

---

COMMODORE 64 / 128

uživatelská příručka

VIC - videočip C64



**COMMOTRONIC**

## V I C – Video Controller 6569

### 1.0 Pár slov úvodem

VIC je jedním z famózních obvodů řady 65xx. V počítači C-64 neplní jen všechny úkoly spojené s výrobou obrazu, nýbrž provádí k odlehčení práce procesoru také časování dynamických pamětí.

### 2.0 Technické údaje

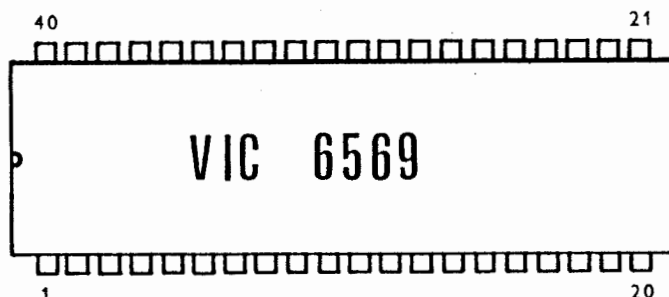
Práci a vlastnosti VICu v C-64 lze shrnout do následujících bodů:

- generuje 16 barev,
- generuje textovou grafiku 40 znaků/25 řádků,
- zabezpečuje HIRES grafiku s rastroem 320 x 200 bodů,
- má 5 způsobů provozu,
- ovládá 8 sprajtů po 24 x 24 bodech,
- vyrábí normovaný PAL TV signál,
- samostatně ovládá 16 KB dynamickou paměť,
- má posuvatelny generátor znaků,
- má posuvatelnu video-RAM.

Funkci obvodu dobře ukazují bloková schemata na obr.2 až 4 v příloze.

### 2.1 Zapojení obvodu

VIC je dodáván v pouzdru se 40 vývody, které mají následující zapojení:



**Obr.1** Zapojení pouzdra VIC 6569

## 2.2 Popis vvod VIC

- 1-7 D6 - D0, datov sbrnice procesoru.
- 8 IR0, LOW pokud odpovd jeden bit IMR a IRR.
- 9 LP, vstup light-pen-strobe.
- 10 CS, akce procesoru na sbrnici se konj pouze tehdy, kdy je CS=LOW.
- 11 R/W, LOW - pebrn dat ze sbrnice.
- 12 B, LOW pokud pi ten data jet nejsou pipravena.
- 13 VDD, + 12 V.
- 14 COLOR, informace o barv, vstup.
- 15 SYNC, řdkovn a obrazov synchronizan dlen.
- 16 ŔEC, LOW - VIC pouiv systmovou sbrnici, HIGH - sbrnice je voln.
- 17 Ŕ OUT, vstup systmovho taktu.
- 18 RAS, řdic signl pro dynamick RAM.
- 19 CAS, jako pedchoz.
- 20 GND, zem.
- 21 Ŕ COLOR, vstup frekvence pro barvy.
- 22 Ŕ IN, vstup frekvence bodu.
- 23 A11, adresov sbrnice procesoru.
- 24-29 A0/A8 - A5/A13, multiplexovan video-RAM adresov sbrnice.
- 30-31 A6 - A7, video-RAM adresov sbrnice.
- 32-34 A8-A10, adresov sbrnice procesoru.
- 35-38 D11-D8, data z barevn RAM.
- 39 D7, datov sbrnice procesoru.
- 40 Vcc, + 5V.

## 3.0 Popis registr VIC

Zkladn adresa VICu v C-64 je hexadecimaln (\$)D000, dekadicky 53248. To je adresa registru 0. VIC m k dispozici celkem 47 registr, kter jsou dle popsny.

**Reg. 0** X-souřdnice pro sprajtu 0.

Zadn pozice sprajtu 0 na obrazovce. Registr udv hodnotu pro vodorovnou souřdnici v hodnotch 0 - 255. Pro hodnotu nad 255 mus bt nastaven odpovdajc bit v registru 16.

