

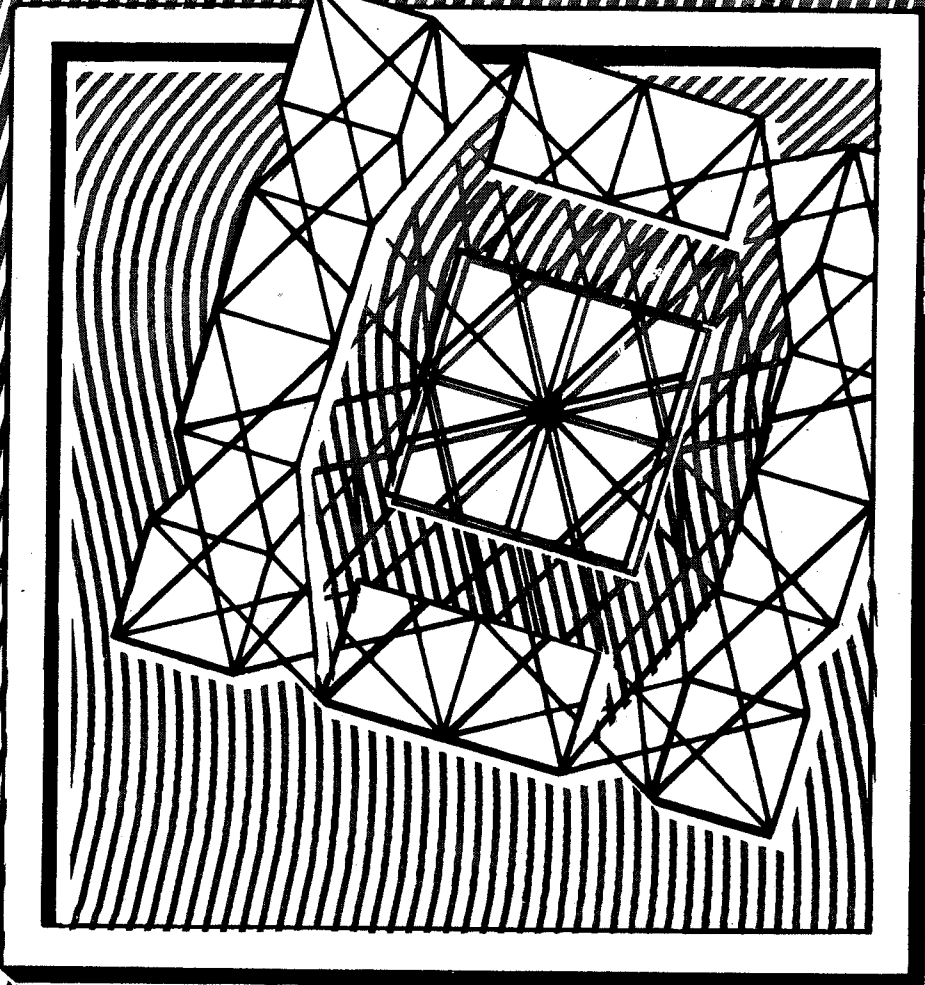


602

*COMMODORE*

1

88



## Programování ve strojovém kódu SELSON část 4.



Dnes se seznámíme se zvukovým procesorem, který je v našem C64 umístěn a zkráceně označován SID (Sound Interface Device). Tento zvukový procesor je velmi výkonný a k popisu všech jeho funkcí bychom spotřebovali více místa než máme k dispozici. Proto jen telegraficky.

Ovládání registrů SID je zprostředkováno 29 adresovými místy (54272-54300). Prvních 21 adres je rozděleno do třech stejných bloků určených k řízení jednotlivých hlasů procesoru (viz. základní příručka C64 Příloha O). HLAS 1 používá adresy 54272-54278 a každá adresa má svou speciální funkci. Obdobnou funkci mají registry použité pro ovládání HLASu 2 a HLASu 3, které začínají na adresách 54279 resp. 54286. Do všech uvedených paměťových míst lze pouze zapisovat např. příkazem POKE. Přečteme-li obsah paměti příkazem PEEK nebude přečtené číslo odpovídat obsahu registru.

Význam jednotlivých registrů pro HLAS 1 je následující:

### **Volba kmitočtu (54272 & 54273):**

Tyto dvě adresy mohou obsahovat číslo 0-65535. Vynásobíme-li toto číslo 0,0596 dostaneme frekvenci v Hz. Poznamenejme, že na adrese 54272 je méně významný byte.

### **Střída obdélníkového generátoru (54274 & 54275):**

12-bitů na těchto adresách je použito k volbě poměru délky pulzu obdélníkového generátoru. Bity 4 až 7 na adrese 54275 nejsou použity. Měníme-li číslo na těchto adresách z minima do maxima, měníme poměr mezi střídou a mezerou obdélníkového impulzu na výstupu generátoru.

