, je sloupcový index ve vzorci vždy zvětšen o 1. Vzorec je aplikován postupně podél řádku. Na př. je-li vzorec 1.03*A1 v položce A2 opakovan podle řádku A, pak v položce A3 bude vzorec 1.03*A2 a v položce A4 bude 1.03*A3 atd. Stejný relativní postup se opakuje ve sloupcích nebo blocích, kde se písmena řádku ve vzorci opět zvětšují.

Přejete-li si indexovat absolutní položku, která se nemění při opakování, musí mít index položky ve vzorci prefix \$. Tak pro výše uvedený příklad vzorec 1.03*\$A1 se bude i při opakování v řadě vztahovat pouze k obsahu položky A1.

PŘÍKAZY

Ve VU-CALC existují různé příkazy pro plnění paměti z pásky, ukládání na pásku, pro vkládání nových dat nebo změnu dat. Tyto příkazy jsou zahájeny pomocí # na prázdném vstupním řádku, který je následován prvním znakem příslušného příkazu.

#B	Blank	-mazání běžné položky
#C	Calculate	-přeodčítání celé tabulky, požaduje se při změnách vzorců nebo dat v tabulce
#E	Edit	-umožňuje změnit vzorec na políčku označeném kurzorem
#F,c,f,i	Format	-určuje uspořádání ve sloupcích a je definován třemi parametry c, i, f. Parametr c musí být jedno až dvojciferné číslo nebo "A". V případě čísla se formát vztahuje pouze na uvedený sloupec, "A" platí pro celou tabulku. Parametr i určuje typ formátu (i - integer, f - formát se dvěma desetinnými místy, G - obecní formát). Parametr z určuje rovnání do sloupců (L - zleva, R - zprava).
#G,rc	Go	-přesunuje kurzor přímo na políčko rc
#L	Load	-vypřázdňuje obrazovku a ptě se na jméno bloku, který má být nahrán z pásky
#P	Print	-vypis obrazovku na tiskárnu
#Q	Quit	-vypřázdňuje formulář nebo ukončí program

Velký systémový soubor pro sběratele.

Ať sbíráte cokoli, tento soubor Vám dá dokonalý přehled o všech podrobnostech Vaší sbírky. Do souboru se vejde minimálně 600 záznamů, přičemž každý z nich může obsahovat až 9 údajů. Počítač seřídí všechny údaje podle Vámi stanoveného pořádku.

Program běží ve dvou fázích:

- 1) Nejdříve stanovíte, jakou strukturu má Váš soubor mít, aby počítač věděl, v jaké formě má záznamy ukládat.
- 2) Pak už jen zadáváte nebo doplňujete svůj seznam.

Program může pracovat úplně potichu, avšak můžete zvolit také zvukovou signalizaci:

- očekává-li počítač vstupní data,
- rozezná-li na vstupu chybné zadání (bručivý tón),
- každé stisknutí klávesy je potvrzeno krátkým pípnutím.

Zvukové efekty však program zpomalují.

Požadujete zvukovou signalizaci?

Prosím stiskněte "Y", jestliže ano, jinak stiskněte "N".

A nyní zadávejte, jak má Váš seznam vypadat:

- počet polí v jednom záznamu,
- jméno každého pole,
- délku každého pole,
- pořadí, ve kterém mají být záznamy uloženy.

Počítač pak zkonstruuje Váš displej.

Následuje příklad souboru pro filatelistu:

Katalogové číslo ...

Hodnota ...

Barva ...

Popis ...

Katalogová hodnota ...

V tomto příkladu má každý záznam 5 polí. Délka pole, která je k dispozici, je patrná ze světlejšího pozadí. - Celková délka záznamu je v tomto případě 37 úhozů. Takových záznamů může počítač uložit 800.

Pro kvalitu souboru je velmi důležité, aby pole a jejich délky byly voleny podle zkušenosti.

V našem příkladě můžete zkrácením pole "Popis" na 10 úhozů a pole "Barva" na 4 úhozy ušetřit 28 znaků. Počítač pak může uložit 1050 takových záznamů, to je o 250 více.

Kolik polí chcete ?

(až 9)

Prosím stiskněte klávesu od 1 do 9.

Počet polí ...

Je to tak správně ?

V případě změny stisknout "n", jinak nějakou jinou klávesu.

Prosím zadání jména pro každé pole
(např. "Hodnota", "Popis" apod.)

Jméno pole 1
(až 15 úhozů)

Prosím zadání délek polí
(maximálně 30 úhozů, celková délka záznamu maximálně 50 úhozů)

Které pole je rozhodující pro vyhledávání záznamů ?
(Od této chvíle bude vyznačeno červeně)

Od této chvíle je počítač připraven záznamy ukládat.
Podle zadané struktury je počítač schopen uložit až ... záznamů.
Stačí Vám to ?

Na tomto místě máte poslední možnost Vaše zobrazení modifikovat.
R - modifikovat , C - pokračovat dál

- 1 - nový záznam
- 2 - změna záznamu
- 3 - výmaz záznamu
- 4 - vyhledání záznamu
- 5 - výtisk záznamu
- 6 - "save"
- 7 - stop

Jaký příkaz ? Prosím stiskněte 1 - 7.

Návod na program "Plošné spoje * 1"

Všeobecné poznámky:

Původní program "Plošné spoje" umožňuje zadat 20 typů součástek, 100 prvků a 500 spojů. Pro usnadnění práce s tímto programem byla vytvořena nová verze tohoto programu, která má rozšířenou knihovnu typů na 120. Program má standardně vestvený katalog 108 typů různých součástek, bodů a řad bodů.

Seznam je navržen pro základní rastr 1,27 mm (palcový) respektive 1,25 mm (metrický). Zbýlých 12 typů si může uživatel doplnit sám podle vlastní momentální potřeby.

U rozměrnějších kondenzátorů nelze zadat pozici pinu č.1 větší jak 6, protože systém umístí součástku v základní poloze tak, že pin č.1 dá do bodu 6,6 a pouzdro by vyšlo mimo spoj. Proto vývody rozměrnějších kondenzátorů nejsou umístěny přesně ve středu pouzdra.

Při otáčení součástek pozor! Systém otáčí součástku vždy okolo pinu č.1. Pokud by vývody nebo pouzdro vyšly mimo obraz, systém se zhroutí! Proto umístěte součástku nejdříve do bezpečné vzdálenosti od okrajů desky.