**Reg. 1** Y-souřdnice pro sprajtu 0.

Zadn pozice sprajtu 0 na obrazovce, jeho svisl souřdnice v rozmez 0 - 199.

**Reg. 2 - 15** Souřdnice pro sprajty 1 - 7.

**Reg. 16** Petečn souřdnice X. Kdy m vodorovn souřdnice pekroit hodnotu 255, mus bt nastaven jeden bit v tomto registru.

Každmu sprajtu je v registru piřzen 1 bit. Sprajtu 0 bit 0, sprajtu 1 bit 1, atd.

- Reg. 17** Řídicí registr  
 Bit 0 - 2: posun obrazu (offset) nahoru/dolů.  
 Bit 3 = 1:25 řádků, = 0:24 řádků.  
 Bit 4 = 0: obrazovka vypnuta, = 1: obrazovka zapnuta.  
 Bit 5 = 1: Hires grafický mód.  
 Bit 6 = 1: rozšířený barevný mód.  
 Bit 7: Přetečení obsahu registru 18 (číslo rastrového řádku).
- Reg. 18** Rastrový řádek  
 Zde jsou ukládána čísla řádků rastru, při nichž proběhnutí paprsku řádkem může uvolnit IRQ. Přetečení tohoto registru je v registru 17
- Reg. 19** Světelné pero, souřadnice X.  
 X-ová souřadnice pozice na obrazovce, na které přišel signál od světelného pera (byl uvolněn jeho STROBE - pin LP=LOW).
- Reg. 20** Světelné pero, souřadnice Y.  
 viz registr 19.
- Reg. 21** Sprite enable - zapínání/vypínání sprajtů.  
 Zapínání a vypínání sprajtů. Každému sprajtu je přiřazen 1 bit (viz reg.16).
- Reg. 22** Řídicí registr 2.  
 Bit 0-2:posun obrazu vlevo/vpravo (offset zobrazení od levého kraje v bodech rastru).  
 Bit 3 =0: 38 sloupců, =1: 40 sloupců (znaků na řádek).  
 Bit 4 =1: vícebarevný mód.  
 Bit 5-7: nevyužito.
- Reg. 23** Sprite Expand X - zvětšení sprajtu ve směru X.  
 Sprajty se zobrazí ve dvounásobné šířce.  
 Každému bitu je přiřazen jeden bit (viz reg.16).
- Reg. 24** VIC Basic Adreses - adresy generátoru znaků a video-RAM (také pro HIRES bitovou mapu).  
 Bit 0 - nevyužit.  
 Bit 1-3: adresové bity znakového souboru (adresové bity 11-13).  
 Bit 4-7: adresové bity pro video-RAM (adresové bity 10-13).
- Reg. 25** Interrupt Request Register (IRR) - vymazání IRQ.  
 Bit 0 - uvolnění z reg. 18,  
 Bit 1 - uvolnění z ger. 31,  
 Bit 2 - uvolnění z reg. 30,  
 Bit 3 - uvolnění z pinu LP.  
 Bit 4-6 nevyužity  
 Bit 7 - nejméně 1 bit 0 - 4 je nastaven (je HIGH, pokud nejméně jeden z bitů 0 - 4 je HIGH).
- Reg. 26** Interrupt Mask Register (IMR) Určení, z jakých důvodů bude povoleno IRQ. Význam jednotlivých bitů odpovídá registru IRR. (Při souhlasu min. 1 bitu v registrech IRR a IMR bude pin IRQ=LOW.