### **Ovládání (54276):**

Všechny bity na této adrese mají svůj zvláštní význam, tak že je popíšeme jednotlivě.

#### **BIT 0 obálka zvuku (Gate):**

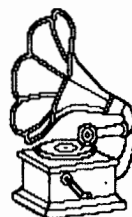
Abychom pochopili funkci tohoto bitu, musíme si něco říci o tvaru obálky zvuku. Obálku zvuku (průběh hlasitosti) tvoří náběh (Attack) až do maxima pokles (Decay), prodleva (Sustain) a doznění (Release). Zvuková obálka se proto často nazývá ADRS. Délky jednotlivých úseků volíme na jiných adresách. Na této adrese pouze určujeme nastavením bitu 0 do hodnoty 1 zapnutí programové obálky, nebo její vypnutí (bit 0 = 0) tj. hlasitost pak pouze klesá z maxima do minima.

**BIT 1 synchronizace (Sync):**

Nastavení tohoto bitu do hodnoty 1 spouští synchronizaci HLASu 1 HLASem 3.

**BIT 2 zámkový mód (Ring Mod):**

Nastavením bitu 2 do hodnoty 1 je možné kombinovat HLAS 1 s HLASem 3 je-li nastaven trojúhelníkový průběh zvuku.

**BIT 3 externí synchronizace (Test):**

Tento bit je určen k zapnutí externí synchronizace HLASu 1

**BIT 4 trojúhelníkový průběh (Triangle Waveform):**

Je-li tento bit nastaven na 1 má základní průběh na výstupu HLASu 1 trojúhelníkový tvar.

**BIT 5 pilovitý průběh (Sawtooth Waveform):**

Nastavením tohoto bitu do 1 je na výstupu HLASu 1 pilovitý průběh.

**BIT 7 bílý šum (White Noise):**

Nastavením tohoto bitu do 1 se na výstupu HLASu 1 objeví bílý šum, tj. hluk. Toto se využívá především k dosažení nemuzikálních zvukových efektů.

**Nárůst (Attack) a pokles (Decay) (54277):**

Horní nyble adresy (nyble = čtyři bity, 1 byte = 2 nyble = 8 bit) je určen k uložení čísla 0 až 15 k určení nárůstové části obálky. Dolní nyble určuje délku trvání poklesu.

**Prodleva (Sustain) a doznění (Release) (54278):**

Horní nyble opět určuje délku prodlevy obálky a dolní nyble délku doznění.

**Výstupní filtr (54293 až 54295):**

Tato tři paměťová místa umožňují filtraci výstupních signálů a budou popsány v příštích pokračováních.

**Hlasitost (54266):**

Bity 4 až 7 této adresy jsou rovněž využívány k předávání parametrů pro filtraci. Méně významný nyble (bity 0 až 3) však slouží k regulaci hlasitosti. Maximální hlasitost je 15, zvuk vypnut je 0.

**Jeden příklad za všechny:**

Subroutine:	GUNSHOT - výstřel
Adresa:	C003
Délka:	31 byte
Akce:	Výstřel.

Registry: Registry A a X jsou použity.  
 Status: Obsah stavového registru může být  
 změněn.

Pomocí příkazu A monitoru vložíme následující program.

```
A C003 LDX #$10
A C005 STX $D405
A C008 DEX
A C009 STX $D418
A C00C LDA #$09
A C00E STA $D406
A C011 LDA #$78
A C013 STA $D401
A C016 LDX #$81
A C018 STX $D404
A C01B DEX
A C01C STX $D404
A C01F RTS
```

Program je tak jednoduchý, že na jeho funkci jistě přijdete sami.

Příště si uvedeme programování dalších zvukových efektů.

**\*\*\* Konec 4.části. \*\*\***

### Zvukové znamení

Následující krátký podprogram může být použit v programech v BASICu, v kompilovaném BASICu nebo ve strojovém kódu jako zvuková indikace určitých akcí (vkládání dat, konec výpočtu ap.). Po spuštění je možno kdykoli v programu vyvolat jednoduše pomocí SYS 50000 pípnutí.

```
100 FOR J=50000 TO J+81: READ A: POKE J, A: NEXT
110 DATA 32, 133, 195, 169, 9, 141, 5, 212, 169, 30, 141, 15
120 DATA 212, 169, 130, 141, 1, 212, 169, 15, 141, 24, 212, 169
130 DATA 40, 141, 4, 212, 169, 7, 141, 163, 195, 32, 145, 195
140 DATA 169, 31, 141, 4, 212, 169, 20, 141, 163, 195, 32, 145
150 DATA 195, 32, 133, 195, 96, 162, 0, 138, 157, 0, 212, 232
160 DATA 224, 24, 208, 248, 96, 165, 162, 24, 109, 163, 195, 141
170 DATA 162, 195, 165, 162, 205, 162, 195, 208, 249, 96
```

## Winter Games

**Winter Games** (=Zimní hry) umožňují 1 až 8 hráčům soutěžit v 7 zimních sportech. Po natažení a spuštění proběhne zahajovací ceremoniál a objeví se **MENU**. Při výběru z následující nabídky použijte joystick.

