

edice C 64/128

COMMODORE
od A do Z

JAN KAVÁLEK

Úvodem

bych snad měl vysvětlit Proč a za jakých okolností vznikl tento sborník.

Každý, kdo se již s nějakým Počítačem trochu zná (byť je to třeba IQ nebo PMD) ví, jak užitečná a potřebná je každá informace o něm. Jistě mi dáte za Pravdu, že se to týká i naší,mimochodem Poměrně Početné, rodiny "šedesátčtyřkařů".

Manuál dodávaný sPolu s Počítačem Pro základní informaci jistě stačí. Nicméně bych si ho dovolil Přirovnat k návodu na obsluhu auta,který začíná odemknutím vozu,Pokračuje očistěním nohou a jejich následným vsunutím Pod volant a končí řazením jedničky.Jistě uznáte,že by s tím byl sPokojen asi málokdo.Ale my jsme v naProsto stejné,ne-li horší, situaci!My máme Počítač,ale neumíme ho Plně využívat.Zkoušíme toho tedy dosáhnout,ale nemáme žádnou dostuPnou literaturu. Proto se dáme cestou "Pokusů a omylů".Tato cesta je nejen velice trnitá a dlouhá,ale Především zbytečná.Proč by měl každý zvlášť objevovat to,co již objeveno bylo?Proč bychom tím měli ztrácet obrovské množství času? Potom se nedivme,že 99,99% dobrých Programů Pochází z ciziny.

V ruce držíte sborník informací,Postřehů,triků,atd.,který je výsledkem studia, Převážně zahraniční,literatury (sehnané všude možně) a nesPočetného množství Pokusů.(Mimochodem,držte ho Pevně,ať vám ho nějaký závistivý Atarista nevytrhne.) Pokusíme se v něm najít různé možnosti,jak Počítač Používat a jak s ním sPrávně zacházet. Podíváme se trochu blíže na jednotlivé Basicovské Příkazy,zmíníme se o zPůsobu uložení Programů, dat a Proměnných v Paměti, o systémových Proměnných,o obrazu a a zvuku. Všechno si vyzkoušíme na různých Příkladech. A Pak se dostaneme k tomu hlavnímu - totiž k Assembleru.Seznámíme se s Programováním Procesoru, s jednotlivými instrukcemi a vše oPět doložíme na

různých příkladech. A nakonec završíme svou touhu po vědění výpisem Basicu a Kernalu,který je přiměřeně obsáhlý.Ve sborníku samozřejmě nemůžou chybět různé, jak se již vžilo,tipy&triky,které mile překvapí i zapřísahlé Basickáře.

Tak tedy s chutí do toho a půl je hotovo. (Ta druhá polovina závisí už jenom na vás.) Přeji vám, aby C=64 byla vaším pomocníkem,který za vás ochotně odře všechnu práci,se kterou byste se jinak lopotili vy.Co to třeba může být,na to vám možná pomůže odpovědět hned následující kapitola. Nicméně ještě si neodpustím poznámku o tom, že všechno má mít svou míru.Doufám,že jste si to správně vyložili, takže už mi nezbývá nic jiného, než vám popřát

hodně štěstí a úspěchu!

KAPITOLA I

K čertu s těmi,
co řekli naše slova
před námi.
(Aurelius Donatus, 4.století n.l.)

Elektronický počítač
je symbolem naší budoucnosti
právě tak, jako jím byl
pro naše předky parní stroj.
(Colin Cherry, 1961)

Zkušenost je jméno,
které dáváme svým omylům.
(Oscar Wilde, 1854 - 1900)

1)Na co můžeme C=64 Používat

2)Jak zacházet s Počítačem

Na co můžeme C=64 používat

Vyjmenovat všechny možné aplikace se mi jistě nepodaří. Stejně jako většinu ostatních počítačů lze i C=64 používat na hraní her.S tím se setkáme většinou u uživatelů -začátečníků,ale není to pravidlo.Existuje množství her,které podporují fantazii, představivost, schopnost rychlé reakce,občas i logiku a různé další vlastnosti.Samostatnou kapitolu pak tvoří různé simulátory,které mohou být někdy i užitečné. Rozhodně si myslím, že hry k počítačům patří. Jednak dávají prostor k odpočinku a pak nám také velice přitažlivou formou ukazují,čeho všeho, že je ten počítač schopen.Ta grafika a zvuk!Zkušenějším dávají příležitost k rozebírání, případně i k přivlastnění si jisté její části.

Náš počítač se však může stát i naším učitelem.Výukové programy jsou velmi oblíbené, zvláště, pokoušejí-li se s žákem nějakým způsobem komunikovat. Některé pomocí klávesnice, jiné přes zvukový výstup -tedy např. hlasem.Předmět výuky je různý : počínaje výukou programování a konče výukou jazyků apod.

Počítač za nás může řešit i různé,zdánlivě od počítačů odvislé,úlohy.Dokáže nakreslit graf zadané funkce, umí ho zvětšit, zmenšit a já nevím co ještě. Umí udělat optimální rodinný rozpočet, tedy takový, který nejlépe vyhovuje zadaným podmínkám. S jeho pomocí můžeme snadno řešit logické i jiné problémy. Zkusme třeba tohle: číslo 25 lze bez obtíží napsat jako 5+5+5+5+5=25, tedy pěti pětkami. Dokážete napsat číslo 100

a)pěti jedničkami? c)pěti pětkami?

b)pěti trojkami? d)pěti devítkami?

Věřím, že se vám to dříve nebo později podaří, ale dříve než dobře naprogramovaný počítač s tím budete asi sotva hotovi. Už slyším tu spoustu námitek, že než počítač dobře naprogramujete,bude to spočí-

táno několikrát. To všechno je Pravda. Zvolil jsem asi příliš jednoduchý příklad. Zkusme třeba tohle:
Nádrž se naPlní Prvním Přítokem o 48 min. dříve než druhým; oběma současně však za 18 minut. Za kolik minut je naPlněna Prvním Přítokem?[24 min]
Anebo:
Na stavbu neustále vozí z nedaleké Pískovny dvě auta Písek.Když současně vyjely Proti sobě, setkaly se PoPrvé ve vzdálenosti 500m od Pískovny, Podruhé Při zPáteční cestě ve vzdálenosti 300 m od stavby.Zdržení Při nakládce a vykládce neuvažujeme. Dokážete určit Poměr rychlostí obou vozidel a vzdálenost Pískovny od stavby?[5:7; 1,2km]
Stále jsem vás nePřesvědčil?No dobře.
Při odstraňování Polomů Po větrné smršti shromáždili dřevorubci na Pasece 135m^3 smrkové kulatiny a 114m^3 borové. Na Pilu svážely dřevo dvě Party. Parta svážející borovice měla denní výkon 6m^3 ,druhá 7,5m^3 .Kdy bylo na Pasece stejné množství borového a smrkového dřeva?[Za čtrnáct dní]
Tak doufám, že to stačilo.Ale musím s vámi souhlasit,že vytvoření dobrého Programu je věc zaPeklitá a nezřídka i náročná.Ani ne tak na matematiku,jako spíš na logické myšlení,ale to chce cvik.

C=64 má ve své třídě velmi kvalitní zvuk a obraz.Toho můžeme využít Při skládání hudby,titulkování filmů,atd.

Počítač se velmi často Používá Při Práci s různými informacemi a daty. Umí je nejen Pořídit, ale i uchovat,třídit.Umí v nich hledat,může je doPlňovat,aPod.

Zkrátka myslím,že bychom se mohli shodnout na tom, že Počítač je velmi užitečná věc, která je tu Proto,aby nám Pomáhala.C=64 je ve své třídě jedním z nejlePších.Pro domácí účely myslím Plně Postačuje. Jeho vnitřní struktura je logická a snadno PochoPitelná.Bude-li se vám někdy zdát, že je to blbec,Pak máte Pravdu: udělá vše,co mu Přikážete.

Jak zacházet s Počítačem

NejPrve trochu historie. Firma Commodore (Plným jménem Commodore Business Machines, nyní Commodore Technology),kterou založil legendární Jack Tramiel má hlavní stan ve West Chestru v U.S.A..Svuj úmysl vyrábět Počítače oznámila v roce 1977.Počítač dostal jméno PET (angl. oblíbené domácí zvíře) a byl, stejně jako jeho Potomci,založen na mikroProcesoru 6502. Tím byla odstartována série stolních Počítačů,která v Průběhu let doznaly celou řadu vylePšení. Všechny modely byly integrovány s obrazovkou. Velikou Předností těchto stolních Počítačů, byl (a je) celostránkový editor,v té době věc téměř nevídaná.

Co se týče softwarového vybavení, nelišilo se Příliš od toho,které známe -Basic v.2.0.

Na Počátku osmdesátých let Přesahovala cena Počítače tisíc dolarů.V té době vzniká Počítač VIC 20, jehož cena byla 300 dolarů. Měl 3.5KB RAM Pro Basic a zobrazení 22 znaků na řádek. Jako jeden z Prvních měl výstuP na běžný TVP, jako První Pracoval v osmi barvách,měl zabudovaný 3-kanálový zvuk. Generátor a umožňoval PřiPojení jednoho křížového ovladače.

V roce 1983 se objevuje na scéně dosud nejProdávanější Počítač na světě - C=64 a krátce Po něm C=128. C=64 má 64KB RAM, 20KB ROM,rozhraní RS-232,tříkanálový zvukový syntezátor a vylePšený zákaznický video-obvod. Dále má dva řídící Porty,sériový Přenos a Přímé ovládání magnetofonu.

A jak bychom měli s Commodorem zacházet, aby co nejdéle vydržel?OdPověď je jednoduchá:co nejlépe. NejPrve je třeba zakrytovat user Port a exPansion Port (Pokud nejsou zakryty konektorem) a to to nejlépe PřelePením samolepící páskou. Totéž se týká výstuPu na magnetofon.Těmito místy totiž uniká do Počítače obrovské množství Prachu a Prach je

náš nepřítel čís.2.Stejně tak doporučujiPřikrývání počítače. Na C=64 se prodává i speciální ochranný kryt z kouřového Plexiskla, ale postačí i utěrka, nebo cosi podobného.

Nepřítelem číslo 1 jsou náboje.Proto se nedotýkejme (čímkoli) žádných vývodů, byť je to třeba jenom chráněný control Port.Obvykle se nedoporučuje vypnutí a znovu zapnutí počítače dříve než za 15-30 sekund.A teď pozor! K zapnutému počítači nic nepřipojujte. Nejcitlivější je výstup na magnetofon, pak expansion port, user port, control port , serial, video a nejméně citlivý je výstup z videomodulátoru na TVP. Přívody k jednotlivým zařízením nepohybujme před obrazovkou!U přívodu k magnetofonu se tím bezpečně zničí výstup z procesoru a, že to není laciná legrace, to vám potvrdí každý třetí majitel Commodorky.Já se k nim počítám také.

A nakonec bych ještě upozornil na to,že zdroj dodávaný k C=64 má sklony k přehřívání a nadělá mu dobře nepřetržitý provoz.Dlouhým provozem naprázdno se může i zničit (zdroj je zapojen v síti, ale počítač je vypnutý).Závada se projeví rozbitím televizního signálu (rozpad obrazu) a zvuku,zastavením,nebo podstatným zpomalením,běhu programu a nesmyslnými výsledky. Závada se obvykle neprojevuje ihned po zapnutí počítače,ale až asi za 45 min..Je velmi nebezpečná, protože místo 5V je na obvodech téměř 10V. Porucha se bohužel svými příznaky velmi podobá nepodařenému programu ve stroj. kódu, ale existuje způsob jak jí odhalit s jistotou a to měření na vývodech control portu (viz. zapojení).Pokud se vám to náhodou někdy stane, ihned vypněte počítač a kupte si zdroj nový, nebo ho předejte, spolu se sekáčkem a kladivem, mimořádně zručnému kutilovi. Vše je totiž zalito v pryskyřici, která se tváří neústupně.Myslím,že se tedy docela vyplatí občas ten zdroj vypnout.

KAPITOLA II

Pok za rokem jsme lépe vybaveni,
abychom dosáhli toho,po čem toužíme.
Ale po čem vlastně toužíme?
(Bertrand de Jouvenel, 1970)

Práce se rozpíná tak,
aby vyplnila čas určený k jejímu
dokončení.
Výdaje stoupají tak,
aby se vyrovnaly příjmům.
(První a druhý Parkinsonův zákon)
(Cyril N. Parkinson, 1957)

Není to tím,že nevidí řešení.
Je to tím,že nevidí problém.
(Gilbert K. Chesterton, 1952)

1)Práce s klávesnicí

2)Proměnná

3)Příkazy

4)Priority operací

5)Jednoduché programy v Basicu

Práce s klávesnicí

je velice jednoduchá. Jednotlivá tlačítka mají poměrně vysoký zdvih a jsou spolehlivá,což zabraňuje různým chybám.

Tlačítka na klávesnici jsou uspořádána podle americké normy,což je pro programování v angličtině jistě výhodnější.

Jistě vás zajímá,jak je to uděláno,že počítač si s klávesnicí tak dobře rozumí. Není to tak složité,jak by se mohlo zdát.Dělá to tedy takto: konstruktér rozdělil klávesnici do řádků a sloupců a tam kde se kříží řádek se sloupcem napojil vždy po jednom tlačítku.Jaký to má význam?Je to nejelegantnější způsob (alespoň zatím),jak lze klávesu tzv. "přečíst".Počítač,nebo přesněji řečeno obvod,který zabezpečuje styk s klávesnicí (ComPlex Interface Adapter -CIA 6526),vyšle na první řádek signál,zatímco ostatní řádky jsou bez signálu a postupně zkouší jednotlivé sloupce, není-li na nich tento signál.Není-li tam,příslušné tlačítko je rozpojené (nevede el. proud).A je-li tam,tlačítko bude nepochybně stlačeno.V tom případě podle zapnutého řádku a k němu příslušejícímu sloupci zjistí v tabulce kód klávesy. Tabulka je uložena v paměti ROM a nutno dodat,že není jenom jedna..Mění se podle toho jestli je příslušná klávesa stlačena samotná, nebo dohromady např. se SHIFTem, nebo C=. Ale k tomu se dostaneme,to teď není podstatné.Důležité je vědět, že každé klávese přísluší jistý kód.Z tohoto existuje jediná výjimka a to klávesa RESTORE, ale o ní až později.A co se děje s kódem klávesy potom?Začne se zjišťovat,není-li to náhodou kód kláves C= , SHIFT,CONTROL a RUN/STOP.Je-li to on,změní se příslušná systémová proměnná.A není-li to on, testuje se na CLR atd..Pokud se jedná o znakovou klávesu a o ty zbylé, zapíše se jejich kód do tzv. vstupního buffru, kde čeká na to,až bude potřeba.

Tento popis je trochu zjednodušený,ale v pod-

statě to tak opravdu chodí.Jak je to úplně přesně, to se včas dozvíme.

Většině kláves,až na zmíněné výjimky,přísluší jistý kód -ASCII kód.A známe-li tabulku znaků,které k jednotlivým kódům přísluší, můžeme toho využívat.Hlavně při Programování v Assembleru. Basic má tu výhodu, že umí ke kódu najít znak a opačně, ale k tomu se dostaneme až o Podkapitolu dál.

Zapneme-li počítač a začneme do něj něco "ládovat", může to vyvolat dojem,že se vcelku nic neděje a,že počítač začne vlastně pracovat až po zadání příkazu RUN. Opak je pravdou.I když si kurzor spokojeně pobliká,počítač pracuje na plné obrátky. Nejenom, že se "čte" klávesnice, ale pracuje i obrazovkový editor a překladač Basicu a to navíc za přispění operačního systému. To všechno jsou ramy ve stroj. kódu (v pamětích ROM), bez kterých bychom byli vyřízení. Neexistoval by žádný kurzor, natož nějaký souvislý text na obrazovce.Takže mějme stále na paměti, že nás obsluhuje Program (a to na rozdíl od načerno kopírovaných her legální), který je dílem CBM.To, že se nám počítač jeví jako dobrý je do značné míry způsobeno právě kvalitou již zmíněných zabudovaných Programů. Takže když by se řeklo,že práce na Programu v Basicu (a nejen v něm) je práce na Programu za pomoci Programu, pak je to pravda a nic víc než čistá pravda.

Ale mám takový pocit,že jsem trochu moc odbočil od základního tématu. Ke klávesnici už se snad ani vracet nebudeme; myslím, že už to stačilo, tak se radši podíváme na následující Podkapitolu.

Proměnná

Přesně definovat pojem proměnné je velice obtížné. Je to něco podobného, jako pochopit správný význam cizího slova.Myslím,že bude nejlepší ukázat si několik příkladů.Bez proměnných se neobejdeme - používají se v Basicu, Assembleru -no zkrátka všude. V našem Basicu má jméno proměnné max. 2 znaky. Když napíšeme jméno delší,uvažují se jen první dva znaky.Ještě dodám,že kód ani jednoho znaku v názvu proměnné nesmí (a po pravdě řečeno ani nemůže) být větší, než 127 (sedmý bit je nulový).Ale teď už se vraťme k příkladům.

Představme si podnik,ve kterém se vedoucí oddělení rozhodl přidat všem na dílně 648,35 Kčs ve formě prémie za splnění úkolu.Seznam všech pracovníků má v počítači a s ním i další potřebné údaje, včetně platu. Protože dílna má více než půldruhé třetiny ze sta pracovníků,rozhodl se udělat ve své databázi krátký program, který ke mzdě přičte prémii.Jak na to asi půjde?No vezme mzdu prvního pracovníka, přičte k ní prémii a výsledek si zapíše. Pak totéž udělá s druhým atd..Aby přímo do programu nemusel psát výšku platu každého zvlášť (to by mohl popoužít kalkulačku), jistě si zavede položku MZDA;do které vždy načte příslušnou hodnotu.Vypadá to takhle:

výplata = MZDA + 648,35

Za položku MZDA tedy dosazujeme jednotlivé hodnoty a v položce výplata pak máme výsledek součtu.Bude-li vedoucí tak štědrý,že se rozhodne přidávat častěji, bude pro něj jistě jednodušší zavést ještě jednu položku -PREMIE, kterou vždy nastaví na patřičnou hodnotu podle toho,jak se dílna činila.Mohlo by to vypadat třeba takhle:

PREMIE = 648,35:výplata = MZDA + PREMIE

Řekněme,že mzda Víta Pecháčka bude 1500,-Kčs.Takže

výplata = MZDA + PREMIE ,takže nyní položka výplata ukrývá hodnotu rovnou součtu 1500+648,35.
(1500) (648,35)

Z hlediska obsazení Paměti a složitosti některých výpočtů je výhodné dělit číselné proměnné do dvou skupin:

a)Reálné (23;45;-0.56;23.1234567;2.7345E+11;47.44)

b)Celočíselné (23;45;55;0;7276;32000;55;-238;-668)

A ještě jak se značí:

Real (reálné)	Integer (celočíselné)
A ,AA ,X ,XY aPod.	A% ,AA% ,X% ,XY% ,atd.
s rozsahem +-1.70141183E+38	s rozsahem +-32767

Jistě vás nepřekvapí,že stejně jako máme číselné proměnné, tak máme i řetězcové.Takoví řetězcová proměnná se značí znakem $ (A$,AA$,X$,XY$) a je třeba vědět, že do jednoho řetězce se vejde max. 255 znaků.Příklad:

A$ = "Pavel":B$="Mudřík":C$=A$+B$

V položce C$ máme PavelMudřík. Na co se proměnné hodí,to se ukáže až budeme znát nějaké příkazy,ale jistě je patrné,že asi budou užitečné.

V počítači jsou však ještě jiné proměnné tzv. systémové. Ukazují nám,jak jinak,stav systému. Jejich hodnota se mění podle toho, jak se mění podmínky, za jakých počítač pracuje. Uvedu příklad: existuje syst. proměnná,která ukazuje na řádek, na kterém je právě kurzor. Je-li např. na nejvrchnějším,má hodnotu 0 (počítají se od nuly).A je-li naopak na posledním,má hodnotu 24.Těmto proměnným se ještě budeme věnovat,protože si to zaslouží.

Příkazy

jsou naším prostředníkem při zapisování programu. Čím jich známe více, tím snadněji se nám programuje.Na následujících několika stránkách najdete seřazené příkazy a standardní proměnné podle abecedy. Příklady použití jsou o podkapitolu dál. Ještě bych měl dodat, že k většině příkazů existuje také jejich zkrácený zápis. Oceníte ho, až budete psát nějaký delší program.A nyní k jednotlivým příkazům a proměnným:

ABS (absolute - absolutní)

A=ABS(-324):B=ABS(X*2-12*Y):IF C=ABS(C) THEN ?"OK"
Numerická funkce, která vypočte absolutní hodnotu členu v závorce.

AND (and - a)

AA=12678AND1563:IF (B*3<10)AND(B*3>=0) THEN B=B+11
Operátor, provádějící logický součin dvou výrazů.

ASC (ASCII =American Standard Code for Information Interchange - soubor alfanumerických znaků ve formě osmibitového kódu)

PRINT ASC("a"):ON ASC(A$)-64 GOTO 300:XY%=ASC("*")
Funkce,která ke znaku vyhledá jeho příslušný ASCII kód.Kódy všech znaků se pohybují od 0 do255.V žádném případě nemůže být větší,nebo menší.

ATN (arcustangens)

XY=ATN(AA):AB=ATN(12):BN=ATN(X+34*Y-12)
Numerická funkce,která je opačná k funkci tangens. Její význam oceníme při matematických výpočtech.

CHR$ (character - znak)

A$="OKA"+CHR$(89)+CHR$(13): PRINT A$: BB$=chr$(66)
Funkce,která ASCII kódu přiřadí příslušný znak. Je opakem funkce ASC.

CLOSE (uzavřít)

OPEN1,4,7:PRINT#1,"Test zavírání souboru":CLOSE 1

Příkaz k uzavření souboru.V praxi to znamená konec práce s některým periferním zařízením.

CLR (clear - nulovat,čistit)

A%=5: A$="Zkouška CLR": CLR: PRINT A%,A$,"Totˇ vše"

Nulování, mazání (zkrátka čistění) hodnot všech proměnných.

CMD (command - povel)

OPEN 9,1,1 : CMD 9 : LIST

Příkaz,který způsobí přesměrování výstupu (původně na obrazovku) do příslušného souboru (v tomto případě na magnetofon). Vše co by se tedy tisklo na obrazovku (výpis progr.), bude se nahrávat na pásku.

CONT (continue - pokračování)

PRINT A,B$:A=34:B$="":CONT

Příkaz, který po zastavení programu příkazem STOP, nebo klávesou STOP spustí program znovu od zastaveného místa. Hodnoty proměnných se zachovávají a můžeme je nejen nechat vypsat,ale i měnit.

COS (cosinus)

A=COS(45*3.14159265/180) : B=COS(3.14159265/4)

Matematická funkce,která vypočte kosinus argumentu v závorce.Nutno dodat, že pracuje s argunentem zadaným v radiánech.Příklad převodu 45 stupňů na rad jsem uvedl. Platí jednoduchý převodní vztah, podle kterého je 180stupňů rovno 3.14159265(Pí)radiánům.

DATA

DATA Ringo,George,Paul,John,4:READ A$,B$,C$,D$:E$=
"BEATLES = "+A$+B$+C$+D$:READ A:PRINT E$,A

Příkaz umožňující uložení čísel,nebo řetězců oddělených čárkou,v programu.

DEF FN (define function - definuj funkci)

```
DEF FN AA(X)=X+71Y+10:PRINT FN AA(5),FN AA(3)+6
```

Příkaz pro definování vlastní funkce. Výhoda tohoto příkazu spočívá v tom, že nemusíme opakující se výpočty psát stále znovu, ale stačí "zavolat" jenom tu správnou funkci. Pro název funkce platí stejná pravidla jako pro název proměnné.

DIM (dimension - rozměr)

```
DIM F$(3,4,5),B$(3,2),A(30,2,3),B(20):A(0,1,1)=552
```

Příkaz, který zavede jedno, dvou, nebo třírozměrné pole čísel, nebo řetězců. Index jednotlivých polí se pohybuje od nuly k "nadimenzované" hodnotě. Zbývá dodat, že pole o rozměru 0-10 je dimenzováno automaticky, tedy bez použití příkazu DIM.

END (konec)

```
PRINT"Konec":END
```

Zakončení programu. Běh programu končí, jak jinak, na posledním programovém řádku. Nicméně následují-li podprogramy, nezbývá nic jiného, než poslední řádek hlavního programu opatřit tímto příkazem.

EXP (exponential function - exponec. funkce)

```
E=EXP(1):PRINT E
```

Matematická exponenciální funkce. Uvedený příklad můžeme matematicky zapsat jako $E=e^1$, kde e je Eulerova konstanta, základ přirozených logaritmů a je rovna přibližně číslu 2,71828183.

FN (function - funkce)

```
DEF FN K(T)=TIME+T:TIME = FN K(250)
```

Po nadefinování můžeme funkci volat klíčovým slovem FN a jménem, popř. parametrem. Ulehčuje práci při různých výpočtech a šetří místo v paměti.

FOR...TO...[STEP...] (pro...do...[krok..]

```
FOR X=10 TO 20:PRINT X:NEXT X
```

FOR Y=20 TO 10 STEP -1:PRINT Y:NEXT Y
Tento soubor Příkazů nám Pomůže vytvářet tzv. cykly,tedy úseky Programu, které se opakují tak dlouho,dokud není splněna Podmínka udaná v záhlaví toho kterého cyklu. Cyklus na Předchozí stránce říká asi tohle: Pro X Pohybující se od 10 do 20 dělej (rozumí se s krokem 1,není-li uvedeno jinak):tiskni hodnotu Proměnné X:Přičti krok k Proměnné v záhlaví cyklu a Porovnej jí s konečnou hodnotou(20). Je-li menší, nebo rovna, Pak cyklus opakuj, je-li větší,Pak Pokračuj dalšími Příkazy za NEXTem.Ještě je důležité vědět,že cykly se mohou Prolínat,takže můžeme v cyklu udělat cyklus v cyklu a to tolikrát,kolikrát nám to dovolí kaPacita zásobníku, ale Pro většinu běžných aPlikací jejich Počet stačí.

FRE (free - volný,neobsazený)

PRINT FRE(X)
Funkce, která nám vypíše, kolik místa v Paměti nám ještě zbývá.

GET (vezmi,seber)

GET A$,B$:IF A$=CHR$(13) OR B$="" THEN STOP
OPEN2,1,0:GET#2,B$:PRINT B$;
Příkaz k přečtení znaku,poPříPadě i čísla.Znak může být v klávesnicovém buffru (GET),nebo v nějakém souboru (GET#).

GOSUB (go to subroutine - jdi do PodProgramu)

GOSUB 1000:A=55:GOSUB 500:END
Příkaz umožňující skok do PodProgramu.Správný,dobře strukturovaný Program,by měl mít dvě části:PodProgramy a hlavní Program, který je Používá.Jako PodProgram vytvoříme tu část Programu, která by se mohla opakovat.Jednak tím šetříme Paměť a Pak taky sebe Při PříPadném "oživování".Na konec PodProgramu dáváme Příkaz RETURN, který zPůsobí návrat na řádek,odkud byl PodProgram volán a to hned za Pří-

kaz GOSUB.S Počtem PodProgramů v PodProgramu je to asi tak jako s Počtem cyklů v cyklu.

GOTO (go to - jdi na [ne do, ale NA!Neplést!])

```
10 GET A$:IF A$<>CHR$(13) THEN PRINT A$;:GOTO 10
```

Příkaz k Pokračování běhu Programu na stanoveném řádku.

IF...THEN... (jestliže...,Pak...)

```
100 INPUT"X";X:IF X>=10 THEN 100
560 IF POS(0)>20 THEN PRINT"Konec řádku"
```

Tyto dva příkazy nám dávají možnost měnit chod programu v závislosti na splnění dané podmínky.Po příkazu THEN může následovat buď číslo řádku, na kterém se má pokračovat (zastupuje příkaz GOTO) , nebo jiné příkazy, které mohou vyplnit zbytek řádku zaTHEN.NedoPoručuji dělat za Pomoci těchto příkazů cykly.Jsou asi 10x Pomalejší,než FOR-TO-NEXT.

INPUT (vstuP)

```
INPUT "Jak se jmenujete" ;A$ : FOR X=1 TO LEN(A$):
PRINT LEFT$(A$,X) : NEXT : INPUT A,B,C
OPEN3,1,0:FORx=0TO3:INPUT#3,A$:PRINT A$:NEXT:CLOSE
```

Příkaz vstuPu.Umí přečíst jak řetězec,tak i hodnotu čísla.V PříPadě,že místo čísla zadáme znak,hlásí to a oPakuje se tak dlouho, dokud nedostane, co žádá. V zadávaném textu se nesmí vyskytovat čárka, protože jí chápe jako oddělovač.Položku za ní přiřadí další Proměnné (je-li tam nějaká),nebo to jen tak mimochodem ohlásí.Opět se liší INPUT z klávesnicového buffru od INPUT#u ze souboru.

INT (integer - celá hodnota)

```
A=INT(-2.6):B=INT(2.1):C=INT(-5.2):D=INT(3.7)
```

Funkce, která od desetinného čísla "odsekne" jeho desetinnou část a tím z něj udělá číslo celé. Nemá nic sPolečného se zaokrouhlováním. Vychází z principu uložení kladných a záporných čísel ve dvojko-

vé soustavě.

LEFT$ (left - levý)

A$="Ahoj.":PRINT LEFT$(A$,2)

Tento příkaz pracuje s řetězcem a dokáže z něj oddělit X znaků zleva.

LEN (length - délka)

INPUTA$:IF LEN(A$) > 10 THEN A$ = LEFT$(A$;10)

Tato funkce dokáže spočítat počet znaků zadaného řetězce.

LET (nechť,ať)

LET X = 55 : LET A$ = "Dukej."

Přiřazovací příkaz.Obvykle se nepoužívá,ale nic se tím nezkazí.

LIST (seznam)

LIST -100 ,nebo LIST 10- ,nebo LIST 10 ,či LIST5-9

Příkaz pro vypsání programu. Může se vypsat jeden řádek,určitý úsek,nebo celý program.Každý,kdo chce ochránit svůj program v Basicu před očima rozebíračů, se ho snaží všelijak blokovat.Jak na to, to si povíme později.

LOAD (zavést,naplnit)

LOAD ["jméno"][,kód periférie][,sekundární adresa]

Příkaz pro čtení dat, např. programu, z periférie, např. magnetofonu,nebo diskety.

LOG (logarithm - logaritmus)

PRINT LOG(2.71828183)

Matematická funkce.Jedná se o logaritmus přirozený (v základě má číslo e). Používá se při různých matematických výpočtech.

MID$ (middle of string - střed řetězce)

N$="Ahoj." : PRINT MID$(N$,3,2)

Příkaz,který z daného řetězce vytvoří řetězec nový tím zPůsbem, že si z něj Prostě kousek vezme. Příklad na Předchozí strónce chápeme takto: vytvoř nový řetězec a to tak,že vezmi z řetězce N$ dva znaky,Počínaje třetím.

NEW (nový)

NEW

Tento Příkaz Přenastaví některé syst. Proměnné a za2048 uloží několik nul,takže Program se jeví jako vymazaný.Oživit jde malým trikem.Jinak Po zadání tohoto Příkazu můžeme začít Psát nový Program.

NEXT (příští,následující)

FORX=0TO999: POKE1024+X,36: FORY=0TO100: NEXT Y,X

Tento Příkaz je součástí cyklu FOR-TO.ZPůsobí Přičtení kroku (není-li uveden, je jednotkový) k dané Proměnné.Pak jí Porovná s cílovou hodnotou, kterou jsme zaPsali v záhlaví cyklu a je-li menší, nebo rovna,oPakuje se úsek Programu mezi FOR-TO a NEXT. Je-li větší,Program Pokračuje dále.

NOT (ne,oPak,negace)

X=5:A=NOT X:B=(-(X+1)):IF NOT A=B THEN ?"Dejte Počítač do oPravy!"

Logický oPerátor negace.Myslím,že není treba k němu cokoli dodávat.

ON (v závislosti,Podle)

ON P GOTO (gosub) 100,200,300,400,500

ON (X>Y)+1 GOTO 100 (IF X<Y THEN 100)

Výhodný Příkaz k větvení Programu v závislosti na sPlnění Podmínky,nebo Podle hodnoty Proměnné. Rozhodování je několikrát rychlejší,než Při klasickém IF-THEN.Je-li Proměnná v záhlaví (P) nulová,Pokračuje se na následujícím řádku. Je-li jedna,Pak budeme Pokračovat na řádku 100,2..200,3..300,4..400, 5..500 a 6 a více oPět na následujícím.

OPEN (otevřít)

OPEN 1,4,7 : PRINT#1,"Zkouška" : CLOSE 1

Příkaz k otevření souboru. 1 je číslo souboru (od 0 do 255); 4 je číselný kód periférie,se kterou budeme pracovat a 7 je sekundární adresa. Myslím, že se k tomuto tématu ještě dostaneme.

OR (nebo)

PRINT 5 OR 2 : IF X>5 OR X<2 THEN 350

OPerátor logického součtu.

PEEK (Pokukovat)

POKE 1024,28 : PRINT PEEK(1024)

Tento příkaz slouží k přečtení obsahu paměťového místa. Náš počítač je osmibitový,proto obsah libovolného paměťového místa může mít hodnotu v rozmezí od 0 do 255. Pomocí tohoto příkazu můžeme číst z paměti RAM,Basic ROM,Character a Kernal ROM.Nelze číst obsah paměti RAM pod Basic a Kernal ROM.

POKE (štourat se,rýpat se,vrtat se do něčeho)

POKE 56325,5:POKE 646,1:POKE 199,1

Příkaz sloužící k zápisu čísla do paměti RAM. Hodnota,kterou zapisujeme na danou adresu se musí pohybovat od 0 do 255 ($2^8-1=255$).

POS (Position - Pozice)

IF POS(0)>30 THEN PRINT CHR$(13)

Pomocí tohoto příkazu můžeme zjistit polohu kurzoru na obrazovkovém řádku.

PRINT (tisknout)

PRINT A+B;"=";A;B,"Dobře":OPEN2,1,1:PRINT#2:CLOSE2

Tento příkaz je jedním z nejpoužívanějších. Tiskne na obrazovku,nebo příslušné údaje zapisuje do souboru (PRINT#).Jeho zkrácený zápis je velice jednoduchý: ? = PRINT.(Nelze ovšem napsat ?#,1.)Doporučuji ohmatat si ho co nejlépe. Zvláštní úlohu má u

tohoto Příkazu čárka a středník (velké rozestupy jsou po čárce,malé po středníku).

READ (čti)

```
READ A$,B$:READ X:PRINT A$+B$,X
```

Příkaz pro načtení položky z datařádku. Položky na datařádku (-cích) jsou odděleny čárkami.

REM (remark - Poznámka)

```
10 PRINT RND(1):REM Zkouška generátoru "náhody"
```

Tento příkaz umožňuje vkládat do programu komentáře, které celý program zpřehledňují.Zbytek počítačového řádku za REM je tedy určen pro vlastní poznámky.

RESTORE (obnovit)

```
RESTORE:READ A$
```

Příkaz, který způsobí, že ukazatel na aktuální položku datařádku (tedy na tu,která leží hned za posledně přečtenou) se nastaví opět na počátek (tedy na první položku datařádku). Data se tedy začínají číst znovu od počátku.

RETURN (návrat)

```
A=3*X↑2:PRINT"Výsledek ";A:RETURN
```

Tento příkaz dáváme na konec podprogramů. Způsobí návrat za příkaz GOSUB, který volal poslední podprogram.

RIGHT$ (right - pravý)

```
A$="Napěťový zdroj" : PRINT RIGHT$(A$,3)
```

Příkaz, který je velice podobný LEFT$.Parametry má dva.Jednak je to řetězec a pak číslo,nebo proměnná určující z kolika znaků (samozřejmě zprava) se vytvoří nový řetězec.

RND (random - náhoda)

```
A=INT(RND(1)*1000) (náhodná čísla 0-999)
```

Příkaz,který generuje náhodné číslo.Velikost tohoto čísla se pohybuje od 0 do 0.99999999 (viz. příklad),proto se obvykle násobí nějakou konstantou a pak se z něj dělá celá hodnota.Tím dostanem číslo, které je v praxi mnohem více použitelné.

RUN (běžet)

RUN ,nebo RUN 100
Příkaz pro spuštění programu (od počátku, nebo od zadaného řádku).Nuluje proměnné.Funguje jenom tehdy, je-li na úplném počátku RAM pro Basic uložená nula..

SAVE (spasit,zachránit,střádat)

SAVE ["jméno"][,číslo periférie][,sekundár.adresa]
Příkaz pro uložení (často opravdu pro zachránění) programu do periferního zařízení (mgf.,disketa,... atd.). Uložení programu na magnetofon je poměrně spolehlivé, ale dosti zdlouhavé.Proto se používají programy tzv. Turba. Záznam na pásce je podstatně kratší,ale poněkud méně spolehlivý.V praxi to více než dostačně postačuje, proto jsou tyto programy velice oblíbené.

SGN (signum)

PRINT SGN(-10),SGN(-5),SGN(0),SGN(4),SGN(1237)
Matematická funkce, která nabývá hodnot -1,0 a 1. Její hodnota závisí na znaménku argumentu.Pro 0 je nulová..

SIN (sinus)

A=SIN(90*3.14159265/180)
Matematická funkce sinus. Argument je třeba udat v radiánech (180stupňů odpovídá 3.14159265 [pí]rad).

SPC (space - prostor)

PRINT"Zdravím";SPC(5);"Vás"
Funkce,která mezi jednotlivé řetězce vkládá stano-

vený Počet mezer.První mezera následuje za Posledním znakem Prvního řetězce a Poslední je Před Prvním znakem druhého řetězce.Tím se tento Příkaz liší od Příkazu TAB.

SQR (square root - druhá odmocnina)

```
FOR X=0 TO 36:PRINT SQR(X),SQR(X)*SQR(X),X:NEXT X
```

Pomocí tohoto Příkazu můžeme vyPočítat druhou odmocninu.Parametrem je číslo od nuly výš.

STATUS (stav)

```
OPEN2,8,4:INPUT#2,A$:IF STATUS AND 64 THEN CLOSE 2
```

Standardní vstuPně/výstuPní Proměnná.Ukazuje konec bloku,odvozuje se z ní hláška LOAD ERROR,aPod.Je v Podstatě obrazem adresy 144 (PoPř. 663,je-li Periférií modem).Protože je to Proměnná, berou se v úvahu Pouze První dva znaky.Stačí tedy Psát ST.

STEP (krok)

```
FOR H=10 TO -10 STEP -1.6253 : PRINT H↑2,H : NEXT
```

Tento Příkaz lze Použít jako součást cyklu FOR-TO-NEXT. K Proměnné se u NEXTu Přičte hodnota uvedená za klíčovým slovem STEP (-1.6253). K funkci cyklu však není bezPodmínečně nutný. Není-li uveden, je hodnota kroku rovna jedné.

STOP (stát)

```
A=SQR(X+34*Y)+3:STOP
```

STOP je velmi užitečný Příkaz. NaPíšeme-li ho do Programu,zPůsobí jeho zastavení a vyPíše číslo Ř., kde k zastavení došlo (obdoba klávesy STOP).Hodnoty Proměnných se zachovávají a můžeme s nimi různě maniPulovat. Program znovu sPustíme Příkazem CONT.

STR$ (straight - Přímo)

```
PR=1991:P$="Budeme Psát rok "+STR$(PR)+" !"
```

Pomocí tohoto Příkazu snadno Převedeme číslo (real i integer) do řetězce.Přímo to totiž nejde(P$=PR)!

SYS (system - systém)

POKE251,141:Poke252,0:Poke253,4:Poke254,96:SYS 251

Příkaz pro skok do programu ve strojovém kódu. Je-li zakončen instrukcí RTS, program v Basicu pokračuje dále za SYS.

TAB (tabulate - sestavit do tabulky)

PRINT TAB(9) "1234567890" TAB(21) "0987654321"

Příkaz k posunutí kurzoru o (X) znaků od okraje. X se pohybuje, podobně jako u SPC, od 0 do 255. Není možné přepsat text napsaný tímtéž příkazem PRINT , který obsluhuje TAB.Pokud bychom se o to pokusili, příkaz TAB se nebude brát v úvahu.

TAN (tangent - tangenta)

X=TAN(0)+Y↑(Y+1)

Goniometrická funkce tangens.Argument funkce zadáváme v radiánech (180stupňů odpovídá 3.141592 [pí] rad.).

TIME (čas,doba)

PRINT TIME,TIME$:IF TI>2000 THEN TI=0:PRINT TI,TI$

Standardní proměnné ukazující čas od zapnutí počítače. TI je numerická a po vydělení 60ti ukazuje čas v sekundách. TI$ je řetězcová a ukazuje čas v hodinách,minutách a sekundách.

USR (use - užívat)

POKE785,251:POKE786,0:POKE251,96:PRINT USR(2)

Příkaz,který umožňuje užití programu ve stroj. kódu s přenosem parametrů. Na adresy 785,786 uložíme adresu počátku programu,který chceme volat.

VAL (value - hodnota)

A$="123.456 O.K." : A=VAL(A$) : PRINT A$,A

Funkce,která z řetězce vybere zleva vše co se může vyskytovat v číselném oboru a přiřadí to dané proměnné.Ostatního si nevšímá.

VERIFY (ověřit)

SAVE "PROGRAM":VERIFY "PROGRAM"

Tento příkaz, po uložení dat (např. na magntofon) zkontroluje celý záznam, je-li opravdu v pořádku. Jak test dopadl, to se dozvíme z hlášky,kterou vypíše.

WAIT (čekat)

WAIT 198,1 : GET A$: PRINT A$;

Tento příkaz zastaví běh programu a čeká, až se na dané adrese (198) objeví požadovaná hodnota (1). V jiném případě můžeme za první hodnotu (1) zapsat ještě hodnotu druhou. Obě samozřejmě oddělené čárkami. Vztah mezi těmito hodnotami je EXCLUSIVE OR (výhradní nebo).V uvedeném příkladu se čeká, až bude v klávesnicovém buffru první znak (v podstatě tedy až stiskneme klávesu). Může se ovšem stát, že buffer obsahuje více než jeden znak a program se v tomto místě "zakousne". Proto bychom měli buffer nastavit do výchozí polohy, kdy v něm je 0 znaků. Lze to udělat např. příkazem POKE 198,0.

Tak toto byl poslední příkaz, který C=64 zná. Zdá se vám jich málo?Možná máte trochu pravdu.Chybí příkazy pro grafiku, zvuk apod. Ale všechno má své pro a proti. Není totiž vůbec žádný problém si některé příkazy "přidělat". Ale bylo by to trochu zbytečné, protože už to je hotovo. Stačí sehnat si třeba známý Simon's Basic, který nabízí rozšíření o dalších 114 příkazů a několik standardních proměnných. Ulehčuje odladování programů,práci s grafikou apod.. Bohužel si také ukrojí z paměti pořádný krajíc. Takže pokud byste byli nespokojeni, pořiďte si ho na cartridgi, nebo si uděljte svůj, lepší.

Co se týče příkazů,mám dojem,že by bylo dobré všechny je vyzkoušet, zkrátka osahat si je. Určitě se to vyplatí.

Priority operací

Když zapisujeme nějakou rovnici, je třeba vědět, které početní operace se provedou nejdříve. Uvedu příklad: X = 10↑4/2. Je to X = 10↑(4/2), nebo spíš X = (10↑4)/2 ?Ano,máte pravdu -je to ta druhá možnost.Ale přesto se podívejme na následující tabulku:

Operátor	Význam	Příklad použití
↑	umocňování	základ ↑ exponent X=2↑8 : Y=4↑5
—	negace (změna znaménka)	X=-X :Y=-B : Z=-X
* /	násobení dělení	X=3*2 : Y=125*34 X=10/2 : Y=625/25
+ —	sčítání odečítání	X=3+4 : Y=1+1+3+2 X=10-35.4 : Y=3-1
< = >	Porovnávání	X=3>1 (-1 Pravda) Y=3<=2 (0 lež)
NOT	logická negace	X=NOT3 : Y=NOT-1 (X=(-(X+1))
AND	logický součin	X=255AND64:Y=3AND2
OR	logický součet	X=64OR32 : Y=3OR2

V tabulce tedy máme jednotlivé operátory seřazeny podle priority, tedy podle toho, jaký mají význam. Nejvyšší prioritu má umocňování, nejnižší logický součet.

Myslím,že jsem se ještě nezmiňoval o tom, jak fungují logické operace s čísly. Tak tedy: všechna čísla jsou v počítači zapsána ve dvojkové soustavě a z toho plynou jisté důsledky. Podívejme se tedy, jak s nimi jednotlivé logické operace zacházejí:

NOT (NOT1=0,NOT0=1)

NOT 00001111=11110000 (15dec) (240dec)

NOT 1111111111111110 = 0000000000000001

AND (1AND1=1,0AND1=0,1AND0=0,0AND0=0)

```
      17    dec                          65534
AND  195    dec          AND                27
  00010001  bin              1111111111111110
AND 11000011  bin        AND 0000000000011011
  00000001  bin              0000000000011010
        1   dec                           26
```

OR (1OR1=1,0OR1=1,1OR0=1,0OR0=0)

```
      25    dec                          65533
OR   129    dec          OR                134
  00011001  bin              1111111111111101
OR 10000001 bin          OR  0000000010000110
  10011001  bin              1111111111111111
     153    dec                          65535
```

a ještě si neodpustím EXCLUSIVE OR:

EOR (1EOR1=0,1EOR0=1,0EOR1=1,0EOR0=0)

```
  00011001  bin              1111111111111101
EOR 10000001 bin         EOR 0000000010000110
  10011000  bin              1111111101111011
```

Jednoduché programy v Basicu

Následujících několik stránek bude patřit dvanácti programům v Basicu.Jsou to:

1)Loterie MDN 1988 (aneb jak snadno přijít o peníze). Po spuštění programu zadáme počet losů, které bychom si koupili. Pak postupně zadáváme jejich čísla. Až s tím budeme hotovi, počítač je porovná s čísly, které jsme mohli najít na tažební listině MDN v roce 1988.Porovnávají se nejen kompletní čísla, ale i trojčísla, zkrátka vše, na co šlo něco vyhrát. Výsledek se vypíše na obrazovku.

2)Římská čísla

Tento program převede zadané číslo na tvar s římskými čísly.Zadané číslo se pohybuje od nuly do 3000.

3)Dec => Hex

Za pomoci tohoto programu můžeme snadno převádět čísla z desítkové soustavy do šestnáctkové. Provedeme-li malou úpravu, může program sloužit jako zkoušející. Náhodně vygeneruje decimální číslo,převede ho na hexa a porovná s naší odpovědí.Tuto úpravu lze ještě vylepšit tím, že vypíše hexa tvar a zeptá se na decimální.

4)Zápis a čtení souboru na pásce

Jedná se o ukázkový program, který má za úkol předvést práci se souborem na pásce. Při zápisu ukládáme jednotlivá slova,nebo věty.Pokud budeme chtít zápis zakončit, stiskneme "naprázdno" klávesu <RETURN>.Při čtení souboru nejprve převineme pásku zpět.Jednotlivé řetězce se vypisují vždy po jednom.

5)Třídění probubláváním

Tento program seřadí řetězce podle abecedy metodou "probublávání". Nesmíme zapomínat, že základem pro rozhodování podle abecedy je tabulka kódů ASCII.Nejprve se tedy seřadí řetězce začínající malým písmenem a až potom ty ostatní.

8)Tabulkové výpočty

Klasický program pro základní tabulkové výpočty (5 řádků a 6 sloupců).

7)Řešení lin. rovnic

Jednoduchý program pro řešení soustavy maximálně 49 lin. rovnic.

9)Datařádky

Tento program převede data uložená v paměti do data-řádků, které lze uložit příkazem SAVE.

1 0)Výpis paměti

Program provede výpis obsahu paměti v decimálním tvaru.Vhodné zejména pro tisk.

1 1)Redaktůrek

Krátký textový editor, který se vždy uloží společně s ukládáným souborem.Ukládá se na pásku v Turbu.Pro čtení platí totéž.Umožňuje kódování a dekódování textu.

1 1)Hodiny

Chcete vědět při práci s počítačem kolik je hodin?Použijte tento program!V datech je rozepsán program ve stroj. kódu.

1 2)Bleskové třídění

Pokud se vám zdá třídění v Basicu pomalé, použijte služeb tohoto programu. Za půl vteřiny seřadí podle abecedy 300 řetězců! V datech je opět rozepsán program ve stroj. kódu.

Všechny programy jsou ověřené a samozřejmě funkční.K výpisům ještě dodávám,že uvozovky v rozdělených příkazech PRINT,INPUT apod.na dalším řádku (pod číslem řádku) nejsou součástí programu,ale signalizací, že uvozovky pokračují z předchozího řádku.Co se týče inverzních znaků, můžeme je zapisovat buď tak jak je vidíme (s tím jsou však trochu obtíže), nebo zjistit,jaká klávesa je "dělá" - zpravidla se jedná o CLR,HOME,CRSR.

Programy nejsou určeny jen pro slepé okopírování, ale především na "rozebírání". Pomůže nám to snadněji proniknout do tajů Basicu.

Loterie MON 1988 (aneb jak snadno Prijit o Penize)

```
10 rem loterie mon 1988
15 dim a$(30):clr
20 inPut"xxxxxxx zadej Pocet losu";n
26 Print:Print
27 if n>1 then Print"bude vas to stat",n*3,"Kcs"
28 if n=1 then Print"bude te to stat 3 Kcs"
29 Print:Print
30 for i=1 to n
40 inPut "zadej cislo losu (8 cifer)";a$(i)
50 next i
60 P=94:Print:Print
70 for K=1to P
80 readr$
90 for i=1ton
100 a=len(r$)
110 if r$=mid$(a$(i),8-a,a) then Print"toto cislo
"    je zde",a$(i),r$:m=1
120 next i:next K
125 if m=0 then Print"*  tentokrat to newyslo.snad
"     Priste... *"
130 data3279169,3908230,5922487,5500738,2224417,26
    23731,3685870
140 data2054278,4272620,2705062,3063066,3457446,49
    57978,3156566
150 data3414574,4290604,4569124,2504460,3967349,44
    59305,6304845
160 data1881287,2022171,4982696,6331549,1637310,17
    22553,2365638
170 data3915340,5261387,5283645,197499,630583,9183
    70,47533,85889
180 data27843,71137,17077,78208,03697,84158,02738,
    80200,87255,92499
190 data53854,9852,3633,7844,3604038,3145507,63172
    23,1669148,2165216
200 data2827759,5934806,6193633,6380896,2231641,23
    86042,6082393
210 data5497819,5515103,155254,632168,634046,64547
    4,651814,872551,913357
220 data989603,42690,46700,63851,68595,24463,25727
    ,45421,61570
230 data01829,42926,51693,61721,98066,20355,43285,
    65251,88370
240 data2862,7718,0284,7259,000000
```

Pozn.

Rimska cisla

```
10 rem rimska cisla
20 print"■■■■ ";chr$(14);
30 input"z=";z
40 b=z:print"■■■■■■■■■■■■■■■";
50 ifb>3000 then print"Prilis velke.":wait198,1:ge
   ta$:goto20
60 f=1000:a$="M":gosub500
70 f=900:a$="CM":gosub500
80 f=500:a$="D":gosub500
90 f=400:a$="CD":gosub500
100 f=100:a$="C":gosub500
110 f=90:a$="XC":gosub500
120 f=50:a$="L":gosub500
130 f=40:a$="XL":gosub500
140 f=10:a$="X":gosub500
150 f=9:a$="IX":gosub500
160 f=5:a$="V":gosub500
170 f=4:a$="IV":gosub500
180 f=1:a$="I":gosub500
190 print"■■■■":end
500 b=int(z/f)
510 ifb<>0thengosub1000
520 z=z-b*f
530 return
1000 fori=1tob
1010 printa$;
1020 nexti
1030 return
```

Pozn.

Dec => Hex

```
5 rem Prevod z decimalni soust. do sestnactkove
10 clr:input"[REVERSE]  dec:";d
15 ifd<0ord>65535then10
20 l=1:a$="":ifd<256thena$="   ":l=3
30 d2=int(d/256):d1=d-d2*256
40 a=d2:a$="":gosub100
50 a=d1:gosub100
60 print"[RVS]  hex:   $";mid$(a$,l,4)
70 wait198,1:geta$:ifa$<>"e"then10
80 end
100 a2=int(a/16):a1=a-a2*16
110 d=a2:gosub150
120 d=a1:gosub150
130 return
150 ifd<10thena$=a$+mid$(str$(d),2,2)
160 ifd=10thena$=a$+"a"
170 ifd=11thena$=a$+"b"
180 ifd=12thena$=a$+"c"
190 ifd=13thena$=a$+"d"
200 ifd=14thena$=a$+"e"
210 ifd=15thena$=a$+"f"
220 return
```

Pozn.

Zapis a cteni souboru na Pasce

```
10 rem zapis a cteni souboru na Pasce
20 Print"■■■■ <Z>apis / <C>teni■■■■";chr$(14)
25 Print"                    ";
30 Poke204,0
40 Geta$:Poke204,1
50 if a$="z"then100
60 if a$="c"then200
70 Goto30
100 Gosub1000
110 oPen1,1,1,a$
120 b$="":inPut"Do souboru zapiseme";b$
130 if len(b$)=0 then Print#1,"@":close1:end
140 Print#1,b$
150 Print:Goto120
200 Gosub1000
210 oPen1,1,0,a$:Print"■"
220 inPut#1,b$
230 if b$="@" then close1:end
240 Print b$:Poke198,0:wait198,1
250 Goto220
1000 inPut"■■■■ Jmeno souboru";a$
1010 if len(a$)>16 then 1000
1020 return
```

Pozn.

Trideni Probublavanim

```
10 rem *** trideni retezcu "Probublavanim" ***
15 Print chr$(14)
20 inPut"▒▒▒▒  Pocet retezcu";an
30 dim a$(an)
40 Print"▒"
50 for i=1 to an:Print i;:inPut" ";a$(i)
60 next
70 rem bublani
100 for i=2 to an
110 for j=an to i steP-1
120 if a$(j-1) < a$(j) then160
130 x$=a$(j-1)
140 a$(j-1)=a$(j)
150 a$(j)=x$
160 next j,i
200 rem tisk serazenych retezcu
210 Print"▒"
220 for i=1 to an
230 Poke198,0
240 Print i;a$(i)
250 wait198,1:Get x$
260 next:end
```

Pozn.

```
10 rem tabulkove vypocty
20 dim aa(6,5):clr
30 print"■";chr$(142):rem tisk tabuky
40 print"  ┌────┬────┬────┬────┬────┬────┬────┬──────┐"
42 print" ■ p1 ■ p2 ■ p3 ■ p4 ■ p5 ■ p6 ■ suma ■"
45 print"  ├────┼────┼────┼────┼────┼────┼────┼──────┤"
49 print" ■    ■    ■    ■    ■    ■    ■      ■"
50 print" ■    ■    ■    ■    ■    ■    ■      ■"
52 print" ■    ■    ■    ■    ■    ■    ■      ■"
55 print" ■    ■    ■    ■    ■    ■    ■      ■"
58 print" ■    ■    ■    ■    ■    ■    ■      ■"
60 print" ■    ■    ■    ■    ■    ■    ■      ■"
62 print" ■    ■    ■    ■    ■    ■    ■      ■"
65 print" ■    ■    ■    ■    ■    ■    ■      ■"
68 print" ■    ■    ■    ■    ■    ■    ■      ■"
70 print" ■    ■    ■    ■    ■    ■    ■      ■"
72 print" ■    ■    ■    ■    ■    ■    ■      ■"
75 print" ■    ■    ■    ■    ■    ■    ■      ■"
78 print" ■    ■    ■    ■    ■    ■    ■      ■"
80 print" ■    ■    ■    ■    ■    ■    ■      ■"
82 print" ■    ■    ■    ■    ■    ■    ■      ■"
85 print" ■    ■    ■    ■    ■    ■    ■      ■"
88 print" ■    ■    ■    ■    ■    ■    ■      ■"
90 print"  ├────┼────┼────┼────┼────┼────┼────┼──────┤"
92 print" ■    ■    ■    ■    ■    ■    ■      ■"
95 print"  └────┴────┴────┴────┴────┴────┴────┴──────┘"
98 if a>0then130:rem tiskne-li se prvne, a=0
100 restore:rem vyplneni tabulky cisly
110 forb=0to4:forc=0to5
120 read a:gosub1000:nextc,b
125 restore:goto30:rem novy tisk tabulky
130 for b=0to4:forc=0to5:rem soucet,tisk hodnot
135 reada:gosub2000
140 aa(6,b)=aa(6,b)+aa(c,b)
150 next:a$="      "+str$(aa(6,b)):print"■■■";right$
    (a$,4):next
155 Print"■■■":Print"■";:rem soucet sloupcu+tisk
160 for c=0to5:forb=0to4
170 aa(c,5)=aa(c,5)+aa(c,b)
180 next:a$="      "+str$(aa(c,5)):print"■";right$(a
    $,4);:next
190 Print"■■■bk?";:rem ceKani,doKud nenitab.oPsana
200 Poke198,0:Poke204,0:wait198,1
210 Print"■":end:rem Konec Programu
490 rem data Pro nastaveni Kurzoru v tabulce
500 data 242,247,252,257,262,267
510 data 362,367,372,377,382,387
520 data 482,487,492,497,502,507
530 data 602,607,612,617,622,627
540 data 722,727,732,737,742,747
990 rem PodProgram Pro nastaveni Kurzoru v tab.
999 rem a Precteni hodnoty z Klavesnice
1000 Print"■";
1010 for x=0toa:print"■";
1020 next:b$="":a$=""
```

```
1030 Poke204,0:Poke198,0
1040 Geta$:ifa$=""then1040
1045 Poke204,1
1050 if  len(b$)>=3 and a$<>chr$(13)then1000
1060 if  a$=chr$(13) then aa(c,b)=val(b$):Print"B";
     :return
1070 Printa$;:b$=b$+a$
1080 Goto1030
1999 rem PodProGram Pro Pretisk hodnot tabulky
2000 Print"■";
2010 forx=0toa:Print"■";:next
2020 a$="    "+str$(aa(c,b)):Print right$(a$,3);
2030 return
```

Pozn.

```
2 rem reseni soust. lin. rovnic
5 dimx(50,50),y(50),x$(50)
10 print"■"
12 print"■ Rovnice ma tvar a(x)+a(x-1)..a(1)=a(0)■
"   "
15 input" Pocet rovnic";n:print
18 ifn>49then10
20 forii=1ton
30 forjj=1ton+1
40 print" a(";ii;",";jj;")=";:inputx(ii,jj)
50 nextjj
55 print"--------------------"
60 nextii
62 gosub1000
63 if a<>0then print" Mnoho reseni":end
65 forii=1ton
70 readx$(ii)
80 nextii
90 print"vYpis PostuPu  (a/n)"
100 getl$
110 ifl$="a"thenPrint"stiskni klavesu":goto140
120 ifl$="n"thenPrint"Pocitam":z=1:goto140
130 goto100
140 forKK=1ton-1
150 forll=KK+1ton
160 ifz=1then270
165 getl$:ifl$=""then165
170 forii=1ton
180 forjj=1ton-1
190 ifx(ii,jj)<>0thenPrintx(ii,jj);"*";x$(jj);"+";
200 nextjj
210 ifx(ii,n)<>0thenPrintx(ii,n);"*";x$(n);
215 Print"=";x(ii,n+1);
220 if(ii=KK)thenPrint"    /*";x(ll,KK);"+";ll;
230 if(ii=ll)thenPrint"    /*";-x(KK,KK);
240 Print
250 nextii
260 Print"-----------"
270 formm=n+1toKstep-1
280 x(ll,mm)=x(ll,mm)*x(KK,KK)-x(KK,mm)*x(ll,KK)
287 nextmm,ll,KK
289 ifz=1then300
290 getl$:ifl$=""then290
291 forii=1ton
292 forjj=1ton-1
293 ifx(ii,jj)<>0thenPrintx(ii,jj);"*";x$(jj);"+";
294 nextjj
295 ifx(ii,n)<>0thenPrintx(ii,n);"*";x$(n);
296 Print"=";x(ii,n+1)
297 nextii
298 Print"-----------"
300 forii=nto1step-1
310 ifii=nthen350
320 forjj=ii+1ton
330 x(ii,n+1)=x(ii,n+1)-x(ii,jj)*y(jj)
```

```
340 nextjj
350 y(ii)=x(ii,n+1)/x(ii,ii)
360 nextii
370 forii=1ton
380 printx$(ii);"=";y(ii)
390 nextii:end
500 dataa,b,c,d,e,f,g,h,i,j,K,l,m,n,o,P,q,r,s,t,u,
    v,w,x,y,z
1000  ifn<2then1050
1005 fori=1ton-1:a=1
1010 forj=1ton+1
1020 ifx(i,j)<>x(i+1,j)thena=0
1030 next j,i
1050 return
```

Pozn.

```
2 rem tvorba dataradku
5 clr:print"■"
8 poke51,159:poke52,15:poke55,159:poke56,15
10 input"data umistime do pemeti od adresy jake";p
   o
15 if (po>65535)or(po<4000)then go to5
20 print:pokepo,0:a=po+1
25 print "■"
30 input"odkud budeme cist data";o
40 ifo<0then 3oto30:print
50 input"kde data konci";k
60 if k<othen goto25
70 for ii=1toint((k-o)/18)
80 b=a+78
90 pokea,b-256*int(b/256)
100 pokea+1,int(b/256)
110 c=int(ii/256):d=ii-256*c
120 pokea+2,d
130 pokea+3,c
140 pokea+4,131
150 gosub300
160 a=a+78
170 next ii
180 pokea,0:pokea+1,0
190 print"■prenos dat je skoncen■"
200 c=int((po+1)/256):d=po+1-256*c
210 e=int((a+1)/256):f=a-256*e+1
220 poke43,d:poke44,c:poke45,f:poke46,e
230 end
300 for x=1to18
305 kde=o+18*ii+x-19
310 a$=right$("    "+str$(peek(kde)),3)
320 pokea+4*x+1,asc(left$(a$,1))
330 pokea+4*x+2,asc(mid$(a$,2,1))
340 pokea+4*x+3,asc(right$(a$,1))
350 pokea+4*x+4,44
360 next x
370 pokea+76,32:pokea+77,0
380 return
```

Pozn.

Výpis Pameti

```
rem vypis pameti v decimalnim tvaru
5 printchr$(14):poke657,128:poke55,159:poke56,15
10 print"■■■ Vytvorime predlohu pro tisK dat (dec)
"    "
20 input"■■■ Od jake adresy umistime vypis";po
25 ifpo<4000orpo>65535thenprint"■■ chyba.":wait198,1
   :getz$:goto10
30 input"■■■ Kde data zacinaji a Konci";o,K
35 ifo>kork>65535thenprint"■■ chyba.":wait198,1:getz
   $:goto30
50 print "■■■■■ ■■ okay◆":pokepo,0:a=po+1:od=o
60 forii=1toint((k-o)/14):b=a+61
70 pokea,b-256*int(b/256):pokea+1,int(b/256)
80 pokea+3,int(od/256):pokea+2,od-256*peek(a+3)
90 gosub150:od=od+14
100 a=a+61:nextii
105 pokea,0:pokea+1,0
110 print"■■■■■■ Mas pripraveny vypis pro tisk T 75
"    L.◆"
120 c=int((po+1)/256):d=po+1-256*c:e=int((a+1)/256
    ):f=a-256*e+1
130 poke1020,d:poke1021,c:poke1022,f:poke1023,e
140 poke43,peek(1020):poke44,peek(1021):poke45,pee
k(1022):poke46,peek(1023):poke55,0:poke56,160:end
150 forx=1to14
160 kde=o+14*ii+x-15
170 a$=right$("    "+str$(peek(kde)),3)
180 pokea+4*x,asc(left$(a$,1))
190 pokea+4*x+1,asc(mid$(a$,2,1))
200 pokea+4*x+2,asc(right$(a$,1))
210 pokea+4*x+3,32
220 nextx:pokea+60,0:return
230 from john's workshop
```

Pozn.

```
2 poke2124,133:poke2125,08:rem"
                  "redakturek      "
10 print"■■■■",chr$(14):poke657,128:poke53280,5:pok
   e53281,9:poke646,7:gosub1000
15 poke1020,peek(45):poke1021,peek(46):poke46,17:p
   oke55,0:poke56,20:poke2185,143
20 clr:po=peek(1020)+256*peek(1021):ifpo<5120thenp
   oke1020,0:poke1021,20:goto20
30 print"_■";:a$="":ifpo>40850thenprint" Plna pame
"  t!":wait198,1:geta$:goto1100
35 getb$:ifb$=chr$(133)then1100
36 ifb$=chr$(134)then30
37 ifb$=chr$(136)thenpo=po-1:print"■";
38 a$=a$+b$:printb$;:if(b$<>chr$(13))and(len(a$)<2
   50)then35
40 d=len(a$)
50 forx=1tod:po=po+1:pokepo,asc(mid$(a$,x,1)):ifpe
   ek(po)=192thenpokepo,44
60 next:goto30
100 print"■■■■":gosub1000:input" Jmeno";a$:a$=a$+"
"                          "
110 a$=left$(a$,16):forx=1to16:a=asc(mid$(a$,x,1))
    :poke2105+x,a:poke2814+x,a
115 next:print"■■":b=int(po/256):a=po-256*b+1:poke
    1022,a:poke1023,b:pokepo+1,32
118 pokepo+2,32
120 poke46,peek(1023):poke45,peek(1022):poke2124,0
    :poke2185,151:poke56,159
125 poke2125,0:sys52045,"redakturek      ":poke46,
    17:poke2124,133:poke2125,8
130 poke56,20:print"■■■■ Budes psat novy text?":p
    oke2185,143:clr
140 geta$:ifa$=""then140
145 po=peek(1020)+256*peek(1021)
150 ifa$="n"thenprint"■■■■":gosub1000:goto20
160 po=0:goto10
200 print"■■■■":gosub1000:poke631,95:poke632,76:po
    ke633,13:poke198,3:goto600
300 print"■■■■":gosub1000:input" Kod";a$:ifa$=""th
    en300
310 c=len(a$):a=1
320 forx=peek(56)*256+1topo:ifa>cthena=1
330 b=peek(x)+asc(mid$(a$,a,1)):ifb>255thenb=b-256
340 pokex,b:a=a+1:next
350 print" Okay,pokracuj":goto30
400 print"■■■■":gosub1000:input" Kod";a$:ifa$=""th
    en400
410 c=len(a$):a=1
420 forx=peek(56)*256+1topo:ifa>cthena=1
430 b=peek(x)-asc(mid$(a$,a,1)):ifb<0thenb=b+256
440 pokex,b:a=a+1:next:goto350
500 print"■■■■":gosub1000:ifpo<5121then30
505 forx=5121topo:printchr$(peek(x));:geta$
510 ifx/100=int(x/100)thenifa$=chr$(13)then30
515 ifa$="e"thenx=po
```

```
518 ifa$="c"thenx=x+200
520 next:goto30
600 poke56,159:print"":poke2124,0:poke2125,0:end
1000 restore:forx=0to39:reada,b:poke1064+x,a:poke5
     5336+x,b:next
1003 poke1021,int(Po/256):poke1020,Po-256*peek(102
     1)
1005 printchr$(14);"";:return
1010 data32,241,97,1,97,1,97,1,117,1,117,241,116,1
     ,101,1,101,145,160,13,198,13
1020 data146,13,143,13,141,13,160,13,202,13,143,13
     ,136,13,142,13,167,13,147,13
1030 data160,13,215,13,143,13,146,157,139,13,147,1
     3,136,13,143,13,144,13,160,13
1040 data103,1,103,17,106,1,118,1,118,1,225,1,225,
     241,225,1,32,7
1100 print"":gosub1000:print" Konec":print"
"     Zapis textu na pasku"
1110 print" Cteni textu z pasky":print" Kodo
"     vani":print" Dekodovani"
1120 print" Cteni":poke1944,62:a=1944:poke198,0
1130 wait198,1:geta$:ifa$=chr$(13)then1160
1140 pokea,32:a=a-160:ifa<1100thena=1944
1150 pokea,62:goto1130
1160 r=int(a/100)-10:onrgoto600,10,100,200,10,300,
     400,10,500
```

!Pozor!
Pri prepisovani tohoto programu je nezbytne nutne dodrzovat presny pocet znaku (mezer apod.).Program totiz meni sam sebe a mohlo by dojit ke kolizi neli ke zhrouceni!Tak tedy jeste jednou POZOR!

Pozn.

Hodiny

```
99 rem hodiny
100 fori=51200to51481
110 readx:pokei,x:s=s+x:next
120 data173,14,220,9,128,141,14,220,173,15,220,41
130 data127,141,15,220,32,121,0,240,101,32,253,174
140 data32,158,173,32,163,182,201,6,208,107,160,0
150 data177,34,56,233,48,201,3,176,96,10,10,10
160 data10,133,251,200,177,34,56,233,48,201,10,176
170 data80,5,251,208,4,169,146,208,15,201,36,176
180 data68,201,19,144,7,56,248,233,18,216,9,128
190 data141,11,220,32,253,200,141,10,220,32,253,20
    0
200 data141,9,220,169,0,141,8,220,32,121,0,240
210 data13,32,253,174,32,158,183,224,16,176,22,142
220 data167,2,120,173,20,3,73,161,141,20,3,173
230 data21,3,73,34,141,21,3,88,96,76,72,178
240 data165,251,72,165,252,72,173,136,2,133,252,16
    9
250 data0,133,251,160,30,173,11,220,201,18,240,17
260 data201,128,144,15,41,127,201,18,240,9,248,24
270 data105,18,216,208,2,169,0,32,219,200,173,10
280 data220,32,219,200,173,9,220,32,219,200,173,8
290 data220,9,48,32,243,200,104,133,252,104,133,25
    1
300 data76,49,234,72,41,240,74,74,74,74,24,105
310 data48,32,243,200,104,41,15,24,105,48,32,243
320 data200,169,58,145,251,173,167,2,153,0,216,200
330 data96,200,177,34,56,233,48,201,6,176,134,10
340 data10,10,10,133,251,200,177,34,56,233,48,201
350 data10,176,238,5,251,96
360 ifs<>32970thenprint"chyba v datech !!":end
370 Print"oK"
390 Print"nasleduje prikaz:sys51200,''hhmmss'',cis
"     lo barvy"
400 Print:Print:Print:Print:
410 Print"sys51200"
420 end
```

Pozn.

BleskOve trideni

```
5 rem bleskove trideni retezcu podle abecedy
9 print"■■prosim chvilku pockej..."
10  i=49152:sum=0
20 read a:ifa=256 then40
30 sum=sum+a:poke i,a:i=i+1:goto 20
40 ifsum<>45295thenprint"chyba v data-radcich":end
50 print"■bleskove trideni je pripraveno":rem kone
   c
100 rem priklad pouziti
110 n=300
120 dimaa$(n)
130 print"vytvarim"n" nahodnych retezcu"
140 sd=-ti:a=rnd(sd)
150 fori=1ton
160 printi"■"
170 n1=int(rnd(1)*10+1)
180 a$=""
190 for j=1ton1
200 b$=chr$(int(rnd(1)*26+65))
210 a$=a$+b$
220 next j
230 aa$(i)=a$
240 next i:rem generovani je dokonceno
250 print"stiskem klavesy odstartuj trideni podle
"   abecedy"
260 geta$:ifa$=""then 260
270 print"tridim..."
280 t1=ti
291 rem sys 49152,n,aa$(1)
300 sys 49152,n,aa$(1)
310 t2=ti
320 print"hotovo"
330 print"stiskem klavesy odstartuj tisk retezcu"
340 get a$:if a$=""then340
350 for i=1 to n:print i,aa$(i):next
360 print:print n" retezcu jsem seradil za"(t2-t1)
    /60"sekund"
10000 rem data programu ve stroj. kodu
49152 data 32,253,174,32,158,173
49158 data 32,247,183,165,20,133
49164 data 253,165,21,133,254,32
49170 data 253,174,32,158,173,162
49176 data 1,165,71,157,85,193
49182 data 157,125,193,165,72,157
49188 data 105,193,157,145,193,165
49194 data 253,208,2,198,254,198
49200 data 253,160,3,24,189,125
49206 data 193,101,253,157,125,193
49212 data 189,145,193,101,254,157
49218 data 145,193,136,208,236,189
49224 data 85,193,133,80,189,105
49230 data 193,133,81,189,125,193
49236 data 133,82,189,145,193,133
49242 data 83,32,21,193,144,4
49248 data 202,208,228,96,165,82
```

```
49254 data 133,78,165,83,133,79
49260 data 160,2,177,78,153,250
49266 data 0,136,16,248,48,11
49272 data 24,165,80,105,3,133
49278 data 80,144,2,230,81,160
49284 data 2,177,80,153,247,0
49290 data 136,16,248,32,32,193
49296 data 144,230,56,165,82,233
49302 data 3,133,82,176,2,198
49308 data 83,32,21,193,176,31
49314 data 160,2,177,82,153,247
49320 data 0,136,16,248,32,32
49326 data 193,176,225,160,2,177
49332 data 80,145,82,185,247,0
49338 data 145,80,136,16,244,48
49344 data 183,160,2,177,80,145
49350 data 79,185,250,0,145,80
49356 data 136,16,244,24,189,85
49362 data 193,125,125,193,133,82
49368 data 189,105,193,125,145,193
49374 data 133,83,102,83,102,82
49380 data 32,21,193,176,22,189
49386 data 85,193,157,86,193,189
49392 data 105,193,157,106,193,32
49398 data 53,193,232,32,69,193
49404 data 76,71,192,189,125,193
49410 data 157,126,193,189,145,193
49416 data 157,146,193,32,69,193
49422 data 232,32,53,193,76,71
49428 data 192,165,81,197,83,208
49434 data 4,165,80,197,82,96
49440 data 160,255,200,196,247,176
49446 data 11,196,250,176,6,177
49452 data 248,209,251,240,241,96
49458 data 196,250,96,24,165,80
49464 data 105,3,157,85,193,165
49470 data 81,105,0,157,105,193
49476 data 96,56,165,80,233,3
49482 data 157,125,193,165,81,233
49488 data 0,157,145,193,96,256
```

Pozn.

