

MILES GORDON TECHNOLOGIE

TECHNICKÝ MANUÁL SAM COUPÉ verze 3.0

Účelem tohoto manuálu je představit vývoj hardware a software k vnější a vnitřní práci na počítači SAM COUPÉ. Je to technický dokument předpokládající určité množství technických znalostí.

Je to pojistka MGT, že SAM Coupé a přidružené MGT produkty budou plně doloženy pro pomoc třetí strany vývoje hardware a software, budou plně výhodné všechny postupné rysy. Jestliže jste objevili výrobek Coupé a požadujete další objasnění nebo výklad, technický štáb MGT bude potěšen pokud bude moci pomáhat.

Veškerá výroba obsahuje ROM 1.0. Tento manuál detailně informuje o ROM 1.0 a ROM 1.2. V období duben-květen 1990 všichni majitelé Coupé budou mít k dispozici zvýšenou ROM 2.0, ROM 1.2 odkázana v tomto manuálu je předvýrobek verze 2.0.

Napsal

Bruce Gordon & Andy Wright

= dodatečnými informacemi od

Gary Thomase & Adriana Parkera

Doplňkové informace uživatelům zvukového čipu SAA 1099

Andy Graham

Překlad

Ing. Vladislav Viták
Ing. Vítězslav Vičan

Každá námaha má být vynaložena k jistotě, že obsah tohoto manuálu je opraven v každém detailu, je třeba si uvědomit, že MGT nemůže garantovat, že tato verze je definitivní. My máme jistotu dalšího pokračování.

SAM Coupé je moderní mikropočítač, používající mikroprocesor Z 80B s frekvencí 6 MHz. Má 256K RAM jako standart, se zásuvkou dovolující rozšíření na 512K.

Hlavní kontrolou přístroje je VLSI VGT-200 vstupní řady, obvykle k řízení mimo hlavní úkoly procesoru (video, listování a řízení všech paměti, plán video paměti, rozvržení barevné palety a všech vstupních a výstupních řídicích portů). Jsou zde 4 módy video operace hardwerové kapacity, které ukazují 16 barev z palety 128 na každé řádce obrazu.

Zvuk z přístroje je generovan zvukovým generátorem Philips SAA 1099 se stereo schopnosti, z 6 kanálů při 8 oktávách s 256 tóny v oktávě, dva bílé hlasové generatory, 6 amplitudových kontrolních a dva moderátory vlnění.

Všeobecný popis

Pohon	Mikroprocesor Z 80B, 6 MHz
Řízení	Obvykle VLSI 10,00 - brána ASIC čipu
ROM	32K * 8 ROM s přístupem 150 ns, obsahující SAB Basic, disk, BIOS
RAM	256K, max. až 512K (256K*4 100ns DRAM)
Zvuk	Slučovač Philips SAA 1099, 6 kanálů, 8 oktáv, stereo s kontrolou amplitudy a obsahu, plus volba tvaru vlny

Grafické znázornění - nabízí 4 módy

- mód 1 32 * 24 znakových článků na obrazovce, každý článek je schopen dvou barev, 16 barev vybraných ze 128 slučitelných se SPECTREM.
- mód 2 obdobný jako mód 1, ale se 32 * 192 články, každý článek schopen 2 barev, 16 barev vybraných ze 128
- mód 3 ukazuje 80 sloupců textu - 512 * 192 bodů, každý bod je schopen zvolit barvu, 4 barvy na řádku volí ze 128
- mód 4 256 * 192 bodů grafického obrazu, každý bod je schopen zvolit barvu, 16 barev na řádku volí ze 128

Ve všech módech mohou být barvy předefinovány v řádku přerušení sledující všech 128 barev ukázaných na obrazovce.

UHF

(TV kanál 36) pro kvalitní přenos barev, digitální a lineární RGB přes SCART, standartní ATARI joystick (schopnost ovládat dva joysticky přes speciální konektor)

Mys - standartní k Coupé

Světelné pero / světelná puška - standart k Coupé

Domácí kazetový magnetofon

MIDI IN, MIDI OUT (MIDI jdoucí přes přepínač)

NETWORK (síť) - stíněný mikrofonní kabel, 7mi kolíkový konektor
DIN

Výstup zásuvky AUDIO

RS 232 a paralelní tisk jdoucí přes vnější MGT interfejs spojený
v rozváděcím portu.

64 kolíkový standardní rozváděcí port pro další periferie

disk drajvy - 1 nebo 2 a uvnitř namontovaný CITIZEN 3,5"; 1 MB
nenaformátovaný; 780 kB po naformátování

Klávesnice má 72 kláves membránového typu, obsahuje 10 funkčních
kláves (nadefinovaných)

Zdroj - 4,75 - 5,25 V

Spotřeba - 11,2 W

Chvění - působící 5 - 500 Hz / 0,5 G; nepůsobící 5 - 500 Hz / 2 G

Prostředí okolní teplota - působící 5-45 °C; uskladnění 20-50 °C

Relativní vlhkost - mokrá žárovka 29,4 °C; bez kondenzace

Spolehlivost - 10 000 provozních hodin

Životnost komponentů - větší než 5 let

Hmotnost - 2,26 kg

Napětí - primární vstupy 240 V; 50 Hz
sekundárni vstupy 5 V; 2 A
12 V; 200 mA

Interfejsy

Všechna spojení ke Coupé, kromě odstranitelných drajvů a RAM
rozšiřující zásuvky jsou vyvedena přes zadní panel.

MIDI IN

Standartní 7-mi kolíkový konektor DIN. Tento sériový vstup pracující ve 31,25 K baudech je zásobován cestou opto-izolátoru k sériově/parallelnímu konventoru, který přeruší CPU, když data mají být shromážděny v MIDI-IN registru (253 dec). Tento INPUT může také být čten vedeně bitem 7 registru video paměti VMPR (252 dec). Tento konektor je také používán při NETWORK (sítí).

PIN	Signál
1	NET - LOOP
2	N.C.
3	NET + LOOP
4	MIDI + IN
5	MIDI - IN
6	NET - LOOP
7	NET + LOOP

MIDI OUT

Standartní 7-mi kolíkový konektor DIN. Má za úkol psaní údajů a dat k MIDI OUT registru (253 dec). MIDI výstup vede proud 7,5 mA ve 31,25 K baudech. Když odesílá i bit z PEN registru (TX FMST z registru 248 dec) je uložen. Tento OUTPUT může být také řízen bitem 7 z VMPR (252 dec). Vnitřní spojení z MIDI IN do MIDI OUT použitím bitu 6 THROM z MIDI BORDER registru (254 dec). Tento konektor je také užíván při NETWORK (počítačových sítích). Přerušení je dáno ukončením odeslání MIDI údajů.

PIN	Signál
1	NET - LOOP
2	GND
3	NET + LOOP
4	MIDI + OUT
5	MIDI - OUT
6	NET - LOOP
7	NET + LOOP

BREAK (NMI) tlačítko

Když stisknete toto tlačítko směřuje CPU na adresu 0066H, kde je stanovená adresa v paměti. Je to mnohostranná funkce pro programátory ke stanovení řízení, k oprávnění použití klávesy jako ESC, BREAK nebo CRASH.

- - -

Když je vložen DOS Spectrum Emulátor, toto tlačítko je používáno k aktivizaci snadného přenosu programu.

JOYSTICK

Standartní 9-ti kolíkový typu "D". Joystick má standartní ATARI spoje spolu s 5 V. MGT dovoluje pro dvojnásobné ovládání s druhým joystickem použití speciálního konektoru-kolíkové zásuvky. Joystick je čten z portu klávesnice (254 dec) a pokrývá číselné klávesy 1-5 pro druhý joystick a 6-0 pro první joystick.

