

Pavel DOČEKAL

1

ADRESY PAMĚTI

počítačů

ATARI®

600XL/800XL

vydání 1984 • 116 stran • 10 Kč • nakladatelství PSW

ADRESY PAMĚTI POČÍTAČŮ ATARI

Pavel DOČEKAL

Příručka poskytuje důležité informace o uspořádání paměti počítače ATARI bez nichž by efektivní programování prakticky nebylo možné. Celý titul "Adresy paměti počítače ATARI" vychází ve dvou dílech proto, aby se v co nejkratším možném čase dostala tolik očekávaná publikace do rukou amatérů i profesionálních programátorů.

Obsah prvního dílu se zabývá adresovým prostorem RAM od adresy 0 do 40959, druhý díl obsahuje popis ROM. V době vzniku tohoto titulu byly v ČSSR dostupné pouze podklady k počítači ATARI 400. U dalších modelů vývojové řady osmibitových počítačů ATARI 600XL, 800XL, 130XE, jsou zde popsány adresy (až na několik výjimek) zachovány. Podle potřeby jsou doplněny dalšími adresami, o nichž bude zmínka v příloze druhého dílu příručky.

SLOVNÍČEK ZKRATEK

- ANTIC:** Obrazový procesor. Spolu s dalšími obvody (GTIA) pomáhá vytvářet obraz počítače.
- BIT, BYTE:** Bit je nejmenší paměťová jednotka počítače, může mít pouze dva stavy: nulu nebo jednotku. Byte (slabika) má osm bitů, maximální hodnota 255.
- CIO:** Centrální V/V procedury umístěné v ROM.
- DCB:** Přístrojový řídicí blok, použitý SIO.
- DL:** Obrazový podprogram, soubor instrukcí pro ANTIC. Program definuje počátek paměti obrazu, způsob zobrazování dat atd.
- DLI:** Přerušeni, generované instrukcí obrazového podprogramu.
- DOS:** Disketový operační systém.
- FMS:** Správa souboru. Součást DOS. Řídí všechny V/V operace volané "D:".
- FP:** Matematické operace probíhající v plovoucí řádové čárce.
- IOCB:** V/V řídicí blok (paměť 832-959). V řídicím bloku je definován druh a způsob provedení V/V operace.
- IRQ:** Maskovatelné přerušeni např. pro účely sériového přenosu dat, časování, obsluhu klávesnice.
- NMI:** Nemaskovatelné přerušeni, použité při tvorbě obrazu a při RESET.
- OS:** Operační systém. Část paměti ROM, ve které jsou umístěny operace v plovoucí řádové čárce, znakové sady, procedura pro obsluhu periferních zařízení, CIO a SIO.
- PMG:** Grafika hráč/střela. Hráči a střely jsou objekty s možností definice tvaru, barvy, průniku s jinými objekty na obrazovce atd. (též tzv. sprajty).
- RAM:** Paměť, do které lze informace zapisovat i číst. Paměť je schopná informace udržet jen v době přítomnosti napájecího napětí. Paměť slouží k ukládání programů, systémových proměnných atd.
- ROM:** Paměť, ze které lze informace pouze číst. V ROM je uchován celý OS a BASIC.
- SIO:** Sériové V/V procedury.
- VBLANK:** Čas návratu elektronového obrazového paprsku z pravého dolního do levého horního rohu. Tento čas může programátor využít k provedení krátkých strojových programů.
- V/V:** Vstup/výstup.

USPOŘÁDÁNÍ PAMĚTI

0	\$0000	nejnižší adresa RAM, používá operační systém
256	\$0100	zásobníková paměť mikroprocesoru 6502
512	\$0200	vektory přerušení, barvové registry apod.
768	\$0300	parametry diskety, IOCB 0-7, buffer tiskárny
1021	\$03FD	buffer magnetofonu
1152	\$0480	paměť pro volně použití
1280	\$0500	používá operační systém
1536	\$0600	pro volné použití uživatelem
1792	\$0700	FMS (správa souboru DOS) nebo volná RAM
5440	\$1540	DUP.SYS (DOS programy) nebo volná RAM
13062	\$3306	konec DUP.SYS a počátek volné RAM
32768	\$8000	RAM nebo zásuvný modul B
40960	\$A000	RAM nebo zásuvný modul A nebo ROM Basic

PODROBNÝ POPIS

0 **\$00** **LNFLG**

Používá se při testu paměti OS po zapnutí počítače.

1 **\$01** **NOFLAG**

Používá se při testu RAM a ROM po zapnutí počítače. Nula znamená vadnou paměť.

2,3 **\$02,\$03** **CASINI**

Inicializační adresa po přenosu dat z magnetofonu do RAM. Byl-li přenos úspěšný, následuje skok na zde uvedenou adresu. Tato adresa je přečtena z prvních šesti byte magnetofonového záznamu. První byte je ignorován, druhý označuje číslo přenosu, třetí a čtvrtý byte tvoří adresu počátku ukládání programu a konečně pátý a šestý byte tvoří inicializační adresu.

4,5 **\$04,\$05** **RAMLO**

RAM adresa při testu paměti po zapnutí počítače. Využívá rovněž disketa k zápisu přenosové adresy. Normálně je zde 1798 (\$0706).

6 **\$06** **TRAMSZ**

Dočasný registr testu velikosti RAM po zapnutí počítače. Výsledná hodnota je pak předána do registru RAMTOP na adrese 106 (\$6A). Poté, je-li zasunut zásuvný modul A nebo je-li přítomný ROM Basic, je zde zapsána hodnota 1.

7 **\$07** **TSTDAT**

Datový registr testu RAM po zapnutí počítače. Poté, je-li zasunut zásuvný modul B, je zde zapsaná hodnota 1.
Návěst CMCMD pro modemy 835 a 1030.

8 **\$08** **WARMST**

Návěst RESET. Po zapnutí počítače se zde zapíše 0. Po prvním stisku tlačítka RESET se zapíše 255.
Po stisku RESET se provede program začínající od 58484 (\$E474).

9

\$09

BOOT?

Návěst úspěšného přenosu dat.

Je-li zde 0, nebyl dosud uskutečněn žádný přenos.

Je-li zde 1, přenos z diskety byl úspěšný a program přejde na adresu danou v DOSVEC (10,11).

Je-li zde 2, přenos z magnetofonu byl úspěšný a při případném stisku RESET, skočí program na adresu uloženou v CASINI (2,3).

Je-li zde 255, počítač se po stisku tlačítka RESET "zadrhne", odmítne s vámi dále spolupracovat a pomůže jen vypnutí a opětovné zapnutí napájení.

Po zapnutí napájení zkouší počítač oba druhy přenosu a podle výsledku pak nastavuje příslušnou hodnotu.

10,11

\$0A, \$0B

DOSVEC

Startovací adresa disketového programu. Basic příkaz DOS skočí na zde uvedenou adresu. Adresa může být změněna a využita ve vašem vlastním programu.

Stisk tlačítka RESET vrací původní hodnotu 6047 (\$179F).

12,13

\$0C, \$0D

DOSINI

Inicializační adresa po přenosu dat z diskety. DOSINI je rovněž využit pro dočasný zápis startovací adresy po přenosu dat z magnetofonu, která je pak přesunuta do registru CASINI (2,3).

Počáteční nastavení: 5440 (\$1540).

14,15

\$0E, \$0F

APPMHI

Adresa vrcholu využité paměti Basic programem. Od této adresy je možné např. ukládat další Basic program nebo umístit obrazovou paměť.

Po zapnutí počítače se zde zapíše nula.

Nezapomeňte, že obrazovou paměť není možné umístit pod adresu udanou zde!

Viz dále RAMTOP (106), MEMTOP (741,742), PMBASE (54279) a CHBASE (54281).

16

\$10

POKMSK

Návěst maskovatelného přerušení IRQ pro obvod POKEY. Sledující registr má adresu 53774 (\$D20E) a hodnota musí být zapsána současně v obou registrech. Každý bit registru se vztahuje k jednomu možnému přerušení nastavením na 1.

BIT	DEKADICKY	FUNKCE
7	128	klávesa BREAK
6	64	jiné klávesy
5	32	vstupní sériová data připravena
4	16	výstupní sériová data požadována
3	8	výstupní sériový přenos ukončen
2	4	přerušeni od hardware časovače 4
1	2	přerušeni od hardware časovače 2
0	1	přerušeni od hardware časovače 1

Zákaz přerušeni programu klávesou BREAK je zrušen vždy po stisku klávesy RESET nebo příkazem OPEN (S: nebo E:) nebo příkazem PRINT po OPEN nebo příkazem GRAPHICS.

Po přerušeni klávesou BREAK, skočí OS na adresu zapsanou v registru BRKKEY 566,567 (\$236,\$237).

Po přerušeni od hardware časovačů přechází program na adresy zapsané v 528-535 (\$210-\$217). K přerušeni dochází při vynulování čítačů (zvukových registrů AUDF). Bližší informace u obvodu POKEY.

17**\$11****BRKKEY**

Návěst klávesy BREAK. Po stisknutí BREAK se zde zapíše 0. Jakoukoliv V/V operací se zpětně zapisuje 128. K testování BRKKEY dochází při obsluze displeje, magnetofonu i klávesnice.

18,19,20**\$12,\$13,\$14****RTCLOCK**

Vnitřní hodiny. V podstatě se jedná o 24 bitový čítač inkrementovaný během VBI, tedy každých 20ms. Po zapnutí počítače nebo po stisku RESET se RTCLOCK vynuluje. Snadno si spočítáte, že k naplnění celého časovače dojde nejdříve za 96 hodin(!). Zkuste si udělat:

```
10 POSITION 10,10: ? INT((PEEK(18)*65536+PEEK(13)*256+PEEK(20))/50)
20 GOTO 10
```

21,22**\$15,\$16****BUFADR**

Registr nepřímé adresy bufferu. Dočasně ukazuje na disketový buffer.

23**\$17****ICCOMT**

Příkaz pro CIO. Používá se k nalezení offsetu v příkazové tabulce pro korekční vektor obslužné procedury.

40,41 **\$28,\$29** **ICBLLZ/HZ**

Čítač délky bufferu při PUT a GET. Registr se dekrementuje každým přeneseným byte.

42 **\$2A** **ICAX1Z**

První byte pomocné informace používané při OPEN ke specifikaci typu souboru.

43 **\$2B** **ICAX2Z**

Druhý byte pomocné informace pracovní proměnné CIO. Je použit rovněž při sériovém přenosu. CIO proměnná.

44,45 **\$2C,\$2D** **ICAX3Z/4Z**

Použit při Basic příkazech NOTE a POINT pro přepis čísel disketových sektorů.

46 **\$2E** **ICAX5Z**

Určuje byte v sektoru ICAX3Z/4Z. Používá rovněž IOCB.

47 **\$2F** **ICAX6Z**

Záložní byte. Používá se pro dočasné uložení znaku právě probíhající operace PUT.

48 **\$30** **STATUS**

Registr vnitřního statusu. SIO procedury v ROM používají tento registr k zápisu statusu právě probíhající SIO operace. STATUS používá registr TSTAT (793; \$319) pro dočasnou úschovu.

49 **\$31** **CHKSUM**

Zkušební součet používaný při SIO operacích. Je to v podstatě aritmetický součet všech vyslaných čísel. Pokud hodnota souhlasí, nastaví se návěst CHKSNT na adrese 59 (\$3B). Při zkušebním součtu se využívá pro komparaci čísel BUFRFL na adrese 56 (\$38) a dále ještě 53773 (\$D20D).

50,51 **\$32,\$33** **BUFRLO/HI**

Adresa ukazující na byte v datovém bufferu, který má být právě vysílán nebo přijímán během V/V operace.

63 **\$3F** **FEOF**

Návěst konce souboru magnetofonového zápisu. Je-li zde nula, nebude konce souboru dosaženo. Aby bylo dosaženo konce souboru, musí mít příkazový byte magnetofonového zápisu hodnotu 254 (\$FE).

64 **\$40** **FREQ**

Počet pípnutí při práci s magnetofonem. Hodnota se zde OS zadá při otevření V/V kanálu. Jedno pípnutí pro čtení, dvě pro záznam.

65 **\$41** **SOUNDR**

Návěst hlasitosti při přenosu dat. Běžně je zde hodnota 3, nula zvukový doprovod při přenosu dat umlčí. Viz ještě adresa 54018 (\$D382).

66 **\$42** **CRITIC**

Návěst kritických V/V operací. Nula zde bude znamenat vysazení řady procesů vykonávajících se během VBI. Nebudou se např. dekrementovat časovače 2, 3, 4 a 5, nebude probíhat přepis hodnot do sledujících registrů, attract mód bude vyřazen z činnosti, nebude možný opakovaný stisk stejné klávesy atd.

67-73 **\$43-\$49** **FMZSPG**

Registry disketového souboru (FMS) v nulové stránce paměti. 7 byte.

74,75 **\$4A,\$4B** **ZCHAIN**

Registry dočasné úschovy informací při obslužných procedurách.

76 **\$4C** **DSTAT**

Statusový registr displeje a klávesnice používaný obsluhou displeje. Je rovněž použit k indikaci nedostatku paměti pro daný obrazový mód nebo k indikaci polohy kurzoru mimo rozsah nebo jako BREAK status.

77**\$4D****ATTRACT**

Časovač attract modu a návěst. Nemění-li se obraz příliš dlouho, mohla by se luminoforová vrstva obrazovky poškodit. Proto konstruktéři ATARI implementovali do systému attract mód, který se aktivuje asi po osmi minutách během nichž nedošlo ke stisku klávesy. Attract poté zabezpečí rotaci obrazových barev a snížení jasu až do prvního stisku klávesy, kdy se ATTRACT vynuluje. Hodnota registru ATTRACT je inkrementována přibližně každé čtyři sekundy během VBLANC až do hodnoty 127 (\$7F). Po přetečení hodnoty do sedmého bitu se v registru nastaví 254 (\$FE). Páka nemá na registr vliv. Pokud rotace barev není žádoucí, stačí periodicky opakovat POKE 77,0.

78**\$4E****DRKMSK**

Návěst jasu při attract modu. Normální je zde 254 (\$FE). Při aktivním attract modu se nastaví 246 (\$F6), což způsobí, že jas obrazovky nepřekročí 50%.

79**\$4F****COLRSH**

Maska rotace barev. Jestliže registr COLRSH nabude hodnotu nula a registr DRKMSK (viz výše) obsahuje 246 (\$F6), jas obrazovky je redukován na 50% a zároveň se vymění barva v barvových registrech. Hodnota je zde převzata z časovače RTCLK (19; \$13), z toho vyplývá rychlost střídání barev v barvových registrech v intervalu 5.12 sekund.

Adresy 80 až 122 (\$50 až \$72) jsou využita obrazovým editorem a budičem displeje.

80**\$50****TEMP**

Dočasný registr používaný obsluhou displeje pro přesun dat do a z obrazové paměti.

81**\$51****HOLD1**

Stejně jako TEMP (viz výše). Je také používán v souvislosti s DL.

82**\$52****LMARGN**

Doraz kurzoru zleva. Inicializuje se na 2.

83**\$53****RMARGN**

Doraz kurzoru zprava. Inicializuje se na 39. Hodnoty v LMARGN a v RMARGN jsou akceptovány pouze grafickým modem 0 a textovým okénkem a s jistým omezením mohou být nastaveny na libovolnou hodnotu (0-39 a LMARGN<RMARGN).

Stisk tlačítka RESET obnovuje v obou registrech původní hodnoty.

84

\$54

ROWCRS

Okamžitá řádková poloha grafického nebo textového kurzoru. Podle zvoleného grafického modu se zde může nacházet hodnota v rozsahu 0-191. Nejnižší hodnota 0 znamená polohu kurzoru v nejvyšším řádku. Spolu s registrem COLCRS (viz níže) umožňuje zjištění nebo zadání nové polohy kurzoru.

85, 86

\$55, \$56

COLCRS

Okamžitá sloupcová poloha grafického nebo textového kurzoru. Podle zvoleného grafického modu se zde může nacházet hodnota v rozsahu 0-319. Nejnižší hodnota 0 znamená polohu kurzoru nejbližše levému okraji obrazovky.

Pro zjištění polohy kurzoru v textovém okénku viz registry TXTROW (656, \$290) a TXTCOL (657,658; \$291,\$292).

Registry COLCRS a ROWCRS jsou využívány funkcemi DRAW a FILL. Hodnoty ve všech registrech 80-86 (\$50-\$56) lze nejen číst, ale i modifikovat. Basic příkaz LOCATE nejen přebírá hodnotu znaku z obrazovky z udaných souřadnic, ale přesunuje také kurzor o jednu pozici doprava. Zpětný návrat kurzoru zařídí krátký ukázkový program:

```
100 REM Obrazovka musí být před pokračováním programu otevřená pro  
čtení nebo pro čtení/zápis.
```

```
120 LOOK=PEEK(84):SEE=PEEK(85)
```

```
130 LOCATE X,Y,THIS
```

```
140 POKE 84,LOOK:POKE 85,SEE
```

87

\$57

DINDEX

V dolních čtyřech bitech se nachází hodnota aktivovaného grafického modu, která je sem přenesena z AUX1. OS při některých operacích DINDEX testuje a tak se změnou hodnoty registru dá změnit vlastně i grafický mód vyvolaný příkazem GRAPHICS. Jak se dají za pomoci DINDEX simulovat nové grafické módy ukáže následující program.

```
10 GRAPHICS 8+16
```

```
20 POKE 87,7
```

```
30 COLOR 3
```

```
40 PLOT 0,0:DRAWTO 159,0
```

```
50 COLOR 2
```

... pokračování na další stránce

```
60 FOR Y=0 TO 95:PLOT 0,Y:PLOT 159,Y:NEXT Y
70 COLOR 3
80 PLOT 0,0:DRAWTO 159,95
90 COLOR 2
100 PLOT 0,95:DRAWTO 159,0
110 COLOR 1
120 PLOT 0,95:DRAWTO 159,95
200 POKE 89,PEEK(89)+15
210 COLOR 3
220 PLOT X,0:DRAWTO 159,0
230 COLOR 2
240 FOR Y=0 TO 95:PLOT 0,Y:PLOT 159,Y:NEXT Y
250 COLOR 1
260 PLOT 0,0:DRAWTO 79,95:DRAWTO 159,0
270 PLOT 0,95:DRAWTO 79,0:DRAWTO 159,95
280 COLOR 2
290 PLOT 0,95:DRAWTO 159,95
300 GOTO 300
```

10: Zvolen GR. 8+16

20: Do registru DINDEX je však vložena hodnota 7.

30-120: Kreslení různých čar v různých barvách.

Všimněte si, že DL zpracovává obraz jako v GR. 8+16, ale příkazy PLOT a DRAWTO lze použít jen v rozsahu hodnot platných pro GR. 7+16. To by ovšem znamenalo možnost kreslit jen na horní polovinu obrazovky. Dá se to nějak ošidit? Samozřejmě. Stačí pouze posunout počátek obrazové paměti v SAVMSC (viz níže). Všechny souřadnice bodu se totiž vztahují k danému grafickému režimu a adrese počátku obrazové paměti.

200: Zvýšení adresy v SAVMSC o 3840.

210-290: Nyní se může popisovat i dolní polovina obrazu.

300: GOTO zabraňuje ukončení programu a tím i zrušení obrazu.

88, 89

\$58, \$59

SAVMSC

Adresa počátku obrazové paměti (levý horní roh obrazovky). V běžných grafických režimech leží konec obrazové paměti před počátkem ROM (40959; \$BFFF).

Následující program využívá možnost změny adresy počátku obrazové paměti a tím i rychlé změny obrazu.

```
10 D=1
20 GRAPHICS 3+16
30 POKE 708,50
40 POKE 709,52
50 POKE 710,54
60 FOR P=0 TO 20
70 COLOR C+1
80 X=INT(RND(0)*12)
90 Y=INT(RND(0)*12)
100 PLOT 19-X,11-Y
```

```
110 PLOT 19+X,11-Y
120 PLOT 19+X,11+Y
130 PLOT 19-X,11+Y
140 PLOT 19-Y,11-X
150 PLOT 19+Y,11-X
160 PLOT 19+Y,11+X
170 PLOT 19-Y,11+X
180 NEXT P
200 D=D+1:C=C+1:IF C=3 THEN C=0
210 IF D=7 THEN 300
220 SM=PEEK(88)+PEEK(89)*256
230 FOR K=0 TO 239
240 POKE SM-D*256+K,PEEK(SM+K)
250 NEXT K
260 GOTO 20
300 DL=PEEK(560)+PEEK(561)*256
310 Z=PEEK(DL+5)
320 FOR J=2 TO 6
330 POKE DL+5,Z-J
340 FOR W=0 TO 50:NEXT W
350 NEXT J
360 FOR J=5 TO 2 STEP -1
370 POKE DL+5,Z-J
380 FOR W=0 TO 50:NEXT W
390 NEXT J
400 GOTO 320
```

30-50: Definování barvových registrů.