Velikost desky je omezena na 128 x 128 bodů což v našem rastru odpovídá 160 x 160 mm nebo v palcovém provedení 162,56 x 162,56 mm.

Pokud vytváříte spoje ze strany součástek, odpovídá rozložení součástek při zobrazení (funkce 8) konfiguraci zadané ve funkci umíst'ování (funkce 4). Při vytváření spojů z opačné strany se obraz ve funkci 8 - zobrazení zrcadlově otočí kolem vodorovné osy. Součástku umístěnou vlevo dole je tedy třeba hledat vlevo nahore.

Při vytváření plošného spoje je vhodné použít nejdříve třídění. Ve funkci propojení systém vypisuje spoje, které již nevládnul udělat. Ty je pak možno zadat do druhé vrstvy spojů. Funkce "Znovu propojit" vytváří jinou kombinaci spojů. Především při propojování 3 bodů vybere 2 kratší spoje. Pokud je žádána jiná kombinace, tak nejdlejší spoj zruší a nahradí ho kratším. Počet spojů které nevládnul se mění. Při pokusech na jednoduchém propojení jsem zjistil, že po prvním použití funkce "znovu propojit" vyšla jedna z nejlepších kombinací. Potom byly několikrát kombinace horší a zase kombinace příznivá. Bylo by vhodné nahrát si počáteční stav seznamu spojů na kazetu a potom zkoušet, po kolika použitích funkce "znovu propojit" vyjde nejlepší kombinace. Vzhledem k tomu, že metoda přetváření spojů se nemění, je možné zreprodukovat kterýkoliv stav.

Ovládání některých funkcí:

Ve funkci "umíst'ování":

šipky nahoru a dolů mění součástku, na kterou systém ukazuje.

C - změna umístění ukázané součástky.

B - menu.

Při otáčení: 0 je základní poloha, každá další jednička otočí součástku o 90 stupňů doleva.

Ve funkci "zobrazení":

šipky posunují výřez zobrazení o 1 bod v odpovídajícím směru.

šipky + Caps shift posunují výřez zobrazení o 10 bodů.

B - menu

Knihovna typu součástek, které jsou k dispozici
 v programu "Plošné spoje * i"

a.) Integrované obvody DIL.

číslo	typ	parametry v programu
1	DIL 8	6 4 2 1 0 8 6
2	DIL 10	6 5 2 1 0 10 6
3	DIL 14	6 7 2 1 0 14 6
4	DIL 16	6 8 2 1 0 16 6
5	DIL 18	6 9 2 1 0 18 6
6	DIL 20	6 10 2 1 0 20 6
7	DIL 24	12 12 2 1 0 24 6
8	DIL 28	12 14 2 1 0 28 6
9	DIL 40	12 20 2 1 0 40 6

b.) Integrované obvody jednořadové, jako je např. 1SS1

číslo	typ	parametry v programu
10	1SS1 apod.	0 4 3 1 1 11 2

c.) Integrované obvody v kovovém pozdře - kulaté - řady vývodů
 jsou definovány blíž u sebe, než u DIL.

číslo	typ	parametry v programu
11	4 vývody	4 2 4 0 0 4 4
12	6 vývodů	4 3 2 0 0 4 4
13	8 vývodů	4 4 2 0 0 6 4
14	10 vývodů	4 5 2 0 0 8 4
15	12 vývodů	6 6 2 0 0 10 6

d.) Tranzistory

- s umístěním vývodů do trojúhelníku:
 uspořádání:

$$\begin{matrix} . & b & . \\ & c & . & e \end{matrix}$$

$$\begin{matrix} e = 3 \\ b = 5 \\ c = 1 \end{matrix}$$

číslo	typ	parametry v programu
16	malý typ (plastikový)	1 3 1 1 1 4 3
17	menší typ (kovový - KC)	1 3 1 1 2 4 4
18	větší typ (kovový - KF)	1 3 1 2 3 6 6

- s umístěním vývodů v řadě:
 uspořádání:

$$\begin{matrix} c & b & e \end{matrix}$$

$$\begin{matrix} e = 3 \\ b = 2 \\ c = 1 \end{matrix}$$

číslo	typ	parametry v programu							
19	miniaturní (rozteč 1.25 mm ¹)	0	3	1	1	1	4	2	
20	střední typ	0	3	2	1	2	6	4	
21	výkonové plastkové typy	0	3	2	2	3	8	6	

e.) Výkonové kovové tranzistory

-uspořádání:



e = 6
b = 4
c = 2

číslo	typ	parametry v programu							
22	menší	12	3	3	5	3	16	26	
24	větší	14	3	5	6	4	22	32	

-uspořádání:



e = 4
b = 6
c = 2

číslo	typ	parametry v programu							
23	menší	8	3	3	5	3	16	26	
25	větší	10	3	5	6	4	22	32	

f.) Diody

číslo	typ	parametry v programu							
26	miniaturní	0	2	6	0	1	6	2	
27	KY 130	0	2	8	0	1	8	3	
28	KY 132	0	2	12	0	2	12	4	
29	KY 701 apod.	0	2	48	0	5	48	10	

g.) Body a čtverce

číslo	typ	parametry v programu							
30	izolovaný bod	0	1	0	0	0	1	1	
31	1.25 x 1.25	1	2	1	0	0	1	1	
32	2.5 x 2.5	2	2	2	0	0	2	2	
33	3.75 x 3.75	3	2	3	0	0	3	3	
34	5 x 5	4	2	4	0	0	4	4	
35	6.25 x 6.25	5	2	5	0	0	5	5	
36	7.5 x 7.5	6	2	6	0	0	6	6	

37	10 x 10	8 2 8 0 0 8 8
38	12.5 x 12.5	10 2 10 0 0 10 10
39	15 x 15	12 2 12 0 0 12 12
40	17.5 x 17.5	14 2 14 0 0 14 14
41	20 x 20	16 2 16 0 0 16 16

h.) Odporý

číslo	typ	parametry v programu
42	miniaturní	0 2 6 0 1 6 2
43	0.25 W	0 2 10 0 2 10 4
44	0.5 W	0 2 14 0 2 14 4
45	1 W	0 2 22 0 4 22 8
46	2 W	0 2 44 0 4 44 8

i.) Potenciometry

-jednoduché (vývody 1 až 3 v základní poloze umístěny dole),
jezdec = 2

číslo	typ	parametry v programu
47	miniaturní - průměr 16	0 3 4 9 2 14 8
48	normální - průměr 28	0 3 8 4 2 24 10

-tandemové a dvojité (systém 1 - kraje 1, 3; jezdec 2; systém
2 - kraje 6, 4; jezdec 5)

číslo	typ	parametry v programu
49	průměr 28, horní systém na okraji pouzdra - tandemové	18 3 8 4 0 24 18
50	průměr 28 - dvojité	10 3 8 4 2 24 22
51	průměr 28, horní systém nad středem pouzdra - tandemové	12 3 8 4 2 24 22

j.) Odporové trimry

Uspořádání:

$\begin{matrix} . & j & . \\ & & \\ k & . & k \end{matrix}$

k - krajní vývody - 1 a 3
j - jezdec - 5

Rozměry udávají vzdálenost krajních vývodů (první číslo) a
vzdálenost jezce od středu mezi krajními vývody (druhé číslo).