PIN	Signal
1	UP - nahoru
2	DOWN - dolu
3	LEFT - doleva
4	RIGHT - doprava
5	0 VOLTS
6	FIRE - střelba
7	+ 5 V
8	STROBE 1
9	STROBE 2

MOUSE - myš

Standartní 8-mi kolíkový konektor DIN. MGT myš přeruší CPU na žádost o čtení řádků signálu X a Y. Software je celý v ROM a pokrývá řidící klávesy. Výstupy myši mohou být čteny z portu klávesnice (254 dec) s adresou řádků A8 až A15 (RDMSEL).

PIN	Signál
1	DOWN
2	UP
3	CTRL
4	LEFT
5	RIGHT
6	MSE INT
7	RDMSEL
8	+ 5 VOLTS
	SCREEN GROUND

RESET tlačítko

Když stisknete toto tlačítko přimějete CPU k vycištění paměti a opakování rádku v počátku paměti. (0000 hex)

To také obnovuje následující:

1. Kontroluje floppy disk
2. Registry stránky paměti
3. Registry přijetí a odeslání MIDI
4. Registr BORDER

Neobnovuje barevnou nabídku, registry zvukového čipu, nebo hodnoty LINE přerušovaného registru (249 dec).

Magnetofonova zásuvka

Standartní 3,5 mono Jack. Je obousměrný. Když nahráváme bude spojen se zásuvkou MIC na standartním kazetovém magnetofonu, když přehráváme bude spojen se zásuvkou EAR na magnetofonu. Použití není jiné u Coupé, jestliže EAR a MIC jsou spojeny spolu, ale jen některé magnetofny to dovolují.

Světelné pero & AUDIO port

Standartní 5-ti kolíkový konektor DIN. Konektor má dvojitou funkci:
- vstup světelného pera / pušky
- výstup stereo zvuku k řízení HIFI na výstup AUX

PIN	Signál
1	+ 5 voltů
2	AUDIO LEFT OUTPUT - levý výstup
3	0 voltů
4	SPEN INPUT
5	AUDIO RIGHT OUTPUT - pravý výstup

Napáťový vstup

Upravený DC vstup z 12 voltů, 100 mA a 5 V, 2 mA. Zcela připravený přístroj má max. spotřebu 15 W.

PIN	Signál
1	+ 5 voltů
2	0 voltů (signál GROUND)
3	0 voltů (DIGITAL GROUND)
4	COMPOSITE VIDEO
5	+ 12 voltů
6	SOUND OUTPUT MONO (zvukový výstup)

EUROKONEKTOR (Expanzni konektor)

Standartní 64 kolikový Eurokonektor zásuvka s řadou A a C. Seznam signálů dosažitelný z této zásuvky je ukázán níže. Znaky řady A jsou v dolní části konektoru a řada C je nahore. Všechny znaky končící na L jsou málo účinné.

Má všechny typické hardware signály pořízené z jiného počítače. Navíc obsahuje na konektoru mnoho zvláštních hardware signálů. To dovoluje hardware projekt dosud neznámé pružnosti a řidit mikropočítač z vnějšího zařízení.

PIN	Signál	PIN	Signál
1A	DBDIRL	1C	IORQL
2A	RDL	2C	MREQL
3A	WRL	3C	HALTL
4A	BUSAKL	4C	NMIL
5A	WAITL	5C	INTL
6A	BUSREQL	6C	CD1
7A	RESETL	7C	CDO
8A	CMIL	8C	CD7
9A	REFRESHL	9C	CD2
10A	0 voltů	10C	+ 5 voltů
11A	A0	11C	CD6
12A	A1	12C	CD5
13A	A2	13C	CD3
14A	A3	14C	CD4
15A	A4	15C	CPU CLK
16A	A5	16C	A15
17A	A6	17C	A14
18A	A7	18C	A13
19A	A8	19C	A12
20A	A9	20C	A11
21A	A10	21C	DISC 2
22A	MSEINTL	22C	ROMCSL
23A	XMEML	23C	EARMIC
24A	8 MHz	24C	DISC 1L
25A	RED 1	25C	PRINTL
26A	GREEN 1 (zelená)	26C	BLUE 1
27A	C SYNCL	27C	ROMCSRL
28A	SPEN	28C	AUDIO RIGHT OUTPUT (pravý výstup)
29A	BLUE 0 (modrá)	29C	AUDIO LEFT OUTPUT (levý výstup)
30A	RED 0 (červená)	30C	COMP VIDEO
31A	BRIGHT	31C	GREEN 0 (zelená)
32A	+ 5 voltů	32C	0 voltů

Když se podíváte na Eurokonektor v zadní části Coupé, jednotlivé piny jsou číslovány následovně:



Pro detaily mikroprocesoru Z 80B doporučujeme nějaký standardní text z 80. Upozorňujeme, že zveřejnime SAM Coupé MultiExpansion (SME) v blízké budoucnosti. (Tento výrobek má být zveřejněn tak jako CARD CAB v některé naší literatuře). Některý ze signálů zde popsánych musí být obsažen v nějakých návrzích k zajištění jejich opravné operace když je používán SME.

DBDIR (1A) Tento signál není používán počítačem, ale musí být používán všemi vnějšími zařízeními k umožnění jejich použití s SME. SME má vyrovnávací paměti k zabránění nevhodných vstupů do přístroje. Nicméně, to je odpovědnost hardwarového konstruktéra k zajištění, že data při vysílání a přijímání mají směr bitu určen v souladu s operací (čtení nebo psaní) splněn. Operace budou nasledující:

- 1) Procesor poslal prýč adresu
- 2) Vnější zařízení dekódovalo adresu
- 3) Vnější zařízení jelo DBDIR rádce z otevřeného sběrače výstupu, LOW pro procesor READ. To je jako WIRED-OR uvnitř počítače.
- 4) Obvykle je pokračováno se Z80B po přemístění údaje

RDL (2A)	Cte signál z procesoru
WRL (3A)	Píše signál z procesoru
BUSAKL (4A)	Bere na vědomí signál z procesoru
WAITL (5A)	Čeká signál procesoru
BUSREQL (6A)	Žádá signál k vnějšímu zařízení
RESETL (7A)	To je systém vynulování generování při 10 k ohnehod a 47 mikro F. To dává puls v trvání přibližně 330 ms. To naznačuje, že pokud ostrost nulování okrajové je požadováno, tento signál je schválen přes Schmittův převrzačec.
CM1L (8A)	Překódovaný signál přiveden z procesoru
REFRESHL (9A)	Obnovený signál z procesoru
O V (11A)	Adresa řádků A0 až A10 z procesoru jsou na vývodech 11A až 21A
MSEINTL (22A)	Mys přeruší signál. Signál je tažen přes vnější zařízení k informování procesoru, že mys má souřadnice změněny.
XMEML (23A)	Signál vnější paměti, který bude veden dolů, když procesor požaduje vypnutí vnitřní paměti a přístup vnejší paměti. Všimněte si, že jenom paměť nad adres-

sou 8000H v každém bloku 64k bude vypnuta. To je odpovědnost konstruktéra periférie k jistotě, že XMEM signál je obsažen v adrese dekódujici obvod konstrukce k vyvarování ze potíží mezi vnitřní RAM a vnějším zařízením.