- 1) **COMPETE IN ALL EVENTS** = Soutěžení ve všech disciplínách
- 2) **COMPETE IN ONE EVENT** = Soutěžení v jedné disciplíně
- 3) **PRACTICE ONE EVENT** = Trening jedné disciplíny
- 4) **NUMBER OF JOYSTICKS** = Počet joysticků
- 5) **SEE WORLD RECORDS** = Přehled světových rekordů
- 6) **OPENING CEREMONIES** = Zahajovací ceremoniál

Po zvolení soutěže zadají všichni soutěžící svoje jména a zvolí si zemi, kterou budou reprezentovat. Zadávání se ukončí stisknutím **RETURN** bez jména.

Následuje podrobný popis jednotlivých soutěžních disciplín:

### Krasobruslení – krátký program (Figure Skating – Short Program)

Jízda začíná po stisknutí spouště (tlačítka joysticku). Cvik se provádí tak, že nastavíte joystick do zvoleného směru a stisknete spoušť.

**Bruslení vpřed** umožňuje skočit **Dvojitého Axela** – joystick vpravo nahoru (získáte 0.6 bodu), **Trojitého Axela** – joystick vpravo dolů (1.1 b.) nebo se otočit a **bruslit pozadu** – joystick vpravo. Ke skokům se odrážejte s nohama maximálně nakročenýma.

**Jízda vzad** vám dovoluje skočit **Dvojitého Lutze** – joystick vlevo nahoru (0.6 b.), **Trojitého Lutze** – joystick vlevo dolů (1.1 b.), provést **vzpřímenou** resp. **skrčenou piruetu** – joystick nahoru resp. dolů (obě za 0.7 b.) nebo se otočit a **bruslit popředu** – joystick vlevo. Při piruetách provádějte přesně 6 otoček. Ke skokům se odrážejte v okamžiku, kdy jsou nohy u sebe. **Přechod z piruety vzpřímené do piruety v dřepu** je oceněn 1.2 bodu. Před dopadem po skoku a při ukončení piruety dejte joystick do střední polohy a stiskněte spoušť.

Celkem můžete získat 6.0 bodů. Za každý pád přijdete o 0.7 bodu, za každou chybu o 0.2 bodu. Na krátký program máte 1 minutu a je třeba končit jízdou vpřed.

### Volná jízda (Free Skating)

Cviky a jejich ovládání je stejné jako u krátkého programu. Cílem je provést třikrát každý ze sedmi cviků v časovém limitu 2 minuty. Bodová ohodnocení za každý úspěšný pokus jsou:

obě piruety - 0.3 bodu

pirueta vstoje s přechodem do piruety v dřepu - 0.5 b.

dvojitý Axel a Lutz - 0.2 b.

trojitý Axel a Lutz - 0.4 b.

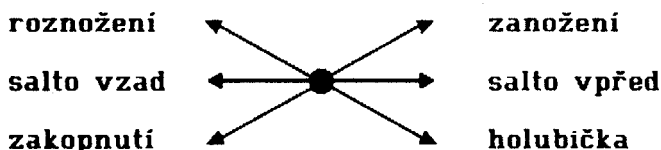
Za každý pád\* se odečítá 0.2 b. a za chybu 0.05 b. Součtem všech bodů vzniká celková známka, která určuje pořadí.

### Rychlobruslení (Speed Skating)

Soutěží dva závodníci v dráhách vedle sebe. Po zobrazení vašeho jména stisknete spoušť. Tím začne odpočítávání. Jakmile se objeví **GO**, začnete pohybovat joystickem do stran. Tajemství rychlosti spočívá v udržení správného rytmu odrazu a skluzu. O pořadí rozhoduje přirozeně čas.

### Hot Dog (akrobatické skoky na lyžích)

Spouští odstartujte skok. Poté nastavte joystick do jedné z následujících 6-ti poloh:



Cvik (s výjimkou salt) se provádí tak dlouho, dokud držíte joystick v uvedeném směru. Ukončíte ho tak, že joystick dáte do střední polohy. Při dopadu buďte v rovné poloze. Maximální skóre je 10.0 bodů. Za každou chybu přijmete o 1.4 bodu. Bodové zisky: 1 figura - 6.3 b., 1 salto - 7.2 b., 2 figury (stejně) - 8.7 b., 2 salta (stejná) - 9.2 b., 2 figury (různé) - 9.6 b., 1 figura a 1 salto - 10.0 b, 2 salta (různá) - 10.0 b.