- Reg. 27** Sprite Background Priority – priorita pozadí sprajtu.  
Každému sprajtu je přiřazen 1 bit. Pokud je bit = LOW, sprajt má přednost před pozadím, pokud je bit HIGH, pozadí má přednost před sprajtem.
- Reg. 28** Multicolor Sprites – vícebarevné sprajty.  
Každému sprajtu odpovídá 1 bit. Pokud je bit = HIGH, sprajt je zobrazován ve vícebarevném módu.
- Reg. 29** Sprite Expand Y – zvětšení sprajtu ve směru Y.  
Už reg. 23, zdvojnásobení výšky sprajtu.
- Reg. 30** Sprite-Sprite Collision – kolize dvou sprajtů.  
Každému sprajtu je přiřazen 1 bit. dotkne-li se jeden sprajt druhého, budou odpovídající bity HIGH. Současně se nastaví druhý bit IRR na HIGH a díky tomu se uvolní IRQ. Potom se musí tento registr vymazat, sám se zpět nepřeklopí.
- Reg. 31** Sprite-Background Collision – kolize sprajtu s pozadím.  
Stejně jako reg. 30. v případě, že se sprajt dotkne znaku pozadí.
- Reg. 32** Exterior Color – barva rámečku.
- Reg. 33** Background Color 0 – barva pozadí 0.
- Reg. 34** Background Color 1 – barva pozadí 1.
- Reg. 35** Background Color 2 – barva pozadí 2.
- Reg. 36** Background Color 3 – barva pozadí 3.
- Reg. 37** Sprite Multicolor 0 – barva sprajtů ve vícebarevném módu 0.
- Reg. 38** Sprite Multicolor 1 – barva sprajtů ve vícebarevném módu 1.
- Reg. 39 – 46** Color Sprite 0-7 – barvy sprajtů 0-7.

## 4.0 Zpusoby provozu VIC

VIC rozeznává 3 základní způsoby zobrazení. Znaky z generátoru, zobrazení jednotlivých bodů a zobrazení sprajtů.

### 4.1 Sprajty

Sprajty mohou měnit svoji pozici. K tomu slouží pár registrů. Rozlišení pohybu sprajtů činí 512 x 256 bodů. Díky tomuto velkému rozsahu se mohou sprajty pohybovat také z obrazu ven. Sprajt se může barevně zobrazit dvěma způsoby. Normální rozlišení je se dvěma barvami a poloviční rozlišení má čtyři barvy. V prvním případě má sprajt rozlišení 24 x 21 bodů. Jedna barva je přiřazena z registru Sprite Color, druhá barva je transparentní, t.j. je shodná s barvou pozadí. Ve druhém případě (mód Multicolor) má být nastaven odpovídající bit v registru 28. Sprajt má rozlišení 12 x 21 bodů. Dva body bitového vzorku sprajtu odpovídají jednomu bodu na obrazovce. Tyto dva body bitové mapy sprajtu určují barvu bodu následovně:

- 00 - transparentní,
- 01 - Multicolor 0,
- 11 - Multicolor 1,
- 10 - Sprite color.

Tím VIC ví, kde má vzít barvu pro sprajt. Pro každý sprajt existuje sprajt-pointer (ukazatel sprajtu). Těchto celkem 8 bodů (ukazatelů) obsahuje horních 8 bajtů ve video-RAM. Uynásobí-li se hodnota jednoho bajtu ukazatele hodnotou 64, obdrží se reálná adresa v paměti, kde má být bitová mapa sprajtu uložena.

## 4.2 Normální zobrazení znaků

Při tomto způsobu provozu si VIC vytáhne 1 bajt z video-RAM. Ten používá jako ukazatel na pozici v generátoru znaků. Bitová mapa uložená na tomto místě se nakonec objeví na obrazovce. Tímto způsobem může být zobrazeno 256 různých znaků.

Informace o barvě pro znak se nachází v barevné RAM. Pro každou pozici obrazovky jsou v barevné RAM 4 bity, jimiž se volí jedna ze 16ti barev. Tato barva však platí jen pro rozsvícený bod vzorku znaků. Body, které nejsou rozsvíceny mají barvu odvozenou z registru barev pozadí 0. Odtud mají také všechny znaky stejnou barvu pozadí.

## 4.3 Normální zobrazení znaků v módu Multicolor (Reg.22 bit 4=HIGH)

Při tomto způsobu provozu se nejprve přezkouší 4 bity barevné RAM. Pokud je nejvýznamnější bit LOW, zobrazí se znak v normální matici 8 x 8 bodů. Barva pro nastavené bity znakové matice bude určena zbylými třemi bity z barevné RAM.