KAPITOLA III

Je zcela zbytečné,abych chtěl
a ukláděl svůj rozum,
k tomu mi stačí kolega.
(Michel de Montaigne, 1954 - 1961)

Zdravý rozum je sbírka předsudků
nashromážděných ve věku osmnácti let.
(Albert Einstein, 1967)

Všechny trampoty jsou z toho,
že hlupáci jsou si jisti sami sebou,
zatímco rozumní lidé
jsou samá pochybnost .
(Bertrand Russel, 1872 - 1970)

1)Základní technické Pojmy

2)Systémové Proměnné

3)Uložení Basicu a Proměnných

4)Organizace Paměti

5)Obraz

6)Zvuk

7)Příklady Programů na obsluhu obrazu a zvuku

Základní technické pojmy

Pokud se chceme blíže podívat na paměť naší Commodorky,musíme si ujasnit některé pojmy.

Nejprve k té paměti.Paměti si můžeme rozdělit do dvou skupin -RWM a ROM.Označení RWM znamená, že paměti tohoto typu jsou určeny pro zápis a čtení informace uživatelem. V praxi se pro tento typ paměti užil, poněkud nesprávně, název RAM. Paměti typu ROM jsou určeny pouze pro čtení jejich obsahu.Jsou totiž "natvrdo" naprogramovány již při výrobě.

Protože informací v paměti bývá velké množství,je třeba dostat se nějakým způsobem k té,kterou potřebujeme. K tomu nám stačí znát adresu té paměťové buňky, která nás zajímá. Data jsou uložena v paměti po jednotlivých slovech - Bytech (bajtech) a to ve dvojkové soustavě. Každý Byte se skládá z osmi bitů (náš Počítač je osmibitový).Podívejme se na obrázek:

Adresa Bytu	0	76543210
Adresa Bytu	1	76543210
Adresa Bytu	2	76543210
⋮	⋮	⋮
Adresa Bytu	65535	76543210

(Každý Byte obsahuje 7 bitů.Bit může mít hodnotu 0,nebo 1.Má-li hodnotu 0,říkáme, že je vypnutý,1-zapnutý. Počet možných kombinací v Bytu je 2^8-1(Počítáme od 0)). První buňka paměti má adresu 0, Poslední 65535 (to je $2^{16}-1$. Adresa se tedy skládá ze dvou Bytů -nižšího (Low) a vyššího (High). Tomu nižšímu říkáme adresa ve stránce a tomu vyššímu stránka. K dispozici tedy máme 256 stránek (0-255) a Paměť má kapacitu 65536 Bytů (0-65535), tedy 64 KB (KB=2^{10} B = =1024 Bytů,ale kB=1000 Bytů).

Zkusme se nyní zamyslet nad tím, jak bychom mohli rozložit adresu do jednotlivých Bytů. Pokud máme adresu menší,než 256 (tedy v nulové stránce), je to jednoduché. Ale mějme třeba adresu 2049. Jak na to?Velice jednoduše.V Basicu bychom to zapsali:

HI = INT(ADRESA/256) : LO = ADRESA-256*HI

Některé obvody mají v Paměti vyhrazena svá místa (naPř. zvukový). Na tato místa ukládáme informace,které si Přečte a Podle jejich obsahu dále zaměřuje svou činnost. Anebo je tam naoPak uloží a je na nás,jak s nimi naložíme. Těmto místům říkáme registry.NaPř. již zmíněný zvukový obvod má vyhrazenu část Paměti od adresy 54272 ($D400) a na této adrese leží jeho nulty registr.Poslední registr má číslo 28, leží tedy na adrese 54300 (54272+28). Do některých registrů zaPisujeme Parametry zvuku (výška tónu,hlasitost aPod.),z jiných registrů můžeme naoPak cosi vyčíst.

A nakonec ještě Pro ty,kdo stále nevěří,že na osmi bitech může nastat Právě 256 kombinací,uvádím část z nich.Ostatní je snadné si odvodit (sledujte jak se mění 1 a 0 ve sloupcích).

Komb.	Hodnota	Hex tvar	76543210
1.	0	$00	00000000
2.	1	$01	00000001
3.	2	$02	00000010
4.	3	$03	00000011
5.	4	$04	00000100
6.	5	$05	00000101
7.	6	$06	00000110
8.	7	$07	00000111
9.	8	$08	00001000
10.	9	$09	00001001
11.	10	$0A	00001010
12.	11	$0B	00001011
13.	12	$0C	00001100
14.	13	$0D	00001101
15.	14	$0E	00001110
16.	15	$0F	00001111
17.	16	$10	00010000
18.	17	$11	00010001
19.	18	$12	00010010
20.	19	$13	00010011
21.	20	$14	00010100
22.	21	$15	00010101
23.	22	$16	00010110
24.	23	$17	00010111
25.	24	$18	00011000
26.	25	$19	00011001
27.	26	$1A	00011010
28.	27	$1B	00011011
29.	28	$1C	00011100
30.	29	$1D	00011101
31.	30	$1E	00011110
32.	31	$1F	00011111
33.	32	$20	00100000
34.	33	$21	00100001
35.	34	$22	00100010
36.	35	$23	00100011
37.	36	$24	00100100
38.	37	$25	00100101
39.	38	$26	00100110
40.	39	$27	00100111
41.	40	$28	00101000
42.	41	$29	00101001
43.	42	$2A	00101010
44.	43	$2B	00101011
45.	44	$2C	00101100
46.	45	$2D	00101101
47.	46	$2E	00101110
48.	47	$2F	00101111
49.	48	$30	00110000
50.	49	$31	00110001
51.	50	$32	00110010
52.	51	$33	00110011
53.	52	$34	00110100
54.	53	$35	00110101
55.	54	$36	00110110
56.	55	$37	00110111
57.	56	$38	00111000
58.	57	$39	00111001
59.	58	$3A	00111010
60.	59	$3B	00111011
61.	60	$3C	00111100
62.	61	$3D	00111101
63.	62	$3E	00111110
64.	63	$3F	00111111
65.	64	$40	01000000
66.	65	$41	01000001
67.	66	$42	01000010
68.	67	$43	01000011
69.	68	$44	01000100
70.	69	$45	01000101
71.	70	$46	01000110
72.	71	$47	01000111
73.	72	$48	01001000
74.	73	$49	01001001
75.	74	$4A	01001010
76.	75	$4B	01001011
77.	76	$4C	01001100
78.	77	$4D	01001101
79.	78	$4E	01001110
80.	79	$4F	01001111
81.	80	$50	01010000
82.	81	$51	01010001
83.	82	$52	01010010
84.	83	$53	01010011
85.	84	$54	01010100
86.	85	$55	01010101
87.	86	$56	01010110
88.	87	$57	01010111
89.	88	$58	01011000
90.	89	$59	01011001
91.	90	$5A	01011010
92.	91	$5B	01011011
93.	92	$5C	01011100
94.	93	$5D	01011101
95.	94	$5E	01011110
96.	95	$5F	01011111
97.	96	$60	01100000
98.	97	$61	01100001
99.	98	$62	01100010
100.	99	$63	01100011
101.	100	$64	01100100
102.	101	$65	01100101
103.	102	$66	01100110
104.	103	$67	01100111
105.	104	$68	01101000
106.	105	$69	01101001
107.	106	$6A	01101010
108.	107	$6B	01101011
109.	108	$6C	01101100
110.	109	$6D	01101101
111.	110	$6E	01101110
112.	111	$6F	01101111
113.	112	$70	01110000
114.	113	$71	01110001
115.	114	$72	01110010
116.	115	$73	01110011
117.	116	$74	01110100
118.	117	$75	01110101
119.	118	$76	01110110
120.	119	$77	01110111
121.	120	$78	01111000
122.	121	$79	01111001
123.	122	$7A	01111010
124.	123	$7B	01111011
125.	124	$7C	01111100
126.	125	$7D	01111101
127.	126	$7E	01111110
128.	127	$7F	01111111
129.	128	$80	10000000
130.	129	$81	10000001
131.	130	$82	10000010
132.	131	$83	10000011
133.	132	$84	10000100
134.	133	$85	10000101
135.	134	$86	10000110
136.	135	$87	10000111
137.	136	$88	10001000
138.	137	$89	10001001
139.	138	$8A	10001010
140.	139	$8B	10001011
141.	140	$8C	10001100
142.	141	$8D	10001101
143.	142	$8E	10001110
144.	143	$8F	10001111
145.	144	$90	10010000
146.	145	$91	10010001
147.	146	$92	10010010
148.	147	$93	10010011
149.	148	$94	10010100
150.	149	$95	10010101
151.	150	$96	10010110
152.	151	$97	10010111
153.	152	$98	10011000
154.	153	$99	10011001
155.	154	$9A	10011010
156.	155	$9B	10011011
157.	156	$9C	10011100
158.	157	$9D	10011101
159.	158	$9E	10011110
160.	159	$9F	10011111
161.	160	$A0	10100000
162.	161	$A1	10100001
163.	162	$A2	10100010
164.	163	$A3	10100011
165.	164	$A4	10100100
166.	165	$A5	10100101
167.	166	$A6	10100110
168.	167	$A7	10100111
169.	168	$A8	10101000
170.	169	$A9	10101001
171.	170	$AA	10101010
172.	171	$AB	10101011
173.	172	$AC	10101100
174.	173	$AD	10101101
175.	174	$AE	10101110
176.	175	$AF	10101111
177.	176	$B0	10110000
178.	177	$B1	10110001
179.	178	$B2	10110010
180.	179	$B3	10110011
181.	180	$B4	10110100
182.	181	$B5	10110101
183.	182	$B6	10110110
184.	183	$B7	10110111
185.	184	$B8	10111000
186.	185	$B9	10111001
187.	186	$BA	10111010
188.	187	$BB	10111011
189.	188	$BC	10111100
190.	189	$BD	10111101
191.	190	$BE	10111110
192.	191	$BF	10111111
193.	192	$C0	11000000
194.	193	$C1	11000001
195.	194	$C2	11000010
196.	195	$C3	11000011
197.	196	$C4	11000100
198.	197	$C5	11000101
199.	198	$C6	11000110
200.	199	$C7	11000111
201.	200	$C8	11001000
202.	201	$C9	11001001
203.	202	$CA	11001010
204.	203	$CB	11001011
205.	204	$CC	11001100
206.	205	$CD	11001101
207.	206	$CE	11001110
208.	207	$CF	11001111
209.	208	$D0	11010000
210.	209	$D1	11010001
211.	210	$D2	11010010
212.	211	$D3	11010011
213.	212	$D4	11010100
214.	213	$D5	11010101
215.	214	$D6	11010110
216.	215	$D7	11010111
217.	216	$D8	11011000
218.	217	$D9	11011001
219.	218	$DA	11011010
220.	219	$DB	11011011
221.	220	$DC	11011100
222.	221	$DD	11011101
223.	222	$DE	11011110
224.	223	$DF	11011111
225.	224	$E0	11100000
226.	225	$E1	11100001
227.	226	$E2	11100010
228.	227	$E3	11100011
229.	228	$E4	11100100
230.	229	$E5	11100101
231.	230	$E6	11100110
232.	231	$E7	11100111
233.	232	$E8	11101000
234.	233	$E9	11101001
235.	234	$EA	11101010
236.	235	$EB	11101011
237.	236	$EC	11101100
238.	237	$ED	11101101
239.	238	$EE	11101110
240.	239	$EF	11101111
241.	240	$F0	11110000
242.	241	$F1	11110001
243.	242	$F2	11110010
244.	243	$F3	11110011
245.	244	$F4	11110100
246.	245	$F5	11110101
247.	246	$F6	11110110
248.	247	$F7	11110111
249.	248	$F8	11111000
250.	249	$F9	11111001
251.	250	$FA	11111010
252.	251	$FB	11111011
253.	252	$FC	11111100
254.	253	$FD	11111101
255.	254	$FE	11111110
256.	255	$FF	11111111

Systémové Proměnné

Adresa (Hex)	Adresa (Dec)	PoPis
$0000	0	řídící registr 6510
$0001	1	in/out registr 6510 (obsluha Paměti a magnetofonu)
$0002	2	nePoužito
$0003-	3-	adresa Procedury Pro Převod
-$0004	-4	real na integer
$0005-	5-	adresa Procedury Pro Převod
-$0006	-6	integer na real
$0007	7	hledaný znak
$0008	8	Příznak Při hlědání řetězce
$0009	9	číslo sloUpce Před dalším TAB
$000A	10	Příznak Basicu;0.LOAD,1.VERIFY
$000B	11	ukazatel vstuPního buffru
$000C	12	Příznak DIM
$000D	13	typ:$00..numerický,$FF..řetězce
$000E	14	typ:$00..real,$FF..integer
$000F	15	Příznak LIST -"Posbírané smetí"
$0010	16	Přízn. Při ident. funkce,Pole
$0011	17	$00.INPUT,$40.GET,$98.READ
$0012	18	místo Pro goniometrické funkce
$0013	19	argument CMD
$0014-	20-	číslo řádku Při GOTO,LIST,ON,
-$0015	-21	GOSUB
$0016	22	ukazatel místa v řetězcích
$0017-	23-	adresa Předchozího řetězce
-$0018	-24	
$0019-	25-	zásobník Pro 3 řetězce (délka,
-$0021	-33	adresa)
$0022-	34-	místo sloužící různým cílům
-$0025	-37	
$0026-	38-	místo Pro násobení s Pohyblivou
-$002A	-42	desetinnou tečkou

Adresa (Hex)	Adresa (Dec)	Popis
$002B- -$002C	43- -44	ukazatel Počátku Programu v Basicu
$002D- -$002E	45- -46	ukazatel konce Progr. v Basicu; Počátek Proměnných
$002F- -$0030	47- -48	ukazatel Počátku Polí
$0031- -$0032	49- -50	ukazatel konce Polí
$0033- -$0034	51- -52	ukazatel Počátku řetězců
$0035- -$0036	53- -54	Pomocný ukazatel řetězců
$0037- -$0038	55- -56	ukazatel konce Paměti vyhrazené Pro Programy v Basicu
$0039- -$003A	57- -58	číslo Překládané řádky Programu
$003B- -$003C	59- -60	číslo Předchozí řádky Programu
$003D- -$003E	61- -62	ukazatel následující instrukce (Pro CONT)
$003F- -$0040	63- -64	číslo aktuálního data-řádku
$0041- -$0042	65- -66	adresa aktuální Položky data-řádku (vč. oddělovací čárky)
$0043- -$0044	67- -68	ukazatel na vstupní rutinu
$0045- -$0046	69- -70	jméno aktuální Proměnné Basicu
$0047- -$0048	71- -72	adresa aktuální Proměnné
$0049- -$004A	73- -74	adresa aktuální Proměnné cyklu FOR-TO-NEXT
$004B- -$004C	75- -76	Pomocný ukazatel Při Překladu

Adresa (Hex)	Adresa (Dec)	Popis
$004D	77	maska při operaci relace
$004E-	78-	místo sloužící různým cílům
-$0053	-83	
$0054-	84-	vektor skoku pro funkci
-$0056	-86	
$0057-	87-	místo pro numerické operace
-$0060	-96	
$0061.	97	Real akumulátor 1 (exponent)
$0062-	98-	Real akumulátor 1 (mantisa)
-$0065	-101	
$0066	102	Real akumulátor 1 (označení)
$0067	103	Počet elementů výrazu
$0068	104	Přestupný byte z akumulátoru 1
$0069-	105-	Real akumulátor 2
-$006E	-110	
$006F	111	výsledek porovnání akumul. 1,2
$0070	112	zaokrouhlený byte akumulátoru 1
$0071-	113-	délka kazetového buffru
-$0072	-114	
$0073-	115-	Procedura CHRGET;čtení znaku
-$008A	-138	
$008B-	139-	báze pro generování náhodných
-$008F	-143	čísel (RND)
$0090	144	stavový byte přenosu in/out
$0091	145	příznak stisknuté klávesy STOP
$0092	146	časová konstanta LOAD z mgf.
$0093	147	Příz.pro Kernal:0.LOAD,1.VERIFY
$0094	148	sériová sběrnice:znak v buffru
$0095	149	znak před vysláním na ser. Bus
$0096	150	sync. na konci pásky
$0097	151	Pomocná paměť pro registry X,Y
$0098	152	Počet otevřených souborů
$0099	153	kód vstupu: 0...klávesnice
$009A	154	kód výstupu:3...obrazovka

Adresa (Hex)	Adresa (Dec)	Popis
$009B	155	Parita Při čtení z Pásky
$009C	156	Příznak Při odebírání Bytu
$009D	157	typ:$00.Programový,$80.Přímý
$009E	158	kontrolní suma
$009F	159	korekce chyb Při nahrávání
$00A0-$00A2	160-162	aktuální čas od zapnutí Počíta-če
$00A3	163	místo Pro bity (Pro ser. Bus)
$00A4	164	obsluha impulsů
$00A5	165	zápis bitů na Pásku
$00A6	166	ukazatel buffru m3f.
$00A7-$00AB	167 171	obsah sloužící Pro práci RS-232
$00AC-$00AD	172-173	ukazatel na Počátek načítání dat
$00AE-$00AF	174-175	ukazatel konce Pro LOAD,SAVE, VERIFY
$00B0-$00B1	176-177	časová konstanta Při čtení z Pásky
$00B2-$00B3	178-179	ukazatel Počátku buffru Při čtení z Pásky
$00B4	180	Příznak Pro čtení z Pásky;RS232
$00B5	181	Přízn.konce Pásky;násl.bitRS232
$00B6	182	Byte Před vyslánim RS-232
$00B7	183	délka názvu souboru
$00B8	184	číslo aktuálního kanáku
$00B9	185	sekundární adresa
$00BA	186	kód Periférie
$00BB-$00BC	187-188	ukazatel na adresu,kde je uložen název souboru
$00BD	189	znak čt/zaPis.na Pás.;Parita RS
$00BE	190	Počet bloků dat
$00BF	191	buffer Pro čtení Byte z Pásky
$00C0	192	Příznak ovládání magnetofonu

Adresa (Hex)	Adresa (Dec)	Popis
$00C1-	193-	Počáteční adresa při operacích
-$00C2	-194	in/out
$00C3-	195-	Pomocný příznak při práci s
-$00C4	-196	páskou
$00C5	197	kód dříve stisknuté klávesy
$00C6	198	Počet znaků v Buffru klávesnice
$00C7	199	příznak:zapnutá inverze tisku
$00C8	200	Počet znaků na počítačovém řád.
$00C9	201	číslo řádky s kurzorem
$00CA	202	pozice kurzoru ve sloupci
$00CB	203	kód právě stisknuté klávesy
$00CC	204	příznak kurzoru:$00...bliká
$00CD	205	typ blikání kurzoru
$00CE	206	znak,nad kterým je kurzor
$00CF	207	negativ/pozitiv znaku pod kurz.
$00D0	208	znak vys. z klávesnice/obraz.
$00D1-	209-	adresa akt. řád. v obraz. RAM
-$00D2	-210	
$00D3	211	sloupec,ve kterém je kurzor
$00D4	212	Příznak (uvozovky):0..nebyly
$00D5	213	max. délka řádku z obrazovky
$00D6	214	řádek,na kterém je kurzor
$00D7	215	kód akt. znaku
$00D8	216	Počet znaků před přečtením
$00D9-	217	pomocné údaje pro jednotlivé
-$00F2	242	obrazovkové řádky
$00F3-	243-	adresa aktuální řádky v paměti
-$00F4	-244	barev
$00F5-	245-	ukazatel pro dekódování znaku
-$00F6	-246	
$00F7-	247-	adresa poč. vstup. buffru pro
-$00F8	-248	RS-232
$00F9-	249-	adresa poč. výstupního buffru
-$00FA	-250	pro RS-232

Adresa (Hex)	Adresa (Dec)	Popis
$00FB- -$00FE	251- -254	čtyři volné Byty určené pro využití uživatelem
$00FF- -$010A	255- -266	místa užívaná při převodu čísel na řetězce
$0100- -$013E	256- -318	čísla Bytů špatně přečtených z pásky
$0100- -$01FF	256- -511	zásobník Procesoru (je obsazován shora)
$0200- -$0258	512- -600	vstupní buffer Basicu
$0259- -$0262	601- -610	tabulka Pro Kernal: aktuální čísla souborů
$0263- -$026C	611- -620	tabulka Pro Kernal: aktuální kódy Periférií
$026D- -$0276	621- -630	tabulka Pro Kernal: aktuální sekundární adresy
$0277- -$0280	631- -640	vstupní buffer klávesnice
$0281- -$0282	641- -642	ukazatel počátku Paměti Pro Basic
$0283- -$0284	643- -644	ukazatel konce Paměti vyhrazené Pro Basic
$0285	645	příznak Pro IEC-625
$0286	646	aktuální kód barvy znaků
$0287	647	aktuální kód barvy Pozadí
$0288	648	Hi Byte adresy obrazovkové RAM
$0289	649	velikost kl. buffru
$028A	650	oPakovací funkce kláves
$028B	651	zPoždění mezi oPětovným čtením
$028C	652	zPoždění Pro Potvrzení
$028D	653	příznak (stisk.CONTROL,C=,SH,..
$028E	654	Předchozí stav příznaku SHIFT,.
$028F- -$0290	655- -656	ukazatel na dekódování kláves-nice

Adresa (Hex)	Adresa (Dec)	Popis
$0291	657	Přep.tisk.módu($00.zap,$80.vyp)
$0292	658	rolování txt:$00.zap,<>$00.vyp)
$0293	659	RS-232:6551 Povelový registr
$0294	660	RS-232:6551 řídící registr
$0295-$0296	661-662	RS-232:nestandardní Parametry Přenosu
$0297	663	RS-232:6551 stavový registr
$0298	664	RS-232:Počet bitů Před vysláním
$0299-$029A	665-666	RS-232:rychlost Přenosu (bit za mikrosekundu)
$029B	667	RS:ukaz.v buffru odebran. znaků
$029C	668	RS:Poč buffru na odebír. znaků
$029D	669	RS:ukaz.Poč.v buffru vyslan.zn.
$029E	670	RS:ukaz.konce v buffru vysl.zn.
$029F-$02A0	671-672	stav IRQ vektoru Před spoluprá-ci s magnetofonem
$02A1	673	Příznak NMI Pro CIA 2
$02A2-$02A4	674-676	Parametry Pro CIA 1 Při spolu-Práci s páskou
$02A5	677	Pomocný uk.řádku Pro rolování
$02A6	678	Příznak:0..NTSC,1..PAL
$02A7-$02FF	679-767	volný Prostor určený Pro uživa-tele (704-766 SPrite 11)
$0300-$0301	768-769	vektor:Procedura Pro tisk chy-bových hlášek
$0302-$0303	770-771	vektor:hlavní část Překladače (teplý start Basicu)
$0304-$0305	772-773	vektor:Přepis textu na Příkazy, zpracování v buffru
$0306-$0307	774-775	vektor:rutina tisknoucí Příkazy (LIST)
$0308-$0309	776-777	vektor:zpracování následujícího Příkazu Basicu

Adresa (Hex)	Adresa (Dec)	Popis
$030A- -$030B	778- -779	vektor:Procedura pro aritmetické vyjádření
$030C	780	stav A (akumulátoru) před SYS
$030D	781	stav X před SYS
$030E	782	stav Y před SYS
$030F	783	stav SP před SYS
$0310	784	kód $4C...JMP Hi Lo (pro USR)
$0311- -$0312	785- -786	Lo skok do programu ve strj. Hi kódu (USR)
$0313	787	volný Byte
$0314- -$0315	788- -789	vektor:Hardware přerušení (IRQ-stand.:$EA31)
$0316- -$0317	790- -791	vektor:přerušení po instrukci BRK ($FE66)
$0318- -$0319	792- -793	vektor:nemaskovatelné přerušení (NMI-FE47)
$031A- -$031B	794- -795	vektor:rutina OPEN
$031C- -$031D	796- -797	vektor:rutina CLOSE
$031E- -$031F	798- -799	vektor:rutina CHKIN
$0320- -$0321	800- -801	vektor:rutina CHKOUT
$0322- -$0323	802- -803	vektor:rutina CLRCHN
$0324- -$0325	804- -805	vektor:rutina CHRIN
$0326- -$0327	806- -807	vektor:rutina CHROUT
$0328- -$0329	808- -809	vektor:rutina STOP
$032A- -$032B	810- -811	vektor:rutina GETIN

Adresa (Hex)	Adresa (Dec)	Popis
$032C-	812-	vektor:rutina CLALL
-$032D	-813	
$032E-	814-	Použitý IRQ ($FE66)
-$032F	-815	
$0330-	816-	vektor:rutina LOAD
-$0331	-817	
$0332-	818-	vektor:rutina SAVE
-$0333	-819	
$0334-	820-	osm volných Bytů
-$033B	-827	
$033C-	828-	buffer užívaný Pouze Při Práci s mgf.(Sprite 13,14,15)
-$03FB	-1019	
$03FC-	1020-	čtyři volné Byty
-$03FF	-1023	
$0400-	1024-	obrazovková Paměť RAM
-$07E7	-2023	
$07E8-	2024-	šestnáct volných Bytů
-$07F7	-2039	
$07F8-	2040-	nastavení Spritů
-$07FF	-2047	
$0800-	2048-	Paměť určená na Programy v Basicu
-$9FFF	-40959	
$8000-	32768-	toto místo může zaujímat vnější Paměť ROM
-$9FFF	-40959	
$A000-	40960-	Basic ROM (nebo 8KB RAM)
-$BFFF	-49151	
$C000-	49152-	4KB RAM (na Programy v Assem-
-$CFFF	-53247	bleru,Popř. i v Basicu,obraz.)
$D000-	53248-	in/out+color RAM,nebo znaková
-$DFFF	-57343	ROM,nebo 4KB RAM
$E000-	57344-	Kernal ROM (nebo 8KB RAM)
-$FFFF	-65535	

Podívejme se ještě na uspořádání paměti pro obvody vstupu a výstupu:

Adresa (Hex)	Adresa (Dec)	Popis
$0000	0	MOS 6510 řídící registr (1-výstup,0-vstup).Norm.:[xx101111] x...libovolná hodn. (1,nebo 0) MOS 6510 registr bit význam 0..LORAM (0-Basic ROM vyřazena) 1..HIRAM (0-Kernal ROM vyřazena) 2..CHARAM(0-znak. ROM zařazena) 3..data na mgf - výstup 4..spínač v mgf (1-sepnutý) 5..ovl. motoru (0-zap,1-vyp) 6..nevyužit 7..nevyužit
$0001	1	
$D000-$D02E	53248-54271	MOS 6566 Video obvod
$D000	53248	Sprite 0,souřadnice X
$D001	53249	Sprite 0,souřadnice Y
$D002	53250	Sprite 1,souřadnice X
$D003	53251	Sprite 1,souřadnice Y
$D004	53252	Sprite 2,souřadnice X
$D005	53253	Sprite 2,souřadnice Y
$D006	53254	Sprite 3,souřadnice X
$D007	53255	Sprite 3,souřadnice Y
$D008	53256	Sprite 4,souřadnice X
$D009	53257	Sprite 4,souřadnice Y
$D00A	53258	Sprite 5,souřadnice X
$D00B	53259	Sprite 5,souřadnice Y
$D00C	53260	Sprite 6,souřadnice X
$D00D	53261	Sprite 6,souřadnice Y
$D00E	53262	Sprite 7,souřadnice X
$D00F	53263	Sprite 7,souřadnice Y
$D010	53264	Sprite 0-7:9-tý bit X-ové souř. [S7|S6|S5|S4|S3|S2|S1|S0] (Pokud je X>255 zapíná se příslušný bit a X jde opět od 0.)

Adresa (Hex)	Adresa (Dec)	Popis
$0011	53265	VIC-nastavovací registr bit — význam 0-2 .. zarolování ve směru osy Y 3 .... text.mód 24/25 řád (1..25) 4 .... vyp/zap obrazovky (0..vyp) 5 .... jemná grafika (1..zap) 6 .... vícebarevný text (1..zap) 7 .... znaménkový bit reg. $0012
$0012	53266	registr rastru
$0013	53267	světelné pero,X-ová souřadnice
$0014	53268	světelné pero,Y-ová souřadnice
$0015	53269	zapínání Spritů (1..zapnut)
$0016	53270	VIC-nastavovací registr bit — význam 0-2 .. zarolování ve směru osy X 3 .... text.mód 38/40 sl. (1..40) 4 .... barevný mód (1..zap) 5 .... tento bit musí mít hodn.0! 6-7 .. neobsazeno
$0017	53271	zvětšení jednotl. Spritů 2x ve směru osy Y
$0018	53272	VIC regis.Pro nastavení Paměti bity 1-3 ... základní adresa zn. Paměti bity 4-7 ... základní adresa video Paměti
$0019	53273	VIC-přerušovací registr (1..IRQ zap) bit — význam 0 ... IRQ Pro rastr 1 ... IRQ -Sprite/pozadí kolize 2 ... IRQ -Sprite/Sprite kolize 3 ... IRQ pro světelné pero 4-6 .nepoužity 7 ... možnost zapnutí IRQ
$001A	53274	IRQ-maskovací registr
$001B	53275	Sprite/Pozadí Priorita (1...Sprite)
$001C	53276	vícebarevný mód Spritů (1..zap)
$001D	53277	zvětšení Spritů 2x v ose X(1=z)

Adresa (Hex)	Adresa (Dec)	Popis
$D01E	53278	kolize mezi jednotlivými SPrity (1..kolize,0..nepřekrývají se)
$D01F	53279	SPrite/Pozadí kolize
$D020	53280	barva rámečku
$D021	53281	barva Pozadí 0
$D022	53282	barva Pozadí 1
$D023	53283	barva Pozadí 2
$D024	53284	barva Pozadí 3
$D025	53285	SPrite-barva 0 v barevném módu
$D026	53286	SPrite-barva 1 v barevném módu
$D027	53287	barva sPritu 0
$D028	53288	barva sPritu 1
$D029	53289	barva sPritu 2
$D02A	53290	barva sPritu 3
$D02B	53291	barva sPritu 4
$D02C	53292	barva sPritu 5
$D02D	53293	barva sPritu 6
$D02E	53294	barva sPritu 7
$D400-$D7FF	54272-55295	MOS 6581 Zvukový obvod
$D400	54272	Hlas 1:Lo Byte frekvence tónu
$D401	54273	Hlas 1:Hi Byte frekvence
$D402	54274	Hlas 1:Lo Byte -střída(jen Pro
$D403	54275	Hlas 1:Hi střída obdélník)
$D404	54276	Hlas 1:nastavovací registr bit — význam 0..gate(1-start ADS,0-start R) 1..osc.1 a osc.3 sync.(1-zap) 2..osc.1-osc.3 modulace (1-zap) 3..testbit (1-osc.1je zapnut) 4..trojúhelníkový Průběh 5..pilovitý Průběh 6..pravoúhlý Průběh 7..šum
$D405	54277	Hlas 1:obalová křivka bity 0-3:Decay ;bity 4-7:Attack (Parametry obálky - ADSR)

Adresa (Hex)	Adresa (Dec)	PoPis
$D40E	54278	Hlas 1:obalová křivka bity 0-3:Release ;4-7:Sustain
$D407- -$D40D	54279- -54285	Hlas 2:obsazení stejné jako u hlasu 1,Pouze sync. a modulace je s oscilátorem 1.
$D40E- -$D414	54286- -54292	Hlas 3:obsazení stejné jako u hlasu 1,sync. a modulace je s oscilátorem 2.
$D415	54293	Lo frekvence filtru (bity 0-2)
$D416	54294	Hi frekvence filtru (bity 0-7)
$D417	54295	Rezonance/filtry bit — význam 0...filtr zařazen do hl.1 (1=ano) 1...filtr zařazen do hl.2 (1=ano) 2...filtr zařazen do hl.3 (1=ano) 3...filtr zařazen do ext.sign. 4-7...rezonance (0-min;$FX max)
$D418	54296	typ filtru/hlasitost bit — význam 0-3...hlasitost (0=min;$XF=max) 4...dolní propust (1=zap) 5...pásmová propust (1=zap) 6...horní propust (1=zap) 7...vypínání hlasu 3 (1=vyp)
$D419	54297	výstup A/D převodníku z portu 1
$D41A	54298	výstup A/D převodníku z portu 2
$D41B	54299	výstup osc.3,náhodné spouštění
$D41C	54300	výstup křivky hlasu 3
$D500- -$D7FF	54528- -55295	Obrazy adres zvukového obvodu
$D800- -$DBFF	55296- -56319	Paměť barev jednotlivých pixelů
$DC00- -$DCFF	56320- -56575	MOS 6526 Mezistykový obvod #1
		(Tyto obvody máme v počítači dva.Slouží ke styku počítače s okolím (kláves.,joy,RS-232,atd)

Adresa (Hex)	Adresa (Dec)	PoPis
$DC00	56320	Datový Port A (klávesnice,joy, Potenciometry,světelné Pero) bit — význam 0-3...joystick-výstuP 4.....střelba z joysticku 5.....nePoužit 6-7...Potenciometry-vstuP (10-Port A,01-Port B) Při čtení klávesnice jsou bity 0-7 Použity Pro výběr slouPce.
$DC01	56321	Datový Port B (klávesnice,joy) bit — význam 0-3...joystick-výstuP 4.....střelba z joysticku 5.....nePoužit 6.....časovač A:výstuP imPulsů 7.....časovač B:výstuP imPulsů Při čtení klávesnice slouží bity 0-7 Pro výběr řádky.
$DC02	56322	řídící registr Portu A (56320)
$DC03	56323	řídící registr Portu B (56321)
$DC04	56324	časovač A:Lo Byte
$DC05	56325	časovač A:Hi Byte
$DC06	56326	časovač B:Lo Byte
$DC07	56327	časovač B:Hi Byte
$DC08	56328	hodiny:1/10 sekundy
$DC09	56329	hodiny:sekundy
$DC0A	56330	hodiny:minuty
$DC0B	56331	hodiny+Příznak AM/PM (doPoledne/odPoledne)
$DC0C	56332	registr Pro sériovou sběrnici
$DC0D	56333	Přerušovací registr CIA #1 bit — význam 0...časovač A -přerušení 1...časovač B -přerušení 2...hodiny -přerušení 3...sériova sbernice -přerušení 4...příznak IRQ (čtení z mgf.-ze sériové sběrnice) 5-6.nevyužity 7...příznak IRQ (zaPínání)

Adresa (Hex)	Adresa (Dec)	Popis
$DC0E	56334	nastavovací registr A bit — význam 0...start/stop časovače A (0-stop; 1-start) 1...výstup na PB6 (0-výp,1-zap) 2...mód výstupu (0-impuls) 3...typ(0-nepřetržitý,1-impuls) 4...zatížení-časovač A(1-ano) 5...vstup časovače A (0-ze sys. taktu Ø2,1-CNT signál) 6...in/out-ser.sběrnice (0-in) 7...výstupní kmitočet (hodinový) (0-60Hz, 1-50Hz)
$DC0F	56335	nastavovací registr B bit — význam 0...start/stop časovače B 1...výstup na PB7 2...mód výstupu 3...typ 4...zatížení 5-6.výstup (00-z taktu Ø2,10-CNT signál,01-z časovače A, 11-z časovače A,je-li CNT=1) 7...hodiny/budík (0-hodiny,1-budík)-směr čítání
$DD00-$DDFF	56576-56831	MOS 6526 Mezistykový obvod #2
$DD00	56576	datový Port A (sériová sběrnice,RS-232,VIC) bit — význam 0-1.nastavení video-banky 2...RS-232:datový výstup 3...sér.sběrnice:ATN výstup 4...sériová sb.:výstup impulsů 5...sér.sběr.:datový výstup 6...sér.sběr.:vstup impulsů 7...sér.sběr.:datový vstup
$DD01	56577	datový Port B (RS-232) bit — význam 0...Received (přijato) 1...Request to send (P.na vysl) 2...Data -Terminal (data) 3...Ring indicator (vyzvánění) 4...Carrier detect (nosný sg.) 5...uživatelský 6...Clear to send (Přípravena) 7...Data ready (data připravena) (viz výstupy User Portu)

Adresa (Hex)	Adresa (Dec)	Popis
$DD02	56578	řídící registr portu A
$DD03	56579	řídící registr portu B
$DD04-$DD0F	56580-56591	totéž jako $DC04-$DC0F (56324-56335)
$DE00-$DEFF	56832-57087	rezervováno pro další možné rozšíření obvodů in/out
$DF00-$DFFF	57088-57343	rezervováno pro další možné rozšíření obvodů in/out

Poznámky:

Uložení Basicu a Proměnných

Mezi příklady programů,psaných v Basicu,byl uveden jeden,který jistě vzbudil vaší pozornost- jmenoval se "Datařádky".Jak je možné,že jeden program dokáže vyrobit jiný?Jak lze takový program sestavit a- to hlavně - uložit do paměti? Tak na tyto a mnohé jiné otázky vám pomůže odpovědět tato podkapitola.

Programy v Basicu mají v paměti vyhrazenu určitou oblast(2048-40959).V ní tedy je program zapsán (počínaje 2048).Podívejme se na příklad:

```
10 PRINT"AHOJ":POKE198,1
20 GETA$:PRINTA$:GOTO10
```

tento program se uloží takto:

```
$0800 (2048)   $00   (pokud zde není 0, nejedná se o
                     program v Basicu-nejde spustit)
$0801 (2049)   $14   lo  ukazatel na počátek dalšího
$0802 (2050)   $08   Hi  programového řádku
$0803 (2051)   $0A   lo  číslo programového řádku 10
$0804 (2052)   $00   Hi
$0805 (2053)   $99   PRINT (kód příkazu)
$0806 (2054)   $22   " (kód uvozovek)
$0807 (2055)   $41   A (kód znaku A)
$0808 (2056)   $48   H (kód znaku H)
$0809 (2057)   $4F   O (kód znaku O)
$080A (2058)   $4A   J (kód znaku J)
$080B (2059)   $22   " (kód uvozovek)
$080C (2060)   $3A   : (kód dvojtečky)
$080D (2061)   $97   POKE (kód příkazu)
$080E (2062)   $31   1 (kód znaku 1)
$080F (2063)   $39   9 (kód znaku 9)
$0810 (2064)   $38   8 (kód znaku 8)
$0811 (2065)   $2C   , (kód čárky)
$0812 (2066)   $31   1 (kód znaku 1)
$0813 (2067)   $00   (konec programové řádky)
$0814 (2068)   $24   lo  ukazatel na počátek dalšího
$0815 (2069)   $08   Hi  programového řádku
$0816 (2070)   $14   lo  číslo řádku 20 (desítkově)
$0817 (2071)   $00   Hi
$0818 (2072)   $A1   GET (kód příkazu)
$0819 (2073)   $41   A (kód znaku A)
```

```
$081A (2074)   $24   $ (kód znaku $)
$081B (2075)   $3A   : (kód dvojtečky)
$081C (2076)   $99   PRINT (kód příkazu)
$081D (2077)   $41   A (kód znaku A)
$081E (2078)   $24   $ (kód znaku $)
$081F (2079)   $3A   : (kód dvojtečky)
$0820 (2080)   $89   GOTO (kód příkazu)
$0821 (2081)   $31   1 (kód znaku 1)
$0822 (2082)   $30   0 (kód znaku 0)
$0823 (2083)   $00   (konec programového řádku)
$0824 (2084)   $00   (konec programu v Basicu)
$0825 (2085)   $00
```

V systémových proměnných jsme si všimli různých ukazatelů pro Basic.Podívejme se,kam ukazují po zapsání tohoto programu:

```
$0281-$0282 (641-642) ukazatel poč. paměti pro Basic.Ukazuje na $0800 (2048)
$0283-$0284 (643-644) ukazatel konce paměti pro Basic.Ukaz. na $A000 (40960)
$0037-$0038 ( 55- 56) ukazatel na konec paměti pro program v Basicu a proměnné. Při překročení hlásí "Out of memory".Ukaz.na $9FFF(40959)
$0033-$0034 ( 51- 52) ukazatel na poč. řetězců. Ukazuje na $A000 (40960).
$0031-$0032 ( 49- 50) konec polí
$002F-$0030 ( 47- 48) počátek polí
$002D-$002E ( 45- 46) poč. proměnných,konec Basicu Všechny tyto ukazatele ukazují na $0826 (2086).
$002B-$002C ( 43- 44) počátek programu v Basicu. Ukazuje na $0801 (2049).
```

Ještě připomínám,že jsme popsali stav před spuštěním programu. Až se bude ukládat proměnná, některé ukazatele se změní ($2F-$34).Tak teď již víme, jak je vlastně náš program uložen. Ze způsobu uložení vyplývá, že přebytečné mezery mezi příkazy by nám zbytečně zabíraly paměť.A jak je to s proměnnými?

Proměnné se ukládají do paměti určené pro Basic. Proto jim program v Basicu (který používá nějaké proměnné) musí nechat trochu místa.Při letmém pohledu na ukazatele (viz.Předch.str.; syst.prom.) je zřejmé,kam se proměnné budou ukládat.Stručně to můžeme říci takto: proměnné real a integer a parametry řetězců hned za konec programu. Za ně se pak ukládají pole. Naopak od konce paměti směrem dolů se ukládají jednotlivé znaky řetězců. Při ukládání se příliš místem nešetří,proto se může stát, že se paměť naplní daty a mohlo by dojít ke kolizi.V tom okamžiku se však,před uložením další proměnné, zavolá rutina, která proměnné trochu srovná a rázem je místa dost.Nicméně pokud je program dlouhý,může se stát,že nás překvapí hláška "Out of memory".Jedinou možností je pracovat s co nejmenším počtem proměnných. A teď se podíváme,jak jsou jednotlivé proměnné ukládány(stav po NEW:CLR):

Integer

Adresa:	$0803	$0804	$0805	$0806	$0807	$0808	$0809
AA%=257	$C1	$C1	$01	$01	$00	$00	$00
A%=-257	$C1	$80	$FE	$FF	$00	$00	$00
AB%=8	$C1	$C2	$00	$08	$00	$00	$00
AB%=-8	$C1	$C2	$FF	$F8	$00	$00	$00
	↑ OR$80	↑ OR$80	↑ Hi	↑ Lo	↑ Tato místa nepoužita		jsou
Chr$ kódy	↑ Proměnných		↑ Hodnota se znaménkem				

Proměnná se ukládá na místo kam ukazuje ukazatel $2D,$2E.Na místo,kam by se ukládala další proměnná ukazuje $2F,$30. Na $0E se uloží typ aktuální proměnné ($00-real,$80-integer). Adresy $45,$46 obsahují jméno aktuální proměnné a $47,$48 ukazuje do jejích dat (na Hi hodnotu).Poslední tři $00-ly nemají pro hodnotu proměnné žádný význam ("srovnávací" rutina je vyřazuje).

Real

Adresa:	$0803	$0804	$0805	$0806	$0807	$0808	$0809
RA=10	$41	$41	$84	$20	$00	$00	$00
RB=513	$41	$42	$8A	$00	$40	$00	$00
RC=-300	$41	$43	$89	$96	$00	$00	$00
RD=-256	$41	$44	$89	$80	$00	$00	$00
	↑	↑	↑	↑	↑	↑	↑
	Chr$ kód	jména	a	z+c1	c2	c3	c4

Desetinný tvar čísla real (např.X) dostaneme podle následujícího vzorce (decimálně):

$X=(2^{a-129})(-1^{z})(1+c)$,kde

a...číslo od 0..255 (exponent)

z...7-mý bit:znaménko (hodnota 0,nebo 1)

c...mantisa (je menší,maximálně rovna,jedné).

Mantisa vznikne z jednotlivých bitů slov c1-c4:

Byte:	Bity:							
	7	6	5	4	3	2	1	0
z+c1:	z	1	2	3	4	5	6	7
c2:	8	9	10	11	12	13	14	15
c3:	16	17	18	19	20	21	22	23
c4:	24	25	26	27	28	29	30	31

$c=2^{-(zapnutý\ bit)}(...+2^{-(zapnutý\ bit)})$.

Zkusme si to pro RB=513:

a=138 ; z=0 ; Byte z+c1 je nulový,tedy nejen z=0, ale i c1=0. Byte c2 je nenulový.Rozepíšeme si ho v binárním tvaru: c2=01000000 .Byty c3 a c4 jsou nulové,takže na hodnotu nemají vliv.A teď počítejme zapnutý bit v c2 má ozn.9.Tedy:

$c=2^{-9}=0.001953125$.

Po dosazení:

$X=(2^{138-129})(-1^{0})(1+0.001953125)=513$.

A teď to zkusme třeba pro číslo X=100:

a=135;z+c1=72 (01001000 Bin):c2=c3=c4=0 ,takže

$X=(2^{135-129})(-1^{0})(1+2^{-1}+2^{-4})=64\times1\times1.562500=100$.

Hodnota real zabírá, na rozdíl od int., sedm Bytů.

String

Při ukládání řetězců musíme od sebe odlišovat dvě věci. Za prvé je to samotný řetězec skládající se pouze ze znaků a za druhé jsou to data,nebo údaje, o tomto řetězci. Mezi tyto údaje patří jméno Proměnné řetězce, počet znaků, ze kterých se skládá a nakonec je to adresa místa,kde řetězec začíná.Tyto údaje se ukládají mezi Proměnné, ale samotný řetězec se ukládá od konce Paměti směrem dolů a to tak že, Poslední znak řetězce je blíže ke konci Paměti a První blíže k jejímu Počátku.Takže:

```
Adresa:   $0803 $0804 $0805 $0806 $0807 $0808 $0809
AA$ =      $41   $C1   $04   $FC   $9F   $00   $00
="AHOJ"     ↑     ↑     ↑     ↑     ↑     ↑     ↑
            1   OR$80  Délka  Lo    Hi    nePoužito
           Chr$ kódy           adresa ře-
           Proměnných            tězce
```

```
Adresa:   $97FC $97FD $97FE $97FF
Kód:       $41   $48   $4F   $4A
          ( A     H     O     J )
```

Většina řetězců se začne ukládat o Byte níže, než kam ukazuje $33,$34. Tento ukazatel se Pak Přenastaví tak, že ukazuje na První znak Posledně vloženého řetězce.Jistě jste nePřehlédli slůvko "většina" a jistě vás zajímá,kam,že se ukládá ten zbytek - tak tedy: do této oblasti se neukládají řetězce, které už jsou uloženy jinde (byl by to velký Přepych) - zPravidla Přímo v Programu.Jako Příklad uvedu:

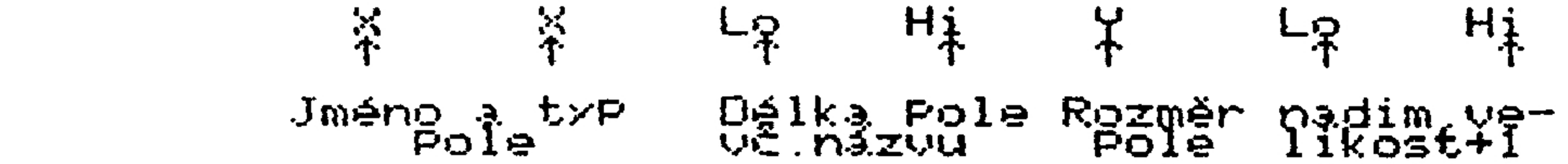
```
10 C$="AHOJ"
```

v tomto PříPadě se jako Počáteční adresa řetězce nastaví adresa,na které leží znak A a ta se -sPolu s délkou a názvem řetězce uloží mezi Proměnné.ZPůsob ukládání těchto údajů do Proměnných je naProsto stejný,jako zPůsob ukládání jakékoliv jiné Proměnné (integer,real) včetně Použití ukazatelů.

Ještě se zmíním o ukazateli $35, $36 - tento uk. Při ukládání řetězců ukazuje na PředPoslední.

Pole

A zbývá nám zjistit,jak se ukládají do Paměti Pole (tabulky).Vyjdeme,stejně jako v předchozích případech,ze stavu po NEW:CLR.Jistě nás nepřekvapí mnohé podobnosti s ukládáním Proměnných. Stejně jako máme tři druhy Proměnných, máme i tři druhy Polí : real,integer,string. Poznávacím znamením je opět 7 bit názvu Proměnné.Takže:



Je-li Pole vícerozměrné,následují dva Byte (Lo, Hi nadimenzovaná velikost +1) Pro každý další rozměr. Pak následuje Prostor Pro hodnoty. Pro integer to jsou vždy 2 Byty (Hi,Lo hodnota),Pro real je to už 5 Bytů a Pro řetězce jsou to 3 Byty.

Počátek Polí ukazuje ukazatel $2F, $30.Na konec Polí (a zároveň na místo Pro případné další Pole) ukazuje $31,$32.

Znalost způsobu uložení Proměnných je užitečná nejen Pro výhodnou tvorbu Programů v Basicu,ale také nám dává možnost vyměňovat si (i větší množství) údaje s Programem v Assembleru (viz.Bleskové třídění).

Organizace Paměti

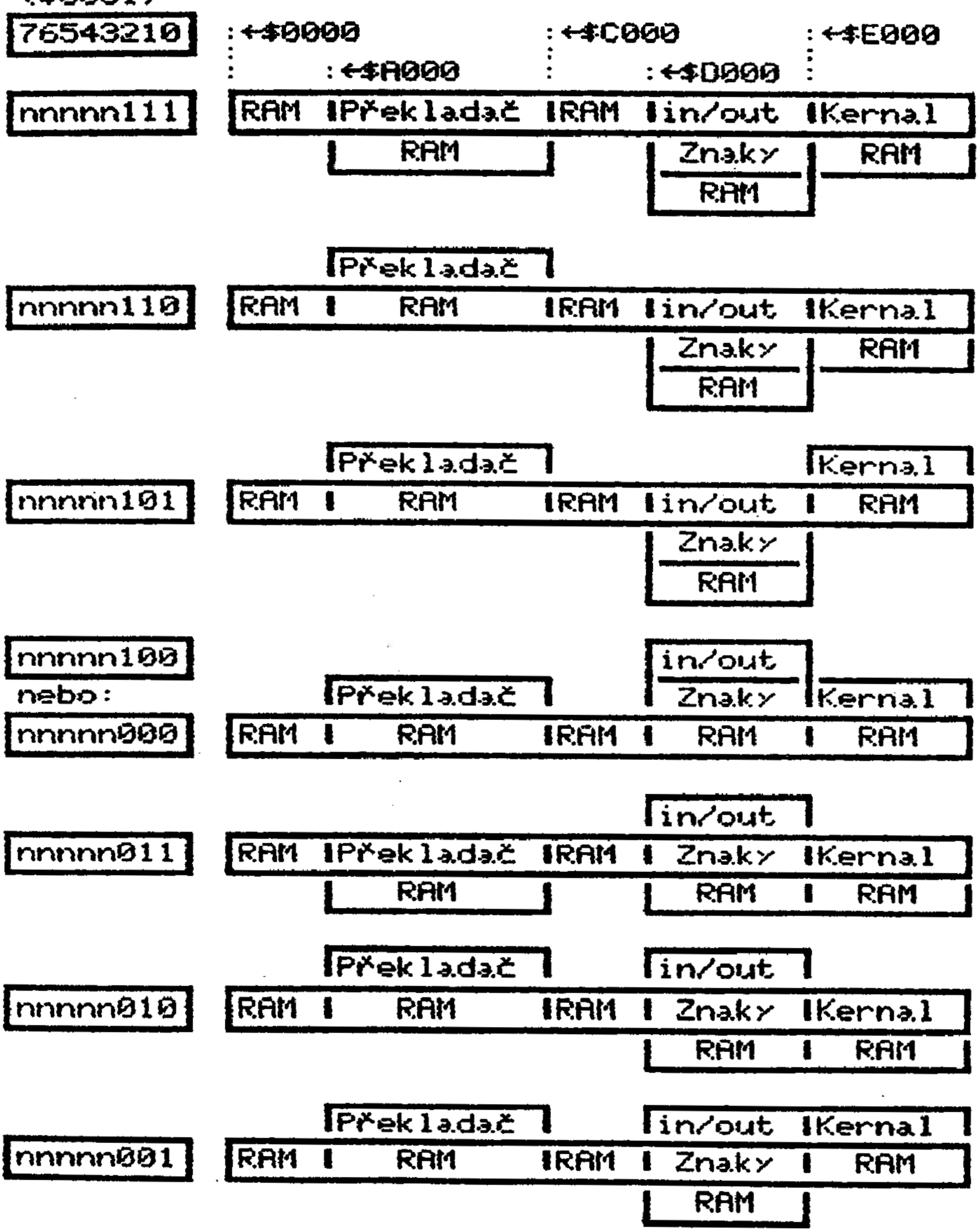
Na konci systémových Proměnných vám jistě neušlo, že se tam několikrát objevilo spojení "ROM, nebo RAM". Máme přeci jenom jednu hlavní datovou a adresovou sběrnici. Jak to tedy vlastně je?

V našem Počítači jsou opravdu zapojeny paralelně (tedy vedle sebe) dva různé typy Pamětí. Protože však na sběrnice může být připojena vždy jen jedna z těchto Pamětí, musíme je přepínat. Tento systém je velice výhodný. Pevné Paměti, které by normálně zabraly asi čtvrtinu našich 64KB můžeme, v případě, že je nebudeme potřebovat, prostě odpojit. V praxi toto uprázdněné místo můžeme využít např. pro úschovu dat, pro uložení programů v Assembleru a nakonec i pro grafickou či obrazovkovou paměť. S pamětí RAM, schovanou pod ROM-kami, však můžeme pracovat i v Basicu. Přepínání za nás obstará překladač. Jistě jste viděli, že v jednom místě jsou paralelně spojeny dokonce tři Paměti. Jednou z nich je znaková ROM, která se z Basicu jeví jako odpojená, ale přesto se v ní vyhledávají tvary jednotlivých znaků a přenášejí se na obrazovku. Bez přepínání by to však také nešlo. Tohle přepínání má zase na svědomí video-obvod. A protože byl vyroben na zakázku, dokáže na obrazovku přenést i data z grafické RAM, která jsou uložena "pod" zařazeným Kernalem.

Jiným příkladem použití paralelních pamětí je např. modifikace překladače. Nějakým způsobem vylepšíme Basic a uložíme ho do RAM pod Basic (který je v ROM) - no a pak už jenom přepneme.

Při přepínání pamětí je však třeba dodržovat jistá pravidla. Hlavně musíme dávat pozor, jestli v paměti, kterou budeme vyřazovat zrovna neběží nějaký program. Pokud tomu tak není a ani nebude, můžeme přepínat. Při přepínání v oblasti vstupů/výstupů je třeba, pokud už se tak nestalo, vypnout přerušovací časovač, nebo použít instrukci pro zákaz přerušení. Samotné přepínání se uskutečňuje pomocí registru

in/out Procesoru 6510.Má adresu $0001.Podívejme se na obrázek (n...nePoužito Pro PřePínání Paměti):



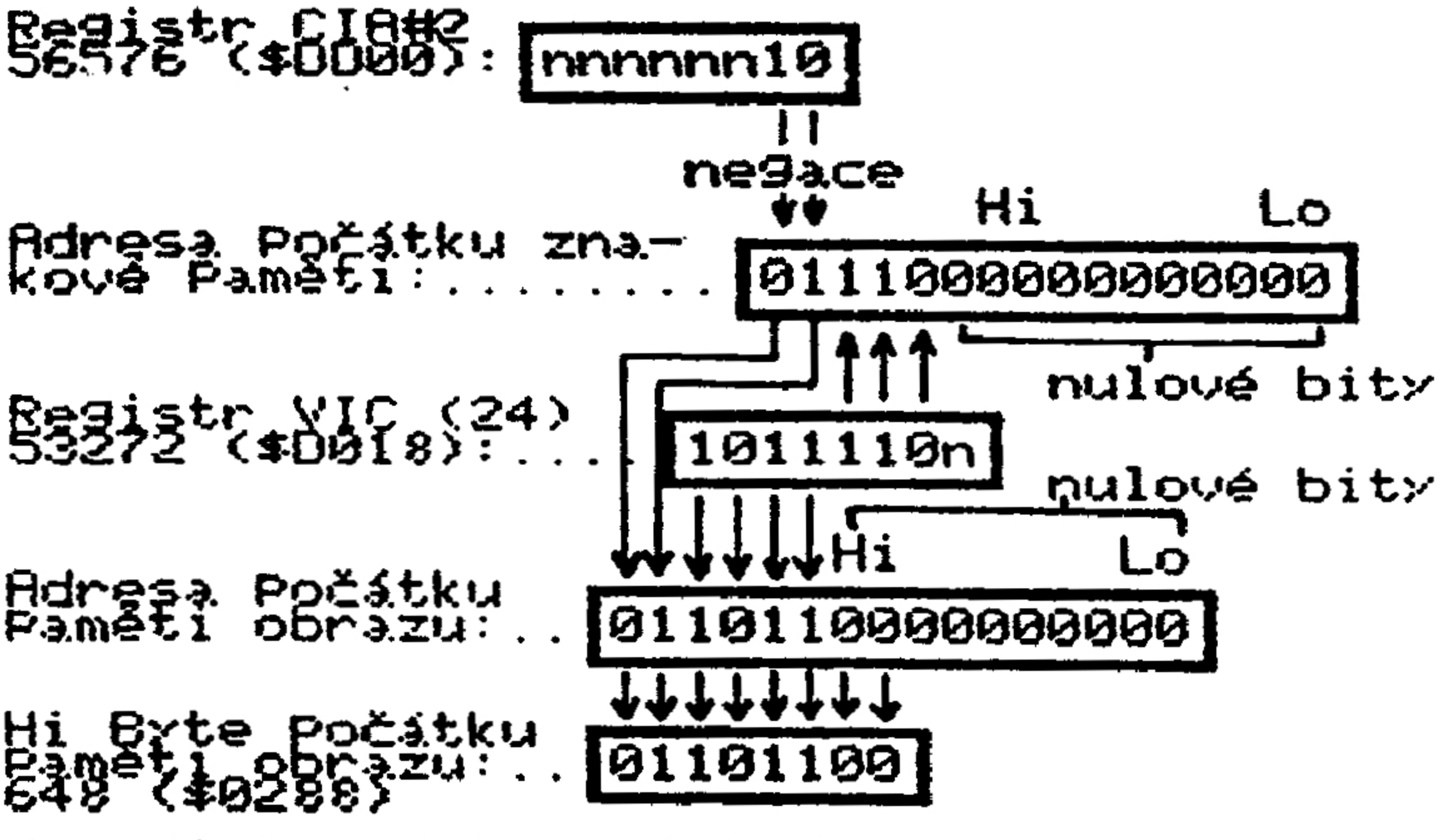
V ExPansion Portu najdeme vývody GAME a EXROM,které jsou určeny Pro vyřazení vnitřních Pamětí v oblasti $8000-$A000($C000) a zařazení Pam. vnějších.

Obraz

Jednou z přednostÍ C=64 je jeho video-obvod.V této podkapitole bychom se měli podÍvat na některé jeho vlastnosti.

Hrubá grafika,znakové sady

Pod pojmem hrubá grafika si můžeme představit třeba obrázek nakreslený pomocÍ jedné ze standardních znakových sad. Jako příklad může posloužit program pro tabulkové výpočty.Kód znaku se ukládá do obrazovkové RAM, která má 40(zn./řádek)x25(řádků)=1000 Bytů. Kód znaku můžeme do této paměti zapsat buď přÍmo (POKE,STR$),nebo můžeme použÍt prostřednÍka-v Basicu příkaz PRINT (popř. rutinu CHROUT). VIC si přečte kód a podle něj vyhledá ve znakové ROM tvar příslušného znaku a ten přenese na obrazovku. Aby obvod věděl,kde má co hledat,musÍme mu dát potřebné informace. Informaci potřebuje také rutina, která zajišťuje tisk na obrazovku. Tyto informace zapÍšeme do dvou registrů a na jednu syst. proměnnou. PodÍvejme se na obrázek (n...nevýznamné pro nastavenÍ obrazovkové a znakové paměti):



V registru CIA#2 ($DD00) nastavujeme,ve které části paměti bude ležet obrazovková a znaková paměť. Podle tohoto schématu tedy dokážeme umÍstit paměť obrazu a znaků. NepokoušejÍte se to však prověřovat na standardnÍm režimu. Jak jsem se již zmÍnil, VIC

byl udělán na zakázku, Proto je částečně PředPro-
gramován. Ve standardním režimu začíná obrazovková
RAM na adrese $0400(-$07E7) a znaková ROM na $D000
(-$DFFF).Pokud umístíte obrazovkovou Paměť jinam,
ihned Proveďte tisk CHR$(147)..mazání obrazovky.
Tím se aktivují syst. Proměnné Pro Kernal. Pokud
bychom to neudělali,znaky by se dále zaPisovaly do
předchozí obrazovkové Paměti. Dělá to dojem, že se
Počítač "zakousnul".

V obrazovkové Paměti jsou kódy znaků uloženy
Po jednotlivých obrazovkových řádkách,ale jak jsou
uložena data jednotlivých znaků? Co kdybychom si
chtěli udělat "svoje"znaky,jak na to?Podívejme se,
jak jsou uložena data Pro znaky A a B ve znakové
ROM (tyto znaky jdou za sebou):

Adr.	Znak	76543210	Adr.	Znak	76543210
$D008		00011000	$D010		01111100
$D009		00111100	$D011		01100110
$D00A		01100110	$D012		01100110
$D00B		01111110	$D013		01111100
$D00C		01100110	$D014		01100110
$D00D		01100110	$D015		01100110
$D00E		01100110	$D016		01111100
$D00F		00000000	$D017		00000000

Tak tedy tímto zPůsobem můžeme vyrábět své vlast-
ní znaky. Použití je široké. Můžeme si udělat čes-
ké znaky nebo různé grafické symboly,PoPř. Pozadí
k nějaké hře. Záleží jen a jen na naší fantazii.
Některé Příklady na změnu znakové sady jsou uve-
deny v Poslední Podkapitole. Kolik Bytů zabírá Pl-
ná (256 znaků) znaková sada? 8*256=2048,tedy 2KB.
V C=64 jsou znakové sady dvě, Proto zabírají 4KB.

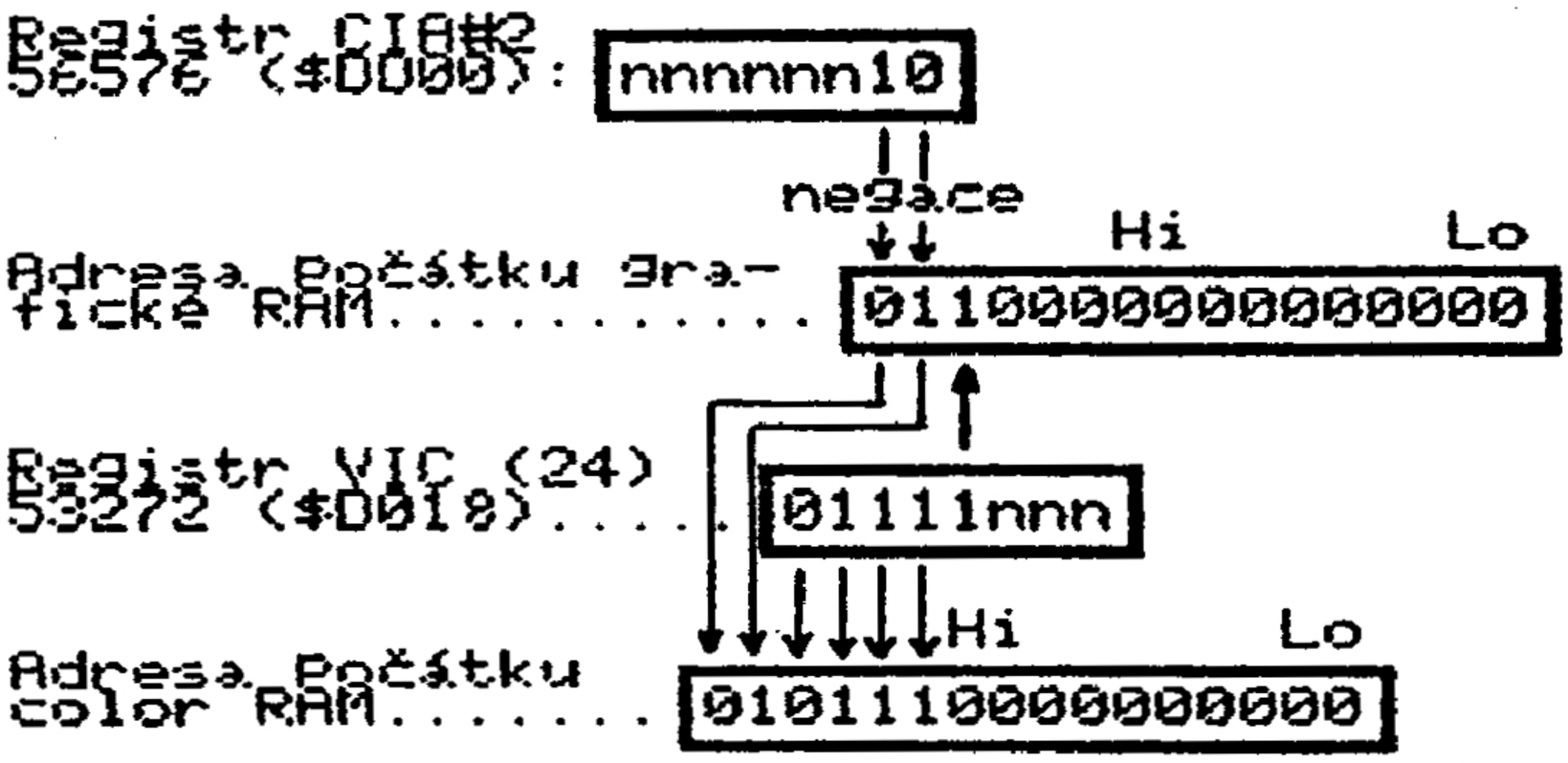
Jemná grafika,barvy

C=64 má standardní jemnou grafiku 320(bodů/řádek)x
x200(bodů/sloUPec). Grafická Paměť RAM by tedy za-
bírá (320/8)x8x25=8000 Bytů.Jeden bod na obrazovce
tedy odPovídá jednomu zaPnutému bitu. Jednotlivé
Byty jsou usPořádány stejně, jako ve znakové ROM-
tedy Po osmi v Pixelu a jednotlivé Pixely jsou se-
řazeny Po obrazovkových řádkách.Ostatně budete mít
možnost ověřit si to na některém z Příkladů.

Podívejme se, jak lze přepočítat souřadnice bodu X[x,y] na jednotlivé Byty.Mějme tedy bod X popsaný souřadnicemi x(0...319) a y(0...199).Pak platí, že RADEK=INT (y/8) : PIXEL=INT (x/8) : LINKA=y AND 7: BIT=7-(x AND 7) : POCATEK= (počátek grafické RAM) a pokud chceme bod zobrazit, uděláme to takhle:

```
BYTE= POCATEK + RADEK*320 + PIXEL*8 + LINKA
POKE BYTE,PEEK(BYTE) OR 2↑BIT.
```

Před tím, než začneme jednotlivé body vynášet je dobré smazat celou grafickou RAM (zapíšeme do ní nuly) a nastavit color RAM. Kde tyto úseky Paměti hledat? Podívejme se na následující obrázek:



(n...nepodstatné pro nastav. grafické a color RAM) Stejně jako jsem to udělal dříve,musím i teď znovu upozornit na to, že VIC je částečně předprogramován.

Grafický režim zapínáme v registru 17 VIC,tedy na adrese 53265 ($D011).Slouží k tomu 5-tý bit, takže to lze provést třeba takto:

```
POKE 53265,PEEK(53265) OR 32.
```

Zpět do textového režimu se dostaneme:

```
POKE 53265,PEEK(53265) AND 223,
```

a zpět do standardního režimu textové a color RAM se dostaneme po přepsání registrů na adresách: 56576 ($DD00) a 53272 ($D018).

Pokud budeme pracovat ve vícebarevném módu, sníží se nám rozlišovací schopnost na 160x200 bodů.Každý může mít jednu ze čtyř možných barev. Bu-

de to záležet na tom, jakou hodnotu má vždy dvojice bitů v Bytu (bráno zleva). Barvy jim přísluší takto

Bity	Grafický režim	Textový režim	Sprites
00	barva Podkladu 53281 ($D021)	barva Podkladu 53281 ($D021)	b. Podkl. 53281
01	horní 4 bity v graf.color RAM	barva #1 53282 ($D022)	multic. #0 53285
10	dolní 4 bity v graf.color RAM	barva #2 53283 ($D023)	b. Sprite (r. 39-46)
11	dolní 4 bity z text.color RAM	dolní 4 bity z text.color RAM	multic. #1 53286

Protože grafické režimy jsou si velmi podobné, uvedl jsem potřebné údaje i pro textový režim a pro Sprity. Kódy barev jsou dokonce stejné:

0 ... černá
1 ... bílá
2 ... červená
3 ... zelenomodrá
4 ... purpurová
5 ... zelená
6 ... modrá
7 ... žlutá
8 ... oranžová
9 ... hnědá
10 ... světle červená
11 ... šedá 1
12 ... světle zelená
13 ... světle modrá
14 ... šedá 2
15 ... šedá 3

Odlišnosti jsou v zapínání vícebarevných módů. V jemné grafice se to provádí 4-tým bitem registru na adrese 53270 ($D016). Tedy např. takto:
POKE 53270,PEEK(53270) OR 16.
V textovém režimu se to provádí stejně. Dokonce můžeme použít i tzv. rozšířený barevný mód, který se zapíná 6-tým bitem registru na adrese 53265($D011) a kód barvy se ukládá do registru na adrese 53284 ($D024).

Barevný mód lze zapínat pro jednotlivé Sprity v registru 28 ($1C), tedy na adrese 53276 ($D01C) a to tak, že nejvyšší bit odpovídá Spritu 7, nejnižší Spritu 0. A tím se dostáváme ke Spritům.

Sprite

Další přednostní našeho počítače jsou tzv. Sprity (česky něco jako duchové, skřítci nebo víly). Sprite je opravdu malý skřítek, kterého si můžeme zobrazit na obrazovce v textovém i grafickém režimu. Jak bude vypadat, to záleží jen na tom, jakými daty ho naplníme. Tento skřítek se umí zvětšovat (2x), dokáže se schovat za okraj nebo znak a ještě další věci.

Spritů máme celkem osm a každý můžeme ovládat samostatně.Každý Sprite má šířku 3x8 bitů a délku 21 bitů.Data se ukládají po jednotlivých řádkách a to zleva.Tedy nejPrve Byte,který je vlevo nahoře, Pak Byte Prostřední a nakonec ten vPravo nahoře. Tím jsme uložili První řádku. Stejně uložíme i všechny ostatní;celkem to tedy bude 3x21=63 Bytů.Každý bit jednotlivých Bytů odPovídá jednomu rozsvícenému bodu na obrazovce. Zkuste schválně trochu pozměnit data Pro myšku (viz. Příklady Programů) a hned vám bude vše jasné.

Máme tedy hotová data Pro nějakého skřítka, ale kam s nimi? V registru 56576 ($DD00) nastavujeme-v negaci- jednu ze čtyř bank,se kterou je VIC schoPen komunikovat. Paměť se tedy v Podstatě rozPadne na čtyři 16KB-tové části.Ve standardním módu jsou v registru $DD00 na příslušných bitech zaPsány hodnoty 1 a 1.Což Po negaci dává hodnoty 0 a 0, tedy Pracujeme v bance 0, která začíná na adrese 0 a končí na 16383 (banka 1 začíná 16384 a končí na 32767,atd.). Data Pro jednotlivé "skřítky" můžeme ukládat kdekoliv uvnitř nastavené banky. A teď musíme dát video-obvodu informace,kde data leží.Provedeme to záPisem do registrů, které leží o 1016 B dále, než Počátek obrazové RAM. Těchto registrů je celkem 8 (Pro každý Sprite jeden) a jdou za sebou, takže První je registr Pro Sprite 0. Do tohoto registru zaPíšeme,ve které 64-ci od Počátku data leží (0-255).Podívejme se na Příklad: data chceme umístit ve standardním módu do kazetového buffru a to od adresy 832.Pracujeme v bance 0,tedy od adresy 832 budeme ukládat 63 dat (+1 nadbyteč.) do 13-té 64-ce.Data jsou určena Pro Sprite 0, takže tuto informaci zaPíšeme do registru na adrese:
Počátek obraz.RAM (1024) + 1016 = 2040 (na 2041 je registr Pro Sprite 1 atd.).