8MHz (24A)	Signal 8 MHz.
RED 1 (25A)	Kazda barva je garantovana pres tri barvy rídící signály, intenzita kazdé barvy je určena třemi bity RED 1 je MSB (nejvýznamější bit) ze signálu červené barvy v běžné pozici obrazového prvku.
GREEN 1 (26A)	GREEN 1 je MSB ze signálu zelené barvy (viz bod 25 A)
CSYNC (27A)	COMPOSITE SYNC pro video okruh jako na konektor SCART
SPEN (28A)	Signál světelného pera,který je obvykle nízko.Když se světelné pero dotyka obrazu a rastrový hrot pero,určitý okraj je vrácen na tento signál ze zásuvky světelného pera. To je příčinou dvou registrů LPEN a HPEN k obsazení bežného koordinátu x a y samostatně ze světelného pera.
BLUEO (29A)	BLUEO je druhý bit ze signálu modré barvy (viz bod 25A pro uplný popis)
REDO (30A)	REDO je druhý bit ze signálu červené barvy(viz bod 25A pro úplný popis)
BRIGHT (31A)	BRIGHT je LSB (nejméně významny bit) ze všech barevných bitů. (viz bod 25A pro úplný popis)
+ 5 V (32A)	+5 V obstarává vedení, max 250 mA.
IORQL (1C)	Vstup/Výstup požaduje aktivitu, když adresa obahuje běžnou branu adresy z procesoru.
MREQL (2C)	Paměť požaduje aktivitu, když adresa obsahuje běžnou adresu z paměti procesoru.
HALT (3C)	Zastavení naznačuje procesoru, že má provést HALT (zastavení) instrukci a čeká na přerušení.
NMIL (4C)	(NON MASKABLE INTERRUPT) Nedosahuje přerušení důvod procesoru k provedení skoku,z kterého je snadno naveden k uživateli. Je použit jako BREAK tlačítka pro DOS. Je tažen nahoru přes vnitřní registr 10 kilohmů a proto pojede níže přes otevřený sběrový obvod.
INTL (5C)	Dovoluje přerušení, je používán k přerušení procesoru v jeho běžné úloze, jestliže přerušení jsou obvykle umožněny.
OD1 (6C)	Data řádků D1,D0,D7 a D2 jsou na jehličkách 6C až 9C.
+ 5V (10C)	+5 V zásobuje vedení
OD6 (11C)	Data řádků D6,D5,D3 a D4 jsou na jehličkách 11C až 14C
CPUCLK (15C)	Signál 6 MHz z procesoru

A15 (16C)	Adresa rádků A 15 až A11 jsou na jehličkách 16C až 20C
DISK2 (21C)	Disk 2 je dekódován ze základní adresy osmi portů drafy 2
ROMCSL (22C)	ROM čip volba je tažená hlavně k oslabení vnitřní ROM (použit ve spojení s ROMCSR na jehličce 27C). To nepřímo ukazuje, když ROM je přístupná.
EARMIC (23C)	EARMIC je spojen s logickou úrovní strany obvodu kazety.
DISK1L (24C)	DISK1 je dekódován ze základní adresy osmi portů drafy 1
PRINTL (25C)	PRINT je dekódován z dalších dvou možných tiskových adres (246 a 248)
BLUE1 (26C)	BLUE1 je MSB ze signálu modré barvy (viz bod 24 A pro úplný popis)
ROMCSRL (27C)	ROMCSR je spojená s ROMCS (22C) přes resistor 1kohm. Jestliže, ROMCS je tažena hlavně přes vnější obvod pak ROMCSR bude použita jako výběrový čip pro vnější ROM, která pak bude umístěna ve vnitřní ROM.
AUDIORIGHT (28C)	Pravy audiosignál, jako na zásuvce světelného pěra je přinesen mimo. (Musí být zesílen před použitím)
AUDIOLEFT (29C)	Levy audiosignál je přítomen na této jehličce. Některá opatření a rovina signálu jsou aplikovány jako u AUDIORIGHT.
CVID (30C)	Složené video, odesláno do modulátoru a zásuvka SCART je umístěna na tuto jehličku.
GREENO (31C)	GREENO je druhý bit ze signálu zelené barvy (viz bod @% A pro úplný popis)
O V (32C)	O voltů - zásobuje vedení

SCART zásuvka

SCART zásuvka se používá k ovládání všech video a zvukových výstupů. Ne všechny spoje jsou standartní.

PIN	Signál	PIN	Signál
1	AUDIO OUT RH	2	SPEN
3	AUDIO OUT LH	4	AUDIO EARTH
5	BLUE EARTH	6	BLUE TTL OUT
7	BLUE LIN.OUT	8	RED TTL OUT
9	GREEN EARTH	10	GREEN TTL OUT
11	GREEN LIN.OUT	12	+5V POWER IN
13	RED EARTH	14	CSYNC EARTH
15	RED LIN.OUT	16	CSYNC
17	C.VIDEO EARTH	18	+12 V POWER IN
19	C.VID OUT	20	BRIGHT TTL OUT
21	GND.		

UHF výstup

UHF televizní standartní video a mono zvuk mají výstupy na kanálu 36; UHF výstup je ve zdrojové skřínce.

Úzká šterbina v disk drivu

Disk Drayvy používané v SAM Coupé jsou typu CITIZEN 3,5", 1MB (formátován na 780 kB)

Coupé může řídit nejvíce dva disk drayvy. Spojení se dvěma drayvy jsou vyrobeny jako dva 32 kolikové Eurokonektory s řádky A a B.

PIN	Signál	PIN	Signál
1A	0 Voltů	1B	WR
2A	0 voltů	2B	A0
3A	0 voltů	3B	A1
4A	0 voltů	4B	D0
5A	0 voltů	5B	D1
6A	0 voltů	6B	D2
7A	0 voltů	7B	D3
8A	0 voltů	8B	D4
9A	0 voltů	9B	D5
10A	0 voltů	10B	D6
11A	5 voltů	11B	D7
12A	5 voltů	12B	8 MHz
13A	5 voltů	13B	RST
14A	5 voltů	14B	No connection
15A	5 voltů	15B	A2
16A	5 voltů	16B	DISK 1 nebo DISK 2

DISK DRIVE

SAM Coupé může být spojen s jedním nebo dvěma diskami. Tyto budou normálně uvnitř SAM Coupé. Ale máme uděláno opatření pro zájemce k použití vnějšího disku (Shugart 400 typu 5,25" nebo 3,5") pokud připoji SAM External Drive interface (SDI). Oba vnitřní disky a SDI používá řídící čip VL-1772-02 vyroben VLSI.
NB A11 všechny hodnoty jsou dány v desítkách.

SAM Coupé řídí každý disk druhý přes 8 I/O plánovaných portů jako podsestavu pro oba disky.

 Disk 1 zahrnuje šířku 224 až 231
 Disk 2 zahrnuje šířku 240 až 247

V každém případě adresa může být zadána takto:

 Disk 1 base (základ) (224) + odstup (0 až 7)
 Disk 2 base (základ) (240) + odstup (0 až 7)

Oba údaje z odstupů splní určité funkce na příslušných stranách disket. První čtyři odstupy (0-3) se vztahují na první stranu diskety, další čtyři odstupy (4-7) se vztahují na druhou stranu diskety. Odstupy (4-7) mohou být ignorovány pro jednoduché stránkování disku.

Odstup určuje, který registr z 1772 bude přístupný, jak je ukázáno níže:

Odstup	Čtení	Píšení
0	Postavení (disketa strana 1)	Povel (disketa strana 1)
1	Stopa 1	Stopa 1
2	Sektor 1	Sektor 1
3	Data 1	Data 1
4	Postavení (disketa strana 2)	Povel (disketa strana 2)
5	Stopa 2	Stopa 2
6	Sektor 2	Sektor 2
7	Data 2	Data 2

Například ke čtení běžného obsahu dat registru na straně 2 diskety 1:

 IN A, (231)

kde 231 = disk 1 základ (224) + odstupy (7)

Pro úplnou informaci ke kontrole čipu 1772 se vztahuji data uvedená v dodatku B.