Je-li nejvýznamnější bit HIGH, potom představují každé dva bity v bitové mapě znaku jen jeden bod. Znak se zobrazí v matici 4 x 8 bodů s dvojnásobnou šířkou bodů. Páry bitů určují nyní barvu dvojitého bodu.

- 00 - pozadí - registr barvy 0
- 01 - pozadí - registr barvy 1
- 10 - pozadí - registr barvy 2
- 11 - ostatní 3 bity z barevné RAM

V tomto případě může mít jeden znak současně 4 barvy.

## 4.4 Mód Extended Color (Reg.17, bit 6=HIGH)

Tento způsob provozu připomíná normální mód, neboť i zde každý znak může být jen ze dvou barev, avšak barva pozadí není nezbytně stejná pro každý znak.

Barva pro nastavené bity (HIGH) bitové mapy znaku pochází z barevné RAM. Pro nenastavené bity (LOW) je ale odvozena z obou nejvýznamnějších bitů video-RAM.

- 00 - pozadí - barevný registr 0
- 01 - pozadí - barevný registr 1
- 10 - pozadí - barevný registr 2
- 11 - pozadí - barevný registr 3.

Protože však nyní zbývá již jen 6 bitů jako ukazatel na bitovou mapu znaků, je možno vybírat jen z 64 znaků ( $2^6=64$ ).

#### 4.5 Normální mód jednotlivých bodů (reg.17, bit 5=HIGH)

V tomto provozním módu existuje přímá souvislost mezi pamětí a obrazovkou. Bit v paměti odpovídá bitu na obrazovce. K tomu se používá paměť o velikosti 8 KB. Normální video-RAM je zneužita pro funkci barevné RAM. Obvyklá barevná RAM není v tomto případě využita. Rozlišení činí  $320 \times 200$  bodů. Každý z bitů video-RAM dává informaci o barvě pro skupinu  $8 \times 8$  bodů a sice tím způsobem, že významnější nibble (půlbajt) pro nastavené bity a méně významný nibble pro nenastavené bity.

#### 4.6 Mód vícebarevných jednotlivých bodů.

(reg.17, bit 5=HIGH a reg.22, bit 4=HIGH)

Rozlišení zde činí  $160 \times 200$  bodů, protože každé dva bity paměti jsou přiřazeny dvounásobně širokému bodu na obrazovce.

- 00 - pozadí - registr barvy 0
- 01 - video RAM, významnější nibble
- 10 - video RAM, méně významný nibble
- 11 - barevná RAM.

#### 4.7 Spojení s procesorem

VIC je připojen na procesor jako normální periferní obvod. Bez omezení lze do něj zapisovat či z něj číst. Nicméně je zodpovědný za kompletní časování. Umožňuje přístup nebo jej zakazuje, obsluhuje dynamickou RAM atd. Toto je možné proto, že VIC vyrábí systémový takt. VIC proto v každém okamžiku ví, kdy se má konat jaký přenos. To je nutné vždy, když musí VIC komunikovat s video RAM, generátorem znaků nebo barevnou RAM. Toto cyklické přístupy musí být proto, aby se na obrazovce objevil souvislý obraz. Aby VIC nerušil procesor při jeho práci, využívá prodlev, kdy procesor nepoužívá sběrnice. Spojení je jednoduché. Vždy, když je  $\overline{\text{Q2}} = \text{HIGH}$ , obsazuje sběrnici procesor a při  $\overline{\text{Q2}} = \text{LOW}$  pak VIC.

Aby byla sběrnice uvolněna i fyzikálně, dává VIC pin AEC na nulu a potom procesor dává ze své strany sběrnici do stavu vysoké impedance (tri-state). Tímto způsobem jeden obvod neruší druhý.