A teď jak dostaneme Sprity na obrazovku. Není to vůbec nic složitého,Protože tam jsou Pořád. Jde

jen o to,kde a v jakém stavu.Podívejme se proto na ty registry VIC,které mají co dělat se SPrity:

0 (53248)...x-ová souřadnice levého horního rohu SP 0 (je-li x<24,je SP0 za l.okrajem)

1 (53249)...x-ová souřadnice levého horního rohu SP 0 (je-li y<50, nebo y>229, je SP 0 schován za okrajem)

2,4,6,8,10,12,14...x-ové souřadnice SP 1 - SP 7

3,5,7,9,11,13,15...y-ové souřadnice SP 1 - SP 7

16 (53264)...8-mý bit x-ové souřadnice SP 0 - SP 7 (je-li x=256,zapne se příslušný bit v tomto registru a x=x-256)

21 (53269)...zapínání SP 0 - SP 7 (bit 7 odpovídá SP 7,bit 6 SP 6,atd.)

23 (53271)...zvětšení jednotl.Sprity 2x ve směru osy y (bit 7...SP 7,atd.)

27 (53275)...Priorita Sprite/Pozadí (bit 7-SP7,..)

28 (53276)...vícebarevný mód

29 (53277)...zvětšení Sprity 2x ve směru osy x

30 (53278)...kolize (dotýkání,překrývání) Sprity (bit 7...SP 7,atd.)

31 (53279)...kolize Sprite/Pozadí

37,38.....barva 0,1 ve vícebarevném módu

39-46.....barva Sprity 0 - 7.

Teď,když už si tykáme se Sprity,se můžeme podívat ještě na další registry VIC:

17...dolní 3 bity slouží k zarolování části okraje přes užitečnou plochu obrazovky (ve směru y)

22...totéž jako 17,ale ve směru osy x

Pak tu máme registry rastru a přerušení.Jsou velice zajímavé. Dávají nám informaci o chodu paprsku na obrazovce. Jejich použitím lze dosáhnout velmi pěkných (někdy i užitečných) obrazových efektů.Jako příklad uvedu režim,kdy horních,řekněme dvacet, řádků slouží pro grafiku a dolních pět pro text. Text je přenášen VIC-em přímo ze znakové RAM.

Zvuk

C=64 obsahuje speciální zvukový obvod (SID), který je schopen provést tříhlasou syntézu a další věci. Registry 0-6 jsou pro první hlas, 7-13 pro druhý a 14-20 pro třetí. Registr 0 leží na adrese 54272. V registru 0,1 (7,8 a 14,15) nastavujeme výšku tónu, neboli frekvenci (Lo,Hi). Kmitočet je získáván dělením jisté základní frekvence, proto pokud chceme vyluzovat tóny,musíme si k nim najít příslušné Lo, Hi hodnoty.

Registry 2,3 (9,10 a 16,17) používáme jen pro pravoúhlý signál.Nastavuje se v nich střída Lo,Hi, přičemž pro Hi je určena jen dolní polovina bytu.

V registru 4 (11 a 18) nastavujeme typ průběhu a další parametry.Průběhy můžeme generovat čtyři: šum,Pravoúhlý,Pilovitý a trojúhelníkový. Další bity slouží k ovládání oscilátoru a ksynchronizaci (popř: modulaci) s jiným hlasem.Bitem 0 startujeme cyklus ADS (=1) a R (=0). Je to vlastně spouštění. Ještě k tomu, co je to ADSR.Zvuk se skládá nejen z tónu,ale i z intenzity signálu.Attack je náběh,tedy čas, který uplyne od startu k dosažení max. intenzity zvuku. Decay -pokles- je čas, který uplyne od okamžiku, kdy zvuk dosáhne max. intenzity do okamžiku,kdy intenzita dosáhne jisté stálé hodnoty. Jak dlouho v ní setrvá,to říká Sustain.Po přepnutí GATE zpět do nuly se spustí Release, tedy doznění. Tyto hodnoty se zapisují vždy po dvou do registrů 5,6 (12,13 a 19,20) podle této tabulky:

Hodnota	A[ms]	D,R[ms]	Hodnota	A[ms]	D,R[ms]
0	2	6	8	100	300
1	8	24	9	250	750
2	16	48	10	500	1500
3	24	72	11	800	2400
4	38	114	12	1000	3000
5	56	168	13	3000	9000
6	68	204	14	5000	15000
7	80	240	15	8000	24000

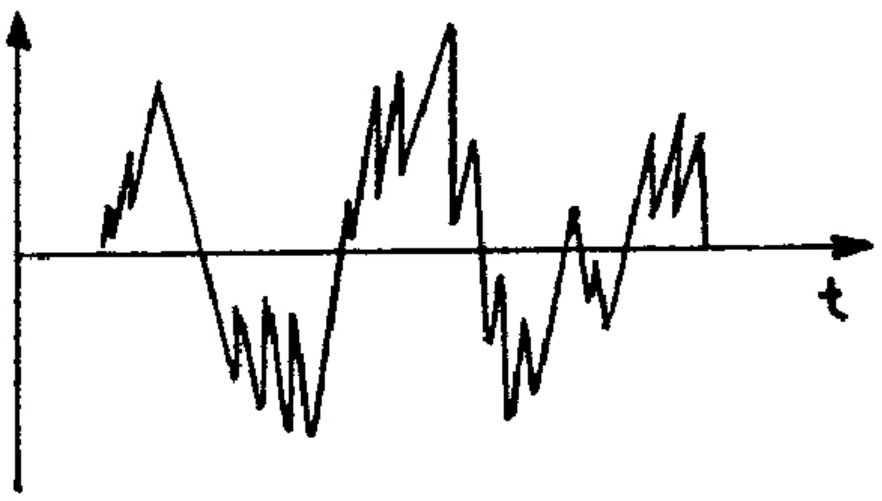
Do registrů 21,22 zapisujeme Lo,Hi mezní frekvenci

filtru, přičemž pro Lo jsou vyhrazeny pouze dolní tři bity. Filtry se do jednotlivých hlasů zařazují v registru 23.Třetí bit slouží k zařazení filtru do vnějšího signálu. Horní čtyři bity slouží k nastavení rezonance od minima po maximum.

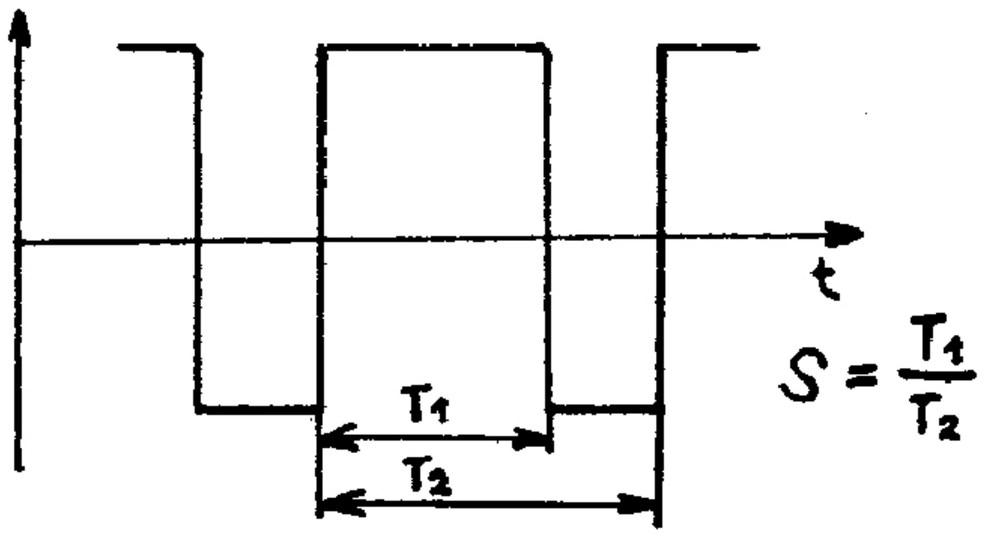
Bez registru 24 bychom se neobešli. Dolními čtyřmi bity řídíme hlasitost zvuku a horní čtyři slouží k výběru filtrů (horní propust,dolní,pásmová) a k vypínání hlasu 3.Ostatní registry viz.Systémové proměnné.

Podívejme se, jak vypadají průběhy jednotlivých typů signálů:

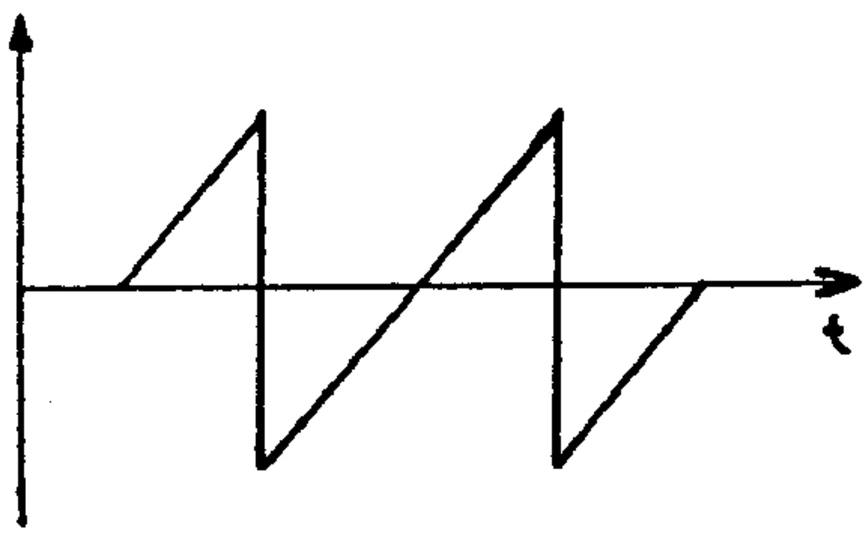
a)šum



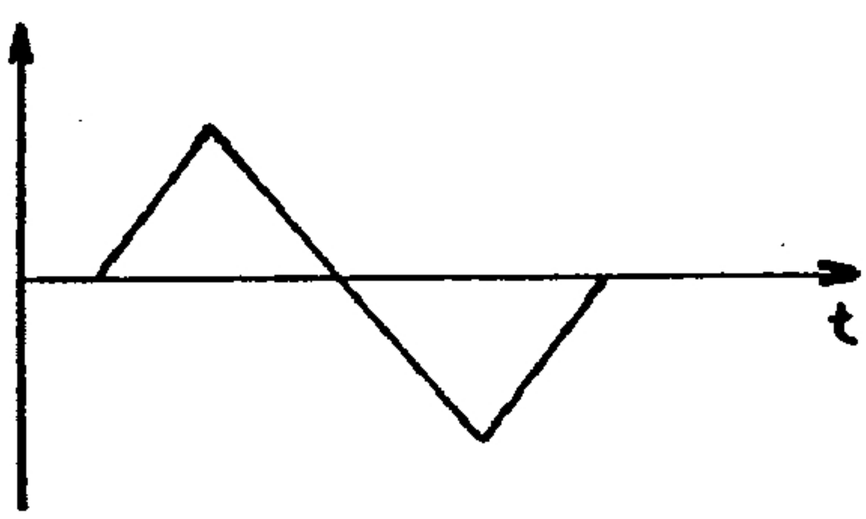
b)pravoúhlý



c)pilovitý



d)trojúhelníkový



Práce se zvukem nepatří mezi nejsnadnější, zejména pro nesnadné pořizování dat. Existují však různé programy,které tuto práci usnadňují.Nejlepší jsou ty, které digitalizují signál přicházející na vstup A/D převodníku a vytvoří program, který po spuštění vydává zvuky podobné digitalizovaným.

Příklady programů na obsluhu obrazu a zvuku

Tato podkapitola přinášší několik programů,které by měly sloužit především jako příklady. Ukazují, jak co nastavit, abychom dosáhli požadovaných efektů. Jsou to:

1)Barevné znaky

Krátký program,který ukazuje,jak se uplatní vícebarevný mód v textovém režimu.Myslím,že stojí za vyzkoušení,Protože uvidíte nevídané.

2)Nový znak

Tento program překopíruje část standardní znakové sady a znak s Poke kódem 0 ("zavináč") zamění za čtvereček.

3)Nový kurzor

Kdo by ho nechtěl,krásný, usměvavý a okatý kurzor. Tento program ho dokáže udělat za několik málo vteřin. V podstatě však jde opět o změnu jednoho znaku ve znakové sadě...

4)Sinusoida

Tento program nám předvede, jak můžeme v jemné grafice nakreslit graf funkce sinus. Za pozornost stojí umístění grafické a color RAM v paměti.

5)Kružnice

V jemné grafice dokážeme nakreslit i kružnici Abychom nemuseli uchovávat data pro každý bod bod kružnice, budeme jí kreslit opět pomocí funkcí.

6)Vícebarevná jemná grafika.

Tento program má za úkol předvést, jak funguje vícebarevná jemná grafika. Doporučuji provést s tímto programem různé pokusy.Bude se to hodit.

7)Tancující myška.

Viděli jste už někdy,jak tancuje malá myška?Tak to máte jedinečnou příležitost. Myška, která se schovává v datech pro Sprity totiž dokáže tancovat i podle hudby.Nevěříte? Tak si to zkuste! Uvidíte (a uslyšíte), že je to pravda.

8)Muzika

Jak používat zvuk, to vám předvede třeba tento, mimochodem poměrně zajímavý,Program.Trocha hudby jistě zalahodí i vašemu uchu.

9)The Show

Velká show je na obzoru! Stačí správně přepsat tento Program a zadat příkaz RUN.Ještě před tím si však nezapomeňte připravit svou Hi-fi aparaturu.

10)Tajemný hrad v Karpatech

Jste-li starší 15-ti let a chcete-li se trochu bát - pak právě pro vás je určen tento Program. Ale je to opravdu jen pro silné povahy.

11)Sprite editor

Pokud nevíte,jak co nejrychleji zpracovat datařádky pro naše skřítky -Spritíky, pak jistě využijete služeb tohoto Programu. Program myslím nepotřebuje další komentář.

Za jednotlivými Programy je vynecháno trochu místa na vaše Poznámky a já pevně věřím,že ho využijete.

```
10 rem vicebarevny mod znaku
19 rem nastaveni kodu barev
20 poke 53281,7
30 poke 53282,3
40 poke 53283,8
49 rem nastaveni vicebarevneho modu
50 poke 53270,peek(53270)or16
60 print"■ahoj                ■  kod:"
70 for x=0to15:fory=0to3
80 poke 55296+y,x
90 next
100 print"■■■
```

Novy znak

```
2 rem zmena znaku s Poke kodem 0
5 Kodznaku=0:rem kod znaku,ktery zmenime
9 rem vypnuti prusovaciho casovace
10 Poke56334,Peek(56334)and254
19 rem prepnuti na znakovou rom
20 Poke1,Peek(1)and251
29 rem preneseni znakove rom
30 for i=0 to 512*8-1
40 Poke i+28672,Peek(i+53248)
50 next
59 rem prepnuti pameti na obvody vstupu/vystupu
60 Poke 1,Peek(1)or4
69 rem zapnuti preruseovaciho casovace
70 Poke56334,Peek(56334)or1
79 rem nastaveni konce pameti pro basic
80 Poke56,108:Poke52,108:clr
89 rem preneseni dat na misto zvoleneho znaku
90 for i=Kodznaku*8+28672 to Ko*8+28672+7
100 read l:Pokei,l
110 next
119 rem nastaveni mista,kde je nova zn. sada
120 Poke56576,(Peek(56576)and252)or2
130 Poke53272,188
140 Poke648,108:Print"A";"B"
149 rem data znaku
150 data 255,129,129,129,129,129,129,255
```

Pozn.

Novy Kurzor

```
2 rem novy kurzor
5 printchr$(142):rem velka. pismena+graf.znaky
9 rem nastaveni konce pameti pro basic
10 poke52,47:poke56,47:clr
19 rem vypnuti prerusovaciho casovace
20 poke56334,peek(56334)and254
29 rem zarazeni znakove rom
30 poke1,peek(1)and251
39 rem prenos poloviny znakove sady
40 fori=0to2047
50 pokei+12288,peek(i+53248)
60 next
69 rem zarazeni obvodu vstupu a. vystupu
70 poke1,peek(1)or4
79 rem zapnuti prerusovaciho casovace
80 poke56334,peek(56334)or1
89 rem prenos dat novych znaku
90 for i=0to7:read a.
100 rem normal:pokei+13568,a.:pokei+14080,a.
109 rem inverzni
110 poke i+13568,255-a.:poke i+14080,255-a.
120 next
130 poke53272,(peek(53272)and240)+12
140 print"■":end
150 data 60,66,165,129,165,153,66,60
```

Pozn.

Sinusoida

```
2 rem Kresleni grafu funkce sinus
4 rem mala pismena,nastaveni konce pameti
5 printchr$(14):poke52,31:poke56,31:clr
10 print"■■■■ Rezim: <N>ormal  -  <I>nverzni"
20 print"■■■■■■■■■■■■■■■■";:gosub1000
30 if rezim<>111then60
40 print"■■■■■■■■■■■■■■■■■■■■■■■■■■■■";:gosub1000
50 if rezim=111 then 10
60 poke204,1:print"■"
70 pocatek=8192:kodbarvy=1:rem poc.graficke ram
80 poke53272,peek(53272)or8
85 poke 53265,peek(53265)or32
90 for i=pocatek to pocatek+7999:rem nulovani
100 poke i,rezim
110 next
120 for i=1024 to 2023
130 poke i,kodbarvy
140 next
150 for x=0 to 319 step2 :rem vypocet hodnot fce
160 y=int(100+80*sin(x/10))
170 byte=pocatek+int(y/8)*320+int(x/8)*8+(x and 7)
180 bit=7-(x and 7)
190 z=1:if peek(byte)=0 then z=-1
200 cislo=rezim-z*((rezim-peek(byte))or(2^bit))
210 poke byte,cislo
220 next
229 rem cekani,nastaveni pocatecnich hodnot
230 poke198,0:wait198,1:poke53272,23:poke53265,27
240 print"■":geta$:end
999 rem cteni rezimu (inverzni,normalni)
1000 poke204,0:poke198,0:wait198,1:geta$
1010 rezim=111
1020 ifa$<>chr$(13) anda$<>"n" anda$<>"i" then1050
1030 if pos(0)>8 and pos(0)<10 then rezim=0
1040 if pos(0)>21 and pos(0)<23 then rezim=255
1050 return
```

Pozn.

Kruznice

```
5 rem Kruznice v jemne grafice
9 rem nastaveni graficke ram
10 po=8192:poke53272,peek(53272)or8:b=0
19 rem zapnuti grafickeho modu
20 poke53265,peek(53265)or32
29 rem mazani grafiky
30 for i=po to po+7999:poke i,0:next
39 rem nastaveni baruy pixelu
40 for i=1024 to 2023:pokei,3:next
47 rem vypocet hodnot kruznice
48 rem b=0...3 a 4 kvadrant
49 rem b=1...2 a 1 kvadrant
50 for x=0 to 160
60 gosub1000
70 xx=x+80
80 zn=int(xx/8):r=int(y/8):ra=y and 7
90 by=po+r*320+zn*8+ra
100 bi=7-(xxand 7)
110 poke by,peek(by)or(2tbi)
120 nextx
125 if b=0then b=1:goto50
129 rem cekani
130 poke198,0:wait198,1:get a$
140 poke53265,27:poke53272,23
150 rem konec
160 print"#":end
999 rem vypocet hodnoty y prisl. kvadrantech
1000 ifb=1then1030
1010 y=100+sqr(160*x-x*x)
1020 return
1030 y=100-sqr(160*x-x*x)
1040 return
```

Pozn.

```
8 rem prepnuti do vicebarevne grafiky
9 rem instalace graficke ram
10 poke56576,198:poke53272,120
20 po=24576:pc=23552
30 poke53265,peek(53265)or32
39 rem nulovani obr. pameti,nastaveni barvy
40 for i=po to po+7999:poke i,0:next
50 for i=pc to pc+999 :poke i,3:next
59 rem preneseni dat krouzku
60 for i=po+8 to po+23:read k:poke i,k:next
69 rem preneseni dat krouzku
70 for i=po+328 to po+343:read k:poke i,k:next
79 rem nastaveni barev
80 poke pc+1,97:poke pc+41,97
90 poke pc+2,33:poke pc+42,33
99 rem cekani na stisk. klavesy
100 poke 198,0:wait198,1:get a$
110 poke 53265,peek(53265)and223
120 poke 53272,20:poke 56576,199
129 rem data krouzku
130 data 7,28,48,96,64,192,128,128
140 data 224,56,12,6,2,3,1,1
150 data 128,128,192,64,96,48,28,7
160 data 1,1,3,2,6,12,56,224
```

Pozn.

```
1 rem tancujici myska
4 rem nastaveni zvuku -sid
5 s=54272:pokes+24,15:pokes,66:pokes+1,3
7 poke s+5,15:pokes+6,215:pokes+7,20:pokes+8,117
10 pokes+12,15:pokes+13,215
14 rem nastaveni obrazu -vic
15 print"■":v=53248:pokev+21,1
19 rem preneseni dat spritu
20 fors1=12288to12350:readq1:pokes1,q1:next
25 fors2=12352to12414:readq2:pokes2,q2:next
30 fors3=12416to12478:readq3:pokes3,q3:next
35 pokev+39,15:pokev+1,68
37 print"■■■■";chr$(14)
38 print"  ■■■■
```

```
10 rem muzika
15 poke 53272,21:rem velka pismena+graf.znaky
19 rem pocatecni adresa zvuk. obvodu -sid
20 s=54272
29 rem nulovani registru
30 for l=s to s+24:poke l,0:next
39 rem nastaveni parametru adsr hlasu 1
40 poke s+5,9:poke s+6,0
49 rem hlasitost -max
50 poke s+24,15
60 b$="■■■■ sum":pr=129:gosub 130
65 b$="■■■■ pravouhly prubeh 1":pr=65
70 poke s+3,1:gosub 130
75 b$="■■■■ pravouhly prubeh 2":pr=65
80 poke s+3,8:gosub 130
85 b$="■■■■ pravouhly prubeh 3":pr=65
90 poke s+3,15:gosub 130
95 b$="■■■■ pilovity prubeh":pr=33:gosub130
100 b$="■■■■ trojuhelnikovy prubeh":pr=17:gosub 13
    0
110 print"■":end
128 rem podprogram pro vyluzovani tonu
130 print b$:gosub 400
139 rem hf-hi kmitocet,lf-lo kmitocet,d-delka
140 read hf,lf,d
150 if hf<0 then 300
160 poke s+1,hf:poke s,lf
169 rem nastaveni typu prubehu a spusteni 'gate'
170 poke s+4,pr
179 rem zpozdovaci smycky
180 for t=1 to d:next:poke s+4,pr-1
190 for t=1 to 50:next:goto 140
200 data 26,20,250,29,69,250
210 data 26,20,250,26,20,250
220 data 29,69,250,26,20,250
230 data 32,219,750,26,20,250
240 data 29,69,250,19,137,250
250 data 19,137,250,19,137,250
260 data 21,237,63,24,157,63
270 data 26,20,750,-1,-1,-1
300 restore
310 print"■■■■■■ soky?";:poke198,0:wait 198,1:ge
    t a$
320 if a$<>chr$(13) and a$<>" " then 130
330 return
400 print"■■■■■■■"tab(240)" filtr: a) ‾‾P  b) __/  c)
"    __/P__  d) P/‾‾"
410 poke 198,0:wait 198,1:get a$:f=1
420 if a$<"a" or a$>"d" then print"■ zadny":f=0
425 print"■ ";a$
430 poke s+22,26
440 if a$="a" then poke s+24,peek(s+24)or16
450 if a$="b" then poke s+24,peek(s+24)or64
460 if a$="c" then poke s+24,peek(s+24)or32
470 if a$="d" then poke s+24,peek(s+24)or(16+64)
```

```
480 Print"■■■ rezonance:  a)min   b)max"
490 Poke 198,0:wait 198,1:get a$
495 if a$=" " then a$="a"
500 if a$<"a" or a$>"b" then 490
505 Print"■ ";a$
510 if a$="a" then Poke s+23, 0*16+f
520 if a$="b" then Poke s+23,15*16+f
530 return
```

Pozn.

```
10 rem the show
19 rem Pocatecni adresa sid
20 s=54272
29 rem nulovani vsech registru
30 for l=0 to 24:Poke s+l,0:next
39 rem strida Pro hlas 1
40 Poke s+3,8
49 rem Parametry adsr hlasu 1
50 Poke s+5,41:Poke s+6,89
59 rem hlas 3-Kmitocet,Prubeh
60 Poke s+14,117:Pokes+18,16
69 rem hlasitost -max
70 Poke s+24,15
80 r=2:Gosub 100:restore:r=.75:Gosub100
85 restore:r=1:Gosub100:Print"■"
90 for x=0 to 500:next:Poke s+24,0:end
99 rem PodProGram Pro hudbu
100 read f,d
110 if f=0 then return
119 rem Prubeh -hlas 1
120 Poke s+4,65
130 for t=1 to d*2
139 rem rozmitani zakladni frekvence Podle r.27
140 fr=f+int(Peek(s+27)/r)
150 hf=int(fr/256):lf=fr and 255
160 Poke s,lf:Poke s+1,hf
170 next
180 Poke s+4,64
190 goto 100
200 data 4817,2,5103,2,5402,2
210 data 8583,4,5407,2,8583,4
220 data 5407,2,8583,12,9634,2
230 data 10207,2,10814,2,8583,2
240 data 9634,4,10814,2,8583,2
250 data 9634,4,8583,12,0,0,0
```

Pozn.

```
10 rem tajemný hrad v karpatech
19 rem pocatecni adresa sid
20 s=54272:print"■"
29 rem nulovani registru
30 for l=0 to 24:poke s+l,0:print"■";:next
39 rem nastaveni kmitoctu hlasu 1
40 poke s+1,104:pokes,78
49 rem nastaveni nabehu a poklesu
50 poke s+5,219
59 rem nastaveni kmitoctu hlasu 2
60 poke s+15,27:poke s+14,160
69 rem hlasitost,prubeh+synchronizace
70 poke s+24,15:pr=19
75 print" *              (jen pro dospele!)";
79 rem opakovani,zmena synchronizace-modulace
80 for l=1 to 8:if l>4 then pr=21
90 poke s+4,pr:for t=0 to 5000:next
100 poke s+4,pr-1
110 for t=0 to 1000:next:next
120 poke s+24,0
130 print"■":end
```

Pozn.

```
2 rem sprite editor
5 dima(21),b(21),c(21)
10 poke53280,6:poke646,7
20 print"■  sestavovani obrazcu sprite-grafiky"
30 print"  ------------------------------------"
40 print"■ na obrazovce se vam objevi obdelnik ,"
50 print" ktery predstavuje zvetsenou plochu    "
60 print" sprite-obrazce . nejprve vytvorte svuj"
65 print" obrazek : pomoci ovladace (port-2)    "
70 print" si vyhledate bod obdelnika , ktery    "
75 print" chcete zakreslit , pak stisknete      "
80 print" tlacitko ovladace . pokud se vam      "
85 print" urcity bod nelibi , stisknete <shift>."
90 print" mozno pouzivat <shift lock,ctrl,com>. "
95 print" jestlize je obrazec hotov , stisknete "
96 print" <return> . chvili pockejte a pak se   "
97 print" na obrazovze vypise sloupec cisel ,"
98 print" ktery predstavuje vas obrazec . "
99 print" pokud budete chtit , tento program"
100 print" vam vytvori soubor data-radku , ktere"
101 print" si muzete nahrat ."
110 print"■ zvolte si : 2-barevny (■2■)"
115 print"                 4-barevny (■4■) obrazec - "
    ;
120 poke204,0
130 getl$
140 ifl$="2"then300
150 ifl$="4"then700
160 goto130
200 print"■"
210 print"  ▒▒▒
```

```
470  a=1106-40+ii*40:a(ii)=0
480  b=a+8:b(ii)=0
490  c=b+8:c(ii)=0
500  forjj=7to0step-1
502  ifpeek(a+7-jj)=32thend=0:goto510
504  d=1
510  a(ii)=a(ii)+d*2↑jj
520  nextjj
530  forjj=7to0step-1
532  ifpeek(b+7-jj)=32thend=0:goto540
534  d=1
540  b(ii)=b(ii)+d*2↑jj
550  nextjj
560  forjj=7to0step-1
562  ifpeek(c+7-jj)=32thend=0:goto570
564  d=1
570  c(ii)=c(ii)+d*2↑jj
580  nextjj
590  nextii
595  print"■                                                    "
596  print"■■■
```

```
910 b=0
920 getl$:ifl$=""then810
930 ifl$=chr$(13)thengoto1000
940 ifl$="*"thend=42
950 ifl$="="thend=61
960 ifl$="/"thend=47
970 ifl$=" "thend=32
980 goto810
1000 Print"■ Prenasim obrazec do Pameti"
1010 forii=1to21
1020 a=1106-40+ii*40:a(ii)=0
1030 b=a+8:b(ii)=0
1040 c=b+8:c(ii)=0
1050 forjj=7to0step-2
1060 ifpeek(a+7-jj)=32thend1=0:d2=0
1061 ifpeek(a+7-jj)=42thend1=1:d2=1
1062 ifpeek(a+7-jj)=61thend1=1:d2=0
1063 ifpeek(a+7-jj)=47thend1=0:d2=1
1070 a(ii)=a(ii)+d1*2↑jj+d2*2↑(jj-1)
1080 nextjj
1090 forjj=7to0step-2
1100 ifpeek(b+7-jj)=32thend1=0:d2=0
1101 ifpeek(b+7-jj)=42thend1=1:d2=1
1102 ifpeek(b+7-jj)=61thend1=1:d2=0
1103 ifpeek(b+7-jj)=47thend1=0:d2=1
1110 b(ii)=b(ii)+d1*2↑jj+d2*2↑(jj-1)
1120 nextjj
1130 forjj=7to0step-2
1140 ifpeek(c+7-jj)=32thend1=0:d2=0
1141 ifpeek(c+7-jj)=42thend1=1:d2=1
1142 ifpeek(c+7-jj)=61thend1=1:d2=0
1143 ifpeek(c+7-jj)=47thend1=0:d2=1
1150 c(ii)=c(ii)+d1*2↑jj+d2*2↑(jj-1)
1160 nextjj
1170 nextii
1180 goto595
1200 Print"■ sestavujj data-radky                                    "
1209 Poke10000,0
1210 a=10001
1220 forii=1to21
1230 b=a+17
1240 Pokea,b-256*int(b/256)
1250 Pokea+1,int(b/256)
1260 Pokea+2,ii
1270 Pokea+3,0
1280 Pokea+4,131
1290 a$=right$("    "+str$(a(ii)),3)
1300 Pokea+5,asc(left$(a$,1))
1310 Pokea+6,asc(mid$(a$,2,1))
1320 Pokea+7,asc(right$(a$,1))
1330 Pokea+8,44
1340 a$=right$("    "+str$(b(ii)),3)
1350 Pokea+9,asc(left$(a$,1))
1360 Pokea+10,asc(mid$(a$,2,1))
1370 Pokea+11,asc(right$(a$,1))
1380 Pokea+12,44
1390 a$=right$("    "+str$(c(ii)),3)
1400 Pokea+13,asc(left$(a$,1))
```

```
1410 Pokea+14,asc(mid$(a$,2,1))
1420 Pokea+15,asc(right$(a$,1))
1430 Pokea+16,0
1440 a=a+17
1450 nextii
1460 Pokea,0
1470 Pokea+1,0
1480 Print"■■■
```

KAPITOLA IV

Vzdělání .je to co zůstane,
když člověk zapomene všechno,
co se naučil.
(Burrhus F. Skinner, 1964)

Nic mi nepřipadá pošetilejší
než se domívat,
že by se myšlenka někoho .jiného
mohla krýt s mou vlastní.
(Louis Aragon, 1964)

Mozek .je nástroj,
.jímž myslíme,
že myslíme.
(Julian Tuwim, 1894 - 1964)

1)Programování ve strojovém kódu

2)Popis registrů;značky

3)Popis instrukcí

4)Popis zabudovaných rutin

5)Příklady programů v Assembleru

Programování ve strojovém kódu

Základem počítače C=64 je mikroProcesor Motorola, 6510.A pokud si někdo myslí, že když pracuje v Basicu,tak ho vlastně může vyndat ze soklu a za tvrdé komusi prodat, pak je na omylu.Procesor zkrátka potřebujeme. Basicovský překladač totiž překládá náš program v Basicu do kódů, kterým rozumí procesor. Stejně tak,ale můžeme napsat program v těchto kódech my. Jaké má tento způsob programování přednosti a nedostatky je zřejmé.

Programujeme-li ve strojovém kódu, tedy v kódu, kterému přímo rozumí procesor,pak vlastně překladač Basicu nepotřebujeme.Náš program bude běhat podstatně rychleji, než by běhal v Basicu, ale na druhé straně přijdeme o veškerý komfort, který nám Basic nabízí. A teď si vyberte. Výběr nám naštěstí trochu ulehčí požadavky praxe. Podívejte se třeba na programy z minulé kapitoly.Vždy, když jsme přepínali do jemné grafiky,bylo potřeba grafickou RAM vynulovat. A šlo to setsakramentsky pomalu. Pokud bychom udělali program na nulování grafické RAM ve strojovém kódu,je celé nulování záležitostí zlomku vteřiny.

V programu "Sinusoida." jsme počítali hodnoty funkce sinus, které jsme pak vynášeli v patřičném měřítku na obrazovku. Numerické výpočty (a výpočty hodnot funkcí zvláště) patří k nejzdlouhavějším operacím v Basicu.Ale když uvážím jak dlouho by mi trvalo,než bych vymyslel program svůj ve strojovém kódu,tak tuto práci rád přenechám Basicu. A i kdybych ten program udělal,zjhy bych zjistil, že zase (při stejné přesnosti) tak o moc rychlejší není.Je však ještě jedno řešení: Použití rutiny zabudované v překladači,ale to jsme se dostali moc daleko.Tušíme tedy, kdy budeme používat programy v Basicu a kdy naopak ve strojovém kódu.

Programování ve strojovém kódu je poměrně náročné na čas a bezchybnost (a tím i na hlavu).Nic-

méně mám takový pocit, že by si to měl každý několikrát zkusit. Kódy se ukládají za sebou do volné oblasti RAM příkazem POKE, čtou se příkazem PEEK a program se spouští příkazem SYS [počátek].Na konec programu dejte instrukci RTS (návrat). A až vám to celkem půjde, opatřete si nějaký monitor. Program monitor,např. SUPERMON 64,MONITOR C=64 apod..Tento program nám podstatně ulehčí programování. Za prvé totiž nemusíme psát program v kódech, ale přímo v jednotlivých instrukcích (nemusíme hledat kódy-minimální riziko chyby) a za druhé nám taky něco nabízí, co by se mohlo hodit. Je to například přenos úseku paměti jinam,ukládání programů na pásku,nebo disketu, výpis úseku paměti v datech, nebo v instrukcích atd..Monitory obvykle neumějí pracovat v desítkové soustavě.Místo toho používají šestnáckovou,která je, jak brzy zjistíte,velmi výhodná. Jde jen o zvyk. Jediné co se nesmí je psát program do některých systémových proměnných,obvodů vst./výst. a pak do samotného monitoru. Delší verze vás na to upozorní, kratší se v lepším případě zhroutí a v horším dělají přesně to co nejvíc nechceme.

Popis registrů; značky

Procesor 6510 má několik registrů. K čemu slouží a jak?Jak je budeme značit? Na podobné otázky se nyní pokusíme odpovědět.

A...akumulátor (8b).Hlavní registr Procesoru.

X...indexový a Pomocný registr (8b)

Y...druhý indexový a Pomocný registr (8b)

SR...stavový registr. Jednotlivé bity mají tento význam:

0...Carry (Přenos).Nastavuje se do 1,jestliže hodnota Přetéče 8bitu a0d.

1...Zero (nula). Je nastaven, jestliže je výsledek Po operaci nulový a0d.

2...Irq (Požadavek na přerušení). Je-li tento bit nastaven,je Přerušení zakázáno. Jinak je Povoleno

3...Decimal mode (desítkový mód).Pokud je nastaven,pracujeme v BCD módu.Jinak v Bin. Př.:%11010011=10E v binárním módu, nebo 69 v decimálním -8b číslo je Prezentováno jako dvě 4b v kódu BCD

4...Break (Přerušení).Je-li nastaven,pak ukazuje, že došlo k Programovému Přerušení (BRK)

5...nekomentován

6...V overflow (Přetečení).Je-li nastaven,Pak došlo k Přenosu ze šestého bitu

7...Negative (záporný).Tento bit kopíruje bit 7 čísla.

SP...ukazatel zásobníku Procesoru (8b). Zásobník Procesor zaPlňuje odshora,takže je-li prázdný,má SP hodnotu $FF. Ukazuje vždy na místo, kam bude Procesor ukládat další hodnotu

PC...Programový čítač. Obsahuje aktuální adresu Paměti.V tomto místě Procesor Pracuje.Má 16b rozložených do dvou Bytů (Lo,Hi).

Tak to bychom měli registry.Podívejme se nyní blíže na instrukci LDA (načtení hodnoty do A).V monitoru ji můžeme zapsat různými zPůsoby,s různým významem (obdobně to Platí i Pro ostatní instrukce):

LDA#dd...načtení šestnáctkové hodnoty dd do A

LDA a.a ...načtení obsahu Paměťové buňky o adrese a.a (v nulové stránce)

LDA a.a,X..načtení obsahu Paměťové buňky o adre-

se a.a+X (Pouze v nulové stránce)

LDA a.a.a.a. .načtení obsahu Paměťové buňky o adrese a.a.a.a (kdekoliv v Paměti)

LDA a.a.a.a,X. . .načtení obsahu Paměťové buňky z adresy a.a.a.a+X (s Přenosem do vyššího Bytu)

LDA a.a.a.a,Y. . .totéž jako LDA a.a.a.a,X

LDA (a.a,X). . .načtení hodnoty z adresy,na kterou ukazuje vektor uložený na a.a+X a a.a+1+X (Lo,Hi)

LDA (a.a),X. . .načtení hodnoty z adresy,na kterou ukazuje vektor uložený na adresách [a.a (Lo) , a.a+1 (Hi)]+X

Ještě jednou PřiPomínám,že Při indexovaném načítání z nulové stránky (Zero Page) zůstaneme stále v nulové stránce.Vyšší (Hi) Byte je totiž vždy nulový. NaProti tomu Při indexovaném čtení z libovolné části Paměti se Přenos do vyššího Bytu uPlatňuje.

Ještě se zmíním o tom,co se stane,je-li v registru, řekněme A, uloženo číslo $FF a Přičteme k němu jedničku. $FF je decimálně 255,tedy maximální hodnota, kterou lze na 8 bitů zaPsat. Po Přičtení jedničky bude v registru hodnota následující, tedy $00.Přitom se nastaví Příznaky Z (a C).

A na konec se Podívejme, jak vyPadají některé Příznaky Po Porovnání (R...registr, který Porovnáváme s oPerandem O):

Stav	Z	N	C
R>O	0	0	1
R=O	1	0	1
R<O	0	1	0

(Tato tabulka platí Pro Porovnávání obecně, ne jenom Pro R-O)

ADC (add with carry) součet s přenosem

Zápis	Kód	Byty	Cykly
ADC #dd	69	2	2
ADC a.a.	65	2	3
ADC a.a.,X	75	2	4
ADC a.a.a.a.	6D	3	4
ADC a.a.a.a.,X	7D	3	4+1*
ADC a.a.a.a.,Y	79	3	4+1
ADC (a.a.,X)	61	2	6
ADC (a.a.),Y	71	2	5+1

NV-BDIZC
SR: **....**

A+M+C → A,C

*..Počet cyklů závisí na tom, jestli přecházíme do nové str.

AND logický součin

Zápis	Kód	Byty	Cykly
AND #dd	29	2	2
AND a.a.	25	2	3
AND a.a.,X	35	2	4
AND a.a.a.a.	2D	3	4
AND a.a.a.a.,X	3D	3	4+1
AND a.a.a.a.,Y	39	3	4+1
AND (a.a.,X)	21	2	6
AND (a.a.),Y	31	2	5+1

NV-BDIZC
SR: *.....*.

A ∧ M → A

ASL (arithmetic shift left) rotace vlevo

Zápis	Kód	Byty	Cykly
ASL A	0A	1	2
ASL a.a.	06	2	5
ASL a.a.,X	16	2	6
ASL a.a.a.a	0E	3	6
ASL a.a.a.a.,X	1E	3	7

NV-BDIZC
SR: *.....**

C ← [A] ← 0

BCC (branch on carry clear) skok při C=0

Zápis	Kód	Byty	Cykly
BCC rel.a.	90	2	3+1

NV-BDIZC
....

BCS (branch on carry set) skok při C=1

Zápis	Kód	Byty	Cykly
BCS rel.a.	B0	2	3+1

NV-BDIZC
SR: ....

BEQ (branch if equal to zero) skok při Z=1

Zápis	Kód	Byty	Cykly
BEQ rel.a.	F0	2	3+1

NV-BDIZC
SR: ....

BIT (compare memory bits with accumulator)
testování bitů akumulátor-paměť

Zápis	Kód	Byty	Cykly
BIT a.a.	24	2	3
BIT a.a.a.a.	2C	3	4

NV-BDIZC
SR: ab....*.

$\overline{RAM}$,M7→a,M6→b

BMI (branch on minus) skok při N=1

Zápis	Kód	Byty	Cykly
BMI rel.a.	30	2	3+1

NV-BDIZC
SR: ....

BNE (branch on not equal to zero) skok při Z=0

Zápis	Kód	Byty	Cykly
BNE rel.a.	D0	2	3+1

NV-BDIZC
SR: ....

BPL (branch on plus) skok při N=0

Zápis	Kód	Byty	Cykly
BPL rel.a.	10	2	3+1

NV-BDIZC
SR: ....

BRK (break) programové přerušení

Zápis	Kód	Byty	Cykly
BRK	00	1	7

NV-BDIZC
SR: ...1.1..

(PC+2)↓,P↓,PCL ← FFFE,PCH ← FFFF

BVC (branch on overflow clear) skok při V=0

Zápis	Kód	Byty	Cykly
BVC rel.a.	50	2	3+1

NV-BDIZC
SR: ....

BVS (branch on overflow set) skok při V=1

Zápis	Kód	Byty	Cykly
BVS rel.a.	70	2	3+1

NV-BDIZC
SR: ....

CLC (clear carry flag) nulování Příznaku C

Zápis	Kód	Byty	Cykly
CLC	18	1	2

NV-BDIZC
SR: 0

0 → C

CLD (clear decimal mode) nulování Příznaku D

Zápis	Kód	Byty	Cykly
CLD	D8	1	2

NV-BDIZC
SR: ....0...

0 → D

CLI (clear interrupt mask) Povolení přerušení

Zápis	Kód	Byty	Cykly
CLI	58	1	2

NV-BDIZC
SR: .....0..

0 → I

CLV (clear overflow flag) nulování Příznaku V

Zápis	Kód	Byty	Cykly
CLV	B8	1	2

NV-BDIZC
SR: .0......

0 → V

CMP (compare to accumulator) Porovnávání A

Zápis	Kód	Byty	Cykly
CMP #dd	C9	2	2
CMP a.a.	C5	2	3
CMP a.a.,X	D5	2	4
CMP a.a.a.a.	CD	3	4

NV-BDIZC
SR: *.....**

A <> M

CMP aaaa,X	DD	3	4+1
CMP aaaa,Y	D9	3	4+1
CMP (aa,X)	C1	2	6
CMP (aa),Y	D1	2	5+1

CPX (compare to register X) Porovnávání X

Zápis	Kód	Byty	Cykly
CPX #dd	E0	2	2
CPX aa	E4	2	3
CPX aaaa	EC	3	4

NV-BDIZC
SR: *.....**

X <> M

CPY (compare to register Y) Porovnávání Y

Zápis	Kód	Byty	Cykly
CPY #dd	C0	2	2
CPY aa	C4	2	3
CPY aaaa	CC	3	4

NV-BDIZC
SR: *.....**

X <> M

DEC (decrement) odečtení jedničky

Zápis	Kód	Byty	Cykly
DEC aa	C6	2	5
DEC aa,X	D6	2	6
DEC aaaa	CE	3	6
DEC aaaa,X	DE	3	7

NV-BDIZC
SR: *.....*.

M - 1 → M

DEX (decrement X) odečtení jedničky od X

Zápis	Kód	Byty	Cykly
DEX	CA	1	2

NV-BDIZC
SR: *.....*.

X - 1 → X

DEY (decrement Y) odečtení jedničky od Y

Zápis	Kód	Byty	Cykly
DEY	88	1	2

NV-BDIZC
SR: *.....*.

Y - 1 → Y

EOR (exlusive OR)

Zápis	Kód	Byty	Cykly
EOR #dd.	49	2	2
EOR a.a.	45	2	3
EOR a.a.,X	55	2	4
EOR a.a.a.a.	4D	3	4
EOR a.a.a.a.,X	5D	3	4+1
EOR a.a.a.a.,Y	59	3	4+1
EOR (a.a.,X)	41	2	6
CPX (a.a.),Y	51	2	5+1

NV-BDIZC
SR: *.....*.

A eor M → A

INC (increment) přičtení jedničky

Zápis	Kód	Byty	Cykly
INC a.a.	E6	2	5
INC a.a.,X	F6	2	6
INC a.a.a.a.	EE	3	6
INC a.a.a.a.,X	FE	3	7

NV-BDIZC
SR: *.....*.

M + 1 → M

INX (increment X) přičtení jedničky k X

Zápis	Kód	Byty	Cykly
INX	E8	1	2

NV-BDIZC
SR: *.....*.

X + 1 → X

INY (increment Y) přičtení jedničky k Y

Zápis	Kód	Byty	Cykly
INY	C8	1	2

NV-BDIZC
SR: *.....*.

Y + 1 → Y

JMP (jump) skok

Zápis	Kód	Byty	Cykly
JMP a.a.a.a.	4C	3	3
JMP (a.a.a.a.)	6C	3	5

NV-BDIZC
SR: ....

a.a.a.a. → PC,na-
bo M → PCL,A

JSR (jump to subroutine) skok do PodProgramu

Zápis	Kód	Byty	Cykly
JSR a.a.a.a.	20	3	6

NV-BDIZC
SR: ....

PC+2↓, a.a.a.a→PC

LDA (load accumulator) načtení do A

Zápis	Kód	Byty	Cykly
LDA #dd	A9	2	2
LDA a.a.	A5	2	3
LDA a.a.,X	B5	2	4
LDA a.a.a.a.	AD	3	4
LDA a.a.a.a.,X	BD	3	4+1
LDA a.a.a.a.,Y	B9	3	4+1
LDA (a.a.,X)	A1	2	6
LDA (a.a.),X	B1	2	5+1

NV-BDIZC
SR: *.....*.

M → A

LDX (load register X) načtení X

Zápis	Kód	Byty	Cykly
LDX #dd	A2	2	2
LDX a.a.	A6	2	3
LDX a.a.,Y	B6	2	4
LDX a.a.a.a.	AE	3	4
LDX a.a.a.a.,Y	BE	3	4+1

NV-BDIZC
SR: *.....*.

M → X

LDY (load register Y) načtení Y

Zápis	Kód	Byty	Cykly
LDY #dd	A0	2	2
LDY a.a.	A4	2	3
LDY a.a.,X	B4	2	4
LDY a.a.a.a.	AC	3	4
LDY a.a.a.a.,X	BC	3	4+1

NV-BDIZC
SR: *.....*.

M → Y

LSR (logical shift right) log. Posun vPravo

Zápis	Kód	Byty	Cykly
LSR A	4A	1	2

NV-BDIZC
SR: 0.....**

(Pokračování)

LSR a.a	46	2	5
LSR a.a,X	56	2	6
LSR a.a.a.a	4E	3	6
LSR a.a.a.a,X	5E	3	7

0 → [A] → C

NOP (no operation) Prázdná operace

Zápis	Kód	Byty	Cykly
NOP	EA	1	2

NV-BDIZC
SR: ....

ORA (OR with accumulator) OR s akumulátorem

Zápis	Kód	Byty	Cykly
ORA #dd	09	2	2
ORA a.a	05	2	3
ORA a.a,X	15	2	4
ORA a.a.a.a	0D	3	4
ORA a.a.a.a,X	1D	3	4+1
ORA a.a.a.a,Y	19	3	4+1
ORA (a.a,X)	01	2	6
ORA (a.a),X	11	2	5+1

NV-BDIZC
SR: *.....*.

A v M → A

PHA (Push accumulator) ulož A (do zásobníku)

Zápis	Kód	Byty	Cykly
PHA	48	1	3

NV-BDIZC
SR: ....

A↓

PHP (Push Processor status) ulož SR

Zápis	Kód	Byty	Cykly
PHP	08	1	3

NV-BDIZC
SR: ....

SR↓

PLA (Pull accumulator) naplň A ze zásobníku

Zápis	Kód	Byty	Cykly
PLA	68	1	4

NV-BDIZC
SR: *.....*.

A↑

PLP (pull SR from stack) naplň SR ze zásobníku

Zápis	Kód	Byty	Cykly
PLP	28	1	4

SR: ze zásob.

SR↑

ROL (rotate left) rotace vlevo

Zápis	Kód	Byty	Cykly
ROL A	2A	1	2
ROL a.a.	26	2	5
ROL a.a.,X	36	2	6
ROL a.a.a.a.	2E	3	6
ROL a.a.a.a.,X	3E	3	7

NV-BDIZC

SR: *.....**

ROR (rotate right) rotace vpravo

Zápis	Kód	Byty	Cykly
ROR A	6A	1	2
ROR a.a.	66	2	5
ROR a.a.,X	76	2	6
ROR a.a.a.a.	6E	3	6
ROR a.a.a.a.,X	7E	3	7

NV-BDIZC

SR: *.....**

RTI (return from interrupt) návrat z přerušení

Zápis	Kód	Byty	Cykly
RTI	40	1	6

SR: ze zásob.

SR↑,PC↑

RTS (return from subroutine) návrat z podprogr.

Zápis	Kód	Byty	Cykly
RTS	60	1	6

NV-BDIZC

SR: ....

(PC↑)+1

SBC (substract with carry) odečítání s přenosem

Zápis	Kód	Byty	Cykly
SBC #dd	E9	2	2

NV-BDIZC

SR: **....**

$A-M-\overline{C} \rightarrow A$

Zápis	Kód	Byty	Cykly
SBC a.a.	E5	2	3
SBC a.a.,X	F5	2	4
SBC a.a.a.a.	ED	3	5
SBC a.a.a.a.,X	FD	3	4+1
SBC a.a.a.a.,Y	F9	3	4+1
SBC (a.a.,X)	E1	2	6
SBC (a.a),Y	F1	2	5+1

SEC (set carry) nastavení příznaku C

Zápis	Kód	Byty	Cykly
SEC	38	1	2

NV-BDIZC
SR: 1

1 → C

SED (set decimal mode) zapnutí decimál. módu

Zápis	Kód	Byty	Cykly
SED	F8	1	2

NV-BDIZC
SR: ....1...

1 → D

SEI (set interrupt disable) zákaz přerušení

Zápis	Kód	Byty	Cykly
SEI	78	1	2

NV-BDIZC
SR: .....1..

1 → I

STA (store A in memory) ulož A do paměti

Zápis	Kód	Byty	Cykly
STA a.a.	85	2	3
STA a.a.,X	95	2	4
STA a.a.a.a.	8D	3	4
STA a.a.a.a.,X	9D	3	5
STA a.a.a.a.,Y	99	3	5
STA (a.a.,X)	81	2	6
STA (a.a.),Y	91	2	6

NV-BDIZC
SR: ....

A → M

STX (store X in memory) ulož X do paměti

Zápis	Kód	Byty	Cykly
STX a.a.	86	2	3
STX a.a,Y	96	2	4
STX a.a.a.a.	8E	3	4

NV-BDIZC
SR: ....

X → M

STY (store Y in memory) ulož Y do Paměti

Zápis	Kód	Byty	Cykly
STY a.a.	84	2	3
STY a.a,X	94	2	4
STY a.a.a.a.	8C	3	4

NV-BDIZC
SR: ....

Y → M

TAX (transfer A into X) přenes A do X

Zápis	Kód	Byty	Cykly
TAX	AA	1	2

NV-BDIZC
SR: *.....*.

A → X

TAY (transfer A into Y) přenes A do Y

Zápis	Kód	Byty	Cykly
TAY	A8	1	2

NV-BDIZC
SR: *.....*.

A → Y

TSX (transfer SP into X) přenes SP do X

Zápis	Kód	Byty	Cykly
TSX	BA	1	2

NV-BDIZC
SR: *.....*.

SP → X

TXA (transfer X into A) přenes X do A

Zápis	Kód	Byty	Cykly
TXA	8A	1	2

NV-BDIZC
SR: *.....*.

X → A

TXS (transfer X into SP)

Zápis	Kód	Byty	Cykly
TXS	9A	1	2

NV-BDIZC
SR: ....

X → SP

TYA (transfer Y into A) Přenes Y do A

Zápis	Kód	Byty	Cykly
TYA	98	1	2

NV-BDIZC
SR: *.....*.

Z popisu instrukcí vidíme,na co je můžeme používat a jaké příznaky ovlivňují (Podle toho pak větvíme program). Zastavme se teď u některých z nich:

ASL a jiné typy rotací a posunů. Jedná se o posun jednotlivých bitů v daném směru,s tím, že přetékající bit se zapíše do C a na místo uvolněného bitu se uloží buď 0, nebo předchozí stav příznaku C (Podle typu instrukce).

BCC apod. jsou skoky v závislosti na stavu příznaku. Následuje relativní adresa místa, kde bude program pokračovat. Relativní adresa je číslo kladné (7 bit je nulový) nebo záporné (7 b je 1),které se přičítá k programovému čítači. Ten se pak zvětší o 1 a z místa, na které ukazuje budeme pokračovat.

BIT instrukce pro testování bitů A a paměti.Protože neovlivňuje příznak C, používá se jí i trochu jinak.Podívejme se na příklad:

```
.,$1000 BIT$A901 a tatáž instrukce od $1001:
.,$1001 LDA#$01  .
```

CLC a podobné instrukce slouží k nastavení příznaku do požadovaného stavu.

DEC,INC a jiné instrukce tohoto typu nenastavují při přetečení příznak C.Byl by totiž pouze kopií příznaku Z.Je to častá chyba,která se stane i profesionálovi.

JSR při použití této instrukce se nejprve, na rozdíl od JMP,uloží návratová adresa,tedy stav PC +2 do zásobníku.Tato adresa ukazuje těsně před instrukci, která po JSR následuje.Ze zásobníku se vybere instrukcí RTS.

LDA,LDX,LDY Při načítání do registru se stará hodnota přepíše novou,dojde tedy k její ztrátě. S tím musíme počítat. Proto budeme-li ještě starou hodnotu potřebovat, je vhodné jí předem uložit.

NOP této instrukce můžeme používat různě.Něco by o ní možná věděli i programátoři,kteří jsou placení od délky programu. Méně často se používá, potřebujeme-li vyplnit v programu vzniklé místo,nebo pokud program časujeme.

PHA touto instrukcí uložíme do zásobníku kopii A, to znamená, že v okamžiku po proběhnutí této instrukce je stav A uložen v zásobníku,přičemž se A nemění.

PHP tuto instrukci použijeme jednak když chceme uložit SR, nebo v případě, že ho chceme přenést do A.Udělali bychom to takto:

```
.,1000 PHP
.,1001 PLA
```

Poznámky

Popis zabudovaných rutin

V Kernalu je uloženo několik Programů,které se obvykle používají jako PodProgramy.Používá je Basic, ale používat je můžeme i my. Zrovna tak je ovšem můžeme i předělat a na jejich počátek přenastavit příslušný vektor. Tabulka skoků pro jednotlivé rutiny je uložena na konci Kernalu od $FF81 do $FFF5 (včetně). Protože jejich použití je poměrně výhodné,uvedeme si jejich základní popis (v závorce je uvedena adresa podle tabulky skoků).

ACPTR $FFA5 [EE13]:čtení Bytu ze sériové Bus

Používá regis.: A,X

Komunikační r.: A

Zakazuje přerušení,v A je Byte přečtený ze sběrnice.

Příklad:

```
;čtení Byte ze sériové sběrnice
JSR ACPTR
STA DATA
```

CHKIN $FFC6 [(031C)]:příprava soub.pro vstup

Používá regis.: A,X

Komunikační r.: X

Nejprve zjišťuje byl-li již soubor otevřen,jestliže ano, pak přenese aktuální data na $B8,$B9,$BA. Následuje test,může-li být soubor vstupní.

Příklad:

```
;připrav pro vstup logický soubor 2
LDX#$02
JSR CHKIN
```

CHKOUT $FFC9 [(0320)]:příPr. soub. pro výst.

Používá regis.: A,X

Komunikační r.: X

Obdoba CHKIN. Používá některé shodné PodProgramy , jako CHKIN.

Příklad:

```
;připrav pro výstup logický soubor 4
LDX#$04
JSR CHKOUT
```

CHRIN $FFCF [(0324)]:vstuP znaků

Používá regis.: A,X
Komunikační r.: A
Nejprve přečte kód vstupního zařízení.Pokud je Periférie určena pro vstuP, Pak z ní čte data zakončená např. $0D (RETURN).
Příklad:

```
LDY#$00     ;příPrava Y Pro ukládání dat
JSR CHRIN
STA DATA,Y ;uložení dat
INY
CMP#$0D     ;test na RETURN
BNE <JSR CHRIN>  ;není-li,Pak vše oPakuj od JSR...
```

CHROUT $FFD2 [(0326)]:výslání znaku

Používá regis.: A
Komunikační r.: A
Pomocí této rutiny můžeme vyslat na obrazovku,nebo na Periférii kód uložený v A.
Příklad:

```
LDA#$0D     ;kód RETURNu
JSR CHROUT  ;odrolování řádky na obrazovce
```

CIOUT $FFA8 [E0D0]:vyslání znaku na sér. Bus

Používá regis.: A
Komunikační r.: A
Tato rutina obstarává vysílání znaků na sériovou sběrnici.
Příklad:

```
LDA#$41     ;kód znaku "A"
JSR CIOUT   ;vyslání
```

CINT $FF81 [FF5B]:inicializace editoru a VIC

Používá regis.: A,X,Y
Nastaví standardní vstuP a výstuP,nastaví registry $00-$2E VIC (data čte od $ECB9),nastaví standardní grafický mód, blikání kurzoru, dekódovací rutinu klávesnice na $EB47,velikost kl.buffru na $0A, rychlost odečítání znaků, barvu znaku $0286 na $0E, Pomocné ukazatele řádek,další SP editoru,PAL/NTSC, přerušovací registr na adrese $DC0D a další.

Příklad:

```
JSR CINT    ;nastavení editoru a VIC
JMP RUN
```

CLALL $FFE7 [(032C)]:uzavření všech souborů

Používá regis.: A,X

Uzavře všechny soubory a nastaví standardní vstup a výstup (klávesnice,obrazovka). A=$00,X=$03.

Příklad:

```
JSR CLALL
JMP RUN
```

CLOSE $FFC3 [F291]:uzavření daného souboru

Používá regis.: A,X,Y

Komunikační r.: A

Tato rutina uzavře soubor s daným číslem a přerovná tabulku kanálů.

Příklad:

```
;uzavření souboru číslo $15
LDA#$15
JSR CLOSE
```

CLRCHN $FFCC [(0322)]:nastavení stand.in/out

Používá regis.: A,X

Nejprve porovná výstup na $03. Je-li kód Periférie menší,volá UNTALK,jinak UNLSN.Pak nastaví st.vstup (klávesnice) a výstup (obrazovka).

Příklad:

```
JSR CLRCHN
```

GETIN $FFE4 [(032A)]:čtení znaku

Používá regis.: A (X,Y)

Komunikační r.: A

Nejprve zjistí, z jaké Periférie má znak odečítat. Je-li to modem, odečte znak z jeho buffru, přičemž C=0 a Y se zachovává. Je-li vstupem klávesnice, pak odečte znak z jejího buffru. V A a Y je kód tohoto znaku a v X je Počet znaků, které v buffru ještě zbyly. Je-li vstupem něco jiného, pak končí skokem do CHRIN na adresu $F166.

Příklad:

```
;odečtení znaku z klávesice
```

```
JSR GETIN
CMP#$00                 ;klávesa není stisknuta?
BEQ <JSR GETIN>  ;ne,opakuj vše od JSR GETIN
```

IOBASE $FFF3 [E500]:zjistí počátek r. CIA#1

```
Používá regis.: X,Y
Komunikační r.: X,Y
Tato rutina přečte do X a Y  Lo,Hi  adresu počátku
CIA#1 (X=$00,Y=$DC).
```

IOINIT $FF84 [FDA3]:inicializace in/out

```
Používá regis.: A,X,Y
Nastaví přerušovací registr CIA#1, dataPor A atd.,
totéž u CIA#2,vynuluje SID registry,a pak pokračuje na $FDDD (část CINT).
```

LISTEN $FFB1 [ED0C]:nastavení per. na příjem

```
Používá regis.: A
Komunikační r.: A
Tato rutina vyšle na periférii patřičné kódy,podle kterých se nastaví na příjem.
```

Příklad:

```
;nastav periférii s kódem $08 na příjem
LDA#$08
JSR LISTEN
```

LOAD $FFD5 [F49E]:naplnění RAM z periférie

```
Používá regis.: A,X,Y
Komunikační r.: A,X,Y
Podle kódu v A se provádí buď  LOAD  (A=$00)  nebo VERIFY (A=1). Do X,Y ukládáme Lo,Hi adresu možného počátku nahrávání (Pokud nejsou data nahrána v auto-save,tak jistého).Po load tam máme adresu konce nahraných dat.
```

Příklad:

```
;Přečti data z magnetofonu
LDA#$01          ;Periférie s kódem 1-magnetofon
LDX#$00          ;logické číslo souboru
LDY#$00          ;sekundární adresa ($01-nahraj data tam,odkud byla nahrána)
JSR SETLFS       ;rutina pro nastavení $B8,$B9,$BA
LDA#$0F          ;Počet znaků názvu souboru
LDX#<Lo
```

```
LDY#$Hi          ;adresa počátku názvu
JSR SETNAM       ;rutina pro nastavení param. názvu
LDA#$00          ;0...Load
LDX#$01
LDY#$08          ;Počátek ukládání programu
JSR LOAD
STX$2D
STY$2E           ;nastavení konce po load Basicu
JMP RUN
```

MEMBOT $FF9C [FE34]:operace s počát.BasicRAM

Používá regis.: X,Y

Komunikacní r.: X,Y

Je-li C=1,pak se do X,Y přenese adresa počátku Ba-sic RAM $0281,$0282.Je-li C=0,pak se naopak na ta-to místa zapíše obsah X,Y.

MEMTOP $FF99 [FE25]:operace s konc. BasicRAM

Používá regis.: X,Y

Komunikacní r.: X,Y

Tato rutina je naprosto shodná s MEMBOT, jenom s tím rozdílem, že se jedná o konec paměti RAM pro Basic,tedy $0283,$0284.

OPEN $FFC0 [(031A)]:otevření souboru

Používá regis.: A,X,Y

Tato rutina provede otevření požadovaného souboru.

Príklad:

```
LDA#$00          ;žádný název
JSR SETNAM       ;nastavení jména
LDA#$15          ;číslo souboru
LDX#$08          ;kód periférie
LDY#$15          ;sekundární adresa
JSR SETLFS       ;nastavení parametrů souboru
JSR OPEN         ;otevření souboru
```

PLOT $FFF0 [E50A]:Poloha kurzoru

Používá regis.: A,X,Y

Komunikační r.: X,Y

Je-li C=1,pak do X,Y přečte pozici kurzoru $D3,$D6 jinak při C=0 na tato místa naopak uloží obsah X a Y.V X je řádek,v Y je sloupec.

Příklad:

```
;nastavení kurzoru na Pozici $05,$10
LDX#$10
LDY#$05
CLC
JSR PLOT
```

RAMTAS $FF87 [FD50]:inicializace RAM

Používá regis.: A,X,Y

Nuluje oblasti $0002-$0101,$0200-$02FF,$0300-$03FF Pak nastavuje $B2,$B3 na $033C, $C1,$C2 na $0400 a inicializuje tuto oblast,nastaví Poč.a konec Basic RAM,$0288 (obraz).

RDTIM $FFDE [F6DD]:čtení času od zapnutí

Používá regis.: A,X,Y

Komunikační r.: A,X,Y

Zakazuje přerušení.Pak přečte do A $A2, do X $A1 a do Y $A0.Pak přerušení Povoluje.

READST $FFB7 [FE07]:čtení stav.r. in/out

Používá regis.: A

Komunikační r.: A

Tato rutina čte stavový registr obvodů vstupu a výstupu, tedy v Podstatě obsah adresy $0297 (Pro modem),nebo $90 (Pro ostatní).

Příklad:

```
;kontrola na konec během čtení
JSR READST
AND$#40
BNE <KONEC>      ;skok při konci souboru
```

RESTOR $FF8A [FD15]:nastavení vektorů in/out

Používá regis.: A,X,Y

Tato rutina nastaví $C2,$C3 na $FD30 a dále Pak vektory v oblasti $0314-$0333.

SAVE $FFD8 [F5DD]:uložení dat z RAM

Používá regis.: A,X,Y

Komunikační r.: A,X,Y

Tato rutina uloží do Periférie data, která jsou v RAM.Pokud je před zavoláním A nenulové,Pak ukazuje na Lo adresu (za ní je Hi) v nulové stránce, která

ukazuje na počátek nahrávaných dat. Tento mód má tu výhodu, že při LOAD se data nahrají zpět na totéž místo,ze kterého byla zaznamenána (auto-save).

Příklad:

```
;záznam na magnetofon v auto-save
LDX#$01          ;Periférie:magnetofon
JSR SETLFS       ;nastavení kódů pro in/out
LDA#$00          ;žádné jméno
JSR SETNAM
LDA#$2B          ;na $2B,$2C je uložena LO,Hi adresa
                  počátku dat
LDX$2D
LDY$2E           ;Lo,Hi adresa konce dat+1
JSR SAVE         ;uložení
```

SCNKEY $FF9F [EA87]:příprava pro čtení znaku

Používá regis.: A,X,Y

Tato rutina přenastaví SP pro čtení znaku z klávesnice do výchozího stavu (ukazatele SHIFT, dekódovací tabulku,atd.),Pak čte kód stisknuté klávesy a porovnává ho na STOP, nastavuje správnou dekódovací tabulku,dekóduje některé znaky a podle toho o jaké jde buď změní příslušné SP, nebo znak zapíše do buffru. Poměrně často se používá před rutinou GETIN pro vynulování a inicializaci SP.

SCREEN $FFED [E505]:formát textové obrazovky

Používá regis.: X,Y

Komunikační r.: X,Y

Do X načte hodnotu ##28 a do Y ##19. Pak se vrací. Tyto hodnoty odpovídají formátu 40x25.

SECOND $FF93 [EDB9]:vyslání sekundár. adresy

Používá regis.: A

Komunikační r.: A

Tato rutina vyšle sekundární adresu obsaženou v A.

Příklad:

```
;Periférie $08 se sekundární adresou $0F
LDA#$08
JSR LISTEN
LDA#$0F
JSR SECOND
```

SETLFS $FFBA [FE00]:uložení kódu in/out

Používá regis.: A,X,Y

Komunikační r.: A,X,Y

Tato rutina zapíše A (číslo souboru) na $B8, X (kód periférie) na $BA a Y (sekundární adresa) na $B9.

SETMSG $FF90 [FE18]:obsluha hlášek

Používá regis.: A

Komunikační r.: A

Tato rutina zapíše A na adresu $9D.Je-li A=$40,pak se tisknou jen řídící hlášky,A=$80 chybové hlášky, A=$00 nic,hlášky jsou odpojené. Basic je však vždy zapíná!

SETNAM $FFBD [FDF9]:nastavení názvu souboru

Používá regis.: A,X,Y

Komunikační r.: A,X,Y

Po zavolání přepíše A (délka názvu) na $B7, X (Lo adresa počátku názvu) na $BB a Y (Hi adresa počát-ku názvu) na $BC.Používá se před OPEN, LOAD, SAVE apod.

SETTIM $FFDB [F6E4]:nastavení systém. hodin

Používá regis.: A,X,Y

Komunikační r.: A,X,Y

Nejprve zakáže přerušení. Pak přepíše jednotlivé registry na adresy: A na $A2, X na $A1 a Y na $A0. Pak přerušení opět povolí a vrací se.Nejvyšší hod-notu má A,nižší X a nejnižší Y. Jednu minutu zapí-šeme takto: 1min=60s=60*60uteřinek (jiffies). Tuto hodnotu převedeme do hexa tvaru a rozdělíme ji do jednotlivých registrů.

SETTMO $FFA2 [FE21]:ukazatel pro IEEE-Bus

Používá regis.: A

Komunikační r.: A

Hodnotu z A přepíše na $0285.

STOP $FFE1 [(0328)]:test na STOP (a jiné)

Používá regis.: A,X

Komunikační r.: A

Test na stisk kláves STOP,C=,SHIFT,CONTROL,←,1,2 a

jejich kombinací (Podle stavu $91). Je-li stisknutá STOP-ka, Pak se Provede CLRCHN a Zse nastaví do 1. Není-li stisknutá, je Z=0.

Příklad:

```
JSR UDTIM                ;aktualizace syst.taktu
                          (není bezPodmínečně nutné)
JSR STOP
BNE <nebyl stlač.>
```

TALK $FFB4 [ED09]:nast. Periférie Pro vysílání

Používá regis.: A

Komunikační r.: A

Nast. Periférii s kódem, který je v A na vysílání. Přenos je Po sériové sběrnici.

TKSA $FF96 [EDC7]:sekund. adresa Pro vysílání

Používá regis.: A

Komunikační r.: A

Tato rutina vyšle sekundární adresu na sér. Bus.

Příklad:

```
;nastavení Periférie $04 na vys. se sekun.adr. $07
LDA#$04
JSR TALK
LDA#$07
JSR TKSA
```

UDTIM $FFEA [F69B]:aktualizace syst. taktu

Používá regis.: A,X

Tato rutina Posune o jedničku hodiny, od $A2,A1,A0 odečte #$4F,1A,01 a není-li C=1, nastaví hodiny na 00 00 00. Pak sPoluPracuje s CIA#1.

UNLSN $FFAE [EDFE]:Povel Pro sériovou Bus

Používá regis.: A

Vyšle Povel před novým LISTEN.

UNTLK $FFAB [EDEF]:Povel Pro sér. sběrnici

Používá regis.: A

Vyšle Povel na sběrnici Před novým TALK.

VECTOR $FF8D [FD1A]:obsluha vektorů in/out

Používá regis.: A,X,Y

Komunikační r.: X,Y

V X a Y je Lo,Hi adresa Počátku místa, kde jsou uloženy hodnoty Pro vektory $0314 a dále. Při C=0

se vektory přenesou z této oblasti do $0314...., a při C=1 je tomu přesně opačně.
A na závěr této podkapitoly se ještě podíváme na kódy chybových hlášek, které najdeme buď v A, nebo si je můžeme přečíst pomocí READST:
$00...Přerušeno klávesou STOP
$01...Příliš mnoho otevřených souborů
$02...soubor již byl otevřen
$03...soubor není otevřen
$04...soubor nebyl nalezen
$05...Periférie není po ruce
$06...není vstupní soubor
$07...není výstupní soubor
$08...soubor tohoto jména chybí
$09...nepřípustný kód periférie
$F0...konec změněn (pro RS-232)

Všechny tyto rutiny můžeme libovolně používat, ale je třeba vědět, co udělají, aby nedošlo k zakousnutí,nebo zhroucení,systému.

Poznámky

Příklady programů v Assembleru

V této podkapitole si ukážeme několik krátkých programů,které jsou psány v Assembleru.Do počítače je můžeme vložit buď v monitoru, nebo po převedení do decimálního tvaru příkazem POKE (převod zajistí např.program v Basicu).

1)Dvoubytové přičítání a odečítání s přenosem

V registru X máme Lo Byte, v Y Hi Byte. K těmto hodnotám přičteme postupně #$10 a #$05 (podprogram začíná na $1000),nebo od nich tyto hodnoty odečteme (podprogram začíná na $100A). Výsledky obdržíme opět v registrech X a Y. Je-li přenos do vyššího bytu,je při sčítání příznak C (bit 0 SR) nastaven,při odečítání vynulován. Oba programy spouštíme (po naplnění registrů) od jejich počátečních adresy.

2)Hrubý test klíčového slova

Tento program provede test klíčového slova,které zadáváme z klávesnice.V tomto případě se bude testovat slovo OKAY.

3)CHR$ => POKE

Pomocí tohoto programu můžeme převádět některé CHR$ kódy na POKE kódy. Místo zakázaných ukládá kód mezery <SPACE>. Před zavoláním naplníme A CHR$ kódem.Po návratu máme v A POKE kód.

4)Tisk v grafickém módu

Abychom mohli na tiskárně tisknout v grafice,je nutné nejprve přepnout tiskárnu do grafického módu (řídící znak #$08) a pak na ní vysílat kódy, které odpovídají jednotlivým jehličkám.Kódy jsou osmibitové.Z těchto osmi bitů je pouze sedm určeno pro jehličky, osmý je totiž tzv. grafický bit, který musíme nastavovat do jedničky. Tento program tedy vytiskne grafický znak,který je uložen na adresách $6140-$6146 (7Bytů odshora). Po jeho vytisknutí odřádkuje. Pro běh programu jsou ještě potřebné adresy $613E,F,ale lze ho i snadno převést do libovolné části RAM.