Každý závodník má tři pokusy a pořadí závisí na bodovém zisku z nejlepšího skoku.

## Skok na lyžích (Ski Jump)

Po stisknutí spouště se vrhnete na rozjezdovou dráhu. Na odrazové hraně můstku stisknete znovu spoušť. V pravém horním rohu se objeví váš skokan a chcete-li získat maximum bodů za styl, musíte neustále udržovat jeho polohu. Stylové chyby opravíte následujícím způsobem:

- ▣ **pokrčené nohy** => joystick **nahoru**
- ▣ **náklon dopředu** => joystick **vlevo**
- ▣ **záklon** => joystick **vpravo**
- ▣ **zkřížené lyže** => joystick **dolů**

Skóre se vypočítá: **vzdálenost \* 3 + body za styl**.  
Vzdálenost závisí na okamžiku odrazu a na aerodynamice letu.  
Každý má tři pokusy.

## Biathlon

Odstartujte spouští a při **běhu na rovině** pohybujte joystickem vpravo a vlevo v rytmu lyžaře. Nejrychlejší **jízdu z kopce** docílíte odražením obouruč (joystick dolů). Při **běhu do kopce** pohybujte joystickem co nejrychleji do stran. Pozor však na srdce. Běží se 4 kola a v každém je **střelba** na 5 ran. Před výstřelem pohněte joystickem **dolů** (otevření komory), **nahoru** (zasunutí náboje). Poté stisknete **spoušť**. Za každý nezasáhnutý terč se k času přičítá 5 sekund. O vítězi rozhoduje celkový čas.

## Boby (Bobsled)

Po odstartování spouští řídíte svůj bob pohybem joysticku vlevo a vpravo. Na spodním okraji obrazovky je zobrazena vaše rychlost, která závisí na perfektním projetí každé zatáčky. O konečném pořadí rozhoduje čas v jedné ze tří jízdy.

Po ukončení poslední disciplíny budou vyhlášeni vítězové Zimních her. Konečné pořadí závisí na celkovém zisku bodů (každá zlatá medaile = 5 bodů, stříbrná = 3 body, bronzová = 1 bod). Pokud jste v některé disciplíně dosáhli **světového rekordu**, bude tento výkon spolu s vaším jménem zaznamenán na disketě. Přerušit hru a vrátit se do počátečního MENU můžete kdykoli stisknutím **RUN/STOP + RESTORE**.

## Tři nerovní bratři

Pod tímto názvem byl ve zvláštním čísle časopisu 64'er (8/86) uveřejněn zajímavý článek pro majitele mikropočítačů Commodore 16, 116 a Plus 4.

Zde je výtah podstatných informací:

Firma Commodore uvedla tyto počítače na trh již v roce 1984. Commodore Plus 4 stál tehdy 1298,- DM. Tato poměrně vysoká cena odradila případné zájemce. Také počítače Commodore 16 a 116 se prodávaly za cenu několikanásobně vyšší než dnes. Teprve od konce roku 1985 začaly ceny těchto počítačů klesat. Co se však mezitím stalo s Commodore Plus 4? Tento počítač zcela zmizel z pultů, aby se později navrátil jako hit s prodejní cenou pod 500,- DM včetně nové rychlejší disketové jednotky 1551. (Pozn.: Dnes aktuální cena počítače je 399,- DM.) Počítač je vybaven pamětí 64 kB RAM a vestavěným programovým vybavením (textový editor, kalkulační program apod.).

Počítače Commodore 16 a 116 se prodávají buď se standardní pamětí 16 kB RAM nebo s rozšířenou pamětí 64kB RAM. Tuto verzi s rozšířenou pamětí lze pořídit za cca 200,- DM.

Jsou všechny tyto počítače kompatibilní?

Lze pro ně použít shodný software?

Co se týká C 16 a C 116, lze říci, že v oblasti software jsou tyto dva počítače zcela kompatibilní. Liší se pouze vnějším uspořádáním, krytem, klávesnicí a prostorovým uspořádáním desek plošných spojů. Všechny elektronické části jsou zcela shodné.

Poměrně chudou paměť 16 kB lze snadno rozšířit na 64 kB RAM. Tím však mohou vznikat problémy s kompatibilitou softwaru. Některé programy, určené pro základní verzi C 16, nemusí fungovat ve verzi 64 kB RAM (tzn. v C 16 s rozšířenou pamětí RAM nebo v C +4). Jedním z problémů je nahrávání strojových programů, které musí být vždy uloženy na přesně určeném místě paměti. To lze jednoduše vyřešit tím, že budeme programy nahrávat se sekundární adresou 1. Tedy:

LOAD "jméno programu",1,1 ..... pro magnetofon

LOAD "jméno programu",8,1 ..... pro disc drive



Bez sekundární adresy se programy nahrávají na počátek basicové paměti, ale málokteré tam skutečně patří.