60-180: Program generuje několik náhodných obrazců.

220-250: Náhodně vygenerované obrazce jsou postupně zapisovány do různých stránek paměti.

300: Počátek DL (display list).

310: Vyšší byte v adrese instrukce LMS (viz ABC o počítačích ATARI, str. 7).

320-390: Postupně se zobrazuje pět stránek paměti, do kterých byly předtím uloženy náhodně vygenerované obrazce.

Možnosti animace apod.

Adresa počátku obrazové paměti textového okénka je zapsána v registru TXTMSC (660,661; \$294,\$295).

90

\$5A

OLDROW

Předešlý řádek grafického kurzoru. Je obnovován před každou operací hodnotou z registru ROWCRS (84; \$54). Takto je pro příkazy DRAWTO nebo FILL k dispozici počáteční řádková souřadnice.

91,92

\$5B,\$5C

OLDCOL

Předešlý sloupec grafického kurzoru. Je obnovován před každou operací

hodnotou z registru COLCRS (85,86; \$55,\$56). Takto je pro příkazy DRAWTO nebo FILL k dispozici počáteční sloupcová souřadnice.

93**\$5D****OLDCHR**

Obsahuje kód znaku právě překrytého kurzorem. Využívá se k obnovení znaku při pohybu kurzoru.

94, 95**\$5E, \$5F****OLDADR**

Adresa paměťového místa, ve kterém se právě vyskytuje kurzor. Spolu s OLDCHR (viz výše) se registr OLDADR využívá k obnovení znaku při pohybu kurzoru.

96, 97**\$60, \$61****FKDEF**

ATARI 1200XL má k dispozici čtyři předefinovatelné funkční klávesy. V FKDEF se nachází adresa definiční, 8 byte dlouhé, tabulky kláves F1-F4 a SHIFT F1 - SHIFT F4. Každý byte je shodný s interním (ne s ASCII) kódem. Kódy kláves jsou 138-141 (\$8A-\$8D). Počáteční inicializace registru je 64529 (\$FC11). Funkční klávesy mají tuto počáteční funkci:

KLÁVES A**FUNKCE**

F1	kurzor nahoru (ATASCII 28; \$1C)
F2	kurzor dolů (29; \$1D)
F3	kurzor doleva (30; \$1E)
F4	kurzor doprava (31; \$1F)

KLÁVES A+SHIFT**FUNKCE**

F1	kurzor do levého horního rohu (28; \$1C)
F2	kurzor do levého dolního rohu (29; \$1D)
F3	kurzor na počátek fyzického řádku (30; \$1E)
F4	kurzor na konec fyzického řádku (31; \$1F)

KLÁVES A+CTRL**FUNKCE**

F1	klávesnice zapojena / vyřazena
F2	výpis obrazu povolen / zakázán
F3	zvukový doprovod klávesnice povolen / zakázán
F4	mezinárodní znaky

Funkční klávesy jsou ignorovány při současném stisku SHIFT a CTRL. Není možné redefinovat CTRL-funkce.

98**\$62****PALNTS**

Návěst televizní normy. V předešlých verzích 400/800 byla tato informace uložena na adrese 53268 (\$D014). Nula značí NTSC (americká televizní norma). Jedna značí PAL (evropská televizní norma).

99 **\$63** **LOGCOL**

Pozice kurzoru uvnitř logického řádku. Logický řádek může být dlouhý až tři fyzické řádky GRAPHICS 0, tedy $3 \times 40 = 120$. Registr nabývá hodnot 0-119. Používá obsluha displeje.

100,101 **\$64,\$65** **ADRESS**

Používá obsluha displeje k přechodnému uložení různých hodnot pro DL, řádkový buffer (583 až 622; \$247 až \$26E), hodnotu DMASK, novou hodnotu pro MEMTOP po změně DL, rolování, vypouštění znaku, mazání obrazovky a pro adresu obrazové paměti (88,89; \$58,\$59).

102,103 **\$66,\$67** **MLTTMP**

První byte je použit příkazem OPEN jako dočasný registr, je rovněž používán obsluhou displeje jako dočasný registr.

104,105 **\$68,\$69** **SAVADR**

Dočasný registr, používaný společně s ADRESS (viz výše) při přesunu řádkových dat na obrazovku.

106 **\$6A** **RAMTOP**

Počet stránek paměti RAM volně přístupných uživateli (jedna stránka=256 byte). Po zapnutí počítače se zde přesune hodnota z TRAMSZ (6). U ATARI 800XL je to 160 (\$A0).

Tohoto registru může být využito k vytvoření chráněné oblasti paměti, kde se pak může ukládat strojový program, oblast PM-grafiky, nově definované sady znaků - bez obav, že bychom o ně přišli.

Následující programový řádek zajistí potřebný počet chráněných "n" stránek:

```
POKE 106,PEEK(106)-n
```

Hodnota v registru RAMTOP nesmí být menší než hodnota v registru MEMTOP (144,145). Je žádoucí, aby po změně hodnoty v registru RAMTOP následoval příkaz GRAPHICS nebo CLEAR. Na nové adresy se přesunou DL, obrazová paměť a vynuluje se 64 byte nad registrem RAMTOP. Nezapomeňte, že v grafickém modu s textovým okénkem se při rolování okénka roluje zároveň 800 byte nad RAMTOP (20 řádků), neboť rolovací procedura obsahuje pouze rolování celé obrazovky. Proto v těchto modech myslíte na větší rezervu chráněné oblasti.

V grafických režimech 7 a 8 je žádoucí přesunovat RAMTOP o 4k byte (16 stránek), jinak dochází k nežádoucím jevům.

Jiná možnost zřízení chráněné oblasti paměti je pod MEMLO (743,744; \$2E7,\$2E8).

Viz ještě také MEMTOP (741,742; \$2E5,\$2E6).

107 **\$6B** **BUFCNT**

Čítač bufferu. Běžná velikost logického řádku obrazového editoru.

108,109 **\$6C,\$6D** **BUFSTR**

Ukazatel procedury GETCH obrazového editoru (62867; \$F593). Vrací znak, na který ukazuje BUFCNT.

110 **\$6E** **BITMSK**

Bitová maska procedury obsluhy displeje (64235 až 64305; \$FAEB až \$FB31). Slouží také jako dočasný registr obsluhy displeje.

111 **\$6F** **SHFAMT**

Nastavení obrazových elementů (Pixel). Tato hodnota je stejná jako v registru DMASK (672; \$2A0).

112,113 **\$70,\$71** **ROWAC**

ROWAC a COLAC (viz níže) společně slouží jako pracovní akumulátory pro řízení zápisu bodu a k funkční inkrementaci nebo dekrementaci.

114,115 **\$72,\$73** **COLAC**

Viz ROWAC výše.

116,117 **\$74,\$75** **ENDPT**

Koncové souřadnice kreslené čáry. Zapiše se zde hodnota DELTAR nebo z DELTAC podle toho, která je vyšší. Registr je využit ve spojení s ROWAC/COLAC k řízení kreslení čary.

118 **\$76** **DELTAR**

Řádková diference. Obsahuje absolutní rozdíl mezi NEWROW a ROWCRS.

119,120 **\$77,\$78** **DELTAC**

Sloupcová diference. Obsahuje absolutní rozdíl mezi NEWCOL a COLCRS. DELTAR a DELTAC vlastně definují sklon kreslené čáry.

121,122**\$79,\$7A****KEYDEF**

Adresa definiční tabulky klávesnice inicializována na 64337 (\$FB51).
Tabulka sestává ze tří 64 byte bloku. První blok definuje běžné znaky:

BYTE	KLÁVES A	BYTE	KLÁVES A
00	l	32	,
01	j	33	mezera
02	;	34	.
03	F1 (1200XL)	35	n
04	F2 (1200XL)	36	(128)
05	k	37	m
06	+	38	/
07	*	39	inverze (114)
08	o	40	r
09	(128 viz dále)	41	(128)
10	p	42	e
11	u	43	y
12	RETURN	44	TAB
13	i	45	t
14	-	46	w
15	=	47	q
16	v	48	9
17	HELP (128)	49	(128)
18	c	50	0
19	F3 (1200XL)	51	7
20	F4 (1200XL)	52	BACKSPACE
21	b	53	8
22	x	54	<
23	z	55	>
24	4	56	f
25	(128)	57	h
26	3	58	d
27	6	59	(128)
28	ESC	60	CAPS (130)
29	5	61	g
30	2	62	s
31	1	63	a

Další 64 byte blok je pro SHIFT+klávesa a poslední pro CTRL+klávesa.
Můžete si v RAM definovat vlastní tabulku. Pak zde změňte adresu.
Několik hodnot pro klávesový dekodér má specifické použití:

DEC/HEX	POUŽITÍ
128/88	nepoužito, chybná kombinace
129/81	inverzní výstup
130/82	velké/malé znaky
131/83	CAPS funkce
132/84	CTRL funkce
133/85	konec zápisu

137/89	píp klávesy
138-141/8A-8D	funkční klávesy F1-F4 (pouze pro 1200XL)
	kurzor nahoru
	kurzor dolů
	kurzor doleva
	kurzor doprava
142/8E	kurzor do levého horního rohu
143/8F	kurzor do levého dolního rohu
144/90	kurzor k levému okraji (1200XL)
145/91	kurzor k pravému okraji (1200XL)

Klávesy BREAK, SHIFT a CTRL není možné předefinovat. Po stisku RESET se do KEYDEF vrací původní adresa.

123 **\$7B** **SWPFLG**

Řídící návěst kurzoru při dělení obrazovém modu. Jestliže je RAM textového okénka a regulérní RAM přehozena, je zde 255. Jinak je vždy nula.

124 **\$7C** **HOLDCH**

Dočasně se zde ukládá hodnota znaku před rozhodnutím logiky mezi SHIFT a CTRL.

125 **\$7D** **INSDAT**

Dočasný registr používaný obsluhou displeje pro znak pod kurzorem a pro rozeznávání konce řádku.

126,127 **\$7E,\$7F** **COUNTR**

Začíná vyšší hodnotou buď z DELTAR (118; \$76) nebo z DELTAC (119; \$77). Hodnota odpovídá počtu bodů, které mají být nakresleny příkazem DRAWTO. Hodnota se postupně během příkazu snižuje o jednu až do konečné hodnoty nula.

128,129 **\$80,\$81** **LOMEM**

Nachází se zde adresa počátku Basic programu. Prvních 256 byte slouží jako výstupní buffer, který je použit Basicem k překladu všech zadávaných výroků do numerické podoby.

Při SAVE se zapisují dva bloky informací:

Prvním blokem je sedm Basic adres LOMEM až STARP (128-141; \$80-\$8D), přičemž tyto hodnoty jsou nahrávány tak, že LOMEM je roven nule a od dalších adres se původní LOMEM hodnota odečítá. Při LOAD je postup obrácený, přičemž LOMEM získává hodnotu, která je v MEMLO (743,744; \$2E7,\$2E8).

Druhý blok obsahuje tabulku názvů proměnných, tabulku hodnot proměnných, přeložený Basic program a řádek v přímém módu. Není-li v paměti nahrán DOS nebo jiní aplikační program, je v LOMEM adresa 1792 (\$700). Je-li přítomen DOS, LOMEM obsahuje adresu 7420 (\$1CFC).

130,131

\$82,\$83

VNTP

Počáteční adresa tabulky názvů proměnných. Názvy se do tabulky ukládají v ASCII kódu v pořadí, v jakém byly postupně zadávány, přičemž může být definováno maximálně 128 proměnných.

Názvy proměnných mohou být teoreticky tak dlouhé, jako logický řádek a interpreter je bezpečně schopen tak dlouhé názvy rozlišit. Aby se poznal konec názvu proměnné, je sedmý bit posledního znaku názvu nastaven na jedna.

Skaláry končí posledním znakem názvu (s nastaveným sedmým bitem), řetězcové proměnné používají poslední znak "\$" a konečně názvy polí končí "(".

Někdy se při tvorbě obzvlášť dlouhého programu stane, že na další názvy proměnných už nezbyvá místo. K tomu pár cenných rad:

1. Volte všechny názvy proměnných co nejkratší. Dlouhé názvy zbytečně obsazují paměť a zpomalují program. Je ovšem pravda, že delší názvy jsou výstižnější a snadněji zapamatovatelné. Pokud to bude nutné, pište delší názvy a ty potom v případě nutnosti kratte, ale přímo! v tabulce názvů proměnných. Ušetříte si tím spoustu zbytečného hledání zapomenutých nezkrácených názvů.
2. Není-li již v tabulce názvů místo, pomůže někdy malá zacházka. Je velká pravděpodobnost, že při programování došlo k definování nějaké proměnné, která není z jakýchkoliv důvodů vůbec využita. V tabulce přesto existuje a zabírá místo. Pomůže uložení programu pomocí LIST, pak NEW a zpětné zavedení programu přes ENTER. Všechny nevyužité názvy se zpět do programu již nedostanou.
3. Používejte často pomocných proměnných. Neexistuje důvod, proč byste nemohli použít v sobě nevložených smyčkách FOR-NEXT stejný název proměnné.

Můžete si zkusit krátkým programem vylistovat všechny názvy v tabulce:

```
1 POKE 1664,PEEK(130):POKE 1665,PEEK(131)
2 IF PEEK(1664) = PEEK(132) THEN IF PEEK(1665) = PEEK(133) THEN STOP
3 PRINT CHR$(PEEK(PEEK(1664)+PEEK(1665)*256));
4 IF PEEK(1664)+PEEK(1665)*256 > 127 THEN PRINT " ";
5 IF PEEK(1664) = 255 THEN POKE 1664,0:POKE 1665,PEEK(1665)+1:GOTO 2
6 POKE 1664,PEEK(1664)+1:GOTO 2
```

132,133**\$84,\$85****VNTD**

Adresa konce tabulky názvů proměnných + 1 byte. Je-li v tabulce všech 128 proměnných, je ve VNTD adresa prvního byte za posledním znakem poslední proměnné. Je-li v tabulce méně proměnných, ukazuje VNTD na byte ležící hned za poslední proměnnou.

134,135**\$86,\$87****VVTP**

Počáteční adresa tabulky hodnot proměnných. Každá proměnná obsadí v tabulce 8 byte.

Byte	1	2	3	4	5	6	7	8
Proměnná								
skalár	00	var#	6 byte BCD konstanty					
pole (dimenz.)	65	var#	offset	první	druhý			
pole (nedimenz.)	84		od	DIM+1	DIM+1			
			STARPU					
řetězec (dimenz.)	129	var#	offset	délka	DIM			
řetězec (nedimenz.)	128		od					
			STARPU					

Skalární proměnná obsahuje ve třetím až osmém byte desítkovou hodnotu, vyjádřenou v pohyblivé řádové čárce. Třetí byte je exponent a čtvrtý až osmý byte jsou desítková čísla (osmý byte je MSB).

Pátý a šestý byte pole obsahuje první dimenzovaný rozměr+1 (LSB/MSB) a sedmý a osmý byte pak druhý dimenzovaný rozměr+1 (LSB/MSB).

U řetězců obsahuje pátý a šestý byte běžnou délku proměnné (LSB/MSB), a sedmý a osmý byte aktuální dimenzovaný rozměr (max. 32767).

Ve všech případech označuje první byte druh proměnné. Druhý byte bývá v rozsahu 0-127.

Offset je číslo udávající odstup proměnné od registru STARPU na adrese 140,141 (\$8C,\$8D).

Kteroukoliv proměnnou ve VVTP lze vypsát následovně:

```
1000 VVTP=PEEK(134)+PEEK(135)*256
1005 INPUT VAR: REM pořadí proměnné
1010 FOR LOOP=0 TO 7
1815 PRINT VVTP+LOOP+8*VAR
1820 NEXT LOOP
```

136,137**\$88,\$89****STMTAB**

Tabulka výroků. Od adresy udané v STMTAB jsou uloženy všechny přeložené Basic řádky včetně řádku zadaného v přímém módu. První dva byte tabulky nesou číslo prvního Basic řádku, přičemž řádek v přímém módu má číslo 32768 (\$8000). Následující byte udává velikost celého přeloženého Basic řádku.

Podrobnější informace naleznete v ABC o počítačích na straně 45-59.

138,139

\$8A,\$8B

STMCUR

Adresa právě vykonávaného výroku, nacházejícího se v tabulce výroků. Čeká-li Basic na vklad, je v STMCUR uložena adresa řádku přímého vkladu (32768).

140,141

\$8C,\$8D

STARP

Adresa počátku tabulky polí a řetězců. Adresa v registru STARP zároveň udává konec přeloženého Basic programu.

Číselné hodnoty jsou ukládány v 6byte BCD formě, řetězcové znaky zabírají jeden byte každý. Z toho jasně vyplývá, že po DIM A(100) se v paměti rezervuje 600 byte a po DIM A\$(100) se rezervuje 100 byte. Počáteční adresu řetězců a polí lze získat pomocí funkce ADR.

142,143

\$8E,\$8F

RUNSTK

Počáteční adresa zásobníkové paměti pro GOSUB (zabere vždy 4 byte) a FOR-NEXT (zabere vždy 16 byte).

První byte GOSUB je vždy 0, následuje celočíselný 2 byte integer (LSB/MSB) řádku, ve kterém byl GOSUB volán a poslední byte udává offset GOSUB od počátku Basic řádku, aby se po RETURN mohl interpreter vrátit do správného místa v Basic řádku.

Při FOR-NEXT je prvních 6 byte číselná hodnota BCD udaná za TO. Dalších 6 byte je rovněž BCD číslo a vyjadřuje STEP. Třináctý byte je kód pro proměnnou čítače s nastaveným sedmým bitem na 1. Čtrnáctý a patnáctý byte nese číslo řádku, ve kterém je umístěn FOR a konečně šestnáctý byte vyjadřuje offset FOR od počátku řádku. Adresa v RUNSTK současně znamená konec tabulky proměnných.

144,145

\$90,\$91

MEMTOP

Adresa vrcholu Basic programu. Prostor nad MEMTOP až po počátek DL lze vyjádřit pomocí FRE(0).

Buňky paměti 146-202 (\$92-\$C2) jsou využity 8K Basic ROM.

Buňky paměti 176-207 (\$B0-\$CF) jsou využity zásuvným modulem Assembler Editor.

182**\$B6****DATAD**

Počet čtení v Basic DATA řádku od příkazu RESTORE. Vždy po RESTORE se v DATAD nastaví nula a po každém čtení příkazem READ v jednotlivém řádku se hodnota v registru zvýší o jednotku.

183,184**\$B7,\$B8****DATALN**

Číslo Basic řádku (vyjádřené jako 2 byte integer), který byl nastaven příkazem RESTORE nebo číslo naposledy čteného Basic DATA řádku. Funkci registru DATAD a DATALN blíže objasní demonstrační program:

```
10 REM Demonstrační program DATA-RESTORE-READ
20 DIM C$(2),A$(20)
30 C$=CHR$(125)
100 DATA JEDNA-1,DVE-2,TRI-3,CTYRI-4,0
110 DATA PET-1,SEST-2,SEDM-3,OSM-4,0
120 DATA <9-1>,<10-2>,<11-3>,<12-4>,1
150 PRINT C$: RESTORE 100
160 READ A$: IF A$="" THEN 200
170 IF PEEK(182)=1 THEN PRINT: PRINT "Na řádku ";
PEEK(183)+PEEK(184)*256; " je zapsáno:"
180 IF A$="1" THEN 300
190 PRINT "#"; PEEK(182); " "; A$; " ": GOTO 160
200 PRINT: GOTO 160
300 PRINT: PRINT
310 TRAP 400: PRINT "Který datový řádek (1, 2, 3) ": INPUT DATALINE
320 PRINT "Která pozice v řádku (1, 2, 3, 4) ": INPUT ITEM
330 LET DATALINE=90+10*DATALINE
340 POKE 184,INT(DATALINE/256): POKE 183,DATALINE-INT(DATALINE/256)
350 POKE 182,ITEM-1
368 READ A$: PRINT "Data na udané pozici: "; A$
370 GOTO 300
400 END
```

186,187**\$BA,\$BB****STOPLN**

Ukládá se zde číslo řádku, ve kterém byl program zastaven klávesou BREAK nebo příkazem STOP nebo při ERROR. Registry STOPLN a ERRSAVE (viz níže) spolu s příkazem TRAP a dalším nezbytným programovým vybavením mohou zamezit zastavení programu s chybovým hlášením ERROR při jakékoliv chybě, která by mohla nastat vlivem špatného vkladu, přenosu dat apod.