#### 4.8 Spojení s RAM

K podrobnějšímu objasnění slouží blokové schéma na obr.2 a 3. VIC může adresovat 16 KB RAM. Proto musí, a to je důležité, při přesouvání z video RAM na generátor znaků oba chybějící adresové bity vzít na pomoc zvenci. Tyto bity jsou dány k dispozici na FA0-1 obvodu CIA2 (adresa \$DD00 - 56576). Pokud se tyto bity změní, posouvá se video RAM v 16K krocích. Tyto bity tedy představují adresové bity 14 a 15 video RAM. Za předpokladu, že je k dispozici dostatečně velká paměť, může se těmito bity přepínat více obrazových stránek. Nezapomeňte však, že tyto bity jsou aktivní ve stavu LOW. Pokud má být zvolena nejspodnější 16KB stránka, musí být oba bity CIA2 na HIGH. Pro nejvyšší stránku pak musí být oba bity LOW. V módu HIRES by mohla být případně bylo možno zobrazovat jednu stránku, zatím co druhá by se mohla připravovat. Video RAM lze posouvat také v jemnějších krocích po 1K pomocí registru 24. Bity 4 až 7 zde znamenají adresy RAM A10 - A13.

#### 4.9 Spojení s generátorem znaků.

Spojení s generátorem znaků pracuje obdobně jako spojení s RAM. Bity CIA2 přesouvají současně s video RAM také generátor znaků ve stejných krocích. Pomocí bitů 1 - 3 v registru 24 VIC lze generátor znaků posouvat ve 2K krocích. Tyto bity představují adresové bity A11 - A13. Díky adresovému manažeru získal znakový generátor v C-64 zvláštní status. Je oslovován VICem jen tehdy, pohybuje-li se relativní adresa v rozmezí \$1000 - \$1FFF nebo \$9000 - \$9FFF. Ve skutečnosti leží znakový generátor v adresovém prostoru procesoru od \$D000 a v adresovém prostoru od \$1000 se pouze zrcadlí. To znamená, že VIC v tomto adresovém prostoru nemá přístup k RAM, ale nepřímo na znakový generátor uložený od adresy \$D000. To však také znamená, že v tomto adresovém prostoru RAM nelze uložit žádné sprajty nebo jiná grafická data. To stejné platí samozřejmě i pro RAM v rozsahu \$D000 - \$1FFF.

Aby, čistě fyzikálně, znaková ROM neležela v oblasti přístupu všech video adres, musí se chybějící bity uložit přechodně do zvláštního bafru. Jedná se o adresové bity A0 - A7.

Báze znakové ROM v registru 24 hraje ještě další roli. Nepřifazuje totiž počáteční adresy generátoru pevně, nýbrž bit 3 určuje v módu jednotlivých bitů také polohu paměti obrazovky. Na bity 1 a 2 se v tomto případě nebere zřetel.

#### 4.10 Spojení s barevnou RAM

Barevná RAM je na rozdíl od video RAM a generátoru znaků nepohyblivá. Adresování může přicházet stejně jako u znakové ROM ze dvou směrů (od VICu nebo procesoru). Také zde chybějí VICu k adresování



bitů A0 - A7. Získají se stejným způsobem jako u generátoru znaků. S barevnou RAM spojuje VIC vlastní datová sběrnice. Prostřednictvím analogového spínače je odpojitelná od datové sběrnice procesoru. Celá záležitost funguje jednoduše:

Při AEC=LOW (VIC řídí bus) je spínač nastaven ve směru VIC. Při AEC = HIGH v jiném směru. dále jsou kontakty čtení/zápis do RAM barev řízeny mimo procesor adresovým manažerem PLA.

## 5.0 Programování barev a grafiky

U C-64 je hardwarově umožněno pohybovat po obrazovce až 8 volně definovatelnými sprajty. Každý z těchto sprajtů může být jinak zbarven. K dispozici je 16 barev. Případně jsou možné i vícebarevné sprajty (Multicolor), tvořené až ze tří barev. Sprajt obsazuje matici 24 x 21 bodů, srovnatelnou s blokem 3 x 3 písmen. U každém čase je možno kontrolovat, zda došlo ke kolizi sprajtů na obrazovce a kde se sprajty na obrazovce nacházejí.

### 5.1. Grafika a barva

U C-64 můžete volit ze 16 barev rámečku, pozadí a barvy znaků. Změna barev znaků je možná přímo z klávesnice pomocí kláves CTRL(CBM) + (1 až 8).