číslo	typ	parametry v programu
52	miniaturní nastojato 10 x 5	4 3 4 2 2 12 8
53	miniaturní naležato 10 x 12.5	10 3 4 2 2 12 14
54	miniaturní nastojato 5 x 5	4 3 2 3 2 10 8
55	keramický naležato 5 x 10	8 3 2 3 2 10 12
56	normální naležato 10 x 17,5	14 3 4 2 0 12 14
57	normální nastojato 15 x 5	4 3 6 4 2 18 10

k.) Elektrolytické kondenzátory kovové

Poznámka - u všech rozměrů dvoupólových součástek značí první číslo šířku nebo průměr, druhé číslo značí délku.

číslo	typ	parametry v programu						
58	3.2 x 11	0	2	16	0	1	16	3
59	4.2 x 11	0	2	16	0	2	16	4
60	5.3 x 11	0	2	16	0	2	16	5
61	6.5 x 16	0	2	20	0	3	20	6
62	8.5 x 16	0	2	20	0	3	20	7
63	8.5 x 24	0	2	24	0	3	24	7
64	10 x 24	0	2	24	0	4	24	9
65	11.5 x 29	0	2	32	0	5	32	10
66	14 x 29	0	2	32	0	6	32	12

i.) Plastikové elektrolyty do plošných spojů

číslo	typ	parametry v programu						
67	4.5 x 7.5	0	2	2	2	2	6	4
68	6.5 x 9.5	0	2	2	3	3	8	5
69	10.5 x 13.5	0	2	4	4	4	12	8

j.) Tantalové elektrolyty válcové

číslo	typ	parametry v programu						
70	7.5 x 15	0	2	16	0	4	16	8

k.) Tantalové kapky, keramika

Vzdálenost vývodů je 2.5 mm.

číslo	typ	parametry v programu						
71	do délky 4.5	0	2	2	1	2	4	4
72	do délky 6	0	2	2	2	2	6	4
73	do délky 8	0	2	2	3	2	8	4

l.) Válcové kondenzátory MP

číslo	typ	parametry v programu						
74	9 x 19	0	2	20	0	4	20	8
75	12 x 26.5	0	2	24	0	5	24	10
76	14 x 29	0	2	28	0	6	28	12
77	16 x 33	0	2	32	0	6	32	14

m.) Polyesterové kondenzátory

číslo	typ	parametry v programu						
78	5 x 13	0	2	14	0	2	14	4
79	7.5x 13	0	2	14	0	3	14	6
80	10 x 13	0	2	14	0	4	14	8
81	5 x 16	0	2	16	0	2	16	4
82	7.5x 16	0	2	16	0	3	16	6
83	10 x 16	0	2	16	0	4	16	8
84	10 x 20	0	2	20	0	4	20	8
85	12 x 28	0	2	26	0	5	26	10
86	16 x 28	0	2	26	0	6	26	14
87	24 x 28	0	2	26	0	6	26	20
88	20 x 33	0	2	30	0	6	30	16
89	24 x 35	0	2	32	0	6	36	16
90	20 x 40	0	2	36	0	6	36	16

n.) Univerzální řady bodů pro kontaktní pole a konektory

- pro rozteč 2.5 mm respektive 2.54 mm (číslo udává počet bodů v řadě) - počty bodů jsou voleny podle konektoru FRB.

číslo	typ	parametry v programu						
91	10	0	10	2	0	0	18	0
92	15	0	15	2	0	0	28	0
93	24	0	24	2	0	0	46	0
94	31	0	32	2	0	0	60	0
95	36	0	36	2	0	0	70	0
96	45	0	45	2	0	0	80	0

- pro rozteč 3.75 mm respektive 3.81 mm

číslo	typ	parametry v programu						
97	6	0	6	3	0	0	15	0
98	10	0	10	3	0	0	27	0
99	15	0	15	3	0	0	42	0
100	24	0	24	3	0	0	69	0
101	30	0	30	3	0	0	87	0
102	36	0	36	3	0	0	105	0

- pro rozteč 5 mm respektive 5.08 mm

číslo	typ	parametry v programu						
103	4	0	4	4	0	0	12	0
104	6	0	6	4	0	0	20	0
105	10	0	10	4	0	0	36	0
106	15	0	15	4	0	0	56	0
107	21	0	24	4	0	0	92	0
108	26 (EC 1033 apod)	0	26	4	0	0	100	0

Opravy knihoven v programu "Plošné spoje"

Program "Plošné spoje" neumožňuje jednoduchou editaci a opravy údajů ve svých knihovnách. Nesprávnému vložení údajů se sice můžete vyhnout tím, že nepotvrdíte naposled vložené údaje. To ale může být škoda, zvláště, když vkládáte údaje ve větších dávkách. Dalším nedostatkem programu je, že neumožňuje uložit na kazetu momentální stav seznamu spojů. Proto uvádím údaje o knihovnách programu "Plošné spoje":

a.) Knihovna typů

- je uložena v poli e\$ (20,7), u nové verze *1 s rozměry (120,7). Každému řádku odpovídá 1 typ.

Pozice v řádku	Význam
1	hg - rozteč řad pinů
2	pn - počet pinů v řadě
3	dz - rozteč pinů
4, 5	xx, yy - souřadnice pinu čís. 1 od levého spodního rohu pouzdra
6, 7	x1, y1 - rozměry pouzdra

Na poslední uložený typ ukazuje proměnná mxe

b.) Knihovna prvků

- je uložena v poli b\$ (100,7).