192**\$CO****IOCMD**

V/V příkaz.

193 **\$C1** **IODVC**
V/V zařízení.

194 **\$C2** **PROMPT**
Okamžitý znak.

195 **\$C3** **ERRSAVS**
Obsahuje kód chybového hlášení, které vedlo nebo mohlo vést (pokud bylo použito TRAP) k zastavení programu.

200 **\$C8** **COLOR**
Je zde zapsána barva definovaná COLOR a používaná výroky PLOT a DRAWTO. Předefinování barvy lze provést jak COLOR x, tak i POKE 200,x. Stejná hodnota je i v registru ATACHR (763; \$2FB).

201 **\$C9** **PTABW**
Určuje počet mezer mezi příkazem PRINT s čárkou, a následným tiskem. Například PRINT A\$,LOOP,C(12). Není zde žádná souvislost s klávesou TAB. Nejvyšší možná hodnota může být až 255, tj. odstup 257 sloupců. Zapiše-li se do PTABW nula, systém se po první čárce za PRINT zavěsí a může být vrácen jen tlačítkem RESET. Minimální hodnota zapsaná do PTABW je 3. Inicializuje se na 10. RESET ponechává v registru původní hodnotu.
Zbývá ještě dodat, že PTABW pracuje ve všech grafických módech.

203-207 **\$CB-\$CF** -----
Paměť využívá pouze zásuvný modul Assembler.

203-209 **\$CB-\$D1** -----
Jediné buňky nulové stránky paměti, které Basic ROM ani OS nevyužívá.

210-211 **\$D2-\$D3** -----
Rezervováno pro Basic, pracovní oblast operací v pohyblivé řádové čárce.

Adresy 214-255 (\$D6-\$FF) jsou rezervovány pro matematické operace v pohyblivé řádové čárce (FP-floating point) dále jen FP. Matematické programy jsou uloženy v ROM od 55296 do 57393 (\$D800-\$E031). Tyto buňky nulové stránky paměti lze využívat, ovšem pod podmínkou, že současně nebudou volány matematické programy. Každé číslo vyjádřené v FP potřebuje v tomto počítači 6 byte. První byte je exponent přičemž nejvyšší sedmý bit nese informaci o znaménku exponentu (0=kladný, 1=záporný). Dalších pět byte je mantisa. Vše v BCD kódu.

212-217 \$D4-\$D9 FR0

FP registr nula. Slouží pro převod 2 byte celočíselné hodnoty (integer) na desítkové FP vyjádření a naopak, přičemž integer se umísťuje před převodem na adresy 212,213. Pro převod integer/FP slouží podprogram začínající od adresy 55722 (\$D9AA) a pro převod FP/integer podprogram od 55762 (\$D9D2).

Registr je rovněž využíván Basic funkcíUSR.

218-223 \$DA-\$DF FRE

FP extra registr (?)

224-229 \$E0-\$E5 FR1

FP registr jedna. Obsahuje šest byte interní formy FP čísla stejně jako FR0. FP programy velmi často vyměňují mezi oběma registry data a používají jich při aritmetických operacích se dvěma vstupními hodnotami.

230-235 \$E6-\$EB FR2

FP registr dva.

236 \$EC FRX

Záložní FP registr.

237 \$ED EEXP

Hodnota exponentu E.

238 \$EE NSIG

Návěst znaménka FP čísla mantisy.

239 **\$EF** **ESIGN**

Návěst znaménka FP čísla exponentu.

240 **\$F0** **FCHRFLG**

První znaková návěst.

241 **\$F1** **DIGRT**

Počet čísel za desetinnou čárkou.

242 **\$F2** **CIX**

Znakový index. Používán jako offset ve vstupním textovém bufferu INBUFF (viz níže).

243, 244 **\$F3, \$F4** **INBUFF**

Adresa vstupního textového bufferu (ASCII). Použit při změně ASCII kódu na FP číslo. Výstupní buffer se nachází od adresy 1488 do 1535 (od \$588 do \$5FF).

245, 246 **\$F5, \$F6** **ZTEMP1**

Dočasný registr.

247, 248 **\$F7, \$F8** **ZTEMP2**

Dočasný registr.

249, 250 **\$F9, \$FA** **ZTEMP3**

Dočasný registr.

251 **\$FB** **RADFLG**

Je-li RADFLG roven nule, probíhají všechny trigonometrické funkce v radiánech. Je-li zde zapsána hodnota šest pomocí POKE nebo DEG, počítá se v gradech. Přepnutí zpět do základního stavu lze provést RAD, NEW nebo RESET.

252,253**\$FC,\$FD****FLPTR**

Ukazuje na první FP číslo.

254,255**\$FE,\$FF****FPTR2**

Ukazuje na druhé FP číslo, které bude použito v následující FP operaci.

Konec nulové stránky RAM.

PRVNÍ STRÁNKA: ZÁSOBNÍK

Adresy 256-511 (\$100-\$1FF) tvoří zásobníkovou paměť pro OS, DOS a Basic.

Strojové instrukce JSR, PHA a přerušeni vždy ukládají data zde. RTS, PLA a RTI je odsud zase vybírají. Po zapnutí počítače nebo po RESET je zásobník inicializován od nejvyšší adresy 511 (\$1FF). Zásobník není nijak chráněn proti přeplnění.

STRÁNKY DVĚ AŽ ČTYŘI

Adresa 512 až 1151 (\$200-\$47F) jsou použity OS k různým účelům: tabulky, datové buffery, skokové vektory po přerušeni NMI a IRQ, skokový vektor po BREAK, pracovní proměnné atd.

512-551**\$200-\$227****INTERRUPT**

Následující adresa využívá operační systém pro skokové vektory po interruptu (přerušeni).

VEKTOR	NÁZEV	UKAZUJE NA
512,513 (\$200,\$201)	VDSLST	49358 (\$C0CE)
514,515 (\$202,\$203)	VPRCED	49357 (\$C0CD)
516,517 (\$204,\$205)	VINTER	49397 (\$C0CD)
518,519 (\$206,\$207)	VBREAK	49397 (\$C0CD)
520,521 (\$208,\$209)	VKEYBD	64537 (\$FC19)
522,523 (\$20A,\$20B)	VSERIN	6691 (\$1A23)
524,525 (\$20C,\$20D)	VSEROR	6630 (\$19E6)
526,527 (\$20E,\$20F)	VSEROC	60140 (\$EAEC)
528,529 (\$210,\$211)	VTIMR1	49357 (\$C0CD)
530,531 (\$212,\$213)	VTIMR2	49357 (\$C0CD)
532,533 (\$214,\$215)	VTIMR4	49357 (\$C0CD)
534,535 (\$216,\$217)	VIMIRQ	49200 (\$C030)
546,547 (\$222,\$223)	VVBLKI	49378 (\$C0E2)
548,549 (\$224,\$225)	VVBLKD	49802 (\$C28A)
550,551 (\$226,\$227)	CDTMA1	60433 (\$EC11)

512,513

\$200,\$201

VDSLST

Vektor pro NMI po DLI. Nachází se zde adresa, na kterou se po DLI skočí a vykoná se příslušný podprogram zakončený instrukcí RTI. Po zapnutí počítače nebo po RESET je ve VDSLST adresa, směřující do ROM na instrukci RTI.

Aby byl podprogram skutečně vykonán, musí se nejprve změnit adresa v VDSLST tak, aby směřovala na počátek vašeho programu a do registru 54286 (\$D40E) zapsat 192, aby ANTIC rozeznal požadavek. Nakonec je nutno nastavit 7. bit v příslušné instrukci DL. V závislosti na grafickém módu jsou pak k dispozici 14 - 61 strojových cyklů. Krátký program využívající DLI si můžete vyzkoušet sami:

```
10 START=PEEK(560)+PEEK(561)*256
20 POKE START+16,130
30 PAGE=1536
40 FOR PGM=PAGE TO PAGE+7
50 READ BYTE
60 POKE PGM,BYTE
70 NEXT PGM
80 DATA 72,169,4,141,1,212,104,64
90 POKE 512,0
100 POKE 513,6
110 POKE 54286,192
120 FOR TEST=1 TO 24
130 PRINT "+++++++"
140 NEXT TEST
150 GOTO 150
```

514,515

\$202,\$203

VPRCED

Vektor sériového přenosu. Prvotní adresa ve VPRCED ukazuje na instrukce PLA, RTI. VPRCED je využit při V/V operacích obvodu PIA vyvolávajícího IRQ.

516,517

\$204,\$205

VINTER

Vektor sériového přenosu. Podobně jako u VPRCED (viz výše).

518,519

\$206,\$207

VBREAK

Programový vektor instrukce BRK mikroprocesoru 6502 (\$00). Nemá nic společného s klávesou BREAK. Dá se použít při odladování strojových programů.

520,521 **\$208,\$209** **VKEYBD**

Vektor přerušení po stisku jakékoliv klávesy kromě BREAK a bočních tlačítek. Přerušení vyvolává POKEY.

522,523 **\$20A,\$20B** **VSERIN**

POKEY vektor přerušení pro příjem dat sériovým kanálem.

524,525 **\$20C,\$20D** **VSEROR**

POKEY vektor přerušení pro vysílání dat sériovým kanálem.

526,527 **\$20E,\$20F** **VSEROC**

Vektor přerušení po ukončení přenosu dat sériovým kanálem. SIO používá tři poslední vektory k řízení sériového kanálu. Během sériového přenosu je vykonávání všech ostatních programů zastaveno.

528,529 **\$210,\$211** **VTIMR1**

Vektor přerušení od hardware časovače POKEY jedna. Normálně se zde nachází adresa směřující na instrukce PLA a RTI. Přerušení přichází v okamžiku, kdy se vynuluje registr AUDF1 (53760; \$D200) po čítání dolů. Hodnota z AUDF1 se přesouvá do registru STIMER (53769; \$D209).(?) Pro spuštění časovače je potřeba dodržet následující postup: Nejprve se definuje v AUDCTL (53768; \$D208) vstupní frekvence časovače (0=64kHz, 1=15kHz, 36=1,79MHz). Následuje nastavení řídicího registru (53761; \$D201). Po uložení přerušovacího podprogramu do volné paměti se do tohoto registru uloží startovací adresa. Pak POKE 53769,0. Nakonec se povolí přerušení POKE 16,x a POKE 53774,x. Přerušení bude generováno po dekrementaci čítače na nulu.

530,531 **\$212,\$213** **VTIMR2**

Vektor přerušení od hardware časovače POKEY dva. Normálně se zde nachází adresa směřující na instrukce PLA a RTI. Přerušení přichází v okamžiku, kdy se vynuluje registr AUDF2 (53762; \$D202) po čítání dolů.

532,533 **\$214,\$215** **VTIMR4**

Vektor přerušení od hardware časovače POKEY čtyři. Normálně se zde nachází adresa směřující na instrukce PLA a RTI. Přerušení přichází v okamžiku, kdy se vynuluje registr AUDF4 (53766; \$D206) po čítání dolů.

534,535

\$216,\$217

VIMIRQ

Hlavní vektor přerušení IRQ, Více informací získáte prostudováním paměti od 53774 (\$D20E).

Adresy od 536 do 558 (od \$218 do \$22E) jsou využity software časovači. Hardware časovače, popsané výše, jsou umístěny v POKEY. Software časovače jsou oproti nim čistě software záležitosti. Dekrementace a ovládání všech pěti software časovačů probíhá během VBLANK procesu (v době snímkového zatemnění obrazovky) každou 1/50 sec.

536,537

\$218,\$219

CDTMV1

Čítač časovače 1. Po dokončené dekrementaci provede program skok na adresu zapsanou v CDTM1 (550,551; \$226,\$227). Je-li nastaven registr CRITIC (66; \$42), pak je během VBLANK zpracováván pouze RTCLK (18-28; \$12-\$14), tento první časovač a registr ATTRACT. Je tomu tak proto, že operační systém využívá prvního časovače k V/V operacím. Proto používejte raději dalších časovačů, aby nedošlo ke kolizím s programy systému. Všechny časovače (1-5) se po dokončené dekrementaci na nulu zastaví.

538,539

\$21A,\$21B

CDTMV2

Čítač časovače 2. Po dokončené dekrementaci provede program skok na adresu zapsanou v CDTM2 (552,553; \$228,\$229). Časovač není dekrementován během V/V operací.

540,541

\$21C,\$21D

CDTMV3

Čítač časovače 3. Na rozdíl od časovačů 1 a 2, časovače 3, 4 a 5 pouze nastaví po dokončeném časování příslušnou návěst. Pro časovač 3 je to návěst CDTMF3 (554; \$22A). OS užívá časovače 3 pro odčasování pauzy před nahráváním nebo čtením z magnetofonu. Původní hodnota je po této operaci ztracena.

542,543

\$21E,\$21F

CDTMV4

Čítač časovače 4. Po dokončené dekrementaci je nastavena návěst CDTMF4 (556; \$22C).

544,545 **\$220,\$221** **CDTMV5**

Čítač časovače 5. Po dokončené dekrementaci je nastavena návěst CDTMF5 (558; \$22E).

546,547 **\$222,\$223** **WBLKI**

Adresa počátku programu VBLANK uloženého v ROM od adresy 49378 (\$C0E2). Tento program se vykonává každou 1/50 sec během NMI.

548,549 **\$224,\$225** **VVBLKD**

Adresa počátku druhé části programu VBLANK obnovujícího všechny registry a zakončeného instrukcí RTI, a který začíná v ROM od 49802 (\$C28A). Adresu ve VVBLKD je možné samozřejmě změnit tak, aby ukazovala na námi sestavený podprogram, který se vykoná vždy během snímkového zatemnění obrazu. Na konci tohoto podprogramu však musí být instrukce JMP VVBLKD! Vykonávání programu může trvat až 20 000 strojových taktů. Normálně je přerušeni snímkovým zatemňováním povoleno nastavením 6. bitu v 54286 (\$D40E).

550,551 **\$226,\$227** **CDTMA1**

Adresa počátku programu, který se vykoná po ukončeném čítání časovače 1. Strojový program musí být zakončen instrukcí RTS. Počáteční inicializace je 60433 (\$EC11).

552,553 **\$228,\$229** **CDTMR2**

Adresa počátku programu, který se vykoná po ukončeném čítání časovače 2. Strojový program musí být zakončen instrukcí RTS. Počáteční inicializace je na adresu 0.

554 **\$22A** **CDTMF3**

Návěst časovače 3. Po dokončené dekrementaci časovače 3 se v CDTMF3 změní hodnota z 255 na 0.

555 **\$22B** **SRTIMR**

Interním programem řízený opakovací časovač. Po každém stisku klávesy se zde zapíše hodnota 40(\$28), která bude postupně dekrementována jestliže SRTIMR dosáhne nulu, klávesa bude ještě stále stisknutá, bude kód klávesy průběžně přepisován do registru CH (764; \$2FC).

556 **\$22C** **CDTMF4**

Návěst časovače 4. Po dokončené dekrementaci časovače 4 se v CDTMF4 změni hodnota z 255 na 0.

557 **\$22D** **INTEMP**

Registr používaný podprogramem SETVBL.

558 **\$22E** **CDTMF5**

Návěst časovače 5. Po dokončené dekrementaci časovače 5 se v CDTMF5 změni hodnota z 255 na 0.

559 **\$22F** **SDMCTL**

Řízení přímého přístupu do paměti (DMA). Zapsáním nula do SDMCTL se urychlí všechny programy asi o 30%, ovšem pokud nám nevdá, že obrazovka je dokonale černá. Pro opětné obnovení zobrazování stačí provést POKE 559,34.

V SDMCTL se využívá pouze šesti bitů.

VOLBA	BIT	DEC.
šířka zobrazování	0	1
šířka zobrazování	1	2
povolení PMG střely	2	4
povolení PMG hráče	3	8
jedno/dvou řádkové rozlišení PMG	4	16
povolení vykonávání obraz. pgm.	5	32

Sledující registr DMACTL je na adrese 54272 (\$D400).

560,561 **\$230,\$231** **SDLSTL**

Počáteční adresa obrazového programu DL. Je to krátký strojový program. Podle kterého pracuje obrazový mikroprocesor ANTIC. DL obsahuje pokyny, kde leží obrazová data, jak mají být interpretována atd. Pro každý grafický mód existují zvláštní instrukce, jiné umožňují přerušování, rolování, skoky.

Sledující registry jsou na adresách 54274,54275 (\$D482,\$D403).

Počáteční adresu DL lze rovněž nalézt pomocí MEMTOP, vrcholu volné paměti RAM:

PRINT PEEK(741)+PEEK(742)*256+1

Bezprostředně za DL začíná v základních grafických módech obrazová paměť.

Pokud se chcete na DL podívat, zkuste tento program:

```
10 DIM D(281)
20 ? CHR$(125),
30 ? : ? "ZADEJTE GRAFICKÝ MÓD: ";
100 INPUT G
110 GRAPHICS G
120 DL=PEEK(560)+PEEK(561)*256
130 FOR J=0 TO 201
140 D(J)=PEEK(DL+J)
150 Z=Z+1
160 IF D(J)=65 THEN POP :GOTO 180
170 NEXT J
180 J=Z: D(J)=PEEK(DL+J)
190 D(J+1)=PEEK(DL+J+1)
200 GRAPHICS 0
210 POKE 201,1: POKE 83,37
220 FOR J=0 TO Z+1
230 ? D(J),
240 NEXT J
250 GOTO 250
```

10: Nejdelší DL je dlouhý 202 byte.

100: Hodnota v rozsahu 0-31.

120: DL získává hodnotu počátku obrazového programu.

140: Do D jsou postupně zapsány všechny instrukce obrazového programu.

150: Z slouží ke zjištění délky obrazového programu.

160: Poslední příkaz je vždy JVB s kódem 65. Za ním následují dva byte adresy.

180: Do D zapsána první část adresy.

190: Do D zapsána druhá část adresy.

200: Normální textový mód.

210: Definování tabulátoru a pravého okraje.

220-240: Tisk celého obrazového programu na obrazovku.

250: Program se ukončí až po stisknutí tlačítka BREAK nebo RESET.

562

\$232

SSKCTL

Řídící registr sériového portu. Sledující registr je na adrese 53775 (\$D20F). Inicializován na 19 (\$13).

Každý bit v registru má svou funkci:

BIT	FUNKCE
0	povolení obsluhy obvodu klávesnice
1	povolení obsluhy obvodu klávesnice
2	převod hodnoty potenciometru je proveden během trvání dvou obrazových řádků místo během trvání jednoho televizního snímku
3	sériový přenos je prováděn dvoutónově místo logického výstupu L/H
4-6	řízení módu sériového portu
7	nastavení výstupu sériového portu na nulu

563 **\$233** **LCOUNT**

Dočasný čítač pro přenos dat.

564 **\$234** **LPENH**

Hodnota horizontální polohy světelného pera. Sledující registr je na 54284 (\$D40C). Obsah registru může nabýt hodnoty od 0 do 227.

565 **\$235** **LPENV**

Hodnota vertikální polohy světelného pera. Sledující registr je na 54285 (\$D40D). Obsah registru může nabýt hodnoty od 16 do 111. Hodnota je stejná jako ve VCOUNT pro dvouřádkové rozlišení (viz 54283; \$D40B).

Oba registry pozice světelného pera jsou modifikovány při stisknutí tlačítka trigger (spoušť).

566, 567 **\$236, \$237** **BRKKY**

Vektor přerušeni po stisku klávesy BREAK. Je zde uložena adresa, na níž OS přejde po zmáčknutí klávesy BREAK. Inicializace na 49298 (\$C092). Adresu je samozřejmě možné změnit a napsat si vlastní BREAK program.

568, 569 **\$238, \$239** **RELADR**

Používá pouze 1200XL.

568, 569 **\$238, \$239** **VPIRQ**

Používají všechny XL/XE kromě 1200XL. Vektor přerušeni paralelní sběrnici.

570 **\$23A** **CDEVIC**

Adresový byte. Používá SIO při sériové komunikaci.

571 **\$23B** **CCOMND**

Příkazový kód používaný SIO.

572 **\$23C** **CAUX1**

První pomocný příkazový byte. Je naplňován z adresy 778 (\$30A).

573 **\$23D** **CRUX2**

Druhý pomocný příkazový byte. Je naplňován z adresy 779 (\$30B).

574 **\$23E** **TEMP**

Dočasný registr pro SIO.

575 **\$23F** **ERRFLG**

Chybová návěst SIO.

576 **\$240** **DFLAGS**

Disketová návěst nacházející se v prvním byte souboru v sektoru jedna disku.