Jiný problém se objeví, používá-li se v programu grafického modu. V tomto případě se u verzi 64 kB RAM posune počátek basicové paměti z původní adresy 1000H (hexadecimálně) na adresu 4000H. Uvolněný paměťový prostor slouží pro uložení grafické obrazovky (cca 10 kB). Jinak je tomu u verze 16 kB, kde se ukládá grafická obrazovka na stejné místo jako u verze 64 kB, ale počátek basicové paměti zůstává na adrese 1000H. Pro program se pak využívá zbylý paměťový prostor od adresy 1000H do adresy 1800H, tedy cca 2 kB. Tento problém lze vyřešit softwarovým omezením paměti RAM. Horní hranice použitelné paměti RAM je uložena v oblasti systémových proměnných na adresách 55 a 56. Používáme-li paměť 64 kB, jsou zde uloženy hodnoty 0 a 253. U standardní verze C 16 s paměti 16 kB zde nalezneme hodnoty 246 a 63 (viz pozn. 1). Chceme-li tedy omezit paměť RAM na 16 kB, zadáme:

POKE 55,246: POKE 56,63

Zeptáme-li se nyní na počet volných bytů (PRINT FRE(0)), objeví se na obrazovce 12275 bytes free jako u základní verze C 16. Po stisknutí tlačítka RESET se obsah paměťových míst nastaví na původní hodnoty.

Při výměně programů mezi počítači C +4 a C 16 s rozšířenou pamětí nevznikají žádné problémy, tyto dva počítače jsou zcela kompatibilní.

Z originálu "Drei ungleiche Brueder" (64'er Sonderheft 8/86, str. 56) přeložil ing. Miloš Domorázek.

pozn. 1 - Horní hranice použitelné paměti RAM je uložena v paměťových buňkách s adresou 55 a 56 způsobem Lo-Hi. Skutečnou hodnotu získáme výpočtem:  $63 * 256 + 246 = 16374$  tedy 3FF6H (pro 16 kB RAM) a  $253 * 256 + 0 = 64768$  tedy FD00H (pro 64 kB RAM).

pozn. 2 - U některých programů, které využívají vlastní generátor znaků (např. Chess, Grandmaster ap.), je nutno před spuštěním změnit obsah paměťové buňky 65299 TED CHIP registru. U verzi s pamětí 64 kB RAM spustíme tyto programy příkazem: POKE 65299,16: RUN.

(J.S.)

## Počítač ! A co s ním ?

Pod tímto názvem bude v našem časopisu vycházet volný seriál, který je určen především začínajícím majitelům počítačů Commodore 16, 116 a Plus 4. Rozsah, odbornost i zaměření našeho seriálu bude záviset především na Vás, vážení čtenáři. Své dotazy podněty a připomínky zasílejte na moji adresu:

Jiří Strnad  
Jeseniova 17  
130 00 Praha 3

Všechny dotazy se budu snažit za pomoci svých kolegů na stránkách našeho časopisu zodpovědět. Bude-li problém obsírnější, je možno dané téma probrat na některé schůzce klubu.

Otázka první: Jaký počítač vlastně máme?

Commodore 16 je osmibitový mikropočítač s kapacitou paměti 32 KB ROM a 16 resp. 64 KB RAM.

Pro většinu z Vás jsou tyto údaje srozumitelné, ale vzhledem k tomu, že tento seriál je určen především našim začínajícím kolegům, rozebereme si úvodní údaje podrobněji.

Všechny uvedené parametry počítače se týkají **hardware**. Nejprve si vysvětlíme význam tohoto termínu. Hardware je veškeré technické vybavení počítače, všechny jeho integrované obvody, deska plošných spojů, klávesnice, display (TV přístroj), tiskárna, magnetofon, disketová jednotka ap.

Tato zařízení by samozřejmě nemohla fungovat bez souboru řídicích příkazů. Tento soubor příkazů nazýváme programovým vybavením neboli **software**. Software (program) je souborem příkazů srozumitelným našemu mikroprocesoru, který může být uložen buď na vnějším paměťovém médiu (magnetofonové pásce nebo disketě) nebo přímo v paměti počítače.

Nyní je třeba si vysvětlit jaké druhy paměti jsou použity v našem počítači.

Je to paměť **ROM** (z angl. Read only memory) tj. paměť, ve které je trvale uchováno základní programové vybavení počítače. Které programy to jsou si povíme později. Jak již

vyplývá z názvu lze z tohoto typu paměti informace pouze číst a nelze do ní zapisovat.