Pozice v řádku	Význam
1 až 3	jméno prvku
4	číslo typu v knihovně typů
5 až 7	při vytváření se dosazují čísla 0,6,6

Na poslední prvek ukazuje proměnná mxb

c.) Knihovna spojů

- je uložena v poli v\$ (500,7).

Pozice v řádku	Význam
1	číslo prvku "od"
2	číslo PINU "od"
3	číslo prvku "do"
4	číslo PINU "do"
5 až 7	při vytváření se nedosazují

Na poslední spoj ukazuje proměnná mxv

Příklady využití těchto údajů:

Uděláte-li chybu v zadávání a nechcete ztratit předtím zadané hodnoty, můžete vypustit posledních n prvků snížením řídicí proměnné pro daný seznam. Například chcete vypustit posledních 5 spojů:

```
LET mxv = mxv - 5
```

Jednotlivý údaj lze opravit přepsáním příslušného místa v seznamu. Například chcete opravit v seznamu prvků číslo typu u 27. prvku na 10:

```
LET b$(27,4) = CHR$(10)
```

Seznam spojů lze zaznamenat příkazem:

```
SAVE "jmeno" DATA v$()
```

Přitom je nutné vypsát si obsah proměnné mxv a poznamenat si ho. Pro obnovení seznamu spojů z kazety pak stačí:

```
LOAD "jmeno" DATA v$():LET mxv = xx (původní obsah)
```

Po provedených úpravách se lze dostat do menu příkazem:

```
GO TO 9200
```

INSTALACE SYSTÉMU

POČÍTAČ

LOAD "" CODE ENTER

Bytes: SpecForth
SPECT ZX-FORTH 1.1.
(C)1982 ARTIC COMPUTING LTD.
ALL RIGHTS RESERVED
32470 BYTES FREE

Po natažení programu FORTH stop maj. Následuje Editor

1 LOAD ENTER READY CASSETTE
 ENTER

start mgf-po natažení stop R MSG # 4
 READY CASSETTE

start mgf-ENTER-po natažení stop I MSG # 4
 READY CASSETTE

start mgf-ENTER-po natažení stop OK

Pokud vše proběhlo v tomto pořadí, je systém natažen a schopen provozu. Následují již stránky (screens) vlastní konstrukce.
4 LOAD ENTER atd.

Grafický znak (je to znak, který má code 136) je vaším kurzorem a označuje, kdy je systém READY, (připraven) pro vstup z klávesnice.

Nyní můžete zadávat slova a ukončovat stisknutím tlačítka ENTER. Než stisknete ENTER, můžete případně chybné zadání opravit nebo zrušit pomocí DELETE.

Stránkový editor se užívá při opravě chyb. Provádí se kopírováním správného textu. Pomocí kurzoru (Caps Shift + 5-8) najdeme začátek textu ke kopírování. Samotné kopírování provádíme tlačítkem EDIT (Caps Shift + 1).

Chybová hlášení.

- 0 není ve slovníku
- 1 prázdný zásobník
- 2 slovník přeplněn
- 3 nedovolený způsob adresace
- 4 jméno je použito (varování)
- 7 zásobník přeplněn
- 17 musí být použito v definici
- 18 nesmí být použito v definici
- 19 nevhodná podmínka
- 20 nedokončená definice
- 21 chráněný slovník
- 22 použito pouze při load
- 23 mimo rozsah aktuální stránky
- 24 deklarace slovníku

FORTH EDITOR

=====

Při interaktivní práci s jazykem FORTH nemáme možnost změnit nebo vypsat definice slov. Proto byl vytvořen EDITOR, který používá zápis na očíslované stránky. Jedna stránka (screen) obsahuje 16 řádků po 64 znacích. Programy (definice slov) se ukládají pomocí EDITORU do těchto stránek. Číslo stránky může být 0 až 32768. V paměti je uložena vždy jen jedna stránka a je označena jako aktuální stránka.

EDITOR vyvoláme zápisem: editor (ENTER)

PŘÍKAZY EDITORU.

```
n clear   vymaže a naformátuje stránku n
n list    určí aktuální stránku
n p....   zápis do řádku n
n e       zruší obsah řádku n
n d       zruší obsah řádku n, ostatní řádky nahoru
n s       uvolní řádku n, ostatní řádky dolů
n h       řádek n uložit do mezipaměti
n r       řádek n naplnit obsahem z mezipaměti
n i       řádek n naplnit obsahem z mezipaměti, ostatní řádky dolů

l         vylistuje aktuální stránku
```

Na konci stránek:

```
-->      přechod na další stránku
; s       poslední stránka
first?    vypíše číslo aktuální stránky
```

Změna čísla stránky:

```
n dup first l scr! (n=číslo nové stránky)
```

Kursor editoru. (podtržítkový kursor)

Tento kursor ulehčuje vyhledávání a opravu vadných částí textu na aktuální stránce.

```
top       kursor na začátek stránky
f text    hledá vpřed řetězec text, po nalezení umístí řádek do editační oblasti a vypíše číslo řádku. Nenašel-li vypíše ? MSG # 0.
b         používá se po f a vrací kursor o délku hledaného řetězce zpět.
n         hledá další výskyt textu dle f.
c text    kopíruje text na místo označené kurorem.
n m       přesun kursoru o n míst vpřed nebo vzad (-n).
till text vymaže řádky od kursoru až po text. Používáme pouze bezprostředně po posicování.
x text    vyhledá a zruší další výskyt řetězce text.
```

Zadáme-li c bez textu dojde ke stopu překladače, tuto chybu rušíme pomocí : top x (ENTER).

Ukládání programu.

Program (definice slov) zapsaný na aktuální stránce můžeme uložit na kazetu běžným postupem při zápisu. Příkazem:

```
flush (ENTER) zahájíme zápis  
systém sdělí: READY CASSETTE
```

Po zapnutí záznamu stiskneme Enter, sledujeme pruhu na obrazovce a po zprávě OK zastavíme záznam.

Takto zapsaná stránka se může zavést do systému pouze opět jako stejná stránka.

Natažení programu (stránky).

Příkaz : n list (n=číslo stránky)

Je-li stránka již obsazena pak následuje výpis stránky na obrazovce. Je-li prázdná, pak následuje výpis READY CASSETTE. Stiskneme ENTER a pustíme mgf. Nedojde-li k odpovědi OK nesouhlasí číslo stránky.