577 **\$241** **DBSECT**

Číslo sektoru čteného z prvního diskového zápisu.

578,579 **\$242,\$243** **BOOTAD**

Adresa, od které se budou ukládat data čtená z disku, formálně (s DOS) je zde adresa 1792 (\$700) a dočasně se ukládá rovněž do registru RAMLO (4,5).

560 **\$244** **COLDST**

Návěst startu počítače. Normálně je zde nula a při této hodnotě se po stisku RESET provede běžná RESET inicializace. Bude-li zde však při RESET jednička, provede se inicializace paměti jako po zapnutí počítače.

561 **\$245** **RECLEN**

Proměnná pro délku záznamu RLR.

582 **\$246** **DSKTIM**

Diskový registr časového zpoždění.

583-622 **\$247-\$26E** **LINBUF**

Čtyřicet bytů řádkového bufferu používaných k dočasné úschově fyzického řádku textu v době, kdy obrazový editor přesouvá obrazová data. Do bufferu ukazuje ADDRESS (100,101; \$64,\$65).

583 **\$247** **PDVMSK**

Maska výběru zařízení. Sledující registr na 53759 (\$D1FF) je aktivní pouze v době, kdy OS nepracuje s ROM, ve které jsou uloženy matematické programy. Na paralelní sběrnici může být až osm přístrojů a každý bit tohoto registru přísluší jednomu přístroji. Pouze pro 1200XL.

584 **\$248** **SHPDVS**

Každý bit tohoto registru reprezentuje jeden z osmi přístrojů na paralelní sběrnici. Pouze pro 1200XL.

585 **\$249** **PDMSK**

Maska přerušení od přístroje na paralelní sběrnici. Pouze pro 1200XL.

586,587 **\$24A,\$24B** **RELADR**

Relativní adresa při práci s paralelní sběrnici. Pouze pro 1200XL.

588 **\$24C** **PPTMPA**

Dočasně používaný registr při práci s paralelní sběrnici. Pouze pro 1200XL.

589 **\$24D** **PPTMPX**

Dočasně používaný registr při práci s paralelní sběrnici. Pouze pro 1200XL.

590-618 **\$24E-\$26A** **-----**

Volná paměť.

619 **\$26B** **CHSALT**

Stránkový vektor alternativní znakové sady pro 1200XL, inicializovaný na 204 (\$CC). Řada XL má k dispozici dvě znakové sady; první je od 52224 (\$CC00); druhá od 57344 (\$E000).

620 **\$26C** **VSFLAG**

Dočasně používaný registr při jemném rolování.

621 **\$26D** **KEYDIS**

Hodnota 255 zcela eliminuje použití klávesnice. POKE 621,0 použití klávesnice opět obnovuje. U 1200XL lze obnovit použití klávesnice pomocí CTRL-F1.

622 **\$26E** **FINE**

Povolení jemného rolování v grafickém modu 0 (text). POKE 622,0 je pro běžné rolování (základní nastavení), 255 je pro jemné rolování. Při dlouhém listování programem se únavou projeví trhavé pohyby řádku. Zkuste jemné rolování obrazu při LISTu. DL je delší jen o jeden byte. Pokud se vám nechce měnit DL zkuste dosadit do FINE nenulovou hodnotu a provést LIST.

623 **\$26F** **GPRIOR**

Registr prioritního zobrazení. Sledující registr je na 53275 (\$D01B). Umožňuje vytvářet optický efekt přední - zadní. Rovněž umožňuje použít všechny čtyři střely, jako pátého hráče a hráčům dát různé barvy. Registr můžete rovněž využít k výběru grafického módu GTIA (tj. 9,10,11).

PRIORITNÍ VOLBA ZOBRAZENÍ	BIT
hráč 0-3, obraz 0-3, pozadí	8
hráč 0-1/ obraz 0-3, hráč 2-3, pozadí	1
obraz 0-1/ hráč 0-3, pozadí	2
obraz 0-1/ hráč 0-3, obraz 2-3, pozadí	3
DALŠÍ FUNKCE:	
čtyři střely = pátý hráč	4
překrytí hráči mají třetí barvu	5
GRAPHICS 9 (GTIA mód)	6
GRAPHICS 10 (GTIA mód)	7
GRAPHICS 11 (GTIA mód)	6,7

Nastavením pátého bitu vzniká třetí barva v okamžiku překrytí dvou hráčů; 0 a 1 nebo 2 a 3. Příslušné barvové registry jsou svázané logickou funkcí OR. Jestliže je např. hráč 1 růžový a hráč 0 modrý, pak

sečtením vznikne zelená. Nulový pátý bit zobrazí překryté plochy vždy černě.

V GTIA modu 3 lze získat 16 odstínů jediné barvy. Zkuste:

```
10 GRAPHICS 3:SETCOLOR 4,9,0
20 FOR LOOP=1 TO 15:COLOR LOOP
30 FOR LINE=1 TO 2
40 FOR TEST=1 TO 25:PLOT 4+TEST,LOOP+LINE+SPACE:NEXT TEST
45 NEXT LINE
50 SPACE=SPACE+4
60 NEXT LOOP
70 GOTO 70:REM BEZ TOHOTO RADKU BY SE OBRAZOVKA OKAMZITE NASTAVILA DO
MODU 0
```

V GTIA módu 10 máte k dispozici všech devět barvových registrů, přičemž každý z nich nese informaci o barvě i jasu.

```
10 N=0:GRAPHICS 10
20 FOR Q=1 TO 2
30 FOR B=0 TO 8:POKE 704+B,N*16+A
35 IF A>15 THEN A=0
40 COLOR B
45 A=A+1:N=N+1
58 IF N>14 THEN N=0
60 NEXT B
65 TRAP 70:NEXT Q
70 POP:N=N+1:FOR Z=1 TO 200:NEXT Z
75 GOTO 30
```

GTIA mód 11 je podobný módu 3. Máte možnost zobrazit 16 různých barev v jednom odstínu.

```
10 GRAPHICS 11
20 FOR LOOP=0 TO 79:COLOR LOOP:PLOT LOOP,0:DRAWTO LOOP,191:NEXT LOOP
30 GOTO 30
```

GTIA módy jsou nulovány příkazem OPEN.

624

\$270

PADDL0

Poloha potenciometru 0 (controller 1). Pohybuje se v rozsahu 0-228 (\$E4). Sledující registr je na adrese 53760 (\$D200).

625

\$271

PADDL1

Poloha potenciometru 1 (controller 1). Pohybuje se v rozsahu 0-228 (\$E4). Sledující registr je na adrese 53761 (\$D201).

626 **\$272** **PADDL2**

Poloha potenciometru 2 (controller 2). Pohybuje se v rozsahu 0-228 (\$E4). Sledující registr je na adrese 53762 (\$D202).

627 **\$273** **PADDL3**

Poloha potenciometru 3 (controller 2). Pohybuje se v rozsahu 0-228 (\$E4). Sledující registr je na adrese 53763 (\$D203).

628-631 **\$274-\$277** **PADDL4-7**

Série XL má pouze dva porty, proto jsou aktivní pouze potenciometry 0-3. Jinak se nachází hodnoty stejné jako v PADDL0-3 i zde.

632 **\$278** **STICK0**

Hodnota páky 0 (controller 1). Sledující registr je na adrese 54016 (\$D300). Hodnota v registru závisí na poloze páky:

POLOHA	DEKADICKY	BINÁRNĚ
vlevo	11	1011
vlevo nahoře	10	1010
nahoře	14	1110
vpravo nahoře	06	0110
vpravo	07	0111
vpravo dole	05	0101
dole	13	1101
vlevo dole	09	1001
klidová poloha	15	1111

633 **\$279** **STICK1**

Stejně jako STICK0. Sledující registr na adrese 54017 (\$D301).

634,635 **\$27A,\$27B** **STICK2-3**

Série XL má k dispozici pouze porty pro dvě páky 0 a 1. Nachází se zde hodnoty stejné jako ve STICK0-1.

636 **\$27C** **PTRIG0**

Řídicí vstup potenciometru (controller 1). Při nestisknutém tlačítku je zde 1, při stisknutém 0. Další tři registry mají obdobnou funkci. Sledující registr je na adrese 54016 (\$D300).

637 **\$27D** **PTRIG1**

Viz výše (controller 1).

638 **\$27E** **PTRIG2**

Viz výše. (controller 2).

639 **\$27F** **PTRIG3**

Viz výše. (controller 2).

640-643 **\$280-\$283** **PTRIG4-7**

Nepoužito (viz PADDL4-7).

644 **\$284** **STRIG0**

Spoušť páky (controller 1). Při nestisknutém tlačítku je zde 1, při stisknutém 0. Další registr má obdobnou funkci. Sledující registry jsou na adresách 53264-53265 (\$D010-\$D811).

645 **\$285** **STRIG1**

Viz výše. (controller 2).

646, 647 **\$286, \$287** **STRIG2-3**

Nepoužito (viz STICK2-3).

648 **\$288** **HIBYTE**

Registr vyššího byte RLR.

649 **\$289** **WMODE**

Registr čtecího nebo zapisovacího módu magnetofonu. Nula značí čtení, 128 (\$80) zápis.

650 **\$28A** **BLIM**

Množství dat v bufferu magnetofonu (buffer začíná na adrese 1021; \$3FD). Obsahuje číslo, udávající množství dat, které budou během čtení z magnetofonu přijaty v jednom bloku. Při zápisu na magnetofon udává množství platných vysílaných dat opět během jednoho bloku. Hodnota v BLIM se pohybuje v rozsahu 0-128.

651 **\$28B** **IMASK**

Nepoužít.

652,653 **\$28C,\$28D** **JVECK**

Dočasný skokový vektor. Nepoužít.

654,655 **\$28E,\$28F** **NEWADR**

Nový adresový vektor. Použit v RLR.

Registry 656 až 703 (\$290 až \$2BF) používají obslužné procedury displeje.

V módech s děleným stínítkem je textové okénko řízeno textovým editorem (E:), zatímco grafická část obrazu je řízena obsluhou displeje (S:), který používá dva zvláštní IOCB bloky.

656 **\$290** **TXTRW**

Řádková poloha kurzoru v textovém okénku. Muže nabýt velikosti od 0 do 3 (okénko má pouze čtyři řádky).

657,658 **\$291,\$292** **TXTCOL**

Sloupcová poloha kurzoru v textovém okénku. Muže nabýt velikosti od 0 do 39. Vyšší byte bude vždy nulový, neboť okénko má pouze 40 sloupců. Příkazy POSITION, PLOT, LOCATE se k textovému okénku nevztahují, proto je nutné ke změně hodnot v TXTRW a TXTCOL použít POKE.

659 **\$293** **TINDEX**

Obsahuje číslo GRAPHICS módu s děleným stínítkem. Je to ekvivalentní hodnota k DINDEX (87; \$57) a vždy je zde nula pokud je nula i v registru 128 (\$80). TINDEX se inicializuje na nulu (GRAPHICS 0). Pro textové okénko ve kterémkoliv grafickém módu můžete použít upravený DL, musí se však patřičně přizpůsobit hodnota v tomto registru.

660,661 **\$294,\$295** **TXTMSC**

Adresa počátku paměti textového okénka (odpovídá levému hornímu rohu textového okénka). Při děleném stínítku je to obdoba SAVMSC 88,89 (\$58,\$59).

662-667 **\$296-\$29B** **TXTOLD**

Tyto adresy jsou při děleném stínítku obdobami OLDROW (90; \$5A), OLDCOL (91,92; \$5B,\$5C), OLDCHR (93; \$5D) a OLDADR (94,95; \$5E,\$5F).

668 **\$29C** **CRETRY**

Počet opakování příkazu po neúspěšném pokusu o čtení sektoru nebo formátování disku, případně počet opakování pokusů o komunikaci s periferním zařízením. Prvotní nastavení je 13.

669 **\$29D** **HOLD3**

Dočasný registr.

670 **\$29E** **SUBTMP**

Dočasný registr.

671 **\$29F** **HOLD2**

Dočasný registr.

672 **\$2A0** **DMASK**

Maska obrazových bodů. DMASK obsahuje nuly ve všech bitech, které nenáleží k danému obrazovému bodu a jedničky v bitech, které k obrazovému bodu náleží. Například:

11111111	Módy 0, 1 a 2:	jeden bod na obrazový byte
11110000	Módy 9, 10 a 11:	dva body na byte
00001111		
11000000	Módy 3, 5 a 7:	čtyři body na byte
00110000		
00001100		
00000011		
10000000	Módy 4, 6 a 8:	osm bodů na byte
01000000		
atd.		
00000001		

Bod (vlastně obrazová buňka nebo obrazový element) je logická jednotka velikosti obrazu, která záleží na zvoleném grafickém módu. Nejmenší bod je v GRAPHICS 8 (pouze jeden obrazový řádek).

Následující tabulka obsahuje základní údaje prvních osmi módů:

	TEXTOVÉ MODY				GRAFICKÉ MODY					
grafický mód obr. řádky	0	1	2	3	4	5	6	7	8	
na bod	8	8	16	8	4	4	2	2	1	
bitů										
na bod	1	1	1	2	1	2	1	2	1	
barvových bodů										
na bod	.5	1	1	4	2	2	1	1	.5	
znaků										
na řádek	40	20	20	-	-	-	-	-	-	
bodů										
na šířku obr.	-	-	-	40	80	80	160	160	320	

673**\$2A1****T MPLBT**

Dočasný registr bitové masky.

674**\$2A2****ESCFLG**

Escape návěst. Normálně je zde nula. Po zmáčknutí klávesy ESC se zde nastaví 128 a k vynulování dojde okamžitě po výstupu dalšího znaku. Pokud je nutné tisknout na obrazovku řídicí ATASCII znaky bez použití klávesy ESC, stačí nastavit DSPFLG (766; \$2FE) na nenulovou hodnotu.

675-689**\$2A3-\$2B1****TABMAP**

Maska zarážky tabulátoru. Všechny 128 bitů 15 byte registru TABMAP se vztahuje k logickému řádku. Jednička ve kterémkoliv bitu znamená nastavení zarážky tabulátoru. K dispozici je tedy celkem 128 zarážek, neboť logický řádek má v GR.0 délku 3 fyzických řádků x 40 sloupců. Z toho plyne také to, že každý ze tří fyzických řádků v logickém řádku může mít různé nastavené zarážky.

Chceme-li například nastavit zarážky na pozicích 5, 23, 27 a 32:

```
0000100000000000000000001000100001
```

bude mít prvních pět byte hodnoty:

```
00001000 = 8
00000000 = 0
00000010 = 2
00100001 = 33
00000000 = 0
```


Základní nastavení zářezek po zapnutí počítače nebo po OPEN (S: nebo E:) je 1 (tj. 00000001). Znamená to, že zářezky jsou na pozicích 7, 15, 23 atd. až 119.

TABMAP funguje i v textovém okénku děleného stínítka.

690-693**\$2B2-\$2B5****LOGMAP**

Bitový záznam začátku logických řádků. Každý ze 24 bitů (3 byte x 8 bitů) zastupuje jeden fyzický řádek na obrazovce v módu 0. Bit v LOGMAP se nastaví na 1, začíná-li na odpovídající pozici logický řádek. Formát je patrný z následující tabulky:

BIT	7	6	5	4	3	2	1	0	BYTE
ŘÁDEK	0	1	2	3	4	5	6	7	690
	8	9	10	11	12	13	14	15	691
	16	17	18	19	20	21	22	23	692
	-	-	-	-	-	-	-	-	

Čtvrtý byte je ignorován. Všechny bity registru LOGMAP jsou nastaveny na 1 při OPEN (textový mód), při CLEAR obrazovky, při GRAPHICS nebo při RESET.

694**\$2B6****INVFLG**

Návěst invertování znaku. Inicializován na nula. Při hodnotě 128 (\$80) jsou při psaní znaků skrze klávesnici všechny invertovány. INVFLG lze změnit pomocí POKE nebo stiskem klávesy INVERSE.

Obsluha displeje pomocí funkce OR vždy sčítá ATASCII kód znaku přicházející z klávesnice s hodnotou v INVFLG.

Znamená to tedy, že INVFLG pracuje při vstupu, nikoli při výstupu.

Budete se například marně pokoušet vytisknout po POKE 694,128:PRINT "ahoj" inverzně. Větší úspěch zaznamenáte při POKE 694,128 a pak INPUT A\$. Všechny následně vložené znaky budou na obrazovku tištěny inverzně. Viz rovněž SHFLOK (702; \$2BE).

695**\$2B7****FILFLG**

Návěst funkce FILL (XIO 18, ...). Při DRAWTO je zde nula, při nenulové hodnotě FILFLG probíhá FILL.

696**\$2B8****TMPROW**

Dočasný registr používaný ROWCRS (84; \$54).

697,698**\$2B9, \$2BA****TMPCOL**

Dočasný registr používaný COLCRS (85,86; \$55,\$56).

699**\$2BB****SCRFLG**

Návěst rolování je nastavena vždy, když probíhá rolování. Čítá počet fyzických řádků minus jedna, které byly smazány od horního okraje obrazovky. Protože logický řádek zaujímá tři fyzické řádky, může registr nabýt hodnot od 0 do 3.

Rolování textového okénka je ekvivalentní k rolování celé obrazovky v GR.0. To znamená, že 20 řádků (800 byte), umístěných v paměti pod textovým okénkem roluje rovněž (není to samozřejmě vidět). Může to vyvolat problémy kupříkladu při nevhodném umístění PMG.

700**\$2BC****HOLD4**

Dočasný registr používaný pouze příkazem DRAWT0. Použit k zápisu a obnově hodnoty v ATACHR (763; \$2FB) během operace FILL.

701**\$2BD****DRETRY**

Počet pokusů o komunikaci s periferním zařízením (tiskárna). Prvotní hodnota je 1.

702**\$2BE****SHFLOK**

Návěst kláves SHIFT a CTRL. Při psaní malých znaků je v registru nula, při psaní velkých znaků 64 (\$40) (základní nastavení). Po vložení 128 (\$80) do SHFLOK budou tištěny pseudografické znaky.

Jakékoliv jiné hodnoty mohou, způsobit zhroucení systému. Krátký program převádí jakýkoliv klávesový vstup (inverze, malé znaky, pseudografika) na normální velké znaky:

```
10 OPEN #2,4,0,"K:"
```

```
20 GET #2,A
```

```
30 GOSUB 1000
```

```
40 PRINT CHR$(A);
```

```
50 GOTO 20
```

```
.
```

```
.
```

```
1000 IF A=155 THEN 1030:REM OŠETŘENÍ PŘI STISKU KLÁVESY RETURN
```

```
1010 IF A>128 THEN A=A-128:REM ZMĚNA INVERZNÍCH ZNAKŮ NA NORMÁLNÍ
```

```
1020 IF PEEK(702)=0 AND A>96 THEN A=A-32:REM ZMĚNA MALÝCH ZNAKŮ NA VELKÉ
```

```
1830 POKE 702,64
```

```
1040 POKE 694,0
```

```
1050 RETURN
```

703**\$2BF****BOTSCR**

Udává celkové množství řádků použitelných pro tisk. 24 (\$18) je zde pro GR.0, pro módy s děleným stínítkem hodnota 4 a pro ostatní módy je BOTSCR vždy 0.

Pokud sem vložíte 4 při GR.0, budou se dolní čtyři řádky chovat jako textové okénko, tzn. normální rolování a tisk pomocí PRINT. Horních 20 řádků pak bude dosažitelných přes PRINT #6 bez možnosti rolování.

Registry 704 až 712 (\$2C0 až \$2C8) jsou barvé registry.

Sledující registry jsou na adresách 53266 až 53274 (\$D012 až \$D01A). Některé jsou nastavitelné příkazem SETCOLOR (COLOR0-4), všechny pomocí POKE použitím následujícího předpisu:

VÝSLEDNÁ BARVA = BARVA * 16 + JAS

přičemž hodnoty barev jsou v následující tabulce:

BARVA	HODNOTA
šedá	0
světle oranžová	1
oranžová	2
červeně oranžová	3
růžová	4
nachová	5
nachově modrá	6
azurově modrá	7
blankytně modrá	8
světle modrá	9
tyrkysová	10
zeleně modrá	11
zelená	12
žlutozelená	13
oranžově zelená	14
světle oranžová	15

Obsazení bitů v jednotlivých registrech PCOLR a COLOR vypadá:

BIT	7	6	5	4	3	2	1	0
	-barva-			-	jas		-	nepoužit

Následuje program, který ve strojové rutině rotuje barvami v registrech 705 až 712.

```
10 DIM ROT$(30)
20 FOR LOOP=1 TO 27
30 READ BYTE
40 ROT$(LOOP,LOOP)=CHR$(BYTE)
50 NEXT LOOP
```

... program pokračuje na další stránce

```
100 CHANGE=USR(ADR(ROT$))
105 FOR LOOP=1 TO 200:NEXT LOOP:GOTO 100
110 DATA 104,162,0,172,193,2,189,194,2,157
120 DATA 193,2,232,224,8,144,245,140,280,2
130 DATA 96,65,65,65,65,65,65
```

704 **\$2C0** **PCOLR0**

Barvový registr pro hráče 0 a střelu 0. Sledující registr je na 53266 (\$D012). Čtyři barvové registry 704-707 jsou využívány při PMG a v GTIA módu a žádný z nich nelze nastavit příkazem SETCOLOR.