5)Kurzor

Posledním Prográmem, jehož výPis najdete na následujících stránkách je tento.Po jeho PřePsání ho sPustíme Příkazem SYS od adresy $C44A. Abychom viděli nějaký efekt, je ještě Potřeba zaPnout SPrite 0 (naPř.LDA$D015:ORA#$01:STA$D015). Pak se vám, Pod normálním kurzorem, objeví malý medvídek,který slouží jako kurzor nový.Jeho data jsou umístěna od $02C0 a je zobrazen ve vícebarevném módu. Ještě snad dodám,že se jmenuje Boban.

Prográm Pracuje v Prerušení,Proto s ním Po sPuštění neProvádějte žádné Pokusy (mohlo by to skončit šPatně).

```
.,  1000 8a          txa ; Pricitani
.,  1001 18          clc
.,  1002 69 10       adc #$10
.,  1004 aa          tax
.,  1005 98          tya
.,  1006 69 05       adc #$05
.,  1008 a8          tay
.,  1009 60          rts
.,  100a 8a          txa ; odecitani
.,  100b 38          sec
.,  100c e9 10       sbc #$10
.,  100e aa          tax
.,  100f 98          tya
.,  1010 e9 05       sbc #$05
.,  1012 a8          tay
.,  1013 60          rts
```

Priklady funkce

```
   pc  sr ac xr yr sp
.;2006 37 00 00 00 fc

JSR $1000

   pc  sr ac xr yr sp
.;200a 34 05 10 05 fc

   pc  sr ac xr yr sp
.;2015 b4 88 fa ff fc

JSR $1000

   pc  sr ac xr yr sp
.;200a 35 05 0a 05 fc


   pc  sr ac xr yr sp
.;2006 37 00 00 00 fc

JSR $100A

   pc  sr ac xr yr sp
.;200e b4 fa f0 fa fc


   pc  sr ac xr yr sp
.;2015 b4 88 fa ff fc

JSR $100A

   pc  sr ac xr yr sp
.;200e b5 fa ea fa fc
```

Hruby test Klicoveho slova

```
.,  1000 a0 04      ldy #$04; Pocet znaku
.,  1002 a2 00      ldx #$00
.,  1004 20 cf ff   jsr $ffcf; rutina CHRIN
.,  1007 8e 0c 10   stx $100c
.,  100a 18         clc
.,  100b 69 e9      adc #$e9
.,  100d aa         tax
.,  100e 88         dey
.,  100f d0 f3      bne $1004
.,  1011 c9 34      cmp #$34? ; Porovnani Pseudo-
.,  1013 00         brk ; Kodu slova (OKAY)
```

Pri pouziti pri LOAD se mozna bude hodit test na

Load Error

```
JSR $FFB7           ; READST
AND #$BF
BEQ <Load O.K.>
BNE <Load Error>
```

Pozn.

```
.,  c4b0 c9 ff     cmp #$ff
.,  c4b2 d0 02     bne $c4b6
.,  c4b4 a9 7e     lda #$7e
.,  c4b6 c9 c0     cmp #$c0
.,  c4b8 90 07     bcc $c4c1
.,  c4ba c9 e0     cmp #$e0
.,  c4bc b0 03     bcs $c4c1
.,  c4be 38        sec
.,  c4bf e9 60     sbc #$60
.,  c4c1 c9 e0     cmp #$e0
.,  c4c3 90 03     bcc $c4c8
.,  c4c5 38        sec
.,  c4c6 e9 40     sbc #$40
.,  c4c8 c9 20     cmp #$20
.,  c4ca 90 1c     bcc $c4e8
.,  c4cc c9 40     cmp #$40
.,  c4ce 90 1a     bcc $c4ea
.,  c4d0 c9 60     cmp #$60
.,  c4d2 b0 04     bcs $c4d8
.,  c4d4 38        sec
.,  c4d5 e9 40     sbc #$40
.,  c4d7 60        rts
.,  c4d8 c9 80     cmp #$80
.,  c4da b0 04     bcs $c4e0
.,  c4dc 38        sec
.,  c4dd e9 20     sbc #$20
.,  c4df 60        rts
.,  c4e0 c9 a0     cmp #$a0
.,  c4e2 90 04     bcc $c4e8
.,  c4e4 38        sec
.,  c4e5 e9 40     sbc #$40
.,  c4e7 60        rts
.,  c4e8 a9 20     lda #$20
.,  c4ea 60        rts
```

Pozn.

```
., 6000 a9 01     lda #$01
., 6002 a2 04     ldx #$04
., 6004 a0 00     ldy #$00
., 6006 20 ba ff  jsr $ffba; SETLFS
., 6009 a9 00     lda #$00
., 600b 20 bd ff  jsr $ffbd; SETNAM
., 600e 20 c0 ff  jsr $ffc0; OPEN
., 6011 a2 01     ldx #$01
., 6013 20 c9 ff  jsr $ffc9; CHKOUT
., 6016 a9 08     lda #$08
., 6018 20 d2 ff  jsr $ffd2; CHROUT
., 601b a9 00     lda #$00
., 601d 8d 3e 61  sta $613e
., 6020 a2 00     ldx #$00
., 6022 bd 40 61  lda $6140,x
., 6025 18        clc
., 6026 0a        asl
., 6027 9d 40 61  sta $6140,x
., 602a 6e 3f 61  ror $613f
., 602d e8        inx
., 602e e0 07     cpx #$07
., 6030 d0 f0     bne $6022
., 6032 38        sec
., 6033 6e 3f 61  ror $613f; + graf. bit
., 6036 a2 01     ldx #$01
., 6038 20 c9 ff  jsr $ffc9
., 603b ad 3f 61  lda $613f
., 603e 20 d2 ff  jsr $ffd2
., 6041 ee 3e 61  inc $613e
., 6044 a9 08     lda #$08; test na delku
., 6046 cd 3e 61  cmp $613e
., 6049 d0 d5     bne $6020
., 604b a9 0d     lda #$0d
., 604d 20 d2 ff  jsr $ffd2; odrol. radky
., 6050 20 cc ff  jsr $ffcc; CLRCHN
., 6053 a2 01     ldx #$01
., 6055 20 c3 ff  jsr $ffc3; CLOSE
., 6058 60        rts
```

Pozn.

```
.,  c3e8  a9 0b       lda #$0b
.,  c3ea  8d f8 07    sta $07f8; definice Sprite 0
.,  c3ed  a9 01       lda #$01
.,  c3ef  8d 1c d0    sta $d01c; Sprite 0 -barevny
.,  c3f2  8d 25 d0    sta $d025; barva 0
.,  c3f5  a9 0a       lda #$0a
.,  c3f7  8d 26 d0    sta $d026; barva 1
.,  c3fa  a5 d3       lda $d3; Poloha kurzoru
.,  c3fc  c9 28       cmp #$28; Prestupny radek?
.,  c3fe  90 03       bcc $c403
.,  c400  38          sec
.,  c401  e9 28       sbc #$28
.,  c403  c9 1e       cmp #$1e
.,  c405  90 12       bcc $c419
.,  c407  aa          tax
.,  c408  ad 10 d0    lda $d010
.,  c40b  09 01       ora #$01
.,  c40d  8d 10 d0    sta $d010; Hi bit sour. X
.,  c410  8a          txa
.,  c411  38          sec
.,  c412  e9 1e       sbc #$1e
.,  c414  a2 00       ldx #$00
.,  c416  4c 25 c4    jmp $c425
.,  c419  a8          tay
.,  c41a  ad 10 d0    lda $d010
.,  c41d  29 fe       and #$fe
.,  c41f  8d 10 d0    sta $d010; nul. Hi bitu X
.,  c422  98          tya
.,  c423  a2 10       ldx #$10
.,  c425  a8          tay
.,  c426  8a          txa
.,  c427  c0 00       cpy #$00
.,  c429  f0 07       beq $c432
.,  c42b  18          clc
.,  c42c  69 08       adc #$08
.,  c42e  88          dey
.,  c42f  4c 27 c4    jmp $c427
.,  c432  8d 00 d0    sta $d000; Poloha v X
.,  c435  a9 2d       lda #$2d
.,  c437  a4 d6       ldy $d6; Poloha kurzoru
.,  c439  c0 00       cpy #$00
.,  c43b  f0 07       beq $c444
.,  c43d  18          clc
.,  c43e  69 08       adc #$08
.,  c440  88          dey
.,  c441  4c 39 c4    jmp $c439
.,  c444  8d 01 d0    sta $d001; Poloha v Y
.,  c447  4c 31 ea    jmp $ea31; Pokracovani IRQ

.,  c44a  a2 e8       ldx #$e8; start
.,  c44c  a0 c3       ldy #$c3
.,  c44e  78          sei
.,  c44f  8e 14 03    stx $0314
.,  c452  8c 15 03    sty $0315
.,  c455  58          cli
```

```
., c456 60                rts

Data Sprite 0 (medvidek)
.:02c0 00 c3 00 00 ff 00 03 3c
.:02c8 c0 03 3c c0 00 ff 00 00
.:02d0 3c 00 00 d7 00 03 3c c0
.:02d8 01 28 40 00 3c 00 00 d7
.:02e0 00 00 c3 00 01 43 00 00
.:02e8 01 40 00 00 00 00 00 00
.:02f0 00 00 00 00 00 00 00 00
.:02f8 00 00 00 00 00 00 00 00
```

Pozn.

KAPITOLA V

Když mi bylo čtrnáct,
byl můj táta takový ignorant,
že jsem ho sotva vydržel.
Ale když mi pak bylo jednadvacet,
byl jsem překvapen,
kolik se stařík za těch sedm let
naučil.
(Mark Twain, 1835 - 1910)

Přežívání lidské kultury,
stejně jako přežívání organismů
a společností,
je závislé na vnitřní interakci,
jíž lze dosíci jen tím,
že věda podrží svou smyslupInost
pro lidskou životní praxi.
(René Dubos, 1964)

Intelektuál je někdo,
jehož mysl se má na pozoru.
(Albert Camus, 1957)

1)Výpis Basic ROM

2)Výpis Kernal ROM

```
a000 94 e3 7b e3 43 42 4d 42; vektory pro RESET
($e394) a RESTORE ($e37b);    kody 'CBMB'
a008 41 53 49 43 30 a8 41 a7; 'ASIC'; tabulka ne-
Primych skoku (ulozi se do zasobniku a vyvolaji se
RTS) na Prikazy                END;FOR
a010 1d ad f7 a8 a4 ab be ab;  NEXT;DATA;INPUT#;
                             ; INPUT
a018 80 b0 05 ac a4 a9 9f a8;  DIM;READ;LET;GOTO
a020 70 a8 27 a9 1c a8 82 a8;  RUN;IF;RESTORE;GOSUB
a028 d1 a8 3a a9 2e a8 4a a9;  RETURN;REM;STOP;ON
a030 2c b8 67 e1 55 e1 64 e1;  WAIT;LOAD;SAVE;
                             ; VERIFY
a038 b2 b3 23 b8 7f aa 9f aa;  DEF;POKE;PRINT#;
                             ; PRINT
a040 56 a8 9b a6 5d a6 85 aa;  CONT;LIST;CLR;CMD
a048 29 e1 bd e1 c6 e1 7a ab;  SYS;OPEN;CLOSE;GET
                             ; (i GET#)
a050 41 a6 39 bc cc bc 58 bc;  NEW; tabulka Primych
skoku na rutiny pro Prikazy    SGN;INT;ABS
a058 10 03 7d b3 9e b3 71 bf;  USR;FRE;POS;SQR
a060 97 e0 ea b9 ed bf 64 e2;  RND;LOG;EXP;COS
a068 6b e2 b4 e2 0e e3 0d b8;  SIN;TAN;ATN;PEEK
a070 7c b7 65 b4 ad b7 8b b7;  LEN;STR$;VAL;ASC
a078 ec b6 00 b7 2c b7 37 b7;  CHR$;LEFT$;RIGHT$;
                             ; MID$
Tabulka skoku s dodatkovou hodnotou (h Lo Hi)
a080 79 69 b8 79 52 b8 7b 2a;  +;-;*
a088 ba 7b 11 bb 7f 7a bf 50;  *;/;↑;AND
a090 e8 af 46 e5 af 7d b3 bf;  AND;OR;>
a098 5a d3 ae 64 15 b0 45 4e;  =;<;
Tabulka Prikazu                en
a0a0 c4 46 4f d2 4e 45 58 d4;  D;foR;nexT
a0a8 44 41 54 c1 49 4e 50 55;  datA;inpu
a0b0 54 a3 49 4e 50 55 d4 44;  t#;inPuT;d
a0b8 49 cd 52 45 41 c4 4c 45;  iM;reaD;le
a0c0 d4 47 4f 54 cf 52 55 ce;  T;gotO;ruN
a0c8 49 c6 52 45 53 54 4f 52;  iF;restor
a0d0 c5 47 4f 53 55 c2 52 45;  E;gosuB;re
a0d8 54 55 52 ce 52 45 cd 53;  turN;reM;s
a0e0 54 4f d0 4f ce 57 41 49;  toP;oN;wai
a0e8 d4 4c 4f 41 c4 53 41 56;  T;loaD;sav
a0f0 c5 56 45 52 49 46 d9 44;  E;verifY;d
a0f8 45 c6 50 4f 4b c5 50 52;  eF;PoKE;Pr
a100 49 4e 54 a3 50 52 49 4e;  int#;Prin
a108 d4 43 4f 4e d4 4c 49 53;  T;conT;lis
a110 d4 43 4c d2 43 4d c4 53;  T;clR;cmD;s
a118 59 d3 4f 50 45 ce 43 4c;  yS;oPeN;cl
a120 4f 53 c5 47 45 d4 4e 45;  osE;geT;ne
a128 d7 54 41 42 a8 54 cf 46;  W;tab(;tO;f
a130 ce 53 50 43 a8 54 48 45;  N;sPc(;the
a138 ce 4e 4f d4 53 54 45 d0;  N;noT;steP
a140 ab ad aa af de 41 4e c4;  +;-;*;/;↑;anD
(CHR$ Kod + [-;*;/;↑] OR$80...zapnuti 7. bitu)
a148 4f d2 be bd bc 53 47 ce;  oR;>;=;<;sgN
a150 49 4e d4 41 42 d3 55 53;  inT;abS;us
```

```
a.158 d2 46 52 c5 50 4f d3 53;  R;frE;poS;s
a.160 51 d2 52 4e c4 4c 4f c7;  qR;rnD;loG
a.168 45 58 d0 43 4f d3 53 49;  exP;coS;si
a.170 ce 54 41 ce 41 54 ce 50;  N;taN;atN;p
a.178 45 45 cb 4c 45 ce 53 54;  eeK;leN;st
a.180 52 a4 56 41 cc 41 53 c3;  r$;vaL;asC
a.188 43 48 52 a4 4c 45 46 54;  chr$;left
a.190 a4 52 49 47 48 54 a4 4d;  $;right$;m
a.198 49 44 a4 47 cf 00 54 4f;  id$;gO;$00-Konec
Tabulka chybovych hlasek        to
a.1a0 4f 20 4d 41 4e 59 20 46;  o many f
a.1a8 49 4c 45 d3 46 49 4c 45;  ileS;file
a.1b0 20 4f 50 45 ce 46 49 4c;  opeN;fil
a.1b8 45 20 4e 4f 54 20 4f 50;  e not oP
a.1c0 45 ce 46 49 4c 45 20 4e;  eN;file n
a.1c8 4f 54 20 46 4f 55 4e c4;  ot founD
a.1d0 44 45 56 49 43 45 20 4e;  device n
a.1d8 4f 54 20 50 52 45 53 45;  ot prese
a.1e0 4e d4 4e 4f 54 20 49 4e;  nT;not in
a.1e8 50 55 54 20 46 49 4c c5;  put filE
a.1f0 4e 4f 54 20 4f 55 54 50;  not outP
a.1f8 55 54 20 46 49 4c c5 4d;  ut filE;m
a.200 49 53 53 49 4e 47 20 46;  issing f
a.208 49 4c 45 20 4e 41 4d c5;  ile namE
a.210 49 4c 4c 45 47 41 4c 20;  illegal
a.218 44 45 56 49 43 45 20 4e;  device n
a.220 55 4d 42 45 d2 4e 45 58;  umbeR;nex
a.228 54 20 57 49 54 48 4f 55;  t withou
a.230 54 20 46 4f d2 53 59 4e;  t foR;syn
a.238 54 41 d8 52 45 54 55 52;  taX;retur
a.240 4e 20 57 49 54 48 4f 55;  n withou
a.248 54 20 47 4f 53 55 c2 4f;  t gosuB;o
a.250 55 54 20 4f 46 20 44 41;  ut of da
a.258 54 c1 49 4c 4c 45 47 41;  tA;illega
a.260 4c 20 51 55 41 4e 54 49;  l quanti
a.268 54 d9 4f 56 45 52 46 4c;  tY;overfl
a.270 4f d7 4f 55 54 20 4f 46;  oW;out of
a.278 20 4d 45 4d 4f 52 d9 55;  memorY;u
a.280 4e 44 45 46 27 44 20 53;  ndef'd s
a.288 54 41 54 45 4d 45 4e d4;  tatemenT
a.290 42 41 44 20 53 55 42 53;  bad subs
a.298 43 52 49 50 d4 52 45 44;  cripT;red
a.2a0 49 4d 27 44 20 41 52 52;  im'd arr
a.2a8 41 d9 44 49 56 49 53 49;  aY;divisi
a.2b0 4f 4e 20 42 59 20 5a 45;  on bY ze
a.2b8 52 cf 49 4c 4c 45 47 41;  rO;illega
a.2c0 4c 20 44 49 52 45 43 d4;  l direct
a.2c8 54 59 50 45 20 4d 49 53;  taPe mis
a.2d0 4d 41 54 43 c8 53 54 52;  matcH;str
a.2d8 49 4e 47 20 54 4f 4f 20;  ing too
a.2e0 4c 4f 4e c7 46 49 4c 45;  lonG;file
a.2e8 20 44 41 54 c1 46 4f 52;  datA;for
a.2f0 4d 55 4c 41 20 54 4f 4f;  mula too
a.2f8 20 43 4f 4d 50 4c 45 d8;  comPleX
a.300 43 41 4e 27 54 20 43 4f;  can't co
a.308 4e 54 49 4e 55 c5 55 4e;  ntinuE;un
a.310 44 45 46 27 44 20 46 55;  def'd fu
a.318 4e 43 54 49 4f ce 56 45;  nctioN;ve
```

```
a320 52 49 46 d9 4c 4f 41 c4; rifY;loaD
Ukazatele na Pocatek Predchozich chybovych hla-
sek na hlasku BREAK
a328 9e a1 ac a1 b5 a1 c2 a1
a330 d0 a1 e2 a1 f0 a1 ff a1
a338 10 a2 25 a2 35 a2 3b a2
a340 4f a2 5a a2 6a a2 72 a2
a348 7f a2 90 a2 9d a2 aa a2
a350 ba a2 c8 a2 d5 a2 e4 a2
a358 ed a2 00 a3 0e a3 1e a3
a360 24 a3 83 a3 0d 4f 4b 0d; od $a364 jsou
doPlnkove hlasky                OK
a368 00 20 20 45 52 52 4f 52; error
a370 00 20 49 4e 20 00 0d 0a; in
a378 52 45 41 44 59 2e 0d 0a; ready.
a380 00 0d 0a 42 52 45 41 4b; break
a388 00;                        Konec hlasek

Pocatek bloku Programu
```

```
a389  a0 ba       ldy #$ba
a38b  e8          inx
a38c  e8          inx
a38d  e8          inx
a38e  e8          inx
a38f  bd 01 01    lda $0101,x
a392  c9 81       cmp #$81; Kod 'FOR'
a394  d0 21       bne $a3b7
a396  a5 4a       lda $4a
a398  d0 0a       bne $a3a4
a39a  bd 02 01    lda $0102,x
a39d  85 49       sta $49; Lo
a39f  bd 03 01    lda $0103,x
a3a2  85 4a       sta $4a; Hi Promenna FOR
a3a4  dd 03 01    cmp $0103,x
a3a7  d0 07       bne $a3b0
a3a9  a5 49       lda $49
a3ab  dd 02 01    cmp $0102,x
a3ae  f0 07       beq $a3b7
a3b0  8a          txa
a3b1  18          clc
a3b2  69 12       adc #$12
a3b4  aa          tax
a3b5  d0 d8       bne $a38f
a3b7  60          rts
```

```
a3b8  20 08 a4    jsr $a408
a3bb  85 31       sta $31; Lo
a3bd  84 32       sty $32; Hi Konec Poli
a3bf  38          sec
a3c0  a5 5a       lda $5a
a3c2  e5 5f       sbc $5f
a3c4  85 22       sta $22
a3c6  a8          tay
a3c7  a5 5b       lda $5b
a3c9  e5 60       sbc $60
a3cb  aa          tax
a3cc  e8          inx
```

```
a.3cd  98        tya
a.3ce  f0 23     beq  $a.3f3
a.3d0  a5 5a.    lda  $5a.
a.3d2  38        sec
a.3d3  e5 22     sbc  $22
a.3d5  85 5a.    sta  $5a.
a.3d7  b0 03     bcs  $a.3dc
a.3d9  c6 5b     dec  $5b
a.3db  38        sec
a.3dc  a.5 58    lda  $58
a.3de  e5 22     sbc  $22
a.3e0  85 58     sta  $58
a.3e2  b0 08     bcs  $a.3ec
a.3e4  c6 59     dec  $59
a.3e6  90 04     bcc  $a.3ec
a.3e8  b1 5a.    lda  ($5a.),y
a.3ea. 91 58     sta  ($58),y
a.3ec  88        dey
a.3ed  d0 f9     bne  $a.3e8
a.3ef  b1 5a.    lda  ($5a.),y
a.3f1  91 58     sta  ($58),y
a.3f3  c6 5b     dec  $5b
a.3f5  c6 59     dec  $59
a.3f7  ca.       dex
a.3f8  d0 f2     bne  $a.3ec
a.3fa. 60        rts
-------------------------------------------------------------
a.3fb  0a.       asl
a.3fc  69 3e     adc  #$3e
a.3fe  b0 35     bcs  $a.435
a.400  85 22     sta  $22
a.402  ba.       tsx
a.403  e4 22     cpx  $22
a.405  90 2e     bcc  $a.435
a.407  60        rts
-------------------------------------------------------------
a.408  c4 34     cpy  $34; Hi
a.40a. 90 28     bcc  $a.434
a.40c  d0 04     bne  $a.412
a.40e  c5 33     cmp  $33; Lo Pocatek retezcu
a.410  90 22     bcc  $a.434
a.412  48        pha
a.413  a.2 09    ldx  #$09
a.415  98        tya
a.416  48        pha
a.417  b5 57     lda  $57,x
a.419  ca.       dex
a.41a. 10 fa.    bpl  $a.416
a.41c  20 26 b5  jsr  $b526
a.41f  a.2 f7    ldx  #$f7
a.421  68        pla
a.422  95 61     sta  $61,x
a.424  e8        inx
a.425  30 fa.    bmi  $a.421
a.427  68        pla
a.428  a.8       tay
a.429  68        pla
a.42a. c4 34     cpy  $34
```

```
a.42c  90 06        bcc $a.434
a.42e  d0 05        bne $a.435
a.430  c5 33        cmp $33
a.432  b0 01        bcs $a.435
a.434  60           rts
-------------------------------------------------------
a.435  a.2 10       ldx #$10; 'Out of memory'
a.437  6c 00 03     jmp ($0300); rutina pro chyby
a.43a  8a.          txa
a.43b  0a.          asl
a.43c  a.a.         tax
a.43d  bd 26 a.3    lda $a.326,x; Lo
a.440  85 22        sta $22
a.442  bd 27 a.3.   lda $a.327,x; Hi ukazatel hlasky
a.445  85 23        sta $23
a.447  20 cc ff     jsr $ffcc; CLRCHN
a.44a. a.9 00       lda #$00
a.44c  85 13        sta $13; argument CMD
a.44e  20 d7 a.a.   jsr $a.ad7; vyslani <RETURN>
a.451  20 45 a.b    jsr $a.b45; vyslani '?'
a.454  a.0 00       ldy #$00; tisk hlasky
a.456  b1 22        lda ($22),y
a.458  48           pha
a.459  29 7f        and #$7f
a.45b  20 47 a.b    jsr $a.b47
a.45e  c8           iny
a.45f  68           pla
a.460  10 f4        bpl $a.456
a.462  20 7a. a.6   jsr $a.67a.; nast.pocat.stavu
a.465  a.9 69       lda #$69; Lo
a.467  a.0 a.3      ldy #$a.3; Hi adresa 'ERROR'
a.469  20 1e a.b    jsr $ab1e
a.46c  a.4 3a.      ldy $3a; Hi preladany radek
a.46e  c8           iny
a.46f  f0 03        beq $a.474
a.471  20 c2 bd     jsr $bdc2; 'in'+cislo radku
a.474  a.9 76       lda #$76; Lo
a.476  a.0 a.3      ldy #$a.3; Hi adresa 'READY.'
a.478  20 1e a.b    jsr $a.b1e; tisk hlasky
a.47b  a.9 80       lda #$80
a.47d  20 90 ff     jsr $ff90; SETMSG (zap)
a.480  6c 02 03     jmp ($0302); hlavni blok prekl.
-------------------------------------------------------
a.483  20 60 a.5    jsr $a.560; naplneni buffru
a.486  86 7a.       stx $7a; Lo
a.488  84 7b        sty $7b; Hi adresa CHRGET
a.48a. 20 73 00     jsr $0073; cteni znaku-CHRGET
a.48d  a.a.         tax
a.48e  f0 f0        beq $a.480
a.490  a.2 ff       ldx #$ff; primy (naprogr.) mod
a.492  86 3a.       stx $3a; Hi cislo preklad.r.
a.494  90 06        bcc $a.49c
a.496  20 79 a.5    jsr $a.579; skok na dekod.prikazu
a.499  4c e1 a.7    jmp $a.7e1; nasledujici prikaz
-------------------------------------------------------
a.49c  20 6b a.9    jsr $a.96b; programovy mod
a.49f  20 79 a.5    jsr $a.579; dekodovani prikazu
a.4a.2 84 0b        sty $0b; ukazatel vst.buffru
```

```
a4a4  20 13 a6     jsr $a613
a4a7  90 44        bcc $a4ed
a4a9  a0 01        ldy #$01
a4ab  b1 5f        lda ($5f),y
a4ad  85 23        sta $23
a4af  a5 2d        lda $2d; Lo Konec Basicu
a4b1  85 22        sta $22
a4b3  a5 60        lda $60
a4b5  85 25        sta $25
a4b7  a5 5f        lda $5f
a4b9  88           dey
a4ba  f1 5f        sbc ($5f),y
a4bc  18           clc
a4bd  65 2d        adc $2d
a4bf  85 2d        sta $2d
a4c1  85 24        sta $24
a4c3  a5 2e        lda $2e; Hi Konec Basicu
a4c5  69 ff        adc #$ff
a4c7  85 2e        sta $2e
a4c9  e5 60        sbc $60
a4cb  aa           tax
a4cc  38           sec
a4cd  a5 5f        lda $5f
a4cf  e5 2d        sbc $2d
a4d1  a8           tay
a4d2  b0 03        bcs $a4d7
a4d4  e8           inx
a4d5  c6 25        dec $25
a4d7  18           clc
a4d8  65 22        adc $22
a4da  90 03        bcc $a4df
a4dc  c6 23        dec $23
a4de  18           clc
a4df  b1 22        lda ($22),y
a4e1  91 24        sta ($24),y
a4e3  c8           iny
a4e4  d0 f9        bne $a4df
a4e6  e6 23        inc $23
a4e8  e6 25        inc $25
a4ea  ca           dex
a4eb  d0 f2        bne $a4df
a4ed  20 59 a6     jsr $a659; CLR
a4f0  20 33 a5     jsr $a533
a4f3  ad 00 02     lda $0200; Pocatek vst. buffru
a4f6  f0 88        beq $a480
a4f8  18           clc
a4f9  a5 2d        lda $2d
a4fb  85 5a        sta $5a
a4fd  65 0b        adc $0b
a4ff  85 58        sta $58
a501  a4 2e        ldy $2e
a503  84 5b        sty $5b
a505  90 01        bcc $a508
a507  c8           iny
a508  84 59        sty $59
a50a  20 b8 a3     jsr $a3b8
a50d  a5 14        lda $14; Lo
a50f  a4 15        ldy $15; Hi r. GOTO apod.
```

```
a511  8d fe 01     sta $01fe
a514  8c ff 01     sty $01ff; Pocatek zasobniku
a517  a5 31        lda $31; Lo
a519  a4 32        ldy $32; Hi Konec tabulek
a51b  85 2d        sta $2d; Lo
a51d  84 2e        sty $2e; Hi Pocatek promennych
a51f  a4 0b        ldy $0b; ukaz. vstup.buffru
a521  88           dey
a522  b9 fc 01     lda $01fc,y
a525  91 5f        sta ($5f),y
a527  88           dey
a528  10 f8        bpl $a522
a52a  20 59 a6     jsr $a659
a52d  20 33 a5     jsr $a533
a530  4c 80 a4     jmp $a480; hl.blok prekladace
```

```
a533  a5 2b        lda $2b; Lo
a535  a4 2c        ldy $2c; Hi Pocatek Basicu
a537  85 22        sta $22
a539  84 23        sty $23
a53b  18           clc
a53c  a0 01        ldy #$01
a53e  b1 22        lda ($22),y
a540  f0 1d        beq $a55f
a542  a0 04        ldy #$04
a544  c8           iny
a545  b1 22        lda ($22),y
a547  d0 fb        bne $a544
a549  c8           iny
a54a  98           tya
a54b  65 22        adc $22
a54d  aa           tax
a54e  a0 00        ldy #$00
a550  91 22        sta ($22),y
a552  a5 23        lda $23
a554  69 00        adc #$00
a556  c8           iny
a557  91 22        sta ($22),y
a559  86 22        stx $22
a55b  85 23        sta $23
a55d  90 dd        bcc $a53c
a55f  60           rts
```

Naplneni vst. buffru Basicu

```
a560  a2 00        ldx #$00
a562  20 12 e1     jsr $e112
a565  c9 0d        cmp #$0d; <RETURN>
a567  f0 0d        beq $a576
a569  9d 00 02     sta $0200,x; zapis do buffru
a56c  e8           inx
a56d  e0 59        cpx #$59; max. 89 znaku
a56f  90 f1        bcc $a562
a571  a2 17        ldx #$17; 'String too long'
a573  4c 37 a4     jmp $a437; chybova hlaska
a576  4c ca aa     jmp $aaca; zakonceni buffru
```

```
a579  6c 04 03     jmp ($0304); zpracovani prikazu
a57c  a6 7a        ldx $7a
```

```
a.57e  a0 04      ldy #$04
a.580  84 0f      sty $0f
a.582  bd 00 02   lda $0200,x; cteni z buffru
a.585  10 07      bpl $a58e
a.587  c9 ff      cmp #$ff
a.589  f0 3e      beq $a5c9
a.58b  e8         inx
a.58c  d0 f4      bne $a582
a.58e  c9 20      cmp #$20; <SPACE>?
a.590  f0 37      beq $a5c9
a.592  85 08      sta $08
a.594  c9 22      cmp #$22; uvozovky?
a.596  f0 56      beq $a5ee
a.598  24 0f      bit $0f
a.59a  70 2d      bvs $a5c9
a.59c  c9 3f      cmp #$3f; Kod '?'
a.59e  d0 04      bne $a5a4
a.5a0  a9 99      lda #$99; Kod prikazu PRINT
a.5a2  d0 25      bne $a5c9
a.5a4  c9 30      cmp #$30; Kod '0'
a.5a6  90 04      bcc $a5ac
a.5a8  c9 3c      cmp #$3c; Kod '<'
a.5aa  90 1d      bcc $a5c9
a.5ac  84 71      sty $71
a.5ae  a0 00      ldy #$00
a.5b0  84 0b      sty $0b; ukazatel buffru
a.5b2  88         dey
a.5b3  86 7a      stx $7a; Lo CHRGET
a.5b5  ca         dex
a.5b6  c8         iny
a.5b7  e8         inx
a.5b8  bd 00 02   lda $0200,x
a.5bb  38         sec
a.5bc  f9 9e a0   sbc $a09e,y; Poc. tab. prikazu
a.5bf  f0 f5      beq $a5b6
a.5c1  c9 80      cmp #$80; test na konc. znak
a.5c3  d0 30      bne $a5f5
a.5c5  05 0b      ora $0b
a.5c7  a4 71      ldy $71
a.5c9  e8         inx
a.5ca  c8         iny
a.5cb  99 fb 01   sta $01fb,y
a.5ce  b9 fb 01   lda $01fb,y
a.5d1  f0 36      beq $a609
a.5d3  38         sec
a.5d4  e9 3a      sbc #$3a
a.5d6  f0 04      beq $a5dc
a.5d8  c9 49      cmp #$49
a.5da  d0 02      bne $a5de
a.5dc  85 0f      sta $0f
a.5de  38         sec
a.5df  e9 55      sbc #$55
a.5e1  d0 9f      bne $a582
a.5e3  85 08      sta $08
a.5e5  bd 00 02   lda $0200,x
a.5e8  f0 df      beq $a5c9
a.5ea  c5 08      cmp $08
a.5ec  f0 db      beq $a5c9
```

```
a5e9  c8                  iny
a5ef  99 fb 01            sta $01fb,y
a5f2  e8                  inx
a5f3  d0 f0               bne $a5e5
a5f5  a6 7a               ldx $7a; Lo CHRGET
a5f7  e6 0b               inc $0b; ukazatel zasobniku
a5f9  c8                  iny
a5fa  b9 9d a0            lda $a09d,y; tab. prikazu
a5fd  10 fa               bpl $a5f9
a5ff  b9 9e a0            lda $a09e,y
a602  d0 b4               bne $a5b8
a604  bd 00 02            lda $0200,x
a607  10 be               bpl $a5c7
a609  99 fd 01            sta $01fd,y
a60c  c6 7b               dec $7b; Hi CHRGET
a60e  a9 ff               lda #$ff
a610  85 7a               sta $7a; Lo CHRGET
a612  60                  rts
```

```
a613  a5 2b               lda $2b; Lo
a615  a6 2c               ldx $2c; Hi PocateK Basicu
a617  a0 01               ldy #$01
a619  85 5f               sta $5f
a61b  86 60               stx $60
a61d  b1 5f               lda ($5f),y
a61f  f0 1f               beq $a640
a621  c8                  iny
a622  c8                  iny
a623  a5 15               lda $15; Hi radeK pri GOTO...
a625  d1 5f               cmp ($5f),y
a627  90 18               bcc $a641
a629  f0 03               beq $a62e
a62b  88                  dey
a62c  d0 09               bne $a637
a62e  a5 14               lda $14; Lo r. pri GOTO apod.
a630  88                  dey
a631  d1 5f               cmp ($5f),y
a633  90 0c               bcc $a641
a635  f0 0a               beq $a641
a637  88                  dey
a638  b1 5f               lda ($5f),y
a63a  aa                  tax
a63b  88                  dey
a63c  b1 5f               lda ($5f),y
a63e  b0 d7               bcs $a617
a640  18                  clc
a641  60                  rts
```

```
NEW
a642  d0 fd               bne $a641
a644  a9 00               lda #$00
a646  a8                  tay
a647  91 2b               sta ($2b),y; Lo
a649  c8                  iny
a64a  91 2b               sta ($2b),y; Hi PocatekK Basicu
a64c  a5 2b               lda $2b; Lo Poc. Basicu
a64e  18                  clc
a64f  69 02               adc #$02
```

```
a651  85 2d        sta $2d; Lo Konec Basicu
a653  a5 2c        lda $2c; Hi PocateK Basicu
a655  69 00        adc #$00
a657  85 2e        sta $2e; Hi Konec Basicu
CLR
a659  20 8e a6     jsr $a68e; Posunuti CHRGET
a65c  a9 00        lda #$00
a65e  d0 2d        bne $a68d
a660  20 e7 ff     jsr $ffe7; CLALL
a663  a5 37        lda $37; Lo
a665  a4 38        ldy $38; Hi Konec Basic RAM
a667  85 33        sta $33; Lo
a669  84 34        sty $34; Hi PocateK retezcu
a66b  a5 2d        lda $2d; Lo
a66d  a4 2e        ldy $2e; Hi Konec Basicu
a66f  85 2f        sta $2f; Lo
a671  84 30        sty $30; Hi Pocatek Poli
a673  85 31        sta $31; Lo
a675  84 32        sty $32; Hi Konec Poli
a677  20 1d a8     jsr $a81d; rutina <RESTORE>
a67a  a2 19        ldx #$19
a67c  86 16        stx $16
a67e  68           pla
a67f  a8           tay
a680  68           pla
a681  a2 fa        ldx #$fa
a683  9a           txs
a684  48           pha
a685  98           tya
a686  48           pha
a687  a9 00        lda #$00
a689  85 3e        sta $3e; Hi adresa pro CONT
a68b  85 10        sta $10; ident. funkce;Pole
a68d  60           rts
-------------------------------------------------
Nastaveni CHRGET o Pozici bliz nez je Poc.Basicu
a68e  18           clc
a68f  a5 2b        lda $2b
a691  69 ff        adc #$ff
a693  85 7a        sta $7a
a695  a5 2c        lda $2c
a697  69 ff        adc #$ff
a699  85 7b        sta $7b
a69b  60           rts
-------------------------------------------------
LIST
a69c  90 06        bcc $a6a4
a69e  f0 04        beq $a6a4
a6a0  c9 ab        cmp #$ab; '-' ?
a6a2  d0 e9        bne $a68d
a6a4  20 6b a9     jsr $a96b; cteni cisla a dec=>hex
a6a7  20 13 a6     jsr $a613
a6aa  20 79 00     jsr $0079; CHRGET bez posunuti
a6ad  f0 0c        beq $a6bb
a6af  c9 ab        cmp #$ab;
a6b1  d0 8e        bne $a641
a6b3  20 73 00     jsr $0073; CHRGET
a6b6  20 6b a9     jsr $a96b
```

```
a.6b9  d0 86      bne $a.641
a.6bb  68         pla
a.6bc  68         pla
a.6bd  a.5 14     lda $14
a.6bf  05 15      ora $15
a.6c1  d0 06      bne $a.6c9
a.6c3  a.9 ff     lda #$ff
a.6c5  85 14      sta $14; Lo radka pri GOTO
a.6c7  85 15      sta $15; Hi LIST a.Pod.
a.6c9  a.0 01     ldy #$01
a.6cb  84 0f      sty $0f
a.6cd  b1 5f      lda ($5f),y
a.6cf  f0 43      beq $a.714
a.6d1  20 2c a.8  jsr $a.82c; test na BREAK
a.6d4  20 d7 a.a. jsr $a.ad7; vyslani <RETURN>
a.6d7  c8         iny
a.6d8  b1 5f      lda ($5f),y
a.6da  a.a.       tax
a.6db  c8         iny
a.6dc  b1 5f      lda ($5f),y
a.6de  c5 15      cmp $15
a.6e0  d0 04      bne $a.6e6
a.6e2  e4 14      cpx $14
a.6e4  f0 02      beq $a.6e8
a.6e6  b0 2c      bcs $a.714
a.6e8  84 49      sty $49
a.6ea. 20 cd bd   jsr $bdcd
a.6ed  a.9 20     lda #$20
a.6ef  a.4 49     ldy $49
a.6f1  29 7f      and #$7f
a.6f3  20 47 a.b  jsr $a.b47; tisk
a.6f6  c9 22      cmp #$22; uvozovky?
a.6f8  d0 06      bne $a.700
a.6fa. a.5 0f     lda $0f
a.6fc  49 ff      eor #$ff
a.6fe  85 0f      sta $0f
a.700  c8         iny
a.701  f0 11      beq $a.714
a.703  b1 5f      lda ($5f),y
a.705  d0 10      bne $a.717
a.707  a.8        tay
a.708  b1 5f      lda ($5f),y
a.70a. a.a.       tax
a.70b  c8         iny
a.70c  b1 5f      lda ($5f),y
a.70e  86 5f      stx $5f
a.710  85 60      sta $60
a.712  d0 b5      bne $a.6c9
a.714  4c 86 e3   jmp $e386
a.717  6c 06 03   jmp ($0306); tisk prikazu
-------------------------------------------------
a.71a. 10 d7      bpl $a.6f3
a.71c  c9 ff      cmp #$ff
a.71e  f0 d3      beq $a.6f3
a.720  24 0f      bit $0f
a.722  30 cf      bmi $a.6f3
a.724  38         sec
a.725  e9 7f      sbc #$7f
```

```
a.727  aa              tax
a.728  84 49           sty $49
a.72a  a0 ff           ldy #$ff
a.72c  ca              dex
a.72d  f0 08           beq $a737
a.72f  c8              iny
a.730  b9 9e a0        lda $a09e,y
a.733  10 fa           bpl $a72f
a.735  30 f5           bmi $a72c
a.737  c8              iny
a.738  b9 9e a0        lda $a09e,y; tabulka prikazu
a.73b  30 b2           bmi $a6ef
a.73d  20 47 ab        jsr $ab47; tisk
a.740  d0 f5           bne $a737
FOR
a.742  a9 80           lda #$80
a.744  85 10           sta $10
a.746  20 a5 a9        jsr $a9a5; rutina <LET>
a.749  20 8a a3        jsr $a38a; ulozeni do zasobniku
a.74c  d0 05           bne $a753
a.74e  8a              txa
a.74f  69 0f           adc #$0f
a.751  aa              tax
a.752  9a              txs
a.753  68              pla
a.754  68              pla
a.755  a9 09           lda #$09
a.757  20 fb a3        jsr $a3fb; 'Out of memory' ?
a.75a  20 06 a9        jsr $a906
a.75d  18              clc
a.75e  98              tya
a.75f  65 7a           adc $7a; Lo
a.761  48              pha
a.762  a5 7b           lda $7b; Hi CHRGET
a.764  69 00           adc #$00
a.766  48              pha
a.767  a5 3a           lda $3a; Hi
a.769  48              pha
a.76a  a5 39           lda $39; Lo cislo r. programu
a.76c  48              pha
a.76d  a9 a4           lda #$a4
a.76f  20 ff ae        jsr $aeff
a.772  20 8d ad        jsr $ad8d
a.775  20 8a ad        jsr $ad8a
a.778  a5 66           lda $66
a.77a  09 7f           ora #$7f
a.77c  25 62           and $62
a.77e  85 62           sta $62
a.780  a9 8b           lda #$8b
a.782  a0 a7           ldy #$a7
a.784  85 22           sta $22
a.786  84 23           sty $23
a.788  4c 43 ae        jmp $ae43
-------------------------------------------------------------------
a.78b  a9 bc           lda #$bc; Lo
a.78d  a0 b9           ldy #$b9; Hi tabulka dat
a.78f  20 a2 bb        jsr $bba2
a.792  20 79 00        jsr $0079; CHRGET bez posunuti
```

```
a795  c9 a9       cmp #$a9; Kod STEP
a797  d0 06       bne $a79f
a799  20 73 00    jsr $0073; CHRGET
a79c  20 8a ad    jsr $ad8a
a79f  20 2b bc    jsr $bc2b
a7a2  20 38 ae    jsr $ae38
a7a5  a5 4a       lda $4a; Hi
a7a7  48          pha
a7a8  a5 49       lda $49; Lo Promenna FOR
a7aa  48          pha
a7ab  a9 81       lda #$81; Kod FOR
a7ad  48          pha
a7ae  20 2c a8    jsr $a82c; test na BREAK
a7b1  a5 7a       lda $7a; Lo
a7b3  a4 7b       ldy $7b; Hi CHRGET
a7b5  c0 02       cpy #$02; ctu z buffru?
a7b7  ea          nop
a7b8  f0 04       beq $a7be
a7ba  85 3d       sta $3d; Lo adresa nasled.
a7bc  84 3e       sty $3e; Hi Prikazu Pro CONT
a7be  a0 00       ldy #$00
a7c0  b1 7a       lda ($7a),y
a7c2  d0 43       bne $a807
a7c4  a0 02       ldy #$02
a7c6  b1 7a       lda ($7a),y
a7c8  18          clc
a7c9  d0 03       bne $a7ce
a7cb  4c 4b a8    jmp $a84b
-------------------------------------------------------------
a7ce  c8          iny
a7cf  b1 7a       lda ($7a),y
a7d1  85 39       sta $39; Lo
a7d3  c8          iny
a7d4  b1 7a       lda ($7a),y
a7d6  85 3a       sta $3a; Hi cislo zPrac. r.
a7d8  98          tya
a7d9  65 7a       adc $7a
a7db  85 7a       sta $7a
a7dd  90 02       bcc $a7e1
a7df  e6 7b       inc $7b
a7e1  6c 08 03    jmp ($0308); realizace Prikazu
a7e4  20 73 00    jsr $0073; CHRGET
a7e7  20 ed a7    jsr $a7ed; hl. skoku k Prikazu
a7ea  4c ae a7    jmp $a7ae; zaver
-------------------------------------------------------------
a7ed  f0 3c       beq $a82b; nic
a7ef  e9 80       sbc #$80
a7f1  90 11       bcc $a804; Promenna
a7f3  c9 23       cmp #$23; TAB( a vyssi?
a7f5  b0 17       bcs $a80e
a7f7  0a          asl
a7f8  a8          tay
a7f9  b9 0d a0    lda $a00d,y; Hi skok podle Prika-
a7fc  48          pha
a7fd  b9 0c a0    lda $a00c,y; Lo zu (z tabulky)
a800  48          pha
a801  4c 73 00    jmp $0073; CHRGET
a804  4c a5 a9    jmp $a9a5; rutina <LET>
```

```
a807 c9 3a        cmp #$3a; Kod dvojtecky
a809 f0 d6        beq $a7e1; vektor pro nasl.prikaz
a80b 4c 08 af     jmp $af08; 'Syntax error'
```

```
a80e c9 4b        cmp #$4b; Kod GO bez 7 bitu
a810 d0 f9        bne $a80b
a812 20 73 00     jsr $0073; CHRGET
a815 a9 a4        lda #$a4; Kod TO
a817 20 ff ae     jsr $aeff
a81a 4c a0 a8     jmp $a8a0; rutina <GOTO>
```

RESTORE

```
a81d 38           sec
a81e a5 2b        lda $2b; Lo
a820 e9 01        sbc #$01
a822 a4 2c        ldy $2c; Hi Pocatek Basicu
a824 b0 01        bcs $a827
a826 88           dey
a827 85 41        sta $41; Lo adresa aktualniho
a829 84 42        sty $42; Hi znaku na dataradku
a82b 60           rts
```

```
a82c 20 e1 ff     jsr $ffe1; test na BREAK
a82f b0 01        bcs $a832
a831 18           clc
a832 d0 3c        bne $a870
a834 a5 7a        lda $7a; Lo
a836 a4 7b        ldy $7b; Hi CHRGET
a838 a6 3a        ldx $3a; Hi cislo zprac. r.
a83a e8           inx
a83b f0 0c        beq $a849
a83d 85 3d        sta $3d; Lo adresa nasledu-
a83f 84 3e        sty $3e; Hi jiciho prikazu
a841 a5 39        lda $39
a843 a4 3a        ldy $3a
a845 85 3b        sta $3b
a847 84 3c        sty $3c
a849 68           pla
a84a 68           pla
a84b a9 81        lda #$81; Lo adresa hlasky
a84d a0 a3        ldy #$a3; Hi BREAK
a84f 90 03        bcc $a854
a851 4c 69 a4     jmp $a469; chybova hlaska
a854 4c 86 e3     jmp $e386
```