Příkaz : n load provede kompilaci definic na akt.stránce

ÚVOD

Program obsahuje 16k BASICu a 1165 bytů strojového kódu od adresy 62k.

Strojový podprogram programu DEBUGGER má za úkol z daného úseku paměti vzít byty, naplnit jimi registry, provést jednu instrukci a při tom musí nad ní zajistit kontrolu t.j. nasimulovat provedení různých skoků atd. a opět uklidit registry a dříve než bude provádět další instrukci, vrátí se do BASICu.

Jeho důležitým úkolem je obnovování registrů. V BASICovém manuálu je řečeno, že se nesmí používat registry I a IV, ovšem tyto lze použít, když zakážeme přerušení. Dále je tu registr HL', který musí mít při návratu do BASICu stejný obsah jako měl tehdy, když řízení bylo předáno do strojového podprogramu. V případě, že tato podmínka není splněna, systém může havarovat. Toto není v manuálu uvedeno, ale je to důležité. Může na tom ztroskotat většina složitějších programů, které používají čárkované registry. DEBUGGER se snaží prověřit váš program a umožní kontrolu, zda vaše programy těmto požadavkům vyhoví.

Příkaz INIT vynuluje registry mikroprocesoru, na což si není vhodné zvyknout a nesmíme z tohoto předpokladu vycházet při tvorbě programu. Důležité je, že do HL' se vloží nějaké číslo. v našem případě to je #AAAA. Potom se vloží do IV konstanta (normálně nesmíte měnit IV, ale v IV je vždy stejná konstanta).

Připraví se ukazatel zásobníku a na vrchol zásobníku se uloží návratová adresa do BASICu (#2B2B) a PC se nastaví na adresu vloženou vstupem INIT.

Program DEBUGGER má vlastní ukazatel zásobníku, aby nemohl být přemazáván ukazatelem zásobníku systému ZX Spectrum. Rozsah tohoto ukazatele zásobníku lze nastavit vložením dolní a horní hranice: viz seznam systémových proměnných.

Vždy, když začne probíhat podprogram ve strojovém kódu, uchová se obsah HL', I, IV a zakáže se přerušení, pak se provede vaše instrukce a opět se zakáže přerušení pro případ, že by vaše instrukce byla EI. Znamená to, že chybné nastavení IV při debuggování nic nezničí, ale kdyby váš program běžel bez DEBUGGERU, došlo by k havárii, neboť každou 1/50 sec. dochází k přerušení v masce IM 1. Adresa #2B2B umístěna v zásobníku DEBUGGERU slouží k návratu do BASICu (stejná adresa se ukládá i příkazemUSR). Když počítač zjistí, že se ocitl na adrese #2B2B, přeruší zrychlený běh zobrazování a ozve se pípnutí a v plném zobrazovacím režimu se na žlutém okraji objeví nápis:

SUCCESSFULL RETURN nebo FAILURE RETURN

Zpráva SUCCESSFULL RETURN vám říká, že byl nasimulován návrat do BASICu a registry HL' a IV jsou v pořádku. FAILURE RETURN znamená, že některý z registrů HL' nebo IV je špatně nastaven.

Další chybové hlášení, které přerušuje zrychlený běh:

ILLEGAL INSTRUCTION

Označuje, že ve vašem programu je instrukce, která není v seznamu instrukcí Z 80. Např. po prefixu #ED je řada instrukcí, které nejsou definovány a proto je DEBUGGER odmítá provést.

Zpráva:

PC or SP OUT OF THE RANGE

Znamená, že čítač adresy (PC) nebo ukazatel zásobníku (SP) je mimo meze.

PC je normálně nastaven od #0 do #FFFF a proto se (pokud nezměníte tyto hodnoty) hlášení objeví pouze v případě překročení SP, který je umístěn v paměti tiskárny (PRTBUF) a je omezen jeho rozsahem.

Použití dat mimo povolenou oblast není signalizováno. Standardně nastavené meze lze změnit nastavením systémových proměnných.

POUŽITÍ

Před LOAD programu DEBUGGER nastavíme RAMTOP na méně než 63488 příkazem CLEAR. Nahraje se 16k programu BASIC a asi 1k strojového kódu od adresy 63488. Program sám začne pracovat.

Na displeji se zobrazí ve žlutém rámečku dva podobné obrázky nad sebou. Horní obrázek je nedoplněnou variantou dolního. Dolní polovina obrazovky bude vyplněna v první řádce pod žlutým předělem písmeny:

AF BC DE HL

Jsou to označení registrů mikroprocesoru a sdružená do dvojic tak, aby vytvářely 16 bitový pár. Pod nimi v další řádce je uveden jejich obsah v hexadecimální soustavě. O řádce níže je uvedeno číslo jako 2 bytové slovo uložené na adrese, která je obsahem daného registru (nižší byte - LSB je na adrese, která je obsahem registru a více významný byte - MSB je na adrese +1).

Pod tímto je stejným způsobem zobrazen obsah 16 bitových registrů PC, SP, IX a IY.

Vpravo od zobrazených registrů je zobrazen obsah registru F znackami C, Z, S a P. Např. je-li zde C, pak je nastaven stavový registr CARRY. Stavové registry N a A se nezobrazují, protože jsou použity pouze v instrukci DAA. Pod stavovými registry se zobrazují registry určené pro periferie a přerušování a jsou označeny I= a R=.

Čárkované registry se pro nedostatek místa na displeji nezobrazují, ale lze je vyvolat.

Pod zobrazením registru je čtyřmístné hexadecimální číslo v závorce a za ním následují dvojčíslí v šestnáctkové soustavě. Toto je okénko (VIEW). Číslo v závorce můžeme změnit na libovolnou hodnotu a zobrazí se hodnoty bytů následujících za touto adresou.

Poslední řádka zobrazuje dekodovanou instrukci, na kterou ukazuje adresa v PC.

Horní pole, které je zatím nevyplněno, bude vždy obsahovat výpis registrů před provedením instrukce nebo příkazu.

Po provedení příkazu, kdy se provede jedna nebo více instrukcí zkoumaného programu, bude horní polovina obrazovky obsahovat výpis registrů a stavových registrů a také okénka i dekodovanou instrukci před provedením příkazu. Dolní polovina obrazovky bude zobrazovat stav po provedení příkazu a instrukci, která bude provedena jako následující.

PŘÍKAZY

Příkazy vkládáme stisknutím příslušné klávesy.