Druhým použitým typem je paměť **RAM** nebo přesněji RWM (z angl. Read/write memory). Do tohoto typu paměti lze informace zapisovat i je číst. Její nevýhodou je, že informace do ní vložená zmizí okamžitě po odpojení napájecího napětí.

Kapacita paměti je udávána počtem **byte** (čti bajtů), které je možno v této paměti uschovat. Když řekneme, že náš počítač je osmibitový, znamená to, že jeden byte je složen z osmi **bitů**. Vzhledem k tomu, že každý bit může nabývat dvou hodnot (0 nebo 1), zjistíme jednoduchým výpočtem ( $2^8 = 256$ ), že náš mikroprocesor je schopen rozeznat 256 různých informací (instrukcí nebo operandů - význam těchto termínů si vysvětlíme později). Má-li naše paměť ROM kapacitu 32 KB, znamená to, že je v ní uloženo asi 32 tisíc osmibitových informací. Proč asi 32 tisíc vyplývá ze způsobu adresování jednotlivých paměťových míst.

Náš mikropočítač používá pro adresování šestnáctibitovou adresovou sběrnici. To znamená, že mikroprocesor je schopen adresovat  $2^{16}$  tedy 65 536 adresových míst. Z toho vyplývá, že kapacita naší ROM paměti je přesně 32 768 B.

Závěrem prvního dílu našeho seriálu si ještě povězte, proč se pro označování adres i dat ve výpočetní technice používá zpravidla šestnáctkové soustavy. Toto pravidlo vychází z již výše uvedeného faktu, že základní jednotka informace - bit - může nabývat pouze dvou hodnot (0 nebo 1). Využívá se zde tedy dvojková soustava. Uznáte ale, že zápis adresy ve tvaru 1000101110010100 by byl značně nepřehledný. Další nepříjemnou skutečností je fakt, že dvojková soustava resp. mocniny čísla 2 nekorespondují s námi běžně používanou desítkovou soustavou (viz předcházející výpočty).

Využijme tedy skutečnost, že čtyřciferné číslo dvojkové soustavy resp. počet možných kombinací čtyř bitů je šestnáct. Z toho vyplývá, čtyřbitová informace může nabývat hodnoty od 0000 do 1111 ve dvojkové soustavě, což odpovídá hodnotě 0 až 15 v desítkové a hodnotě 0 až F v šestnáctkové soustavě. Hovoříme-li o adrese, která je šestnáctibitová, použijeme tedy čtyřciferného čísla šestnáctkové soustavy.

Jak jistě uznáte je přehlednější napsat, že naše paměť ROM začíná na adrese 8000H, než napsat, že paměť ROM začíná na adrese 32 768. O tom, co je v této paměti uloženo a jak toho při práci s počítačem využíváme, si povíme příště.

## KUDI 64

Kudi 64 je ideální program pro všechny matematické a geometrické problémy řešené pomocí základního diferenciálního počtu. Pomocí tohoto programu můžeme spočítat libovolnou derivaci zadané funkce, velice jednoduchým způsobem umožňuje grafické znázornění funkcí jedné proměnné, výpočet nulových bodů, lokálních maxim a minim, inflexních bodů.

KUDI 64 nám poskytuje celkem 9 příkazů rozšiřujících základní BASIC. Tyto příkazy se mohou používat jednak přímo, kdy se provedou po stisku klávesy RETURN, nebo je můžeme používat do BASIC - programů.

Odlišnost od standardního BASICu je při psaní programu u příkazu THEN. Jestliže napíšete za příkaz THEN některý z nových příkazů, počítač vám ohlásí »SYNTAX ERROR«. Proto se tyto rozšiřující příkazy musí oddělovat dvojtečkou, jako např..

```
50 IF A=1 THEN : DERIVE X
```

Hodnoty se do jednotlivých příkazů mohou zadat třemi způsoby:

- jako konstanty (např. HIRES 2)
- jako proměnné (např. HIRES A)
- jako výrazy (např. HIRES A+B)

U příkazu DERIVE, DRAW, DISCUSS a ADD se musí zadávat funkce pro proměnnou x. Při jejím zadávání se musíte dát pozor na správnou BASIC syntaxi.

Je možno používat všechny numerické funkce, které poskytuje standartní BASIC.

Dále je třeba dát si pozor, aby se u všech funkčních proměnných vyskytovala pouze proměnná »X«.

Délka funkčních proměnných je dána možnostmi překladače, tedy kupříkladu 80 znaků na jednom řádku. Dalším důležitým omezením je délka derivace (především u vyšších stupňů), která nesmí přesáhnout 255 znaků, jinak je hlášena chyba »Formula too Complex Error«. Některá další omezení jsou ještě popsána v dalším textu.