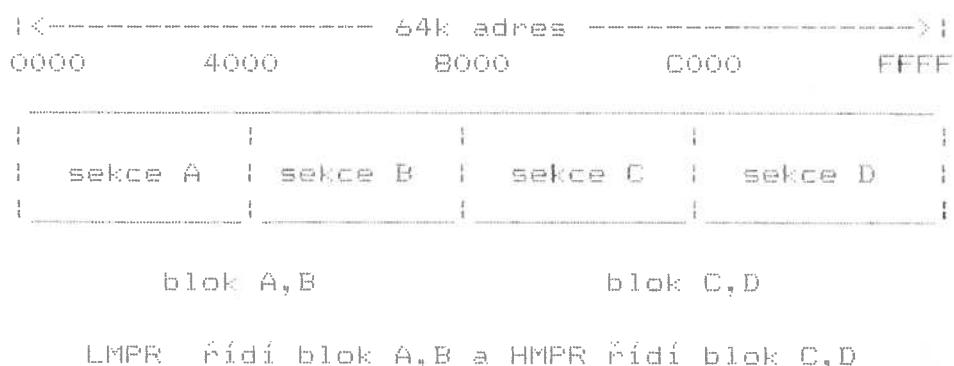
Paměť

Coupé může mít kapacitu 512kB RAM (s vnější pamětí umožňuje připojit přes rozšiřovací konektor použitím signálu XMEML). Strojový BASIC je vybaven 256 kB s vnitřním rozšiřovacím konektorem pro dalších 256 kB. Použitá paměť je Dynamic Random Access (DRAM) 256*4 bity - 100 ns, 20-ti jehličkový dvojitý plastikový komplex.

Protože Z 80B CPU je limitován 64k adresovaného rozsahu, jedna z funkcí ASIC (používání určitých celých obvodů) v Coupé je řízení adresování této paměti rozdělující to na 32 stran po 16k.

ASIC řídí stránkování přes 8-mi bitový čtecí / psací registry LMPR (nižší pamětová stránka registru) I/O adresa 250 dec HMPR (vyšší pamětová stránka registru) I/O adresa 251 dec Nížších 5 bitů každého registru představuje stránku 0 až 31 (ovšem na basic přistroji s 256k paměti stránkuje od 0 do 15).

Na ilustraci systému stránkování na Coupé je nejlepší ukázat 64k adresného obvodu Z 80 jako 2 bloky dvou sekci 16-ti kilových reprezentujících písmena A, B a C, D:



LMPR řídí blok A,B a HMPR řídí blok C,D

Jestliže písem 00 hex na LMPR, tak stránka 0 paměti je přidělena sekci A adresného obvodu CPU. Sekce B je vždy automaticky přidělena jedné stránce nad sekcí A, v tomto případě stránka 1.

Jestliže písem 02 hex na HMPR, tak stránka 2 paměti je přidělena sekci C adresného obvodu CPU. Sekce D je vždy automaticky přidělena jedné stránce nad sekcí C, v tomto případě stránka 3.

V našem případě mapa adres CPU bude vypadat takto:



Obrazové módy Coupé

Jsou zde 4 obrazové módy používané v Coupé, každé použití mění atributy paměti a každý mód má rozdílné atributy pro použití programátorem.

Hardwérový ukazatelé, kteří ukazují obsahy paměti jsou řízeny VMPR (video registr stránky paměti - 252 dec).

Použitím nížších 5-ti bitů tohoto registru můžeme zpřístupnit až 32 stran obrazové paměti. Je nutné si povšimnout, že módy 3 a 4 používají 24kB, které jdou přes hranice stránky 16kB. Kde se to stalo, video adresující hardware sbalí do další stránky mimo tentýž blok.

Například: Při vstupu strany 12 do VMPR, video sbalí na stranu 13. To je jednoduchý příklad. Totéž bude použito u kterékoliv stránky.

Mód 1

32 buněk*24 řádků ve dvou barvách ze 16 z palety 128 barev, dostanete 768 znakových buněk (8*8) používá 6kB ze zmapované paměti a 0,75 kB ze znakové paměti. Tento mód napodobuje paměť Spectra.

Mód 2

32 buněk*192 řádků ve dvou barvách ze 16 z palety 128 barev, dostanete 6144 znakových buněk (8*1) používá 6kB ze zmapované paměti a 6kB ze znakové paměti. Tento mód má styčnou paměť adresovanou ve dvou blocích.

Mód 3

512 bodů * 192 řádků ve 4 barvách z palety 128 barev, dostanete 98304 bodů, užívá 24kB paměti. Pokud v tomto módu použijete znaky 6 bodů široké dostaneme 85 znaků na řádek.

Mód 4

256 bodů*192 řádků v 16-ti barvách z palety 128 barev, dostanete 49252 bodů, užívá 24kB paměti. Tento mód je ideální pro grafické znázornění a když ho použijete v kombinaci s LINE INT registrém můžete znázornit všech 128 barev na obrazovce.

Určení běžné obrazovky

SAM Coupe poskytuje port volající VMPR (adresa 252). K určení základní adresy běžné obrazovky použijte následující program.

```
10 LET A = IN 252 BAND 31      REM Vlož běžnou stránku
20 LET BASE = (A+1)*16384      REM BASE nyní vymovná počátek
                                REM rozlohy obrazu
```

Klávesnice

Coupe má celkem 72 kláves na klávesnici. Je použitá jako forma 9*8, užívající dva porty (brány) KEYBOARD (klávesnice) (254 dec) a STATUS (postavení) (249 dec).

8 vstupních řad je provedeno jako vstupní KEYBOARD port nížších 5 bitů je reprezentováno K1, K2, K3, K4 a K5 zatím co vstupní port STATUS vyšším 3 byty je reprezentován K6, K7 a K8. Dvět vstupních řádků je provedeno jako CPU řádky adres AD8, AD9, AD10, AD11, AD12, AD13, AD14, AD15 a ASIC řádek RDMSEL.

Typický příklad prohlížení klávesnice:

```
LD      HL,SCAN      ;soubor HL s buferem statutem
LD      B,11111110 bin ;soubor horních řad adresy
LOOP LD  C,HIKEY     ;port adresy K6-K8
IN      A,(C)
AND    11100000 bin  ;pruh nepotřebných bitů
LD      (HL),A        ;podřídit to
LD      C,LOKEY      ;port adresy K1-K5
IN      A,C
AND    00011111 bin  ;pruh nepotřebných bitů
OR     (HL)
LD      (HL),A        ;zachovat to
INC    HL
SCF
RLC    B
JR     C,LOOP
IN      A,(C)
AND    00011111      ;pruh nepotřebných bitů
LD      (HL),A        ;zachovat to
RET
```

Vstupní / výstupní porty

SAM Coupe může adresovat 64 kB z portů. Adresy 224 (EOH) až 255 (FFH) jsou přiděleny Coupe. Odkazují na diagram, připravený pro přehled těchto portů a jejich obsahu.

Psaci (výstupní) porty

Zvukové porty (adresa portu 511 dec) a (data portu 255 dec). Zvukový čip Philips SAA 1099 je řízen přes tyto dva porty adres. Zvukový čip je řízen v BASIC použitím povelu SOUND a,d. Viz připojený návod použití Philips a doplněk SAA 1099, který je obsažen nakonec pro úplnou informaci.