INIT =i (počátek) nastaví počáteční adresu zkoumaného programu, vymaže horní část obrazovky a až na výjimky (HL, IY) se všechny registry vynulují. V dolní části obrazovky se objeví INIT a #. což znamená, že máme vložit počáteční adresu, odkud bude DEBUGGER startovat v hexadec. soustavě. Číslice 10 až 15 můžeme zaplat malými písmeny a až i, při použití velkých písmen je třeba dát pozor, aby nezůstal

nastaven C (CAPS LOCK) při stisknutí ENTER. Program by pak mohl havarovat.

EXT =n (příští) provede jednu následující instrukci vašeho programu. Pokud je to CALL nebo JUMP, provede její a hned zobrazí první instrukci podprogramu nebo instrukci na adrese skoku. To umožní krokovat podprogramy a skoky. Po skončení příkazu NEXT je v horní polovině obrazovky stav registru před provedením NEXT a v dolní polovině po provedení NEXT. Tento způsob je nazván režimem plného zobrazení. Nyní může následovat další příkaz (i NEXT) ap.

GO =g (jdi) je příkaz, který zrychlí činnost DEBUGGERU, ale nepracuje v režimu plného zobrazení. Zobrazí pouze adresy a dekodované instrukce, které procházejí po obrazovce zdola nahoru. Je to režim zrychleného zobrazení, kde se zobrazují již provedené instrukce.

Pro volbu rychlosti tohoto zobrazení můžeme vždy použít klávesy 0, 1 a 2 (čím je menší číslice, tím rydlejší je činnost).

Při stisknutí 2 se zobrazí SLOW a další instrukce se provede jen když stisknete N nebo ENTER.

Při stisknutí 1 se zobrazí NORMAL. Instrukce dlynule prochází po obrazovce a provádí se. Do tohoto režimu se počítač přepne příkazem GO.

Při stisknutí 0 se zobrazí nápis FAST, instrukce se provádějí nejrychleji, ale nezobrazují se.

Jednotlivé režimy můžeme podle potřeby volit a střídat.

Příkaz GO ukončíme stisknutím SPACE (bez SHIFT). Program přejde do režimu plného zobrazení, kde v horní polovině obrazovky bude stav registrů před provedením příkazu GO a v dolní je současný stav registrů a následující instrukce.

V případě, že po displeji začnou procházet podivné a vám neznámé instrukce, lze běh zastavit BREAK + CAPS SHIFT

BREAK =b (místo přerušení). Dole na displeji se vám objeví BREAKPOINT + #, zadáte hexadecimální adresu, na které chcete přejít do režimu plného zobrazení a klávesou ENTER spustíte běh programu. V režimu zrychleného zobrazení se pokračuje až k instrukci, jejíž adresa je rovna adrese zadané příkazem BREAKPOINT, proto je třeba zadat adresu přesně a před provedením instrukce přejde do režimu plného zobrazení. Příkaz lze ukončit před zadanou adresou stisknutím PAUSE. Jinak je možné volit rychlost zobrazení 0, 1 a 2 jako v příkazu GO.

Příkazy BREAKPOINT a GO je vhodné použít pokud víte, že určitá oblast programu je bez chyby.

CALL =c (volání). Usnadňuje volání podprogramu. Pokud v plném zobrazovací režimu vidíte, že má být proveden podprogram a nechcete-li jej prohlížet, neboť víte, že je správný, stisknete C a počítač přejde do rychlého zobrazovacího režimu a ten běží tak dlouho, dokud se nevrátí z podprogramu, při tom podprogram může mít několik úrovní podprogramů a až návrat z volaného podprogramu přenechá zobrazovací režim.

Tento příkaz pracuje i když to co provádíte není volání podprogramu, pokud je to obyčejná instrukce např. ADD nebo LD, instrukce se provede a pouze ji uvidíte prokliknout na vrchu obrazovky, pokud je to CALL s podmínkou a tato není splněna, provede se opět pouze tato jediná instrukce a opět se přenechá na plný zobrazovací režim.

RETURN =r (návrat). Po stisknutí R, počítač přejde do zrychleného zobrazovacího režimu a vrátí se do režimu plného zobrazení

až po návratu z podprogramu, ve kterém jste příkaz použil. Je vhodné příkaz RETURN použít v případě, že vyvoláte podprogram a pak nabydete dojmu, že je dále v pořádku a chcete se vrátit.

Rozdíl mezi příkazy CALL a RETURN je v tom, že CALL v režimu zrychleného zobrazení provede podprogram, který má být právě proveden a RETURN dokončí ve zrychleném režimu podprogram, který procházíte.

Příkaz RETURN zvládne několik úrovní podprogramu, problém ale nastane pokud váš podprogram ukončíte jiným způsobem než RET, např. POP HL, tzn. že při práci se zásobníkem není vhodné použít příkaz CALL ani RETURN.

Tento příkaz lze kdykoliv přerušit SPACE a volit rychlost zobrazení stisknutím 0, 1 nebo 2.

WHILE =w
UNTIL =u Představte si, že máte v programu smyčky DJNZ a JP s podmínkou. Při tom existují dva druhy řízení:

- 1) smyčka pokračuje dokud je podmínka splněna.
- 2) smyčka pokračuje dokud podmínka není splněna.

Příkaz WHILE se volí stisknutím W, testuje a zapamatuje si při které instrukci jste jej stiskl a při návratu na tuto instrukci testuje, zda je podmínka splněna, je-li splněna pak pokračuje (např. ve smyčce) v režimu rychlého zobrazení, ale pokud instrukce podmínku nesplnila, přepne se na režim plného zobrazení. Příkaz je vhodný např. pro smyčky typu DJNZ, kterou lze jednou projít v režimu plného zobrazení a pak třeba 100x nechat proběhnout např. ve FAST režimu.

Příkaz UNTIL, který se volí stisknutím U, je obdobou WHILE pro smyčky, které se opakují pokud není podmínka splněna.

STEP =s (krok). Stisknete S a ve spodní části obrazovky se objeví nápis STEP bez # a vložíte číslo v desítkové soustavě, které udává počet instrukcí, které se mají provést ve zrychleném zobrazovacím režimu po příkazu JUMP.

JUMP =j (skok) Proběhne po stisknutí J a provede ve zrychleném zobrazovacím režimu počet instrukcí zadané příkazem STEP. Počet instrukcí v příkazu můžeme měnit, nebo ponechat stejný, při opakování JUMP.