705 **\$2C1** **PCOLR1**

Barvový registr pro hráče 1 a střelu 1 PM grafiky. Sledující registr na 53267 (\$D013).

706 **\$2C2** **PCOLR2**

Barvový registr pro hráče 2 a střelu 2 PM grafiky. Sledující registr na 53268 (\$D014).

707 **\$2C3** **PCOLR3**

Barvový registr pro hráče 3 a střelu 3 PM grafiky. Sledující registr na 53269 (\$D015).

Pokud jsou čtyři střely použity jako pátý hráč, je jeho barva určena registrem COLOR3.

708 **\$2C4** **COLOR0**

Barvový registr pozadí 0. Sledující registr na 53270 (\$D016). Registr je možné modifikovat Basic příkazem SETCOLOR i POKE. V GRAPHICS 1 a 2 je zde uložena barva velkých znaků.

709 **\$2C5** **COLOR1**

Další čtyři registry jsou stejné jako COLOR0 jen s rozdíly v příkazu SETCOLOR. V GR.1 a 2 nese registr informaci o barvě malých znaků. Sledující registr na 53271 (\$D017).

710 **\$2C6** **COLOR2**

V GR.1 a 2 obsahuje registr barvu velkých inverzních znaků. Sledující registr na 53272 (\$D018).

711 **\$2C7** **COLOR3**

V GR.1 a 2 obsahuje registr barvu malých inverzních znaků. Sledující registr na 53273 (\$D019).

712 **\$2C8** **COLOR4**

Tento barvový registr obsahuje v některých grafických módech barvu okraje.

Prvotní nastavení všech registrů PCOLR je 0.

Prvotní nastavení registrů COLOR ukazuje následující tabulka:

REGISTR	VÝSL.	BARVA	BARVA	JAS
708 (CO.0)	40	2	2	8
709 (CO.1)	202	12	12	10
710 (CO.2)	148	9	9	4
711 (CO.3)	70	4	4	6
712 (CO.4)	0	0	0	0

713, 714 **\$2C9, \$2CA** **RUNADR**

Registr startovací adresy RLR.

715, 716 **\$2CB, \$2CC** **HIUSED**

Používá RLR.

717, 718 **\$2CD, \$2CE** **ZHIUSE**

Používá RLR.

719, 720 **\$2CF, \$2D0** **GBYTEA**

Používá RLR.

721, 722 **\$2D1, \$2D2** **LOADAD**

Používá RLR.

734

\$2DE

PBPNT

Ukazatel bufferu tiskárny. Ukazuje běžnou pozici (byte) v bufferu tiskárny. Rozsah je od nula po hodnotu PBUFSZ (viz níže).

735

\$2DF

PBUFSZ

Velikost bufferu tiskárny. Normální velikost bufferu je 40 byte. Obsluha tiskárny kontroluje před tiskem, zda je hodnota stejná jako PBPNT a pokud ano, vyšle celý obsah bufferu na tiskárnu.

736-739

\$2E0-\$2E3

GLBABS

DOS využívá tuto část paměti k uložení globálních proměnných. Využití je popsáno níže (viz RUNAD a INITAD).

736,737

\$2E0,\$2E1

RUNAD

Používá DOS jako startovací adresu přečtenou z diskového sektoru jedna nebo z binárního souboru. Po kompletním binárním načtení přechází řízení normálně do DOS menu. DOS ale také může rovnou po vykonaném čtení přejít na adresu zapsanou právě zde. Je-li zde např. zapsána adresa 40960 (\$A000), bude volán Basic.

738,739

\$2E2,\$2E3

INITAD

Inicializační adresa čtení z disku. Zaváděný soubor musí nejprve přenést hodnotu adresy do RUNAD nebo do INITAD. Spuštění programu, jehož startovací adresa je v RUNAD bude provedeno až po načtení celého souboru. Návrat do DOS z programu je umožněn instrukcí RTS.

740

\$2E4

RAMSIZ

Velikost RAM, množství stránek RAM (stránka = 256 byte). Obsahuje stejnou hodnotu jako RAMTOP (106; \$6A), tzn. že se po zapnutí počítače inicializuje na 160 (pro 800XL).

741,742

\$2E5,\$2E6

MEMTOP

Nachází se zde adresa vrcholu volné paměti RAM, do které může Basic ještě ukládat data a program. Hodnota je zde inicializována po zapnutí počítače přenesením hodnoty z TRAMSZ (6). Tato hodnota je obnovována po RESET, po změně grafického módu nebo po OPEN (E: nebo S:). V paměti bezprostředně nad touto adresou začíná DL a za ním se nachází obrazová paměť. Používá Basic i OS.

743,744

\$2E7,\$2E8

MEMLO

Nachází se zde adresa dolní hranice volné paměti RAM.

Inicializováno na 1792 (\$700). Ke změně adresy dochází např. zavedením DOS (7420; \$1CFC) nebo jakýmkoliv aplikačním programem. Používá OS.

Adresa dolní hranice volné paměti pro Basic je v LOMEM (128,129; \$80,\$81).

Je to adresa první volné buňky v RAM pro programové použití.

Hodnota v MEMLO je měněna OS jen po RESET. Po Basic příkazu NEW se odsud přečte hodnota do LOMEM, nikdy však ne po RUN, LOAD nebo RESET.

Pod MEMLO lze vytvořit chráněnou oblast paměti podobně jako nad RAMTOP.

745

\$E9

HNDLOD

Návěst RLR.

746-749

\$2EA-\$2ED

DVSTAT

Čtyři statusové registry použité při V/V operacích:

746 (\$2EA) - chybový status a příkazový statusový byte. Jednotlivé bity nastavené na 1 znamenají:

<u>BIT</u>	<u>CHYBA</u>
------------	--------------

0	bylo přijato chybné složení příkazu
1	bylo přijato chybné složení dat
2	výstupní operace nebyla úspěšná
3	disk je chráněn proti zápisu
4	system není aktivní
7	periferní zařízení má vlastní "inteligenci" (disk má vlastní mikroprocesor).

747 (\$2EB) - statusový byte. Při diskových operacích je zde hodnota statusového registru diskového řadiče. Pro interface 850 je zde status pro DSR, CTS, CRX a RCV pokud není souběžný V/V kanál aktivní. Může zde také být hodnota AUX2 z předešlé operace (viz IOCB popsany od 832 do 959).

748 (\$2EC) - hodnota v sekundách maximální časové prodlevy.

749 (\$2ED) - počet byte ve výstupním bufferu. Viz 850 Manual, str. 43.

Pokud je souběžný V/V kanál aktivní, STATUS příkaz vrací hodnotu znaku ve vstupním bufferu do 747 a 748 a hodnotu znaku ve výstupním bufferu do 749.

750,751

\$2EE,\$2EF

CBAUDL/H

Obsahuje rychlost v baudech pro magnetofon. Inicializuje se na 1484 (\$5CC) což představuje přenosovou rychlost 600 baudů (bitů za sekundu). Po výpočtu baudové rychlosti bude CBAUDL/H obsahovat hodnotu od POKEY pro rychlostní korekci. Začátek každého magnetofonového zápisu totiž obsahuje krátkou sekvenci jedniček a nul, která je pro korekci využita.

752

\$2F0

CRSINH

Návěst potlačení kurzoru. Nula znamená viditelný kurzor a nastaví se vždy po zapnutí počítače, po RESET, po BREAK nebo po příkazu OPEN (E: nebo S:). Jakákoliv jiná hodnota než nula, činí kurzor neviditelným. Ale pozor, až po prvním následujícím přesunu kurzoru nebo po smazání obrazovky. Viz rovněž CHART (755; \$2F3).

753

\$2F1

KEYDEL

Návěst zpožděné klávesy. Po stlačení kterékoliv klávesy se do KEYDEL запиše 3. Po uvolnění klávesy se během každého druhého VBLANK registr dekrementuje až na nulu. Dokud je registr nenulový, systém ignoruje stisk kterékoliv jiné klávesy.

754

\$2F2

CH1

Obsahuje kód předcházející stlačené klávesy, uložené sem z CH (764; \$2FC).

Jestliže bude nový kód klávesy roven hodnotě v tomto registru, kód bude akceptován až po patřičné časové prodlevě.

755

\$2F3

CHART

Registr způsobu zobrazení znaků.

0 - Inverze znaků.

1 - Inverzní znaky budou tištěny jako prázdná mezera.

2 - Normální znaky.

3 - Inverzní znaky budou tištěny jako prázdná inverzní mezera.

4 - 7 je totéž jako 0 - 3, ale s opačně zobrazovanými znaky.

CHART rovněž určuje způsob zobrazení kurzoru: je viditelný s hodnotami 2 a 6, neprůhledný s 3 a 7. Kurzor zcela chybí s hodnotami 0, 1, 4 a 5. Sledující registr na 54273 (\$D401). Inicializováno na 2. Malý program ukáže možnosti CHART:

10 GRAPHICS 0:POKE 703,4

20 POSITION 5,12

... program pokračuje na další stránce

```
30 PRINT #6;"BLIK blik BLIK blik"  
40 FOR I=0 TO 7  
50 FOR J=0 TO 7  
60 PRINT I;"/";J:PRINT  
70 POKE 755,I  
80 FOR W=0 TO 100:NEXT W  
90 POKE 755,J  
100 FOR W=0 TO 100:NEXT W  
110 IF PEEK(764)=255 THEN GOTO 70  
120 OPEN #1,4,0,"K:"  
130 GET #1,D  
140 CLOSE #1,D  
150 NEXT J  
160 NEXT I
```

Pozn: malé znaky v řádku 30 musí být tištěny inverzně.

10 až 30: Obrazovka rozdělena na grafickou a textovou část v GR.0. Text je tištěn do středu obrazovky.

40 a 50: Smyčka postupně projdou všemi možnými kombinacemi 8*8.

60: V textovém okénku je kontrolní aktuální kombinace.

80 až 100: Možnost změny časové prodlevy.

110 až 130: Stlačením klávesy započne nová kombinace.

756

\$2F4

CHBAS

Stránková adresa znakové sady (pouze vyšší byte adresy, skutečnou hodnotu adresy obdržíte po PRINT PEEK(756)*256). Sledující registr je na 54281 (\$D409). Inicializuje se na 224 (\$E0) pro velké znaky a čísla. Vložení hodnoty 226 (\$E2) do CHBAS se v GR. 1 a GR.2 budou zobrazovat malé a grafické znaky. V obou uvedených módech však není možné současně zobrazovat malé i velké znaky.

Jedná se samozřejmě o standardní americkou sadu znaků. Nicméně v rámci elektronické kolonizace je u XL k dispozici také evropská sada se španělskými, německými atd. specifickými znaky. Je dosažitelná po POKE 756,204.

Každý znak je ve znakové sadě definován osmicí byte. Jeden byte šířka x osm byte výška. Celá znaková sada zabírá 1024 byte (128 znaků x 8byte). Dva příklady poslouží k názornějšímu vysvětlení:

jeden byte šířka:

```
00100000 = 32 #  
00010000 = 16 #  
00010000 = 16 #  
00010000 = 16 #  
00011110 = 30 #####  
00000010 = 2 #  
00001100 = 12 ##  
00010000 = 16 #
```

```

10000001 = 129      #      #
10011001 = 153      #    ##  #
10111101 = 189      #   #### #
11111111 = 255      #####
11111111 = 255      #####
10111101 = 189      #   #### #
10011001 = 153      #    ##  #
10000001 = 129      #      #
    
```

Českou abecedu si můžete do textových programů vložit následujícím programem, který voláte pomocí GOSUB. České znaky se tisknou se stisknutou klávesou CTRL.

```

10000 REM CESKA ABACEDA
10005 REM VYTVORENI CHRANENEHO PROSTORU
10010 ORIG=PEEK(106):CHSET=(ORIG-4)*256:POKE 106,ORIG-8 GRAPHICS 0:POKE
756,ORIG-4:POKE 559,0
10015 REM PRESUN STANDARDNI ZNAKOVE SADY DO RAM
10020 REM BLOKY VLASTNICH ZNAKU SE VLOZI DO RAM
10025 FOR X=0 TO 1023:POKE CHSET+X,PEEK(224*256+X):NEXT X
10030 RESTORE 10100
10035 FOR A=0 TO 215:READ B:POKE 512+A+CHSET,B:NEXT A
10040 FOR A=0 TO 7:READ B:POKE 768+A+CHSET,B:NEXT A
10050 FOR A=0 TO 7:READ B:POKE 984+A+CHSET,B:NEXT A
10060 POKE 559,34:RETURN
10100 DATA 36,0,60,102,102,102,60,0
10101 DATA 4,8,60,6,62,102,62,0
10102 DATA 28,126,12,24,48,96,126,0
10103 DATA 28,8,60,96,96,96,60,0
10104 DATA 56,22,6,62,102,102,62,0
10105 DATA 4,8,68,182,126,96,60,8
10106 DATA 7,58,102,96,96,102,60,8
10107 DATA 7,58,96,60,6,6,60,0
10108 DATA 16,82,70,102,102,102,126,0
10109 DATA 4,8,0,56,24,24,60,0
10110 DATA 24,24,66,102,102,102,62,0
10111 DATA 2,122,24,24,24,24,126,0
10112 DATA 1,61,102,102,102,102,60,0
10113 DATA 40,24,60,102,102,126,102,0
10114 DATA 56,16,124,102,102,102,102,0
10115 DATA 4,8,60,102,102,102,60,0
10116 DATA 0,124,100,124,100,124,96,0
10117 DATA 2,26,60,102,102,126,102,0
10118 DATA 28,8,124,102,96,96,96,0
10119 DATA 28,8,62,96,60,6,124,0
10120 DATA 7,26,126,24,24,24,14,0
10121 DATA 4,8,98,102,102,102,62,0
10122 DATA 7,122,102,102,124,108,102,0
10123 DATA 28,8,60,102,126,96,60,0
10124 DATA 1,125,96,124,96,96,126,0
    
```

10125 DATA 4,8,98,102,102,62,12,120
10126 DATA 28,8,126,12,24,48,126,0
10127 DATA 36,0,182,102,102,102,62,0
10128 DATA 36,0,60,6,62,102,62,0

757 **\$2F5** **NEWROW**
Řádková hodnota pro DRAWTO nebo XIO 18 (FILL).

758,759 **\$2F6,\$2F7** **NEWCOL**
Sloupcová hodnota pro DRAWTO nebo XIO 18 (FILL).

760 **\$2F8** **ROWINC**
Řádková dekrementační nebo inkrementační hodnota.

761 **\$2F9** **COLINC**
Sloupcová dekrementační nebo inkrementační hodnota. ROWINC a COLINC při grafických procedurách.

762 **\$2FA** **CHAR**
Obsahuje interní kód naposledy čteného nebo psaného znaku.
Registr není vhodné používat Basicem, většinou přečteme hodnotu 128 (viditelný kurzor) nebo 8 (neviditelný kurzor).

763 **\$2FB** **ATACHR**
Obsahuje naposledy čtený nebo zapisovaný ATASCII znak a používá se při převodu ATASCII kódu na interní. Příkazy DRAW a FILL (XIO) sem ukládají barvu právě kreslené linie. Pro lepší pochopení činnosti registru ATACHR zkuste následující program:

```
10 OPEN #2,4,0,"K:"  
20 GET #2,A  
30 PRINT PEEK(763);" ";CHR$(A)  
40 GOTO 20
```

764 **\$2FC** **CH**
Interní hardware hodnota naposledy zmáčknuté klávesy. POKE CH,255 registr čistí. Stejná hodnota se sem zapíše, pokud je stisk klávesy obsluhou klávesnice akceptován. Hodnota z CH se pak zároveň převede do CH1 (754; \$2F2). Jestliže je hodnota v CH stejná jako v CH1, stisk klávesy bude akceptován až po uplynutí dané časové prodlevy.

Všechny základní klávesové kódy využívají bity 0-5. Při stisku SHIFT se ještě nastavuje bit 6 na 1 a konečně při stisku CONTROL se nastavuje bit 7. Bity 6 a 7 se samozřejmě nastavují jen při stisku ještě další klávesy kromě BREAK nebo ESC, CAPS, RETURN a INVERS. Nejlépe bude, když si CH zase řádně sami otestujete.

```
10 LOOK=PEEK(764)
20 PRINT "STISKNUTA KLAVESA = ";LOOK
30 POKE 764,255
40 FOR LOOP=1 TO 250:NEXT LOOP
50 GOTO 10
```

765

\$2FD

FILDAT

Barva pro oblast vyplňovanou příkazem XIO 18 (FILL). Jak s FILL zacházet, ukazuje následující program:

```
10 GRAPHICS 31
20 POKE 708,4:POKE 709,50:POKE 710,144
30 POKE 765,1
40 COLOR 2
45 REM PLOT 80,80
50 PLOT 139,95;
60 DRAWTO 109,0
70 DRAWTO 49,0
80 POSITION 19,95
90 XIO 18,#6,0,0,"S:"
95 GOTO 95
100 PLOT 109,191
110 DRAWTO 139,96:REM COLOR 1
120 DRAWTO 19,96
130 POSITION 49,191
140 XIO 18,#6,0,0,"S:"
150 GOTO 150
160 COLOR 3
170 PLOT 159,191
180 DRAWTO 159,0
190 DRAWTO 0,0
200 POSITION 0,191
210 POKE 765,2
220 XIO 18,#6,0,0,"S:"
230 GOTO 230
240 PLOT 159,191:REM COLOR 2
250 DRAWTO 159,0
260 DRAWTO 110,0
270 COLOR 2:POSITION 140,95
280 XIO 18,#6,0,0,"S:"
290 PLOT 159,96
300 DRAWTO 140,96
310 POSITION 110,191
```

320 XIO 18,#6,0,0,"S:"
330 GOTO 330

10: Grafický mód 15 bez textového okénka.

20: Barvové hodnoty COLOR 1, 2 a 3.

30: Barvová hodnota pro FILDAT.

40: Barva pro následující PLOTy.

50 až 90: Funkce FILL potřebuje definovaný prostor: PLOTová linie jako horní hrana a PLOTová linie jako levé omezení. Tyto dvě linie určují tři rohové body: dole vpravo, nahoře vpravo, nahoře vlevo. Poslední rohový bod se vytvoří příkazem POSITION. FILL zásadně vyplňuje čtyřrohé plochy jinak libovolného tvaru. Levý horní roh však musí ležet výše než roh vpravo nahoře, neboť:

90: příkaz FILL provádí jen postupné PLOTování linie od levého horního rohu k bodu, udanému příkazem POSITION. Od každého bodu, vytvořeného PLOTem je přitom tažena linie až k pravému ohraničení, až je vyplněn celý určený prostor. Barva je určena ve FILDAT. K demonstraci efektu přidejte řádek 45 a zadejte RUN. Uvidíte, že až se dostane FILL do řádku 80, bude vyplněn jen sloupec 79, po 80 zůstane řádek ve starém stavu COLOR.

95: řádek spolu s řádkem 45 odstraňte.

100 až 140: šestiúhelník, který se má umístit na obrazovce, musí být rozdělen na dva lichoběžníky.

110: horní hrana dolní poloviny šestiúhelníku je v barvě, která platí pro PLOT. Chcete-li pokračovat, odstraňte REM v řádku 110.

150: po zjištění účinku na obrazovce, odstraňte i tento řádek.

160 až 220: nebude vyplněna celá zbývající plocha obrazovky, nýbrž jen plocha od levého okraje po šestiúhelník.

230: odstraňte další záchytný bod.

240 až 280: aby se vyplnila pomocí FILL také zbývající plocha vedle šestiúhelníku, je zapotřebí trošku více námahy. Není totiž možné vyplnit tuto plochu jedním příkazem. Proto tyto řádky vyplní horní polovinu a o zbytek se postarají řádky 290 až 320.

766

\$2FE

DSPFLG

Návěst výpisu na obrazovku. Jestliže je zde nula (prvotní nastavení), jsou vykonávány všechny běžné řídicí funkce zadávané přes klávesnici (ATASCII kódy 27-31, 123-127, 251,255) jako např. CTRL-INSERT, CTRL-CLEAR, TAB. Po vložení libovolné nenulové hodnoty do DSPFLG se místo klávesové funkce vytiskne na obrazovce příslušný znak, stejně jako po ESC.