Barvy rámečku a pozadí se dají změnit změnou hodnot uložených na adresách:

barva rámečku - 53280,

barva pozadí - 53281,

pomocí příkazů POKE adresa, číslo.

Hodnoty čísel odpovídající jednotlivým barvám:

|                  |                    |
|------------------|--------------------|
| 0 - černé        | 8 - světle hnědá   |
| 1 - bílá         | 9 - hnědá          |
| 2 - červená      | 10 - růžová        |
| 3 - zelená       | 11 - tmavě šedá    |
| 4 - fialová      | 12 - šedá          |
| 5 - tmavě zelená | 13 - světle zelená |
| 6 - modrá        | 14 - světle modrá  |
| 7 - žlutá        | 15 - světle šedá.  |

Barva znaků se dá změnit také změnou čísla uloženého pro daný znak v barevné RAM. Barevná RAM má počáteční adresu 55296 a koncovou 56295. To znamená, že uložením čísla 0 na tuto adresu nám znak zapsaný v levém horním rohu obrazovky zčerná.

Provedeme to příkazem **POKE 55296,0**.

Pro každý z 1000 znaků, které lze na obrazovce zobrazit je tedy k dispozici jeden bajt.

Pro zobrazení znaků je však potřebná ještě další paměť, paměť obrazovky. U této paměti se nacházejí skutečné znaky bez údajů o barvě. Tento paměťový prostor leží od 1024 do 2023.