VMPR - Video stránka paměti registr (252dec)

Tento čteči / psaci registr hlavně řídí stranu adresovanou paměti zobrazené na obrazovce.

bit 0 R/W	BCD 1	řízení stránky video
bit 1 R/W	BCD 2	řízení stránky video
bit 2 R/W	BCD 4	řízení stránky video
bit 3 R/W	BCD 8	řízení stránky video
bit 4 R/W	BCD 16	řízení banky video, používá se k přepnutí mezi bankami 256 kB
bit 5 R/W	MDE 0	první bit řízení módu obrazovky
bit 6 R/W	MDE 1	druhy bit řízení módu obrazovky
bit 7 -/W	TXMIDI	výstupy bitu k písmenu řízení kanálu MIDI OUT
bit 7 R/-	RXMIDI	vstup bitu z kanálu MIDI IN

HMPR - High memory page registr (251 dec)

Tento čteči/psací registr je používán hlavně k řízení stránkování vyšší paměti adresovaného rozsahu v CPU.

bit 0 R/W	BCD 1	řízení vyšší stránky paměti
bit 1 R/W	BCD 2	řízení vyšší stránky paměti
bit 2 R/W	BCD 4	řízení vyšší stránky paměti
bit 3 R/W	BCD 8	řízení vyšší stránky paměti
bit 4 R/W	BCD 16	řízení vyšší banky paměti
bit 5 R/W	MD3S0	BCD 4 vyhledá adresu barvy dosažitelné pouze v módu 3
bit 6 R/W	MD3S1	BCD 8 vyhledá adresu barvy dosažitelné pouze v módu 3
bit 7 R/W	MCNTRL	Pokud je tento bit použit, když CPU adresuje vyšší paměť, pak vnější signál XMEM jede niže a Coupé ukazuje na rozváděcí konektor paměti sekci C a D. (adresy 32768 až 65536)

LMPR = regist stránky nižší paměti (250 dec)

Tento čtecí/písací registr je používán hlavně k řízení stánkování paměti adresovaného rozsahu CPU.

bit 0 R/W	BCD 1	řízení nižší stránky paměti
bit 1 R/W	BCD 2	řízení nižší stránky paměti
bit 2 R/W	BCD 4	řízení nižší stránky paměti
bit 3 R/W	BCD 8	řízení nižší stránky paměti
bit 4 R/W	BCD 16	řízení nižší banky paměti
bit 5 R/W	ROM 0	když bit umístí výše, RAM nahradí první polovinu ROM v sekci A mapy adresy CPU
bit 6 R/W	ROM 1	když bit umístí výše, druhá polovina ROM nahradí RAM v sekci D mapy adresy CPU
bit 7 R/W	WPRAM	napsání ochrany RAM v sekci A mapy adresy CPU je umožněno, když tento bit je umístěn výše

MIDI OUT port (253 dec)

Po napsání údaje tomuto portu hardware automaticky odesílá skrze kanál MIDI OUT v 31.25 k baudu, standartní pro MIDI protokol. Bit 4 registru STATUS (249 dec) je umístěn výše, pokud je tento registr plný.

BORDER port (254 dec)

Tento vystupní port hlavně řídí barvu okoli (border) obrazovky. Je opatřený 4 bitovou adresou ke CLUT (color look up Table) k oprávnění barvy a k ukázání během času okoli.

bit 0	BCD 1	adresa CLUT pro barvu okoli
bit 1	BCD 2	adresa CLUT pro barvu okoli
bit 2	BCD 4	adresa CLUT pro barvu okoli
bit 3	MIC	výstup řídícího bitu, normálně umístěn vysoko
bit 4	BEEP	výstup řídícího bitu, normálně umístěn nízko
bit 5	BCD 8	adresa CLUT pro barvu okoli
bit 6	THROM	bit umístěn vysoko, přístupný přes operaci MIDI
bit 7	SOFF	bit umístěn vysoko, k oslabení obrazovky, účinný pouze v obrazových módech 3 a 4, také odkládá paměť během sporu do konce období

Registr přerušení řádku (249 dec)

Tento pouze psací registr způsobuje přerušení CPU na konci zkoumaného řádku hodícího se obsahu (na začátku pravého okolí). Čísla řádků jsou 0 až 191. Tato funkce je vždy schopná, tedy překazi působení chybneho čísla řádku od 192 do 255, které může být vloženo. Užitečná funkce tohoto registru může být použita k přepnutí video módu nebo vyměně barevných hodnot.

CLUT - Colour Look Up Table (základní port 248 dec s AD 8-AD 11)

Zde je pojednání o 16 pouze 7-mi bitových registrech v CLUT.

barva	0	registrová adresa	0	na portu	248
	1		1		504
	2		2		760
	3		3		1016
	4		4		1272
	5		5		1528
	6		6		1784
	7		7		2040
	8		8		2296
	9		9		2552
	10		A		2808
	11		B		3064
	12		C		3320
	13		D		3576
	14		E		3832
	15		F		4088

Každý registr má 7 bitů ke znázornění jedné ze 128 možných barev

bit 0	BLU0	nejmenší významný bit modré
bit 1	REDO	nejmenší významný bit červené
bit 2	GRNO	nejmenší významný bit zelené
bit 3	BRIGHT	polovina síly bitu ve všech barvách
bit 4	BLU1	největší významný bit modré
bit 5	RED1	největší významný bit červené
bit 6	GRN1	největší významný bit zelené

Registry při zapnutí budou naloženy s chybou z ROM, umístění bude odpovídat hodnotám.

registrová adresa	0	černá	(0)	registrová adresa	8	černá	(0)
	1	modrá	(16)		9	světle modrá	(17)
	2	červená	(32)		A	jasmné červená	(34)
	3	oranžová	(48)		B	jasmné oranžová	(51)

registr 4	zelená	(64)	registr C	jasně zelená	(68)
5	šedá	(18)	D	jasně šedá	(85)
6	žlutá	(96)	E	jasně žlutá	(102)
7	bílá	(120)	F	výrazně bílá	(127)

Čísla barvy jsou ukázány: FOR c=0 TO 15

```
PRINT PEEK ( &55D8 + c )
NEXT c
```

Typický postup pro naložení CLUT se seznamu paměti:

```
LD      HL, TABLE      ; základ HL z vrcholu seznamu
LD      B, 16            ; základ B z objemu seznamu
LD      C, 248           ; základ C s adresou portu
OTDR
RET                ; vykonat základ B
                    ; výstup portu (C) a
                    ; návrat k povolání pravidelného postupu
```

Vysvětlení CLUT v módu 4 & módu 3

Na obrazovce v modu 4 jsou 4 bity užity k adresování CLUT pro barvu bodu. Kromě bajtu, nejdůležitější odkazuje na první bod a nejméně důležitý odkazuje na druhý bod.

Situace je obdobná v módu 3, ačkoliv v této době zde jsou pouze dva bity za bod, který adresuje CLUT. Mimo 8-mi bitového údaje bajtu, první dva nejdůležitější bity jsou použity jako adresa pro první bod.

Přirozeně pouze 4 ze 16 možných registrů budou dostupné. To lze překonat použitím mimořádných dvou bitů z HMPR (251 dec). Bit 5 z HMPR je užívá ke zpřístupnění BCD 4 vyhledání adresy barvy a bit 6 z HMPR je užíván ke zpřístupnění BCD 8 vyhledáním adresy barvy.

Tento cestou můžeme ještě vyhledat 16 barev uvedených v CLUT avšak vyšší rozhodnutí má grafické znázornění.

Čtecí (vstupní) porty

Registr atributů (255 dec)

Tento registr opravňuje programátora ke čtení atributů z běžně ukázaného znaku článku v módu 1 a 2, a třetí bajt ve čtyřech displeích v módech 3 a 4.

Registr klávesnice (254 dec)

Tento pouze čtecí registr se hlavně používá jako vstupních 5 bitů - nižších - z klávesnice. Je to také vstup pro myš, když adresy řádků AD 8 - AD 15 jsou vyšší.

bit 0	K1	klávesnice řádek 1	ovládání myši
bit 1	K2		2 myš nahoru
bit 2	K3		3 myš dolů - knoflík 2
bit 3	K4		4 myš doléva - knoflík 1
bit 4	K5		5 myš doprava - knoflík 3
bit 5	SPEN	signál STROBE pro světelné pero/sériový vstupní bit	
bit 6	EAR	sériový vstup z EAR kazetového magnetofonu	
bit 7	SOFF	postavení bitu je ukázáno, jestliže je umístěna vnější paměť.	