CONT

```
a857 d0 17        bne $a870
a859 a2 1a        ldx #$1a
a85b a4 3e        ldy $3e; Hi adresa pro CONT
a85d d0 03        bne $a862
a85f 4c 37 a4     jmp $a437; 'Can't continue'
```

```
a862 a5 3d        lda $3d
a864 85 7a        sta $7a
a866 84 7b        sty $7b
a868 a5 3b        lda $3b
a86a a4 3c        ldy $3c
```

```
a.86c  85 39        sta $39
a.86e  84 3a        sty $3a
a.870  60           rts
```

```
RUN
a.871  08           php
a.872  a9 00        lda #$00
a.874  20 90 ff     jsr $ff90; SETMSG (vyp)
a.877  28           plp
a.878  d0 03        bne $a87d
a.87a  4c 59 a6     jmp $a659; rutina <CLR>
a.87d  20 60 a6     jsr $a660; CLR bez nast. CHRGET
a.880  4c 97 a8     jmp $a897
```

```
GOSUB
a.883  a9 03        lda #$03
a.885  20 fb a3     jsr $a3fb; 'Out of memory' ?
a.888  a5 7b        lda $7b; Hi
a.88a  48           pha
a.88b  a5 7a        lda $7a; Lo CHRGET
a.88d  48           pha
a.88e  a5 3a        lda $3a; Hi cislo zPracovavane
a.890  48           pha
a.891  a5 39        lda $39; Lo radky programu
a.893  48           pha
a.894  a9 8d        lda #$8d; kod GOSUB
a.896  48           pha
a.897  20 79 00     jsr $0079; CHRGET bez posunuti
a.89a  20 a0 a8     jsr $a8a0; rutina <GOTO>
a.89d  4c ae a7     jmp $a7ae
```

```
GOTO
a.8a0  20 6b a9     jsr $a96b; cteni cisla a dec=>hex
a.8a3  20 09 a9     jsr $a909
a.8a6  38           sec
a.8a7  a5 39        lda $39
a.8a9  e5 14        sbc $14
a.8ab  a5 3a        lda $3a
a.8ad  e5 15        sbc $15
a.8af  b0 0b        bcs $a8bc
a.8b1  98           tya
a.8b2  38           sec
a.8b3  65 7a        adc $7a
a.8b5  a6 7b        ldx $7b
a.8b7  90 07        bcc $a8c0
a.8b9  e8           inx
a.8ba  b0 04        bcs $a8c0
a.8bc  a5 2b        lda $2b; Lo
a.8be  a6 2c        ldx $2c; Hi poc. Basicu
a.8c0  20 17 a6     jsr $a617
a.8c3  90 1e        bcc $a8e3
a.8c5  a5 5f        lda $5f
a.8c7  e9 01        sbc #$01
a.8c9  85 7a        sta $7a
a.8cb  a5 60        lda $60
a.8cd  e9 00        sbc #$00
a.8cf  85 7b        sta $7b
a.8d1  60           rts
```

```
RETURN
a8d2  d0 fd        bne $a8d1
a8d4  a9 ff        lda #$ff
a8d6  85 4a        sta $4a
a8d8  20 8a a3     jsr $a38a; hl. v zasobniku
a8db  9a           txs
a8dc  c9 8d        cmp #$8d; Kod GOSUB
a8de  f0 0b        beq $a8eb
a8e0  a2 0c        ldx #$0c; 'Return without gosub'
a8e2  2c a2 11     bit $11a2; 'Undef'd statement'
a8e5  4c 37 a4     jmp $a437; chybova hlaska
a8e8  4c 08 af     jmp $af08; 'Syntax error'
```

```
a8eb  68           pla
a8ec  68           pla
a8ed  85 39        sta $39; Lo cislo zpracovavane
a8ef  68           pla
a8f0  85 3a        sta $3a; Hi radky programu
a8f2  68           pla
a8f3  85 7a        sta $7a; Lo
a8f5  68           pla
a8f6  85 7b        sta $7b; Hi CHRGET
DATA
a8f8  20 06 a9     jsr $a906
a8fb  98           tya
a8fc  18           clc
a8fd  65 7a        adc $7a
a8ff  85 7a        sta $7a
a901  90 02        bcc $a905
a903  e6 7b        inc $7b
a905  60           rts
```

```
a906  a2 3a        ldx #$3a
a908  2c a2 00     bit $00a2
a90b  86 07        stx $07
a90d  a0 00        ldy #$00
a90f  84 08        sty $08
a911  a5 08        lda $08
a913  a6 07        ldx $07
a915  85 07        sta $07
a917  86 08        stx $08
a919  b1 7a        lda ($7a),y
a91b  f0 e8        beq $a905
a91d  c5 08        cmp $08
a91f  f0 e4        beq $a905
a921  c8           iny
a922  c9 22        cmp #$22; uvozovky ?
a924  d0 f3        bne $a919
a926  f0 e9        beq $a911
IF
a928  20 9e ad     jsr $ad9e
a92b  20 79 00     jsr $0079; CHRGET bez posunuti
a92e  c9 89        cmp #$89; Kod GOTO
a930  f0 05        beq $a937
a932  a9 a7        lda #$a7; Kod THEN
a934  20 ff ae     jsr $aeff
a937  a5 61        lda $61
```

```
a.939 d0 05      bne $a940
a.93b 20 09 a9   jsr $a909
a.93e f0 bb      beq $a8fb
a.940 20 79 00   jsr $0079; CHRGET bez posunuti
a.943 b0 03      bcs $a948
a.945 4c a0 a8   jmp $a8a0; rutina <GOTO>
a.948 4c ed a7   jmp $a7ed; zprac.zbytku buffru
-------------------------------------------------
ON
a.94b 20 9e b7   jsr $b79e
a.94e 48         pha
a.94f c9 8d      cmp #$8d; kod GOSUB
a.951 f0 04      beq $a957
a.953 c9 89      cmp #$89; kod GOTO
a.955 d0 91      bne $a8e8; 'Syntax error'
a.957 c6 65      dec $65
a.959 d0 04      bne $a95f
a.95b 68         pla
a.95c 4c ef a7   jmp $a7ef; zprac.ostat.prikazu
-------------------------------------------------
Rutina pro hledani cisla radku podle parametru
a.95f 20 73 00   jsr $0073; CHRGET
a.962 20 6b a9   jsr $a96b
a.965 c9 2c      cmp #$2c; carka ?
a.967 f0 ee      beq $a957
a.969 68         pla
a.96a 60         rts
-------------------------------------------------
a.96b a2 00      ldx #$00
a.96d 86 14      stx $14; Lo
a.96f 86 15      stx $15; Hi radka programu
a.971 b0 f7      bcs $a96a
a.973 e9 2f      sbc #$2f
a.975 85 07      sta $07
a.977 a5 15      lda $15
a.979 85 22      sta $22
a.97b c9 19      cmp #$19
a.97d b0 d4      bcs $a953
a.97f a5 14      lda $14
a.981 0a         asl
a.982 26 22      rol $22
a.984 0a         asl
a.985 26 22      rol $22
a.987 65 14      adc $14
a.989 85 14      sta $14
a.98b a5 22      lda $22
a.98d 65 15      adc $15
a.98f 85 15      sta $15
a.991 06 14      asl $14
a.993 26 15      rol $15
a.995 a5 14      lda $14
a.997 65 07      adc $07
a.999 85 14      sta $14
a.99b 90 02      bcc $a99f
a.99d e6 15      inc $15
a.99f 20 73 00   jsr $0073; CHRGET
a.9a2 4c 71 a9   jmp $a971
```

```
LET
a.9a5  20 8b b0     jsr $b08b; hl.v promennych
a.9a8  85 49        sta $49
a.9aa  84 4a        sty $4a
a.9ac  a9 b2        lda #$b2; kod '='
a.9ae  20 ff ae     jsr $aeff; test na nasl. znak
a.9b1  a5 0e        lda $0e; integer/real
a.9b3  48           pha
a.9b4  a5 0d        lda $0d; retezce/cisla
a.9b6  48           pha
a.9b7  20 9e ad     jsr $ad9e
a.9ba  68           pla
a.9bb  2a           rol
a.9bc  20 90 ad     jsr $ad90
a.9bf  d0 18        bne $a9d9
a.9c1  68           pla
a.9c2  10 12        bpl $a9d6
a.9c4  20 1b bc     jsr $bc1b
a.9c7  20 bf b1     jsr $b1bf
a.9ca  a0 00        ldy #$00
a.9cc  a5 64        lda $64
a.9ce  91 49        sta ($49),y
a.9d0  c8           iny
a.9d1  a5 65        lda $65
a.9d3  91 49        sta ($49),y
a.9d5  60           rts
a.9d6  4c d0 bb     jmp $bbd0
```

```
a.9d9  68           pla
a.9da  a4 4a        ldy $4a
a.9dc  c0 bf        cpy #$bf
a.9de  d0 4c        bne $aa2c
a.9e0  20 a6 b6     jsr $b6a6
a.9e3  c9 06        cmp #$06
a.9e5  d0 3d        bne $aa24
a.9e7  a0 00        ldy #$00
a.9e9  84 61        sty $61
a.9eb  84 66        sty $66
a.9ed  84 71        sty $71
a.9ef  20 1d aa     jsr $aa1d
a.9f2  20 e2 ba     jsr $bae2
a.9f5  e6 71        inc $71
a.9f7  a4 71        ldy $71
a.9f9  20 1d aa     jsr $aa1d
a.9fc  20 0c bc     jsr $bc0c
a.9ff  aa           tax
a.a00  f0 05        beq $aa07
a.a02  e8           inx
a.a03  8a           txa
a.a04  20 ed ba     jsr $baed
a.a07  a4 71        ldy $71
a.a09  c8           iny
a.a0a  c0 06        cpy #$06
a.a0c  d0 df        bne $a9ed
a.a0e  20 e2 ba     jsr $bae2
a.a11  20 9b bc     jsr $bc9b
a.a14  a6 64        ldx $64;  =
```

```
a.a16  a4 63        ldy $63; <
a.a18  a5 65        lda $65; >
a.a1a  4c db ff     jmp $ffdb; SETTIM
-------------------------------------------
a.a1d  b1 22        lda ($22),y
a.a1f  20 80 00     jsr $0080
a.a22  90 03        bcc $a.a27
a.a24  4c 48 b2     jmp $b248
-------------------------------------------
a.a27  e9 2f        sbc #$2f
a.a29  4c 7e bd     jmp $bd7e
-------------------------------------------
a.a2c  a0 02        ldy #$02
a.a2e  b1 64        lda ($64),y
a.a30  c5 34        cmp $34
a.a32  90 17        bcc $a.a4b
a.a34  d0 07        bne $a.a3d
a.a36  88           dey
a.a37  b1 64        lda ($64),y
a.a39  c5 33        cmp $33
a.a3b  90 0e        bcc $a.a4b
a.a3d  a4 65        ldy $65
a.a3f  c4 2e        cpy $2e; Hi
a.a41  90 08        bcc $a.a4b
a.a43  d0 0d        bne $a.a52
a.a45  a5 64        lda $64
a.a47  c5 2d        cmp $2d; Lo Konec Basicu
a.a49  b0 07        bcs $a.a52
a.a4b  a5 64        lda $64
a.a4d  a4 65        ldy $65
a.a4f  4c 68 a.a.   jmp $a.a68
-------------------------------------------
a.a52  a0 00        ldy #$00
a.a54  b1 64        lda ($64),y
a.a56  20 75 b4     jsr $b475
a.a59  a5 50        lda $50
a.a5b  a4 51        ldy $51
a.a5d  85 6f        sta $6f
a.a5f  84 70        sty $70
a.a61  20 7a. b6    jsr $b67a.
a.a64  a9 61        lda #$61
a.a66  a0 00        ldy #$00
a.a68  85 50        sta $50
a.a6a  84 51        sty $51
a.a6c  20 db b6     jsr $b6db
a.a6f  a0 00        ldy #$00
a.a71  b1 50        lda ($50),y
a.a73  91 49        sta ($49),y
a.a75  c8           iny
a.a76  b1 50        lda ($50),y
a.a78  91 49        sta ($49),y
a.a7a  c8           iny
a.a7b  b1 50        lda ($50),y
a.a7d  91 49        sta ($49),y
a.a7f  60           rts
-------------------------------------------
PRINT#
a.a80  20 86 a.a.   jsr $a.a86
```

```
a.a83  4c  b5  ab      jmp  $abb5
-------------------------------------------------------
CMD
a.a86  20  9e  b7      jsr  $b79e
a.a89  f0  05          beq  $a.a90
a.a8b  a9  2c          lda  #$2c; Kod carky
a.a8d  20  ff  ae      jsr  $aeff; Porovnani
a.a90  08              php
a.a91  86  13          stx  $13; cislo periferie CMD
a.a93  20  18  e1      jsr  $e118
a.a96  28              plp
a.a97  4c  a0  a.      jmp  $a.a.a0
-------------------------------------------------------
PRINT
a.a9a  20  21  ab      jsr  $ab21
a.a9d  20  79  00      jsr  $0079; CHRGET bez posunuti
a.aa0  f0  35          beq  $aad7
a.aa2  f0  43          beq  $aae7
a.aa4  c9  a3          cmp  #$a3; Kod TAB(
a.aa6  f0  50          beq  $aaf8
a.aa8  c9  a6          cmp  #$a6; Kod SPC(
a.aaa  18              clc
a.aab  f0  4b          beq  $aaf8
a.aad  c9  2c          cmp  #$2c; Kod carky
a.aaf  f0  37          beq  $aae8
a.ab1  c9  3b          cmp  #$3b; Kod strednik
a.ab3  f0  5e          beq  $ab13
a.ab5  20  9e  ad      jsr  $ad9e
a.ab8  24  0d          bit  $0d
a.aba  30  de          bmi  $a.a9a
a.abc  20  dd  bd      jsr  $bddd
a.abf  20  87  b4      jsr  $b487
a.ac2  20  21  ab      jsr  $ab21
a.ac5  20  3b  ab      jsr  $ab3b
a.ac8  d0  d3          bne  $a.a9d
a.aca  a9  00          lda  #$00
a.acc  9d  00  02      sta  $0200,x; vstupni buffer
a.acf  a2  ff          ldx  #$ff
a.ad1  a0  01          ldy  #$01
a.ad3  a5  13          lda  $13; Kanal CMD
a.ad5  d0  10          bne  $a.ae7
a.ad7  a9  0d          lda  #$0d; Kod <RETURN>
a.ad9  20  47  ab      jsr  $ab47; tisk
a.adc  24  13          bit  $13
a.ade  10  05          bpl  $a.ae5
a.ae0  a9  0a          lda  #$0a
a.ae2  20  47  ab      jsr  $ab47
a.ae5  49  ff          eor  #$ff
a.ae7  60              rts
-------------------------------------------------------
a.ae8  38              sec
a.ae9  20  f0  ff      jsr  $fff0; PLOT (cteni)
a.aec  98              tya
a.aed  38              sec
a.aee  e9  0a          sbc  #$0a
a.af0  b0  fc          bcs  $a.aee
a.af2  49  ff          eor  #$ff
a.af4  69  01          adc  #$01
```

```
a.af6  d0 16        bne $a.b0e
TAB( a. SPC(
a.af8  08           php
a.af9  38           sec
a.afa  20 f0 ff     jsr $fff0; PLOT (cteni)
a.afd  84 09        sty $09
a.aff  20 9b b7     jsr $b79b
a.b02  c9 29        cmp #$29
a.b04  d0 59        bne $a.b5f
a.b06  28           plp
a.b07  90 06        bcc $a.b0f
a.b09  8a.          txa
a.b0a. e5 09        sbc $09
a.b0c  90 05        bcc $a.b13
a.b0e  a.a.         tax
a.b0f  e8           inx
a.b10  ca.          dex
a.b11  d0 06        bne $a.b19
a.b13  20 73 00     jsr $0073; CHRGET
a.b16  4c a2 a.a.   jmp $a.a.a2; dalsi polozka PRINT
-------------------------------------------------------
a.b19  20 3b a.b    jsr $a.b3b
a.b1c  d0 f2        bne $a.b10
a.b1e  20 87 b4     jsr $b487
a.b21  20 a.6 b6    jsr $b6a.6; nastaveni retezce
a.b24  a.a.         tax ; Pocet znaku A=>X
Tisk retezce na. ktery ukazuje $22,23
a.b25  a.0 00       ldy #$00
a.b27  e8           inx
a.b28  ca.          dex
a.b29  f0 bc        beq $a.a.e7; RTS
a.b2b  b1 22        lda ($22),y; cteni znaku retezce
a.b2d  20 47 a.b    jsr $a.b47; tisk
a.b30  c8           iny
a.b31  c9 0d        cmp #$0d
a.b33  d0 f3        bne $a.b28
a.b35  20 e5 a.a.   jsr $a.a.e5
a.b38  4c 28 a.b    jmp $a.b28
-------------------------------------------------------
a.b3b  a.5 13       lda $13; argument CMD
a.b3d  f0 03        beq $a.b42
a.b3f  a.9 20       lda #$20; 'SPACE'
a.b41  2c a.9 1d    bit $1da.9; CRSR vpravo
a.b44  2c a.9 3f    bit $3fa.9; '?'
a.b47  20 0c e1     jsr $e10c; vyslani znaku
a.b4a. 29 ff        and #$ff
a.b4c  60           rts
-------------------------------------------------------
a.b4d  a.5 11       lda. $11; INPUT? GET? READ?
a.b4f  f0 11        beq $a.b62
a.b51  30 04        bmi $a.b57
a.b53  a.0 ff       ldy #$ff
a.b55  d0 04        bne $a.b5b
a.b57  a.5 3f       lda. $3f; Lo
a.b59  a.4 40       ldy $40; Hi cislo data.radku
a.b5b  85 39        sta. $39; Lo
a.b5d  84 3a.       sty $3a.; Hi cislo preklad.r.
a.b5f  4c 08 a.f    jmp $a.f08; 'Syntax error'
```

```
a.b62  a5 13          lda $13; argument CMD
a.b64  f0 05          beq $ab6b
a.b66  a2 18          ldx #$18
a.b68  4c 37 a4       jmp $a437; 'File data'
-------------------------------------------------
a.b6b  a9 0c          lda #$0c
a.b6d  a0 ad          ldy #$ad
a.b6f  20 1e ab       jsr $ab1e; '?Redo from start'
a.b72  a5 3d          lda $3d; Lo adresa nasledujiciho
a.b74  a4 3e          ldy $3e; Hi prikazu pro CONT
a.b76  85 7a          sta $7a; Lo
a.b78  84 7b          sty $7b; Hi CHRGET
a.b7a  60             rts
-------------------------------------------------
GET a GET#
a.b7b  20 a6 b3       jsr $b3a6
a.b7e  c9 23          cmp #$23; Kod #
a.b80  d0 10          bne $ab92
a.b82  20 73 00       jsr $0073; CHRGET
a.b85  20 9e b7       jsr $b79e
a.b88  a9 2c          lda #$2c
a.b8a  20 ff ae       jsr $aeff
a.b8d  86 13          stx $13; argument CMD
a.b8f  20 1e e1       jsr $e11e
a.b92  a2 01          ldx #$01; Lo adresa znaku pro
a.b94  a0 02          ldy #$02; Hi GET (buffer)
a.b96  a9 00          lda #$00
a.b98  8d 01 02       sta $0201; nulovani
a.b9b  a9 40          lda #$40; ident. Kod GET
a.b9d  20 0f ac       jsr $ac0f; rutina pro cteni
a.ba0  a6 13          ldx $13; CMD argument
a.ba2  d0 13          bne $abb7
a.ba4  60             rts
-------------------------------------------------
INPUT#
a.ba5  20 9e b7       jsr $b79e
a.ba8  a9 2c          lda #$2c
a.baa  20 ff ae       jsr $aeff
a.bad  86 13          stx $13
a.baf  20 1e e1       jsr $e11e
a.bb2  20 ce ab       jsr $abce
a.bb5  a5 13          lda $13
a.bb7  20 cc ff       jsr $ffcc; CLRCHN
a.bba  a2 00          ldx #$00
a.bbc  86 13          stx $13
a.bbe  60             rts
-------------------------------------------------
INPUT
a.bbf  c9 22          cmp #$22; uvozovky?
a.bc1  d0 0b          bne $abce
a.bc3  20 bd ae       jsr $aebd
a.bc6  a9 3b          lda #$3b
a.bc8  20 ff ae       jsr $aeff
a.bcb  20 21 ab       jsr $ab21
a.bce  20 a6 b3       jsr $b3a6
a.bd1  a9 2c          lda #$2c
a.bd3  8d ff 01       sta $01ff
```

```
abd6  20 f9 ab     jsr $abf9
abd9  a5 13        lda $13
abdb  f0 0d        beq $abea
abdd  20 b7 ff     jsr $ffb7; READST
abe0  29 02        and #$02
abe2  f0 06        beq $abea
abe4  20 b5 ab     jsr $abb5
abe7  4c f8 a8     jmp $a8f8
------------------------------------------
abea  ad 00 02     lda $0200; pocatek buffru
abed  d0 1e        bne $ac0d
abef  a5 13        lda $13; argument CMD
abf1  d0 e3        bne $abd6
abf3  20 06 a9     jsr $a906
abf6  4c fb a8     jmp $a8fb
------------------------------------------
abf9  a5 13        lda $13; argument CMD
abfb  d0 06        bne $ac03
abfd  20 45 ab     jsr $ab45; tisk '?'
ac00  20 3b ab     jsr $ab3b; Podle arg.CMD <SPACE>
ac03  4c 60 a5     jmp $a560; cteni znaku do buffru
------------------------------------------
READ
ac06  a6 41        ldx $41; Lo adresa aktualniho
ac08  a4 42        ldy $42; Hi znaku z dataradku
ac0a  a9 98        lda #$98; ident. kod READ
ac0c  2c a9 00     bit $00a9; (ident. kod INPUT)
ac0f  85 11        sta $11; typ nacitani (GET...)
ac11  86 43        stx $43; Lo ukazatel pro pri-
ac13  84 44        sty $44; Hi kazy vstupu
ac15  20 8b b0     jsr $b08b; ct. a hledani promenne
ac18  85 49        sta $49; Lo
ac1a  84 4a        sty $4a; Hi promenna
ac1c  a5 7a        lda $7a; Lo
ac1e  a4 7b        ldy $7b; Hi CHRGET
ac20  85 4b        sta $4b
ac22  84 4c        sty $4c
ac24  a6 43        ldx $43
ac26  a4 44        ldy $44
ac28  86 7a        stx $7a; Lo
ac2a  84 7b        sty $7b; Hi prenast. CHRGET
ac2c  20 79 00     jsr $0079; CHRGET bez posunuti
ac2f  d0 20        bne $ac51
ac31  24 11        bit $11
ac33  50 0c        bvc $ac41
ac35  20 24 e1     jsr $e124; obsluha GETIN
ac38  8d 00 02     sta $0200
ac3b  a2 ff        ldx #$ff
ac3d  a0 01        ldy #$01
ac3f  d0 0c        bne $ac4d
ac41  30 75        bmi $acb8
ac43  a5 13        lda $13
ac45  d0 03        bne $ac4a
ac47  20 45 ab     jsr $ab45
ac4a  20 f9 ab     jsr $abf9
ac4d  86 7a        stx $7a
ac4f  84 7b        sty $7b
ac51  20 73 00     jsr $0073
```

```
a.c54  24 0d       bit $0d
a.c56  10 31       bpl $a.c89
a.c58  24 11       bit $11
a.c5a  50 09       bvc $a.c65
a.c5c  e8          inx
a.c5d  86 7a       stx $7a
a.c5f  a9 00       lda #$00
a.c61  85 07       sta $07
a.c63  f0 0c       beq $a.c71
a.c65  85 07       sta $07
a.c67  c9 22       cmp #$22
a.c69  f0 07       beq $a.c72
a.c6b  a9 3a       lda #$3a
a.c6d  85 07       sta $07
a.c6f  a9 2c       lda #$2c
a.c71  18          clc
a.c72  85 08       sta $08
a.c74  a5 7a       lda $7a
a.c76  a4 7b       ldy $7b
a.c78  69 00       adc #$00
a.c7a  90 01       bcc $a.c7d
a.c7c  c8          iny
a.c7d  20 8d b4    jsr $b48d
a.c80  20 e2 b7    jsr $b7e2
a.c83  20 da a9    jsr $a9da
a.c86  4c 91 a.c   jmp $a.c91
-------------------------------------------------------------
a.c89  20 f3 bc    jsr $bcf3
a.c8c  a5 0e       lda $0e; typ  integer/real
a.c8e  20 c2 a9    jsr $a9c2
a.c91  20 79 00    jsr $0079; CHRGET bez posunuti
a.c94  f0 07       beq $a.c9d
a.c96  c9 2c       cmp #$2c; carka?
a.c98  f0 03       beq $a.c9d
a.c9a  4c 4d ab    jmp $ab4d
-------------------------------------------------------------
a.c9d  a5 7a       lda $7a; Lo
a.c9f  a4 7b       ldy $7b; Hi CHRGET
a.ca1  85 43       sta $43; Lo ukazatel pro pri-
a.ca3  84 44       sty $44; Hi kazy vstupu
a.ca5  a5 4b       lda $4b
a.ca7  a4 4c       ldy $4c
a.ca9  85 7a       sta $7a
a.cab  84 7b       sty $7b
a.cad  20 79 00    jsr $0079
a.cb0  f0 2d       beq $a.cdf
a.cb2  20 fd ae    jsr $aefd
a.cb5  4c 15 a.c   jmp $a.c15
-------------------------------------------------------------
a.cb8  20 06 a9    jsr $a906
a.cbb  c8          iny
a.cbc  aa          tax
a.cbd  d0 12       bne $a.cd1
a.cbf  a2 0d       ldx #$0d
a.cc1  c8          iny
a.cc2  b1 7a       lda ($7a),y
a.cc4  f0 6c       beq $a.d32
a.cc6  c8          iny
```

```
acc7 b1 7a      lda ($7a),y
acc9 85 3f      sta $3f; Lo
accb c8         iny
accc b1 7a      lda ($7a),y
acce c8         iny
accf 85 40      sta $40; Hi cislo dataradku
acd1 20 fb a8   jsr $a8fb
acd4 20 79 00   jsr $0079; CHRGET bez posunuti
acd7 aa         tax
acd8 e0 83      cpx #$83; Kod DATA
acda d0 dc      bne $acb8
acdc 4c 51 ac   jmp $ac51
```

```
acdf a5 43      lda $43; Lo ukazatel pro pri-
ace1 a4 44      ldy $44; Hi kazy vstupu
ace3 a6 11      ldx $11
ace5 10 03      bpl $acea
ace7 4c 27 a8   jmp $a827
```

```
acea a0 00      ldy #$00
acec b1 43      lda ($43),y
acee f0 0b      beq $acfb
acf0 a5 13      lda $13
acf2 d0 07      bne $acfb
acf4 a9 fc      lda #$fc
acf6 a0 ac      ldy #$ac
acf8 4c 1e ab   jmp $ab1e; '?Extra ignored'
acfb 60         rts
```

```
acfc 3f 45 58 54 52 41 20 49; ?extra i
ad04 47 4e 4f 52 45 44 0d 00; gnored
ad0c 3f 52 45 44 4f 20 46 52; ?redo fr
ad14 4f 4d 20 53 54 41 52 54; om start
ad1c 0d 00; <RETURN>;$00-konec hlasky
```

```
NEXT
ad1e d0 04      bne $ad24
ad20 a0 00      ldy #$00
ad22 f0 03      beq $ad27
ad24 20 8b b0   jsr $b08b
ad27 85 49      sta $49; Lo
ad29 84 4a      sty $4a; Hi promenna FOR
ad2b 20 8a a3   jsr $a38a; hledani v zasobniku
ad2e f0 05      beq $ad35
ad30 a2 0a      ldx #$0a
ad32 4c 37 a4   jmp $a437; 'Next without for'
```

```
ad35 9a         txs
ad36 8a         txa
ad37 18         clc
ad38 69 04      adc #$04
ad3a 48         pha
ad3b 69 06      adc #$06
ad3d 85 24      sta $24
ad3f 68         pla
ad40 a0 01      ldy #$01
```

```
a.d42  20 a2 bb     jsr $bba2
a.d45  ba           tsx
a.d46  bd 09 01     lda $0109,x; zasobnik
a.d49  85 66        sta $66
a.d4b  a5 49        lda $49; Lo
a.d4d  a4 4a        ldy $4a; Hi promenna FOR
a.d4f  20 67 b8     jsr $b867
a.d52  20 d0 bb     jsr $bbd0
a.d55  a0 01        ldy #$01
a.d57  20 5d bc     jsr $bc5d
a.d5a  ba           tsx
a.d5b  38           sec
a.d5c  fd 09 01     sbc $0109,x
a.d5f  f0 17        beq $ad78
a.d61  bd 0f 01     lda $010f,x
a.d64  85 39        sta $39
a.d66  bd 10 01     lda $0110,x
a.d69  85 3a        sta $3a
a.d6b  bd 12 01     lda $0112,x
a.d6e  85 7a        sta $7a
a.d70  bd 11 01     lda $0111,x
a.d73  85 7b        sta $7b
a.d75  4c ae a7     jmp $a7ae
-------------------------------------------------------
a.d78  8a           txa
a.d79  69 11        adc #$11
a.d7b  aa           tax
a.d7c  9a           txs
a.d7d  20 79 00     jsr $0079; CHRGET bez posunuti
a.d80  c9 2c        cmp #$2c; carka?
a.d82  d0 f1        bne $ad75
a.d84  20 73 00     jsr $0073; CHRGET
a.d87  20 24 ad     jsr $ad24
a.d8a  20 9e ad     jsr $ad9e
a.d8d  18           clc
a.d8e  24 38        bit $38
a.d90  24 0d        bit $0d
a.d92  30 03        bmi $ad97
a.d94  b0 03        bcs $ad99
a.d96  60           rts
-------------------------------------------------------
a.d97  b0 fd        bcs $ad96
a.d99  a2 16        ldx #$16
a.d9b  4c 37 a4     jmp $a437; 'Tape mismatch'
-------------------------------------------------------
a.d9e  a6 7a        ldx $7a
a.da0  d0 02        bne $ada4
a.da2  c6 7b        dec $7b
a.da4  c6 7a        dec $7a
a.da6  a2 00        ldx #$00
a.da8  24 48        bit $48
a.daa  8a           txa
a.dab  48           pha
a.dac  a9 01        lda #$01
a.dae  20 fb a3     jsr $a3fb
a.db1  20 83 ae     jsr $ae83
a.db4  a9 00        lda #$00
a.db6  85 4d        sta $4d
```

```
a.db8  20 79 00     jsr $0079; CHRGET bez posunuti
a.dbb  38           sec
a.dbc  e9 b1        sbc #$b1
a.dbe  90 17        bcc $a.dd7
a.dc0  c9 03        cmp #$03
a.dc2  b0 13        bcs $a.dd7
a.dc4  c9 01        cmp #$01
a.dc6  2a           rol
a.dc7  49 01        eor #$01
a.dc9  45 4d        eor $4d
a.dcb  c5 4d        cmp $4d
a.dcd  90 61        bcc $a.e30
a.dcf  85 4d        sta $4d
a.dd1  20 73 00     jsr $0073; CHRGET
a.dd4  4c bb a.d    jmp $a.dbb
-------------------------------------------------------
a.dd7  a.6 4d       ldx $4d
a.dd9  d0 2c        bne $a.e07
a.ddb  b0 7b        bcs $a.e58
a.ddd  69 07        a.dc #$07
a.ddf  90 77        bcc $a.e58
a.de1  65 0d        a.dc $0d
a.de3  d0 03        bne $a.de8
a.de5  4c 3d b6     jmp $b63d
-------------------------------------------------------
a.de8  69 ff        a.dc #$ff
a.dea. 85 22        sta. $22
a.dec  0a.          asl
a.ded  65 22        a.dc $22
a.def  a.8          tay
a.df0  68           pla
a.df1  d9 80 a.0    cmp $a.080,y; tab.skoku s dodatkem
a.df4  b0 67        bcs $a.e5d
a.df6  20 8d a.d    jsr $a.d8d
a.df9  48           pha
a.dfa. 20 20 a.e    jsr $a.e20
a.dfd  68           pla
a.dfe  a.4 4b       ldy $4b
a.e00  10 17        bpl $a.e19
a.e02  a.a.         tax
a.e03  f0 56        beq $a.e5b
a.e05  d0 5f        bne $a.e66
a.e07  46 0d        lsr $0d
a.e09  8a.          txa
a.e0a  2a.          rol
a.e0b  a.6 7a.      ldx $7a
a.e0d  d0 02        bne $a.e11
a.e0f  c6 7b        dec $7b
a.e11  c6 7a.       dec $7a
a.e13  a.0 1b       ldy #$1b
a.e15  85 4d        sta $4d
a.e17  d0 d7        bne $adf0
a.e19  d9 80 a.0    cmp $a.080,y; dodatkova hodnota
a.e1c  b0 48        bcs $a.e66
a.e1e  90 d9        bcc $a.df9
a.e20  b9 82 a.0    lda $a.082,y; Hi skok (+;-;atd.)
a.e23  48           pha
a.e24  b9 81 a.0    lda $a.081,y; Lo skok (+;-;atd.)
```

```
a.e27 48          pha
a.e28 20 33 a.e   jsr $a.e33; Provedeni operace
a.e2b a5 4d       lda $4d
a.e2d 4c a9 a.d   jmp $a.da9
-------------------------------------------------
a.e30 4c 08 af    jmp $af08; 'Syntax error'
-------------------------------------------------
a.e33 a5 66       lda $66
a.e35 be 80 a0    ldx $a080,y
a.e38 a8          tay
a.e39 68          pla
a.e3a 85 22       sta $22
a.e3c e6 22       inc $22
a.e3e 68          pla
a.e3f 85 23       sta $23
a.e41 98          tya
a.e42 48          pha
a.e43 20 1b bc    jsr $bc1b
a.e46 a5 65       lda $65
a.e48 48          pha
a.e49 a5 64       lda $64
a.e4b 48          pha
a.e4c a5 63       lda $63
a.e4e 48          pha
a.e4f a5 62       lda $62
a.e51 48          pha
a.e52 a5 61       lda $61
a.e54 48          pha
a.e55 6c 22 00    jmp ($0022); skok podle operace
-------------------------------------------------
a.e58 a0 ff       ldy #$ff
a.e5a 68          pla
a.e5b f0 23       beq $a.e80
a.e5d c9 64       cmp #$64
a.e5f f0 03       beq $a.e64
a.e61 20 8d a.d   jsr $a.d8d
a.e64 84 4b       sty $4b
a.e66 68          pla
a.e67 4a          lsr
a.e68 85 12       sta $12
a.e6a 68          pla
a.e6b 85 69       sta $69
a.e6d 68          pla
a.e6e 85 6a       sta $6a
a.e70 68          pla
a.e71 85 6b       sta $6b
a.e73 68          pla
a.e74 85 6c       sta $6c
a.e76 68          pla
a.e77 85 6d       sta $6d
a.e79 68          pla
a.e7a 85 6e       sta $6e
a.e7c 45 66       eor $66
a.e7e 85 6f       sta $6f
a.e80 a5 61       lda $61
a.e82 60          rts
-------------------------------------------------
a.e83 6c 0a 03    jmp ($030a); aritmet. procedura
```

```
---------------------------------------------------------------
ae86  a9 00           lda #$00
ae88  85 0d           sta $0d; $00-cisla
ae8a  20 73 00        jsr $0073; CHRGET
ae8d  b0 03           bcs $ae92
ae8f  4c f3 bc        jmp $bcf3
---------------------------------------------------------------
ae92  20 13 b1        jsr $b113
ae95  90 03           bcc $ae9a
ae97  4c 28 af        jmp $af28
---------------------------------------------------------------
ae9a  c9 ff           cmp #$ff
ae9c  d0 0f           bne $aead
ae9e  a9 a8           lda #$a8; Lo
aea0  a0 ae           ldy #$ae; Hi adresa dat
aea2  20 a2 bb        jsr $bba2
aea5  4c 73 00        jmp $0073; CHRGET
---------------------------------------------------------------

aea8  82 49 0f da a1; data

---------------------------------------------------------------
aead  c9 2e           cmp #$2e; tecka?
aeaf  f0 de           beq $ae8f
aeb1  c9 ab           cmp #$ab; '-'
aeb3  f0 58           beq $af0d
aeb5  c9 aa           cmp #$aa; '+'
aeb7  f0 d1           beq $ae8a
aeb9  c9 22           cmp #$22; uvozovky?
aebb  d0 0f           bne $aecc
aebd  a5 7a           lda $7a; Lo
aebf  a4 7b           ldy $7b; Hi CHRGET
aec1  69 00           adc #$00
aec3  90 01           bcc $aec6
aec5  c8              iny
aec6  20 87 b4        jsr $b487
aec9  4c e2 b7        jmp $b7e2
---------------------------------------------------------------
aecc  c9 a8           cmp #$a8; Kod NOT
aece  d0 13           bne $aee3
aed0  a0 18           ldy #$18
aed2  d0 3b           bne $af0f
aed4  20 bf b1        jsr $b1bf
aed7  a5 65           lda $65
aed9  49 ff           eor #$ff
aedb  a8              tay
aedc  a5 64           lda $64
aede  49 ff           eor #$ff
aee0  4c 91 b3        jmp $b391
---------------------------------------------------------------
aee3  c9 a5           cmp #$a5; Kod FN
aee5  d0 03           bne $aeea
aee7  4c f4 b3        jmp $b3f4
---------------------------------------------------------------
aeea  c9 b4           cmp #$b4; Kod SGN
aeec  90 03           bcc $aef1
aeee  4c a7 af        jmp $afa7
aef1  20 fa ae        jsr $aefa
```

```
a.ef4 20 9e ad     jsr $ad9e
a.ef7 a9 29        lda #$29
a.ef9 2c a9 28     bit $28a9
a.efc 2c a9 2c     bit $2ca9
a.eff a0 00        ldy #$00
a.f01 d1 7a        cmp ($7a),y
a.f03 d0 03        bne $a.f08
a.f05 4c 73 00     jmp $0073
-------------------------------------------------
a.f08 a2 0b        ldx #$0b
Hlaska s Kodem $0b-'Syntax error'
a.f0a 4c 37 a4     jmp $a.437
-------------------------------------------------
Rutiny pro aritmeticke a logicke operace
a.f0d a0 15        ldy #$15
a.f0f 68           pla
a.f10 68           pla
a.f11 4c fa ad     jmp $adfa
a.f14 38           sec
a.f15 a5 64        lda $64
a.f17 e9 00        sbc #$00
a.f19 a5 65        lda $65
a.f1b e9 a0        sbc #$a0
a.f1d 90 08        bcc $af27
a.f1f a9 a2        lda #$a2
a.f21 e5 64        sbc $64
a.f23 a9 e3        lda #$e3
a.f25 e5 65        sbc $65
a.f27 60           rts
-------------------------------------------------
a.f28 20 8b b0     jsr $b08b
a.f2b 85 64        sta $64
a.f2d 84 65        sty $65
a.f2f a6 45        ldx $45
a.f31 a4 46        ldy $46
a.f33 a5 0d        lda $0d
a.f35 f0 26        beq $af5d
a.f37 a9 00        lda #$00
a.f39 85 70        sta $70
a.f3b 20 14 af     jsr $af14
a.f3e 90 1c        bcc $af5c
a.f40 e0 54        cpx #$54
a.f42 d0 18        bne $af5c
a.f44 c0 c9        cpy #$c9
a.f46 d0 14        bne $af5c
a.f48 20 84 af     jsr $af84
a.f4b 84 5e        sty $5e
a.f4d 88           dey
a.f4e 84 71        sty $71
a.f50 a0 06        ldy #$06
a.f52 84 5d        sty $5d
a.f54 a0 24        ldy #$24
a.f56 20 68 be     jsr $be68
a.f59 4c 6f b4     jmp $b46f
a.f5c 60           rts
-------------------------------------------------
a.f5d 24 0e        bit $0e
a.f5f 10 0d        bpl $af6e
```

```
af61  a0 00        ldy #$00
af63  b1 64        lda ($64),y
af65  aa           tax
af66  c8           iny
af67  b1 64        lda ($64),y
af69  a8           tay
af6a  8a           txa
af6b  4c 91 b3     jmp $b391
------------------------------------------------------------
af6e  20 14 af     jsr $af14
af71  90 2d        bcc $afa0
af73  e0 54        cpx #$54
af75  d0 1b        bne $af92
af77  c0 49        cpy #$49
af79  d0 25        bne $afa0
af7b  20 84 af     jsr $af84
af7e  98           tya
af7f  a2 a0        ldx #$a0
af81  4c 4f bc     jmp $bc4f
------------------------------------------------------------
af84  20 de ff     jsr $ffde; RDTIM
af87  86 64        stx $64; T+1
af89  84 63        sty $63; T
af8b  85 65        sta $65; T+2
af8d  a0 00        ldy #$00
af8f  84 62        sty $62
af91  60           rts
------------------------------------------------------------
af92  e0 53        cpx #$53
af94  d0 0a        bne $afa0
af96  c0 54        cpy #$54
af98  d0 06        bne $afa0
af9a  20 b7 ff     jsr $ffb7; READST
af9d  4c 3c bc     jmp $bc3c
------------------------------------------------------------
afa0  a5 64        lda $64
afa2  a4 65        ldy $65
afa4  4c a2 bb     jmp $bba2
------------------------------------------------------------
afa7  0a           asl
afa8  48           pha
afa9  aa           tax
afaa  20 73 00     jsr $0073; CHRGET
afad  e0 8f        cpx #$8f
afaf  90 20        bcc $afd1
afb1  20 fa ae     jsr $aefa
afb4  20 9e ad     jsr $ad9e
afb7  20 fd ae     jsr $aefd
afba  20 8f ad     jsr $ad8f
afbd  68           pla
afbe  aa           tax
afbf  a5 65        lda $65
afc1  48           pha
afc2  a5 64        lda $64
afc4  48           pha
afc5  8a           txa
afc6  48           pha
afc7  20 9e b7     jsr $b79e
```

```
a.fca 68          pla
a.fcb a8          tay
a.fcc 8a          txa
a.fcd 48          pha
a.fce 4c d6 af    jmp $afd6
------------------------------------------------------------
a.fd1 20 f1 ae    jsr $aef1
a.fd4 68          pla
a.fd5 a8          tay
a.fd6 b9 ea 9f    lda $9fea,y
a.fd9 85 55       sta $55; Lo
a.fdb b9 eb 9f    lda $9feb,y
a.fde 85 56       sta $56; Hi vektor skoku pro FN
a.fe0 20 54 00    jsr $0054; skok na funkci
a.fe3 4c 8d ad    jmp $ad8d
------------------------------------------------------------
DR
a.fe6 a0 ff       ldy #$ff
(AND)
a.fe8 2c a0 00    bit $00a0
a.feb 84 0b       sty $0b
a.fed 20 bf b1    jsr $b1bf
a.ff0 a5 64       lda $64
a.ff2 45 0b       eor $0b
a.ff4 85 07       sta $07
a.ff6 a5 65       lda $65
a.ff8 45 0b       eor $0b
a.ffa 85 08       sta $08
a.ffc 20 fc bb    jsr $bbfc
a.fff 20 bf b1    jsr $b1bf
b002 a5 65        lda $65
b004 45 0b        eor $0b
b006 25 08        and $08
b008 45 0b        eor $0b
b00a a8           tay
b00b a5 64        lda $64
b00d 45 0b        eor $0b
b00f 25 07        and $07
b011 45 0b        eor $0b
b013 4c 91 b3     jmp $b391
------------------------------------------------------------
<
b016 20 90 ad     jsr $ad90
b019 b0 13        bcs $b02e
b01b a5 6e        lda $6e
b01d 09 7f        ora #$7f
b01f 25 6a        and $6a
b021 85 6a        sta $6a
b023 a9 69        lda #$69
b025 a0 00        ldy #$00
b027 20 5b bc     jsr $bc5b
b02a aa           tax
b02b 4c 61 b0     jmp $b061
------------------------------------------------------------
b02e a9 00        lda #$00
b030 85 0d        sta $0d; typ-numericky
b032 c6 4d        dec $4d
b034 20 a6 b6     jsr $b6a6
```

```
b037  85 61        sta $61
b039  86 62        stx $62
b03b  84 63        sty $63
b03d  a5 6c        lda $6c
b03f  a4 6d        ldy $6d
b041  20 aa b6     jsr $b6aa
b044  86 6c        stx $6c
b046  84 6d        sty $6d
b048  aa           tax
b049  38           sec
b04a  e5 61        sbc $61
b04c  f0 08        beq $b056
b04e  a9 01        lda #$01
b050  90 04        bcc $b056
b052  a6 61        ldx $61
b054  a9 ff        lda #$ff
b056  85 66        sta $66
b058  a0 ff        ldy #$ff
b05a  e8           inx
b05b  c8           iny
b05c  ca           dex
b05d  d0 07        bne $b066
b05f  a6 66        ldx $66
b061  30 0f        bmi $b072
b063  18           clc
b064  90 0c        bcc $b072
b066  b1 6c        lda ($6c),y
b068  d1 62        cmp ($62),y
b06a  f0 ef        beq $b05b
b06c  a2 ff        ldx #$ff
b06e  b0 02        bcs $b072
b070  a2 01        ldx #$01
b072  e8           inx
b073  8a           txa
b074  2a           rol
b075  25 12        and $12
b077  f0 02        beq $b07b
b079  a9 ff        lda #$ff
b07b  4c 3c bc     jmp $bc3c
------------------------------------------------------------
b07e  20 fd ae     jsr $aefd
DIM
b081  aa           tax
b082  20 90 b0     jsr $b090
b085  20 79 00     jsr $0079; CHRGET bez posunuti
b088  d0 f4        bne $b07e
b08a  60           rts
------------------------------------------------------------
b08b  a2 00        ldx #$00
b08d  20 79 00     jsr $0079
b090  86 0c        stx $0c
b092  85 45        sta $45
b094  20 79 00     jsr $0079
b097  20 13 b1     jsr $b113
b09a  b0 03        bcs $b09f
b09c  4c 08 af     jmp $af08; 'Syntax error'
------------------------------------------------------------
b09f  a2 00        ldx #$00
```

```
b0a1  86 0d       stx $0d; $00-cisla
b0a3  86 0e       stx $0e; $00-real
b0a5  20 73 00    jsr $0073; CHRGET
b0a8  90 05       bcc $b0af
b0aa  20 13 b1    jsr $b113
b0ad  90 0b       bcc $b0ba
b0af  aa          tax
b0b0  20 73 00    jsr $0073
b0b3  90 fb       bcc $b0b0
b0b5  20 13 b1    jsr $b113
b0b8  b0 f6       bcs $b0b0
b0ba  c9 24       cmp #$24; '%' ?
b0bc  d0 06       bne $b0c4
b0be  a9 ff       lda #$ff
b0c0  85 0d       sta $0d
b0c2  d0 10       bne $b0d4
b0c4  c9 25       cmp #$25; '$' ?
b0c6  d0 13       bne $b0db
b0c8  a5 10       lda $10
b0ca  d0 d0       bne $b09c
b0cc  a9 80       lda #$80
b0ce  85 0e       sta $0e
b0d0  05 45       ora $45
b0d2  85 45       sta $45
b0d4  8a          txa
b0d5  09 80       ora #$80
b0d7  aa          tax
b0d8  20 73 00    jsr $0073; CHRGET
b0db  86 46       stx $46
b0dd  38          sec
b0de  05 10       ora $10
b0e0  e9 28       sbc #$28
b0e2  d0 03       bne $b0e7
b0e4  4c d1 b1    jmp $b1d1
-------------------------------------------------------
b0e7  a0 00       ldy #$00
b0e9  84 10       sty $10
b0eb  a5 2d       lda $2d; Lo
b0ed  a6 2e       ldx $2e; Hi Pocatek promennych
b0ef  86 60       stx $60
b0f1  85 5f       sta $5f
b0f3  e4 30       cpx $30; Hi Pocatek Poli
b0f5  d0 04       bne $b0fb
b0f7  c5 2f       cmp $2f; Lo Pocatek Poli
b0f9  f0 22       beq $b11d
b0fb  a5 45       lda $45; Lo jmeno aktual-
b0fd  d1 5f       cmp ($5f),y
b0ff  d0 08       bne $b109
b101  a5 46       lda $46; Hi ni Promenne
b103  c8          iny
b104  d1 5f       cmp ($5f),y
b106  f0 7d       beq $b185
b108  88          dey
b109  18          clc
b10a  a5 5f       lda $5f
b10c  69 07       adc #$07
b10e  90 e1       bcc $b0f1
b110  e8          inx
```

```
b111  d0 dc        bne $b0ef
b113  c9 41        cmp #$41
b115  90 05        bcc $b11c
b117  e9 5b        sbc #$5b
b119  38           sec
b11a  e9 a5        sbc #$a5
b11c  60           rts
-------------------------------------------------------
b11d  68           pla
b11e  48           pha
b11f  c9 2a        cmp #$2a
b121  d0 05        bne $b128
b123  a9 13        lda #$13
b125  a0 bf        ldy #$bf
b127  60           rts
-------------------------------------------------------
b128  a5 45        lda $45; Lo jmeno aktual-
b12a  a4 46        ldy $46; Hi ni promenne
b12c  c9 54        cmp #$54; T? (cast TI)
b12e  d0 0b        bne $b13b
b130  c0 c9        cpy #$c9; I$ ? (cast TI$)
b132  f0 ef        beq $b123
b134  c0 49        cpy #$49; I ? (cast TI)
b136  d0 03        bne $b13b
b138  4c 08 af     jmp $af08; 'Syntax error'
b13b  c9 53        cmp #$53; S ? (cast ST)
b13d  d0 04        bne $b143
b13f  c0 54        cpy #$54; T ? (cast ST)
b141  f0 f5        beq $b138
b143  a5 2f        lda $2f; Lo
b145  a4 30        ldy $30; Hi Pocatek Poli
b147  85 5f        sta $5f
b149  84 60        sty $60
b14b  a5 31        lda $31; Lo
b14d  a4 32        ldy $32; Hi Konec Poli
b14f  85 5a        sta $5a
b151  84 5b        sty $5b
b153  18           clc
b154  69 07        adc #$07
b156  90 01        bcc $b159
b158  c8           iny
b159  85 58        sta $58
b15b  84 59        sty $59
b15d  20 b8 a3     jsr $a3b8
b160  a5 58        lda $58
b162  a4 59        ldy $59
b164  c8           iny
b165  85 2f        sta $2f
b167  84 30        sty $30
b169  a0 00        ldy #$00
b16b  a5 45        lda $45
b16d  91 5f        sta ($5f),y
b16f  c8           iny
b170  a5 46        lda $46
b172  91 5f        sta ($5f),y
b174  a9 00        lda #$00
b176  c8           iny
b177  91 5f        sta ($5f),y
```

```
b179  c8           iny
b17a  91 5f        sta ($5f),y
b17c  c8           iny
b17d  91 5f        sta ($5f),y
b17f  c8           iny
b180  91 5f        sta ($5f),y
b182  c8           iny
b183  91 5f        sta ($5f),y
b185  a5 5f        lda $5f
b187  18           clc
b188  69 02        adc #$02
b18a  a4 60        ldy $60
b18c  90 01        bcc $b18f
b18e  c8           iny
b18f  85 47        sta $47
b191  84 48        sty $48
b193  60           rts
--------------------------------------------------------------
b194  a5 0b        lda $0b
b196  0a           asl
b197  69 05        adc #$05
b199  65 5f        adc $5f
b19b  a4 60        ldy $60
b19d  90 01        bcc $b1a0
b19f  c8           iny
b1a0  85 58        sta $58
b1a2  84 59        sty $59
b1a4  60           rts
--------------------------------------------------------------
b1a5  90 80        bcc $b127
b1a7  00           brk
b1a8  00           brk
b1a9  00           brk
--------------------------------------------------------------
Procedura Pro Prevod Real => Integer
b1aa  20 bf b1     jsr $b1bf
b1ad  a5 64        lda $64
b1af  a4 65        ldy $65
b1b1  60           rts
--------------------------------------------------------------
b1b2  20 73 00     jsr $0073; CHRGET
b1b5  20 9e ad     jsr $ad9e
b1b8  20 8d ad     jsr $ad8d
b1bb  a5 66        lda $66
b1bd  30 0d        bmi $b1cc
b1bf  a5 61        lda $61
b1c1  c9 90        cmp #$90
b1c3  90 09        bcc $b1ce
b1c5  a9 a5        lda #$a5
b1c7  a0 b1        ldy #$b1
b1c9  20 5b bc     jsr $bc5b
b1cc  d0 7a        bne $b248
b1ce  4c 9b bc     jmp $bc9b
--------------------------------------------------------------
b1d1  a5 0c        lda $0c
b1d3  05 0e        ora $0e
b1d5  48           pha
b1d6  a5 0d        lda $0d
```

```
b1d8  48          pha
b1d9  a0 00       ldy #$00
b1db  98          tya
b1dc  48          pha
b1dd  a5 46       lda $46
b1df  48          pha
b1e0  a5 45       lda $45
b1e2  48          pha
b1e3  20 b2 b1    jsr $b1b2
b1e6  68          pla
b1e7  85 45       sta $45; Lo nazev aktual-
b1e9  68          pla
b1ea  85 46       sta $46; Hi ni promenne
b1ec  68          pla
b1ed  a8          tay
b1ee  ba          tsx
b1ef  bd 02 01    lda $0102,x
b1f2  48          pha
b1f3  bd 01 01    lda $0101,x
b1f6  48          pha
b1f7  a5 64       lda $64
b1f9  9d 02 01    sta $0102,x
b1fc  a5 65       lda $65
b1fe  9d 01 01    sta $0101,x
b201  c8          iny
b202  20 79 00    jsr $0079
b205  c9 2c       cmp #$2c
b207  f0 d2       beq $b1db
b209  84 0b       sty $0b
b20b  20 f7 ae    jsr $aef7
b20e  68          pla
b20f  85 0d       sta $0d; typ cisla/retezce
b211  68          pla
b212  85 0e       sta $0e; integer/real
b214  29 7f       and #$7f
b216  85 0c       sta $0c
b218  a6 2f       ldx $2f; Lo
b21a  a5 30       lda $30; Hi pocatek poli
b21c  86 5f       stx $5f
b21e  85 60       sta $60
b220  c5 32       cmp $32; Hi konec poli
b222  d0 04       bne $b228
b224  e4 31       cpx $31; Lo konec poli
b226  f0 39       beq $b261
b228  a0 00       ldy #$00
b22a  b1 5f       lda ($5f),y
b22c  c8          iny
b22d  c5 45       cmp $45
b22f  d0 06       bne $b237
b231  a5 46       lda $46
b233  d1 5f       cmp ($5f),y
b235  f0 16       beq $b24d
b237  c8          iny
b238  b1 5f       lda ($5f),y
b23a  18          clc
b23b  65 5f       adc $5f
b23d  aa          tax
b23e  c8          iny
```

```
b23f  b1 5f     lda ($5f),y
b241  65 60     adc $60
b243  90 d7     bcc $b21c
b245  a2 12     ldx #$12; 'Bad subscript'
b247  2c a2 0e  bit $0ea2; ('Illegal quantity')
b24a  4c 37 a4  jmp $a437; chybova hlaska
--------------------------------------------------
b24d  a2 13     ldx #$13
b24f  a5 0c     lda $0c
b251  d0 f7     bne $b24a
b253  20 94 b1  jsr $b194
b256  a5 0b     lda $0b
b258  a0 04     ldy #$04
b25a  d1 5f     cmp ($5f),y
b25c  d0 e7     bne $b245
b25e  4c ea b2  jmp $b2ea
--------------------------------------------------
b261  20 94 b1  jsr $b194
b264  20 08 a4  jsr $a408
b267  a0 00     ldy #$00
b269  84 72     sty $72
b26b  a2 05     ldx #$05
b26d  a5 45     lda $45
b26f  91 5f     sta ($5f),y
b271  10 01     bpl $b274
b273  ca        dex
b274  c8        iny
b275  a5 46     lda $46
b277  91 5f     sta ($5f),y
b279  10 02     bpl $b27d
b27b  ca        dex
b27c  ca        dex
b27d  86 71     stx $71
b27f  a5 0b     lda $0b
b281  c8        iny
b282  c8        iny
b283  c8        iny
b284  91 5f     sta ($5f),y
b286  a2 0b     ldx #$0b
b288  a9 00     lda #$00
b28a  24 0c     bit $0c
b28c  50 08     bvc $b296
b28e  68        pla
b28f  18        clc
b290  69 01     adc #$01
b292  aa        tax
b293  68        pla
b294  69 00     adc #$00
b296  c8        iny
b297  91 5f     sta ($5f),y
b299  c8        iny
b29a  8a        txa
b29b  91 5f     sta ($5f),y
b29d  20 4c b3  jsr $b34c
b2a0  86 71     stx $71
b2a2  85 72     sta $72
b2a4  a4 22     ldy $22
b2a6  c6 0b     dec,$0b
```

```
b2a8  d0 dc     bne $b286
b2aa  65 59     adc $59
b2ac  b0 5d     bcs $b30b
b2ae  85 59     sta $59
b2b0  a8        tay
b2b1  8a        txa
b2b2  65 58     adc $58
b2b4  90 03     bcc $b2b9
b2b6  c8        iny
b2b7  f0 52     beq $b30b
b2b9  20 08 a4  jsr $a408
b2bc  85 31     sta $31
b2be  84 32     sty $32
b2c0  a9 00     lda #$00
b2c2  e6 72     inc $72
b2c4  a4 71     ldy $71
b2c6  f0 05     beq $b2cd
b2c8  88        dey
b2c9  91 58     sta ($58),y
b2cb  d0 fb     bne $b2c8
b2cd  c6 59     dec $59
b2cf  c6 72     dec $72
b2d1  d0 f5     bne $b2c8
b2d3  e6 59     inc $59
b2d5  38        sec
b2d6  a5 31     lda $31
b2d8  e5 5f     sbc $5f
b2da  a0 02     ldy #$02
b2dc  91 5f     sta ($5f),y
b2de  a5 32     lda $32
b2e0  c8        iny
b2e1  e5 60     sbc $60
b2e3  91 5f     sta ($5f),y
b2e5  a5 0c     lda $0c
b2e7  d0 62     bne $b34b
b2e9  c8        iny
b2ea  b1 5f     lda ($5f),y
b2ec  85 0b     sta $0b
b2ee  a9 00     lda #$00
b2f0  85 71     sta $71
b2f2  85 72     sta $72
b2f4  c8        iny
b2f5  68        pla
b2f6  aa        tax
b2f7  85 64     sta $64
b2f9  68        pla
b2fa  85 65     sta $65
b2fc  d1 5f     cmp ($5f),y
b2fe  90 0e     bcc $b30e
b300  d0 06     bne $b308
b302  c8        iny
b303  8a        txa
b304  d1 5f     cmp ($5f),y
b306  90 07     bcc $b30f
b308  4c 45 b2  jmp $b245
b30b  4c 35 a4  jmp $a435
```

```
b30e  c8        iny
```

```
b30f  a5 72        lda $72
b311  05 71        ora $71
b313  18           clc
b314  f0 0a        beq $b320
b316  20 4c b3     jsr $b34c
b319  8a           txa
b31a  65 64        adc $64
b31c  aa           tax
b31d  98           tya
b31e  a4 22        ldy $22
b320  65 65        adc $65
b322  86 71        stx $71
b324  c6 0b        dec $0b
b326  d0 ca        bne $b2f2
b328  85 72        sta $72
b32a  a2 05        ldx #$05
b32c  a5 45        lda $45
b32e  10 01        bpl $b331
b330  ca           dex
b331  a5 46        lda $46
b333  10 02        bpl $b337
b335  ca           dex
b336  ca           dex
b337  86 28        stx $28
b339  a9 00        lda #$00
b33b  20 55 b3     jsr $b355
b33e  8a           txa
b33f  65 58        adc $58
b341  85 47        sta $47
b343  98           tya
b344  65 59        adc $59
b346  85 48        sta $48
b348  a8           tay
b349  a5 47        lda $47
b34b  60           rts
-------------------------------------------------------
b34c  84 22        sty $22
b34e  b1 5f        lda ($5f),y
b350  85 28        sta $28
b352  88           dey
b353  b1 5f        lda ($5f),y
b355  85 29        sta $29
b357  a9 10        lda #$10
b359  85 5d        sta $5d
b35b  a2 00        ldx #$00
b35d  a0 00        ldy #$00
b35f  8a           txa
b360  0a           asl
b361  aa           tax
b362  98           tya
b363  2a           rol
b364  a8           tay
b365  b0 a4        bcs $b30b
b367  06 71        asl $71
b369  26 72        rol $72
b36b  90 0b        bcc $b378
b36d  18           clc
b36e  8a           txa
```

```
b36f  65 28        adc $28
b371  aa           tax
b372  98           tya
b373  65 29        adc $29
b375  a8           tay
b376  b0 93        bcs $b30b
b378  c6 5d        dec $5d
b37a  d0 e3        bne $b35f
b37c  60           rts
```

```
FRE
b37d  a5 0d        lda $0d; retezce/cisla
b37f  f0 03        beq $b384
b381  20 a6 b6     jsr $b6a6
b384  20 26 b5     jsr $b526
b387  38           sec
b388  a5 33        lda $33; Lo Pocatek retezcu
b38a  e5 31        sbc $31; Lo Konec Poli
b38c  aa           tax
b38d  a5 34        lda $34; Hi Pocatek retezcu
b38f  e5 32        sbc $32; Hi Konec Poli
b391  a2 00        ldx #$00
b393  86 0d        stx $0d; typ $00-cisla
b395  85 62        sta $62
b397  84 63        sty $63
b399  a2 90        ldx #$90
b39b  4c 44 bc     jmp $bc44
```

```
POS
b39e  38           sec
b39f  20 f0 ff     jsr $fff0; PLOT (cteni)
b3a2  a9 00        lda #$00
b3a4  f0 eb        beq $b391
```

```
b3a6  a6 3a        ldx $3a; Hi Prekladany r.
b3a8  e8           inx
b3a9  d0 a0        bne $b34b; RTS
b3ab  a2 15        ldx #$15; Kod 'Illegal direct'
b3ad  2c a2 1b     bit $1ba2; ('Undef'd function')
b3b0  4c 37 a4     jmp $a437; chybova hlaska
```

```
DEF
b3b3  20 e1 b3     jsr $b3e1
b3b6  20 a6 b3     jsr $b3a6
b3b9  20 fa ae     jsr $aefa; test na '('
b3bc  a9 80        lda #$80
b3be  85 10        sta $10
b3c0  20 8b b0     jsr $b08b
b3c3  20 8d ad     jsr $ad8d
b3c6  20 f7 ae     jsr $aef7; test na ')'
b3c9  a9 b2        lda #$b2
b3cb  20 ff ae     jsr $aeff; test na '='
b3ce  48           pha
b3cf  a5 48        lda $48; Hi
b3d1  48           pha
b3d2  a5 47        lda $47; Lo adr. Promenne
b3d4  48           pha
b3d5  a5 7b        lda $7b; Hi
```

```
b3d7  48                 pha
b3d8  a5 7a              lda $7a; Lo adresa CHRGET
b3da  48                 pha
b3db  20 f8 a8           jsr $a8f8; rutina <DATA>
b3de  4c 4f b4           jmp $b44f
-------------------------------------------------------
b3e1  a9 a5              lda #$a5
b3e3  20 ff ae           jsr $aeff; test na 'FN'
b3e6  09 80              ora #$80
b3e8  85 10              sta $10
b3ea  20 92 b0           jsr $b092
b3ed  85 4e              sta $4e
b3ef  84 4f              sty $4f
b3f1  4c 8d ad           jmp $ad8d
-------------------------------------------------------
b3f4  20 e1 b3           jsr $b3e1; test na 'FN'+ostatni
b3f7  a5 4f              lda $4f
b3f9  48                 pha
b3fa  a5 4e              lda $4e
b3fc  48                 pha
b3fd  20 f1 ae           jsr $aef1
b400  20 8d ad           jsr $ad8d
b403  68                 pla
b404  85 4e              sta $4e
b406  68                 pla
b407  85 4f              sta $4f
b409  a0 02              ldy #$02
b40b  b1 4e              lda ($4e),y
b40d  85 47              sta $47
b40f  aa                 tax
b410  c8                 iny
b411  b1 4e              lda ($4e),y
b413  f0 99              beq $b3ae
b415  85 48              sta $48
b417  c8                 iny
b418  b1 47              lda ($47),y
b41a  48                 pha
b41b  88                 dey
b41c  10 fa              bpl $b418
b41e  a4 48              ldy $48
b420  20 d4 bb           jsr $bbd4
b423  a5 7b              lda $7b
b425  48                 pha
b426  a5 7a              lda $7a
b428  48                 pha
b429  b1 4e              lda ($4e),y
b42b  85 7a              sta $7a
b42d  c8                 iny
b42e  b1 4e              lda ($4e),y
b430  85 7b              sta $7b
b432  a5 48              lda $48
b434  48                 pha
b435  a5 47              lda $47
b437  48                 pha
b438  20 8a ad           jsr $ad8a
b43b  68                 pla
b43c  85 4e              sta $4e
b43e  68                 pla
```

```
b43f  85 4f        sta $4f
b441  20 79 00     jsr $0079
b444  f0 03        beq $b449
b446  4c 08 af     jmp $af08
b449  68           pla
b44a  85 7a        sta $7a
b44c  68           pla
b44d  85 7b        sta $7b
b44f  a0 00        ldy #$00
b451  68           pla
b452  91 4e        sta ($4e),y
b454  68           pla
b455  c8           iny
b456  91 4e        sta ($4e),y
b458  68           pla
b459  c8           iny
b45a  91 4e        sta ($4e),y
b45c  68           pla
b45d  c8           iny
b45e  91 4e        sta ($4e),y
b460  68           pla
b461  c8           iny
b462  91 4e        sta ($4e),y
b464  60           rts
-------------------------------------------------------
STR$
b465  20 8d ad     jsr $ad8d
b468  a0 00        ldy #$00
b46a  20 df bd     jsr $bddf
b46d  68           pla
b46e  68           pla
b46f  a9 ff        lda #$ff
b471  a0 00        ldy #$00
b473  f0 12        beq $b487
-------------------------------------------------------
b475  a6 64        ldx $64
b477  a4 65        ldy $65
b479  86 50        stx $50
b47b  84 51        sty $51
b47d  20 f4 b4     jsr $b4f4
b480  86 62        stx $62
b482  84 63        sty $63
b484  85 61        sta $61
b486  60           rts
-------------------------------------------------------
b487  a2 22        ldx #$22
b489  86 07        stx $07
b48b  86 08        stx $08
b48d  85 6f        sta $6f
b48f  84 70        sty $70
b491  85 62        sta $62
b493  84 63        sty $63
b495  a0 ff        ldy #$ff
b497  c8           iny
b498  b1 6f        lda ($6f),y
b49a  f0 0c        beq $b4a8
b49c  c5 07        cmp $07
b49e  f0 04        beq $b4a4
```

```
b4a0  c5 08     cmp  $08
b4a2  d0 f3     bne  $b497
b4a4  c9 22     cmp  #$22
b4a6  f0 01     beq  $b4a9
b4a8  18        clc
b4a9  84 61     sty  $61
b4ab  98        tya
b4ac  65 6f     adc  $6f
b4ae  85 71     sta  $71
b4b0  a6 70     ldx  $70
b4b2  90 01     bcc  $b4b5
b4b4  e8        inx
b4b5  86 72     stx  $72
b4b7  a5 70     lda  $70
b4b9  f0 04     beq  $b4bf
b4bb  c9 02     cmp  #$02
b4bd  d0 0b     bne  $b4ca
b4bf  98        tya
b4c0  20 75 b4  jsr  $b475
b4c3  a6 6f     ldx  $6f
b4c5  a4 70     ldy  $70
b4c7  20 88 b6  jsr  $b688
b4ca  a6 16     ldx  $16
b4cc  e0 22     cpx  #$22
b4ce  d0 05     bne  $b4d5
b4d0  a2 19     ldx  #$19
b4d2  4c 37 a4  jmp  $a437; 'Formula too complex'
```

```
b4d5  a5 61     lda  $61
b4d7  95 00     sta  $00,x
b4d9  a5 62     lda  $62
b4db  95 01     sta  $01,x
b4dd  a5 63     lda  $63
b4df  95 02     sta  $02,x
b4e1  a0 00     ldy  #$00
b4e3  86 64     stx  $64
b4e5  84 65     sty  $65
b4e7  84 70     sty  $70
b4e9  88        dey
b4ea  84 0d     sty  $0d
b4ec  86 17     stx  $17
b4ee  e8        inx
b4ef  e8        inx
b4f0  e8        inx
b4f1  86 16     stx  $16
b4f3  60        rts
```

```
b4f4  46 0f     lsr  $0f
b4f6  48        pha
b4f7  49 ff     eor  #$ff
b4f9  38        sec
b4fa  65 33     adc  $33
b4fc  a4 34     ldy  $34
b4fe  b0 01     bcs  $b501
b500  88        dey
b501  c4 32     cpy  $32
b503  90 11     bcc  $b516
b505  d0 04     bne  $b50b
```

```
b507  c5 31        cmp  $31
b509  90 0b        bcc  $b516
b50b  85 33        sta  $33
b50d  84 34        sty  $34
b50f  85 35        sta  $35
b511  84 36        sty  $36
b513  aa           tax
b514  68           pla
b515  60           rts
-------------------------------------------------------------------
b516  a2 10        ldx  #$10
b518  a5 0f        lda  $0f
b51a  30 b6        bmi  $b4d2
b51c  20 26 b5     jsr  $b526
b51f  a9 80        lda  #$80
b521  85 0f        sta  $0f
b523  68           pla
b524  d0 d0        bne  $b4f6
b526  a6 37        ldx  $37; Lo
b528  a5 38        lda  $38; Hi Konec Basic RAM
b52a  86 33        stx  $33; Lo
b52c  85 34        sta  $34; Hi Poc. retezcu
b52e  a0 00        ldy  #$00
b530  84 4f        sty  $4f
b532  84 4e        sty  $4e
b534  a5 31        lda  $31; Lo
b536  a6 32        ldx  $32; Hi Konec Poli
b538  85 5f        sta  $5f
b53a  86 60        stx  $60
b53c  a9 19        lda  #$19
b53e  a2 00        ldx  #$00
b540  85 22        sta  $22
b542  86 23        stx  $23
b544  c5 16        cmp  $16
b546  f0 05        beq  $b54d
b548  20 c7 b5     jsr  $b5c7
b54b  f0 f7        beq  $b544
b54d  a9 07        lda  #$07
b54f  85 53        sta  $53
b551  a5 2d        lda  $2d
b553  a6 2e        ldx  $2e
b555  85 22        sta  $22
b557  86 23        stx  $23
b559  e4 30        cpx  $30
b55b  d0 04        bne  $b561
b55d  c5 2f        cmp  $2f
b55f  f0 05        beq  $b566
b561  20 bd b5     jsr  $b5bd
b564  f0 f3        beq  $b559
b566  85 58        sta  $58
b568  86 59        stx  $59
b56a  a9 03        lda  #$03
b56c  85 53        sta  $53
b56e  a5 58        lda  $58
b570  a6 59        ldx  $59
b572  e4 32        cpx  $32; Hi
b574  d0 07        bne  $b57d
b576  c5 31        cmp  $31; Lo Konec Poli
```

```
b578 d0 03        bne $b57d
b57a 4c 06 b6     jmp $b606
--------------------------------------------------
b57d 85 22        sta $22
b57f 86 23        stx $23
b581 a0 00        ldy #$00
b583 b1 22        lda ($22),y
b585 aa           tax
b586 c8           iny
b587 b1 22        lda ($22),y
b589 08           php
b58a c8           iny
b58b b1 22        lda ($22),y
b58d 65 58        adc $58
b58f 85 58        sta $58
b591 c8           iny
b592 b1 22        lda ($22),y
b594 65 59        adc $59
b596 85 59        sta $59
b598 28           plp
b599 10 d3        bpl $b56e
b59b 8a           txa
b59c 30 d0        bmi $b56e
b59e c8           iny
b59f b1 22        lda ($22),y
b5a1 a0 00        ldy #$00
b5a3 0a           asl
b5a4 69 05        adc #$05
b5a6 65 22        adc $22
b5a8 85 22        sta $22
b5aa 90 02        bcc $b5ae
b5ac e6 23        inc $23
b5ae a6 23        ldx $23
b5b0 e4 59        cpx $59
b5b2 d0 04        bne $b5b8
b5b4 c5 58        cmp $58
b5b6 f0 ba        beq $b572
b5b8 20 c7 b5     jsr $b5c7
b5bb f0 f3        beq $b5b0
b5bd b1 22        lda ($22),y
b5bf 30 35        bmi $b5f6
b5c1 c8           iny
b5c2 b1 22        lda ($22),y
b5c4 10 30        bpl $b5f6
b5c6 c8           iny
b5c7 b1 22        lda ($22),y
b5c9 f0 2b        beq $b5f6
b5cb c8           iny
b5cc b1 22        lda ($22),y
b5ce aa           tax
b5cf c8           iny
b5d0 b1 22        lda ($22),y
b5d2 c5 34        cmp $34
b5d4 90 06        bcc $b5dc
b5d6 d0 1e        bne $b5f6
b5d8 e4 33        cpx $33
b5da b0 1a        bcs $b5f6
b5dc c5 60        cmp $60
```

```
b5de  90 16     bcc $b5f6
b5e0  d0 04     bne $b5e6
b5e2  e4 5f     cpx $5f
b5e4  90 10     bcc $b5f6
b5e6  86 5f     stx $5f
b5e8  85 60     sta $60
b5ea  a5 22     lda $22
b5ec  a6 23     ldx $23
b5ee  85 4e     sta $4e
b5f0  86 4f     stx $4f
b5f2  a5 53     lda $53
b5f4  85 55     sta $55
b5f6  a5 53     lda $53
b5f8  18        clc
b5f9  65 22     adc $22
b5fb  85 22     sta $22
b5fd  90 02     bcc $b601
b5ff  e6 23     inc $23
b601  a6 23     ldx $23
b603  a0 00     ldy #$00
b605  60        rts
-------------------------------------------------------------------
b606  a5 4f     lda $4f
b608  05 4e     ora $4e
b60a  f0 f5     beq $b601
b60c  a5 55     lda $55
b60e  29 04     and #$04
b610  4a        lsr
b611  a8        tay
b612  85 55     sta $55
b614  b1 4e     lda ($4e),y
b616  65 5f     adc $5f
b618  85 5a     sta $5a
b61a  a5 60     lda $60
b61c  69 00     adc #$00
b61e  85 5b     sta $5b
b620  a5 33     lda $33; Lo
b622  a6 34     ldx $34; Hi poc. retezcu
b624  85 58     sta $58
b626  86 59     stx $59
b628  20 bf a3  jsr $a3bf
b62b  a4 55     ldy $55
b62d  c8        iny
b62e  a5 58     lda $58
b630  91 4e     sta ($4e),y
b632  aa        tax
b633  e6 59     inc $59
b635  a5 59     lda $59
b637  c8        iny
b638  91 4e     sta ($4e),y
b63a  4c 2a b5  jmp $b52a
-------------------------------------------------------------------
b63d  a5 65     lda $65
b63f  48        pha
b640  a5 64     lda $64
b642  48        pha
b643  20 83 ae  jsr $ae83
b646  20 8f ad  jsr $ad8f
```

```
b649 68          pla
b64a 85 6f       sta $6f
b64c 68          pla
b64d 85 70       sta $70
b64f a0 00       ldy #$00
b651 b1 6f       lda ($6f),y
b653 18          clc
b654 71 64       adc ($64),y
b656 90 05       bcc $b65d
b658 a2 17       ldx #$17
b65a 4c 37 a4    jmp $a437; 'String too long'
-------------------------------------------------
b65d 20 75 b4    jsr $b475
b660 20 7a b6    jsr $b67a
b663 a5 50       lda $50
b665 a4 51       ldy $51
b667 20 aa b6    jsr $b6aa
b66a 20 8c b6    jsr $b68c
b66d a5 6f       lda $6f
b66f a4 70       ldy $70
b671 20 aa b6    jsr $b6aa
b674 20 ca b4    jsr $b4ca
b677 4c b8 ad    jmp $adb8
-------------------------------------------------
b67a a0 00       ldy #$00
b67c b1 6f       lda ($6f),y
b67e 48          pha
b67f c8          iny
b680 b1 6f       lda ($6f),y
b682 aa          tax
b683 c8          iny
b684 b1 6f       lda ($6f),y
b686 a8          tay
b687 68          pla
b688 86 22       stx $22
b68a 84 23       sty $23
b68c a8          tay
b68d f0 0a       beq $b699
b68f 48          pha
b690 88          dey
b691 b1 22       lda ($22),y
b693 91 35       sta ($35),y
b695 98          tya
b696 d0 f8       bne $b690
b698 68          pla
b699 18          clc
b69a 65 35       adc $35
b69c 85 35       sta $35; Lo Pomocny ukaza-
b69e 90 02       bcc $b6a2
b6a0 e6 36       inc $36; Hi tel Pro retezce
b6a2 60          rts
-------------------------------------------------
b6a3 20 8f ad    jsr $ad8f
b6a6 a5 64       lda $64
b6a8 a4 65       ldy $65
b6aa 85 22       sta $22
b6ac 84 23       sty $23
b6ae 20 db b6    jsr $b6db
```

```
b6b1  08           php
b6b2  a0 00        ldy #$00
b6b4  b1 22        lda ($22),y
b6b6  48           pha
b6b7  c8           iny
b6b8  b1 22        lda ($22),y
b6ba  aa           tax
b6bb  c8           iny
b6bc  b1 22        lda ($22),y
b6be  a8           tay
b6bf  68           pla
b6c0  28           plp
b6c1  d0 13        bne $b6d6
b6c3  c4 34        cpy $34
b6c5  d0 0f        bne $b6d6
b6c7  e4 33        cpx $33
b6c9  d0 0b        bne $b6d6
b6cb  48           pha
b6cc  18           clc
b6cd  65 33        adc $33
b6cf  85 33        sta $33; Lo
b6d1  90 02        bcc $b6d5
b6d3  e6 34        inc $34; Hi Poc. retezcu
b6d5  68           pla
b6d6  86 22        stx $22
b6d8  84 23        sty $23
b6da  60           rts
-------------------------------------------------------------
b6db  c4 18        cpy $18
b6dd  d0 0c        bne $b6eb
b6df  c5 17        cmp $17
b6e1  d0 08        bne $b6eb
b6e3  85 16        sta $16
b6e5  e9 03        sbc #$03
b6e7  85 17        sta $17
b6e9  a0 00        ldy #$00
b6eb  60           rts
-------------------------------------------------------------
CHR$
b6ec  20 a1 b7     jsr $b7a1
b6ef  8a           txa
b6f0  48           pha
b6f1  a9 01        lda #$01
b6f3  20 7d b4     jsr $b47d
b6f6  68           pla
b6f7  a0 00        ldy #$00
b6f9  91 62        sta ($62),y
b6fb  68           pla
b6fc  68           pla
b6fd  4c ca b4     jmp $b4ca
-------------------------------------------------------------
LEFT$
b700  20 61 b7     jsr $b761
b703  d1 50        cmp ($50),y
b705  98           tya
b706  90 04        bcc $b70c
b708  b1 50        lda ($50),y
b70a  aa           tax
```

```
b70b  98          tya
b70c  48          pha
b70d  8a          txa
b70e  48          pha
b70f  20 7d b4    jsr $b47d
b712  a5 50       lda $50
b714  a4 51       ldy $51
b716  20 aa b6    jsr $b6aa
b719  68          pla
b71a  a8          tay
b71b  68          pla
b71c  18          clc
b71d  65 22       adc $22
b71f  85 22       sta $22
b721  90 02       bcc $b725
b723  e6 23       inc $23
b725  98          tya
b726  20 8c b6    jsr $b68c
b729  4c ca b4    jmp $b4ca
```

```
RIGHT$
b72c  20 61 b7    jsr $b761
b72f  18          clc
b730  f1 50       sbc ($50),y
b732  49 ff       eor #$ff
b734  4c 06 b7    jmp $b706
```

```
MID$
b737  a9 ff       lda #$ff
b739  85 65       sta $65
b73b  20 79 00    jsr $0079
b73e  c9 29       cmp #$29
b740  f0 06       beq $b748
b742  20 fd ae    jsr $aefd
b745  20 9e b7    jsr $b79e
b748  20 61 b7    jsr $b761
b74b  f0 4b       beq $b798
b74d  ca          dex
b74e  8a          txa
b74f  48          pha
b750  18          clc
b751  a2 00       ldx #$00
b753  f1 50       sbc ($50),y
b755  b0 b6       bcs $b70d
b757  49 ff       eor #$ff
b759  c5 65       cmp $65
b75b  90 b1       bcc $b70e
b75d  a5 65       lda $65
b75f  b0 ad       bcs $b70e
b761  20 f7 ae    jsr $aef7
b764  68          pla
b765  a8          tay
b766  68          pla
b767  85 55       sta $55
b769  68          pla
b76a  68          pla
b76b  68          pla
b76c  aa          tax
```

```
b76d  68          pla
b76e  85 50       sta  $50
b770  68          pla
b771  85 51       sta  $51
b773  a5 55       lda  $55
b775  48          pha
b776  98          tya
b777  48          pha
b778  a0 00       ldy  #$00
b77a  8a          txa
b77b  60          rts
```

LEN

```
b77c  20 82 b7    jsr  $b782
b77f  4c a2 b3    jmp  $b3a2
```

```
b782  20 a3 b6    jsr  $b6a3
b785  a2 00       ldx  #$00
b787  86 0d       stx  $0d
b789  a8          tay
b78a  60          rts
```

ASC

```
b78b  20 82 b7    jsr  $b782
b78e  f0 08       beq  $b798
b790  a0 00       ldy  #$00
b792  b1 22       lda  ($22),y
b794  a8          tay
b795  4c a2 b3    jmp  $b3a2
b798  4c 48 b2    jmp  $b248
```

```
b79b  20 73 00    jsr  $0073;  CHRGET
b79e  20 8a ad    jsr  $ad8a
b7a1  20 b8 b1    jsr  $b1b8
b7a4  a6 64       ldx  $64
b7a6  d0 f0       bne  $b798
b7a8  a6 65       ldx  $65
b7aa  4c 79 00    jmp  $0079
```