Ale C-64 neumí přece zobrazovat jen znaky. Má také jemnou grafiku, nazývanou HIRES, což je zkratka z anglického High Resolution - vysoké rozlišení. Programování jemné grafiky však není jednoduché. I namalování pouhé kružnice nebo čáry mezi dvěma různými obdvy je na C-64 mnohem komplikovanější než na mnoha jiných počítačích. Přes absenci vhodných grafických příkazů pro malování čar a křivek v Basicu V 2 dokáže se C-64 i s takovým problémem vypořádat. Navíc existuje pro C-64 řada programových doplňků k Basicu V 2, které tento problém elegantně řeší. Abychom si demonstrovali přístupným způsobem, jak programování jemné grafiky probíhá, uvedeme si dále příklad programu, který maluje sinusoidu.

```

10 REM malování sinusoidy
20 U=53248 : REM počáteční adresa VIC
30 AD=8192 : REM počáteční adresa HIRES bitové mapy
40 POKE U+17,59: REM zapnutí jemné grafiky
50 POKE U+24, 24: REM zapnutí jemné grafiky
60 FOR I= 1024 TO 2023 : REM nastavení barevné RAM.
70 : POKE I,16: REM číslo barvy
80 NEXT I
90 FOR I=8192 TO 16383 : REM vymazání bitové mapy pro HIRES
100 : POKE I,0
110 NEXT I
120 FOR X=0 TO 319: REM rýsování osy X
130 : Y=100: REM pozice osy X
140 : GOSUB 1000: REM volání zobrazovací rutiny
150 NEXT X
160 FOR Y =0 TO 199: REM rýsování osy Y
170 X=160: REM pozice osy Y
180 : GOSUB 1000: REM volání rutiny rýsování
190 NEXT Y
200 X=0
210 FOR I=-3.14159265 TO 3.1459265 STEP 0.0196349541
220 : REM hranice intervalu
230 : Y=100+99*SIN(I): REM funkce SINUS
240 : GOSUB 1000
250 : X=X+1
260 : NEXT X
270 : GOTO 270: REM tím se zabrání změně obrazovky
1000 OY = 320*INT(Y/8)+(Y AND 7): REM počítání bodů
1010 OX = 8*INT(X/8)

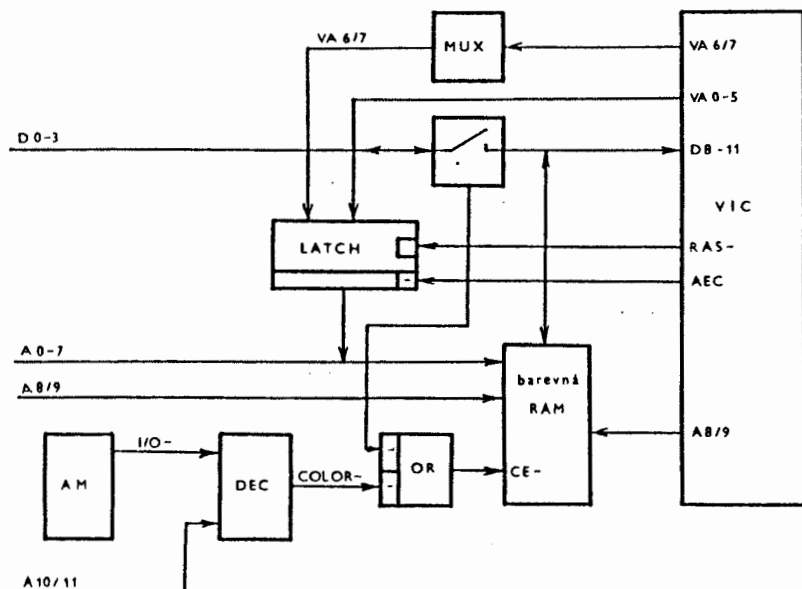
```

```

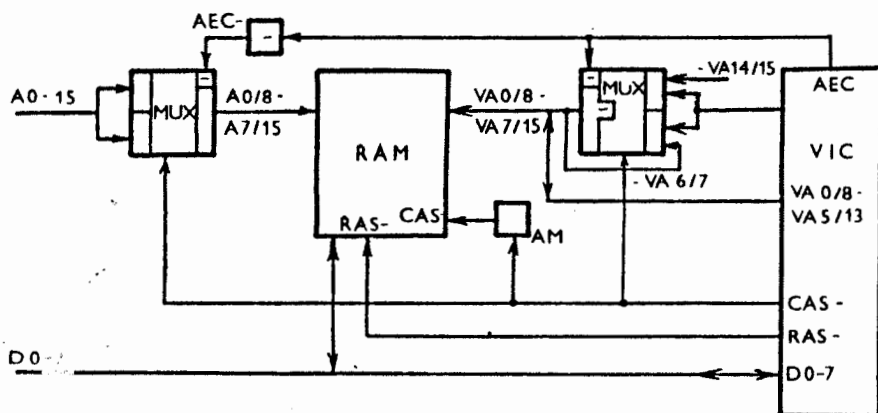
1020 MA = 2*(7-(X AND 7))
1030 AV = AD+0Y+0X
1040 POKE AV,PEEK(AV) OR MA: REM zobrazení bodu
1050 RETURN

```

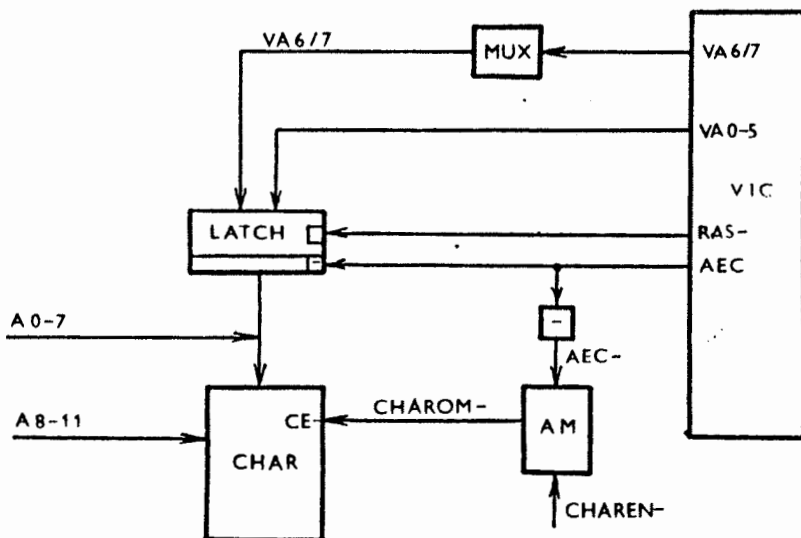
Fokud chcete po ukončení programu softwarově přepnout zpět do znakového módu, musíte nastavit zpět registry 17 a 24. K tomu je vhodné si tyto hodnoty před přepnutím do jemné grafiky uložit někde do paměti a při přepínání zpět na znakový mód je přenést zpět do uvedených registrů.