Registr MIDI IN (253 dec)

Tento pouze čtecí registr provádí přerušení CPU, když údaje bajtu má číst ze sériového vstupu MIDI. Údaje musí být načteny do CPU před dalším přerušením (obvykle 320 msec).

Registr postavení (STATUS) (249 dec)

Tento pouze čtecí registr se používá hlavně pro načtení přerušení postavení. Ačkoliv všech 5 přerušení jde přes působící CPU pod modem i se přeruší. Zde není známá cesta, která je požadovaná, proto načtení tohoto registru je nezbytné. Přibližný čas přerušení je asi 20 msec.

bit 0	LINE int ...	když vydává zvuk, signalizuje rádku přerušení požadovaného registru
bit 1	MOUSE int ..	když vydává zvuk, signalizuje myši, že požaduje přerušení
bit 2	MIDI IN int.	když vydává zvuk, signalizuje, že MIDI panel má údaj bajtu
bit 3	FRAME int ..	když vydává zvuk, signalizuje, že zkoumaná stavba má být úplná (50 za sec)
bit 4	MIDI OUT int	když vydává zvuk, oznamuje, že registr MIDI OUT má právě úplné údaje výstupu
bit 5	K6 int	klávesnice řádek 6
bit 6	K7 int	klávesnice řádek 7
bit 7	K8 int	klávesnice řádek 8

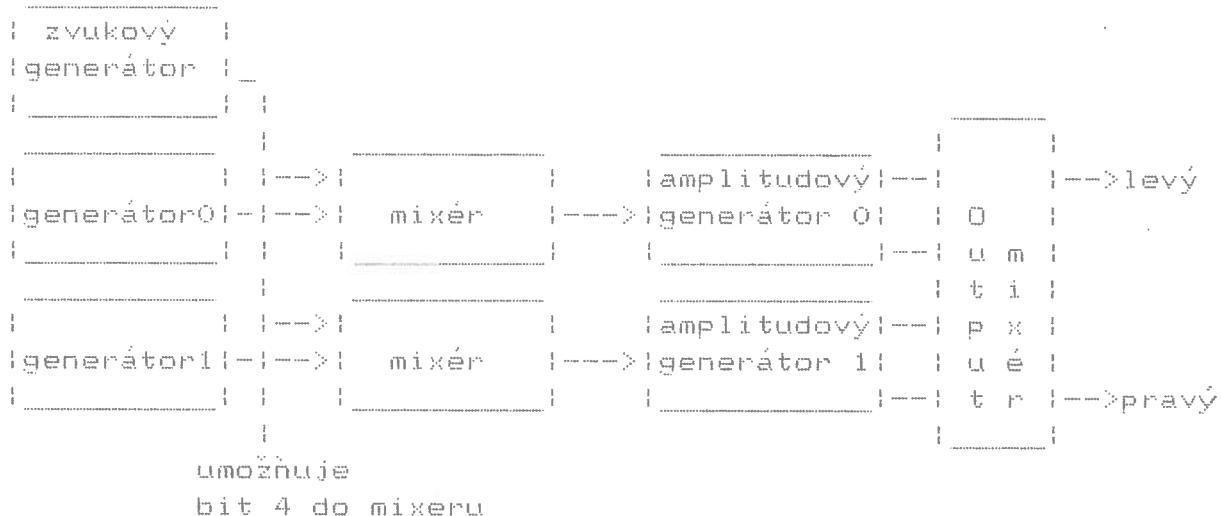
Registry pero (LPEN 248 dec) (HPEN 504 dec)

Jestliže světelné pero není zapojeno, tyto pouze čtecí registry jsou nepřetržitě nahoře datovány s běžnou polohou zkoumaného. Registr LPEN je napojen na vodorovný, nebo-li koordinát X funkce zkoumaného a HPEN je napojen na svislý, nebo-li koordinát Y funkce zkoumaného. První dva bity registru LPEN jsou:

bit 0 - vstupy BCD 1 běžně CLUT adresy
bit 1 - TXFMST je postavení bitu pro MIDI OUT. Pokud je vyšší, ukazuje, že bajt je odesilán.

Zvukový čip PHILIPS SAA 1099

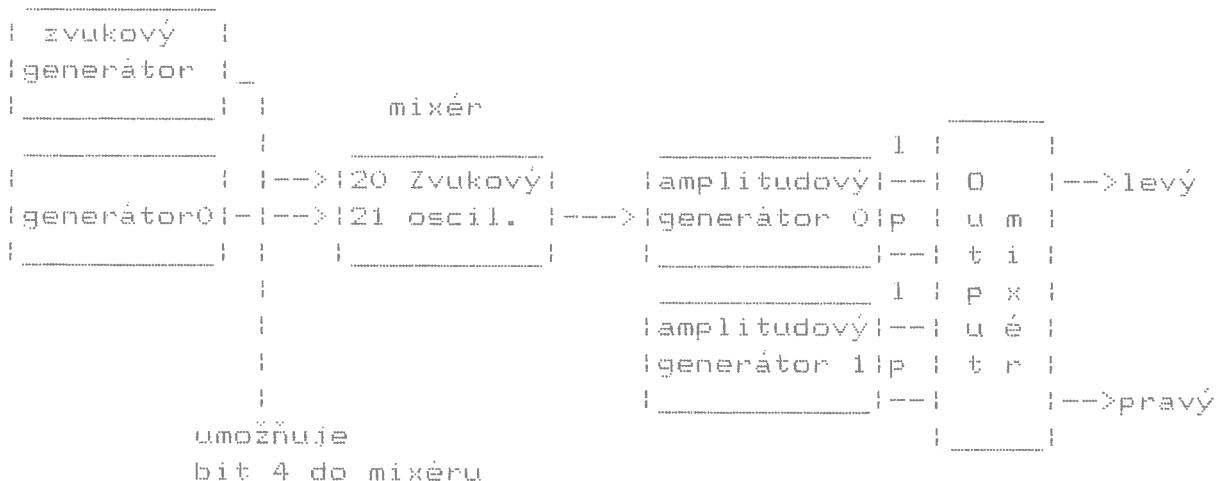
Zvukový čip má 6 registrů tónu, nebo oscilátorů a dva hlasové generátory. Ty všechny mohou být vyplňeny zleva doprava přes stereo pole použitím jedné z 256 možných poloh (16 různých hlasitostí na panelu, kdy 16×16 kombinací tvoří 256 poloh). Poloha ve stereo poli, vztahující se k tomuto jednotlivému oscilátoru, je řízena amplitudovým registrarem. Vše zahrnující zvuk může být zesílen, nebo zeslaben použitím výkonného směsovače. Pro slyšitelný zvuk bude registr "1", k utlumení zvuku bude registr "0".



Šest oscilátorů je rozděleno, tři na hlasové generátory. To umožnuje dva obalené registry použít samostatně.

THE USE OF SYMBOLS

Směšovač registrů 20-21 směšuje hlas každého oscilátoru. Registr 20 řídí, který z obou zvukových oscilátorů je vynechán nebo potlačen, zatím co registr 21 řídí hlas.



Užitím tohoto směšovače dovoluje kterýkoliv hlas nebo tón, nebo oboji, nebo nic, vyněchat jednoduchý registr amplitudy a také polohu ve stereo poli.

Intenzita

Intenzita každého úplného oscilátoru je řízena 2 parametry. Je možno 8 oktáv na generátoru a 256 tónů v oktávě. Přes pečlivou volbu výběru oktavy a tónového registru, hladký přechod je možný z nejnižší frekvence zvuku-31 Hz k nejvyšší frekvenci 7,81 kHz.