MOVE =m (pohyb). Stisknutím M se zobrazí ve spodní části obrazovky MOVE #, vložíte adresu v hexadecimální soustavě a po ENTER se nastaví simulační PC (čítač adresy) a umožní vám přenést se na jiné místo vašeho programu.

VIEW =v (pohled). Klávesa V vyvolá zobrazení VIEW #. Vložíte hexadecimální číslo a pohled se vám nastaví na tuto adresu a zobrazí se obsah následujících 8 bytů (pokud se nepřekročí adresa #FFFF).

EXCH =e (výměna). Po stisknutí E se prochází zobrazení registrů za sadu čárkovaných registrů 2 B0 obdobně jako instrukce EXX, ale nemění registry mikroprocesoru. Zobrazené stavové registry také náleží F i když nejsou označeny čárkou.

Druhé stisknutí E opět zobrazí nečárkované registry. Stav se nezapamatovává a po provedení jakéhokoliv jiného příkazu se zobrazí nečárkovaná sada registrů.

QUIT =q (konec). Ukončí běh programu DEBUGGER. Opětovný start můžeme provést příkazem GO TO B700.

MENU =symbol SHIFT + C. Přehled všech symbolů ruší se stisknutím ENTER.

Speciální "SYSTÉMOVÉ PROMĚNNÉ"

ADRESA	POČET BYTŮ	VÝZNAM
64587	26	Prostor odkud se berou a kam se ukládají registry mikroprocesoru během simulace v tomto pořadí: F' A' C' B' E' D' L' H' F A C B E D L H PC SP IX IY R I
64603	8	Prostor pro uložení registrů IY, HL', SP, R a I, které nastaví BASIC a které jsou po provedení jedné instrukce a přesunutí na adresu 64587 opět nastaveny a předány BASICU.
64621	2	Dolní hranice pro PC.
64623	2	Horní hranice pro PC.
64625	2	Dolní hranice pro SP.
64627	2	Horní hranice pro SP.
64629	2	Adresa, kterou DEBUGGER považuje za návrat do BASICU.
64631	2	Hodnota IY.
64633	2	Hodnota HL', kterou má mít registr při návratu do BASICU.
64635	8	Byty, které DEBUGGER nastaví na 0 nebo 1 podle bitů registru F' (méně významné byty LSB - nejdříve).
64643	8	Totéž pro F.
64651	2	Adresa instrukce, která byla právě simulována.

=====

UVOD.

Monitor potřebuje něco přes 4K v horní části paměti. RAMTOP se automaticky nastaví pod MONITOR, přičemž MONITOR leží ihned pod uživatelskou grafikou. Pokud tam máte svoje strojové programy, musíte RAMTOP posunout ještě směrem dolů.

ZAVEDENÍ DO PAMĚTI.

```
LOAD "MONITOR 48 K" nebo LOAD""
LOAD "MONITOR 16 K"
```

SPUŠTĚNÍ MONITORU.

```
RANDOMIZE USR 63247 (pro 48K)
RANDOMIZE USR 30479 (pro 16K)
Z Basicu lze aktivovat MONITOR také po hlášení:
"Press BREAK for MONITOR".
Po tomto hlášení stiskneme BREAK a jsme v MONITORU. Objeví se
kursor, který se zobrazuje jako bílý čtvereček a ukazatel jako .
Tyto značky se objeví, jenom když je ukončena nějaká rutina a
MONITOR čeká na nový příkaz.
Blikající kursor znamená, že čeká na povel z klávesnice, kursor
je na místě, kde se vstup zobrazí.
```

POPIS POVELŮ (přehled v příloze)

- M Zobrazí se blikající M (modré písmeno v bílém čtverci). Zadejte hexadecimálně adresu a objeví se hodnota uložená na této adrese (hexa). Pokud dodáme novou hexa hodnotu (např. FF) uloží se na danou adresu. Když nechceme hodnotu měnit, přejdeme pomocí ENTER na další adresu.
Příklad: M 6000 00 FF
6001 00
- X Návrat k ukazateli a kursoru. Dovoluje stornování příkazu a návrat zpět do MONITORU. Pracuje se všemi příkazy mimo R a K.
- I Forma zadávacího příkazu je I aaaa bbbb nn, přičemž I je vstupní příkaz, aaaa je hexadecimální adresa první adresy od které se má zadávat, bbbb je hexadecimální hodnota posledního bytu, odsunutého bloku paměti a nn je počet bytů. Blok od adresy aaaa do adresy bbbb se tedy posune o nn bytů, takže vznikne možnost vložení (INSERT) nn bytů, přičemž nn může být maximálně 255.
- D Mazání (DELETE) až 255 bytů strojové rutiny.
Příklad: X - zavolání kursoru
D - povel k mazání
6004 - začáteční adresa oblasti k mazání
6014 - konečná adresa oblasti, určené k posunu
05 - počet bytů, které mají být vymazány (od počáteční adresy)
- ENTER
- A Povel pro posunutí bloku atd.
Formát zadání je A bbbb cccc dddd
Příklad: X - zavolání kursoru
A - povel
6000 - startovací adresa bloku
600F - konečná adresa bloku
6200 - nová startovací adresa bloku
ENTER

F Vložení stejných hodnot do určité oblasti paměti.

Formát zadání F aaaa bbbb xxx

Příklad: X - zavolání kursoru

F - povel

6020 - startovací adresa

6100 - konečná adresa

AA - HEX hodnota, která bude uložena na adresy
6020 až 6100.

ENTER

Y Return do BASICU

Po povelu Y-RETURN můžeme používat libovolné Basicové povely. Protože monitor nemá SAVE a LOAD, musíme se před jejich použitím vrátit do Basicu. Pokud se chceme dostat zpět do MONITORU, musíme zadat:

16K : RANDOMIZEUSR 30479 + ENTER

48K : RANDOMIZEUSR 63247 + ENTER

B BREAKPOINT - nastavuje místa přerušení průběhu strojového programu a zpětný návrat k řízení MONITOREMm

Povel má formát B aaaa (aaaa - adresa přerušení musí být adresa prvního bytu u vícebytového příkazu)

Pokud zvolíme aaaa=6009 budou na adresách 6009, 600A a 600B hodnoty CD OF F7 (pro 48k), nebo CD OF 77 (pro 16k) místo původních hodnot. Tento kód znamená příkaz CALL, pomocí kterého se vystoupí z MONITORU.