VAL

```
b7ad  20 82 b7    jsr  $b782
b7b0  d0 03       bne  $b7b5
b7b2  4c f7 b8    jmp  $b8f7
```

```
b7b5  a6 7a       ldx  $7a
b7b7  a4 7b       ldy  $7b
b7b9  86 71       stx  $71
b7bb  84 72       sty  $72
b7bd  a6 22       ldx  $22
b7bf  86 7a       stx  $7a
b7c1  18          clc
b7c2  65 22       adc  $22
b7c4  85 24       sta  $24
b7c6  a6 23       ldx  $23
b7c8  86 7b       stx  $7b
b7ca  90 01       bcc  $b7cd
b7cc  e8          inx
b7cd  86 25       stx  $25
```

```
b7cf  a0 00        ldy #$00
b7d1  b1 24        lda ($24),y
b7d3  48           pha
b7d4  98           tya
b7d5  91 24        sta ($24),y
b7d7  20 79 00     jsr $0079; CHRGET bez posunuti
b7da  20 f3 bc     jsr $bcf3
b7dd  68           pla
b7de  a0 00        ldy #$00
b7e0  91 24        sta ($24),y
b7e2  a6 71        ldx $71
b7e4  a4 72        ldy $72
b7e6  86 7a        stx $7a; Lo
b7e8  84 7b        sty $7b; Hi CHRGET
b7ea  60           rts
-------------------------------------------
b7eb  20 8a ad     jsr $ad8a
b7ee  20 f7 b7     jsr $b7f7
b7f1  20 fd ae     jsr $aefd
b7f4  4c 9e b7     jmp $b79e
-------------------------------------------
b7f7  a5 66        lda $66
b7f9  30 9d        bmi $b798
b7fb  a5 61        lda $61
b7fd  c9 91        cmp #$91
b7ff  b0 97        bcs $b798
b801  20 9b bc     jsr $bc9b
b804  a5 64        lda $64
b806  a4 65        ldy $65
b808  84 14        sty $14; Lo radka pri GOTO
b80a  85 15        sta $15; Hi GOSUB;LIST aPod.
b80c  60           rts
-------------------------------------------
PEEK
b80d  a5 15        lda $15
b80f  48           pha
b810  a5 14        lda $14
b812  48           pha
b813  20 f7 b7     jsr $b7f7
b816  a0 00        ldy #$00
b818  b1 14        lda ($14),y
b81a  a8           tay
b81b  68           pla
b81c  85 14        sta $14
b81e  68           pla
b81f  85 15        sta $15
b821  4c a2 b3     jmp $b3a2
-------------------------------------------
POKE
b824  20 eb b7     jsr $b7eb
b827  8a           txa
b828  a0 00        ldy #$00
b82a  91 14        sta ($14),y
b82c  60           rts
-------------------------------------------
WAIT
b82d  20 eb b7     jsr $b7eb
b830  86 49        stx $49
```

```
b832 a2 00        ldx #$00
b834 20 79 00     jsr $0079; CHRGET bez posunuti
b837 f0 03        beq $b83c
b839 20 f1 b7     jsr $b7f1
b83c 86 4a        stx $4a
b83e a0 00        ldy #$00
b840 b1 14        lda ($14),y
b842 45 4a        eor $4a
b844 25 49        and $49
b846 f0 f8        beq $b840
b848 60           rts
-------------------------------------------------
b849 a9 11        lda #$11; Lo
b84b a0 bf        ldy #$bf; Hi adresa dat
b84d 4c 67 b8     jmp $b867
-------------------------------------------------
b850 20 8c ba     jsr $ba8c
b853 a5 66        lda $66
b855 49 ff        eor #$ff
b857 85 66        sta $66
b859 45 6e        eor $6e
b85b 85 6f        sta $6f
b85d a5 61        lda $61
b85f 4c 6a b8     jmp $b86a
-------------------------------------------------
b862 20 99 b9     jsr $b999
b865 90 3c        bcc $b8a3
b867 20 8c ba     jsr $ba8c
b86a d0 03        bne $b86f
b86c 4c fc bb     jmp $bbfc
-------------------------------------------------
b86f a6 70        ldx $70
b871 86 56        stx $56
b873 a2 69        ldx #$69
b875 a5 69        lda $69
b877 a8           tay
b878 f0 ce        beq $b848
b87a 38           sec
b87b e5 61        sbc $61
b87d f0 24        beq $b8a3
b87f 90 12        bcc $b893
b881 84 61        sty $61
b883 a4 6e        ldy $6e
b885 84 66        sty $66
b887 49 ff        eor #$ff
b889 69 00        adc #$00
b88b a0 00        ldy #$00
b88d 84 56        sty $56
b88f a2 61        ldx #$61
b891 d0 04        bne $b897
b893 a0 00        ldy #$00
b895 84 70        sty $70
b897 c9 f9        cmp #$f9
b899 30 c7        bmi $b862
b89b a8           tay
b89c a5 70        lda $70
b89e 56 01        lsr $01,x
b8a0 20 b0 b9     jsr $b9b0
```

```
b8a3  24 6f        bit $6f
b8a5  10 57        bpl $b8fe
b8a7  a0 61        ldy #$61
b8a9  e0 69        cpx #$69
b8ab  f0 02        beq $b8af
b8ad  a0 69        ldy #$69
b8af  38           sec
b8b0  49 ff        eor #$ff
b8b2  65 56        adc $56
b8b4  85 70        sta $70
b8b6  b9 04 00     lda $0004,y
b8b9  f5 04        sbc $04,x
b8bb  85 65        sta $65
b8bd  b9 03 00     lda $0003,y
b8c0  f5 03        sbc $03,x
b8c2  85 64        sta $64
b8c4  b9 02 00     lda $0002,y
b8c7  f5 02        sbc $02,x
b8c9  85 63        sta $63
b8cb  b9 01 00     lda $0001,y
b8ce  f5 01        sbc $01,x
b8d0  85 62        sta $62
b8d2  b0 03        bcs $b8d7
b8d4  20 47 b9     jsr $b947
b8d7  a0 00        ldy #$00
b8d9  98           tya
b8da  18           clc
b8db  a6 62        ldx $62
b8dd  d0 4a        bne $b929
b8df  a6 63        ldx $63
b8e1  86 62        stx $62
b8e3  a6 64        ldx $64
b8e5  86 63        stx $63
b8e7  a6 65        ldx $65
b8e9  86 64        stx $64
b8eb  a6 70        ldx $70
b8ed  86 65        stx $65
b8ef  84 70        sty $70
b8f1  69 08        adc #$08
b8f3  c9 20        cmp #$20
b8f5  d0 e4        bne $b8db
b8f7  a9 00        lda #$00
b8f9  85 61        sta $61
b8fb  85 66        sta $66
b8fd  60           rts
-------------------------------------------------------
b8fe  65 56        adc $56
b900  85 70        sta $70
b902  a5 65        lda $65
b904  65 6d        adc $6d
b906  85 65        sta $65
b908  a5 64        lda $64
b90a  65 6c        adc $6c
b90c  85 64        sta $64
b90e  a5 63        lda $63
b910  65 6b        adc $6b
b912  85 63        sta $63
b914  a5 62        lda $62
```

```
b916 65 6a        adc $6a
b918 85 62        sta $62
b91a 4c 36 b9     jmp $b936
--------------------------------------------------------
b91d 69 01        adc #$01
b91f 06 70        asl $70
b921 26 65        rol $65
b923 26 64        rol $64
b925 26 63        rol $63
b927 26 62        rol $62
b929 10 f2        bpl $b91d
b92b 38           sec
b92c e5 61        sbc $61
b92e b0 c7        bcs $b8f7
b930 49 ff        eor #$ff
b932 69 01        adc #$01
b934 85 61        sta $61
b936 90 0e        bcc $b946
b938 e6 61        inc $61
b93a f0 42        beq $b97e
b93c 66 62        ror $62
b93e 66 63        ror $63
b940 66 64        ror $64
b942 66 65        ror $65
b944 66 70        ror $70
b946 60           rts
--------------------------------------------------------
b947 a5 66        lda $66
b949 49 ff        eor #$ff
b94b 85 66        sta $66
b94d a5 62        lda $62
b94f 49 ff        eor #$ff
b951 85 62        sta $62
b953 a5 63        lda $63
b955 49 ff        eor #$ff
b957 85 63        sta $63
b959 a5 64        lda $64
b95b 49 ff        eor #$ff
b95d 85 64        sta $64
b95f a5 65        lda $65
b961 49 ff        eor #$ff
b963 85 65        sta $65
b965 a5 70        lda $70
b967 49 ff        eor #$ff
b969 85 70        sta $70
b96b e6 70        inc $70
b96d d0 0e        bne $b97d
b96f e6 65        inc $65
b971 d0 0a        bne $b97d
b973 e6 64        inc $64
b975 d0 06        bne $b97d
b977 e6 63        inc $63
b979 d0 02        bne $b97d
b97b e6 62        inc $62
b97d 60           rts
--------------------------------------------------------
b97e a2 0f        ldx #$0f
b980 4c 37 a4     jmp $a437; 'Overflow'
```

```
b983  a2 25        ldx #$25
b985  b4 04        ldy $04,x
b987  84 70        sty $70
b989  b4 03        ldy $03,x
b98b  94 04        sty $04,x
b98d  b4 02        ldy $02,x
b98f  94 03        sty $03,x
b991  b4 01        ldy $01,x
b993  94 02        sty $02,x
b995  a4 68        ldy $68
b997  94 01        sty $01,x
b999  69 08        adc #$08
b99b  30 e8        bmi $b985
b99d  f0 e6        beq $b985
b99f  e9 08        sbc #$08
b9a1  a8           tay
b9a2  a5 70        lda $70
b9a4  b0 14        bcs $b9ba
b9a6  16 01        asl $01,x
b9a8  90 02        bcc $b9ac
b9aa  f6 01        inc $01,x
b9ac  76 01        ror $01,x
b9ae  76 01        ror $01,x
b9b0  76 02        ror $02,x
b9b2  76 03        ror $03,x
b9b4  76 04        ror $04,x
b9b6  6a           ror
b9b7  c8           iny
b9b8  d0 ec        bne $b9a6
b9ba  18           clc
b9bb  60           rts
```

```
b9bc  81 00 00 00 00 03 7f 5e; Data
b9c4  56 cb 79 80 13 9b 0b 64
b9cc  80 76 38 93 16 82 38 aa
b9d4  3b 20 80 35 04 f3 34 81
b9dc  35 04 f3 34 80 80 00 00
b9e4  00 80 31 72 17 f8
```

```
LOG
b9ea  20 2b bc     jsr $bc2b
b9ed  f0 02        beq $b9f1
b9ef  10 03        bpl $b9f4
b9f1  4c 48 b2     jmp $b248
```

```
b9f4  a5 61        lda $61
b9f6  e9 7f        sbc #$7f
b9f8  48           pha
b9f9  a9 80        lda #$80
b9fb  85 61        sta $61
b9fd  a9 d6        lda #$d6
b9ff  a0 b9        ldy #$b9
ba01  20 67 b8     jsr $b867
ba04  a9 db        lda #$db
ba06  a0 b9        ldy #$b9

200
```

```
ba08  20 0f bb     jsr $bb0f
ba0b  a9 bc        lda #$bc
ba0d  a0 b9        ldy #$b9
ba0f  20 50 b8     jsr $b850
ba12  a9 c1        lda #$c1
ba14  a0 b9        ldy #$b9
ba16  20 43 e0     jsr $e043
ba19  a9 e0        lda #$e0
ba1b  a0 b9        ldy #$b9
ba1d  20 67 b8     jsr $b867
ba20  68           pla
ba21  20 7e bd     jsr $bd7e
ba24  a9 e5        lda #$e5
ba26  a0 b9        ldy #$b9
ba28  20 8c ba     jsr $ba8c
ba2b  d0 03        bne $ba30
ba2d  4c 8b ba     jmp $ba8b
-------------------------------------------------------
ba30  20 b7 ba     jsr $bab7
ba33  a9 00        lda #$00
ba35  85 26        sta $26
ba37  85 27        sta $27
ba39  85 28        sta $28
ba3b  85 29        sta $29
ba3d  a5 70        lda $70
ba3f  20 59 ba     jsr $ba59
ba42  a5 65        lda $65
ba44  20 59 ba     jsr $ba59
ba47  a5 64        lda $64
ba49  20 59 ba     jsr $ba59
ba4c  a5 63        lda $63
ba4e  20 59 ba     jsr $ba59
ba51  a5 62        lda $62
ba53  20 5e ba     jsr $ba5e
ba56  4c 8f bb     jmp $bb8f
-------------------------------------------------------
ba59  d0 03        bne $ba5e
ba5b  4c 83 b9     jmp $b983
-------------------------------------------------------
ba5e  4a           lsr
ba5f  09 80        ora #$80
ba61  a8           tay
ba62  90 19        bcc $ba7d
ba64  18           clc
ba65  a5 29        lda $29
ba67  65 6d        adc $6d
ba69  85 29        sta $29
ba6b  a5 28        lda $28
ba6d  65 6c        adc $6c
ba6f  85 28        sta $28
ba71  a5 27        lda $27
ba73  65 6b        adc $6b
ba75  85 27        sta $27
ba77  a5 26        lda $26
ba79  65 6a        adc $6a
ba7b  85 26        sta $26
ba7d  66 26        ror $26
ba7f  66 27        ror $27
```

```
ba81  66 28        ror $28
ba83  66 29        ror $29
ba85  66 70        ror $70
ba87  98           tya
ba88  4a           lsr
ba89  d0 d6        bne $ba61
ba8b  60           rts
--------------------------------------------------------
ba8c  85 22        sta $22
ba8e  84 23        sty $23
ba90  a0 04        ldy #$04
ba92  b1 22        lda ($22),y
ba94  85 6d        sta $6d
ba96  88           dey
ba97  b1 22        lda ($22),y
ba99  85 6c        sta $6c
ba9b  88           dey
ba9c  b1 22        lda ($22),y
ba9e  85 6b        sta $6b
baa0  88           dey
baa1  b1 22        lda ($22),y
baa3  85 6e        sta $6e
baa5  45 66        eor $66
baa7  85 6f        sta $6f
baa9  a5 6e        lda $6e
baab  09 80        ora #$80
baad  85 6a        sta $6a
baaf  88           dey
bab0  b1 22        lda ($22),y
bab2  85 69        sta $69
bab4  a5 61        lda $61
bab6  60           rts
--------------------------------------------------------
bab7  a5 69        lda $69
bab9  f0 1f        beq $bada
babb  18           clc
babc  65 61        adc $61
babe  90 04        bcc $bac4
bac0  30 1d        bmi $badf
bac2  18           clc
bac3  2c 10 14     bit $1410
bac6  69 80        adc #$80
bac8  85 61        sta $61
baca  d0 03        bne $bacf
bacc  4c fb b8     jmp $b8fb
--------------------------------------------------------
bacf  a5 6f        lda $6f
bad1  85 66        sta $66
bad3  60           rts
--------------------------------------------------------
bad4  a5 66        lda $66
bad6  49 ff        eor #$ff
bad8  30 05        bmi $badf
bada  68           pla
badb  68           pla
badc  4c f7 b8     jmp $b8f7
--------------------------------------------------------
badf  4c 7e b9     jmp $b97e
```

```
baе2  20 0c bc     jsr $bc0c
baе5  aa           tax
baе6  f0 10        beq $baf8
baе8  18           clc
baе9  69 02        adc #$02
baeb  b0 f2        bcs $badf
baed  a2 00        ldx #$00
baef  86 6f        stx $6f
baf1  20 77 b8     jsr $b877
baf4  e6 61        inc $61
baf6  f0 e7        beq $badf
baf8  60           rts
```

```
baf9  84 20        sty $20
bafb  00           brk
bafc  00           brk
bafd  00           brk
```

```
bafe  20 0c bc     jsr $bc0c
bb01  a9 f9        lda #$f9
bb03  a0 ba        ldy #$ba
bb05  a2 00        ldx #$00
bb07  86 6f        stx $6f
bb09  20 a2 bb     jsr $bba2
bb0c  4c 12 bb     jmp $bb12
```

```
bb0f  20 8c ba     jsr $ba8c
/ (deleni)
bb12  f0 76        beq $bb8a
bb14  20 1b bc     jsr $bc1b
bb17  a9 00        lda #$00
bb19  38           sec
bb1a  e5 61        sbc $61
bb1c  85 61        sta $61
bb1e  20 b7 ba     jsr $bab7
bb21  e6 61        inc $61
bb23  f0 ba        beq $badf
bb25  a2 fc        ldx #$fc
bb27  a9 01        lda #$01
bb29  a4 6a        ldy $6a
bb2b  c4 62        cpy $62
bb2d  d0 10        bne $bb3f
bb2f  a4 6b        ldy $6b
bb31  c4 63        cpy $63
bb33  d0 0a        bne $bb3f
bb35  a4 6c        ldy $6c
bb37  c4 64        cpy $64
bb39  d0 04        bne $bb3f
bb3b  a4 6d        ldy $6d
bb3d  c4 65        cpy $65
bb3f  08           php
bb40  2a           rol
bb41  90 09        bcc $bb4c
bb43  e8           inx
bb44  95 29        sta $29,x
bb46  f0 32        beq $bb7a
bb48  10 34        bpl $bb7e
```

```
bb4a  a9 01     lda #$01
bb4c  28        plp
bb4d  b0 0e     bcs $bb5d
bb4f  06 6d     asl $6d
bb51  26 6c     rol $6c
bb53  26 6b     rol $6b
bb55  26 6a     rol $6a
bb57  b0 e6     bcs $bb3f
bb59  30 ce     bmi $bb29
bb5b  10 e2     bpl $bb3f
bb5d  a8        tay
bb5e  a5 6d     lda $6d
bb60  e5 65     sbc $65
bb62  85 6d     sta $6d
bb64  a5 6c     lda $6c
bb66  e5 64     sbc $64
bb68  85 6c     sta $6c
bb6a  a5 6b     lda $6b
bb6c  e5 63     sbc $63
bb6e  85 6b     sta $6b
bb70  a5 6a     lda $6a
bb72  e5 62     sbc $62
bb74  85 6a     sta $6a
bb76  98        tya
bb77  4c 4f bb  jmp $bb4f
--------------------------------------------------------------------------------
bb7a  a9 40     lda #$40
bb7c  d0 ce     bne $bb4c
bb7e  0a        asl
bb7f  0a        asl
bb80  0a        asl
bb81  0a        asl
bb82  0a        asl
bb83  0a        asl
bb84  85 70     sta $70
bb86  28        plp
bb87  4c 8f bb  jmp $bb8f
--------------------------------------------------------------------------------
bb8a  a2 14     ldx #$14
bb8c  4c 37 a4  jmp $a437; 'Division by zero'
--------------------------------------------------------------------------------
bb8f  a5 26     lda $26
bb91  85 62     sta $62
bb93  a5 27     lda $27
bb95  85 63     sta $63
bb97  a5 28     lda $28
bb99  85 64     sta $64
bb9b  a5 29     lda $29
bb9d  85 65     sta $65
bb9f  4c d7 b8  jmp $b8d7
--------------------------------------------------------------------------------
bba2  85 22     sta $22
bba4  84 23     sty $23
bba6  a0 04     ldy #$04
bba8  b1 22     lda ($22),y
bbaa  85 65     sta $65
bbac  88        dey
bbad  b1 22     lda ($22),y
```

```
bbaf  85 64     sta $64
bbb1  88        dey
bbb2  b1 22     lda ($22),y
bbb4  85 63     sta $63
bbb6  88        dey
bbb7  b1 22     lda ($22),y
bbb9  85 66     sta $66
bbbb  09 80     ora #$80
bbbd  85 62     sta $62
bbbf  88        dey
bbc0  b1 22     lda ($22),y
bbc2  85 61     sta $61
bbc4  84 70     sty $70
bbc6  60        rts
```

```
bbc7  a2 5c     ldx #$5c
bbc9  2c a2 57  bit $57a2
bbcc  a0 00     ldy #$00
bbce  f0 04     beq $bbd4
bbd0  a6 49     ldx $49
bbd2  a4 4a     ldy $4a
bbd4  20 1b bc  jsr $bc1b
bbd7  86 22     stx $22
bbd9  84 23     sty $23
bbdb  a0 04     ldy #$04
bbdd  a5 65     lda $65
bbdf  91 22     sta ($22),y
bbe1  88        dey
bbe2  a5 64     lda $64
bbe4  91 22     sta ($22),y
bbe6  88        dey
bbe7  a5 63     lda $63
bbe9  91 22     sta ($22),y
bbeb  88        dey
bbec  a5 66     lda $66
bbee  09 7f     ora #$7f
bbf0  25 62     and $62
bbf2  91 22     sta ($22),y
bbf4  88        dey
bbf5  a5 61     lda $61
bbf7  91 22     sta ($22),y
bbf9  84 70     sty $70
bbfb  60        rts
```

```
bbfc  a5 6e     lda $6e
bbfe  85 66     sta $66
bc00  a2 05     ldx #$05
bc02  b5 68     lda $68,x
bc04  95 60     sta $60,x
bc06  ca        dex
bc07  d0 f9     bne $bc02
bc09  86 70     stx $70
bc0b  60        rts
```

```
bc0c  20 1b bc  jsr $bc1b
bc0f  a2 06     ldx #$06
bc11  b5 60     lda $60,x
bc13  95 68     sta $68,x
```

```
bc15  ca        dex
bc16  d0 f9     bne $bc11
bc18  86 70     stx $70
bc1a  60        rts
-------------------------------------------------------
bc1b  a5 61     lda $61
bc1d  f0 fb     beq $bc1a
bc1f  06 70     asl $70
bc21  90 f7     bcc $bc1a
bc23  20 6f b9  jsr $b96f
bc26  d0 f2     bne $bc1a
bc28  4c 38 b9  jmp $b938
-------------------------------------------------------
bc2b  a5 61     lda $61
bc2d  f0 09     beq $bc38
bc2f  a5 66     lda $66
bc31  2a        rol
bc32  a9 ff     lda #$ff
bc34  b0 02     bcs $bc38
bc36  a9 01     lda #$01
bc38  60        rts
-------------------------------------------------------
SGN
bc39  20 2b bc  jsr $bc2b
bc3c  85 62     sta $62
bc3e  a9 00     lda #$00
bc40  85 63     sta $63
bc42  a2 88     ldx #$88
bc44  a5 62     lda $62
bc46  49 ff     eor #$ff
bc48  2a        rol
bc49  a9 00     lda #$00
bc4b  85 65     sta $65
bc4d  85 64     sta $64
bc4f  86 61     stx $61
bc51  85 70     sta $70
bc53  85 66     sta $66
bc55  4c d2 b8  jmp $b8d2
-------------------------------------------------------
ABS
bc58  46 66     lsr $66
bc5a  60        rts
-------------------------------------------------------
bc5b  85 24     sta $24
bc5d  84 25     sty $25
bc5f  a0 00     ldy #$00
bc61  b1 24     lda ($24),y
bc63  c8        iny
bc64  aa        tax
bc65  f0 c4     beq $bc2b
bc67  b1 24     lda ($24),y
bc69  45 66     eor $66
bc6b  30 c2     bmi $bc2f
bc6d  e4 61     cpx $61
bc6f  d0 21     bne $bc92
bc71  b1 24     lda ($24),y
bc73  09 80     ora #$80
bc75  c5 62     cmp $62
```

```
bc77  d0 19        bne  $bc92
bc79  c8           iny
bc7a  b1 24        lda  ($24),y
bc7c  c5 63        cmp  $63
bc7e  d0 12        bne  $bc92
bc80  c8           iny
bc81  b1 24        lda  ($24),y
bc83  c5 64        cmp  $64
bc85  d0 0b        bne  $bc92
bc87  c8           iny
bc88  a9 7f        lda  #$7f
bc8a  c5 70        cmp  $70
bc8c  b1 24        lda  ($24),y
bc8e  e5 65        sbc  $65
bc90  f0 28        beq  $bcba
bc92  a5 66        lda  $66
bc94  90 02        bcc  $bc98
bc96  49 ff        eor  #$ff
bc98  4c 31 bc     jmp  $bc31
-------------------------------------------------------
bc9b  a5 61        lda  $61
bc9d  f0 4a        beq  $bce9
bc9f  38           sec
bca0  e9 a0        sbc  #$a0
bca2  24 66        bit  $66
bca4  10 09        bpl  $bcaf
bca6  aa           tax
bca7  a9 ff        lda  #$ff
bca9  85 68        sta  $68
bcab  20 4d b9     jsr  $b94d
bcae  8a           txa
bcaf  a2 61        ldx  #$61
bcb1  c9 f9        cmp  #$f9
bcb3  10 06        bpl  $bcbb
bcb5  20 99 b9     jsr  $b999
bcb8  84 68        sty  $68
bcba  60           rts
-------------------------------------------------------
bcbb  a8           tay
bcbc  a5 66        lda  $66
bcbe  29 80        and  #$80
bcc0  46 62        lsr  $62
bcc2  05 62        ora  $62
bcc4  85 62        sta  $62
bcc6  20 b0 b9     jsr  $b9b0
bcc9  84 68        sty  $68
bccb  60           rts
-------------------------------------------------------
INT
bccc  a5 61        lda  $61
bcce  c9 a0        cmp  #$a0
bcd0  b0 20        bcs  $bcf2
bcd2  20 9b bc     jsr  $bc9b
bcd5  84 70        sty  $70
bcd7  a5 66        lda  $66
bcd9  84 66        sty  $66
bcdb  49 80        eor  #$80
bcdd  2a           rol
```

```
bcda  a9 a0          lda #$a0
bce0  85 61          sta $61
bce2  a5 65          lda $65
bce4  85 07          sta $07
bce6  4c d2 b8       jmp $b8d2
--------------------------------------------------------
bce9  85 62          sta $62
bceb  85 63          sta $63
bced  85 64          sta $64
bcef  85 65          sta $65
bcf1  a8             tay
bcf2  60             rts
--------------------------------------------------------
bcf3  a0 00          ldy #$00
bcf5  a2 0a          ldx #$0a
bcf7  94 5d          sty $5d,x
bcf9  ca             dex
bcfa  10 fb          bpl $bcf7
bcfc  90 0f          bcc $bd0d
bcfe  c9 2d          cmp #$2d; Kod '-'
bd00  d0 04          bne $bd06
bd02  86 67          stx $67
bd04  f0 04          beq $bd0a
bd06  c9 2b          cmp #$2b; Kod '+'
bd08  d0 05          bne $bd0f
bd0a  20 73 00       jsr $0073; CHRGET
bd0d  90 5b          bcc $bd6a
bd0f  c9 2e          cmp #$2e; Kod '.'
bd11  f0 2e          beq $bd41
bd13  c9 45          cmp #$45
bd15  d0 30          bne $bd47
bd17  20 73 00       jsr $0073; CHRGET
bd1a  90 17          bcc $bd33
bd1c  c9 ab          cmp #$ab
bd1e  f0 0e          beq $bd2e
bd20  c9 2d          cmp #$2d
bd22  f0 0a          beq $bd2e
bd24  c9 aa          cmp #$aa
bd26  f0 08          beq $bd30
bd28  c9 2b          cmp #$2b
bd2a  f0 04          beq $bd30
bd2c  d0 07          bne $bd35
bd2e  66 60          ror $60
bd30  20 73 00       jsr $0073; CHRGET
bd33  90 5c          bcc $bd91
bd35  24 60          bit $60
bd37  10 0e          bpl $bd47
bd39  a9 00          lda #$00
bd3b  38             sec
bd3c  e5 5e          sbc $5e
bd3e  4c 49 bd       jmp $bd49
--------------------------------------------------------
bd41  66 5f          ror $5f
bd43  24 5f          bit $5f
bd45  50 c3          bvc $bd0a
bd47  a5 5e          lda $5e
bd49  38             sec
bd4a  e5 5d          sbc $5d
```

```
bd4c  85 5e        sta  $5e
bd4e  f0 12        beq  $bd62
bd50  10 09        bpl  $bd5b
bd52  20 fe ba     jsr  $bafe
bd55  e6 5e        inc  $5e
bd57  d0 f9        bne  $bd52
bd59  f0 07        beq  $bd62
bd5b  20 e2 ba     jsr  $bae2
bd5e  c6 5e        dec  $5e
bd60  d0 f9        bne  $bd5b
bd62  a5 67        lda  $67
bd64  30 01        bmi  $bd67
bd66  60           rts
bd67  4c b4 bf     jmp  $bfb4
```

```
bd6a  48           pha
bd6b  24 5f        bit  $5f
bd6d  10 02        bpl  $bd71
bd6f  e6 5d        inc  $5d
bd71  20 e2 ba     jsr  $bae2
bd74  68           pla
bd75  38           sec
bd76  e9 30        sbc  #$30
bd78  20 7e bd     jsr  $bd7e
bd7b  4c 0a bd     jmp  $bd0a
```

```
bd7e  48           pha
bd7f  20 0c bc     jsr  $bc0c
bd82  68           pla
bd83  20 3c bc     jsr  $bc3c
bd86  a5 6e        lda  $6e
bd88  45 66        eor  $66
bd8a  85 6f        sta  $6f
bd8c  a6 61        ldx  $61
bd8e  4c 6a b8     jmp  $b86a
```

```
bd91  a5 5e        lda  $5e
bd93  c9 0a        cmp  #$0a
bd95  90 09        bcc  $bda0
bd97  a9 64        lda  #$64
bd99  24 60        bit  $60
bd9b  30 11        bmi  $bdae
bd9d  4c 7e b9     jmp  $b97e
```

```
bda0  0a           asl
bda1  0a           asl
bda2  18           clc
bda3  65 5e        adc  $5e
bda5  0a           asl
bda6  18           clc
bda7  a0 00        ldy  #$00
bda9  71 7a        adc  ($7a),y
bdab  38           sec
bdac  e9 30        sbc  #$30
bdae  85 5e        sta  $5e
bdb0  4c 30 bd     jmp  $bd30
```

```
bdb3 9b 3e bc 1f fd 9e 6e 6b; Data
bdbb 27 fd 9e 6e 6b 28 00
```

```
bdc2 a9 71          lda #$71; Lo
bdc4 a0 a3          ldy #$a3; Hi hlaska 'in'
bdc6 20 da bd       jsr $bdda; vystup
bdc9 a5 3a          lda $3a; Hi
bdcb a6 39          ldx $39; Lo prekl. radka
bdcd 85 62          sta $62
bdcf 86 63          stx $63
bdd1 a2 90          ldx #$90
bdd3 38             sec
bdd4 20 49 bc       jsr $bc49
bdd7 20 df bd       jsr $bddf
bdda 4c 1e ab       jmp $ab1e
```

```
bddd a0 01          ldy #$01
bddf a9 20          lda #$20
bde1 24 66          bit $66
bde3 10 02          bpl $bde7
bde5 a9 2d          lda #$2d
bde7 99 ff 00       sta $00ff,y
bdea 85 66          sta $66
bdec 84 71          sty $71
bdee c8             iny
bdef a9 30          lda #$30
bdf1 a6 61          ldx $61
bdf3 d0 03          bne $bdf8
bdf5 4c 04 bf       jmp $bf04
```

```
bdf8 a9 00          lda #$00
bdfa e0 80          cpx #$80
bdfc f0 02          beq $be00
bdfe b0 09          bcs $be09
be00 a9 bd          lda #$bd
be02 a0 bd          ldy #$bd
be04 20 28 ba       jsr $ba28
be07 a9 f7          lda #$f7
be09 85 5d          sta $5d
be0b a9 b8          lda #$b8
be0d a0 bd          ldy #$bd
be0f 20 5b bc       jsr $bc5b
be12 f0 1e          beq $be32
be14 10 12          bpl $be28
be16 a9 b3          lda #$b3
be18 a0 bd          ldy #$bd
be1a 20 5b bc       jsr $bc5b
be1d f0 02          beq $be21
be1f 10 0e          bpl $be2f
be21 20 e2 ba       jsr $bae2
be24 c6 5d          dec $5d
be26 d0 ee          bne $be16
be28 20 fe ba       jsr $bafe
be2b e6 5d          inc $5d
be2d d0 dc          bne $be0b
be2f 20 49 b8       jsr $b849
be32 20 9b bc       jsr $bc9b
```

```
be35  a2 01     ldx #$01
be37  a5 5d     lda $5d
be39  18        clc
be3a  69 0a     adc #$0a
be3c  30 09     bmi $be47
be3e  c9 0b     cmp #$0b
be40  b0 06     bcs $be48
be42  69 ff     adc #$ff
be44  aa        tax
be45  a9 02     lda #$02
be47  38        sec
be48  e9 02     sbc #$02
be4a  85 5e     sta $5e
be4c  86 5d     stx $5d
be4e  8a        txa
be4f  f0 02     beq $be53
be51  10 13     bpl $be66
be53  a4 71     ldy $71
be55  a9 2e     lda #$2e
be57  c8        iny
be58  99 ff 00  sta $00ff,y
be5b  8a        txa
be5c  f0 06     beq $be64
be5e  a9 30     lda #$30
be60  c8        iny
be61  99 ff 00  sta $00ff,y
be64  84 71     sty $71
be66  a0 00     ldy #$00
be68  a2 80     ldx #$80
be6a  a5 65     lda $65
be6c  18        clc
be6d  79 19 bf  adc $bf19,y
be70  85 65     sta $65
be72  a5 64     lda $64
be74  79 18 bf  adc $bf18,y
be77  85 64     sta $64
be79  a5 63     lda $63
be7b  79 17 bf  adc $bf17,y
be7e  85 63     sta $63
be80  a5 62     lda $62
be82  79 16 bf  adc $bf16,y
be85  85 62     sta $62
be87  e8        inx
be88  b0 04     bcs $be8e
be8a  10 de     bpl $be6a
be8c  30 02     bmi $be90
be8e  30 da     bmi $be6a
be90  8a        txa
be91  90 04     bcc $be97
be93  49 ff     eor #$ff
be95  69 0a     adc #$0a
be97  69 2f     adc #$2f
be99  c8        iny
be9a  c8        iny
be9b  c8        iny
be9c  c8        iny
be9d  84 47     sty $47
be9f  a4 71     ldy $71
```

```
bea1  c8          iny
bea2  aa          tax
bea3  29 7f       and #$7f
bea5  99 ff 00    sta $00ff,y
bea8  c6 5d       dec $5d
beaa  d0 06       bne $beb2
beac  a9 2e       lda #$2e
beae  c8          iny
beaf  99 ff 00    sta $00ff,y
beb2  84 71       sty $71
beb4  a4 47       ldy $47
beb6  8a          txa
beb7  49 ff       eor #$ff
beb9  29 80       and #$80
bebb  aa          tax
bebc  c0 24       cpy #$24
bebe  f0 04       beq $bec4
bec0  c0 3c       cpy #$3c
bec2  d0 a6       bne $be6a
bec4  a4 71       ldy $71
bec6  b9 ff 00    lda $00ff,y
bec9  88          dey
beca  c9 30       cmp #$30
becc  f0 f8       beq $bec6
bece  c9 2e       cmp #$2e
bed0  f0 01       beq $bed3
bed2  c8          iny
bed3  a9 2b       lda #$2b
bed5  a6 5e       ldx $5e
bed7  f0 2e       beq $bf07
bed9  10 08       bpl $bee3
bedb  a9 00       lda #$00
bedd  38          sec
bede  e5 5e       sbc $5e
bee0  aa          tax
bee1  a9 2d       lda #$2d
bee3  99 01 01    sta $0101,y
bee6  a9 45       lda #$45
bee8  99 00 01    sta $0100,y
beeb  8a          txa
beec  a2 2f       ldx #$2f
beee  38          sec
beef  e8          inx
bef0  e9 0a       sbc #$0a
bef2  b0 fb       bcs $beef
bef4  69 3a       adc #$3a
bef6  99 03 01    sta $0103,y
bef9  8a          txa
befa  99 02 01    sta $0102,y
befd  a9 00       lda #$00
beff  99 04 01    sta $0104,y
bf02  f0 08       beq $bf0c
bf04  99 ff 00    sta $00ff,y
bf07  a9 00       lda #$00
bf09  99 00 01    sta $0100,y
bf0c  a9 00       lda #$00
bf0e  a0 01       ldy #$01
bf10  60          rts
```

```
Data pro výpočet mocnin a odmocnin
bf11 80 00 00 00 00 fa 0a 1f
bf19 00 00 98 96 80 ff f0 bd
bf21 c0 00 01 86 a0 ff ff d8
bf29 f0 00 00 03 e8 ff ff ff
bf31 9c 00 00 00 0a ff ff ff
bf39 ff ff df 0a 80 00 03 4b
bf41 c0 ff ff 73 60 00 00 0e
bf49 10 ff ff fd a8 00 00 00
bf51 3c ec aa aa aa aa aa aa
bf59 aa aa aa aa aa aa aa aa
bf61 aa aa aa aa aa aa aa aa
bf69 aa aa aa aa aa aa aa aa
```

```
SQR
bf71 20 0c bc     jsr $bc0c
bf74 a9 11        lda #$11
bf76 a0 bf        ldy #$bf
bf78 20 a2 bb     jsr $bba2
↑
bf7b f0 70        beq $bfed
bf7d a5 69        lda $69
bf7f d0 03        bne $bf84
bf81 4c f9 b8     jmp $b8f9
```

```
bf84 a2 4e        ldx #$4e
bf86 a0 00        ldy #$00
bf88 20 d4 bb     jsr $bbd4
bf8b a5 6e        lda $6e
bf8d 10 0f        bpl $bf9e
bf8f 20 cc bc     jsr $bccc
bf92 a9 4e        lda #$4e
bf94 a0 00        ldy #$00
bf96 20 5b bc     jsr $bc5b
bf99 d0 03        bne $bf9e
bf9b 98           tya
bf9c a4 07        ldy $07
bf9e 20 fe bb     jsr $bbfe
bfa1 98           tya
bfa2 48           pha
bfa3 20 ea b9     jsr $b9ea
bfa6 a9 4e        lda #$4e
bfa8 a0 00        ldy #$00
bfaa 20 28 ba     jsr $ba28
bfad 20 ed bf     jsr $bfed
bfb0 68           pla
bfb1 4a           lsr
bfb2 90 0a        bcc $bfbe
bfb4 a5 61        lda $61
bfb6 f0 06        beq $bfbe
bfb8 a5 66        lda $66
bfba 49 ff        eor #$ff
bfbc 85 66        sta $66
bfbe 60           rts
```

```
bfbf  81 38 aa 3b 29 07 71 34; Data
bfc7  58 3e 56 74 16 7e b3 1b
bfcf  77 2f ee e3 85 7a 1d 84
bfd7  1c 2a 7c 63 59 58 0a 7e
bfdf  75 fd e7 c6 80 31 72 18
bfe7  10 81 00 00 00 00
```

```
bfed  a9 bf     lda #$bf
bfef  a0 bf     ldy #$bf
bff1  20 28 ba  jsr $ba28
bff4  a5 70     lda $70
bff6  69 50     adc #$50
bff8  90 03     bcc $bffd
bffa  20 23 bc  jsr $bc23
bffd  4c 00 e0  jmp $e000
```

Tabulka skoků

NePřímé skoky pro příkazy [uloženo $a00c-$a09d]

END	$a830	FOR	$a741	NEXT	$ad1d
DATA	$a8f8	INPUT#	$ab47	INPUT	$ab4a
DIM	$b081	READ	$b1aa	LET	$a9a3
GOTO	$a93a	NEW	$a7ae	IF	$aa06
RUN	$a871	GOSUB	$a883	RESTORE	$a933
RETURN	$a8d2	REM	$a8f8	STOP	$a831
ON	$a94b	PRINT#	$aa80	CONT	$a857
LIST	$a9f0	PRINT	$aa86	CLEAR	$a88a
CLR	$a8ac	GET#	$b1bf	CMD	$aa4a
SYS	$e12a	NEW	$a642	CLOSE	$ab1e

Přímé skoky na rutiny pro jednotlivé příkazy

SGN	$bc39	INT	$bccc	ABS	$bc58
SQR	$bf71	ERR$	$b976	FRE	$b37d
EXP	$bfed	SPC(	$b99a	POS	$b39e
LOG	$bf2b	STR$	$b94a	SYS	$e157
COS	$e264	VAL	$b7ad	USR	$0310
SIN	$e26b	ASC	$b78b	WAIT	$a80d
TAN	$e2b4	CHR$	$b7d6	PEEK	$b80d
ATN	$e30e	LEFT$	$b700	POKE	$b824
RIGHT$	$b72c	MID$	$b737		

NePřímé skoky (na rutiny) s dodatkovou hodnotou

+	$79	$b86a	-	$79	$b853	*	$7b	$ba2b
/	$7b	$bb12	↑	$7f	$bf7b	AND	$50	$afe9
OR	$46	$afe6	>	$7d	$bfb3	=	$5a	$aed3
<	$64	$b016						

Chybové hlášky (+"error")

$01	Too many files	$02	File open
$03	File not open	$04	File not found
$05	Device not present	$06	Not input file
$07	Not output file	$08	Missing file name
$09	Illegal device number	$0A	Next without for
$0B	Syntax	$0C	Return without gosub
$0D	Out of data	$0E	Illegal quantity
$0F	Overflow	$10	Out of memory
$11	Undef'd statement	$12	Bad subscript
$13	Redim'd array	$14	Division by zero
$15	Illegal direct	$16	Type mismatch
$17	String too long	$18	File data
$19	Formula too complex	$1A	Can't continue
$1B	Undef'd function	$1C	Verify
$1D	Load	$1E	Break

```
e000  85 56     sta $56
e002  20 0f bc  jsr $bc0f
e005  a5 61     lda $61
e007  c9 88     cmp #$88
e009  90 03     bcc $e00e
e00b  20 d4 ba  jsr $bad4
e00e  20 cc bc  jsr $bccc
e011  a5 07     lda $07
e013  18        clc
e014  69 81     adc #$81
e016  f0 f3     beq $e00b
e018  38        sec
e019  e9 01     sbc #$01
e01b  48        pha
e01c  a2 05     ldx #$05
e01e  b5 69     lda $69,x
e020  b4 61     ldy $61,x
e022  95 61     sta $61,x
e024  94 69     sty $69,x
e026  ca        dex
e027  10 f5     bpl $e01e
e029  a5 56     lda $56
e02b  85 70     sta $70
e02d  20 53 b8  jsr $b853
e030  20 b4 bf  jsr $bfb4
e033  a9 c4     lda #$c4
e035  a0 bf     ldy #$bf
e037  20 59 e0  jsr $e059
e03a  a9 00     lda #$00
e03c  85 6f     sta $6f
e03e  68        pla
e03f  20 b9 ba  jsr $bab9
e042  60        rts
```

```
e043  85 71     sta $71
e045  84 72     sty $72
e047  20 ca bb  jsr $bbca
e04a  a9 57     lda #$57
e04c  20 28 ba  jsr $ba28
e04f  20 5d e0  jsr $e05d
e052  a9 57     lda #$57
e054  a0 00     ldy #$00
e056  4c 28 ba  jmp $ba28
```

```
e059  85 71     sta $71
e05b  84 72     sty $72
e05d  20 c7 bb  jsr $bbc7
e060  b1 71     lda ($71),y
e062  85 67     sta $67
e064  a4 71     ldy $71
e066  c8        iny
e067  98        tya
e068  d0 02     bne $e06c
e06a  e6 72     inc $72
e06c  85 71     sta $71
```

```
e06a  a4 72        ldy $72
e070  20 28 ba     jsr $ba28
e073  a5 71        lda $71
e075  a4 72        ldy $72
e077  18           clc
e078  69 05        adc #$05
e07a  90 01        bcc $e07d
e07c  c8           iny
e07d  85 71        sta $71
e07f  84 72        sty $72
e081  20 67 b8     jsr $b867
e084  a9 5c        lda #$5c
e086  a0 00        ldy #$00
e088  c6 67        dec $67
e08a  d0 e4        bne $e070
e08c  60           rts
-------------------------------------------------
e08d  98 35 44 7a 00 68 28 b1; Data pro RND
e095  46 00
-------------------------------------------------
RND
e097  20 2b bc     jsr $bc2b
e09a  30 37        bmi $e0d3
e09c  d0 20        bne $e0be
e09e  20 f3 ff     jsr $fff3; IOBASE
e0a1  86 22        stx $22
e0a3  84 23        sty $23
e0a5  a0 04        ldy #$04
e0a7  b1 22        lda ($22),y
e0a9  85 62        sta $62
e0ab  c8           iny
e0ac  b1 22        lda ($22),y
e0ae  85 64        sta $64
e0b0  a0 08        ldy #$08
e0b2  b1 22        lda ($22),y
e0b4  85 63        sta $63
e0b6  c8           iny
e0b7  b1 22        lda ($22),y
e0b9  85 65        sta $65
e0bb  4c e3 e0     jmp $e0e3
-------------------------------------------------
e0be  a9 8b        lda #$8b; Lo adresa dat pro
e0c0  a0 00        ldy #$00; Hi urceni RND (baze)
e0c2  20 a2 bb     jsr $bba2
e0c5  a9 8d        lda #$8d; Lo adresa bloku
e0c7  a0 e0        ldy #$e0; Hi dat 1
e0c9  20 28 ba     jsr $ba28
e0cc  a9 92        lda #$92; Lo adresa bloku
e0ce  a0 e0        ldy #$e0; Hi dat 2
e0d0  20 67 b8     jsr $b867
e0d3  a6 65        ldx $65
e0d5  a5 62        lda $62
e0d7  85 65        sta $65
e0d9  86 62        stx $62
e0db  a6 63        ldx $63
e0dd  a5 64        lda $64
e0df  85 63        sta $63
e0e1  86 64        stx $64
```

```
e0e3  a9 00        lda #$00
e0e5  85 66        sta $66
e0e7  a5 61        lda $61
e0e9  85 70        sta $70
e0eb  a9 80        lda #$80
e0ed  85 61        sta $61
e0ef  20 d7 b8     jsr $b8d7
e0f2  a2 8b        ldx #$8b
e0f4  a0 00        ldy #$00
e0f6  4c d4 bb     jmp $bbd4
-------------------------------------
Zaver po volani nasledujicich rutin
e0f9  c9 f0        cmp #$f0
e0fb  d0 07        bne $e104
e0fd  84 38        sty $38; Hi adresa konce
e0ff  86 37        stx $37; Lo RAM pro Basic
e101  4c 63 a6     jmp $a663
-------------------------------------
e104  aa           tax ; preneseni kodu chyby
e105  d0 02        bne $e109
e107  a2 1e        ldx #$1e; kod 'break error'
e109  4c 37 a4     jmp $a437; chybova hlaska
-------------------------------------
e10c  20 d2 ff     jsr $ffd2; CHROUT
e10f  b0 e9        bcs $e0f9
e111  60           rts
-------------------------------------
e112  20 cf ff     jsr $ffcf; CHRIN
e115  b0 e2        bcs $e0f9
e117  60           rts
-------------------------------------
e118  20 ad e4     jsr $e4ad
e11b  b0 dc        bcs $e0f9
e11d  60           rts
-------------------------------------
e11e  20 c6 ff     jsr $ffc6; CHKIN
e121  b0 d6        bcs $e0f9
e123  60           rts
-------------------------------------
e124  20 e4 ff     jsr $ffe4; GETIN
e127  b0 d0        bcs $e0f9
e129  60           rts
-------------------------------------
Obsluha skoku pri GOTO;GOSUB apod.
e12a  20 8a ad     jsr $ad8a
e12d  20 f7 b7     jsr $b7f7
e130  a9 e1        lda #$e1; Hi
e132  48           pha
e133  a9 46        lda #$46; Lo navratova adresa
e135  48           pha
e136  ad 0f 03     lda $030f; Nacteni predchoziho
e139  48           pha
e13a  ad 0c 03     lda $030c; stavu SR;A;X;Y
e13d  ae 0d 03     ldx $030d
e140  ac 0e 03     ldy $030e
e143  28           plp
e144  6c 14 00     jmp ($0014)
-------------------------------------
```

```
Obsluha navratu z podprogramu (GOSUB-RETURN)
e147 08          php
e148 8d 0c 03    sta $030c; Ulozeni soucasneho
e14b 8e 0d 03    stx $030d; stavu SR;A;X;Y do
e14e 8c 0e 03    sty $030e; syst. promennych
e151 68          pla
e152 8d 0f 03    sta $030f
e155 60          rts
-------------------------------------------------
SAVE
e156 20 d4 e1    jsr $e1d4
e159 a6 2d       ldx $2d; Lo
e15b a4 2e       ldy $2e; Hi Konec programu
e15d a9 2b       lda #$2b; neprima adresa pocatku
e15f 20 d8 ff    jsr $ffd8; SAVE
e162 b0 95       bcs $e0f9
e164 60          rts
-------------------------------------------------
VERIFY
e165 a9 01       lda #$01
LOAD (od $e168)
e167 2c a9 00    bit $00a9
e16a 85 0a       sta $0a
e16c 20 d4 e1    jsr $e1d4
e16f a5 0a       lda $0a
e171 a6 2b       ldx $2b; Lo
e173 a4 2c       ldy $2c; Hi Pocatek
e175 20 d5 ff    jsr $ffd5; LOAD
e178 b0 57       bcs $e1d1
e17a a5 0a       lda $0a
e17c f0 17       beq $e195
e17e a2 1c       ldx #$1c; Kod 'Verify error'
e180 20 b7 ff    jsr $ffb7; READST
e183 29 10       and #$10
e185 d0 17       bne $e19e
e187 a5 7a       lda $7a; Lo CHRGET
e189 c9 02       cmp #$02; mod (primy/programovy)?
e18b f0 07       beq $e194
e18d a9 64       lda #$64; Lo adresa hlas-
e18f a0 a3       ldy #$a3; Hi ky 'OK'
e191 4c 1e ab    jmp $ab1e; tisk hlasky
e194 60          rts
-------------------------------------------------
e195 20 b7 ff    jsr $ffb7; READST
e198 29 bf       and #$bf
e19a f0 05       beq $e1a1
e19c a2 1d       ldx #$1d; Kod 'Load error'
e19e 4c 37 a4    jmp $a437; chybova hlaska
-------------------------------------------------
e1a1 a5 7b       lda $7b; Hi CHRGET
e1a3 c9 02       cmp #$02
e1a5 d0 0e       bne $e1b5
e1a7 8e 2d       stx $2d; Lo Konec progra-
e1a9 84 2e       sty $2e; Hi mu v Basicu
e1ab a9 76       lda #$76; Lo adresa hlas-
e1ad a0 a3       ldy #$a3; Hi ky 'Ready.'
e1af 20 1e ab    jsr $ab1e; tisk hlasky
e1b2 4c 2a a5    jmp $a52a
```

```
e1b5  20 8e a6     jsr $a68e
e1b8  20 33 a5     jsr $a533
e1bb  4c 77 a6     jmp $a677
```

```
e1be  20 19 e2     jsr $e219
e1c1  20 c0 ff     jsr $ffc0; OPEN
e1c4  b0 0b        bcs $e1d1
e1c6  60           rts
```

```
e1c7  20 19 e2     jsr $e219
e1ca  a5 49        lda $49
e1cc  20 c3 ff     jsr $ffc3; CLOSE
e1cf  90 c3        bcc $e194
e1d1  4c f9 e0     jmp $e0f9
```

Nastaveni jmena a Kodu Periferie Pro LOAD;SAVE a VERIFY

```
e1d4  a9 00        lda #$00
e1d6  20 bd ff     jsr $ffbd; SETNAM
e1d9  a2 01        ldx #$01
e1db  a0 00        ldy #$00
e1dd  20 ba ff     jsr $ffba; SETLFS
e1e0  20 06 e2     jsr $e206
e1e3  20 57 e2     jsr $e257
e1e6  20 06 e2     jsr $e206
e1e9  20 00 e2     jsr $e200
e1ec  a0 00        ldy #$00
e1ee  86 49        stx $49
e1f0  20 ba ff     jsr $ffba
e1f3  20 06 e2     jsr $e206
e1f6  20 00 e2     jsr $e200
e1f9  8a           txa
e1fa  a8           tay
e1fb  a6 49        ldx $49
e1fd  4c ba ff     jmp $ffba
e200  20 0e e2     jsr $e20e
e203  4c 9e b7     jmp $b79e
e206  20 79 00     jsr $0079
e209  d0 02        bne $e20d
e20b  68           pla
e20c  68           pla
e20d  60           rts
```

```
e20e  20 fd ae     jsr $aefd
e211  20 79 00     jsr $0079; CHRGET bez posunuti
e214  d0 f7        bne $e20d
e216  4c 08 af     jmp $af08; 'Syntax error'
```

```
e219  a9 00        lda #$00
e21b  20 bd ff     jsr $ffbd
e21e  20 11 e2     jsr $e211
e221  20 9e b7     jsr $b79e
e224  86 49        stx $49
e226  8a           txa
e227  a2 01        ldx #$01
e229  a0 00        ldy #$00
e22b  20 ba ff     jsr $ffba
e22e  20 06 e2     jsr $e206
```

```
e231  20 00 e2     jsr $e200
e234  86 4a        stx $4a
e236  a0 00        ldy #$00
e238  a5 49        lda $49
e23a  e0 03        cpx #$03
e23c  90 01        bcc $e23f
e23e  88           dey
e23f  20 ba ff     jsr $ffba
e242  20 06 e2     jsr $e206
e245  20 00 e2     jsr $e200
e248  8a           txa
e249  a8           tay
e24a  a6 4a        ldx $4a
e24c  a5 49        lda $49
e24e  20 ba ff     jsr $ffba
e251  20 06 e2     jsr $e206
e254  20 0e e2     jsr $e20e
e257  20 9e ad     jsr $ad9e
e25a  20 a3 b6     jsr $b6a3
e25d  a6 22        ldx $22
e25f  a4 23        ldy $23
e261  4c bd ff     jmp $ffbd; SETNAM
-------------------------------------------------
COS
e264  a9 e0        lda #$e0; Lo data pro vypocet
e266  a0 e2        ldy #$e2; Hi hodn. fce cosinus
e268  20 67 b8     jsr $b867
SIN
e26b  20 0c bc     jsr $bc0c
e26e  a9 e5        lda #$e5; Lo
e270  a0 e2        ldy #$e2; Hi data
e272  a6 6e        ldx $6e
e274  20 07 bb     jsr $bb07
e277  20 0c bc     jsr $bc0c
e27a  20 cc bc     jsr $bccc
e27d  a9 00        lda #$00
e27f  85 6f        sta $6f
e281  20 53 b8     jsr $b853
e284  a9 ea        lda #$ea; Lo
e286  a0 e2        ldy #$e2; Hi data
e288  20 50 b8     jsr $b850
e28b  a5 66        lda $66
e28d  48           pha
e28e  10 0d        bpl $e29d
e290  20 49 b8     jsr $b849
e293  a5 66        lda $66
e295  30 09        bmi $e2a0
e297  a5 12        lda $12
e299  49 ff        eor #$ff
e29b  85 12        sta $12
e29d  20 b4 bf     jsr $bfb4
e2a0  a9 ea        lda #$ea; Lo
e2a2  a0 e2        ldy #$e2; Hi data
e2a4  20 67 b8     jsr $b867
e2a7  68           pla
e2a8  10 03        bpl $e2ad
e2aa  20 b4 bf     jsr $bfb4
e2ad  a9 ef        lda #$ef; Lo

220
```

```
e2af  a0 e2        ldy #$e2; Hi data
e2b1  4c 43 e0     jmp $e043
-------------------------------------------------
TAN
e2b4  20 ca bb     jsr $bbca
e2b7  a9 00        lda #$00
e2b9  85 12        sta $12
e2bb  20 6b e2     jsr $e26b
e2be  a2 4e        ldx #$4e
e2c0  a0 00        ldy #$00
e2c2  20 f6 e0     jsr $e0f6
e2c5  a9 57        lda #$57
e2c7  a0 00        ldy #$00
e2c9  20 a2 bb     jsr $bba2
e2cc  a9 00        lda #$00
e2ce  85 66        sta $66
e2d0  a5 12        lda $12
e2d2  20 dc e2     jsr $e2dc
e2d5  a9 4e        lda #$4e
e2d7  a0 00        ldy #$00
e2d9  4c 0f bb     jmp $bb0f
e2dc  48           pha
e2dd  4c 9d e2     jmp $e29d
-------------------------------------------------
e2e0  81 49 0f da a2 83 49 0f; Data pro vypocet fci
e2e8  da a2 7f 00 00 00 00 05
e2f0  84 e6 1a 2d 1b 86 28 07
e2f8  fb f8 87 99 68 89 01 87
e300  23 35 df e1 86 a5 5d e7
e308  28 83 49 0f da a2
-------------------------------------------------
ATN
e30e  a5 66        lda $66
e310  48           pha
e311  10 03        bpl $e316
e313  20 b4 bf     jsr $bfb4
e316  a5 61        lda $61
e318  48           pha
e319  c9 81        cmp #$81
e31b  90 07        bcc $e324
e31d  a9 bc        lda #$bc; Lo
e31f  a0 b9        ldy #$b9; Hi data
e321  20 0f bb     jsr $bb0f
e324  a9 3e        lda #$3e; Lo
e326  a0 e3        ldy #$e3; Hi data
e328  20 43 e0     jsr $e043
e32b  68           pla
e32c  c9 81        cmp #$81
e32e  90 07        bcc $e337
e330  a9 e0        lda #$e0; Lo
e332  a0 e2        ldy #$e2; Hi data
e334  20 50 b8     jsr $b850
e337  68           pla
e338  10 03        bpl $e33d
e33a  4c b4 bf     jmp $bfb4
e33d  60           rts
-------------------------------------------------
e33e  0b 76 b3 83 bd d3 79 1e; Data pro vypocet fca
```

```
e346 f4 a6 f5 7b 83 fc b0 10
e34e 7c 0c 1f 67 ca 7c de 53
e356 cb c1 7d 14 64 70 4c 7d
e35e b7 ea 51 7a 7d 63 30 88
e366 7e 7e 92 44 99 3a 7e 4c
e36e cc 91 c7 7f aa aa aa 13
e376 81 00 00 00 00
-------------------------------------------------------------
e37b 20 cc ff       jsr $ffcc; CLRCHN
e37e a9 00          lda #$00
e380 85 13          sta $13; argument CMD
e382 20 7a a6       jsr $a67a
e385 58             cli
e386 a2 80          ldx #$80;
e388 6c 00 03       jmp ($0300); obsluha chyb. hlaseK
-------------------------------------------------------------
e38b 8a             txa
e38c 30 03          bmi $e391; skoK pri X>$7f
e38e 4c 3a a4       jmp $a43a; chybova hlaska
e391 4c 74 a4       jmp $a474; 'Ready.'+start Basicu
-------------------------------------------------------------
e394 20 53 e4       jsr $e453; vektory $0300 a dale
e397 20 bf e3       jsr $e3bf; nastaveni syst.prom.
e39a 20 22 e4       jsr $e422; hlavicka po RESETu
e39d a2 fb          ldx #$fb
e39f 9a             txs
e3a0 d0 e4          bne $e386
-------------------------------------------------------------
e3a2 e6 7a          inc $7a; Lo
e3a4 d0 02          bne $e3a8
e3a6 e6 7b          inc $7b; Hi CHRGET
e3a8 ad 60 ea       lda $ea60
e3ab c9 3a          cmp #$3a
e3ad b0 0a          bcs $e3b9
e3af c9 20          cmp #$20
e3b1 f0 ef          beq $e3a2
e3b3 38             sec
e3b4 e9 30          sbc #$30
e3b6 38             sec
e3b7 e9 d0          sbc #$d0
e3b9 60             rts
-------------------------------------------------------------
Baze pro RND
e3ba 80 4f c7 52 58; ($008b - $008f)
-------------------------------------------------------------
e3bf a9 4c          lda #$4c; kod instrukce JMP
e3c1 85 54          sta $54; skoK pro funkci
e3c3 8d 10 03       sta $0310; skoK pro USR
e3c6 a9 48          lda #$48; Lo standardni skoK
e3c8 a0 b2          ldy #$b2; Hi Pri Pouziti USR
e3ca 8d 11 03       sta $0311
e3cd 8c 12 03       sty $0312
e3d0 a9 91          lda #$91; Lo adresa Procedury pro
e3d2 a0 b3          ldy #$b3; Hi Prevod int. => real
e3d4 85 05          sta $05
e3d6 84 06          sty $06
e3d8 a9 aa          lda #$aa; Lo adresa Procedury pro
e3da a0 b1          ldy #$b1; Hi Prevod real => int.
```

```
e3dc 85 03        sta $03
e3de 84 04        sty $04
e3e0 a2 1c        ldx #$1c
e3e2 bd a2 e3     lda $e3a2,x; prepsani CHRGET
e3e5 95 73        sta $73,x;  z ROM do syst. prom.
e3e7 ca           dex
e3e8 10 f8        bpl $e3e2
e3ea a9 03        lda #$03
e3ec 85 53        sta $53
e3ee a9 00        lda #$00
e3f0 85 68        sta $68
e3f2 85 13        sta $13
e3f4 85 18        sta $18
e3f6 a2 01        ldx #$01
e3f8 8e fd 01     stx $01fd
e3fb 8e fc 01     stx $01fc
e3fe a2 19        ldx #$19
e400 86 16        stx $16
e402 38           sec
e403 20 9c ff     jsr $ff9c; MEMBOT
e406 86 2b        stx $2b; Lo
e408 84 2c        sty $2c; Hi pocatek Basicu
e40a 38           sec
e40b 20 99 ff     jsr $ff99; MEMTOP
e40e 86 37        stx $37; Lo konec RAM
e410 84 38        sty $38; Hi pro Basic
e412 86 33        stx $33; Lo
e414 84 34        sty $34; Hi pocatek retezcu
e416 a0 00        ldy #$00; ulozeni $00 na pocatek
e418 98           tya ; Basic RAM
e419 91 2b        sta ($2b),y
e41b e6 2b        inc $2b
e41d d0 02        bne $e421
e41f e6 2c        inc $2c
e421 60           rts
-------------------------------------------------------
e422 a5 2b        lda $2b; Lo pocatek progra-
e424 a4 2c        ldy $2c; Hi mu v Basicu
e426 20 08 a4     jsr $a408; nast.ukaz. promennych
e429 a9 73        lda #$73; Lo adresa hlavicky
e42b a0 e4        ldy #$e4; Hi (po RESETu)
e42d 20 1e ab     jsr $ab1e; tisk
Test - kolik mame volnych bytu pro Basic
e430 a5 37        lda $37
e432 38           sec
e433 e5 2b        sbc $2b
e435 aa           tax
e436 a5 38        lda $38
e438 e5 2c        sbc $2c
e43a 20 cd bd     jsr $bdcd; decim. vypise pocet
e43d a9 60        lda #$60; Lo adresa zbytku
e43f a0 e4        ldy #$e4; Hi uvodni hlasky
e441 20 1e ab     jsr $ab1e; tisk
e444 4c 44 a6     jmp $a644; rutina pro prikaz NEW
-------------------------------------------------------
Hodnoty vektoru ulozenych od $0300 do $030b
e447 8b e3 83 a4 7c a5 1a a7
e44f e4 a7 86 ae
```

```
e453  a2 0b       ldx #$0b; Preneseni vektoru
e455  bd 47 e4    lda $e447,x
e458  9d 00 03    sta $0300,x
e45b  ca          dex
e45c  10 f7       bpl $e455
e45e  60          rts
```

```
e45f  00 20 42 41 53 49 43 20;  .. basic
e467  42 59 54 45 53 20 46 52;  bytes fr
e46f  45 45 0d 00 93 0d 20 20;  ee
e477  20 20 2a 2a 2a 2a 20 43;  ..**** c
e47f  4f 4d 4d 4f 44 4f 52 45;  ommodore
e487  20 36 34 20 42 41 53 49;  .64 basi
e48f  43 20 56 32 20 2a 2a 2a;  c v2 ***
e497  2a 0d 0d 20 36 34 4b 20;  *    64k
e49f  52 41 4d 20 53 59 53 54;  ram syst
e4a7  45 4d 20 20 00            ;  em
```

```
e4ad  48          pha
e4ae  20 c9 ff    jsr $ffc9; CHKOUT
e4b1  aa          tax
e4b2  68          pla
e4b3  90 01       bcc $e4b6
e4b5  8a          txa
e4b6  60          rts
```

```
e4b7  aa          tax
e4b8  aa          tax
e4b9  aa          tax
e4ba  aa          tax
e4bb  aa          tax
e4bc  aa          tax
e4bd  aa          tax
e4be  aa          tax
e4bf  aa          tax
e4c0  aa          tax
e4c1  aa          tax
e4c2  aa          tax
e4c3  aa          tax
e4c4  aa          tax
e4c5  aa          tax
e4c6  aa          tax
e4c7  aa          tax
e4c8  aa          tax
e4c9  aa          tax
e4ca  aa          tax
e4cb  aa          tax
e4cc  aa          tax
e4cd  aa          tax
e4ce  aa          tax
e4cf  aa          tax
e4d0  aa          tax
e4d1  aa          tax
e4d2  aa          tax
e4d3  85 a9       sta $a9; startbit RS-232
e4d5  a9 01       lda #$01
e4d7  85 ab       sta $ab; Parita
```

```
e4d9 60             rts
-------------------------------------------------------
e4da ad 86 02       lda $0286; Kod barvy znaku
e4dd 91 f3          sta ($f3),y; zapis do color-RAM
e4df 60             rts
-------------------------------------------------------
Brzdeni tisku klavesou CONTROL
e4e0 69 02          adc #$02
e4e2 a4 91          ldy $91
e4e4 c8             iny
e4e5 d0 04          bne $e4eb
e4e7 c5 a1          cmp $a1; cas -stredni Byte
e4e9 d0 f7          bne $e4e2
e4eb 60             rts
-------------------------------------------------------
e4ec 19 26 44 19 1a 11 e8 0d; Data
e4f4 70 0c 06 06 d1 02 37 01
e4fc aa 00 69 00
-------------------------------------------------------
IOBASE
e500 a2 00          ldx #$00
e502 a0 dc          ldy #$dc
e504 60             rts
-------------------------------------------------------
SCREEN
e505 a2 28          ldx #$28
e507 a0 19          ldy #$19
e509 60             rts
-------------------------------------------------------
PLOT
e50a b0 07          bcs $e513
e50c 86 d6          stx $d6; radek
e50e 84 d3          sty $d3; sloupec - pro Kurzor
e510 20 6c e5       jsr $e56c
e513 a6 d6          ldx $d6
e515 a4 d3          ldy $d3
e517 60             rts
-------------------------------------------------------
Inicializace VIC (nastaveni Poc. Parametru)
e518 20 a0 e5       jsr $e5a0; nastaveni in/out
e51b a9 00          lda #$00
e51d 8d 91 02       sta $0291; velka Pismena+grafika
e520 85 cf          sta $cf; blik. Kurzoru
e522 a9 48          lda #$48; Lo adresa dekodovaci
e524 8d 8f 02       sta $028f
e527 a9 eb          lda #$eb; Hi rutiny klavesnice
e529 8d 90 02       sta $0290
e52c a9 0a          lda #$0a
e52e 8d 89 02       sta $0289; velikost kl. buffru
e531 8d 8c 02       sta $028c; zpozdeni
e534 a9 0e          lda #$0e
e536 8d 86 02       sta $0286; barva znaku
e539 a9 04          lda #$04
e53b 8d 8b 02       sta $028b; zpozdeni pri cteni
e53e a9 0c          lda #$0c
e540 85 cd          sta $cd; blikani
e542 85 cc          sta $cc; Kurzoru
e544 ad 88 02       lda $0288; Hi Byte obraz. RAM
```

```
e547 09 80      ora #$80
e549 a8         tay
e54a a9 00      lda #$00
e54c aa         tax
e54d 94 d9      sty $d9,x; Parametry obraz. radek
e54f 18         clc
e550 69 28      adc #$28; (40 znaku na radce)
e552 90 01      bcc $e555
e554 c8         iny
e555 e8         inx
e556 e0 1a      cpx #$1a; jsme uz na 26-te radce?
e558 d0 f3      bne $e54d
e55a a9 ff      lda #$ff
e55c 95 d9      sta $d9,x
e55e a2 18      ldx #$18
e560 20 ff e9   jsr $e9ff
e563 ca         dex
e564 10 fa      bpl $e560
e566 a0 00      ldy #$00; 'Crsr Home'
e568 84 d3      sty $d3
e56a 84 d6      sty $d6
e56c a6 d6      ldx $d6
e56e a5 d3      lda $d3
e570 b4 d9      ldy $d9,x
e572 30 08      bmi $e57c; Prestup.rad.?Ne=skok
e574 18         clc
e575 69 28      adc #$28
e577 85 d3      sta $d3
e579 ca         dex
e57a 10 f4      bpl $e570
e57c 20 f0 e9   jsr $e9f0
e57f a9 27      lda #$27
e581 e8         inx
e582 b4 d9      ldy $d9,x
e584 30 06      bmi $e58c
e586 18         clc
e587 69 28      adc #$28
e589 e8         inx
e58a 10 f6      bpl $e582
e58c 85 d5      sta $d5
e58e 4c 24 ea   jmp $ea24
```

```
e591 e4 c9      cpx $c9
e593 f0 03      beq $e598
e595 4c ed e6   jmp $e6ed
e598 60         rts
```

```
e599 ea         nop
e59a 20 a0 e5   jsr $e5a0
e59d 4c 66 e5   jmp $e566
```

```
e5a0 a9 03      lda #$03
e5a2 85 9a      sta $9a; Kod Periferie pro vystup
e5a4 a9 00      lda #$00
e5a6 85 99      sta $99; Kod Periferie pro vstup
Prenos dat do registru VIC
e5a8 a2 2f      ldx #$2f
e5aa bd b8 ec   lda $ecb8,x
```

```
e5ad 9d ff cf     sta $cfff,x
e5b0 ca           dex
e5b1 d0 f7        bne $e5aa
e5b3 60           rts
-------------------------------------------------
Cteni znaku z buffru + posun zbylych
e5b4 ac 77 02     ldy $0277
e5b7 a2 00        ldx #$00
e5b9 bd 78 02     lda $0278,x
e5bc 9d 77 02     sta $0277,x
e5bf e8           inx
e5c0 e4 c6        cpx $c6; Pocet zbylych znaku
e5c2 d0 f5        bne $e5b9
e5c4 c6 c6        dec $c6
e5c6 98           tya
e5c7 58           cli
e5c8 18           clc
e5c9 60           rts
-------------------------------------------------
Obsluha cteni znaku
e5ca 20 16 e7     jsr $e716
e5cd a5 c6        lda $c6
e5cf 85 cc        sta $cc
e5d1 8d 92 02     sta $0292
e5d4 f0 f7        beq $e5cd
e5d6 78           sei
e5d7 a5 cf        lda $cf
e5d9 f0 0c        beq $e5e7
e5db a5 ce        lda $ce; Kod znaku pod Kurzorem
e5dd ae 87 02     ldx $0287; barva
e5e0 a0 00        ldy #$00
e5e2 84 cf        sty $cf
e5e4 20 13 ea     jsr $ea13; znak/barva => RAM
e5e7 20 b4 e5     jsr $e5b4; cteni z buffru
e5ea c9 83        cmp #$83; kod 'SHIFT'+'RUN/STOP'
e5ec d0 10        bne $e5fe
e5ee a2 09        ldx #$09
e5f0 78           sei
e5f1 86 c6        stx $c6; Pocet znaku v buffru
e5f3 bd e6 ec     lda $ece6,x; naplneni buffru kody
e5f6 9d 76 02     sta $0276,x; prikazu 'LOAD'a'RUN'
e5f9 ca           dex
e5fa d0 f7        bne $e5f3
e5fc f0 cf        beq $e5cd;
e5fe c9 0d        cmp #$0d; <RETURN>
e600 d0 c8        bne $e5ca
e602 a4 d5        ldy $d5;
e604 84 d0        sty $d0
e606 b1 d1        lda ($d1),y; cteni z obr. RAM
e608 c9 20        cmp #$20; <SPACE>
e60a d0 03        bne $e60f
e60c 88           dey
e60d d0 f7        bne $e606
e60f c8           iny
e610 84 c8        sty $c8; Pocet zn. na radce
e612 a0 00        ldy #$00
e614 8c 92 02     sty $0292
e617 84 d3        sty $d3
```

```
e619  84  d4        sty  $d4
e61b  a5  c9        lda  $c9
e61d  30  1b        bmi  $e63a
e61f  a6  d6        ldx  $d6
e621  20  91  e5    jsr  $e591
e624  e4  c9        cpx  $c9
e626  d0  12        bne  $e63a
e628  a5  ca        lda  $ca
e62a  85  d3        sta  $d3
e62c  c5  c8        cmp  $c8
e62e  90  0a        bcc  $e63a
e630  b0  2b        bcs  $e65d
```

```
Vetsi  cast rutiny CHRIN
e632  98            tya
e633  48            pha
e634  8a            txa
e635  48            pha
e636  a5  d0        lda  $d0
e638  f0  93        beq  $e5cd
e63a  a4  d3        ldy  $d3
e63c  b1  d1        lda  ($d1),y
e63e  85  d7        sta  $d7
e640  29  3f        and  #$3f
e642  06  d7        asl  $d7
e644  24  d7        bit  $d7
e646  10  02        bpl  $e64a
e648  09  80        ora  #$80
e64a  90  04        bcc  $e650
e64c  a6  d4        ldx  $d4
e64e  d0  04        bne  $e654
e650  70  02        bvs  $e654
e652  09  40        ora  #$40
e654  e6  d3        inc  $d3
e656  20  84  e6    jsr  $e684
e659  c4  c8        cpy  $c8
e65b  d0  17        bne  $e674
e65d  a9  00        lda  #$00
e65f  85  d0        sta  $d0
e661  a9  0d        lda  #$0d
e663  a6  99        ldx  $99
e665  e0  03        cpx  #$03
e667  f0  06        beq  $e66f
e669  a6  9a        ldx  $9a
e66b  e0  03        cpx  #$03
e66d  f0  03        beq  $e672
e66f  20  16  e7    jsr  $e716
e672  a9  0d        lda  #$0d
e674  85  d7        sta  $d7
e676  68            pla
e677  aa            tax
e678  68            pla
e679  a8            tay
e67a  a5  d7        lda  $d7
e67c  c9  de        cmp  #$de; Kod 'PI'
e67e  d0  02        bne  $e682
e680  a9  ff        lda  #$ff
e682  18            clc
```

```
e683 60             rts
------------------------------------------------------
Uvozovkovy mod zap/vyp
e684 c9 22          cmp #$22
e686 d0 08          bne $e690
e688 a5 d4          lda $d4
e68a 49 01          eor #$01
e68c 85 d4          sta $d4
e68e a9 22          lda #$22; Kod uvozovek
e690 60             rts
------------------------------------------------------
e691 09 40          ora #$40
e693 a6 c7          ldx $c7; inverzni znaky
e695 f0 02          beq $e699
e697 09 80          ora #$80; Prepis do inverz. tvaru
e699 a6 d8          ldx $d8
e69b f0 02          beq $e69f
e69d c6 d8          dec $d8
e69f ae 86 02       ldx $0286
e6a2 20 13 ea       jsr $ea13; znak/barva => RAM
e6a5 20 b6 e6       jsr $e6b6
e6a8 68             pla
e6a9 a8             tay
e6aa a5 d8          lda $d8
e6ac f0 02          beq $e6b0
e6ae 46 d4          lsr $d4
e6b0 68             pla
e6b1 aa             tax
e6b2 68             pla
e6b3 18             clc
e6b4 58             cli
e6b5 60             rts
------------------------------------------------------
Obsluha obrazovky
e6b6 20 b3 e8       jsr $e8b3
e6b9 e6 d3          inc $d3
e6bb a5 d5          lda $d5
e6bd c5 d3          cmp $d3
e6bf b0 3f          bcs $e700
e6c1 c9 4f          cmp #$4f
e6c3 f0 32          beq $e6f7
e6c5 ad 92 02       lda $0292
e6c8 f0 03          beq $e6cd
e6ca 4c 67 e9       jmp $e967
e6cd a6 d6          ldx $d6
e6cf e0 19          cpx #$19
e6d1 90 07          bcc $e6da
e6d3 20 ea e8       jsr $e8ea
e6d6 c6 d6          dec $d6
e6d8 a6 d6          ldx $d6
e6da 16 d9          asl $d9,x; je-li radek prestupny
e6dc 56 d9          lsr $d9,x; vynuluje se 7-my bit
e6de e8             inx
e6df b5 d9          lda $d9,x
e6e1 09 80          ora #$80; dalsi radek je uz
e6e3 95 d9          sta $d9,x; normalni
e6e5 ca             dex
e6e6 a5 d5          lda $d5
```

```
e6e8  18           clc
e6e9  69 28        adc #$28
e6eb  85 d5        sta $d5
e6ed  b5 d9        lda $d9,x; hledani pruniho pred-
e6ef  30 03        bmi $e6f4; choziho neprestupneho
e6f1  ca           dex ; radku
e6f2  d0 f9        bne $e6ed
e6f4  4c f0 e9     jmp $e9f0
-------------------------------------------------------
e6f7  c6 d6        dec $d6
e6f9  20 7c e8     jsr $e87c
e6fc  a9 00        lda #$00
e6fe  85 d3        sta $d3
e700  60           rts
-------------------------------------------------------
e701  a6 d6        ldx $d6
e703  d0 06        bne $e70b
e705  86 d3        stx $d3
e707  68           pla
e708  68           pla
e709  d0 9d        bne $e6a8
e70b  ca           dex
e70c  86 d6        stx $d6
e70e  20 6c e5     jsr $e56c
e711  a4 d5        ldy $d5
e713  84 d3        sty $d3
e715  60           rts
-------------------------------------------------------
Vystup na obrazovku - soucast CHROUT
e716  48           pha
e717  85 d7        sta $d7
e719  8a           txa
e71a  48           pha
e71b  98           tya
e71c  48           pha
e71d  a9 00        lda #$00
e71f  85 d0        sta $d0
e721  a4 d3        ldy $d3
e723  a5 d7        lda $d7
e725  10 03        bpl $e72a
e727  4c d4 e7     jmp $e7d4
e72a  c9 0d        cmp #$0d; <RETURN>
e72c  d0 03        bne $e731
e72e  4c 91 e8     jmp $e891
e731  c9 20        cmp #$20; <SPACE>
e733  90 10        bcc $e745
e735  c9 60        cmp #$60; znaky do $60 ?
e737  90 04        bcc $e73d
e739  29 df        and #$df
e73b  d0 02        bne $e73f
e73d  29 3f        and #$3f
e73f  20 84 e6     jsr $e684
e742  4c 93 e6     jmp $e693
e745  a6 d8        ldx $d8
e747  f0 03        beq $e74c
e749  4c 97 e6     jmp $e697
e74c  c9 14        cmp #$14; <DEL>
e74e  d0 2e        bne $e77e
```

```
e750  98           tya
e751  d0 06       bne $e759
e753  20 01 e7    jsr $e701
e756  4c 73 e7    jmp $e773
e759  20 a1 e8    jsr $e8a1
e75c  88          dey
e75d  84 d3       sty $d3
e75f  20 24 ea    jsr $ea24
e762  c8          iny
e763  b1 d1       lda ($d1),y
e765  88          dey
e766  91 d1       sta ($d1),y
e768  c8          iny
e769  b1 f3       lda ($f3),y
e76b  88          dey
e76c  91 f3       sta ($f3),y
e76e  c8          iny
e76f  c4 d5       cpy $d5
e771  d0 ef       bne $e762
e773  a9 20       lda #$20
e775  91 d1       sta ($d1),y
e777  ad 86 02    lda $0286
e77a  91 f3       sta ($f3),y
e77c  10 4d       bpl $e7cb
e77e  a6 d4       ldx $d4
e780  f0 03       beq $e785
e782  4c 97 e6    jmp $e697
e785  c9 12       cmp #$12; <Rvs ON>
e787  d0 02       bne $e78b
e789  85 c7       sta $c7
e78b  c9 13       cmp #$13; <HOME>
e78d  d0 03       bne $e792
e78f  20 66 e5    jsr $e566
e792  c9 1d       cmp #$1d; 'Crsr vpravo'
e794  d0 17       bne $e7ad
e796  c8          iny
e797  20 b3 e8    jsr $e8b3
e79a  84 d3       sty $d3
e79c  88          dey
e79d  c4 d5       cpy $d5
e79f  90 09       bcc $e7aa
e7a1  c6 d6       dec $d6
e7a3  20 7c e8    jsr $e87c
e7a6  a0 00       ldy #$00
e7a8  84 d3       sty $d3
e7aa  4c a8 e6    jmp $e6a8
e7ad  c9 11       cmp #$11; 'Crsr dolu'
e7af  d0 1d       bne $e7ce
e7b1  18          clc
e7b2  98          tya
e7b3  69 28       adc #$28
e7b5  a8          tay
e7b6  e6 d6       inc $d6
e7b8  c5 d5       cmp $d5
e7ba  90 ec       bcc $e7a8
e7bc  f0 ea       beq $e7a8
e7be  c6 d6       dec $d6
e7c0  e9 28       sbc #$28
```

```
e7c2  90 04        bcc $e7c8
e7c4  85 d3        sta $d3
e7c6  d0 f8        bne $e7c0
e7c8  20 7c e8     jsr $e87c
e7cb  4c a8 e6     jmp $e6a8
e7ce  20 cb e8     jsr $e8cb; CHR$ kod barvy znaku?
e7d1  4c 44 ec     jmp $ec44; dekodovani zbylych zn.
--------------------------------------------------
Dekodovani nekterych CHR$ kodu (vetsich nez $7f)
e7d4  29 7f        and #$7f
e7d6  c9 7f        cmp #$7f
e7d8  d0 02        bne $e7dc
e7da  a9 5e        lda #$5e
e7dc  c9 20        cmp #$20; <SPACE>
e7de  90 03        bcc $e7e3
e7e0  4c 91 e6     jmp $e691
e7e3  c9 0d        cmp #$0d; 'SHIFT'+'RETURN'
e7e5  d0 03        bne $e7ea
e7e7  4c 91 e8     jmp $e891
e7ea  a6 d4        ldx $d4
e7ec  d0 3f        bne $e82d
e7ee  c9 14        cmp #$14; <INST>
e7f0  d0 37        bne $e829
e7f2  a4 d5        ldy $d5
e7f4  b1 d1        lda ($d1),y
e7f6  c9 20        cmp #$20
e7f8  d0 04        bne $e7fe
e7fa  c4 d3        cpy $d3
e7fc  d0 07        bne $e805
e7fe  c0 4f        cpy #$4f
e800  f0 24        beq $e826
e802  20 65 e9     jsr $e965
e805  a4 d5        ldy $d5
e807  20 24 ea     jsr $ea24
e80a  88           dey
e80b  b1 d1        lda ($d1),y
e80d  c8           iny
e80e  91 d1        sta ($d1),y
e810  88           dey
e811  b1 f3        lda ($f3),y
e813  c8           iny
e814  91 f3        sta ($f3),y
e816  88           dey
e817  c4 d3        cpy $d3
e819  d0 ef        bne $e80a
e81b  a9 20        lda #$20
e81d  91 d1        sta ($d1),y
e81f  ad 86 02     lda $0286
e822  91 f3        sta ($f3),y
e824  e6 d8        inc $d8
e826  4c a8 e6     jmp $e6a8
e829  a6 d8        ldx $d8
e82b  f0 05        beq $e832
e82d  09 40        ora #$40
e82f  4c 97 e6     jmp $e697
e832  c9 11        cmp #$11; 'Crsr nahoru'
e834  d0 16        bne $e84c
e836  a6 d6        ldx $d6
```

```
e838  f0 37       beq $e871
e83a  c6 d6       dec $d6
e83c  a5 d3       lda $d3
e83e  38          sec
e83f  e9 28       sbc #$28
e841  90 04       bcc $e847
e843  85 d3       sta $d3
e845  10 2a       bpl $e871
e847  20 5c e5    jsr $e55c
e84a  d0 25       bne $e871
e84c  c9 12       cmp #$12; <Rvs OFF>
e84e  d0 04       bne $e854
e850  a9 00       lda #$00
e852  85 c7       sta $c7
e854  c9 1d       cmp #$1d; 'Crsr vlevo'
e856  d0 12       bne $e86a
e858  98          tya
e859  f0 09       beq $e864
e85b  20 a1 e8    jsr $e8a1
e85e  88          dey
e85f  84 d3       sty $d3
e861  4c a8 e6    jmp $e6a8
e864  20 01 e7    jsr $e701
e867  4c a8 e6    jmp $e6a8
e86a  c9 13       cmp #$13; <CLR>
e86c  d0 06       bne $e874
e86e  20 44 e5    jsr $e544
e871  4c a8 e6    jmp $e6a8
e874  09 80       ora #$80
e876  20 cb e8    jsr $e8cb
e879  4c 4f ec    jmp $ec4f
--------------------------------------------------------------
e87c  46 c9       lsr $c9
e87e  a6 d6       ldx $d6
e880  e8          inx
e881  e0 19       cpx #$19
e883  d0 03       bne $e888
e885  20 ea e8    jsr $e8ea
e888  b5 d9       lda $d9,x
e88a  10 f4       bpl $e880
e88c  86 d6       stx $d6
e88e  4c 6c e5    jmp $e56c
e891  a2 00       ldx #$00
e893  86 d8       stx $d8
e895  86 c7       stx $c7
e897  86 d4       stx $d4
e899  86 d3       stx $d3
e89b  20 7c e8    jsr $e87c
e89e  4c a8 e6    jmp $e6a8
e8a1  a2 02       ldx #$02
e8a3  a9 00       lda #$00
e8a5  c5 d3       cmp $d3
e8a7  f0 07       beq $e8b0
e8a9  18          clc
e8aa  69 28       adc #$28
e8ac  ca          dex
e8ad  d0 f6       bne $e8a5
e8af  60          rts
```

```
e8b0  c6 d6           dec $d6
e8b2  60              rts
--------------------------------------------------------------------------------
e8b3  a2 02           ldx #$02
e8b5  a9 27           lda #$27
e8b7  c5 d3           cmp $d3
e8b9  f0 07           beq $e8c2
e8bb  18              clc
e8bc  69 28           adc #$28
e8be  ca              dex
e8bf  d0 f6           bne $e8b7
e8c1  60              rts
e8c2  a6 d6           ldx $d6
e8c4  e0 19           cpx #$19
e8c6  f0 02           beq $e8ca
e8c8  e6 d6           inc $d6
e8ca  60              rts
--------------------------------------------------------------------------------
Dekodovani CHR$ => Kod barvy + zapis
e8cb  a2 0f           ldx #$0f
e8cd  dd da e8        cmp $e8da,x
e8d0  f0 04           beq $e8d6
e8d2  ca              dex
e8d3  10 f8           bpl $e8cd
e8d5  60              rts
e8d6  8e 86 02        stx $0286
e8d9  60              rts
--------------------------------------------------------------------------------
e8da  90 05 1c 9f 9c 1e 1f 9e; CHR$ kody barev
e8e2  81 95 96 97 98 99 9a 9b
--------------------------------------------------------------------------------
e8ea  a5 ac           lda $ac; Lo pocatek progra-
e8ec  48              pha
e8ed  a5 ad           lda $ad; Hi mu pro LOAD apod.
e8ef  48              pha
e8f0  a5 ae           lda $ae; Lo konec progra-
e8f2  48              pha
e8f3  a5 af           lda $af; Hi mu pro LOAD apod.
e8f5  48              pha
e8f6  a2 ff           ldx #$ff
e8f8  c6 d6           dec $d6
e8fa  c6 c9           dec $c9
e8fc  ce a5 02        dec $02a5
e8ff  e8              inx
e900  20 f0 e9        jsr $e9f0
e903  e0 18           cpx #$18
e905  b0 0c           bcs $e913
e907  bd f1 ec        lda $ecf1,x
e90a  85 ac           sta $ac
e90c  b5 da           lda $da,x
e90e  20 c8 e9        jsr $e9c8
e911  30 ec           bmi $e8ff
e913  20 ff e9        jsr $e9ff
e916  a2 00           ldx #$00
e918  b5 d9           lda $d9,x
e91a  29 7f           and #$7f
e91c  b4 da           ldy $da,x
e91e  10 02           bpl $e922
```

```
e920  09 80     ora #$80
e922  95 d9     sta $d9,x
e924  e8        inx
e925  e0 18     cpx #$18
e927  d0 ef     bne $e918
e929  a5 f1     lda $f1
e92b  09 80     ora #$80
e92d  85 f1     sta $f1
e92f  a5 d9     lda $d9
e931  10 c3     bpl $e8f6
e933  e6 d6     inc $d6
e935  ee a5 02  inc $02a5
e938  a9 7f     lda #$7f
e93a  8d 00 dc  sta $dc00
e93d  ad 01 dc  lda $dc01
e940  c9 fb     cmp #$fb
e942  08        php
e943  a9 7f     lda #$7f
e945  8d 00 dc  sta $dc00
e948  28        plp
e949  d0 0b     bne $e956
e94b  a0 00     ldy #$00
e94d  ea        nop
e94e  ca        dex
e94f  d0 fc     bne $e94d
e951  88        dey
e952  d0 f9     bne $e94d
e954  84 c6     sty $c6
e956  a6 d6     ldx $d6
e958  68        pla
e959  85 af     sta $af
e95b  68        pla
e95c  85 ae     sta $ae
e95e  68        pla
e95f  85 ad     sta $ad
e961  68        pla
e962  85 ac     sta $ac
e964  60        rts
-------------------------------------------------
e965  a6 d6     ldx $d6; radek
e967  e8        inx
e968  b5 d9     lda $d9,x; najde prvni nepres-
e96a  10 fb     bpl $e967; tupny radek
e96c  8e a5 02  stx $02a5; rolovani obrazovky
e96f  e0 18     cpx #$18; 24-ty radek?
e971  f0 0e     beq $e981
e973  90 0c     bcc $e981
e975  20 ea e8  jsr $e8ea
e978  ae a5 02  ldx $02a5
e97b  ca        dex
e97c  c6 d6     dec $d6
e97e  4c da e6  jmp $e6da
e981  a5 ac     lda $ac
e983  48        pha
e984  a5 ad     lda $ad
e986  48        pha
e987  a5 ae     lda $ae
e989  48        pha
```

```
e98a  a5 af     lda $af
e98c  48        pha
e98d  a2 19     ldx #$19
e98f  ca        dex
e990  20 f0 e9  jsr $e9f0
e993  ec a5 02  cpx $02a5
e996  90 0e     bcc $e9a6
e998  f0 0c     beq $e9a6
e99a  bd ef ec  lda $ecef,x
e99d  85 ac     sta $ac
e99f  b5 d8     lda $d8,x
e9a1  20 c8 e9  jsr $e9c8
e9a4  30 e9     bmi $e98f
e9a6  20 ff e9  jsr $e9ff
e9a9  a2 17     ldx #$17
e9ab  ec a5 02  cpx $02a5
e9ae  90 0f     bcc $e9bf
e9b0  b5 da     lda $da,x
e9b2  29 7f     and #$7f
e9b4  b4 d9     ldy $d9,x
e9b6  10 02     bpl $e9ba
e9b8  09 80     ora #$80
e9ba  95 da     sta $da,x
e9bc  ca        dex
e9bd  d0 ec     bne $e9ab
e9bf  ae a5 02  ldx $02a5
e9c2  20 da e6  jsr $e6da
e9c5  4c 58 e9  jmp $e958
```

```
e9c8  29 03     and #$03
e9ca  0d 88 02  ora $0288
e9cd  85 ad     sta $ad
e9cf  20 e0 e9  jsr $e9e0
e9d2  a0 27     ldy #$27
e9d4  b1 ac     lda ($ac),y
e9d6  91 d1     sta ($d1),y
e9d8  b1 ae     lda ($ae),y
e9da  91 f3     sta ($f3),y
e9dc  88        dey
e9dd  10 f5     bpl $e9d4
e9df  60        rts
```

```
e9e0  20 24 ea  jsr $ea24
e9e3  a5 ac     lda $ac
e9e5  85 ae     sta $ae
e9e7  a5 ad     lda $ad
e9e9  29 03     and #$03
e9eb  09 d8     ora #$d8
e9ed  85 af     sta $af
e9ef  60        rts
```