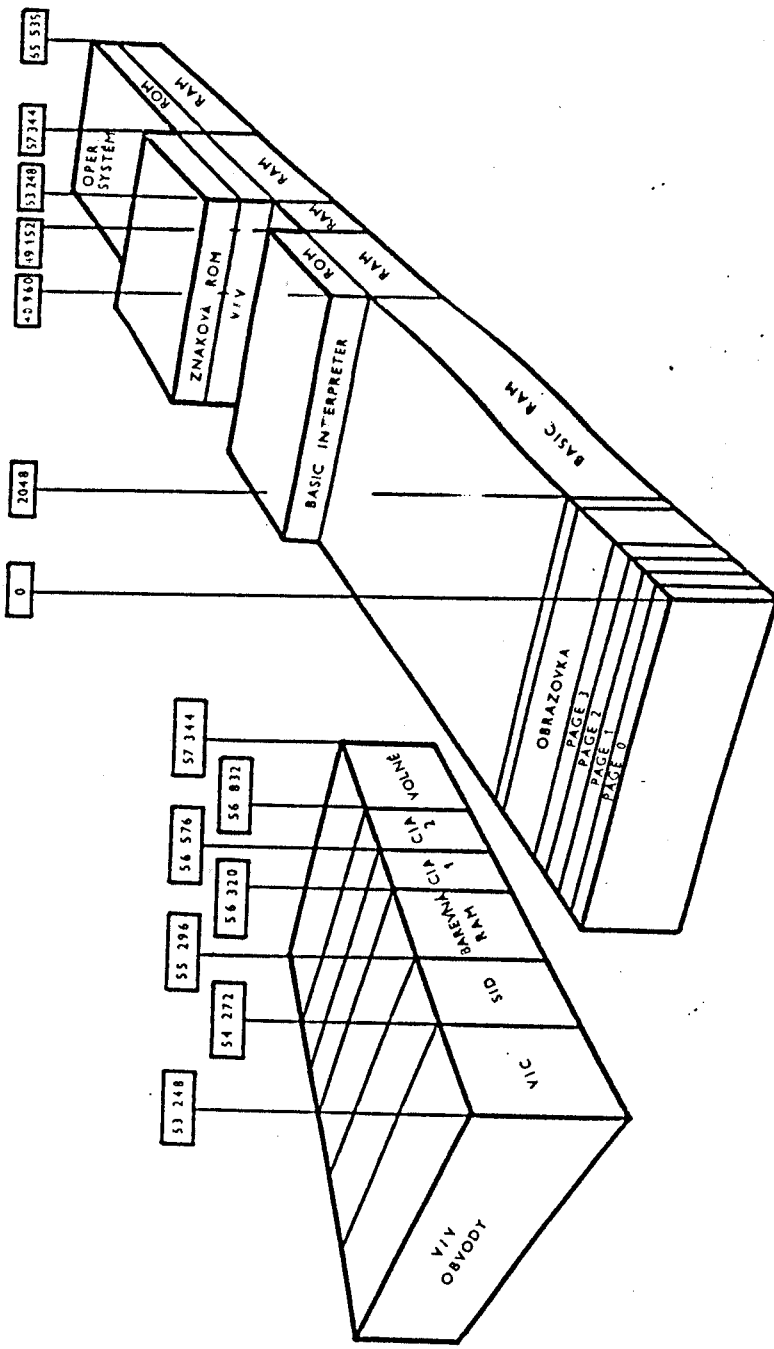
Obr. 2 Spolupráce VIC s barevnou RAM



Obr. 3 Spolupráce VIC s RAM



Obr. 4 Spolupráce VIC s generátorem znaků



Mapa paměti C - 64