J Skok do strojové rutiny - volání určité strojové rutiny.

Formát : J aaaa

Příklad: X - zavolání kursoru

J - příkaz

6000 - startovací adresa

ENTER

Rutina bude spuštěna až do bodu přerušení (Breakpoint), kdy se objeví "Press BREAK for MONITOR".

Po povelu J: 1. vymaže se obrazovka
2. ukazatel zásobníku se nastaví na programový zásobník
3. startovací adresa bude vzata do programového čítače a program bude spuštěn.

K BREAKPOINT zrušit

Tento povel opět uloží správně tři adresy (viz B) zpět.

R Zobrazení registru CPU.

Po stisku se automaticky zobrazí obsahy registru CPU.

A" F"

B" C"

D" E"

H" L"

A F

B C

D E

H L

I X

I Y

SP

PC

C BREAKPOINT CONTINUE
 C + ENTER po přerušení bude program opět spuštěn. Obrazovka bude vymazána, programový zásobník bude nastaven a CPU registry budou znovu naplněny daty.

P Výstup na tiskárnu (ZX printer)
 Vypis adres a jejich HEX hodnot. Výpis je možno zastavit pomocí BREAK, návrat do Basicu, znovu zavolat MONITOR.
 Formát: P aaaa bbbb (aaaa-hex adresa prvního bytu k tisku
 bbbb-hex adresa posledního bytu k tisku)

S Zadávání textů
 Pracuje podobně jako M a dovoluje zadávat text přímo z klávesnice (není to však textový procesor).
 Formát: S aaaa (aaaa-startovací adresa textového bloku)
 Vkládání textu po znacích, ale výstup pomocí STOP (Symbol shift + A). Výstupní povel X nelze použít.
 Můžeme vkládat i symboly mimo GRAPHIC a EXTENDED režimu (to musíme vkládat pomocí M a hexakódu).

Z DISASSEMBLER
 Pracuje pro ROM i RAM, výstup na tiskárnu nebo ne. Dodává hexadresy, hexahodnoty bytu a mnemoniku
 Formát: Z aaaa bbbb (aaaa-hexa adresa startu
 bbbb-hexa adresa konce)

Příklad: Z 0000 0020
 PRINTER? odpovíme jako na scroll?
 0000 F3 DI (ENTER = výpis na obrazovku i tiskárnu
 N = výpis pouze na obrazovku)

ENTER = další adresy : X = konec výpisu
 Na obrazovce se zobrazí 16 řádků, pak se očekává ENTER nebo X

N Převod čísel mezi desítkovou a hexadecimální soustavou.
 Po stisku N se objeví: NUMBER H/D? H = z hex do des
 D = z des do hex

Příklad: N
 H 4000
 ENTER
 Objeví se: H 4000 = 16384

MONITOR V PRAKTICKÉM POUŽITÍ.

Účelem tohoto odstavce je vysvětlit veškeré příkazy MONITORu, také ty, o kterých dosud nebylo pojednáno, jakož i podat všeobecné rady k použití programu ve strojovém kódu.

1. Ukazatel MONITORu pracuje se znaky na modrém pozadí. Podle dlouhodobých testů s různými barevnými kombinacemi má tato kombinace nejlepší čitelnost.
2. Reproduktor vydává krátké tony, je-li stisknuta klávesa. Délka tónu tak zajišťuje, aby se snížila reakční doba klávesnice. Systémové proměnné POP nemají vliv na akustickou odezvu klávesnice z MONITORU.
3. Jak už bylo uvedeno, používá MONITOR interní zásobník, kromě případu, kdy program právě probíhá. Programový zásobník uschovává blok dat do SP, kdykoliv provede příkaz J (JUMP) nebo C (CONTINUE). Když se použije příkaz Y (RETURN), programový zásobník se vymaže a podle zjištění v systémových proměnných ERR SP se vrací k normálnímu startovacímu bodu BASICU.

4. Hodnoty registru CPU, které jsou na určitých adresách uschovány, jsou zavedeny nazpět do CPU, když se provede skok (J) nebo BREAKPOINT CONTINUE (C). Návrat BASICU (aby se náš strojový program aktivoval přes USR funkci) přitom uschovává staré hodnoty CPU registru, jak je definováno v BASIC-ROM.
5. V návodu na ZX Spectrum nám na straně 480 doporučuji nepoužívat IX a IY registr v programech ve strojovém kódu. Používejte alternativní páry registru BC, DE a HL, pak se vrátíte do programu ve strojovém kódu v BASICU.
6. Spectrum MONITOR nemá vlastní rutiny "SAVE" a "LOAD", protože je můžeme použít z BASICU. Po napsání programu ve strojovém kódu použijte příkaz pro převod čísel (N), abyste si převedli startovací a konečnou adresu na desítkové číslo. Po návratu do Basicu uložte (SAVE) a přezkoušejte (VERIFY) váš program.

PŘEHLED PŘÍKAZŮ MONITORU

- =====
- M Zobrazí se adresa a její obsah v hexa, možnost změny hodnoty na určité adrese.
 - X Návrat od libovolného příkazu, kromě R a K, zpět do režimu MONITORU, kdy lze zadávat další příkaz MONITORU.
 - A Příkaz k posunu bloku.
 - F Vložení stejných hodnot byte do určité oblasti paměti.
 - I Vložení až 255 byte strojové rutiny.
 - D Vymazání až 255 byte strojové rutiny.
 - J Zavolání určité strojové rutiny.
 - B Zadání místa přerušeni (BREAKPOINT) k přerušeni průběhu strojové rutiny.
 - K Vložení původních hodnot zpět na adresy, které byly přepsány při BREAKPOINTU.
 - R Zobrazí na obrazovce obsahy registru CPU.
 - C Opět spustí rutinu od BREAKPOINTU.
 - Y Návrat do BASICU.
 - P Výpis hexahodnot na ZX printeru.
 - S Možnost přímého vstupu textu.
 - Z Disassembler. Zpětná konverze ze strojového kódu do mnemoniky ASSEMBLERU Z80, možnost zobrazení na obrazovce i tisku na tiskárně.
 - N Konverze hexadecimálních čísel na desítková a naopak
-

Vydal : Kroužek mikroelektroniky ZO Svazarm,
Karolinka
Schváleno : OK ONV Vsetín, š.j. 403/16/87
o 3810 027 87
Náklad : 1 000 výtisků
Poznámka : N E P R O D E J N Ě