767

\$2FF

SSFLG

Start/stop návěst výpisu na obrazovku. Výpis běží normálně, je-li zde nula. Při 255 (\$FF) se zastaví všechny příkazy PRINT, LIST a DRAWTO. SSFLG lze ovlivnit přes CTRL-1 nebo POKE. Po zapnutí počítače a po RESET je zde nula.

TŘETÍ STRÁNKA

Adresy 768 až 831 (\$300 až \$33F) jsou využity pro obsluhu periferních zařízení (S:, P:, E:, D:, C:, R: a K:).

Adresy 768 až 779 (\$300 až \$30B) jsou používány V/V procedurami nevyjímaje diskety. Disketa ATARI vyměňuje data s počítačem rychlostí 19200 baudů. Má vlastní mikroprocesor 6507, RAM 128 byte, ROM 2716 (2KB), obvod 2332 (RAM s dalšími 128 byte + časovač) a řídicí obvod WD 1771 FD.

768

\$300

DDEVIC

Sériová přístrojová sběrnice. Nastavení je provedeno obslužnými procedurami a není možné jej změnit uživatelem.

disketa	D1-D4	49-52	(\$31-\$34)
tiskárna	P1	64	(\$40)
tiskárna	P2	79	(\$4F)
RS-232	R1-R4	80-83	(\$50-\$53)

769

\$301

DUNIT

Číslo přístroje, může být určeno uživatelem (1-4).

770

\$302

DCOMND

Číslo, značící operaci (příkaz), který se má vykonat.

Nastavitelné uživatelem nebo obslužnými procedurami. Příkazy pro sériovou sběrnici jsou:

read	82	(\$52)
write (verify)	87	(\$57)
status	83	(\$53)
put (no verify)	80	(\$50)
format	33	(\$21)
download	32	(\$20)
read address	84	(\$54)
read spin	81	(\$51)
motor on	85	(\$55)
verify sector	86	(\$56)

Vše jsou diskové povely, vyjma write a status, které používá také tiskárna.

771

\$303

DSTATS

Status kód. Používá se vždy obslužnými procedurami k indikaci SIO jak pokračovat po vyslání nebo příjmu dat. Bit 6 se nastaví od SIO na 1 při

příjmu dat, bit 7 při vysílání dat. Jestliže jsou oba bity na nule, data nejsou zpracovávána. Chybné je nastavení obou bitů na 1.

772,773 **\$304,\$305** **DBUFLO/HI**

Je zde uložena adresa datového bufferu pro zápis nebo příjem dat nebo informace o přístrojovém statusu nebo diskový datový sektor (rozumí se opět adresa).

774 **\$306** **DTIMLO**

Čekací doba do chybového hlášení. Inicializováno na 30.

775 **\$307** **DUNUSE**

Nepoužit.

776,777 **\$308,\$309** **DBYTLO/HI**

Počet byte, které mají být přijaty nebo vyslány z datového bufferu (nebo z disketové jednotky) nebo počet špatných datových sektorů. Do SID se vloudila malá chyba, která zapříčiňuje nesprávnou funkci, končí-li adresa posledního byte bufferu \$FF (např. \$A0FF).

778,779 **\$30A,\$30B** **DAUX1/2**

Použito pro pomocnou specifikaci při disketových operacích. Může se zde objevit pět různých příkazů:

GET sector	(82; \$52)
PUT sector	(80; \$50)
PUT (verify)	(87; \$57)
STATUS	(83; \$53)
FORMAT	(33; \$21)

Adresy 788 až 793 (\$30C až \$319) jsou využity k nejrozmanitějším účelům.

Adresy 794 až 831 (\$31A až \$33F) jsou adresní obslužné tabulky.

780,781 **\$30C,\$30D** **TIMER1**

Inicializační hodnota časovače rychlosti v baudech.

782 **\$30E** **ADDCOR**

Korekční návěst pro časovací registry rychlosti.

783 **\$30F** **CASFLG**

Operace s magnetofonem zde nastaví nenulovou hodnotu. Jinak probíhají standardní SIO operace.

784,785 **\$310,\$311** **TIMER2**

Intervalový časovač. Časovače 1 a 2 obsahují referenční časy počátku a konce bitového přenosu krátké zkušební sekvence. První byte každého časovače obsahuje hodnotu VCOUHT (54283; \$D40B), druhý byte pak reálný čas převzatý z RTCLK 20 (\$14). Rozdílu časovačů se využívá k nalezení opravné hodnoty v tabulce, pomocí které se upraví příjmová rychlost v CBAUDL/H (750,751; \$2EE,\$2EF).

786 **\$312** **TEMP1**

Registr dočasné úschovy používaný SIO při výpočtu opravných rychlostí magnetofonu. Viz 54283 (\$D40B).

787 **\$313** **TEMP2**

Registr dočasné úschovy.

788 **\$314** **PTIMOT**

Přesun ze 28 (\$1C) u starších 400/800.

789 **\$315** **TEMP3**

Registr dočasné úschovy.

790 **\$316** **SAVIO**

?

791 **\$317** **TIMFLG**

Návěst překročení času pro korekci časovače magnetofonu (rychlost příjmu). Je inicializována na 1 a dekrementována během V/V operace. Jestliže je zde nula (po dvou sekundách) před prvním čtením magnetofonového zápisu, operace se nezdaří.

792**\$318****STACKP**

Ukazatel zásobníku SIO. Ukazuje na byte, který bude v následující běžné operaci použit (první stránka paměti).

783**\$318****TSTAT**

Dočasný registr SIO. Viz STATUS (48; \$30).

794-829**\$31A-\$33D****HATABS**

Obslužná adresní tabulka. Po zapnutí počítače je tabulka přepsaná z ROM. První byte z každé trojice obsahuje název přístroje v ATASCII, byte dva a tři adresu startu obslužné procedury.

Nepoužité byte jsou vynulované. Po zapnutí počítače nebo po RESET je tabulka přepsána z ROM.

V HATABS je definováno pouze prvních pět V/V procedur:

794	\$31A	tiskárna	(P:), inicializace na 58416 (\$E430)
797	\$310	magnetofon	(C:), inicializace na 58432 (\$E440)
800	\$320	editor	(E:), inicializace na 58368 (\$E400)
803	\$323	obrazovka	(S:), inicializace na 58384 (\$E410)
806	\$326	klávesnice	(K:), inicializace na 58400 (\$E420)

Další bloky mohou být definovány uživatelem. FMS pátrá v HATABS shora dolů zda je definováno D: a pokud ne, zapíše adresu své obslužné procedury do prvního volného bloku.

Formát HATABS je následující:

Jméno přístroje

Adresa počátku obslužného programu

Další jména

Další adresy

.
. .
.

Každý obslužný program má následující formát:

OPEN vektor

GET BYTE vektor

PUT BYTE vektor

GET STATUS vektor

SPECIAL vektor

CIO v tabulce pátrá zdola nahoru, což dovoluje zapsat do neobsazeného prostoru případné adresy vlastních V/V procedur, budou nalezeny dříve než originální.

829-831**\$33D-\$33F****PUPBT1-3**

Registry 1-3 použité ke kontrole neporušenosti paměti. OS inicializuje po zapnutí počítače registry na 92 (\$5C), 147 (\$93) a 37 (\$25). Po RESET jsou hodnoty v PUPBT1-3 srovnány s inicializační hodnotou a pokud srovnání nedopadne dobře, provede se inicializace jako po zapnutí počítače.

Adresy 832 až 959 (\$340 až \$3BF) jsou rezervovány pro osm IOCB (vstupně/výstupní řídicí blok). IOCB jsou vlastně kanály skrze které se děje výměna informací mezi počítačem a periferním zařízením. Pro každý blok je rezervováno 16 byte. Po každém tisku na obrazovku nebo na tiskárnu, po každém LOAD či SAVE je využit IOCB. Tyto příkazy pracují s IOCB automaticky, bez toho, že by ještě něco vyžadovaly (např. OPEN nebo CLOSE).

Některé IOCB plní specifické funkce, IOCB0 využívá obrazovka. Další mohou být využity k jakékoliv další V/V funkci. IOCB je používán ve stejnou dobu jako blok v nulové stránce paměti 32 až 47 (\$20 až \$2F). OS odsud bere informace a přesouvá je do ZIOCB pro další použití CIO. Nezapomeňte, že vykonáním příkazu DOS se uzavřou všechny IOCB kromě IOCB0.

832-847**\$340-\$34F****IOCB0**

IOCB nula. Normálně je používán obrazovým editorem (E:). Zkuste udělat následující program:

```
10 DIM A$(1):CONSOL=53279:GRAPHICS 0:IOCB0E=838
20 PHDLR=58422
30 EHDLR=58374
40 PL=PEEK(PHDLR):PH=PEEK(PHDLR+1)
50 EL=PEEK(EHDLR):EH=PEEK(EHDLR+1)
60 PRINT "Text bude prubezne tisten."
70 PRINT "SELECT slouzi k vyberu tisku"
80 PRINT "na obrazovku nebo tiskarnu."?
90 PRINT "Aktivuj tiskarnu a stiskni RETURN"
100 INPUT A$
110 PRINT I;" SELECT slouzi k vyberu tisku.":I=I+1
120 IF PEEK(CONSOL)<>5 THEN 110
130 IF DIR THEN POKE IOCB0E,EL:POKE IOCB0E+1,EH
140 IF NOT DIR THEN POKE IOCB0E,PL:POKE IOCB0E+1,PH
150 DIR=NOT DIR
160 IF PEEK(CONSOL)<>7 THEN 160
170 GOTO 110
```

Pokud nemáte tiskárnu, krátké vysvětlení: vhodnou změnou adresy v ICPTL/H je tisk prováděn na obrazovku nebo tiskárnu, vždy však příkazem PRINT.

Každý IOCB je rozčleněn následovně:

NÁVĚŠT	OFFSET	BYTE	POPIS
ICHID	0	1	Index zařízení, s kterým bude kanál komunikovat. Je nastaven OS. Jestliže se nepoužívá, je zde 255 (\$FF), což je také inicializační hodnota.
ICDNO	1	1	Číslo zařízení (nejčastěji diskety). Např. 1 pro D1:. Rovněž nastavován OS.
ICCOM	2	1	Příkaz, který určuje, co se bude s daty a vybranou periferií dít. Je to první proměnná po čísle kanálu příkazu OPEN.
ICSTA	3	1	Hlášení o proběhlé V/V operaci.
ICBAL/H	4,5	2	Adresa bufferu odkud nebo kam se ukládají data.
ICPTL/H	6,7	2	Adresa podprogramu minus jedna, vysílajícího jeden byte. Používá OS i Basic. Po zapnutí počítače ukazuje adresa na "IOCB NOT OPEN".
ICBLL/H	8,9	2	Délka bufferu nastavená na maximální počet byte při přenosu PUT nebo GET. Dekrementuje se po každém čtení nebo zápisu.
ICAX1	10	1	Pomocný byte jedna. Použit při OPEN ke specifikaci čtení nebo zápisu. Čtyři je READ, osm je WRITE, dvanáct READ i WRITE. Všechna zařízení nemohou obě operace používat. Jednotlivé bity v registru mají svůj význam: BIT: 7 6 5 4 3 2 1 0 POUŽITÍ: nepoužity W R D A W značí zápis, R čtení, D adresář, A připojení.
ICAX2	11	1	Pomocný byte dva. Specifické použití každým zařízením. Pomocný byte dva až pět nemají fixované použití.
ICAX3/4	12,13	2	Pomocný byte tři a čtyři. Užívá se k přenesení čísla disketového sektoru při příkazech NOTE a POINT.
ICAX5	14	1	Pomocný byte pět. Používá se při NOTE a POINT k provedení zápisu do sektoru. Zapisuje se odstup v sektoru od nula do 124 (\$7C), a byte 127 (\$7F) je použit jako čítač aktuálních datových byte v daném sektoru.
ICAX6	15	1	Nepoužitý pomocný byte.

Následující údaje se týkají prvního parametru po čísle kanálu příkazu OPEN a přináležejí k ICAX1.

ZAŘÍZENÍ	ICAX1	POPIS
magnetofon	4	čtení
magnetofon	8	zápis
disketa	4	čtení
disketa	6	čtení z adresáře
disketa	8	zápis nového souboru. Každý otevřený soubor bude v tomto módu vyřazen a první byte bude zapsán na počátek souboru.
disketa	9	zápis-přidání. V tomto módu zůstane otevřený soubor nedotčený a data budou zapsána na konec souboru.
disketa	12	čtení i zápis. Čtení i zápis dat bude od počátku souboru.
obrazovka	8	obrazový výstup
editor	12	klávesový vstup a obrazový výstup.
editor	13	obrazový vstup a výstup
klávesnice	4	čtení
tiskárna	8	zápis
RS-232	5	souběžné čtení
RS-232	8	zápis bloku
RS-232	9	souběžný zápis
RS-232	13	souběžný zápis i čtení

Mazání obrazu po GR..C

Rovněž textové okénko T

Operace čtení..... R

Obrazovka	8	ano	ne	ne
(S:)	12	ano	ne	ano
	24	ano	ano	ne
	28	ano	ano	ano
	48	ne	ne	ne
	44	ne	ne	ano
	56	ne	ano	ne
	68	ne	ano	ano

Všechny hodnoty (S:) jsou po provedení OPEN přesunuty do ICAX1. Druhý parametr v příkazu OPEN (přesunut vždy do ICAX2) má jen omezené použití. Většinou je nastaven na nulu. Jestliže se zde nachází 128 (\$80), pak je mezi bloky magnetofonového zápisu pauza krátká. Pro obrazovku (S:) může být druhý parametr použit ke specifikaci GRAPHICS módu nula až devět. Jestliže je zvolen mód nula, pak AUX1 je ignorován. Pro ICCOM přináležejí následující hodnoty (příkaz XIO používá stejné hodnoty):

PŘÍKAZ	HODNOTA	
OPEN - otevření kanálu	3	
GET - čtení bloku	5	Basic: INPUT#n,A
GET - čtení znaku	7	Basic: GET#n,A
PUT - zápis bloku	9	Basic: PRINT#n,A
PUT - zápis znaku	11	Basic: PUT#n,A
CLOSE - uzavření kanálu	12	
STATUS - stavové slovo	13	

Následující příkazy platí jen pro disketu:

RENAME	32
ERASE	33
PROTECT	35
UNPROTECT	36
POINT	37
NOTE	38
FORMAT	254

XIO používá:

GET - čtení znaku	7
PUT - zápis znaku	11
DRAW - kreslení linie	17
FILL - kreslení plochy	18

FILL má v Basic tvar XIO 18,#6,12,0,"S:"

XIO RS-232 (R:) používá:

Výstup bloku	32
Řízení RTS, XMT, DTR	34
Stop bity, velikost slova	36
Přenosový mód	38
Souběžný mód	40

Adresní pole IOCB:

NÁVĚST	IOCB0	IOCB1	IOCB2	IOCB3	IOCB4	IOCB5	IOCB6	IOCB7	POUŽITÍ
ICHID	832	848	864	880	896	912	928	944	index
1CDNO	833	849	865	881	897	913	929	945	přístr
ICCOM	834	850	866	882	898	914	930	946	příkaz
ICSTA	835	851	867	883	899	915	931	947	status
ICBAL/H	836	852	868	884	900	916	932	948	buffer
ICPTL/H	838	854	870	886	902	918	934	950	put buf
ICBLL/H	840	856	872	888	904	920	936	952	délka
ICAX1	842	858	874	890	906	922	938	954	práce
ICAX2	843	859	875	891	907	923	939	955	pomocný
ICAX3	844	860	876	892	908	924	340	956	sektor
ICAX4	845	861	877	893	909	925	341	957	sektor
ICAX5	846	862	878	894	910	926	942	958	byte#
ICAX6	847	863	879	895	911	927	943	959	pomocný

960-999**\$3C0-\$3E7****PRNBUF**

Buffer tiskárny. Do tohoto bufferu se po LPRINT ukládají data a z něj pak vysílají nejednou na tiskárnu po ukončeném řádku nebo po naplnění všech 40 byte bufferu. Jestliže LPRINT generuje více než 40 znaků a končí středníkem nebo více než 38 znaků a končí čárkou, počítač celý buffer vyšle. Po dalším LPRINT bude tisk pokračovat v 41. sloupci tiskárny.

Malý program umožní přepsat texty na obrazovce na tiskárnu:

```
10 DIM TEXT$(1000)
20 OPEN#2,4,0,"S:"
30 TRAP 1050
.
.
1000 FOR LINE=1 TO 24
1005 POSITION PEEK(82),LINE
1010 FOR COL=1 TO 38
1015 GET#2,CHAR:TEXT$(COL,COL)=CHR$(CHAR)
1020 NEXT COL
1025 GET#2,COL
1030 LPRINT TEXT$
1040 NEXT LINE
1050 RETURN
```

V registru PTABW (201; \$C9) lze nastavit počet mezer mezi čárkou a dalším tiskem. Minimální počet akceptovaných mezer jsou 2. LPRINT používá IOCB7, který si automaticky otevírá i zavírá. Není tedy potřeba používat OPEN a CLOSE.

1000**\$3E8****SUPERF**

Registr editoru. Tištěn při vstupu do procedury "PUT byte". Editor mění kódy kláves 142-145 na 28-31 a nastavuje SUPERF na nenulovou hodnotu.

1001**\$3E9****CKEY**

Je-li po zapnutí počítače stisknuto tlačítko START, CKEY se nastaví a do paměti je možné přenést z magnetofonu strojové programy.

1002**\$3EA****CASSBT**

Návěst pro přenos programu z magnetofonu do počítače. Po zapnutí počítače zjišťuje OS zda má data přenášet z magnetofonu nebo z diskety. Pokud přenos z magnetofonu není požadován, nastaví se zde nula.

1003 **\$3EB** **CARTCK**

Kontrolní součet první stránky paměti zásuvného modulu. Jestliže není kontrolní suma správná, dojde k normálnímu startu počítače, jako po jeho zapnutí. Platí pouze pro 1200XL.

1804 **\$3EC** **DERRF**

Chybová návěšt neúspěšného pokusu o inicializaci obrazového editoru.

1005-1015 **\$3ED-\$3F7** **ACMVAR**

Rezervováno pro proměnné OS. Po zapnutí počítače jsou všechny proměnné mezi 1005 a 1023 (\$3ED a \$3FF) vynulovány. Po RESET zůstávají zachovány.

1316 **\$3F8** **BASICF**

Návěšt ROM Basic. Jakákoliv nenulová hodnota vyřadí po RESET ROM Basic z činnosti. Nahraje se DOS a pokus o návrat do Basicu nebude úspěšný.

1017 **\$3F9** **MINTLK**

Nepoužit.

1018 **\$3FA** **GINTLK**

Blokovací registr zásuvného modulu. Je to vlastně komplement k BASICF, jedna signalizuje přítomnost zásuvného modulu, nula naopak značí aktivní ROM Basic. Při inicializaci se zde zapisuje hodnota z TRIG3 (53267; \$D103). Jestliže dojde během provozu k zasunutí modulu, systém se zhroutí.

1019, 1020 **\$3FB, \$3FC** **CHLINK**

Buffer magnetofonu. Tato část paměti je používána obsluhou magnetofonu při čtení nebo zápisu dat z nebo do magnetofonu. 128 datových byte pro každou část zápisu je uloženo od adresy 1024 (\$400 - čtvrtá stránka). Běžnou velikost bufferu lze nalézt v registru BLIM (650; \$28A). Registr BPTR (61; \$3D) udává, který byte bude v bufferu čten nebo zapsán. CASBUF je rovněž využit při prvním čtení z diskety. Každá část magnetofonového zápisu sestává ze 132 byte; dva kontrolní byte s hodnotou 85 (\$55 - střídání jedniček a nul) pro výpočet korekce rychlosti příjmu dat z magnetofonu, jeden řídicí byte (viz níže), 128 datových byte a jeden byte kontrolního součtu. V CASBUF jsou zapsány pouze datové byte.

Hodnoty řídicích byte:

250 (\$FA) Částečný blok. Aktuální hodnota byte se nachází v posledním byte zápisu (127).

252 (\$FC) Zápis dokončen, následuje již jen 128 byte.

254 (\$FE) Poslední blok celého zápisu sestává pouze ze samých nul.

1152-1279 **\$480-\$4FF** -----

128 byte pro volné použití.

Operace v pohyblivé řádové čárce (floating point) využívá adresní prostor 1406-1535 (\$57E-\$5FF).