Nastaveni aktualni radky

```
e9f0  bd f0 ec  lda $ecf0,x
e9f3  85 d1     sta $d1
e9f5  b5 d9     lda $d9,x
e9f7  29 03     and #$03
e9f9  0d 88 02  ora $0288
e9fc  85 d2     sta $d2
```

```
e9fe 60          rts
-------------------------------------------------
e9ff a0 27       ldy #$27
ea01 20 f0 e9    jsr $e9f0; akt. radka
ea04 20 24 ea    jsr $ea24; zn. RAM => bar. RAM
ea07 20 da e4    jsr $e4da; zapis do bar. RAM
ea0a a9 20       lda #$20; kod 'SPACE'
ea0c 91 d1       sta ($d1),y; vymazani radky
ea0e 88          dey
ea0f 10 f6       bpl $ea07
ea11 60          rts
-------------------------------------------------
Zapis znaku z A a barvy z X do pameti
ea12 ea          nop
ea13 a8          tay
ea14 a9 02       lda #$02
ea16 85 cd       sta $cd
ea18 20 24 ea    jsr $ea24
ea1b 98          tya
ea1c a4 d3       ldy $d3
ea1e 91 d1       sta ($d1),y
ea20 8a          txa
ea21 91 f3       sta ($f3),y
ea23 60          rts
-------------------------------------------------
Prepocet adresy znak. RAM na adresu bar. RAM
ea24 a5 d1       lda $d1
ea26 85 f3       sta $f3
ea28 a5 d2       lda $d2
ea2a 29 03       and #$03
ea2c 09 d8       ora #$d8
ea2e 85 f4       sta $f4
ea30 60          rts
-------------------------------------------------
Standardni preruseni - IRQ
ea31 20 ea ff    jsr $ffea; posunuti vnitr. hodin
ea34 a5 cc       lda $cc; obsluha blikani kur-
ea36 d0 29       bne $ea61; zoru
ea38 c6 cd       dec $cd; zpozdeni
ea3a d0 25       bne $ea61
obsluha kurzoru
ea3c a9 14       lda #$14
ea3e 85 cd       sta $cd
ea40 a4 d3       ldy $d3; sloupec
ea42 46 cf       lsr $cf
ea44 ae 87 02    ldx $0287; barva
ea47 b1 d1       lda ($d1),y
ea49 b0 11       bcs $ea5c
ea4b e6 cf       inc $cf
ea4d 85 ce       sta $ce
ea4f 20 24 ea    jsr $ea24
ea52 b1 f3       lda ($f3),y
ea54 8d 87 02    sta $0287
ea57 ae 86 02    ldx $0286
ea5a a5 ce       lda $ce
ea5c 49 80       eor #$80
ea5e 20 1c ea    jsr $ea1c
ea61 a5 01       lda $01
```

```
ea63 29 10        and #$10; motor mgf.
ea65 f0 0a        beq $ea71
ea67 a0 00        ldy #$00
ea69 84 c0        sty $c0; motor mgf.
ea6b a5 01        lda $01
ea6d 09 20        ora #$20
ea6f d0 08        bne $ea79
ea71 a5 c0        lda $c0
ea73 d0 06        bne $ea7b
ea75 a5 01        lda $01
ea77 29 1f        and #$1f
ea79 85 01        sta $01
ea7b 20 87 ea     jsr $ea87; rutina SCNKEY
ea7e ad 0d dc     lda $dc0d
ea81 68           pla ; Konec IRQ
ea82 a8           tay
ea83 68           pla
ea84 aa           tax
ea85 68           pla
ea86 40           rti
-------------------------------------------------------
SCNKEY
ea87 a9 00        lda #$00
ea89 8d 8d 02     sta $028d; ukaz. C=;SHIFT apod.
ea8c a0 40        ldy #$40
ea8e 84 cb        sty $cb; kod stiskn. klavesy
ea90 8d 00 dc     sta $dc00; radek klavesnice
ea93 ae 01 dc     ldx $dc01; sloupec-odezva
ea96 e0 ff        cpx #$ff
ea98 f0 61        beq $eafb
ea9a a8           tay
ea9b a9 81        lda #$81; Lo standardni deko-
ea9d 85 f5        sta $f5
ea9f a9 eb        lda #$eb; Hi dovaci tab. klaves.
eaa1 85 f6        sta $f6
cteni stisknute klavesy a jeji dekodovani
eaa3 a9 fe        lda #$fe
eaa5 8d 00 dc     sta $dc00
eaa8 a2 08        ldx #$08
eaaa 48           pha
eaab ad 01 dc     lda $dc01
eaae cd 01 dc     cmp $dc01
eab1 d0 f8        bne $eaab
eab3 4a           lsr
eab4 b0 16        bcs $eacc
eab6 48           pha
eab7 b1 f5        lda ($f5),y; cteni z tabulky
eab9 c9 05        cmp #$05
eabb b0 0c        bcs $eac9
eabd c9 03        cmp #$03
eabf f0 08        beq $eac9
eac1 0d 8d 02     ora $028d
eac4 8d 8d 02     sta $028d
eac7 10 02        bpl $eacb
eac9 84 cb        sty $cb; kod stiskn. klavesy
eacb 68           pla
eacc c8           iny
eacd c0 41        cpy #$41
```

```
ea.cf  b0 0b        bcs $ea.dc
ea.d1  ca           dex
ea.d2  d0 df        bne $eab3
ea.d4  38           sec
ea.d5  68           Pla
ea.d6  2a           rol
ea.d7  8d 00 dc     sta $dc00
ea.da  d0 cc        bne $ea.a8

ea.dc  68           Pla
ea.dd  6c 8f 02     jmP ($028f); dekodovaci rutina
-------------------------------------------------------
ea.e0  a.4 cb       ldY $cb; Kod znaku Po Prvnim
ea.e2  b1 f5        lda ($f5),Y; dekodovani; tabulka
ea.e4  a.a          ta.x
ea.e5  c4 c5        cpY $c5; Kod Predchoziho znaku
ea.e7  f0 07        beq $ea.f0
ea.e9  a.0 10       ldY #$10
ea.eb  8c 8c 02     sty $028c
ea.ee  d0 36        bne $eb26
ea.f0  29 7f        and #$7f
ea.f2  2c 8a. 02    bit $028a.
ea.f5  30 16        bmi $eb0d
ea.f7  70 49        bvs $eb42
ea.f9  c9 7f        cmP #$7f
ea.fb  f0 29        beq $eb26
ea.fd  c9 14        cmP #$14
ea.ff  f0 0c        beq $eb0d
eb01  c9 20         cmP #$20
eb03  f0 08         beq $eb0d
eb05  c9 1d         cmP #$1d
eb07  f0 04         beq $eb0d
eb09  c9 11         cmP #$11
eb0b  d0 35         bne $eb42
eb0d  a.c 8c 02     ldY $028c
eb10  f0 05         beq $eb17
eb12  ce 8c 02      dec $028c
eb15  d0 2b         bne $eb42
eb17  ce 8b 02      dec $028b
eb1a. d0 26         bne $eb42
eb1c  a0 04         ldY #$04
eb1e  8c 8b 02      sty $028b
eb21  a.4 c6        ldY $c6; Pocet znaku v buffru
eb23  88            dey
eb24  10 1c         bP1 $eb42
eb26  a.4 cb        ldY $cb
eb28  84 c5         stY $c5
eb2a. a.c 8d 02     ldY $028d
eb2d  8c 8e 02      sty $028e
eb30  e0 ff         cPx #$ff
eb32  f0 0e         beq $eb42
eb34  8a.           txa
eb35  a.6 c6        ldx $c6
eb37  ec 89 02      cPx $0289
eb3a. b0 06         bcs $eb42
eb3c  9d 77 02      sta $0277,x
eb3f  e8            inx
eb40  86 c6         stx $c6
```

```
eb42 a9 7f      lda #$7f
eb44 8d 00 dc   sta $dc00
eb47 60         rts
-------------------------------------------------------
eb48 ad 8d 02   lda $028d; ukaz. SHIFT apod.
eb4b c9 03      cmp #$03
eb4d d0 15      bne $eb64
eb4f cd 8e 02   cmp $028e; predch.st.uk. SHIFT...
eb52 f0 ee      beq $eb42
eb54 ad 91 02   lda $0291; blok. prep. zn. sad
eb57 30 1d      bmi $eb76
eb59 ad 18 d0   lda $d018; prepnuti do druhe
eb5c 49 02      eor #$02; zn. sady
eb5e 8d 18 d0   sta $d018
eb61 4c 76 eb   jmp $eb76
eb64 0a         asl
eb65 c9 08      cmp #$08
eb67 90 02      bcc $eb6b
eb69 a9 06      lda #$06; zakladni tabulka
eb6b aa         tax
eb6c bd 79 eb   lda $eb79,x; prenos pocatku deko-
eb6f 85 f5      sta $f5
eb71 bd 7a eb   lda $eb7a,x; dovaci tabulky CHR$
eb74 85 f6      sta $f6
eb76 4c e0 ea   jmp $eae0; dekodovani nekt. kodu
-------------------------------------------------------
Pocatky dekodovacich tabulek pri soucasnem stis-
ku zn. klavesy a SHIFT;C=;Control a niceho
eb79 81 eb c2 eb 03 ec 78 ec
-------------------------------------------------------
eb81  14 0d 1d 88 85 86 87 11; Zakl. dekod. tabulka
eb89  33 57 41 34 5a 53 45 01
eb91  35 52 44 36 43 46 54 58
eb99  37 59 47 38 42 48 55 56
eba1  39 49 4a 30 4d 4b 4f 4e
eba9  2b 50 4c 2d 2e 3a 40 2c
ebb1  5c 2a 3b 13 01 3d 5e 2f
ebb9  31 5f 04 32 20 02 51 03
ebc1  ff 94 8d 9d 8c 89 8a 8b; Druha dek. tabulka
ebc9  91 23 d7 c1 24 da d3 c5
ebd1  01 25 d2 c4 26 c3 c6 d4
ebd9  d8 27 d9 c7 28 c2 c8 d5
ebe1  d6 29 c9 ca 30 cd cb cf
ebe9  ce db d0 cc dd 3e 5b ba
ebf1  3c a9 c0 5d 93 01 3d de
ebf9  3f 21 5f 04 22 a0 02 d1
ec01  83 ff 94 8d 9d 8c 89 8a; treti dek. tabulka
ec09  8b 91 96 b3 b0 97 ad ae
ec11  b1 01 98 b2 ac 99 bc bb
ec19  a3 bd 9a b7 a5 9b bf b4
ec21  b8 be 29 a2 b5 30 a7 a1
ec29  b9 aa a6 af b6 dc 3e 5b
ec31  a4 3c a8 df 5d 93 01 3d
ec39  de 3f 81 5f 04 95 a0 02
ec41  ab 83 ff
-------------------------------------------------------
Dalsi cast dekodovani CHR$
ec44 c9 0e          cmp #$0e; velka./mala pismena ?
```

```
ec46  d0 07        bne  $ec4f
ec48  ad 18 d0     lda  $d018
ec4b  09 02        ora  #$02
ec4d  d0 09        bne  $ec58
ec4f  c9 8e        cmp  #$8e; velka pismena a gr. ?
ec51  d0 0b        bne  $ec5e
ec53  ad 18 d0     lda  $d018
ec56  29 fd        and  #$fd
ec58  8d 18 d0     sta  $d018
ec5b  4c a8 e6     jmp  $e6a8
ec5e  c9 08        cmp  #$08
ec60  d0 07        bne  $ec69
ec62  a9 80        lda  #$80
ec64  0d 91 02     ora  $0291
ec67  30 09        bmi  $ec72
ec69  c9 09        cmp  #$09
ec6b  d0 ee        bne  $ec5b
ec6d  a9 7f        lda  #$7f
ec6f  2d 91 02     and  $0291
ec72  8d 91 02     sta  $0291
ec75  4c a8 e6     jmp  $e6a8
--------------------------------------------------------
Tabulka pro prvni dekodovani klavesnice
ec78  ff ff ff ff ff ff ff ff
ec80  1c 17 01 9f 1a 13 05 ff
ec88  9c 12 04 1e 03 06 14 18
ec90  1f 19 07 9e 02 08 15 16
ec98  12 09 0a 92 0d 0b 0f 0e
eca0  ff 10 0c ff ff 1b 00 ff
eca8  1c ff 1d ff ff 1f 1e ff
ecb0  90 06 ff 05 ff ff 11 ff
ecb8  ff 00 00 00 00 00 00 00; od $ecb9 data pro
ecc0  00 00 00 00 00 00 00 00; VIC
ecc8  00 00 9b 37 00 00 00 08
ecd0  00 14 0f 00 00 00 00 00
ecd8  00 0e 06 01 02 03 04 00
ece0  01 02 03 04 05 06 07
--------------------------------------------------------
Data pro naplneni buffru prikazy LOAD a RUN
(po stisku 'SHIFT' + 'RUN/STOP')
ece7  4c 4f 41 44 0d 52 55 4e; load run
ecef  0d
--------------------------------------------------------
ecf0  00 28 50 78 a0 c8 f0 18; Data
ecf8  40 68 90 b8 e0 08 30 58
ed00  80 a8 d0 f8 20 48 70 98
ed08  c0
--------------------------------------------------------
TALK
ed09  09 40        ora  #$40
LISTEN (od $ed0c)
ed0b  2c 09 20     bit  $2009
ed0e  20 a4 f0     jsr  $f0a4; obsluha RS-232
ed11  48           pha
ed12  24 94        bit  $94
ed14  10 0a        bpl  $ed20
ed16  38           sec
ed17  66 a3        ror  $a3
```

```
ed19  20 40 ed     jsr $ed40
ed1c  46 94        lsr $94
ed1e  46 a3        lsr $a3
ed20  68           pla
ed21  85 95        sta $95
ed23  78           sei
ed24  20 97 ee     jsr $ee97
ed27  c9 3f        cmp #$3f
ed29  d0 03        bne $ed2e
ed2b  20 85 ee     jsr $ee85
ed2e  ad 00 dd     lda $dd00
ed31  09 08        ora #$08
ed33  8d 00 dd     sta $dd00
ed36  78           sei
ed37  20 8e ee     jsr $ee8e
ed3a  20 97 ee     jsr $ee97
ed3d  20 b3 ee     jsr $eeb3
ed40  78           sei
ed41  20 97 ee     jsr $ee97
ed44  20 a9 ee     jsr $eea9
ed47  b0 64        bcs $edad
ed49  20 85 ee     jsr $ee85
ed4c  24 a3        bit $a3
ed4e  10 0a        bpl $ed5a
ed50  20 a9 ee     jsr $eea9
ed53  90 fb        bcc $ed50
ed55  20 a9 ee     jsr $eea9
ed58  b0 fb        bcs $ed55
ed5a  20 a9 ee     jsr $eea9
ed5d  90 fb        bcc $ed5a
ed5f  20 8e ee     jsr $ee8e
ed62  a9 08        lda #$08
ed64  85 a5        sta $a5
ed66  ad 00 dd     lda $dd00
ed69  cd 00 dd     cmp $dd00
ed6c  d0 f8        bne $ed66
ed6e  0a           asl
ed6f  90 3f        bcc $edb0
ed71  66 95        ror $95
ed73  b0 05        bcs $ed7a
ed75  20 a0 ee     jsr $eea0
ed78  d0 03        bne $ed7d
ed7a  20 97 ee     jsr $ee97
ed7d  20 85 ee     jsr $ee85
ed80  ea           nop
ed81  ea           nop
ed82  ea           nop
ed83  ea           nop
ed84  ad 00 dd     lda $dd00
ed87  29 df        and #$df
ed89  09 10        ora #$10
ed8b  8d 00 dd     sta $dd00
ed8e  c6 a5        dec $a5
ed90  d0 d4        bne $ed66
ed92  a9 04        lda #$04
ed94  8d 07 dc     sta $dc07
ed97  a9 19        lda #$19
ed99  8d 0f dc     sta $dc0f
```

```
ed9c  ad 0d dc     lda  $dc0d
ed9f  ad 0d dc     lda  $dc0d
eda2  29 02        and  #$02
eda4  d0 0a        bne  $edb0
eda6  20 a9 ee     jsr  $eea9
eda9  b0 f4        bcs  $ed9f
edab  58           cli
edac  60           rts
-------------------------------------------------------------
edad  a9 80        lda  #$80
edaf  2c a9 03     bit  $03a9
edb2  20 1c fe     jsr  $fe1c
edb5  58           cli
edb6  18           clc
edb7  90 4a        bcc  $ee03
SECOND
edb9  85 95        sta  $95
edbb  20 36 ed     jsr  $ed36
edbe  ad 00 dd     lda  $dd00
edc1  29 f7        and  #$f7
edc3  8d 00 dd     sta  $dd00
edc6  60           rts
-------------------------------------------------------------
TKSA
edc7  85 95        sta  $95
edc9  20 36 ed     jsr  $ed36
edcc  78           sei
edcd  20 a0 ee     jsr  $eea0
edd0  20 be ed     jsr  $edbe
edd3  20 85 ee     jsr  $ee85
edd6  20 a9 ee     jsr  $eea9
edd9  30 fb        bmi  $edd6
eddb  58           cli
eddc  60           rts
-------------------------------------------------------------
CIOUT
eddd  24 94        bit  $94
eddf  30 05        bmi  $ede6
ede1  38           sec
ede2  66 94        ror  $94
ede4  d0 05        bne  $edeb
ede6  48           pha
ede7  20 40 ed     jsr  $ed40
edea  68           pla
edeb  85 95        sta  $95
eded  18           clc
edee  60           rts
-------------------------------------------------------------
UNTLK
edef  78           sei
edf0  20 8e ee     jsr  $ee8e
edf3  ad 00 dd     lda  $dd00
edf6  09 08        ora  #$08
edf8  8d 00 dd     sta  $dd00
edfb  a9 5f        lda  #$5f
UNLSN (cod $edfe)
edfd  2c a9 3f     bit  $3fa9
ee00  20 11 ed     jsr  $ed11
```

```
ee03  20 be ed     jsr $edbe
ee06  8a           txa
ee07  a2 0a        ldx #$0a
ee09  ca           dex
ee0a  d0 fd        bne $ee09
ee0c  aa           tax
ee0d  20 85 ee     jsr $ee85
ee10  4c 97 ee     jmp $ee97
-------------------------------------------------------------
ACPTR
ee13  78           sei
ee14  a9 00        lda #$00
ee16  85 a5        sta $a5
ee18  20 85 ee     jsr $ee85
ee1b  20 a9 ee     jsr $eea9
ee1e  10 fb        bpl $ee1b
ee20  a9 01        lda #$01
ee22  8d 07 dc     sta $dc07
ee25  a9 19        lda #$19
ee27  8d 0f dc     sta $dc0f
ee2a  20 97 ee     jsr $ee97
ee2d  ad 0d dc     lda $dc0d
ee30  ad 0d dc     lda $dc0d
ee33  29 02        and #$02
ee35  d0 07        bne $ee3e
ee37  20 a9 ee     jsr $eea9
ee3a  30 f4        bmi $ee30
ee3c  10 18        bpl $ee56
ee3e  a5 a5        lda $a5
ee40  f0 05        beq $ee47
ee42  a9 02        lda #$02
ee44  4c b2 ed     jmp $edb2
-------------------------------------------------------------
ee47  20 a0 ee     jsr $eea0
ee4a  20 85 ee     jsr $ee85
ee4d  a9 40        lda #$40
ee4f  20 1c fe     jsr $fe1c
ee52  e6 a5        inc $a5
ee54  d0 ca        bne $ee20
ee56  a9 08        lda #$08
ee58  85 a5        sta $a5
ee5a  ad 00 dd     lda $dd00
ee5d  cd 00 dd     cmp $dd00
ee60  d0 f8        bne $ee5a
ee62  0a           asl
ee63  10 f5        bpl $ee5a
ee65  66 a4        ror $a4
ee67  ad 00 dd     lda $dd00
ee6a  cd 00 dd     cmp $dd00
ee6d  d0 f8        bne $ee67
ee6f  0a           asl
ee70  30 f5        bmi $ee67
ee72  c6 a5        dec $a5
ee74  d0 e4        bne $ee5a
ee76  20 a0 ee     jsr $eea0
ee79  24 90        bit $90
ee7b  50 03        bvc $ee80
ee7d  20 06 ee     jsr $ee06
```

```
ee80  a5 a4        lda $a4
ee82  58           cli
ee83  18           clc
ee84  60           rts
-------------------------------------------------
Tabulka pro CIA#2
ee85  ad 00 dd     lda $dd00
ee88  29 ef        and #$ef
ee8a  8d 00 dd     sta $dd00
ee8d  60           rts
ee8e  ad 00 dd     lda $dd00
ee91  09 10        ora #$10
ee93  8d 00 dd     sta $dd00
ee96  60           rts
ee97  ad 00 dd     lda $dd00
ee9a  29 df        and #$df
ee9c  8d 00 dd     sta $dd00
ee9f  60           rts
eea0  ad 00 dd     lda $dd00
eea3  09 20        ora #$20
eea5  8d 00 dd     sta $dd00
eea8  60           rts
eea9  ad 00 dd     lda $dd00
eeac  cd 00 dd     cmp $dd00
eeaf  d0 f8        bne $eea9
eeb1  0a           asl
eeb2  60           rts
-------------------------------------------------
eeb3  8a           txa ; zpozdeni
eeb4  a2 b8        ldx #$b8
eeb6  ca           dex
eeb7  d0 fd        bne $eeb6
eeb9  aa           tax
eeba  60           rts
-------------------------------------------------
Obsluha RS-232
eebb  a5 b4        lda $b4
eebd  f0 47        beq $ef06
eebf  30 3f        bmi $ef00
eec1  46 b6        lsr $b6
eec3  a2 00        ldx #$00
eec5  90 01        bcc $eec8
eec7  ca           dex
eec8  8a           txa
eec9  45 bd        eor $bd
eecb  85 bd        sta $bd
eecd  c6 b4        dec $b4
eecf  f0 06        beq $eed7
eed1  8a           txa
eed2  29 04        and #$04
eed4  85 b5        sta $b5; Konec
eed6  60           rts
-------------------------------------------------
eed7  a9 20        lda #$20
eed9  2c 94 02     bit $0294
eedc  f0 14        beq $eef2
eede  30 1c        bmi $eefc
eee0  70 14        bvs $eef6
```

```
eee2  a5 bd       lda $bd
eee4  d0 01       bne $eee7
eee6  ca          dex
eee7  c6 b4       dec $b4
eee9  ad 93 02    lda $0293
eeec  10 e3       bpl $eed1
eeee  c6 b4       dec $b4
eef0  d0 df       bne $eed1
eef2  e6 b4       inc $b4
eef4  d0 f0       bne $eee6
eef6  a5 bd       lda $bd
eef8  f0 ed       beq $eee7
eefa  d0 ea       bne $eee6
eefc  70 e9       bvs $eee7
eefe  50 e6       bvc $eee6
-------------------------------------------------------------
ef00  e6 b4       inc $b4
ef02  a2 ff       ldx #$ff
ef04  d0 cb       bne $eed1
ef06  ad 94 02    lda $0294
ef09  4a          lsr
ef0a  90 07       bcc $ef13
ef0c  2c 01 dd    bit $dd01
ef0f  10 1d       bpl $ef2e
ef11  50 1e       bvc $ef31
ef13  a9 00       lda #$00
ef15  85 bd       sta $bd
ef17  85 b5       sta $b5
ef19  ae 98 02    ldx $0298; Pocet b pred vyslanim
ef1c  86 b4       stx $b4
ef1e  ac 9d 02    ldy $029d; Poc.v buffru vysl.zn.
ef21  cc 9e 02    cpy $029e; Konec v b. vysl. zn.
ef24  f0 13       beq $ef39
ef26  b1 f9       lda ($f9),y; cteni znaku z buffru
ef28  85 b6       sta $b6
ef2a  ee 9d 02    inc $029d
ef2d  60          rts
-------------------------------------------------------------
ef2e  a9 40       lda #$40
ef30  2c a9 10    bit $10a9
ef33  0d 97 02    ora $0297
ef36  8d 97 02    sta $0297; stavovy rg. RS-232
ef39  a9 01       lda #$01
ef3b  8d 0d dd    sta $dd0d
ef3e  4d a1 02    eor $02a1
ef41  09 80       ora #$80
ef43  8d a1 02    sta $02a1; NMI pro CIA#2
ef46  8d 0d dd    sta $dd0d
ef49  60          rts
-------------------------------------------------------------
ef4a  a2 09       ldx #$09
ef4c  a9 20       lda #$20
ef4e  2c 93 02    bit $0293
ef51  f0 01       beq $ef54
ef53  ca          dex
ef54  50 02       bvc $ef58
ef56  ca          dex
ef57  ca          dex
```

```
ef58 60                 rts
-------------------------------------------------------
ef59 a6 a9              ldx $a9; start
ef5b d0 33              bne $ef90
ef5d c6 a8              dec $a8
ef5f f0 36              beq $ef97
ef61 30 0d              bmi $ef70
ef63 a5 a7              lda $a7
ef65 45 ab              eor $ab
ef67 85 ab              sta $ab
ef69 46 a7              lsr $a7
ef6b 66 aa              ror $aa
ef6d 60                 rts
-------------------------------------------------------
ef6e c6 a8              dec $a8
ef70 a5 a7              lda $a7
ef72 f0 67              beq $efdb
ef74 ad 93 02           lda $0293
ef77 0a                 asl
ef78 a9 01              lda #$01
ef7a 65 a8              adc $a8
ef7c d0 ef              bne $ef6d
ef7e a9 90              lda #$90
ef80 8d 0d dd           sta $dd0d
ef83 0d a1 02           ora $02a1
ef86 8d a1 02           sta $02a1
ef89 85 a9              sta $a9
ef8b a9 02              lda #$02
ef8d 4c 3b ef           jmp $ef3b
ef90 a5 a7              lda $a7
ef92 d0 ea              bne $ef7e
ef94 4c d3 e4           jmp $e4d3
ef97 ac 9b 02           ldy $029b; Konec v buffru od. zn.
ef9a c8                 iny
ef9b cc 9c 02           cpy $029c; Poc. v b. odebir. zn.
ef9e f0 2a              beq $efca
efa0 8c 9b 02           sty $029b
efa3 88                 dey
efa4 a5 aa              lda $aa
efa6 ae 98 02           ldx $0298; Pocet bitu pred vysl.
efa9 e0 09              cpx #$09
efab f0 04              beq $efb1
efad 4a                 lsr
efae e8                 inx
efaf d0 f8              bne $efa9
efb1 91 f7              sta ($f7),y; zapis do vst. buffru
efb3 a9 20              lda #$20
efb5 2c 94 02           bit $0294
efb8 f0 b4              beq $ef6e
efba 30 b1              bmi $ef6d
efbc a5 a7              lda $a7
efbe 45 ab              eor $ab
efc0 f0 03              beq $efc5
efc2 70 a9              bvs $ef6d
efc4 2c 50 a6           bit $a650
efc7 a9 01              lda #$01; Kody pro stav. rs.
efc9 2c a9 04           bit $04a9
efcc 2c a9 80           bit $80a9
```

```
efcf 2c a9 02     bit $02a9
efd2 0d 97 02     ora $0297
efd5 8d 97 02     sta $0297; stavovy registr RS-232
efd8 4c 7e ef     jmp $ef7e
-------------------------------------------------------
efdb a5 aa        lda $aa
efdd d0 f1        bne $efd0
efdf f0 ec        beq $efcd
efe1 85 9a        sta $9a; Kod periferie pro vystup
efe3 ad 94 02     lda $0294
efe6 4a           lsr
efe7 90 29        bcc $f012
efe9 a9 02        lda #$02
efeb 2c 01 dd     bit $dd01
efee 10 1d        bpl $f00d
eff0 d0 20        bne $f012
eff2 ad a1 02     lda $02a1; ukazatel NMI pro CIA#2
eff5 29 02        and #$02
eff7 d0 f9        bne $eff2
eff9 2c 01 dd     bit $dd01
effc 70 fb        bvs $eff9
effe ad 01 dd     lda $dd01
f001 09 02        ora #$02
f003 8d 01 dd     sta $dd01
f006 2c 01 dd     bit $dd01
f009 70 07        bvs $f012
f00b 30 f9        bmi $f006
f00d a9 40        lda #$40
f00f 8d 97 02     sta $0297; stavovy registr RS-232
f012 18           clc
f013 60           rts
-------------------------------------------------------
f014 20 28 f0     jsr $f028
f017 ac 9e 02     ldy $029e; Konec v buffru vys.zn.
f01a c8           iny
f01b cc 9d 02     cpy $029d; Poc. v buffru vys. zn.
f01e f0 f4        beq $f014
f020 8c 9e 02     sty $029e
f023 88           dey
f024 a5 9e        lda $9e
f026 91 f9        sta ($f9),y; zapis do vyst. buf.
f028 ad a1 02     lda $02a1
f02b 4a           lsr
f02c b0 1e        bcs $f04c
f02e a9 10        lda #$10
f030 8d 0e dd     sta $dd0e
f033 ad 99 02     lda $0299; Lo
f036 8d 04 dd     sta $dd04
f039 ad 9a 02     lda $029a; Hi casova konstanta
f03c 8d 05 dd     sta $dd05
f03f a9 81        lda #$81
f041 20 3b ef     jsr $ef3b
f044 20 06 ef     jsr $ef06
f047 a9 11        lda #$11
f049 8d 0e dd     sta $dd0e
f04c 60           rts
-------------------------------------------------------
f04d 85 99        sta $99
```

```
f04f  ad 94 02     lda $0294
f052  4a           lsr
f053  90 28        bcc $f07d
f055  29 08        and #$08
f057  f0 24        beq $f07d
f059  a9 02        lda #$02
f05b  2c 01 dd     bit $dd01
f05e  10 ad        bpl $f00d
f060  f0 22        beq $f084
f062  ad a1 02     lda $02a1
f065  4a           lsr
f066  b0 fa        bcs $f062
f068  ad 01 dd     lda $dd01
f06b  29 fd        and #$fd
f06d  8d 01 dd     sta $dd01
f070  ad 01 dd     lda $dd01
f073  29 04        and #$04
f075  f0 f9        beq $f070
f077  a9 90        lda #$90
f079  18           clc
f07a  4c 3b ef     jmp $ef3b
f07d  ad a1 02     lda $02a1
f080  29 12        and #$12
f082  f0 f3        beq $f077
f084  18           clc
f085  60           rts
```

```
f086  ad 97 02     lda $0297; stavovy registr RS-232
f089  ac 9c 02     ldy $029c; Poc. v b. odeb. zn.
f08c  cc 9b 02     cpy $029b; Konec v b. odebir. zn.
f08f  f0 0b        beq $f09c
f091  29 f7        and #$f7
f093  8d 97 02     sta $0297
f096  b1 f7        lda ($f7),y; cteni ze vstup. buf.
f098  ee 9c 02     inc $029c
f09b  60           rts
f09c  09 08        ora #$08
f09e  8d 97 02     sta $0297
f0a1  a9 00        lda #$00
f0a3  60           rts
```

```
f0a4  48           pha
f0a5  ad a1 02     lda $02a1; NMI pro CIA#2
f0a8  f0 11        beq $f0bb
f0aa  ad a1 02     lda $02a1; cekani
f0ad  29 03        and #$03
f0af  d0 f9        bne $f0aa
f0b1  a9 10        lda #$10
f0b3  8d 0d dd     sta $dd0d
f0b6  a9 00        lda #$00
f0b8  8d a1 02     sta $02a1
f0bb  68           pla
f0bc  60           rts
```

```
Data hlasek
f0bd 0d 49 2f 4f 20 45 52 52; .i/o err
f0c5 4f 52 20 a3 0d 53 45 41; or   sea
f0cd 52 43 48 49 4e 47 a0 46; rching f
```

```
f0d5  4f 52 a0 0d 50 52 45 53;  or  pres
f0dd  53 20 50 4c 41 59 20 4f;  s play o
f0e5  4e 20 54 41 50 c5 50 52;  n tapEPr
f0ed  45 53 53 20 52 45 43 4f;  ess reco
f0f5  52 44 20 26 20 50 4c 41;  rd & pla
f0fd  59 20 4f 4e 20 54 41 50;  y on tap
f105  c5 0d 4c 4f 41 44 49 4e;  E loadin
f10d  c7 0d 53 41 56 49 4e 47;  G saving
f115  a0 0d 56 45 52 49 46 59;  . verifY
f11d  49 4e c7 0d 46 4f 55 4e;  ng  foun
f125  44 a0 0d 4f 4b 8d       ;  d  oK
```

```
f12b  24 9d       bit $9d
f12d  10 0d       bpl $f13c
```

```
Cast hlasky (v Y je relativni adresa pocatku)
f12f  b9 bd f0    lda $f0bd,y
f132  08          php
f133  29 7f       and #$7f
f135  20 d2 ff    jsr $ffd2
f138  c8          iny
f139  28          plp
f13a  10 f3       bpl $f12f
f13c  18          clc
f13d  60          rts
```

```
GETIN
f13e  a5 99       lda $99
f140  d0 08       bne $f14a
f142  a5 c6       lda $c6
f144  f0 0f       beq $f155
f146  78          sei
f147  4c b4 e5    jmp $e5b4
f14a  c9 02       cmp #$02
f14c  d0 18       bne $f166
f14e  84 97       sty $97
f150  20 86 f0    jsr $f086
f153  a4 97       ldy $97
f155  18          clc
f156  60          rts
```

```
CHRIN
f157  a5 99       lda $99; Kod vstup. periferie
f159  d0 0b       bne $f166
f15b  a5 d3       lda $d3; vstup z klavesnice
f15d  85 ca       sta $ca
f15f  a5 d6       lda $d6
f161  85 c9       sta $c9
f163  4c 32 e6    jmp $e632
f166  c9 03       cmp #$03
f168  d0 09       bne $f173
f16a  85 d0       sta $d0; vstup z obraz. RAM
f16c  a5 d5       lda $d5
f16e  85 c8       sta $c8
f170  4c 32 e6    jmp $e632
f173  b0 38       bcs $f1ad
f175  c9 02       cmp #$02
f177  f0 3f       beq $f1b8
```

```
f179  86 97        stx $97; cteni znaku z m9f.
f17b  20 99 f1     jsr $f199
f17e  b0 16        bcs $f196
f180  48           pha
f181  20 99 f1     jsr $f199
f184  b0 0d        bcs $f193
f186  d0 05        bne $f18d
f188  a9 40        lda #$40
f18a  20 1c fe     jsr $fe1c
f18d  c6 a6        dec $a6
f18f  a6 97        ldx $97
f191  68           pla
f192  60           rts
f193  aa           tax
f194  68           pla
f195  8a           txa
f196  a6 97        ldx $97
f198  60           rts
f199  20 0d f8     jsr $f80d
f19c  d0 0b        bne $f1a9
f19e  20 41 f8     jsr $f841
f1a1  b0 11        bcs $f1b4
f1a3  a9 00        lda #$00
f1a5  85 a6        sta $a6
f1a7  f0 f0        beq $f199
f1a9  b1 b2        lda ($b2),y
f1ab  18           clc
f1ac  60           rts
-------------------------------------------------------------
f1ad  a5 90        lda $90; disketa
f1af  f0 04        beq $f1b5
f1b1  a9 0d        lda #$0d
f1b3  18           clc
f1b4  60           rts
f1b5  4c 13 ee     jmp $ee13; rutina ACPTR
-------------------------------------------------------------
f1b8  20 4e f1     jsr $f14e; cteni z modemu
f1bb  b0 f7        bcs $f1b4
f1bd  c9 00        cmp #$00
f1bf  d0 f2        bne $f1b3
f1c1  ad 97 02     lda $0297
f1c4  29 60        and #$60
f1c6  d0 e9        bne $f1b1
f1c8  f0 ee        beq $f1b8
-------------------------------------------------------------
CHROUT
f1ca  48           pha
f1cb  a5 9a        lda $9a
f1cd  c9 03        cmp #$03
f1cf  d0 04        bne $f1d5
f1d1  68           pla
f1d2  4c 16 e7     jmp $e716
f1d5  90 04        bcc $f1db
f1d7  68           pla
f1d8  4c dd ed     jmp $eddd; rutina CIOUT
f1db  4a           lsr
f1dc  68           pla
f1dd  85 9e        sta $9e
```

```
f1df  8a        txa
f1e0  48        pha
f1e1  98        tya
f1e2  48        pha
f1e3  90 23     bcc $f208
f1e5  20 0d f8  jsr $f80d
f1e8  d0 0e     bne $f1f8
f1ea  20 64 f8  jsr $f864
f1ed  b0 0e     bcs $f1fd
f1ef  a9 02     lda #$02
f1f1  a0 00     ldy #$00
f1f3  91 b2     sta ($b2),y; b. pro cteni z pasky
f1f5  c8        iny
f1f6  84 a6     sty $a6; ukazatel buffru
f1f8  a5 9e     lda $9e; kontrolni suma
f1fa  91 b2     sta ($b2),y
f1fc  18        clc
f1fd  68        pla
f1fe  a8        tay
f1ff  68        pla
f200  aa        tax
f201  a5 9e     lda $9e
f203  90 02     bcc $f207
f205  a9 00     lda #$00
f207  60        rts
f208  20 17 f0  jsr $f017
f20b  4c fc f1  jmp $f1fc
```

```
CHKIN
f20e  20 0f f3  jsr $f30f
f211  f0 03     beq $f216
f213  4c 01 f7  jmp $f701
f216  20 1f f3  jsr $f31f
f219  a5 ba     lda $ba
f21b  f0 16     beq $f233
f21d  c9 03     cmp #$03
f21f  f0 12     beq $f233
f221  b0 14     bcs $f237
f223  c9 02     cmp #$02
f225  d0 03     bne $f22a
f227  4c 4d f0  jmp $f04d
f22a  a6 b9     ldx $b9
f22c  e0 60     cpx #$60
f22e  f0 03     beq $f233
f230  4c 0a f7  jmp $f70a
f233  85 99     sta $99
f235  18        clc
f236  60        rts
f237  aa        tax
f238  20 09 ed  jsr $ed09
f23b  a5 b9     lda $b9
f23d  10 06     bpl $f245
f23f  20 cc ed  jsr $edcc
f242  4c 48 f2  jmp $f248
f245  20 c7 ed  jsr $edc7
f248  8a        txa
f249  24 90     bit $90
f24b  10 e6     bpl $f233
```

```
f24d 4c 07 f7     jmp $f707
------------------------------------------------------------
CHKOUT
f250 20 0f f3     jsr $f30f
f253 f0 03        beq $f258
f255 4c 01 f7     jmp $f701
f258 20 1f f3     jsr $f31f
f25b a5 ba        lda $ba; Kod periferie
f25d d0 03        bne $f262
f25f 4c 0d f7     jmp $f70d
f262 c9 03        cmp #$03
f264 f0 0f        beq $f275
f266 b0 11        bcs $f279
f268 c9 02        cmp #$02
f26a d0 03        bne $f26f
f26c 4c e1 ef     jmp $efe1
f26f a6 b9        ldx $b9
f271 e0 60        cpx #$60
f273 f0 ea        beq $f25f
f275 85 9a        sta $9a
f277 18           clc
f278 60           rts
f279 aa           tax
f27a 20 0c ed     jsr $ed0c
f27d a5 b9        lda $b9
f27f 10 05        bpl $f286
f281 20 be ed     jsr $edbe
f284 d0 03        bne $f289
f286 20 b9 ed     jsr $edb9
f289 8a           txa
f28a 24 90        bit $90
f28c 10 e7        bpl $f275
f28e 4c 07 f7     jmp $f707
f291 20 14 f3     jsr $f314; test na otevr. soub.
f294 f0 02        beq $f298
f296 18           clc
f297 60           rts
f298 20 1f f3     jsr $f31f; akt. data pro soubor
f29b 8a           txa
f29c 48           pha
f29d a5 ba        lda $ba
f29f f0 50        beq $f2f1
f2a1 c9 03        cmp #$03
f2a3 f0 4c        beq $f2f1
f2a5 b0 47        bcs $f2ee
f2a7 c9 02        cmp #$02
f2a9 d0 1d        bne $f2c8
f2ab 68           pla ; modem
f2ac 20 f2 f2     jsr $f2f2
f2af 20 83 f4     jsr $f483
f2b2 20 27 fe     jsr $fe27
f2b5 a5 f8        lda $f8
f2b7 f0 01        beq $f2ba
f2b9 c8           iny
f2ba a5 fa        lda $fa
f2bc f0 01        beq $f2bf
f2be c8           iny
f2bf a9 00        lda #$00
```

```
f2c1  85 f8     sta $f8
f2c3  85 fa     sta $fa
f2c5  4c 7d f4  jmp $f47d
f2c8  a5 b9     lda $b9; magnetofon
f2ca  29 0f     and #$0f
f2cc  f0 23     beq $f2f1
f2ce  20 d0 f7  jsr $f7d0
f2d1  a9 00     lda #$00
f2d3  38        sec
f2d4  20 dd f1  jsr $f1dd
f2d7  20 64 f8  jsr $f864
f2da  90 04     bcc $f2e0
f2dc  68        pla
f2dd  a9 00     lda #$00
f2df  60        rts
f2e0  a5 b9     lda $b9
f2e2  c9 62     cmp #$62
f2e4  d0 0b     bne $f2f1
f2e6  a9 05     lda #$05
f2e8  20 6a f7  jsr $f76a
f2eb  4c f1 f2  jmp $f2f1
f2ee  20 42 f6  jsr $f642; tiskarna/disk.
f2f1  68        pla ; klavesnice/obrazovka
f2f2  aa        tax
f2f3  c6 98     dec $98
f2f5  e4 98     cpx $98
f2f7  f0 14     beq $f30d
f2f9  a4 98     ldy $98; pocet otevr. soub.
f2fb  b9 59 02  lda $0259,y; prepsani tabulky
f2fe  9d 59 02  sta $0259,x; zakladnich dat pro
f301  b9 63 02  lda $0263,y; jednotlive soubory
f304  9d 63 02  sta $0263,x
f307  b9 6d 02  lda $026d,y
f30a  9d 6d 02  sta $026d,x
f30d  18        clc
f30e  60        rts
```

```
Rutina na zjistovani poctu otevrenych souboru a
byl-li jiz dany soubor otevren
f30f  a9 00     lda #$00
f311  85 90     sta $90
f313  8a        txa
f314  a6 98     ldx $98
f316  ca        dex
f317  30 15     bmi $f32e
f319  dd 59 02  cmp $0259,x
f31c  d0 f8     bne $f316
f31e  60        rts
```

```
Prepis dat pro aktualni soubor z tabulky
f31f  bd 59 02  lda $0259,x
f322  85 b8     sta $b8
f324  bd 63 02  lda $0263,x
f327  85 ba     sta $ba
f329  bd 6d 02  lda $026d,x
f32c  85 b9     sta $b9
f32e  60        rts
```

```
f32f  a9 00        lda #$00
f331  85 98        sta $98
CLRCHN
f333  a2 03        ldx #$03; (3..obrazovka)
f335  e4 9a        cpx $9a
f337  b0 03        bcs $f33c
f339  20 fe ed     jsr $edfe
f33c  e4 99        cpx $99
f33e  b0 03        bcs $f343
f340  20 ef ed     jsr $edef
f343  86 9a        stx $9a
f345  a9 00        lda #$00
f347  85 99        sta $99; (0..klavesnice)
f349  60           rts
```

```
OPEN
f34a  a6 b8        ldx $b8
f34c  d0 03        bne $f351
f34e  4c 0a f7     jmp $f70a
f351  20 0f f3     jsr $f30f; test otevr. souboru
f354  d0 03        bne $f359
f356  4c fe f6     jmp $f6fe; hlaska
f359  a6 98        ldx $98; pocet otevrenych souboru
f35b  e0 0a        cpx #$0a; neni jich vice nez $09?
f35d  90 03        bcc $f362
f35f  4c fb f6     jmp $f6fb; hlaska
f362  e6 98        inc $98
f364  a5 b8        lda $b8; zapis akt. dat do tab.
f366  9d 59 02     sta $0259,x
f369  a5 b9        lda $b9
f36b  09 60        ora #$60
f36d  85 b9        sta $b9
f36f  9d 6d 02     sta $026d,x
f372  a5 ba        lda $ba
f374  9d 63 02     sta $0263,x
f377  f0 5a        beq $f3d3
f379  c9 03        cmp #$03
f37b  f0 56        beq $f3d3
f37d  90 05        bcc $f384
f37f  20 d5 f3     jsr $f3d5
f382  90 4f        bcc $f3d3
f384  c9 02        cmp #$02
f386  d0 03        bne $f38b
f388  4c 09 f4     jmp $f409
f38b  20 d0 f7     jsr $f7d0
f38e  b0 03        bcs $f393
f390  4c 13 f7     jmp $f713
f393  a5 b9        lda $b9
f395  29 0f        and #$0f
f397  d0 1f        bne $f3b8
f399  20 17 f8     jsr $f817
f39c  b0 36        bcs $f3d4
f39e  20 af f5     jsr $f5af
f3a1  a5 b7        lda $b7
f3a3  f0 0a        beq $f3af
f3a5  20 ea f7     jsr $f7ea
f3a8  90 18        bcc $f3c2
f3aa  f0 28        beq $f3d4
```

```
f3ac  4c  04  f7      jmp  $f704
f3af  20  2c  f7      jsr  $f72c
f3b2  f0  20          beq  $f3d4
f3b4  90  0c          bcc  $f3c2
f3b6  b0  f4          bcs  $f3ac
f3b8  20  38  f8      jsr  $f838
f3bb  b0  17          bcs  $f3d4
f3bd  a9  04          lda  #$04
f3bf  20  6a  f7      jsr  $f76a
f3c2  a9  bf          lda  #$bf
f3c4  a4  b9          ldy  $b9
f3c6  c0  60          cpy  #$60
f3c8  f0  07          beq  $f3d1
f3ca  a0  00          ldy  #$00
f3cc  a9  02          lda  #$02
f3ce  91  b2          sta  ($b2),y
f3d0  98              tya
f3d1  85  a6          sta  $a6
f3d3  18              clc
f3d4  60              rts
-------------------------------------------------------------------
f3d5  a5  b9          lda  $b9
f3d7  30  fa          bmi  $f3d3
f3d9  a4  b7          ldy  $b7
f3db  f0  f6          beq  $f3d3
f3dd  a9  00          lda  #$00
f3df  85  90          sta  $90
f3e1  a5  ba          lda  $ba
f3e3  20  0c  ed      jsr  $ed0c; LISTEN
f3e6  a5  b9          lda  $b9
f3e8  09  f0          ora  #$f0
f3ea  20  b9  ed      jsr  $edb9; SECOND
f3ed  a5  90          lda  $90
f3ef  10  05          bpl  $f3f6
f3f1  68              pla
f3f2  68              pla
f3f3  4c  07  f7      jmp  $f707
-------------------------------------------------------------------
Vyslani nazvu souboru po seriove sbernici
f3f6  a5  b7          lda  $b7
f3f8  f0  0c          beq  $f406
f3fa  a0  00          ldy  #$00
f3fc  b1  bb          lda  ($bb),y
f3fe  20  dd  ed      jsr  $eddd; CIOUT
f401  c8              iny
f402  c4  b7          cpy  $b7; delka nazvu
f404  d0  f6          bne  $f3fc
f406  4c  54  f6      jmp  $f654
-------------------------------------------------------------------
f409  20  83  f4      jsr  $f483
f40c  8c  97  02      sty  $0297
f40f  c4  b7          cpy  $b7
f411  f0  0a          beq  $f41d
f413  b1  bb          lda  ($bb),y
f415  99  93  02      sta  $0293,y
f418  c8              iny
f419  c0  04          cpy  #$04
f41b  d0  f2          bne  $f40f
```

```
f41d  20 4a ef    jsr $ef4a
f420  8e 98 02    stx $0298
f423  ad 93 02    lda $0293
f426  29 0f       and #$0f
f428  f0 1c       beq $f446
f42a  0a          asl
f42b  aa          tax
f42c  ad a6 02    lda $02a6; PAL/NTSC ?
f42f  d0 09       bne $f43a
f431  bc c1 fe    ldy $fec1,x; NTSC
f434  bd c0 fe    lda $fec0,x
f437  4c 40 f4    jmp $f440
f43a  bc eb e4    ldy $e4eb,x; PAL
f43d  bd ea e4    lda $e4ea,x
f440  8c 96 02    sty $0296
f443  8d 95 02    sta $0295
f446  ad 95 02    lda $0295
f449  0a          asl
f44a  20 2e ff    jsr $ff2e
f44d  ad 94 02    lda $0294
f450  4a          lsr
f451  90 09       bcc $f45c
f453  ad 01 dd    lda $dd01
f456  0a          asl
f457  b0 03       bcs $f45c
f459  20 0d f0    jsr $f00d
f45c  ad 9b 02    lda $029b
f45f  8d 9c 02    sta $029c
f462  ad 9a 02    lda $029a
f465  8d 9d 02    sta $029d
f468  20 27 fe    jsr $fe27
f46b  a5 f8       lda $f8
f46d  d0 05       bne $f474
f46f  88          dey
f470  84 f8       sty $f8
f472  86 f7       stx $f7
f474  a5 fa       lda $fa
f476  d0 05       bne $f47d
f478  88          dey
f479  84 fa       sty $fa
f47b  86 f9       stx $f9
f47d  38          sec
f47e  a9 f0       lda #$f0
f480  4c 2d fe    jmp $fe2d
```

Obsluha CIA#2

```
f483  a9 7f       lda #$7f
f485  8d 0d dd    sta $dd0d
f488  a9 06       lda #$06
f48a  8d 03 dd    sta $dd03
f48d  8d 01 dd    sta $dd01
f490  a9 04       lda #$04
f492  0d 00 dd    ora $dd00
f495  8d 00 dd    sta $dd00
f498  a0 00       ldy #$00
f49a  8c a1 02    sty $02a1; ukazatel NMI
f49d  60          rts
```

```
LOAD
f49e 86 c3        stx $c3
f4a0 84 c4        sty $c4
f4a2 6c 30 03     jmp ($0330)
--------------------------------------------------------------------------
f4a5 85 93        sta $93
f4a7 a9 00        lda #$00
f4a9 85 90        sta $90; nulovani stav. reg.
f4ab a5 ba        lda $ba; Kod Periferie
f4ad d0 03        bne $f4b2
f4af 4c 13 f7     jmp $f713; Klavesnice/obrazovka
f4b2 c9 03        cmp #$03
f4b4 f0 f9        beq $f4af
f4b6 90 7b        bcc $f533
f4b8 a4 b7        ldy $b7; delka nazvu souboru
f4ba d0 03        bne $f4bf
f4bc 4c 10 f7     jmp $f710
f4bf a6 b9        ldx $b9; sekundarni adresa
f4c1 20 af f5     jsr $f5af; hlaska 'Searching..'
f4c4 a9 60        lda #$60
f4c6 85 b9        sta $b9
f4c8 20 d5 f3     jsr $f3d5
f4cb a5 ba        lda $ba
f4cd 20 09 ed     jsr $ed09
f4d0 a5 b9        lda $b9
f4d2 20 c7 ed     jsr $edc7
f4d5 20 13 ee     jsr $ee13
f4d8 85 ae        sta $ae
f4da a5 90        lda $90
f4dc 4a           lsr
f4dd 4a           lsr
f4de b0 50        bcs $f530
f4e0 20 13 ee     jsr $ee13
f4e3 85 af        sta $af
f4e5 8a           txa
f4e6 d0 08        bne $f4f0
f4e8 a5 c3        lda $c3
f4ea 85 ae        sta $ae
f4ec a5 c4        lda $c4
f4ee 85 af        sta $af
f4f0 20 d2 f5     jsr $f5d2
f4f3 a9 fd        lda #$fd
f4f5 25 90        and $90
f4f7 85 90        sta $90
f4f9 20 e1 ff     jsr $ffe1; STOP (test na BREAK)
f4fc d0 03        bne $f501
f4fe 4c 33 f6     jmp $f633
f501 20 13 ee     jsr $ee13
f504 aa           tax
f505 a5 90        lda $90
f507 4a           lsr
f508 4a           lsr
f509 b0 e8        bcs $f4f3
f50b 8a           txa
f50c a4 93        ldy $93; (0..LOAD;1..VERIFY)
f50e f0 0c        beq $f51c
f510 a0 00        ldy #$00
f512 d1 ae        cmp ($ae),y
```

```
f514 f0 08       beq $f51e
f516 a9 10       lda #$10
f518 20 1c fe    jsr $fe1c
f51b 2c 91 ae    bit $ae91
f51e e6 ae       inc $ae
f520 d0 02       bne $f524
f522 e6 af       inc $af
f524 24 90       bit $90
f526 50 cb       bvc $f4f3
f528 20 ef ed    jsr $edef
f52b 20 42 f6    jsr $f642
f52e 90 79       bcc $f5a9
f530 4c 04 f7    jmp $f704
```

```
f533 4a          lsr
f534 b0 03       bcs $f539
f536 4c 13 f7    jmp $f713
f539 20 d0 f7    jsr $f7d0
f53c b0 03       bcs $f541
f53e 4c 13 f7    jmp $f713
Load (verify) z magnetofonu
f541 20 17 f8    jsr $f817; hlasky
f544 b0 68       bcs $f5ae
f546 20 af f5    jsr $f5af; hlasky
f549 a5 b7       lda $b7; delka nazvu
f54b f0 09       beq $f556
f54d 20 ea f7    jsr $f7ea; nactni a porov. nazvu
f550 90 0b       bcc $f55d
f552 f0 5a       beq $f5ae
f554 b0 da       bcs $f530
f556 20 2c f7    jsr $f72c; cteni hlavicky bez p.
f559 f0 53       beq $f5ae
f55b b0 d3       bcs $f530
f55d a5 90       lda $90; test stavoveho reg.
f55f 29 10       and #$10
f561 38          sec
f562 d0 4a       bne $f5ae
f564 e0 01       cpx #$01
f566 f0 11       beq $f579
f568 e0 03       cpx #$03
f56a d0 dd       bne $f549
f56c a0 01       ldy #$01
f56e b1 b2       lda ($b2),y
f570 85 c3       sta $c3
f572 c8          iny
f573 b1 b2       lda ($b2),y
f575 85 c4       sta $c4
f577 b0 04       bcs $f57d
f579 a5 b9       lda $b9
f57b d0 ef       bne $f56c
f57d a0 03       ldy #$03
f57f b1 b2       lda ($b2),y
f581 a0 01       ldy #$01
f583 f1 b2       sbc ($b2),y
f585 aa          tax
f586 a0 04       ldy #$04
f588 b1 b2       lda ($b2),y
f58a a0 02       ldy #$02
```

```
f58c  f1 b2        sbc ($b2),y
f58e  a8           tay
f58f  18           clc
f590  8a           txa
f591  65 c3        adc $c3
f593  85 ae        sta $ae
f595  98           tya
f596  65 c4        adc $c4
f598  85 af        sta $af
f59a  a5 c3        lda $c3
f59c  85 c1        sta $c1
f59e  a5 c4        lda $c4
f5a0  85 c2        sta $c2
f5a2  20 d2 f5     jsr $f5d2; hl. 'Loading/Verif.'
f5a5  20 4a f8     jsr $f84a
f5a8  24 18        bit $18
f5aa  a6 ae        ldx $ae
f5ac  a4 af        ldy $af
f5ae  60           rts ; v X a Y je adresa konce
-------------------------------------------------------
f5af  a5 9d        lda $9d; hlasky zap/vyp ?
f5b1  10 1e        bpl $f5d1
f5b3  a0 0c        ldy #$0c
f5b5  20 2f f1     jsr $f12f; hl. 'Searching for'
f5b8  a5 b7        lda $b7; tisk nazvu souboru
f5ba  f0 15        beq $f5d1
f5bc  a0 17        ldy #$17
f5be  20 2f f1     jsr $f12f
f5c1  a4 b7        ldy $b7
f5c3  f0 0c        beq $f5d1
f5c5  a0 00        ldy #$00
f5c7  b1 bb        lda ($bb),y
f5c9  20 d2 ff     jsr $ffd2
f5cc  c8           iny
f5cd  c4 b7        cpy $b7
f5cf  d0 f6        bne $f5c7
f5d1  60           rts
-------------------------------------------------------
f5d2  a0 49        ldy #$49; 'Loading'
f5d4  a5 93        lda $93
f5d6  f0 02        beq $f5da
f5d8  a0 59        ldy #$59; 'Verifying'
f5da  4c 2b f1     jmp $f12b; hlaska
-------------------------------------------------------
SAVE
f5dd  86 ae        stx $ae
f5df  84 af        sty $af
f5e1  aa           tax
f5e2  b5 00        lda $00,x
f5e4  85 c1        sta $c1; Lo Pocatek
f5e6  b5 01        lda $01,x
f5e8  85 c2        sta $c2; Hi Pocatek
f5ea  6c 32 03     jmp ($0332)
-------------------------------------------------------
f5ed  a5 ba        lda $ba; Kod Periferie
f5ef  d0 03        bne $f5f4
f5f1  4c 13 f7     jmp $f713; klavesnice/obrazovka
f5f4  c9 03        cmp #$03
```

```
f5f6  f0 f9        beq $f5f1
f5f8  90 5f        bcc $f659
f5fa  a9 61        lda #$61; tiskarna/disketa
f5fc  85 b9        sta $b9
f5fe  a4 b7        ldy $b7; delka nazvu souboru
f600  d0 03        bne $f605
f602  4c 10 f7     jmp $f710
f605  20 d5 f3     jsr $f3d5
f608  20 8f f6     jsr $f68f; hlasky
f60b  a5 ba        lda $ba
f60d  20 0c ed     jsr $ed0c; LISTEN
f610  a5 b9        lda $b9
f612  20 b9 ed     jsr $edb9; SECOND
f615  a0 00        ldy #$00
f617  20 8e fb     jsr $fb8e; Poc. Programu (dat)
f61a  a5 ac        lda $ac
f61c  20 dd ed     jsr $eddd; CIOUT
f61f  a5 ad        lda $ad
f621  20 dd ed     jsr $eddd
f624  20 d1 fc     jsr $fcd1
f627  b0 16        bcs $f63f
f629  b1 ac        lda ($ac),y
f62b  20 dd ed     jsr $eddd
f62e  20 e1 ff     jsr $ffe1; STOP
f631  d0 07        bne $f63a
f633  20 42 f6     jsr $f642; zavar po BREAK
f636  a9 00        lda #$00
f638  38           sec
f639  60           rts
f63a  20 db fc     jsr $fcdb
f63d  d0 e5        bne $f624
f63f  20 fe ed     jsr $edfe; UNLSN
f642  24 b9        bit $b9; zakonceni po SAVE
f644  30 11        bmi $f657
f646  a5 ba        lda $ba
f648  20 0c ed     jsr $ed0c
f64b  a5 b9        lda $b9
f64d  29 ef        and #$ef
f64f  09 e0        ora #$e0
f651  20 b9 ed     jsr $edb9
f654  20 fe ed     jsr $edfe
f657  18           clc
f658  60           rts
```

```
f659  4a           lsr
f65a  b0 03        bcs $f65f
f65c  4c 13 f7     jmp $f713; modem
```

```
f65f  20 d0 f7     jsr $f7d0; magnetofon
f662  90 8d        bcc $f5f1
f664  20 38 f8     jsr $f838; test zap. mgf.-hlaska
f667  b0 25        bcs $f68e
f669  20 8f f6     jsr $f68f; hlaska
f66c  a2 03        ldx #$03
f66e  a5 b9        lda $b9
f670  29 01        and #$01
f672  d0 02        bne $f676
f674  a2 01        ldx #$01
```

```
f676  8a          txa
f677  20 6a f7    jsr $f76a; vyslani hlavicky
f67a  b0 12       bcs $f68e
f67c  20 67 f8    jsr $f867
f67f  b0 0d       bcs $f68e
f681  a5 b9       lda $b9
f683  29 02       and #$02
f685  f0 06       beq $f68d
f687  a9 05       lda #$05
f689  20 6a f7    jsr $f76a
f68c  24 18       bit $18
f68e  60          rts
-------------------------------------------------------------
f68f  a5 9d       lda $9d; hlasky zap/vyp ?
f691  10 fb       bpl $f68e
f693  a0 51       ldy #$51
f695  20 2f f1    jsr $f12f; hlaska 'Saving'
f698  4c c1 f5    jmp $f5c1
-------------------------------------------------------------
UDTIM
f69b  a2 00       ldx #$00
f69d  e6 a2       inc $a2
f69f  d0 06       bne $f6a7
f6a1  e6 a1       inc $a1
f6a3  d0 02       bne $f6a7
f6a5  e6 a0       inc $a0
f6a7  38          sec
f6a8  a5 a2       lda $a2
f6aa  e9 01       sbc #$01
f6ac  a5 a1       lda $a1
f6ae  e9 1a       sbc #$1a
f6b0  a5 a0       lda $a0
f6b2  e9 4f       sbc #$4f
f6b4  90 06       bcc $f6bc
f6b6  86 a0       stx $a0
f6b8  86 a1       stx $a1
f6ba  86 a2       stx $a2
f6bc  ad 01 dc    lda $dc01
f6bf  cd 01 dc    cmp $dc01
f6c2  d0 f8       bne $f6bc
f6c4  aa          tax
f6c5  30 13       bmi $f6da
f6c7  a2 bd       ldx #$bd
f6c9  8e 00 dc    stx $dc00
f6cc  ae 01 dc    ldx $dc01
f6cf  ec 01 dc    cpx $dc01
f6d2  d0 f8       bne $f6cc
f6d4  8d 00 dc    sta $dc00
f6d7  e8          inx
f6d8  d0 02       bne $f6dc
f6da  85 91       sta $91
f6dc  60          rts
-------------------------------------------------------------
RDTIM
f6dd  78          sei
f6de  a5 a2       lda $a2
f6e0  a6 a1       ldx $a1
f6e2  a4 a0       ldy $a0
```

```
SETTIM
f6e4  78              sei
f6e5  85 a2           sta $a2
f6e7  86 a1           stx $a1
f6e9  84 a0           sty $a0
f6eb  58              cli
f6ec  60              rts
-------------------------------------------------------------------
STOP
f6ed  a5 91           lda $91
f6ef  c9 7f           cmp #$7f
f6f1  d0 07           bne $f6fa
f6f3  08              php
f6f4  20 cc ff        jsr $ffcc
f6f7  85 c6           sta $c6
f6f9  28              plp
f6fa  60              rts
-------------------------------------------------------------------
f6fb  a9 01           lda #$01
f6fd  2c a9 02        bit $02a9
f700  2c a9 03        bit $03a9
f703  2c a9 04        bit $04a9
f706  2c a9 05        bit $05a9
f709  2c a9 06        bit $06a9
f70c  2c a9 07        bit $07a9
f70f  2c a9 08        bit $08a9
f712  2c a9 09        bit $09a9
f715  48              pha
f716  20 cc ff        jsr $ffcc; CLRCHN
f719  a0 00           ldy #$00
f71b  24 9d           bit $9d
f71d  50 0a           bvc $f729
f71f  20 2f f1        jsr $f12f; hlaska 'I/O error'
f722  68              pla
f723  48              pha
f724  09 30           ora #$30
f726  20 d2 ff        jsr $ffd2; CHROUT
f729  68              pla
f72a  38              sec
f72b  60              rts
-------------------------------------------------------------------
PodProgram Pro LOAD
f72c  a5 93           lda $93
f72e  48              pha
f72f  20 41 f8        jsr $f841
f732  68              pla
f733  85 93           sta $93
f735  b0 32           bcs $f769
f737  a0 00           ldy #$00
f739  b1 b2           lda ($b2),y
f73b  c9 05           cmp #$05
f73d  f0 2a           beq $f769
f73f  c9 01           cmp #$01
f741  f0 08           beq $f74b
f743  c9 03           cmp #$03
f745  f0 04           beq $f74b
f747  c9 04           cmp #$04
f749  d0 e1           bne $f72c
```

```
f74b  aa                tax
f74c  24  9d            bit  $9d
f74e  10  17            bpl  $f767
f750  a0  63            ldy  #$63
f752  20  2f  f1        jsr  $f12f
f755  a0  05            ldy  #$05
f757  b1  b2            lda  ($b2),y
f759  20  d2  ff        jsr  $ffd2
f75c  c8                iny
f75d  c0  15            cpy  #$15
f75f  d0  f6            bne  $f757
f761  a5  a1            lda  $a1
f763  20  e0  e4        jsr  $e4e0
f766  ea                nop
f767  18                clc
f768  88                dey
f769  60                rts
```

```
PodProgram pro SAVE
f76a  85  9e            sta  $9e
f76c  20  d0  f7        jsr  $f7d0
f76f  90  5e            bcc  $f7cf
f771  a5  c2            lda  $c2
f773  48                pha
f774  a5  c1            lda  $c1
f776  48                pha
f777  a5  af            lda  $af
f779  48                pha
f77a  a5  ae            lda  $ae
f77c  48                pha
f77d  a0  bf            ldy  #$bf ; vypln eni buffru $20
f77f  a9  20            lda  #$20
f781  91  b2            sta  ($b2),y
f783  88                dey
f784  d0  fb            bne  $f781
f786  a5  9e            lda  $9e
f788  91  b2            sta  ($b2),y
f78a  c8                iny
f78b  a5  c1            lda  $c1
f78d  91  b2            sta  ($b2),y
f78f  c8                iny
f790  a5  c2            lda  $c2
f792  91  b2            sta  ($b2),y
f794  c8                iny
f795  a5  ae            lda  $ae
f797  91  b2            sta  ($b2),y
f799  c8                iny
f79a  a5  af            lda  $af
f79c  91  b2            sta  ($b2),y
f79e  c8                iny
f79f  84  9f            sty  $9f
f7a1  a0  00            ldy  #$00
f7a3  84  9e            sty  $9e
f7a5  a4  9e            ldy  $9e
f7a7  c4  b7            cpy  $b7 ; delka nazvu souboru
f7a9  f0  0c            beq  $f7b7
f7ab  b1  bb            lda  ($bb),y ; zapis nazvu do
f7ad  a4  9f            ldy  $9f ; buffru Pro m9f.
```

```
f7af  91 b2        sta ($b2),y
f7b1  e6 9e        inc $9e
f7b3  e6 9f        inc $9f
f7b5  d0 ee        bne $f7a5
f7b7  20 d7 f7     jsr $f7d7
f7ba  a9 69        lda #$69
f7bc  85 ab        sta $ab; parita
f7be  20 6b f8     jsr $f86b; vyslani hlavicky
f7c1  a8           tay
f7c2  68           pla
f7c3  85 ae        sta $ae
f7c5  68           pla
f7c6  85 af        sta $af
f7c8  68           pla
f7c9  85 c1        sta $c1
f7cb  68           pla
f7cc  85 c2        sta $c2
f7ce  98           tya
f7cf  60           rts
-------------------------------------------------------
f7d0  a6 b2        ldx $b2; Lo buffer pro cteni z
f7d2  a4 b3        ldy $b3; Hi pasky
f7d4  c0 02        cpy #$02
f7d6  60           rts
-------------------------------------------------------
f7d7  20 d0 f7     jsr $f7d0
f7da  8a           txa
f7db  85 c1        sta $c1
f7dd  18           clc
f7de  69 c0        adc #$c0
f7e0  85 ae        sta $ae
f7e2  98           tya
f7e3  85 c2        sta $c2
f7e5  69 00        adc #$00
f7e7  85 af        sta $af
f7e9  60           rts
-------------------------------------------------------
Nacteni hlavicky + test nazvu
f7ea  20 2c f7     jsr $f72c; load hlavicka
f7ed  b0 1d        bcs $f80c
f7ef  a0 05        ldy #$05
f7f1  84 9f        sty $9f
f7f3  a0 00        ldy #$00
f7f5  84 9e        sty $9e
f7f7  c4 b7        cpy $b7
f7f9  f0 10        beq $f80b
f7fb  b1 bb        lda ($bb),y; Porovnani nazvu
f7fd  a4 9f        ldy $9f
f7ff  d1 b2        cmp ($b2),y
f801  d0 e7        bne $f7ea; nesouhl. - cti znovu
f803  e6 9e        inc $9e
f805  e6 9f        inc $9f
f807  a4 9e        ldy $9e
f809  d0 ec        bne $f7f7
f80b  18           clc
f80c  60           rts
-------------------------------------------------------
f80d  20 d0 f7     jsr $f7d0
```

```
f810  e6 a6        inc $a6
f812  a4 a6        ldy $a6
f814  c0 c0        cpy #$c0
f816  60           rts
--------------------------------------------------------------------------------
f817  20 2e f8     jsr $f82e
f81a  f0 1a        beq $f836
f81c  a0 1b        ldy #$1b
f81e  20 2f f1     jsr $f12f ; hlaska
f821  20 d0 f8     jsr $f8d0
f824  20 2e f8     jsr $f82e
f827  d0 f8        bne $f821
f829  a0 6a        ldy #$6a
f82b  4c 2f f1     jmp $f12f ; hlaska
--------------------------------------------------------------------------------
f82e  a9 10        lda #$10
f830  24 01        bit $01
f832  d0 02        bne $f836
f834  24 01        bit $01
f836  18           clc
f837  60           rts
--------------------------------------------------------------------------------
f838  20 2e f8     jsr $f82e
f83b  f0 f9        beq $f836
f83d  a0 2e        ldy #$2e
f83f  d0 dd        bne $f81e
--------------------------------------------------------------------------------
Obsluha LOAD/VERIFY/SAVE a prenast. IRQ vektoru
f841  a9 00        lda #$00
f843  85 90        sta $90
f845  85 93        sta $93
f847  20 d7 f7     jsr $f7d7
f84a  20 17 f8     jsr $f817
f84d  b0 1f        bcs $f86e
f84f  78           sei
f850  a9 00        lda #$00
f852  85 aa        sta $aa
f854  85 b4        sta $b4
f856  85 b0        sta $b0
f858  85 9e        sta $9e
f85a  85 9f        sta $9f
f85c  85 9c        sta $9c
f85e  a9 90        lda #$90
f860  a2 0e        ldx #$0e
f862  d0 11        bne $f875
f864  20 d7 f7     jsr $f7d7
f867  a9 14        lda #$14
f869  85 ab        sta $ab
f86b  20 38 f8     jsr $f838
f86e  b0 6c        bcs $f8dc
f870  78           sei
f871  a9 82        lda #$82
f873  a2 08        ldx #$08
f875  a0 7f        ldy #$7f
f877  8c 0d dc     sty $dc0d
f87a  8d 0d dc     sta $dc0d
f87d  ad 0e dc     lda $dc0e
f880  09 19        ora #$19
```

```
f882  8d 0f dc     sta $dc0f
f885  29 91        and #$91
f887  8d a2 02     sta $02a2
f88a  20 a4 f0     jsr $f0a4
f88d  ad 11 d0     lda $d011
f890  29 ef        and #$ef
f892  8d 11 d0     sta $d011
f895  ad 14 03     lda $0314
f898  8d 9f 02     sta $029f
f89b  ad 15 03     lda $0315
f89e  8d a0 02     sta $02a0
f8a1  20 bd fc     jsr $fcbd
f8a4  a9 02        lda #$02
f8a6  85 be        sta $be
f8a8  20 97 fb     jsr $fb97
f8ab  a5 01        lda $01
f8ad  29 1f        and #$1f
f8af  85 01        sta $01
f8b1  85 c0        sta $c0
f8b3  a2 ff        ldx #$ff
f8b5  a0 ff        ldy #$ff
f8b7  88           dey
f8b8  d0 fd        bne $f8b7
f8ba  ca           dex
f8bb  d0 f8        bne $f8b5
f8bd  58           cli
f8be  ad a0 02     lda $02a0
f8c1  cd 15 03     cmp $0315
f8c4  18           clc
f8c5  f0 15        beq $f8dc
f8c7  20 d0 f8     jsr $f8d0
f8ca  20 bc f6     jsr $f6bc
f8cd  4c be f8     jmp $f8be
-------------------------------------------------
f8d0  20 e1 ff     jsr $ffe1; STOP
f8d3  18           clc
f8d4  d0 0b        bne $f8e1
f8d6  20 93 fc     jsr $fc93; motor off;nast.VIC;IRQ
f8d9  38           sec
f8da  68           pla
f8db  68           pla
f8dc  a9 00        lda #$00
f8de  8d a0 02     sta $02a0
f8e1  60           rts
-------------------------------------------------
Nova preruseni IRQ pro LOAD/VERIFY/SAVE
f8e2  86 b1        stx $b1
f8e4  a5 b0        lda $b0
f8e6  0a           asl
f8e7  0a           asl
f8e8  18           clc
f8e9  65 b0        adc $b0
f8eb  18           clc
f8ec  65 b1        adc $b1
f8ee  85 b1        sta $b1
f8f0  a9 00        lda #$00
f8f2  24 b0        bit $b0
f8f4  30 01        bmi $f8f7
```

```
f8f6  2a        rol
f8f7  06 b1     asl  $b1
f8f9  2a        rol
f8fa  06 b1     asl  $b1
f8fc  2a        rol
f8fd  aa        tax
f8fe  ad 06 dc  lda  $dc06
f901  c9 16     cmp  #$16
f903  90 f9     bcc  $f8fe
f905  65 b1     adc  $b1
f907  8d 04 dc  sta  $dc04
f90a  8a        txa
f90b  6d 07 dc  adc  $dc07
f90e  8d 05 dc  sta  $dc05
f911  ad a2 02  lda  $02a2
f914  8d 0e dc  sta  $dc0e
f917  8d a4 02  sta  $02a4
f91a  ad 0d dc  lda  $dc0d
f91d  29 10     and  #$10
f91f  f0 09     beq  $f92a
f921  a9 f9     lda  #$f9
f923  48        pha
f924  a9 2a     lda  #$2a
f926  48        pha
f927  4c 43 ff  jmp  $ff43
f92a  58        cli
f92b  60        rts
-------------------------------------------------------
f92c  ae 07 dc  ldx  $dc07
f92f  a0 ff     ldy  #$ff
f931  98        tya
f932  ed 06 dc  sbc  $dc06
f935  ec 07 dc  cpx  $dc07
f938  d0 f2     bne  $f92c
f93a  86 b1     stx  $b1
f93c  aa        tax
f93d  8c 06 dc  sty  $dc06
f940  8c 07 dc  sty  $dc07
f943  a9 19     lda  #$19
f945  8d 0f dc  sta  $dc0f
f948  ad 0d dc  lda  $dc0d
f94b  8d a3 02  sta  $02a3
f94e  98        tya
f94f  e5 b1     sbc  $b1
f951  86 b1     stx  $b1
f953  4a        lsr
f954  66 b1     ror  $b1
f956  4a        lsr
f957  66 b1     ror  $b1
f959  a5 b0     lda  $b0
f95b  18        clc
f95c  69 3c     adc  #$3c
f95e  c5 b1     cmp  $b1
f960  b0 4a     bcs  $f9ac
f962  ae 9c     ldx  $9c
f964  f0 03     beq  $f969
f966  4c 60 fa  jmp  $fa60
-------------------------------------------------------
```

```
f969  a6 a3     ldx $a3
f96b  30 1b     bmi $f988
f96d  a2 00     ldx #$00
f96f  69 30     adc #$30
f971  65 b0     adc $b0
f973  c5 b1     cmp $b1
f975  b0 1c     bcs $f993
f977  e8        inx
f978  69 26     adc #$26
f97a  65 b0     adc $b0
f97c  c5 b1     cmp $b1
f97e  b0 17     bcs $f997
f980  69 2c     adc #$2c
f982  65 b0     adc $b0
f984  c5 b1     cmp $b1
f986  90 03     bcc $f98b
f988  4c 10 fa  jmp $fa10
--------------------------------------------------------
f98b  a5 b4     lda $b4
f98d  f0 1d     beq $f9ac
f98f  85 a8     sta $a8
f991  d0 19     bne $f9ac
f993  e6 a9     inc $a9
f995  b0 02     bcs $f999
f997  c6 a9     dec $a9
f999  38        sec
f99a  e9 13     sbc #$13
f99c  e5 b1     sbc $b1
f99e  65 92     adc $92
f9a0  85 92     sta $92
f9a2  a5 a4     lda $a4
f9a4  49 01     eor #$01
f9a6  85 a4     sta $a4
f9a8  f0 2b     beq $f9d5
f9aa  86 d7     stx $d7
f9ac  a5 b4     lda $b4
f9ae  f0 22     beq $f9d2
f9b0  ad a3 02  lda $02a3
f9b3  29 01     and #$01
f9b5  d0 05     bne $f9bc
f9b7  ad a4 02  lda $02a4
f9ba  d0 16     bne $f9d2
f9bc  a9 00     lda #$00
f9be  85 a4     sta $a4
f9c0  8d a4 02  sta $02a4
f9c3  a5 a3     lda $a3
f9c5  10 30     bpl $f9f7
f9c7  30 bf     bmi $f988
f9c9  a2 a6     ldx #$a6
f9cb  20 e2 f8  jsr $f8e2
f9ce  a5 9b     lda $9b
f9d0  d0 b9     bne $f98b
f9d2  4c bc fe  jmp $febc
--------------------------------------------------------
f9d5  a5 92     lda $92
f9d7  f0 07     beq $f9e0
f9d9  30 03     bmi $f9de
f9db  c6 b0     dec $b0
```

```
f9dd  2c  e6  b0      bit  $b0e6
f9e0  a9  00          lda  #$00
f9e2  85  92          sta  $92
f9e4  e4  d7          cpx  $d7
f9e6  d0  0f          bne  $f9f7
f9e8  8a              txa
f9e9  d0  a0          bne  $f98b
f9eb  a5  a9          lda  $a9
f9ed  30  bd          bmi  $f9ac
f9ef  c9  10          cmp  #$10
f9f1  90  b9          bcc  $f9ac
f9f3  85  96          sta  $96
f9f5  b0  b5          bcs  $f9ac
f9f7  8a              txa
f9f8  45  9b          eor  $9b
f9fa  85  9b          sta  $9b
f9fc  a5  b4          lda  $b4
f9fe  f0  d2          beq  $f9d2
fa00  c6  a3          dec  $a3
fa02  30  c5          bmi  $f9c9
fa04  46  d7          lsr  $d7
fa06  66  bf          ror  $bf
fa08  a2  da          ldx  #$da
fa0a  20  e2  f8      jsr  $f8e2
fa0d  4c  bc  fe      jmp  $febc
------------------------------------------------------------
fa10  a5  96          lda  $96
fa12  f0  04          beq  $fa18
fa14  a5  b4          lda  $b4
fa16  f0  07          beq  $fa1f
fa18  a5  a3          lda  $a3
fa1a  30  03          bmi  $fa1f
fa1c  4c  97  f9      jmp  $f997
fa1f  46  b1          lsr  $b1
fa21  a9  93          lda  #$93
fa23  38              sec
fa24  e5  b1          sbc  $b1
fa26  65  b0          adc  $b0
fa28  0a              asl
fa29  aa              tax
fa2a  20  e2  f8      jsr  $f8e2
fa2d  e6  9c          inc  $9c
fa2f  a5  b4          lda  $b4
fa31  d0  11          bne  $fa44
fa33  a5  96          lda  $96
fa35  f0  26          beq  $fa5d
fa37  85  a8          sta  $a8
fa39  a9  00          lda  #$00
fa3b  85  96          sta  $96
fa3d  a9  81          lda  #$81
fa3f  8d  0d  dc      sta  $dc0d
fa42  85  b4          sta  $b4
fa44  a5  96          lda  $96
fa46  85  b5          sta  $b5
fa48  f0  09          beq  $fa53
fa4a  a9  00          lda  #$00
fa4c  85  b4          sta  $b4
fa4e  a9  01          lda  #$01
```

```
fa50  8d 0d dc     sta $dc0d
fa53  a5 bf        lda $bf
fa55  85 bd        sta $bd
fa57  a5 a8        lda $a8
fa59  05 a9        ora $a9
fa5b  85 b6        sta $b6
fa5d  4c bc fe     jmp $febc
--------------------------------------------------------
fa60  20 97 fb     jsr $fb97
fa63  85 9c        sta $9c
fa65  a2 da        ldx #$da
fa67  20 e2 f8     jsr $f8e2
fa6a  a5 be        lda $be
fa6c  f0 02        beq $fa70
fa6e  85 a7        sta $a7
fa70  a9 0f        lda #$0f
fa72  24 aa        bit $aa
fa74  10 17        bpl $fa8d
fa76  a5 b5        lda $b5
fa78  d0 0c        bne $fa86
fa7a  a6 be        ldx $be
fa7c  ca           dex
fa7d  d0 0b        bne $fa8a
fa7f  a9 08        lda #$08
fa81  20 1c fe     jsr $fe1c
fa84  d0 04        bne $fa8a
fa86  a9 00        lda #$00
fa88  85 aa        sta $aa
fa8a  4c bc fe     jmp $febc
--------------------------------------------------------
fa8d  70 31        bvs $fac0
fa8f  d0 18        bne $faa9
fa91  a5 b5        lda $b5
fa93  d0 f5        bne $fa8a
fa95  a5 b6        lda $b6
fa97  d0 f1        bne $fa8a
fa99  a5 a7        lda $a7
fa9b  4a           lsr
fa9c  a5 bd        lda $bd
fa9e  30 03        bmi $faa3
faa0  90 18        bcc $faba
faa2  18           clc
faa3  b0 15        bcs $faba
faa5  29 0f        and #$0f
faa7  85 aa        sta $aa
faa9  c6 aa        dec $aa
faab  d0 dd        bne $fa8a
faad  a9 40        lda #$40
faaf  85 aa        sta $aa
fab1  20 8e fb     jsr $fb8e
fab4  a9 00        lda #$00
fab6  85 ab        sta $ab
fab8  f0 d0        beq $fa8a
faba  a9 80        lda #$80
fabc  85 aa        sta $aa
fabe  d0 ca        bne $fa8a
fac0  a5 b5        lda $b5
fac2  f0 0a        beq $face
```

```
fac4  a9 04       lda #$04
fac6  20 1c fe    jsr $fe1c
fac9  a9 00       lda #$00
facb  4c 4a fb    jmp $fb4a
face  20 d1 fc    jsr $fcd1
fad1  90 03       bcc $fad6
fad3  4c 48 fb    jmp $fb48
fad6  a6 a7       ldx $a7
fad8  ca          dex
fad9  f0 2d       beq $fb08
fadb  a5 93       lda $93; (0..Load;1..Verify)
fadd  f0 0c       beq $faeb
fadf  a0 00       ldy #$00
fae1  a5 bd       lda $bd; odecteny znak
fae3  d1 ac       cmp ($ac),y; Porovnani
fae5  f0 04       beq $faeb
fae7  a9 01       lda #$01
fae9  85 b6       sta $b6
faeb  a5 b6       lda $b6
faed  f0 4b       beq $fb3a
faef  a2 3d       ldx #$3d
faf1  e4 9e       cpx $9e
faf3  90 3e       bcc $fb33
faf5  a6 9e       ldx $9e
faf7  a5 ad       lda $ad; Hi Pocatek ukladani
faf9  9d 01 01    sta $0101,x
fafc  a5 ac       lda $ac; Lo dat z buffru
fafe  9d 00 01    sta $0100,x
fb01  e8          inx
fb02  e8          inx
fb03  86 9e       stx $9e
fb05  4c 3a fb    jmp $fb3a
-------------------------------------------------
fb08  a6 9f       ldx $9f
fb0a  e4 9e       cpx $9e
fb0c  f0 35       beq $fb43
fb0e  a5 ac       lda $ac
fb10  dd 00 01    cmp $0100,x
fb13  d0 2e       bne $fb43
fb15  a5 ad       lda $ad
fb17  dd 01 01    cmp $0101,x
fb1a  d0 27       bne $fb43
fb1c  e6 9f       inc $9f
fb1e  e6 9f       inc $9f
fb20  a5 93       lda $93
fb22  f0 0b       beq $fb2f
fb24  a5 bd       lda $bd
fb26  a0 00       ldy #$00
fb28  d1 ac       cmp ($ac),y
fb2a  f0 17       beq $fb43
fb2c  c8          iny
fb2d  84 b6       sty $b6
fb2f  a5 b6       lda $b6
fb31  f0 07       beq $fb3a
fb33  a9 10       lda #$10
fb35  20 1c fe    jsr $fe1c
fb38  d0 09       bne $fb43
fb3a  a5 93       lda $93
```

```
fb3c  d0 05        bne $fb43
fb3e  a8           tay
fb3f  a5 bd        lda $bd
fb41  91 ac        sta ($ac),y
fb43  20 db fc     jsr $fcdb
fb46  d0 43        bne $fb8b
fb48  a9 80        lda #$80
fb4a  85 aa        sta $aa
fb4c  78           sei
fb4d  a2 01        ldx #$01
fb4f  8e 0d dc     stx $dc0d
fb52  ae 0d dc     ldx $dc0d
fb55  a6 be        ldx $be
fb57  ca           dex
fb58  30 02        bmi $fb5c
fb5a  86 be        stx $be
fb5c  c6 a7        dec $a7
fb5e  f0 08        beq $fb68
fb60  a5 9e        lda $9e
fb62  d0 27        bne $fb8b
fb64  85 be        sta $be
fb66  f0 23        beq $fb8b
fb68  20 93 fc     jsr $fc93
fb6b  20 8e fb     jsr $fb8e
fb6e  a0 00        ldy #$00
fb70  84 ab        sty $ab
fb72  b1 ac        lda ($ac),y
fb74  45 ab        eor $ab
fb76  85 ab        sta $ab
fb78  20 db fc     jsr $fcdb
fb7b  20 d1 fc     jsr $fcd1
fb7e  90 f2        bcc $fb72
fb80  a5 ab        lda $ab
fb82  45 bd        eor $bd
fb84  f0 05        beq $fb8b
fb86  a9 20        lda #$20
fb88  20 1c fe     jsr $fe1c
fb8b  4c bc fe     jmp $febc
```

```
fb8e  a5 c2        lda $c2; Prepis Lo a Hi adre-
fb90  85 ad        sta $ad; sy Pocatku Programu
fb92  a5 c1        lda $c1; (dat)
fb94  85 ac        sta $ac
fb96  60           rts
```

```
fb97  a9 08        lda #$08; nastaveni syst. Prom.
fb99  85 a3        sta $a3; pro Prenos
fb9b  a9 00        lda #$00
fb9d  85 a4        sta $a4
fb9f  85 a8        sta $a8
fba1  85 9b        sta $9b
fba3  85 a9        sta $a9
fba5  60           rts
```