Hudba

Chromaticke stupnice jsou uvedeny v tabulce. Poznámka: Čísla tónů jsou platná také pro jiné oktavy téže noty, ačkoliv naladění může být rozdílné vůči nízším oktavám.

Chromaticka stupnice

Všechna čísla jsou desítkova

Nota	č.tónu (desít.)	č.oktavy	požadov. frekvence (Hz)	skuteč. frekvence (Hz)
stř.C	33	03	261,626	261,506
C#	60	03	277,183	277,162
C	85	03	293,665	293,427
D#	109	03	311,127	310,945
E	132	03	329,628	329,815
F	153	03	349,228	349,162
F#	173	03	369,994	369,822
G	192	03	391,995	391,850
G#	210	03	415,305	415,282
A	227	03	440,000	440,141
A#	243	03	466,164	466,418
B	5	04	493,883	494,071
C	33	04	523,251	523,013

Registry oktáv

Registr 16 je registr oktavy měničí oktávu tónových registrů 8 a 9. To umožňuje použít pouze tři bitů registru tónu k výběru požadované oktavy.

Tedy $2 \times 2 \times 2$ dostaneme našich 8 možných oktáv, jak byla zmínka v předešlém. Tak, k přístupu registru oktavy pro registr tónu 8, musíme dodat číslo (0-7) k nízším třem bitům registru 16. Pokud

jsme potřebovali k přístupu oktávy pro registr tónu 9, budeme pak potřebovat dodat číslo mezi 16 a 64 zajistující, že první 4 bity D0-D3 obsahují nulu, nebo číslo oktávy pro registr tónu 8, jestliže je použít.

Například: Požadovaná oktaava 5, je pro registr tónu 8

D7	D6	D5	D4	D3	D2	D1	D0
X 0 0 0 0 X 1 0 1							
_____ _____ _____ _____ _____ _____ _____ _____							

Číslo výstupu registru 16 = 5

Například: Požadovaná oktaava 6, je pro registr tónu 9

D7	D6	D5	D4	D3	D2	D1	D0
X 1 1 1 0 X 0 0 0							
_____ _____ _____ _____ _____ _____ _____ _____							

Číslo výstupu registru 16 = 96

Registers amplitud

Některý údaj odesílan k jednotlivému registru amplitudy může přesně řídit polohu zvuku ve stereo poli.

To umožňuje použít tutéž techniku jako u registru oktávy. Jediný rozdíl je, že čas čtyř bitů je použit k rozvodu hlasitosti každého kanálu a dva "polohy" binárního čísla vztahující se ke dvěma kanálům (levý a pravý) místo dvou oddělených registrů tónů.

Hlasitost 11 požadovaná
z levého kanálu

D7	D6	D5	D4	D3	D2	D1	D0
1 1 1 1 1 1 1 1							

Hlasitost 7 požadovaná
z pravého kanálu

0 1 1 1 1 1 0 1							
_____ _____ _____ _____ _____ _____ _____ _____							

pravý

levý

Registr amplitudy 2

Toto umístí zvuk směrem k levé straně stereo pole.

SAM Coupe software

Paměť

U Coupé je možné různé uspořádání paměti, s různými výhodami a nevýhodami a každá z těchto částí je pojednána v detailu. Obrazový mod a umístění obrazové paměti je snad první faktor k uvážení

Paměťová mapa SAM Coupé

FFFF H

ROM 1 nebo program, proměnné nebo obraz

OC000 H

program, proměnné nebo obraz

OB000 H

počátek kanálů

CHANS

05CB6 H

systémové proměnné

05A00 H

SVARS

přehled klávesnice

KTAB

Definice klávesnice

DKBUD

05800 H

Prerušování řádek
přehled barev

05600 H

LINICOLS

přehled palety

055D8 H	FALTAB
UDG	UDGS
vzory	
CHR\$ (127 - 168)	
05490 H	
vzory písmen	
CHR\$ (32 - 127)	
05190 H	CHARS
05120 H	
vyměření stránky	
33 bajtů	
(1 stránka po 16 k)	
05100 H	ALLCOT
050FF H	
všeobecný symbol	
bufer prostor	
04F00 H	
strojová hranice	SP
pohyblivá tečka	STKEND
kalkulačor	
04D00 H	STKBOT
04C00 H FARLDIR bufer	
04B50 H	HDL
hlavičky pásku	
04B00 H	
BASIC pro procedury	HDR
GOSUB/DO	STKBOT
štěření pro hromadění	BASSTK
nebo BASIC	BSTKEND
systém hromadění	
2,7k dostupných	HEAPEND
nestránkuje	
04000 H	HEAPST
ROM '0	
00000 H	

Umístění paměti obrazu

Uplný obraz zabírá 6,75k v módu 1 a 12k v módu 2 (nezahrnuje 2k mezery mezi jehličkou a údajem atributu). V těchto módech paměť obsahující obraz může být zapnuta v hranici 16k. Stránkovací systém znamená, že dalších 16k RAM stránky v téže polovině paměti bude stránka v tomtéž čase jako stránka obsahující obraz. Módy 3 a 4 požadují 24k a tedy dvě stránky 16k. Můžete kterýkoliv zanechat v obrazové paměti trvale stránkováné nebo právě tu stránku, ve které chcete čist nebo psát, poskytující tvůj obraz, ovládající kód je stránkován v tomtéž čase. ROM zapíná v obrazu, jak je požadováno, v částech C a D mapy Paměti (8000H).

Hranice

ROM obvykle udržuje strojovou hranici a systémové proměnné v sekci B v rozmezí 4000H - 5CDOH, takže nejsou stránkovány mimo obrazovku. Basicový program nebo rozsah proměnných je stránkován v horní polovině paměti.

ROM

ROM 0 obvykle zapnuta v sekci A paměťové mapy. ROM 1 je obvykle vypnuta, dokud není potřeba; když je použita je umístěna v sekci D. Přehled skoku rutin v ROM 0, v případě potřeby v ROM 1 bude stránkovat automaticky a pak obnoví předchozí postavení.

Použití subrutin ROM

ROM obvykle očekává hranici v sekci B a mnoho rutin žádá neporušené systémové proměnné v tomtéž rozsahu. Nic méně pokud si přejete použít tento pro Vás vlastní kód, můžete stránkovat mimo systémové proměnné a použít ROM rutiny nepřímo přes JSVIN (viz později). Tyto rutiny zapnou systémové proměnné zpět, dočasně v ROM rutině a Vaše volba je provedena. Obrazovka bude dočasně zapnuta v sekci C a D pokud si přejete, ale Váš kód může být obvykle umístěn v téže ploše. Přirozeně, ROM 0 musí umožňovat, pokud si přejete, použít přehled skoků, ale budete moci stránkovat.

Například, budete moci vložit Váš strojový kód bez přerušení vypnutí ROM 0, zapnuta nová RAM do sekce A/B a použita pro Váš program. Budete moci připravit tentýž kód k odhalení a neodhalení přerušení, pokud si přejete.

Z vnější strany sekce A umožňuje ROM 0 před použitím ROM rutiny a užitím JSVIN k vyvolání rutiny, kterou chcete. Pokud vypnete stránku Váš kód je běžně převeden, pamatujte, že další instrukce přijde z nové stránky.