1406 **\$57E** **LBPR1**

LBUFF prefix jedna (LBUFF - řádkový buffer Basic).

1407 **\$57F** **LBPR2**

LBUFF prefix dva.

1408-1335 **\$580-\$5FF** **LBUFF**

Řádkový buffer Basic o délce 128 byte. Slouží k ukládání výsledku matematických operací prováděných v Basicu. Adresa vstupního bufferu je uložena v registru CIX (243,244; \$F3,\$F4).

1504 **\$5E0** **PLYARG**

Polynomický argument.

1510-1515 **\$5E6-\$5EB** **FPSCR**

Registry pohyblivé řádové čárky.

1516-1535

\$5EC-\$5FF

FPSCR1

Registry pohyblivé řadové čárky.

1536-1791

\$600-\$6FF

Celá šestá stránka paměti (256 byte) je volná pro případné strojové programy nebo jiné použití.

Na tomto místě neuškodí krátké zopakování znalostí o funkci USR, používané při skoku do strojových programů z Basicu.

Jedna metoda umožňuje uložit program do chráněného místa paměti (šestá stránka je ideální) a pak stačí provést např.:

U=USR(1536)

Další možnost poskytují řetězce, do nichž se strojový program uloží čtením dat ve smyčce FOR-NEXT a skok do programu se provede pomocí další funkce ADR. U=USR(ADR(PGM\$)). V takovém programu není vhodné používat instrukce JMP a CALL na vlastní program, neboť absolutní adresa počátku programu není předem obvykle známa.

Kratší strojové programy mohou být vloženy přímo do funkce USR tak, že ATASCII znaky v závorce jsou ekvivalentní kódům strojových instrukcí:

U=USR(a S1+?/k...). Na rozdíl od předchozích dvou způsobů je délka tohoto programu omezena maximální délkou logického řádku. Absolutní adresa počátku programu není většinou rovněž předem známa, proto i v tomto případě nelze použít instrukce skoků JMP a CALL na vlastní program.

Ve všech třech případech by měla být první instrukce strojového programu PLA.

1792-5377

\$700-\$1501

Správa souboru FMS (file management system) umožňuje styk mezi Basicem nebo DUP a disketou.

5440-13062

\$1540-\$3306

Prostor pro DUP.SYS (DOS programy). Vrchol prostoru bude kolísat v závislosti na počtu využívaných sektorových bufferů.

6780-7547

\$1A7C-\$1D7B

Disketové & sektorové buffery. Množství použité paměti bude záviset na počtu přidělených bufferů.

7548-MEMLO \$1D7C-\$3306_(maximum)

Nechráněná část DUP.SYS DOS programu, zvláště při práci s Basicem.

Paměťový prostor 1792-2047 (\$700-\$7FF sedmá stránka) využívá DOS nebo pokud není DOS přítomen lze paměť využít pro Basic či strojové programy.

Nejnižší adresa volné paměti je 1792. Basic sedmou stránku využívá jako bufferu při překladu výrokových řádků.

DOS 2.0

1792 \$700 BFLAG

Přenosová návěst nastavena na nulu (nepoužívá se).

1793 \$701 BRCNT

Číslo za sebou jdoucích čtených sektorů. Jestliže je čteným souborem DOS, BRCNT obsahuje jedničku.

1794,1795 \$702,\$703 BLDADR

Adresa počátku ukládání sektoru.

1796,1797 \$704,\$705 BIWTARR

Inicializační adresa.

1788 \$706 JMPXBCONT

Průběžná přenosová adresa (následuje). Kód instrukce JMP.

1799,1800 \$707,\$708

Průběžná přenosová adresa. Viz výše JMPXBCONT.

1801 \$708 SABYTE

Maximální počet souběžně otevřených souborů. Běžně je nastaven na tři, maximum je sedm (jeden IOCB pro každý soubor, IOCB0 používá obsluha obrazovky). Každý otevřený soubor potřebuje 128 byte pro buffer, tzn., že čím více bude souborů otevřených, tím méně bude volné paměti RAM. SABYTE lze modifikovat a změnu lze trvale uchovat funkcí "H".

1802

\$70A

DRVBYT

Maximální počet disketových jednotek zapojených do systému.

Základní nastavení je 2. Jestliže budou v systému použity diskety 1, 3 a 4, je potřeba DRVBYT změnit:

00001101 nebo 13 dekadicky.

Číslo diskety odpovídá nastavený bit v registru.

Každá disketa má v paměti RAM samostatný 128 byte dlouhý buffer.

DRVBYT lze modifikovat a změnu je možno trvale uchovat funkcí "H" DOSu.

1803

\$70B

Nepoužit.

1804,1805

\$70C,70D

SASA

Buffer přidělení počátečních adres. Ukazuje na 1995 (\$7CB) pokud je čten DOS.

1806

\$70E

DSFLG

Návěst DOS. Pro druhou fázi přenosu musí mít nenulovou hodnotu.

Indikuje, že soubor DOS.SYS bude zapsán na disk.

0 - nepřítomnost DOS souboru.

1 - sektor dlouhý 128 byte.

2 - sektor dlouhý 256 byte.

1807,1808

\$70F,\$710

DFINK

Adresa prvního sektoru souboru DOS.SYS.

1809

\$711

BLDISP

Přemístění do sektor link byte. Sektor link byte ukazuje, který další diskový sektor bude čten. Jestliže je zde nula, bude dosaženo konce souboru.

1810,1811 **\$712,\$713** **DFLADR**

Adresa počátku souboru DOS.SYS.

Data z přenosového sektoru jsou umístována na adresách 1792-1916 (\$700-\$77C). Data části souboru DOS.SYS začínají od 1917 (\$77D).

Všechny binární soubory mají první byte 255. Další čtyři byte obsahují počáteční a koncovou ukládací adresu.

1900 **\$76C** **BSIO**

Vstupní adresa FMS diskového sektoru s V/V procedurami.

1906 **\$772** **BSIOR**

Vstupní adresa FMS diskové obsluhy (?).

1913 **\$779** **-----**

Zapísovací kontrolní návěst V/V diskových operací. POKE 1913,80 kontrolu z činnosti vyřadí, 87 kontrolu opět aktivuje. Vyřazení kontroly zápisu na disk sice urychlí, ale jediná chyba v kritickém místě programu může způsobit jeho totální zhroutení. Proto raději opatrně s POKE 1913,80.

1995 **\$7CB** **DFMSDH**

Vstupní adresa 21 byte dlouhého programu obsluhy diskety. Adresa tohoto programu je umístěna v HATABS (794 až 829; \$31A až \$33D) inicializační FMS procedurou.

2016 **\$7E0** **DINT**

FMS inicializační procedura začíná od 1995 (\$7CB). DUP volá FMS touto adresou. K-DOS používá k inicializaci stejnou adresu.

2219 **\$8AB** **DFMOPN**

OPEN procedury obsahující otevření souboru pro připojení, mazání a výstup.

2508 **\$900** **DFMPUT**

PUT byte procedury.

2591 **\$A1F** **WTBUR**

Burstovací V/V procedury.

2751	\$ABF	DFMGET
GET byte procedury, zahrnující i procedury GET souboru.		
2817	\$B01	DFMSTA
STATUS procedury.		
2837	\$B15	DFMCLS
IOCB CLOSE procedury.		
2983	\$BA7	DFMDDC
Počátky podprogramů, zahrnující Basic XIO příkazy.		
3033	\$BD9	XRENAME
Soubor RENAME (přejmenování souboru).		
3122	\$C32	XDELETE
Soubor DELETE (smazání souboru).		
3196	\$C7C	XLOCK
Soubor LOCK (ochrana souboru před přepisem).		
3203	\$C83	XUNLOCK
Soubor UNLOCK (zrušení ochrany souboru před přepisem).		
3258	\$CBA	XPOINT
Basic příkaz POINT.		
3331	\$D03	XNOTE
Basic příkaz NOTE.		

3352 **\$D18** **XFORMAT**

Formátování celého disku.

3501 **\$DAD** **LISTDIR**

List diskovým adresářem.

3742 **\$E9E** **FNDCODE**

Dekódování názvu souboru, obsahuje test platnosti speciálních symbolů v názvu souboru. Na běžné jméno souboru ukazuje registr ZBUF (67,68; \$43,\$44).

3783 **\$EC7** **-----**

Možnost změny speciálního znaku * (ATASCII 42) používaný v DOSu k označení nespecifikovaného souboru. Modifikovaný DOS pak je možno trvale uchovat funkci "H".

3818, 3822 **\$EEA, \$EEE** **-----**

POKE 3818,33 a POKE 3822,123 umožní DOSu akceptovat interpunkční znaménka a čísla v názvech souborů. 33 je spodní hranice a 127 horní hranice běžně využitelného ATASCII kódu (nižší nebo vyšší hodnota jsou řídicí a grafické kódy a inverzní znaky). Změna může být zapsána zpět na disk pomocí "H" v DOS volbě.

3850 **\$F0A** **FDSCHAR**

Zápis znaku jména souboru, který byl předtím testován dekódovací procedurou.

3873 **\$F21** **SFDIR**

Procedura prohlížení adresáře. Hledání specifického jména souboru.

3988 **\$F94** **WRTNXS**

Procedura zápisu dat do sektoru.

4111 **\$100F** **RDNXTS**

Procedura čtení dat ze sektoru.

4206 **\$106E** **FSDDIR**

Procedura čtení i zápisu v adresáři.

4235 **\$108B** **RDVTOC**

Čtení nebo zápis obsahu tabulky obsahu (VTOC) sektoru.

4293 **\$10C5** **FRESECT**

Procedura pro sumarizaci volných sektorů. Dává počet volných sektorů na disku.

4358 **\$1106** **GETSECTOR**

Procedura pro nalezení volného sektoru na disku.

4452 **\$1164** **SETUP**

Počáteční inicializace FMS parametru.

4618 **\$120A** **WRTDOS**

Procedura zápisu nového DOS.SYS souboru.

4789 **\$12B5** **ERRNO**

Začátek tabulky s FMS chybovými hodnotami.

4856-4978 **\$12F8-\$1372** **-----**

Prostor pro nejrůznější zápisy (délka sektoru, úroveň zásobníku, číslo souboru atd.).

4993 **\$1381** **FBC**

Start FMS FCB (1-8) (FCB - soubor řídicích bloků). Osmice FCB jsou 16byte bloky, které korespondují s IOCB. Každý FCB sestává z:

NÁVĚST	BYTE	POPIS
FCBFNO	1	číslo souboru, který bude zpracován.
FCBOTC	1	číslo módu: 1 je připojení, 2 čtení adresáře, 3 vstup, 8 výstup a 20 vymazání
SPARE	1	nepoužit
FCBSLT	1	návěst pro velikost sektoru 128 nebo 256 byte.
FCBFLG	1	pracovní návěst. Jestliže je zde 128, bude soubor otevřen pro výstup nebo připojení a může přitom

		zabrat nové datové sektory. Při 64 bude buffer zapsán na disk.
FCBMLN	1	maximální délka sektorových dat závisující na způsobu formátování disku, 125 nebo 253 byte. Poslední tři byte sektoru jsou rezervovány pro sektor link byte a kontrolní byte.
FCBDLN	1	byte čtený nebo modifikovaný v sektoru.
FCBBUF	1	sděluje FMS, který buffer bude zařazen do souboru.
FCBCSN	2	číslo sektoru, který je právě v bufferu.
FCBLSN	2	číslo dalšího sektoru datového řetězce.
FCBSSN	2	počáteční sektor pro připojování dat, jestliže bude soubor otevřen pro připojení.
FCBCNT	2	sektorový čítač běžných souborů.

5121 **\$1401** **FILDIR**

256 byte dlouhý buffer pro úschovu adresáře disku.

5377 **\$1501** **ENDFMS**

Buffer diskového adresáře 64 byte dlouhý, jeden byte pro každý možný soubor. Rovněž konec souboru FMS.

5440 **\$1540** **MINIDUP**

Inicializační DUP.SYS adresa. Počátek mini-DOSu, část chráněného DUP v RAM. Používá se ke stejným účelům jako K-DOS.

5446-5450 **\$1546-\$154A** **-----**

Obsahuje adresu registru DOSVEC (10). Používá se při volání DOSu v Basicu.

5533 **\$159D** **DUPFLG**

Návěst testu DUP souboru. 0 znamená nepřítomnost DUP.

5534 **\$159E** **OPT**

Použit k zápisu volby DOS funkce.

5535	\$159F	LOADFLG
Jestliže je zde 128, soubor MEM.SAV nemá být čten.		
5540	\$15A4	SFLOAD
Procedura souboru MEM.SAV.		
5899	\$170B	MEMLDD
Jestliže je zde nulová hodnota, soubor MEM.SAV nemá být čten.		
5947	\$173B	-----
Začátek MEM.SAV procedur.		
6044, 6045	\$179C, \$179D	INISAV
Vektor DOSINI (12,13; \$C,\$D). Vstup do DOSu z Basicu.		
6046	\$179E	MEMFLG
Návěst, oznamující zda paměť bude nahrána na disk s použitím souboru MEM.SAV.		
6418	\$1912	CLMJMP
Test, zda musí být před skokem do Basicu použit soubor MEM.SAV.		
6457	\$1939	LDMEM
MEMSAVE procedura (pro MEM.SAV soubor).		
6518	\$1979	INITIO
Počátek DOS.SYS po RESET.		
6630	\$19E6	ISRODN
Počátek přerušovacích procedur sériového výdeje dat.		

6691 **\$1A23** **ISRSIR**

Počátek přerušovacích procedur sériového příjmu dat.

6781 **\$1A7D** **-----**

Počátek disketových a datových bufferů. Disketové buffery jsou číslovány postupně od 1 do 4, datové od 1 do 8. Normálně jsou první dva buffery přiděleny disketám a další tři jsou pro data. Buffery mají délku 128 byte a začínají na 6908 (\$1AFC), 7037 (\$1B7C), 7162 (\$1BFA), 7292 (\$1C7C). Viz 1801 (\$709) a 1802 (\$70A).

7420 **\$1CFC** **-----**

Registr MEMLO (743,744; \$2F7,\$2F8) zde ukazuje, pokud je v paměti chráněná část DOSu. V MEMLO bude 7164 při jedné disketě a dvou datových bufferech (-256 byte).

7548 **\$1D7C** **-----**

Začátek nechráněné části DUP, 40 byte buffer parametrů.

7588 **\$1DA4** **LINE**

80 byte řádkový buffer.

7668 **\$1DF4** **DBUF**

256 byte datový buffer pro procedury COPY.

7924 **\$1EF4** **-----**

Datové buffery a různá další použití.

7951-8278 **\$1F0F-\$2056** **DMENU**

Paměť obrazovky pro DOS.

PROCEDUREY DUP.SYS

Paměť od 8192 do 32767 (\$2000 až \$7FFF) je vhodná pro jakékoliv vlastní použití. DUP.SYS a DOS.SYS rovněž část této paměti využívá (max. do 13062; \$3306).

8309	\$2075	DOSOS
Start DUP.SYS obsahující volání, pokud je funkce výběru ze souboru dokončena a objeví se nápis "SELECT ITEM".		
8505	\$2139	DIRLST
List adresářem.		
8649	\$21C9	DELFIL
Smazání souboru.		
8990	\$231E	-----
Kopírování souboru. Kopírovací procedury jsou od adresy 9080 (\$2378) PYFIL.		
9783	\$2637	RENFIL
Přejmenování souboru.		
9856	\$2680	FMTDSK
Formátování celého disku. Při verzi "C" ROM není žádná možnost formátovat specifický sektor. Umožňuje to až nová verze "E" ROM, která nejen umožňuje výběrové formátování sektoru, ale je i nepoměrně rychlejší.		
3966	\$26EE	STCAR
Start zásuvného modulu nebo Basic.		
10060	\$274C	BRUN
Spuštění binárního souboru (binary file).		
10111	\$277F	-----
Počátek zápisu souboru MEM.SAV.		

10201	\$27D9	WBOOT
Zápis DOS/DUP souboru na disk.		
10483	\$28F3	TESTVER2
Test pro verzi DOS 2.0.		
10522	\$291A	LDFIL
Přenesení binárního souboru do paměti RAM. Jestliže je specifikována startovací adresa, program se sám spustí.		
10608	\$2370	LKFIL
Ochrana souboru na disku.		
10690	\$29C2	DDMG
Kopírování disku.		
11528	\$2D08	DFFM
Kopírování souboru.		
11841	\$2E41	-----
Různé procedury.		
12078	\$2F2E	SAVFIL
Zápis binárního souboru na disk.		
12384	\$303C	-----
Různé podprogramy.		
13062	\$3306	-----
Konec DUP.SYS.		

Verze DOS 2.5 a počítač 130XE

Verze DOS 2.5 obsahuje speciální program nazývaný RAMDISK.SYS, který ale spolupracuje jen s počítačem 130XE, u kterého využívá přidavné 64 KB paměti. RAMdisk pracuje stejně jako reálná disketa, jen podstatně rychleji. Je formátován na 499 sektorů plus adresář. Volá s.e pomocí D8:.

Paměťový prostor 1792-1812 (\$700-\$714) je naplněn přímo z disku ze sektoru 1.

1792

\$700

BFLG

Přenosová návěst. Vždy nula.

Krátkým Basic programem lze přečíst diskový adresář:

```
5 DIM R$(20)
10 OPEN #4,6,0,"D:*.*)"
20 INPUT #4,R$:TRAP 60
30 PRINT R$
40 IF R$(10,16)="SECTORS" THEN 100
50 GOTO 20
60 PRINT R$
100 CLOSE #4
```

Pro DOS 3.0 je nutná malá úprava v řádku 40.

Běžný diskový sektor má 128 byte, 0 až 127. Poslední tři byte využívá DOS.

BYTE POUŽITÍ

125	nižších šest bitů (0-63; \$0-\$3F): číslo souboru; dva nejvyšší bity: číslo dalšího sektoru (viz 126)
126	číslo dalšího sektoru (nižších osm bitů)
127	počet použitých byte v sektoru (0-125; \$0-\$7D)

Číslo dalšího sektoru je desetibitová hodnota zapsaná v 126. byte a dva významově nejvyšší bity ve 125. byte. Toto platí při textových souborech nebo pro Basic programy.

Binární soubory začínají vždy dvakrát \$FF, následují čtyři byte s adresou počátku a konce ukládání souboru.

1793

\$701

BRCNT

Počet sektorů prvotního zavádění. Tři - první tři sektory.

1794, 1795

\$702, \$703

BLDADDR

Adresa počátku ukládání sektorů. Pokud je zaváděn DOS, vždy 1792 (\$700).

1796,1797 **\$704,\$705** **BINTAD**

Inicializační adresa DOSu. Vždy 5440 (\$1540).

1798-1800 **\$706-\$708** **BCONT**

Kód instrukce JMP a adresa. Průběžná přenosová adresa (1812; \$714).

1801 **\$709** **SABYTE**

Maximální počet souběžně otevřených souborů. Běžně je nastaven na tři.

1802 **\$78A** **DRVBYT**

Každému bitu nastavenému v registru na 1 přísluší jedna disketa.

1803 **\$70B** **SAFBFW**

Nepoužit.

1804,1805 **\$70C,\$70D** **SASA**

Buffer rozvržení adres pro diskety a soubory.

1806 **\$70E** **DFSFLG**

Jestliže je na disku přítomen DOS.SYS, má DFSFLG nenulovou hodnotu.

1807,1808 **\$70F,\$710** **DFLINK**

Adresa prvního sektoru souboru DOS.SYS.

1809 **\$711** **BLDISP**

Počet přemístovaných byte sektor link (poslední tři), vždy 125.

1810,1811 **\$712,\$713** **DFLADDR**

Adresa FMS (D:) obslužné tabulky, 1995 (\$7CB).

1812	\$714	XBCONT
Zaváděcí program začíná zde.		
1900	\$76C	BSIO
Basic SIO procedury.		
1906	\$772	BSIOR
FMS disketové obslužné procedury.		
1913	\$779	-----
Zapísovací kontrolní návěst V/V diskových operací. POKE 1913,80 kontrolu z činnosti vyřadí, 87 kontrolu opět aktivuje.		
1995	\$7CB	DFMSDH
Obslužná tabulka FMS.		
2016	\$7E0	DINIT
Inicializační procedura DOS.		
4993	\$1381	FBC
Počátek řídicího bloku souboru FMS. Jedna až osm.		
5121	\$1401	FILDIR
128 byte dlouhý buffer sektoru diskového adresáře.		
5440	\$1540	MINIDUP
Počátek stabilně chráněné části souboru DUP.SYS.		
5450	\$154A	SFLOAD
Vstupní procedura DUP.SYS pro přenos binárních souborů.		