Nastaveni casovace a Preruseni CIA#1

```
fba6  a5 bd        lda $bd
fba8  4a           lsr
fba9  a9 60        lda #$60
```

```
fbab  90 02        bcc $fbaf
fbad  a9 b0        lda #$b0
fbaf  a2 00        ldx #$00
fbb1  8d 06 dc     sta $dc06
fbb4  8e 07 dc     stx $dc07
fbb7  ad 0d dc     lda $dc0d
fbba  a9 19        lda #$19
fbbc  8d 0f dc     sta $dc0f
fbbf  a5 01        lda $01
fbc1  49 08        eor #$08
fbc3  85 01        sta $01
fbc5  29 08        and #$08
fbc7  60           rts
```

```
fbc8  38           sec
fbc9  66 b6        ror $b6
fbcb  30 3c        bmi $fc09
Preruseni IRQ pro obsluhu vstupu/vystupu
fbcd  a5 a8        lda $a8
fbcf  d0 12        bne $fbe3
fbd1  a9 10        lda #$10
fbd3  a2 01        ldx #$01
fbd5  20 b1 fb     jsr $fbb1
fbd8  d0 2f        bne $fc09
fbda  e6 a8        inc $a8
fbdc  a5 b6        lda $b6
fbde  10 29        bpl $fc09
fbe0  4c 57 fc     jmp $fc57
fbe3  a5 a9        lda $a9
fbe5  d0 09        bne $fbf0
fbe7  20 ad fb     jsr $fbad
fbea  d0 1d        bne $fc09
fbec  e6 a9        inc $a9
fbee  d0 19        bne $fc09
fbf0  20 a6 fb     jsr $fba6
fbf3  d0 14        bne $fc09
fbf5  a5 a4        lda $a4
fbf7  49 01        eor #$01
fbf9  85 a4        sta $a4
fbfb  f0 0f        beq $fc0c
fbfd  a5 bd        lda $bd
fbff  49 01        eor #$01
fc01  85 bd        sta $bd
fc03  29 01        and #$01
fc05  45 9b        eor $9b
fc07  85 9b        sta $9b
fc09  4c bc fe     jmp $febc
fc0c  46 bd        lsr $bd
fc0e  c6 a3        dec $a3
fc10  a5 a3        lda $a3
fc12  f0 3a        beq $fc4e
fc14  10 f3        bpl $fc09
fc16  20 97 fb     jsr $fb97
fc19  58           cli
fc1a  a5 a5        lda $a5
fc1c  f0 12        beq $fc30
fc1e  a2 00        ldx #$00
fc20  86 d7        stx $d7
```

```
fc22  c6 a5        dec $a5
fc24  a6 be        ldx $be
fc26  e0 02        cpx #$02
fc28  d0 02        bne $fc2c
fc2a  09 80        ora #$80
fc2c  85 bd        sta $bd
fc2e  d0 d9        bne $fc09
fc30  20 d1 fc     jsr $fcd1
fc33  90 0a        bcc $fc3f
fc35  d0 91        bne $fbc8
fc37  e6 ad        inc $ad
fc39  a5 d7        lda $d7
fc3b  85 bd        sta $bd
fc3d  b0 ca        bcs $fc09
fc3f  a0 00        ldy #$00
fc41  b1 ac        lda ($ac),y
fc43  85 bd        sta $bd
fc45  45 d7        eor $d7
fc47  85 d7        sta $d7
fc49  20 db fc     jsr $fcdb
fc4c  d0 bb        bne $fc09
fc4e  a5 9b        lda $9b
fc50  49 01        eor #$01
fc52  85 bd        sta $bd
fc54  4c bc fe     jmp $febc
fc57  c6 be        dec $be
fc59  d0 03        bne $fc5e
fc5b  20 ca fc     jsr $fcca
fc5e  a9 50        lda #$50
- - - - - - - - - - - - - - -
fc60  85 a7        sta $a7
fc62  a2 08        ldx #$08
fc64  78           sei
fc65  20 bd fc     jsr $fcbd
fc68  d0 ea        bne $fc54
fc6a  a9 78        lda #$78
fc6c  20 af fb     jsr $fbaf
fc6f  d0 e3        bne $fc54
fc71  c6 a7        dec $a7
fc73  d0 df        bne $fc54
fc75  20 97 fb     jsr $fb97
fc78  c6 ab        dec $ab
fc7a  10 d8        bpl $fc54
fc7c  a2 0a        ldx #$0a
fc7e  20 bd fc     jsr $fcbd
fc81  58           cli
fc82  e6 ab        inc $ab
fc84  a5 be        lda $be
fc86  f0 30        beq $fcb8
fc88  20 8e fb     jsr $fb8e
fc8b  a2 09        ldx #$09
fc8d  86 a5        stx $a5
fc8f  86 b6        stx $b6
fc91  d0 83        bne $fc16
- - - - - - - - - - - - - - -
fc93  08           php
fc94  78           sei
fc95  ad 11 d0     lda $d011; zapnuti obrazovky
```

```
fc98 09 10      ora #$10
fc9a 8d 11 d0   sta $d011
fc9d 20 ca fc   jsr $fcca; vypnuti motoru
fca0 a9 7f      lda #$7f
fca2 8d 0d dc   sta $dc0d; nastaveni IRQ rg.
fca5 20 dd fd   jsr $fddd; nast. VIC (viz. CINT)
fca8 ad a0 02   lda $02a0; predchozi stav IRQ
fcab f0 09      beq $fcb6
fcad 8d 15 03   sta $0315; Hi
fcb0 ad 9f 02   lda $029f
fcb3 8d 14 03   sta $0314; Lo IRQ vektor
fcb6 28         plp
fcb7 60         rts
-------------------------------------------------
fcb8 20 93 fc   jsr $fc93
fcbb f0 97      beq $fc54
fcbd bd 93 fd   lda $fd93,x; nast. noveho vekto-
fcc0 8d 14 03   sta $0314
fcc3 bd 94 fd   lda $fd94,x; ru IRQ z tabulky
fcc6 8d 15 03   sta $0315
fcc9 60         rts
-------------------------------------------------
fcca a5 01      lda $01; vypnuti motoru
fccc 09 20      ora #$20
fcce 85 01      sta $01
fcd0 60         rts
-------------------------------------------------
fcd1 38         sec ; vypocet vzdalenosti mezi
fcd2 a5 ac      lda $ac; Pocatkem a Koncem Pr.
fcd4 e5 ae      sbc $ae
fcd6 a5 ad      lda $ad
fcd8 e5 af      sbc $af
fcda 60         rts
-------------------------------------------------
fcdb e6 ac      inc $ac; Posunuti uk. Pocatku
fcdd d0 02      bne $fce1
fcdf e6 ad      inc $ad; Pro Load
fce1 60         rts
-------------------------------------------------
RESET Preruseni (nejvyssi Priorita)
fce2 a2 ff      ldx #$ff
fce4 78         sei
fce5 9a         txs ; Prenast. uk. zasobniku
fce6 d8         cld
fce7 20 02 fd   jsr $fd02; test je-li cartridge
fcea d0 03      bne $fcef
fcec 6c 00 80   jmp ($8000); skok na cart. RESET
fcef 8e 16 d0   stx $d016; zarolovani obrazovky
fcf2 20 a3 fd   jsr $fda3; IOINIT
fcf5 20 50 fd   jsr $fd50; RAMTAS
fcf8 20 15 fd   jsr $fd15; RESTOR
fcfb 20 5b ff   jsr $ff5b; CINT
fcfe 58         cli
fcff 6c 00 a0   jmp ($a000); stand. RESET ($e394)
-------------------------------------------------
Test lze-li ocekavat nove vektory pro RESET a
RESTORE (napr. pri pouziti cartridge)
fd02 a2 05      ldx #$05
```

```
fd04 bd 0f fd     lda $fd0f,x
fd07 dd 03 80     cmp $8003,x; Porovnavani Kodu
fd0a d0 03        bne $fd0f
fd0c ca           dex
fd0d d0 f5        bne $fd04
fd0f 60           rts
```

```
Kody Pro dekodovani (napr. cartridge)
fd10 c3 c2 cd 38 30          ; CBM80
```

```
RESTOR
fd15 a2 30        ldx #$30
fd17 a0 fd        ldy #$fd
fd19 18           clc
fd1a 86 c3        stx $c3
fd1c 84 c4        sty $c4
fd1e a0 1f        ldy #$1f
fd20 b9 14 03     lda $0314,y
fd23 b0 02        bcs $fd27
fd25 b1 c3        lda ($c3),y
fd27 91 c3        sta ($c3),y
fd29 99 14 03     sta $0314,y
fd2c 88           dey
fd2d 10 f1        bpl $fd20
fd2f 60           rts
```

```
Vektory $0314-$0315 a dale
fd30 31 ea 66 fe 47 fe 4a f3
fd38 91 f2 0e f2 50 f2 33 f3
fd40 57 f1 ca f1 ed f6 3e f1
fd48 2f f3 66 fe a5 f4 ed f5
```

```
RAMTAS
fd50 a9 00        lda #$00
fd52 a8           tay
fd53 99 02 00     sta $0002,y; nulovani
fd56 99 00 02     sta $0200,y
fd59 99 00 03     sta $0300,y
fd5c c8           iny
fd5d d0 f4        bne $fd53
fd5f a2 3c        ldx #$3c
fd61 a0 03        ldy #$03
fd63 86 b2        stx $b2; lo Pocatek buffru Pri
fd65 84 b3        sty $b3; Hi Praci s Paskou
fd67 a8           tay
fd68 a9 03        lda #$03
fd6a 85 c2        sta $c2
fd6c e6 c2        inc $c2
fd6e b1 c1        lda ($c1),y
fd70 aa           tax
fd71 a9 55        lda #$55
fd73 91 c1        sta ($c1),y
fd75 d1 c1        cmp ($c1),y
fd77 d0 0f        bne $fd88
fd79 2a           rol
fd7a 91 c1        sta ($c1),y
fd7c d1 c1        cmp ($c1),y
fd7e d0 08        bne $fd88
```

```
fd80  8a          txa
fd81  91  c1      sta  ($c1),y
fd83  c8          iny
fd84  d0  e8      bne  $fd6e
fd86  f0  e4      beq  $fd6c
fd88  98          tya
fd89  aa          tax
fd8a  a4  c2      ldy  $c2
fd8c  18          clc
fd8d  20  2d  fe  jsr  $fe2d
fd90  a9  08      lda  #$08
fd92  8d  82  02  sta  $0282
fd95  a9  04      lda  #$04
fd97  8d  88  02  sta  $0288
fd9a  60          rts
```

IRQ vektory pro praci se vstupy/vystupy

```
fd9b  6a  fc  cd  fb  31  ea  2c  f9
```

IOINIT

```
fda3  a9  7f      lda  #$7f
fda5  8d  0d  dc  sta  $dc0d
fda8  8d  0d  dd  sta  $dd0d
fdab  8d  00  dc  sta  $dc00
fdae  a9  08      lda  #$08
fdb0  8d  0e  dc  sta  $dc0e
fdb3  8d  0e  dd  sta  $dd0e
fdb6  8d  0f  dc  sta  $dc0f
fdb9  8d  0f  dd  sta  $dd0f
fdbc  a2  00      ldx  #$00
fdbe  8e  03  dc  stx  $dc03
fdc1  8e  03  dd  stx  $dd03
fdc4  8e  18  d4  stx  $d418
fdc7  ca          dex
fdc8  8e  02  dc  stx  $dc02
fdcb  a9  07      lda  #$07
fdcd  8d  00  dd  sta  $dd00
fdd0  a9  3f      lda  #$3f
fdd2  8d  02  dd  sta  $dd02
fdd5  a9  e7      lda  #$e7
fdd7  85  01      sta  $01
fdd9  a9  2f      lda  #$2f
fddb  85  00      sta  $00
fddd  ad  a6  02  lda  $02a6; nastaveni registru
fde0  f0  0a      beq  $fdec; Podle PAL/NTSC
fde2  a9  25      lda  #$25
fde4  8d  04  dc  sta  $dc04
fde7  a9  40      lda  #$40
fde9  4c  f3  fd  jmp  $fdf3
fdec  a9  95      lda  #$95
fdee  8d  04  dc  sta  $dc04
fdf1  a9  42      lda  #$42
fdf3  8d  05  dc  sta  $dc05
fdf6  4c  6e  ff  jmp  $ff6e
```

SETNAM

```
fdf9  85  b7      sta  $b7
fdfb  86  bb      stx  $bb
```

```
fdfd 84 bc          sty $bc
fdff 60             rts
-------------------------------------------------
SETLFS
fe00 85 b8          sta $b8
fe02 86 ba          stx $ba
fe04 84 b9          sty $b9
fe06 60             rts
-------------------------------------------------
READST
fe07 a5 ba          lda $ba; Kod periferie
fe09 c9 02          cmp #$02
fe0b d0 0d          bne $fe1a
fe0d ad 97 02       lda $0297
fe10 48             pha
fe11 a9 00          lda #$00
fe13 8d 97 02       sta $0297
fe16 68             pla
fe17 60             rts
-------------------------------------------------
SETMSG
fe18 85 9d          sta $9d
fe1a a5 90          lda $90
fe1c 05 90          ora $90
fe1e 85 90          sta $90
fe20 60             rts
-------------------------------------------------
SETTMO
fe21 8d 85 02       sta $0285
fe24 60             rts
-------------------------------------------------
MEMTOP
fe25 90 06          bcc $fe2d
fe27 ae 83 02       ldx $0283
fe2a ac 84 02       ldy $0284
fe2d 8e 83 02       stx $0283
fe30 8c 84 02       sty $0284
fe33 60             rts
-------------------------------------------------
MEMBOT
fe34 90 06          bcc $fe3c
fe36 ae 81 02       ldx $0281
fe39 ac 82 02       ldy $0282
fe3c 8e 81 02       stx $0281
fe3f 8c 82 02       sty $0282
fe42 60             rts
-------------------------------------------------
RESTORE Preruseni (NMI)
fe43 78             sei
fe44 6c 18 03       jmp ($0318)
-------------------------------------------------
fe47 48             pha
fe48 8a             txa
fe49 48             pha
fe4a 98             tya
fe4b 48             pha
fe4c a9 7f          lda #$7f
fe4e 8d 0d dd       sta $dd0d
```

```
fe51  ac 0d dd     ldy $dd0d
fe54  30 1c        bmi $fe72
fe56  20 02 fd     jsr $fd02; test na cartridge
fe59  d0 03        bne $fe5e
fe5b  6c 02 80     jmp ($8002); novy NMI vektor
fe5e  20 bc f6     jsr $f6bc; viz. UDTIM
fe61  20 e1 ff     jsr $ffe1; test kl. STOP
fe64  d0 0c        bne $fe72
fe66  20 15 fd     jsr $fd15; RESTOR
fe69  20 a3 fd     jsr $fda3; IOINIT
fe6c  20 18 e5     jsr $e518; inicializace VIC
fe6f  6c 02 a0     jmp ($a002); stand. NMI ($e37b)
```

Navrat z preruseni pro praci se vstupy/vystupy

```
fe72  98           tya
fe73  2d a1 02     and $02a1
fe76  aa           tax
fe77  29 01        and #$01
fe79  f0 28        beq $fea3
fe7b  ad 00 dd     lda $dd00
fe7e  29 fb        and #$fb
fe80  05 b5        ora $b5
fe82  8d 00 dd     sta $dd00
fe85  ad a1 02     lda $02a1
fe88  8d 0d dd     sta $dd0d
fe8b  8a           txa
fe8c  29 12        and #$12
fe8e  f0 0d        beq $fe9d
fe90  29 02        and #$02
fe92  f0 06        beq $fe9a
fe94  20 d6 fe     jsr $fed6
fe97  4c 9d fe     jmp $fe9d
fe9a  20 07 ff     jsr $ff07
fe9d  20 bb ee     jsr $eebb
fea0  4c b6 fe     jmp $feb6
fea3  8a           txa
fea4  29 02        and #$02
fea6  f0 06        beq $feae
fea8  20 d6 fe     jsr $fed6
feab  4c b6 fe     jmp $feb6
feae  8a           txa
feaf  29 10        and #$10
feb1  f0 03        beq $feb6
feb3  20 07 ff     jsr $ff07
feb6  ad a1 02     lda $02a1
feb9  8d 0d dd     sta $dd0d
febc  68           pla
febd  a8           tay
febe  68           pla
febf  aa           tax
fec0  68           pla
fec1  40           rti
```

```
fec2  c1 27 3e 1a  c5 11 74 0e; Data
feca  ed 0c 45 06  f0 02 46 01
fed2  b8 00 71 00
```

Nastaveni casovace B CIA#2

```
fed6  ad 01 dd     lda $dd01
fed9  29 01        and #$01
fedb  85 a7        sta $a7
fedd  ad 06 dd     lda $dd06
fee0  e9 1c        sbc #$1c
fee2  6d 99 02     adc $0299
fee5  8d 06 dd     sta $dd06
fee8  ad 07 dd     lda $dd07
feeb  6d 9a 02     adc $029a
feee  8d 07 dd     sta $dd07
fef1  a9 11        lda #$11
fef3  8d 0f dd     sta $dd0f
fef6  ad a1 02     lda $02a1
fef9  8d 0d dd     sta $dd0d
fefc  a9 ff        lda #$ff
fefe  8d 06 dd     sta $dd06
ff01  8d 07 dd     sta $dd07
ff04  4c 59 ef     jmp $ef59
-------------------------------------------------------
Nastaveni nestandardnich parametru pro casovac B
ff07  ad 95 02     lda $0295
ff0a  8d 06 dd     sta $dd06
ff0d  ad 96 02     lda $0296
ff10  8d 07 dd     sta $dd07
ff13  a9 11        lda #$11
ff15  8d 0f dd     sta $dd0f
ff18  a9 12        lda #$12
ff1a  4d a1 02     eor $02a1
ff1d  8d a1 02     sta $02a1
ff20  a9 ff        lda #$ff
ff22  8d 06 dd     sta $dd06
ff25  8d 07 dd     sta $dd07
ff28  ae 98 02     ldx $0298
ff2b  86 a8        stx $a8
ff2d  60           rts
-------------------------------------------------------
ff2e  aa           tax
ff2f  ad 96 02     lda $0296
ff32  2a           rol
ff33  a8           tay
ff34  8a           txa
ff35  69 c8        adc #$c8
ff37  8d 99 02     sta $0299
ff3a  98           tya
ff3b  69 00        adc #$00
ff3d  8d 9a 02     sta $029a
ff40  60           rts
-------------------------------------------------------
ff41  ea           nop ; Preruseni po praci s in/out
ff42  ea           nop ; (napr. pri Load po detekci
ff43  08           php ; nahravky s jinou casovou
ff44  68           pla ; konstantou)
ff45  29 ef        and #$ef
ff47  48           pha
Preruseni IRQ/BREAK (po instrukci BRK)
ff48  48           pha
ff49  8a           txa
ff4a  48           pha
```

```
ff4b  98          tya
ff4c  48          pha
ff4d  ba          tsx
ff4e  bd 04 01    lda $0104,x
ff51  29 10       and #$10
ff53  f0 03       beq $ff58
ff55  6c 16 03    jmp ($0316); vektor pro BRK
ff58  6c 14 03    jmp ($0314); vektor pro IRQ
-------------------------------------------------
CINT
ff5b  20 18 e5    jsr $e518
ff5e  ad 12 d0    lda $d012
ff61  d0 fb       bne $ff5e
ff63  ad 19 d0    lda $d019
ff66  29 01       and #$01
ff68  8d a6 02    sta $02a6
ff6b  4c dd fd    jmp $fddd
-------------------------------------------------
ff6e  a9 81       lda #$81
ff70  8d 0d dc    sta $dc0d
ff73  ad 0e dc    lda $dc0e
ff76  29 80       and #$80
ff78  09 11       ora #$11
ff7a  8d 0e dc    sta $dc0e
ff7d  4c 8e ee    jmp $ee8e
-------------------------------------------------
ff80  03
-------------------------------------------------
Tabulka skoku na nektere rutiny
ff81  4c 5b ff    jmp $ff5b; CINT
ff84  4c a3 fd    jmp $fda3; IOINIT
ff87  4c 50 fd    jmp $fd50; RAMTAS
ff8a  4c 15 fd    jmp $fd15; RESTOR
ff8d  4c 1a fd    jmp $fd1a; VECTOR
ff90  4c 18 fe    jmp $fe18; SETMSG
ff93  4c b9 ed    jmp $edb9; SECOND
ff96  4c c7 ed    jmp $edc7; TKSA
ff99  4c 25 fe    jmp $fe25; MEMTOP
ff9c  4c 34 fe    jmp $fe34; MEMBOT
ff9f  4c 87 ea    jmp $ea87; SCNKEY
ffa2  4c 21 fe    jmp $fe21; SETTMO
ffa5  4c 13 ee    jmp $ee13; ACPTR
ffa8  4c dd ed    jmp $eddd; CIOUT
ffab  4c ef ed    jmp $edef; UNTLK
ffae  4c fe ed    jmp $edfe; UNLSN
ffb1  4c 0c ed    jmp $ed0c; LISTEN
ffb4  4c 09 ed    jmp $ed09; TALK
ffb7  4c 07 fe    jmp $fe07; READST
ffba  4c 00 fe    jmp $fe00; SETLFS
ffbd  4c f9 fd    jmp $fdf9; SETNAM
ffc0  6c 1a 03    jmp ($031a); OPEN
ffc3  6c 1c 03    jmp ($031c); CLOSE
ffc6  6c 1e 03    jmp ($031e); CHKIN
ffc9  6c 20 03    jmp ($0320); CHKOUT
ffcc  6c 22 03    jmp ($0322); CLRCHN
ffcf  6c 24 03    jmp ($0324); CHRIN
ffd2  6c 26 03    jmp ($0326); CHROUT
ffd5  4c 9e f4    jmp $f49e; LOAD
```

```
ffd8  4c dd f5     jmp  $f5dd; SAVE
ffdb  4c e4 f6     jmp  $f6e4; SETTIM
ffde  4c dd f6     jmp  $f6dd; RDTIM
ffe1  6c 28 03     jmp  ($0328); STOP
ffe4  6c 2a 03     jmp  ($032a); GETIN
ffe7  6c 2c 03     jmp  ($032c); CLALL
ffea  4c 9b f6     jmp  $f69b; UDTIM
ffed  4c 05 e5     jmp  $e505; SCREEN
fff0  4c 0a e5     jmp  $e50a; PLOT
fff3  4c 00 e5     jmp  $e500; IOBASE
-------------------------------------------------
Pocatecni Pismena jmen autoru predeslych Progra-
mu a vektory Pro IRQ/BRK; RESET a RESTORE (NMI)
fff6 52 52 42 59 43 fe e2 fc; rrby;BRK/IRQ;RESET
fffa 48 ff                  ; RESTORE (NMI)
```

Z výpisu Překladače Basicu a Kernalu je vidět, jak je vše velice Pečlivě a do detailu ProPracováno. Oni to také nepsali žádní amatéři.

Věřím, že jen málokdo z nás má teď na jazyku otázku, k čemu že nám ten výpis je. Nicméně takováa otázka padnout může a Proto se na ní Pokusím odPovědět.

Již jsem naznačil,že všechny Programy uložené v ROM jsou napsány špičkovým zPůsobem.A Pokud máme v ruce jejich výPis,pak jsme na nejlepší cestě napsat něco v Podobném stylu.Myslím,že lepší studijní literaturu těžko někde seženeme.

Z výPisu můžeme Poměrně snadno odhadnout, jak jednotlivé Programy fungují a kdy Přicházejí ke slovu. Tím máme jednak umožněno jejich využívání a pak také máme jasnou Představu o tom, co všechno si můžeme dovolit.A Pokud se nám něco nelíbí a rádi bychom to vylePšili,pak můžeme buď napsat svojí rutinu a na její Počátek nastavit Příslušný vektor,PoPř. PřekoPírovat Basic (Kernal) ROM do Paralelní RAM, Příslušné části Programů (nebo dat) Pozměnit a Pak na adrese $0001 PřePnout na RAM.

Myslím,že bez znalosti Programů a rutin obsažených v ROM se obejde jen málokdo,Proto doPoručuji Pečlivé Pročtení výPisu a jeho doPlnění vlastními Poznámkami.

KAPITOLA VI

Snaž se naučit něco o všem
a všechno o něčem.
(Thomas H. Huxley, 1876)

Teoretikové - pište zmizíkem!
(Karel Konrád, 1960)

Věda je pro tu,kdo se učí.
Poezie pro tu,kdo vědí.
(Josesh Pouk, 1851)

1)Tipy & triky

2)Tabulky

3)RS-232

4)Konektory a Připojování

5)Popis vybraných integrovaných obvodů

6)Literatura

Tipy & triky

Na několika dalších stránkách najdete, mezi uživateli tolik oblíbené,tipy a triky, které nejen podstatně ulehčují programování,ale mohou nám odhalit i něco z toho, o čem dosud nemáme ani tušení.Triky v této podkapitole jsou řazeny abecedně podle,doufám,že vhodně zvolených,hesel.

Blokování (v závorkách je jeden ze dvou způsobů odblokování; druhým způsobem je vypnutí a opětné zapnutí počítače)

a)Klávesy RUN/STOP: POKE 788,52 (POKE 788,49)
b)Kl. RESTORE+STOP: POKE 808,225 (POKE 808,237)
c)LISTování : POKE 775,1 (POKE 775,167)
d)Spuštění progr. : POKE (PEEK(43)+256*PEEK(44)-1),1 (POKE (PEEK(43)+256*PEEK(44)-1),0)
e)Prohlížení programu před jeho spuštěním

Protože víme,jak jsou programy uloženy,můžeme toho využít i zde. Za každou programovou řádkou je uložena 0.Za ní následuje další programová řádka. Na konci programu jsou,místo další řádky, uloženy nuly dvě.My si tedy zjistíme,jaké kódy, příslušející k další (řekněme druhé) programové řádce leží za koncem předchozí řádky (za nulou). Obě hodnoty si zapamatujeme a na jejich místo uložíme nuly.Po LIST přečteme jen první řádku,ostatní jako by neexistovaly.Teď nám ještě zbývá zařídit to tak, abychom program ještě někdy spustili. Po RUN počítač zpracuje jen první,viditelnou,řádku. V ní tedy musíme oba kódy přepsat na místo doplněných nul.A aby nebyly vidět,uděláme to třeba takhle:

```
10 POKE xxxx,xx:POKE yyyy,yy:REM"několik písmen T
   zapsaných bez uvozovek v inverzním režimu - kód
   DEL  Jméno programu"
```

Jak tuto techniku můžeme použít v praxi,to je vidět například z výpisu programu "Redaktůrek".

Hlášky (české)

Začátečníkům dělá občas potíže zapamatovat si anglické příkazy a hlášky. Uděleite pro ně program,

který jim v začátcích tuto práci ulehčí.Zkusme to třeba takhle (Pro začátek jen s Basicem):

1)PřekoPírujeme Basic ROM do Paralelní RAM

2)PřePíšeme anglické texty českými (u hlášek je nutno zachovávat jejich přesné uložení -místo i zakončení;u příkazů je třeba dodržet zPůsob zakončení a celková délka všech příkazů dohromady musí být stejná,nebo menší)

3)PřePneme na adrese $0001 z ROM do RAM

Jako příklad nechť Poslouží následující Program (nahradí READY.):

```
1 FOR X=40960 TO 49151:POKE X,PEEK (X):NEXT
2 FOR X=41848 TO 41853:READ A:POKE X,A:NEXT
3 POKE 1,48+6
9 DATA 67,79,80,65,75,63
```

InPut (bez otazníku)

a)10 POKE 19,64:INPUT X:POKE 19,0

b)10 OPEN 1,0:INPUT# 1,X:CLOSE 1

Klávesy

a)OPakovací funkce POKE 650,0 (Crsr,Del,SPace)
POKE 650,127 (žádná klávesa)
POKE 650,255 (všechny klávesy)

b)ZaPnutí inverzního režímu: 1) Rvs ON
2) POKE 199,1 (0)

c)Zákaz PřePínání Grafic. módů: POKE 657,127 (0)

d)Mazání vyrovnávací Paměti kl.:POKE 198,0

e)Velikost vyrovn. Paměti kl. : POKE 649,1...10

Kurzor

a)Rychlost blikání: POKE 56325,5 (maximální)
POKE 56325,255 (minimální)
POKE 56325,50 (normální)

b)Umístění kurzoru: POKE 211,sloUpec:POKE 214,řád.
SYS 58640:PRINT"OK"

c)Blikání Při GET : POKE 204,0 (POKE 204,1)

Load/Verify/Save

a)Název,Počáteční a koncová adresa Progr. na Pásce
PEEK(829)+256*PEEK(830)...Počáteční adresa
PEEK(831)+256*PEEK(832)...koncová adresa

Jméno Programu:
FOR X=833 TO 833+15:PRINT CHR$(PEEK (X)):NEXT

b)Adresář kazety (žádný Program se nesmí jmenovat "qwertyqwertyqwer")
LOAD "qwertyqwertyqwer"

c)Autoload
Program nahraný v tímto způsobem se po opětovném zavedení do počítače např. sám spustí.Využívá se při tom tzv. sekundární adresa.Je-li nulová, data (nebo Program) se nahrají tam, kam právě ukazuje vektor $2b,$2c.Je-li nenulová, pak se Program načte tam,odkud byl uložen.Tímto způsobem je možné uložit i vektory.Takže, pokud máme Program uložen např. od $0800,můžeme to provést takto:
1)Přenastavíme vektor $2b,$2c na $0316
2)Od $07e8 uložíme následující Program pro "RUN"
LDX #$f1:LDY #$57:STX $0325:STY $0324:JSR $a68e
LDX #$00:JSR $a871:JMP $a7ae
3)Uděláme Program, který nastaví jméno souboru, aktuální Periférii, přenastaví $0324,$0325 na $07e8 a zavolá rutinu SAVE s Parametry:
X= $2d:Y= $2e:A= #$2b.Nakonec ještě musí přenastavit $0324,$0325 na $f157. Program se po LOAD sám spustí.

Pozadí (například ubíhající)

Jistě víš u některých her zaujal zajímavý efekt, který vypadá velmi složitě - ubíhající Pozadí.Skutečnost je však trochu jiná.Tato hra totiž pro zobrazení používá modifikovanou znakovou sadu a jedním ze znaků (třeba mezerou) vyplní Pozadí.A pak už není nic jednodušší, než matrici příslušného znaku Pomalu měnit.Někdy si to vyzkoušejte,výsledný efekt stojí za to!

Převod čísel v ASCII do bin. kódu a opačně

a) B= A AND 15 (A...ASCII,B...Binární kód)
b) A= B OR 48

Příkazy (zadávání např. z Programů ve stroj. kódu

Poměrně často by se mám,při práci v Assembleru,ho-

dily některé příkazy z Basicu. Jednou z cest, jak je používat je jejich ukládání do klávesnicového buffru (popř. do vstupního buffru Basicu). Podívejme se na následující program v Basicu, který lze jednoduše přepsat do Assembleru. Zadáme jím příkaz LIST:

```
10 FOR X=631 to 631+4:READ A:POKE X,A:NEXT
20 POKE 198,5:END
30 DATA 76,73,83,84,13:REM L,I,S,T,<RETURN>
```

Spojování dvou programů v Basicu

Pokud máme dva samostatné programy v Basicu s různými čísly řádek, můžeme je za pomoci magnetofonu, nebo diskety, spojit do jednoho. Uděláme to takto: (Pro počátek prvního programu $2b,$2c na $0801)

1)Nahrajeme první program do paměti počítače

2)X=PEEK(45)+256*PEEK(46):X=X-2:POKE44,INT(X/256):
  POKE43,X-PEEK(44)*256

3)Nahrajeme druhý program do paměti (za ten první)

4)POKE43,1:POKE44,8

Umístění paměti pro VIC (výhodné)

a)Znaková sada: 57344 ($e000) - 61439 ($efff)
  Obrazovk.RAM: 49152 ($c000) - 50151 ($c3e7)
  Color RAM : 55296 ($d800) - 56295 ($dbe7)
  POKE56334,0:POKE1,51:Přenést znaky:POKE1,55:POKE
  56334,1:POKE648,192:POKE53272,10:POKE56576,PEEK(
  PEEK(56576)AND252:PRINT CHR$(147)

b)Znaková sada: 40960 ($a000) - 45055 ($afff)
  (Bezchybně pracuje jen při odpojeném překladači)

Obrazovková RAM	X	53272	648
35840 ($8c00)	1	56	140
45056 ($b000)	1	200	176
46080 ($b400)	1	216	180
47104 ($b800)	1	232	184
48128 ($bc00)	1	248	188

Na adresy 53272 a 648 uložíme stanovené hodnoty, na 56576: POKE56576,(PEEK(56576)AND252)OR X. Před tiskem aktivujeme obrazovku: PRINT CHR$(147).

c)Grafická RAM: 57344 ($e000) - 65343 ($ff3f)
  Color RAM   : 49152 ($c000) - 50151 ($c3e8)
  POKE56576,PEEK(56576) AND 252:POKE53272,11:
  POKE53265,PEEK(53265) OR 32

Výměna rutiny Pro RESET,nebo RESTORE

Pokud jste dobře Prolistovali výPis Kernalu, Pak vám jistě neušla rutina Pro Porovnávání kódů CBM 80.Toho nyní využijeme.

U některých her se stává,že se Po stisku klá-vesy RESTORE znovu sPustí. Na tom není nic šPatné-ho.Horší je, když stiskneme RESET a Počítač na něj nereaguje buď vůbec,nebo úPlně jinak.Ale jsou tře-ba Programy (naPř. Monitory),které bychom rádi ně-jak takto sPouštěli.Jak na to?

1)Na adresy $8000,$8001 zaPíšeme nový vektor Pro RESET. Na $8002,$8003 zaPíšeme nový vektor Pro RESTORE.
2)Na adresy $8004-$8008 Přeneseme data ze $FD10--$FD14.
3)Po stisku RESTORE,nebo RESET se zakáže Přerušení a Po Proceduře se zásobníkem se testují námi za-Psané kódy.Jestliže souhlasí,Pokračuje běh Prog-ramu z místa,na které ukazují naše vektory.

Záchrana Programu Po NEW nebo Po RESETu

a)Pokud Program nekončil tak, že by Po Přičtení 2 k získané adr. došlo k Přenosu do vyššího Bytu :

```
POKE2050,1:SYS42291:POKE46,PEEK(35):POKE45,PEEK(78
1)+2:CLR
```

b)Pokud by došlo k Přenosu do vyššího Bytu:

```
POKE2050,1:SYS42291:POKE46,PEEK(35)+1:POKE45,PEEK(
781)-254
```

Věřím,že si mezi Popsanými tiPy & triky každý něco vybral.Ještě dodám, že všechny jsou Praktický ověřené.Mezi triky jsem záměrně nezařadil ty,které Pouze adresují Paměť (naPř.nastavení barvy znaků), Protože není Problém Přečíst systémové Proměnné a najít v nich to,co Právě Potřebujeme.

Tabulka kódů příkazů

Příkaz	Dec.	Hex.	Příkaz	Dec.	Hex.
END	128	$80	FOR	129	$81
NEXT	130	$82	DATA	131	$83
INPUT#	132	$84	INPUT	133	$85
DIM	134	$86	READ	135	$87
LET	136	$88	GOTO	137	$89
RUN	138	$8a	IF	139	$8b
RESTORE	140	$8c	GOSUB	141	$8d
RETURN	142	$8e	REM	143	$8f
STOP	144	$90	ON	145	$91
WAIT	146	$92	LOAD	147	$93
SAVE	148	$94	VERIFY	149	$95
DEF	150	$96	POKE	151	$97
PRINT#	152	$98	PRINT	153	$99
CONT	154	$9a	LIST	155	$9b
CLR	156	$9c	CMD	157	$9d
SYS	158	$9e	OPEN	159	$9f
CLOSE	160	$a0	GET	161	$a1
NEW	162	$a2	TAB(	163	$a3
TO	164	$a4	FN	165	$a5
SPC(	166	$a6	THEN	167	$a7
NOT	168	$a8	STEP	169	$a9
+	170	$aa	-	171	$ab
*	172	$ac	/	173	$ad
↑	174	$ae	AND	175	$af
OR	176	$b0	>	177	$b1
=	178	$b2	<	179	$b3
SGN	180	$b4	INT	181	$b5
ABS	182	$b6	USR	183	$b7
FRE	184	$b8	POS	185	$b9
SQR	186	$ba	RND	187	$bb
LOG	188	$bc	EXP	189	$bd
COS	190	$be	SIN	191	$bf
TAN	192	$c0	ATN	193	$c1

Příkaz	Dec.	Hex.	Příkaz	Dec.	Hex.
PEEK	194	$c2	LEN	195	$c3
STR$	196	$c4	VAL	197	$c5
ASC	198	$c6	CHR$	199	$c7
LEFT$	200	$c8	RIGHT$	201	$c9
MID$	202	$ca	GO	203	$cb

Jakýkoliv jiný příkaz s vyšším kódem je dekódován jako 'Syntax error'.

Tabulka kódů standardních periférií

Kód	Periférie	Kód	Periférie
0	klávesnice	1	magnetofon
2	RS-232	3	obrazovka
4 - 7	tiskárna	8 - 15	disk. jedn.

Tabulka kódů barev

Kód	Barva	Kód	Barva
0	černá	1	bílá
2	červená	3	zelenomodrá
4	purpurová	5	zelená
6	modrá	7	žlutá
8	oranžová	9	hnědá
10	sv. červená	11	šedá 1
12	šedá 2	13	sv. zelená
14	sv. modr	15	šedá 3

RS-232

Pokud budeme k počítači připojovat nějakou periférii, použijeme buď PORTy 1 a 2, sériovou sběrnici, nebo standardní rozhraní RS-232. Protože je RS-232 velice oblíbené,podíváme se na něj trochu zblízka.

RS-232 je plně programovatelný vstup/výstup. Pracuje v logických úrovních TTL 5V a jednotlivé linky můžeme zatížit proudem až 3,2mA.

Linky PB0-PB7 můžeme nadefinovat jako vstupní nebo výstupní.Každé lince odpovídá jeden bit v registru $03 (leží na adrese 56579 tj. $DD03). Má-li např. linka PB0 sloužit pro vstup, nastavíme 0-tý bit řídícího registru na 0.Má-li linka být výstupní,pak se příslušný bit nastaví do 1.

Informace,která na lince je,ať už při vstupu, nebo při výstupu,se odráží v registru $01 CIA#2 na adrese 56577 ($DD01).

Abychom nemuseli dělat složité rutiny pro obsluhu ve standardním (časovaném) režimu,máme v paměti ROM některé již hotové.Ty používá i Basic.Jejich obsluha je podobná,obsluze příslušných příkazů.Podívejme se tedy na obsluhu RS z pohledu Basicu.

Abychom mohli s RS-232 nějakým způsobem komunikovat,je třeba nejprve otevřít příslušný soubor.

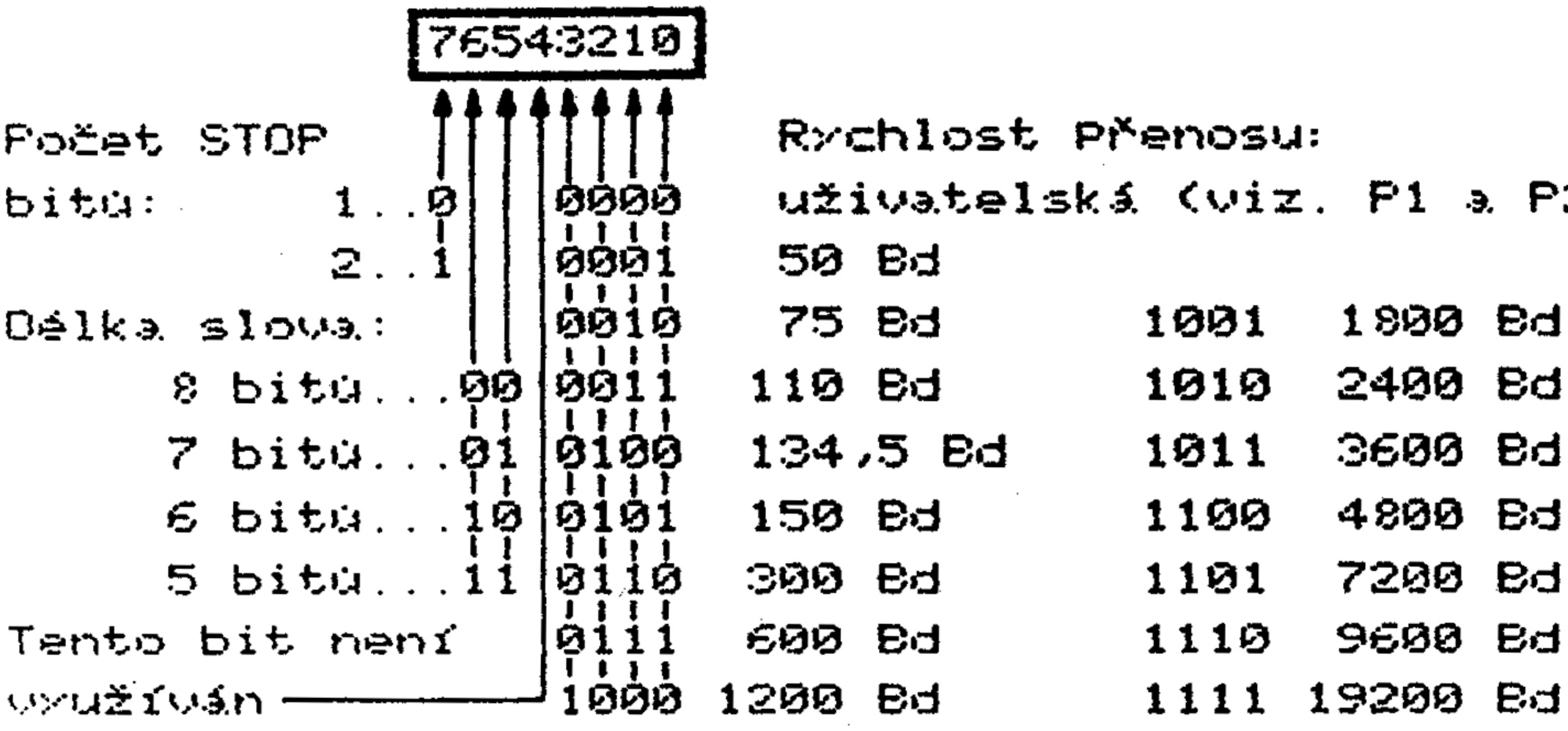
```
OPEN čs,2,0,CHR$(nast.registr)+CHR$(Povelový rg.)+
+CHR$(P1)+CHR$(P2)
čs...číslo souboru pro RS-232
P1,P2...Parametry přenosu (Používají se jen tehdy,
        Pracujem-li v nestandardní rychlosti)
```

V nastavovacím registru nastavujeme rychlost přenosu v Bd,délku slova a počet stop-bitů. V Povelovém registru nastavujeme handshake,duplex a paritu (Pomocné údaje pro přenos).

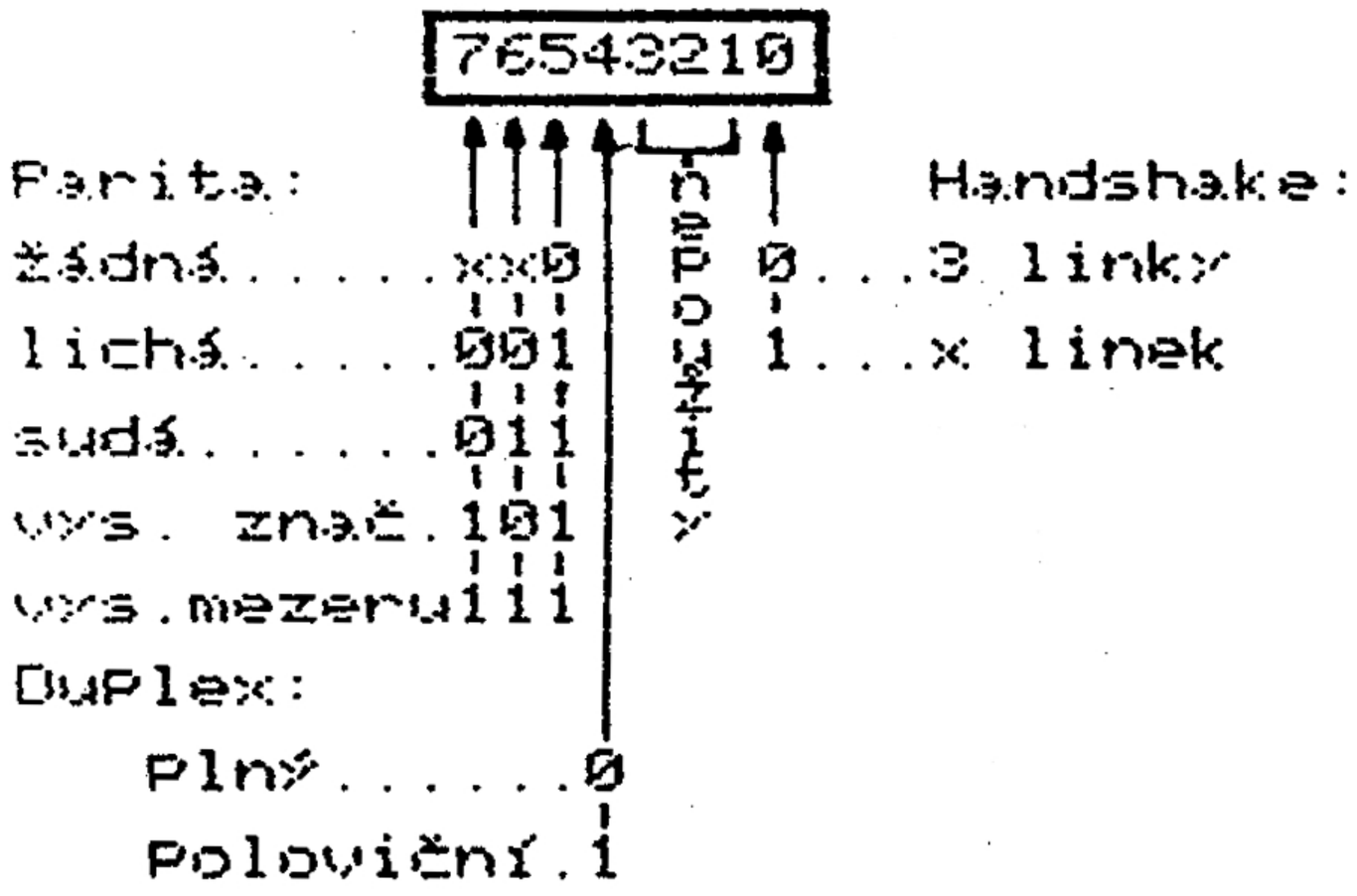
Jak nastavit Parametry P1, 2 pro nestandardní rychlosti? Jedná se o Lo,Hi hodnoty,které získáme:

```
P1=985250/(rychl./2)-100-P2*256     ;rychlost dosa-
P2=INT((985250/(rychl./2)-100)/256) ;zujeme v Bd
```

Nastavovací registr



Povelový registr



Pro vysílání a příjem můžeme použít příkazů PRINT#,INPUT#,GET#,nebo rutin CHKOUT,CHROUT,CHKIN, CHRIN,GETIN. Stavový registr RS-232 si můžeme přečíst Pomocí standardní Proměnné STATUS (ST), nebo Pomocí rutiny READST.

Použití RS-232 vřele doPoručuji. Je to také jedna z mnoha Předností naší "šedesátčtyřky". Jenom je třeba dát si Pozor na to, abychom linky příliš nepřetěžovali.

Konektory a připojování

Pro spojení počítače s okolím (magnetofonem,disketovou jednotkou,apod.) slouží připojovací konektory.Jak jsou zapojeny,to je uvedeno celkem podrobně v uživatelském manuálu. Zaměřme se spíše na jejich praktické použití.

COTROL PORT 1

Všechny datové vývody tohoto portu jsou standardně nastavovány pro vstup.Tento port se obvykle použ-ívá k připojení joysticku. V jaké poloze se právě nachází,to nám ukazuje adresa 56321 ($DC01). Vývod č. 6 tohoto portu můžeme použít pro připojení světelného pera. Jeho polohu ukazují adresy 53267 a 53268.Pokud bychom je četli z Basicu,dostaneme poměrně nepřesné hodnoty. Vývody 5 a 9 slouží k připojení analogového joysticku. Jsou přes analogový přepínač (viz. $DC00) přivedeny na vstup dvou nezávislých A/D převodníků. Příslušné hodnoty čteme na adresách 54297,54298 ($D419,$D41A).Tyto převodníky lze s úspěchem použít i pro digitalizaci zvuku.

Mezi vývody 7,8 je vyvedeno napětí +5V.Protože jsou mi důvěrně známy kvality zdrojů dodávaných k C=64, doporučuji povolenou zátěž 100mA jen krátkodobě.

CONTROL PORT 2

Tento port je velice podobný svému bratříčkovi.Polohu joysticku čteme na adrese 56320 ($DC00).Vývod č. 6 slouží pouze pro připojení tlačítka joysticku a pro A/D převodníky platí totéž,co u portu 1.Ještě upozorním na to,že CELKOVÁ zátěž 5V vedení (tedy součet proudových zátěží na všech vývodech +5V) je max. 100mA.

EXPANSION PORT

Do tohoto portu jsou vyvedeny všechny potřebné a důležité signály. Najdeme tam datovou a adresovou sběrnici apod. Jednotlivé vývody můžeme zatěžovat max. jedním TTL členem.Pozor na poškození ! ! !

AUDIO/VIDEO

Do tohoto konektoru jsou vyvedeny všechny potřebné signály pro zpracování obrazu. Vývod č.3 slouží pro výstup audio-signálu (zvuk) a vývod č.5 naopak pro vstup (viz. SID). Pracuje s běžnými nf úrovněmi.

SERIAL

Tento konektor je určen pro sériovou komunikaci s okolím. Většina vývodů je oddělena invertory, takže jsou poměrně odolné proti přetížení. Obsluhují se přes CIA#2 např. pomocí zabudovaných rutin. Způsob komunikace není složitý, ale i přesto je velmi spolehlivý.

Pokud bychom k tomuto konektoru chtěli připojovat svou nestandardní periférii, je třeba dodržet standardní časové režimy.

USER PORT

Tomuto portu jsme se věnovali v předešlé podkapitole. Ještě jsme nemluvili o vyvedeném RESETU a napájecím napětí.

RESETovací tlačítko je nezbytné. Pokud bychom totiž často vypínali a znovu zapínali počítač, mohli bychom ho za chvíli dát do opravy. Existují totiž i "experti", kteří za hodinu vypnou a zapnou počítač i dvacetkrát. Tak tohle radši opravdu ne, to je lepší pořídit si za tři koruny tlačítko a jeden jeho vývod připojit na GND (nulový potenciál, zem) a druhý na vývod č. 3 (RESET).

Pro vyvedené napětí +5V platí totéž co pro Control Porty - nepřetěžovat. Mezi výwody 10 a 11 je vyvedeno střídavých 9V. Ty lze celkem bez obav zatížit i povolenými 100mA.

CASSETTE

Tento konektor je určen pro připojení magnetofonu. Pokud bychom se rozhodli s ním nějak manipulovat, bude lepší, když si před započetím vlastní práce pořídíme sáček s procesory a sokl, abychom častými výměnami příliš neponičili plošný spoj. Jinak je použití tohoto portu poměrně snadné.

Popis vybraných integrovaných obvodů

Základem C=64 je Procesor,začněme tedy od něj.

MikroProcesor 6510

Zvláštnosti:

* 8 bitový obousměrný I/O Port (logika TTL)
* 8 bitová datová sběrnice
* 16 bitová adresová sběrnice
* 56 instrukcí
* 13 způsobů adresace Paměti
* zásobník s max. délkou 256 B a nast. ukazatelem
* Přerušení
* Přímý Přístup do Paměti (DMA)
* sběrnice slučitelná s M6800
* 1MHz,nebo 2MHz takt

Technické údaje:

-napájecí napětí max. +7V
-Proudová spotřeba 125mA
-zatížitelnost max. 1,6mA
-Pracovní teplota 0..70st.C

Complex Interface Adapter (CIA) 6526

Zvláštnosti:

* 8 bitová datová sběrnice
* 2 osmibitové Porty
* 2 Programovatelné 16 bitové časovače
* 24 hodinový mód s Programovatelným budíkem
* 1MHz,nebo 2MHz takt

Technické údaje:

-napájecí napětí max. +7V
-Proudová spotřeba <100mA
-zatížitelnost max. 3,2mA
-Pracovní teplota 0..70st.C

Video Interface Controller (VIC) 6567

Popisovat tento obvod není nic snadného. Jedná se totiž o špičkový zakázkový obvod, který má tolik předností,že by nebylo snadné je vyjmenovat.Ostatně myslím, že už jste si o něm udělali svou představu.Jediné, co snad dodám je jeho napájecí napětí -max. +7V.

Sound Interface Device (SID) 6581

Zvláštnosti:

* 3 tónové generátory 0-4kHz
* 4 různé průběhy
* 3 amplitudové modulátory (48dB)
* 3 generátory obalové křivky
  Attack...2ms-8s
  Decay....6ms-24s
  Sustain..0 až po přenastavení GATE
  Release..6ms-24s
* synchronizace a modulace jednotlivých oscilátorů
* programovatelné fitry (12dB/oktávu)
* rozmítací generátor
* vnější zvukový vstup (filtr+směšování,nebo pouze směšování s generovaným signálem)
* dva nezávislé A/D převodníky
* dvojí napájecí napětí +5V a +12V

Podrobnosti o jednotlivých obvodech lze vyčíst z firemních katalogů,které vám na požádání příslušná prodejní firma zašle (občas i zdarma). Ceny těchto obvodů postupně klesají, nicméně myslím, že jejich použití je poměrně výhodné. Někdy slyším slova, že osmibitová technika je už dávno za námi, ale není to pravda. Je pravda,že více než 8-mi bitové počítače jsou (většinou) lepší,nicméně jsou složitější a pro uživatele méně přístupné. V takových podmínkách se dobře daří virům a jiné havěti.Kromě toho, čím je počítač složitější, tím větší je pravděpodobnost poruchy a tím i chyby. Navíc osmibitová technika nejsou jen osmibitové osobní počítače.Patří sem např. různé řídící a regulační obvody, kde je použití drahých součástek zbytečným luxusem.Můj názor je,že osmibitová technika se bude ještě velmi dlouho používat.A s ní i osmibitové počítače.

Dělejte tedy dobré,nejlépe české nebo slovenské, programy a nenechávejte si je pro sebe. Třeba je prodávejte, ale nedávejte je do šuplíků ! ! !

Hodně zdaru.

300

Literatura

Nakladatelství	Jméno knihy (sešitu), autor, Pozn.
Addison-Wesley	"Basic and the Personal Computer" (Dwyer, Critchfield)
Commodore Büro-maschinen GmbH	"Alles Uber den Commodore 64" (Basic, Assembler, Popis rutin, zvuku, obrazu a mnoho dalšího)
Compute	"Compute's First Book of PET/CBM"
Coubay Comput-ing	"Feed me, I'm Your PET Computer" (Carol Alexander)
Dilithium Press	"BASIC Basic - English Dictionary for the PET" (Larry Noonan)
Faulk Baker Associates	"MOS Programming Manual" (MOS Technology)
Hofacker	"8K Microsoft BASIC Reference Manual"
	"Expansion Handbook for 6502 and 6800"
	"Small Business Programs"
	"Programming in 6502 Machinelangu-age PET/CBM"
	"The Great Book of Games" (64 Programu Pro C=64)
	"More on the Sixtyfour" (Tipy a triky, zajímavé PodProgra-my)
	"How To Programm Your C=64 in Ma-chine Language" (Podrobný úvod do Programování ve stroj. kódu Pro Basic-áre)
	"Commodore 64 Tune-Up" (Jak rozšiřovat C=64)
	"29 Programme fur den Commodore64" (2-D Plot, 3-D Plot, statistické výpočty, tisk navštívenek, obálek, kalendář, atd. - vše v Basicu)
Osborne/McGraw-Hill	"PET Fun and Games" (R. Jeffries, G. Fisher)
	"PET and the IEEE" (A. Osborne, C. Donahue)
	"Some Common Basic Programs for the PET" (L. Poole, M. Borchers, C. Donahue)
Wydawnictwa Na-ukowo-Techniczne -Mikrokomputery	"Commodore 64" (Bohdan Frelek) (Velmi pěkná knížka. Shrnuje Ba-sic, systémové proměnné, popisuje příkazy programů rozšiřujících Basic. Věnuje se jednotlivým in-strukcím 6510, popisuje práci s Monitorem, zvuk, obraz, způsob ulo-žení dat na pásku a disketu, atd.)

OBSAH

Úvod 5
Kapitola I 7
1)Na co můžeme C=64 Používat 9
2)Jak zacházet s Počítačem 11
Kapitola II 13
1)Práce s klavesnicí 15
2)Proměnná 17
3)Příkazy 19
4)Priority operací 33
5)Jednoduché Programy v Basicu 35
Kapitola III 53
1)Základní technické Pojmy 55
2)Systémové Proměnné 57
3)Uložení Basicu a Proměnných 73
4)Organizace Paměti 79
5)Obraz 81
6)Zvuk 87
7)Př. Programů na obsluhu obrazu a zvuku.. 89
Kapitola IV 107
1)Programování ve strojovém kódu 109
2)Popis registrů;značky 111
3)Popis instrukcí 113
4)Popis zabudovaných rutin 125
5)Příklady Programů v Assembleru 135
Kapitola V 143
1)Výpis Basic ROM 145
2)Výpis Kernal ROM 215
Kapitola VI 285
1)Tipy & triky 287
2)Tabulky 293
3)RS-232 295
4)Konektory a Připojování 297
5)Popis vybraných integrovaných obvodů ...299
6)Literatura 301

A kdybyste náhodou hledali Obsah a nemohli ho najít ani za kredencí, Pak zkuste str.

.....303