### Devět nových příkazů

**DERIVE:** Tento příkaz, je-li zadán bez parametrů v hranaté závorce spočítá první a druhou derivaci zadané funkce  $f(x)$ . Je-li tento příkaz zadán v direkt-modu, jsou první dvě derivace okamžitě vypsány na obrazovku.

Je-li příkaz DERIVE součástí BASIC - programu, můžeme k výpočtu derivací v bodech použít definovaných funkcí  $F_n$ , kde  $n$  je stupeň derivace. ( $F_0$  je tedy funkce sama,  $F_1$  je první derivace, atd.) Např.:

```
10 DERIVE X↑3
20 PRINT "F'(4)="; FNF1(4)    Výsledek po RUN: F'(4)=48
```

Mezi příkaz DERIVE a funkci můžeme do hranatých závorek vložit stupeň derivace, který chceme spočítat, případně nejnižší a nejvyšší stupeň. Např. Chceme-li spočítat druhou až pátou derivaci funkce  $F(X)=X^5$ , pak zadáme: »DERIVE [2,5] X↑3«.

Pozor: program nám spočítá libovolnou derivaci, pokud její délka nepřesahuje 255 znaků.

**DRAW:** Pomocí příkazu DRAW je funkce i se svými derivacemi nakreslena na obrazovce. I zde můžeme jako u příkazu DERIVE zadat libovolné stupně derivací, které chceme zobrazit. Tak např. »DRAW [0]f(x)« zobrazí jen základní funkci. Nezadáme-li hranaté závorky, je zobrazena daná funkce a její první a druhá derivace.

Kreslení na obrazovku se může kdykoliv přerušit stisknutím <RUN/STOP>. Po nakreslení celé funkce se do textového režimu vrátíme stisknutím jakékoliv klávesy, aniž by se nám vymazala grafická obrazovka (můžeme tedy do ní nakreslit další funkci). Chceme-li současně grafickou CRT vymazat, stiskneme <RUN/STOP>.

**COLOUR:** Příkazem COLOUR můžeme měnit barvy na CRT. Za příkazem následuje pět (nebo méně) čísel oddělených čárkami, označujících jednotlivé barvy (celkem je k dispozici 16 barev, jejich čísla jsou shodná s čísly v manuálu C 64). Je-li příkaz zadán bez parametrů, budou všechny barvy nastaveny jako při spuštění programu (viz následující tabulka).

1. parametr	barva zákl. funkce	černá
2. parametr	barva 1. derivace	bílá
3. parametr	barva dalších derivací a os	světle modrá
4. parametr	barva podkladu	tmavě modrá
5. parametr	barva rámečku	světle modrá

**CLEAR:** Tímto povelom se vymaže celá grafická obrazovka a všechny zaznamenané funkce.

**HIRES:** Pomocí tohoto příkazu můžete na CRT zobrazit

předešlou grafickou obrazovku (samozřejmě za předpokladu, že nebyla smazána). Za příkazem HIREs je parametr, který určuje, jak dlouho má být grafická obrazovka zobrazena. Parametrem je číslo mezi 1 a 255, přičemž jednotky odpovídají asi čtyři sekundy. Např.: HIREs 4 zobrazí obrazovku na 12 sekund. Z této obrazovky se může vystoupit i dříve stisknutím libovolné klávesy.

**COPY:** Tento příkaz vytiskne hardcopy grafické obrazovky na připojenou tiskárnu. Podmínkou je samozřejmě, že vaše tiskárna je schopná grafiku tisknout. Rutina pro tisk je napsána pro tiskárny typu Commodore MPS nebo s Commodore interfacem. Hardcopy je provedena v rozměru obrazovky, to je 320 \* 200 bodů.

Za příkazem COPY může následovat číslo, které udává, jak daleko od levého okraje papíru (počet znaků) se má hardcopy vytisknout. Takto můžete výtisk umístit do středu papíru.

**BORDER:** Pomocí příkazu BORDER zadáváte hranice, ve kterých má být daná funkce zobrazena na CRT. Příkaz může obsahovat až čtyři parametry (viz. tabulka).

1. parametr	levá hranice osy x	-8
2. parametr	pravá hranice osy x	+8
3. parametr	spodní hranice osy y	-5
4. parametr	horní hranice osy y	+5

V každém případě se však na CRT musí zobrazit souřadné osy (aby bylo kde odečítat hodnoty). To znamená, že např. interval <2,100> musíte zobrazit jako interval <0,100>, jinak je hlášena chyba.