5542

\$15A6

STLOAD

Používán zároveň se SFLOAD.

5545

\$15A9

LOAD

Používán zároveň se SFLOAD.

Dodatky:

3889

\$F31

DOS 3

Jestliže se zde nachází hodnota 76 (\$4C), je přítomna jednodušší verze DOS 3 (78; \$4E pro novější verzi). Pro opravu několika málo chybiček zadejte tento krátký program:

```
10 FOR N=1 TO 9:READ A,B:POKE A,B:NEXT N
20 DATA 3889,78,3923,78,3943,78,3929,76,3895,76
30 DATA 3901,77,3935,77,3955,77,2117,240
```

20480-22527

\$5000-\$57FF

SELF-TEST

Kontrolní test počítače. Testovací program je uložen v paměti ROM od adresy 53248 (\$D300). Při jeho vyvolání pomocí BYE je celý program přesunut do tohoto prostoru a spuštěn. Stejně tak můžete test spustit zapsáním patřičné hodnoty na PORTB (54017; \$D301) a JMP (nebo USR) na inicializační vektor (viz 58481; \$E471 a 58496-58499; \$E480-\$E483).

NÁVĚSTÍ

NÁZEV	DEC-ADRESA	HEX-ADRESA	STRANA
ABUFPT	28-31	\$1C-\$1F	6
ACMISR	727,728	\$2D7,\$2D8	48
ACMVAR	1005-1015	\$3ED-\$3F7	67
ADDCOR	782	\$30E	59
ADRESS	100,101	\$64,\$65	15
APPMHI	14,15	\$0E,\$0F	4
ATACHR	763	\$2FB	54
ATRACT	77	\$4D	10
BASICF	1316	\$3F8	67
BCONT	1798-1800	\$706-\$708	82
BFENLO/HI	52,53	\$34,\$35	8
BFLAG	1792	\$700	70
BFLG	1792	\$700	81
BINTAD	1796,1797	\$704,\$705	82
BITMSK	110	\$6E	16
BIWTARR	1796,1797	\$704,\$705	70
BLDADDR	1794,1795	\$702,\$703	81
BLDADR	1794,1795	\$702,\$703	70
BLDISP	1809	\$711	71
BLDISP	1809	\$711	82
BLIM	650	\$28A	39
BOOT?	9	\$09	4
BOOTAD	578,579	\$242,\$243	34
BOTSCR	703	\$2BF	45
BPTR	61	\$3D	8
BRCNT	1793	\$701	70
BRCNT	1793	\$701	81
BRKKEY	17	\$11	5
BRKKY	566,567	\$236,\$237	33
BRUN	10060	\$274C	79
BSIO	1900	\$76C	72
BSIO	1900	\$76C	83
BSIOR	1906	\$772	72
BSIOR	1906	\$772	83
BUFADR	21,22	\$15,\$16	5
BUFCNT	107	\$6B	16
BUFRFL	56	\$38	8
BUFRLO/HI	50,51	\$32,\$33	7
BUFSTR	108,109	\$6C,\$6D	16
CARTCK	1003	\$3EB	67
CASFLG	783	\$30F	59
CASINI	2,3	\$02,\$03	3
CASSBT	1002	\$3EA	66
CAUX1	572	\$23C	34
CBAUDL/H	750,751	\$2EE,\$2EF	51
CCOMND	571	\$23B	33

CDEVIC	570	\$23A	33
CDTMA1	550,551	\$226,\$227	30
CDTMF3	554	\$22A	30
CDTMF4	556	\$22C	31
CDTMF5	558	\$22E	31
CDTMR2	552,553	\$228,\$229	30
CDTMV1	536,537	\$218,\$219	29
CDTMV2	538,539	\$21A,\$21B	29
CDTMV3	540,541	\$21C,\$21D	29
CDTMV4	542,543	\$21E,\$21F	29
CDTMV5	544,545	\$220,\$221	30
CH	764	\$2FC	54
CH1	754	\$2F2	51
CHAR	762	\$2FA	54
CHART	755	\$2F3	51
CHBAS	756	\$2F4	52
CHKSNT	59	\$3B	8
CHKSUM	49	\$31	7
CHLINK	1019,102	\$3FB,\$3FC	67
CHSALT	619	\$26B	36
CIX	242	\$F2	25
CKEY	1001	\$3E9	66
CLMJMP	6418	\$1912	77
COLAC	114,115	\$72,\$73	16
COLCRS	85,86	\$55,\$56	11
COLDST	560	\$244	34
COLINC	761	\$2F9	54
COLOR	200	\$C8	23
COLOR0	708	\$2C4	46
COLOR1	709	\$2C5	46
COLOR2	710	\$2C6	47
COLOR3	711	\$2C7	47
COLOR4	712	\$2C8	47
COLRSH	79	\$4F	10
COUNTR	126,127	\$7E,\$7F	18
CRETRY	668	\$29C	41
CRITIC	66	\$42	9
CRSINH	752	\$2F0	51
CRUX2	573	\$23D	34
DATAD	182	\$B6	22
DATALN	183,184	\$B7,\$B8	22
DAUX1/2	778,779	\$30A,\$30B	58
DBSECT	577	\$241	34
DBUF	7668	\$1DF4	78
DBUFLO/HI	772,773	\$304,\$305	58
DBYTLO/HI	776,777	\$308,\$309	58
DCOMND	770	\$302	57
DDEVIC	768	\$300	57
DDMG	10690	\$29C2	80
DELFIL	8649	\$21C9	79

Adresy paměti počítačů ATARI 600XL/800XL

DELTAC	119,12	\$77,\$78	16
DELTAR	118	\$76	16
DERRF	1804	\$3EC	67
DFFM	11528	\$2D08	80
DFINK	1807,1808	\$70F,\$710	71
DFLADDR	1810,1811	\$712,\$713	82
DFLADR	1810,1811	\$712,\$713	72
DFLAGS	576	\$240	34
DFLINK	1807,1808	\$70F,\$710	82
DFMCLS	2837	\$B15	73
DFMDDC	2983	\$BA7	73
DFMGET	2751	\$ABF	73
DFMOPN	2219	\$8AB	72
DFMPUT	2508	\$900	72
DFMSDH	1995	\$7CB	72
DFMSDH	1995	\$7CB	83
DFMSTA	2817	\$B01	73
DFSFLG	1806	\$70E	82
DIGRT	241	\$F1	25
DINDEX	87	\$57	11
DINIT	2016	\$7E0	83
DINT	2016	\$7E0	72
DIRLST	8505	\$2139	79
DMASAV	733	\$2DD	48
DMASK	672	\$2A0	41
DMENU	7951-8278	\$1F0F-\$2056	78
DOS 3	3889	\$F31	84
DOSINI	12,13	\$0C,\$0D	4
DOSOS	8309	\$2075	79
DOSVEC	10,11	\$0A,\$0B	4
DRETRY	701	\$2BD	44
DRKMSK	78	\$4E	10
DRVBYT	1802	\$70A	71
DRVBYT	1802	\$78A	82
DSCTLN	725,726	\$2D5,\$2D6	48
DSFLG	1806	\$70E	71
DSKFMS	24,25	\$18,\$19	6
DSKTIM	582	\$246	35
DSKUTL	26,27	\$1A,\$1B	6
DSPFLG	766	\$2FE	56
DSTAT	76	\$4C	9
DSTATS	771	\$303	57
DTIMLO	774	\$306	58
DUNIT	769	\$301	57
DUNUSE	775	\$307	58
DUPFLG	5533	\$159D	76
DVSTAT	746-749	\$2EA-\$2ED	50
EEXP	237	\$ED	24
ENDFMS	5377	\$1501	76
ENDPT	116,117	\$74,\$75	16

Adresy paměti počítačů ATARI 600XL/800XL

ERRFLG	575	\$23F	34
ERRNO	4789	\$12B5	75
ERRSAVS	195	\$C3	23
ESCFLG	674	\$2A2	42
ESIGN	239	\$EF	25
FBC	4993	\$1381	75
FBC	4993	\$1381	83
FCHRFLG	240	\$F0	25
FDSCHAR	3850	\$F0A	74
FE0F	63	\$3F	9
FILDAT	765	\$2FD	55
FILDIR	5121	\$1401	76
FILDIR	5121	\$1401	83
FILFLG	695	\$2B7	43
FINE	622	\$26E	36
FKDEF	96,97	\$60,\$61	14
FLPTR	252,253	\$FC,\$FD	26
FMTDSK	9856	\$2680	79
FMZSPG	67-73	\$43-\$49	9
FNDCODE	3742	\$E9E	74
FPSCR	1510-1515	\$5E6-\$5EB	68
FPSCR1	1516-1535	\$5EC-\$5FF	69
FPTR2	254,255	\$FE,\$FF	26
FR0	212-217	\$D4-\$D9	24
FR1	224-229	\$E0-\$E5	24
FR2	230-235	\$E6-\$EB	24
FRE	218-223	\$DA-\$DF	24
FREQ	64	\$40	9
FRESECT	4293	\$10C5	75
FRX	236	\$EC	24
FSDDIR	4206	\$106E	75
FTYPE	62	\$3E	8
GBYTEA	719,72	\$2CF,\$2D0	47
GETSECTOR	4358	\$1106	75
GINTLK	1018	\$3FA	67
GLBABS	736-739	\$2E0-\$2E3	49
GPRIOR	623	\$26F	36
HATABS	794-829	\$31A-\$33D	60
HELPPFG	732	\$2DC	48
HIBYTE	648	\$288	39
HIUSED	715,716	\$2CB,\$2CC	47
HNDL0D	745	\$E9	50
HOLD2	671	\$29F	41
HOLD3	669	\$29D	41
HOLD4	700	\$2BC	44
HOLDCH	124	\$7C	18
HOLD1	81	\$51	10
ICAX1Z	42	\$2A	7
ICAX2Z	43	\$2B	7
ICAX3Z/4Z	44,45	\$2C,\$2D	7

Adresy paměti počítačů ATARI 600XL/800XL

ICAX5Z	46	\$2E	7
ICAX6Z	47	\$2F	7
ICBALZ/HZ	36,37	\$24,\$25	6
ICBL LZ/HZ	40,41	\$28,\$29	7
ICCOMT	23	\$17	5
ICCOMZ	34	\$22	6
ICDNOZ	33	\$21	6
ICHIDZ	32	\$20	6
ICPTLZ/HZ	38,39	\$26,\$27	6
ICSTAZ	35	\$23	6
IMASK	651	\$28B	40
INBUFF	243,244	\$F3,\$F4	25
INISAV	6044,6045	\$179C,\$179D	77
INITAD	738,739	\$2E2,\$2E3	49
INITIO	6518	\$1979	77
INSDAT	125	\$7D	18
INTEMP	557	\$22D	31
INTERRUPT	512-551	\$200-\$227	26
INVFLG	694	\$2B6	43
IOCB0	832-847	\$340-\$34F	61
IOCB1	848-863	\$350-\$35F	62
IOCB2	864-879	\$360-\$36F	62
IOCB3	880-895	\$370-\$37F	62
IOCB4	896-911	\$380-\$38F	62
IOCB5	912-927	\$390-\$39F	62
IOCB6	928-943	\$3A0-\$3AF	62
IOCB7	944-959	\$3B0-\$3BF	62
IOCMD	192	\$C0	22
IODVC	193	\$C1	23
ISRODN	6630	\$19E6	77
ISRSIR	6691	\$1A23	78
JMPXBCONT	1788	\$706	70
JVECK	652,653	\$28C,\$28D	40
KEYDEF	121,122	\$79,\$7A	17
KEYDEL	753	\$2F1	51
KEYDIS	621	\$26D	36
KEYREP	730	\$2DA	48
KRPDEL	729	\$2D9	48
LBPR1	1406	\$57E	68
LBPR2	1407	\$57F	68
LBUFF	1408-1335	\$580-\$5FF	68
LCOUNT	563	\$233	33
LDFIL	10522	\$291A	80
LDMEM	6457	\$1939	77
LINBUF	583-622	\$247-\$26E	35
LINE	7588	\$1DA4	78
LISTDIR	3501	\$DAD	74
LKFIL	10608	\$2370	80
LMARGN	82	\$52	10
LNFLG	0	\$00	3

Adresy paměti počítačů ATARI 600XL/800XL

LOAD	5545	\$15A9	84
LOADAD	721,722	\$2D1,\$2D2	47
LOADFLG	5535	\$159F	77
LOGCOL	99	\$63	15
LOGMAP	690-693	\$2B2-\$2B5	43
LOMEM	128,129	\$80,\$81	18
LPENH	564	\$234	33
LPENV	565	\$235	33
LTEMP	54,55	\$36,\$37	8
MEMFLG	6046	\$179E	77
MEMLDD	5899	\$170B	77
MEMLO	743,744	\$2E7,\$2E8	50
MEMTOP	144,145	\$90,\$91	21
MEMTOP	741,742	\$2E5,\$2E6	49
MINIDUP	5440	\$1540	76
MINIDUP	5440	\$1540	83
MINTLK	1017	\$3F9	67
MLTTMP	102,103	\$66,\$67	15
NEWADR	654,655	\$28E,\$28F	40
NEWCOL	758,759	\$2F6,\$2F7	54
NEWROW	757	\$2F5	54
NOCKSM	60	\$3C	8
NOCLIK	731	\$2DB	48
NOFLAG	1	\$01	3
NSIG	238	\$EE	24
OLDADR	94,95	\$5E,\$5F	14
OLDCHR	93	\$5D	14
OLDCOL	91,92	\$5B,\$5C	13
OLDROW	90	\$5A	13
OPT	5534	\$159E	76
PADDL0	624	\$270	37
PADDL1	625	\$271	37
PADDL2	626	\$272	38
PADDL3	627	\$273	38
PADDL4-7	628-631	\$274-\$277	38
PALNTS	98	\$62	14
PBPNT	734	\$2DE	49
PBUFSZ	735	\$2DF	49
PCOLR0	704	\$2C0	46
PCOLR1	705	\$2C1	46
PCOLR2	706	\$2C2	46
PCOLR3	707	\$2C3	46
PDMSK	585	\$249	35
PDVMSK	583	\$247	35
PLYARG	1504	\$5E0	68
POKMSK	16	\$10	4
PPTMPA	588	\$24C	35
PPTMPX	589	\$24D	35
PRNBUF	960-999	\$3C0-\$3E7	66
PROMPT	194	\$C2	23

Adresy paměti počítačů ATARI 600XL/800XL

PTABW	201	\$C9	23
PTIMOT	788	\$314	59
PTRIG0	636	\$27C	38
PTRIG1	637	\$27D	39
PTRIG2	638	\$27E	39
PTRIG3	639	\$27F	39
PTRIG4-7	640-643	\$280-\$283	39
PUPBT1-3	829-831	\$33D-\$33F	61
RADFLG	251	\$FB	25
RAMLO	4,5	\$04,\$05	3
RAMSIZ	740	\$2E4	49
RAMTOP	106	\$6A	15
RDNXTS	4111	\$100F	74
RDVTOC	4235	\$108B	75
RECLEN	561	\$245	34
RECVDN	57	\$39	8
RELADR	568,569	\$238,\$239	33
RELADR	586,587	\$24A,\$24B	35
RENFIL	9783	\$2637	79
RMARGN	83	\$53	10
ROWAC	112,113	\$70,\$71	16
ROWCRS	84	\$54	11
ROWINC	760	\$2F8	54
RTCLOK	18,19,20	\$12,\$13,\$14	5
RUNAD	736,737	\$2E0,\$2E1	49
RUNADR	713,714	\$2C9,\$2CA	47
RUNSTK	142,143	\$8E,\$8F	21
SABYTE	1801	\$708	70
SABYTE	1801	\$709	82
SAFBFW	1803	\$70B	82
SASA	1804,1805	\$70C,70D	71
SASA	1804,1805	\$70C,\$70D	82
SAVADR	104,105	\$68,\$69	15
SAVFIL	12078	\$2F2E	80
SAVIO	790	\$316	59
SAVMSC	88,89	\$58,\$59	12
SCRFLG	699	\$2BB	44
SDLSTL	560,561	\$230,\$231	31
SDMCTL	559	\$22F	31
SELF-TEST	20480-22527	\$5000-\$57FF	84
SETUP	4452	\$1164	75
SFDIR	3873	\$F21	74
SFLOAD	5540	\$15A4	77
SFLOAD	5450	\$154A	83
SHFAMT	111	\$6F	16
SHFLOK	702	\$2BE	44
SHPDVS	584	\$248	35
SOUNDR	65	\$41	9
SRTIMR	555	\$22B	30
SSFLG	767	\$2FF	56

Adresy paměti počítačů ATARI 600XL/800XL

SSKCTL	562	\$232	32
STACKP	792	\$318	60
STARP	140,141	\$8C,\$8D	21
STATUS	48	\$30	7
STCAR	3966	\$26EE	79
STICK0	632	\$278	38
STICK1	633	\$279	38
STICK2-3	634,635	\$27A,\$27B	38
STLOAD	5542	\$15A6	84
STMCUR	138,139	\$8A,\$8B	21
STMTAB	136,137	\$88,\$89	20
STOPLN	186,187	\$BA,\$BB	22
STRIG0	644	\$284	39
STRIG1	645	\$285	39
STRIG2-3	646,647	\$286,\$287	39
SUBTMP	670	\$29E	41
SUPERF	1000	\$3E8	66
SWPFLG	123	\$7B	18
TABMAP	675-689	\$2A3-\$2B1	42
TEMP	80	\$50	10
TEMP	574	\$23E	34
TEMP1	786	\$312	59
TEMP2	787	\$313	59
TEMP3	789	\$315	59
TESTVER2	10483	\$28F3	80
TIMER1	780,781	\$30C,#30D	58
TIMER2	784,785	\$310,\$311	59
TIMFLG	791	\$317	59
TINDEX	659	\$293	40
TMPCOL	697,698	\$2B9,\$2BA	44
TMPLBT	673	\$2A1	42
TMPROW	696	\$2B8	43
TRAMSZ	6	\$06	3
TSTAT	783	\$318	60
TSTDAT	7	\$07	3
TXTCOL	657,658	\$291,\$292	40
TXTMSC	660,661	\$294,\$295	41
TXTOLD	662-667	\$296-\$29B	41
TXTROW	656	\$290	40
VBREAK	518,519	\$206,\$207	27
VDSLST	512,513	\$200,\$201	27
VIMIRQ	534,535	\$216,\$217	29
VINTER	516,517	\$204,\$205	27
VKEYBD	520,521	\$208,\$209	28
VNTD	132,133	\$84,\$85	20
VNTP	130,131	\$82,\$83	19
VPIRQ	568,569	\$238,\$239	33
VPRCED	514,515	\$202,\$203	27
VSERIN	522,523	\$20A,\$20B	28
VSEROC	526,527	\$20E,\$20F	28

Adresy paměti počítačů ATARI 600XL/800XL

VSEROR	524,525	\$20C,\$20D	28
VSFLAG	620	\$26C	36
VTIMR1	528,529	\$210,\$211	28
VTIMR2	530,531	\$212,\$213	28
VTIMR4	532,533	\$214,\$215	28
VVBLKD	548,549	\$224,\$225	30
VVTP	134,135	\$86,\$87	20
WARMST	8	\$08	3
WBLKI	546,547	\$222,\$223	30
WBOOT	10201	\$27D9	80
WMODE	649	\$289	39
WRDOS	4618	\$120A	75
WRTNXS	3988	\$F94	74
WTBUR	2591	\$A1F	72
XBCONT	1812	\$714	83
XDELETE	3122	\$C32	73
XFORMAT	3352	\$D18	74
XLOCK	3196	\$C7C	73
XMTDON	58	\$3A	8
XNOTE	3331	\$D03	73
XPOINT	3258	\$CBA	73
XRENAME	3033	\$BD9	73
XUNLOCK	3203	\$C83	73
ZCHAIN	74,75	\$4A,\$4B	9
ZHIUSE	717,718	\$2CD,\$2CE	47
ZLOADA	723,724	\$2D3,\$2D4	48
ZTEMP1	245,246	\$F5,\$F6	25
ZTEMP2	247,248	\$F7,\$F8	25
ZTEMP3	249,25	\$F9,\$FA	25

ADRESY PAMĚTI POČÍTAČŮ ATARI

© Pavel DOČEKAL

1. díl; popis paměti RAM
2. díl; popis paměti ROM

Neprodejný tisk určený pro vnitřní potřebu členů klubu.
Tisk povolen OK KNV Ostrava ve smyslu zák. 94/1949 Sb.

Vydal: AK ZO SSM SOUŽ, Jimramov
Náklad: 99 výtisků, OSTRAVA-1987
Formát: A5
Technická redakce: O. BURGER