Pro zadávání parametrů příkazu BORDER máte celkem pět možností:

**BORDER** (bez parametrů): všechny hranice se nastaví na počáteční hodnoty (viz. tabulka).

**BORDER a** (jeden parametr): levá hranice je nastavena na hodnotu a. Pravá hranice se nastaví posunutím celé obrazovky tak, že celkový rozsah CRT zůstane stejný jako předtím. (Pravá hranice bude tedy vždy o 16 jednotek větší, než jsme zadali parametr a. Y-ové souřadnice jsou nastaveny jako při zapnutí počítače.

*Petr Bacek*

(dokončení příště)

## TIPY & TRIKY PRO C64

**Zobrazení directory** - Následující krátký program umožní přečtení a zobrazení directory na obrazovce. Je výhodné využít ho jako podprogram ve vlastních programech. Výpis directory lze kdykoli přerušit stisknutím Q.

```
100 OPEN 1,8,0,"$": GET#1,A$,A$
110 GET#1,A$,A$: IF A$="" THEN 170
120 GET#1,B$,C$
130 BL=ASC(B$+CHR$(0)): BH=ASC(C$+CHR$(0))
140 D$=MID$(STR$(BL+256*BH),2)+CHR$(32)
150 PRINT D$,: GET#1,D$: IF D$<>"" THEN 150
160 PRINT: GET E$:IF E$<>"Q" THEN 110
170 CLOSE 1: END
```

**Počet volných bloků** - Program vypočítá počet volných bloků na disketě a hodnotu vypíše na obrazovce. Opět jej lze využít ve vlastních programech v basicu.

```
100 DEF FN A(X)=ASC(A$+CHR$(0))+256*(ASC(B$+CHR$(0)))
110 OPEN 1,8,0,"$0": GET#1,A$,B$
120 GET#1,A$,B$: IF FN A(X)=0 THEN 160
130 GET#1,C$,D$
140 GET#1,E$: IF ASC(E$+CHR$(0)) THEN 140
150 GOTO 120
160 A$=C$: B$=D$: F=FN A(X): CLOSE 1
170 PRINT F;"volnych bloku"
```

**"Directory" kazety** - Následující sekvence příkazů umožní vytisknout názvy všech souborů a programů nacházejících se na kazetě. Na pásce nesmí být soubor, který by se jmenoval 'XXXXX'.

```
OPEN 4,4: CMD 4: LOAD "XXXXX"
```

**Název, počáteční a koncová adresa programu na pásce.**  
Nahrajte program do počítače. A poté:

```
PRINT PEEK(829)+256*PEEK(830) - zobrazí startovací adresu
PRINT PEEK(831)+256*PEEK(832) - zobrazí koncovou adresu
```

název se vytiskne po vložení následující řádky:  
FORJ=833TO848: PRINT CHR\$(PEEK(J));: NEXT



## V příštím čísle:

Programování ve strojovém kódu, 5. část  
Basic a ty ostatní, 3.část  
Kudi 64 (dokončení)  
Počítač! A co s ním?, 2.část



Zavádíme **novou službu pro členy**. Na každé pondělní schůzce je možno vypůjčit si na týden diskety s programovým vybavením C64 (Print Fox, Vizawrite, Ace, F-15, H<sub>i</sub> Eddie+, Hypra Platos). Mimopražským členům budou diskety na požádání zasilány poštou.

**Hledáme** členy klubu ochotné pravidelně nebo jednorázově přispívat do zpravodaje. Příspěvky je možno předat na kterékoli schůzce nebo zaslat písemně na adresu klubu. Přednost dostávají články na kazetě či disketě ve formě souboru textového editoru Printfox nebo Vizawrite. Pokud uvedené programy nevladníte, můžete je získat na každé burze klubu nebo v nově zavedené výpůjční službě.

**Formátovací řádek** textového editoru Print Fox pro příspěvky do zpravodaje: z=10 x=80 l=510 y=40 h=2 v=2 s=1 g=0 t=25  
Maximální délka stránky po přeložení do grafiky je 720 bodů.

Toto číslo Zpravodaje bylo napsáno a editováno textovým procesorem PRINT FOX a přispěli do něj:

Ing. Konstantin Selucký (SELSON)  
Ivan Pavlíček (\*\*\* I.P. \*\*\*)  
Ing. Miloš Domorázek  
Jiří Strnad (J.S.)  
Petr Uacek

Commodore 602, technický zpravodaj pro mikroelektroniku a výpočetní techniku. Uydává 602. 20 Svazarmu pro potřeby vlastního aktivu, zodpovědný redaktor Ing. Pavel Bukovský, sestavení čísla Ivan Pavlíček. Adresa redakce: 602. 20 Svazarmu, Wintrova 8, Praha 6, 160 41. Telefon: 32 85 63. Povoleno UUTEI pod evidenčním číslem 87 006. Cena 12,- Kčs dle CCU č. 1030/202/86. Náklad 500 výtisků. Praha, duben 1988