Použití obrazových módů 3 a 4

Uspořádání obrazové paměti v módech 3 a 4 je velmi jednoduché. Metoda použití ROM v módu 4 (256*192, 16 barev) je ukázána níže. Na vstupu L a H jsou koordináty X a Y (0,0 nejvýše vlevo) a D má barvu k použití v obou pokusech. (např. d=77H pro barvu 7). H a L jsou vráceny a vyměněny. (vhodné např. během kreslení řádku)

PLOTLH: SCF
RR H
RR L
LD A, (HL)
JR C,PLOTOODD ;je pokud je bod lichý

XOR D
AND OFH
XOR D ;směšuje levý z D s pravým
;cte z obrazovky
LD (HL),A
ADD HL,HL ;obnovuje vstup HL
RET

PLOTOODD: XOR D
AND OFOH
XOR D
LD (HL),A
ADD HL,HL
INC L ;obnovuje vstup HL
RET

Adresa další řádky může být vždy získána přidáním 128 k adrese běžného obrazu.

Obraz může být lehce posouván doleva nebo doprava po 1 bodu použitím instrukce RLD a RRD provedenými cestou z 80.LDIR a LDDR mohou být použity po fázích 2,4,6, atd. bodech. Subrutiny ROM podporují největší možnosti.

Zapnutí palety

ROM preruší rutinu použitím horních registrů palety (CLUT) na počátku každé sestavy, užitím jednoho ze dvou přehledů 16 bajtů v PALTAB a přejímajícími mezi nimi, abychom tak dostali žádané barvy. To je dobrý nápad k udržení těchto přehledů up-to-date s běžnými barvami, pokud nechcete použít rutiny ROM, protože registry palety nemohou být čteny. Softwarem vloženým přes NMI bude nalezen správný PALTAB, užitečný, když vytvářený obraz složí nebo změní paletu.

ROM může také vyměnit určitou paměť palety, abychom dostali hodnotu určité rozložené řádky, umožňující mnohem více než 16 barev k ukázání. Jestliže je udělána jen jednoduchá změna řádku, to se stane během návratu a neblikání na obrazovce. Pásy v Coupé na začátku – všechny pásy jsou vlastně barva palety 0 – ale paleta paměti 0 je změněna.

Pokud je programováno více než jedna paleta ke změně, abychom dostali rozložený řádek, později se změní jakmile je obraz ukázán a může vyvolat blikání, pokud je změněna barva na obrazovce v této jednotlivé části. Nejvíce může být sestaveno 127 výměn. Pokud si přejete opatřit vaši vlastní rutinu, to může být dodáno použitím řádky přerušení vektoru adresy (LINIV) nebo vypnutím ROM.

Systém stránkování

Systém stránkování pracuje obvykle ke stránce v nějaké adrese tak, že zde jsou nejmenší 16 kB před FFFFH je vynecháno. To značí že když transakce se strukturou dat bude delší než 16k, počátek dat je stránován v místě, kde není třeba speciálních opatření k zajištění bodů "nezhoršujících" konec paměti. ROM obvykle užívá paměti sekci C a D jako "otáčecí okno" až k paměti a udržuje systémové proměnné a hromady v sekci B. Například, když další Basicový řádek je směrován k připojení čísla k HL (připojuje levý běžný řádek k počátku běžného řádku) kódu podobnému viz níže. Jestliže se HL má pohnout z části C do části B. Pokud se tak stane, hodnota stránky vrátí a HL je upraveno k témuž současnemu bajtu na adresu 4000H a níže v sekci C. Celý řádek pak bude rozdělen do tří složitě. Poněvadž z HL bude třeba udělat ještě další řádek, jak je ukázano

```
BIT 6,H  
JR Z,LAB1
```

```
IN A,(HMPAGE)  
INC A  
OUT (HMPAGE)  
RES 6,H
```

LAB1: atd

Tady není nebezpečí z nižších pěti bitů HMPAGE získaných po 1FH k porušení vyšších bitů do té doby, než má poslední řádek programu omezovač. To bude případ největších jiných strukturálních dat které jste pravděpodobně chtěli použít.

Zdroje ROM

ROM bude použita přes stanovené určité umístění, nebo tabulkou skoku, abychom se vyvarovali problemům s nově zavedenou ROM. Odkazy jsou vyrobeny pro ROM 1.0 a ROM 1.2. Všechny produkty ve strojovém kódu obsahují ROM 1.0 nicméně volný vyšší stupeň dovoluje uživatelům výměnu za ROM 2.0; který je produkt verze ROM 1.2. Strojová výroba bude upravena na ROM 2.0 v druhé polovině roku 1990. Instalovaná verze může být určena přes: PRINT PEEK 15 a rozdělující ukázanou hodnotu po 10.

Stanovené rutiny blízko začátku nízší ROM.

0000	Stroj uveden	
0004	POP HL	Vola k návratu tvé vlastní adresy v HL
0005	JP (HL)	

Dvě instrukce nahore poskytuji schopnost k efektivnímu použití CALL bez relokování kódu. Například:

HERE	CALL 004H	; dostaváme adresu z HERE v HL
	LD DE, THERE-HERE	
	ADD HL, DE	; připojuje přestavení, abychom dostali adresu z THERE
	CALL 0005H	; efektivně, CALL THERE
	atd	
THERE	RET	

Metoda narušuje HL a DE, ale může být upravena k použití střídavého registru nebo IX nebo IY, pokud chcete.

0006	JP (IY)
0008	chybná manipulace
000F	číslo verze ROM
0010	tiskne znak v registru "A" běžnému toku. Panely "K" a "S" odpovídají kontrole kódů pro PEN, PAPER, OVER, INVERSE, FLASH, DELETE, doleva, doprava, nahoru, dolů, AT, TAB a návrat zpět
0013	tisk BC bajtů začíná z DE, používá běžný tok
0016	JP (BC)
0018	dostavá běžný znak z řádku BASIC
0020	dostavá další znak z řádku BASIC
0028	Spouští kalkulačku. Příklad násobků horních dvou čísel na kalkulačce, jak rozděluje součet na 10 a vyměňuje poslední a třetí číslo.

RST 28H	;a,b,c,d
DB MULT	;a,b,c*d
DB STKTEN	;a,b,c*d,10
DB DIVN	;a,b,c*d/10
DB SWOP13	;c*d/10,b,a
DB EXIT	

Použité operační kódy jsou rozdílné od Spectra a dalších; viz samostatná sekce na kalkulačce. Poznámka: Vektor adresy RST 28V to dělá obvykle k přenesení nějakého operačního kódu do jiného před provědením operace.

002B	DJNZ 002B	potřebný pro bezproblémové měření
002D	JP (IX)	
0030	Uživatel RST	určen pro odstranění závady atd. JPs k odstraňování v "RST 30V"
0033	JP (DE)	
0035	DJNZ 0035H	potřebný pro měření
0037	RET	
0038	přerušovače MOD 1	
005C	OUT (HMPR), A	
	JP (HL)	
005F	LD A,B	potřebný pro měření
	OR C	
	DEC BC	
	JR NZ, 005F	
	RET	
0066	nemožno odhalit přerušení	

RAM značí že rutiny ROM obvykle běží nepatrně rychleji, než identické rutiny RAM. Jestliže převedete instrukce bloku Z 80 na velké bloky a rychle je sdělite, může to být vhodné pro vyvolání rutin uvedených níže:

008F	LDI#RET	
0092	LDD#RET	blok běží o 8% rychleji v ROM
0095	CPI#RET	
0098	CPD#RET	
009B	OTI#RET	
009E	OTD#RET	

Přehled skoku v 0100H

Některé podmínky pro vstup několika bajtů po začátku rutiny jsou tak dobré, jaké jsou vstupní podmínky k rutině; kde je to udělano, můžete čist adresu rutiny z přehledu skoku, připojit

Přemístění a vložit část cesty skrz (např.použitím CALL HLJUMP) Prokázaný vstup podminek zůstane konstantní.

Když jsou rutiny popsány jako "chybné", to značí, že chybná manipulace (užití RST 0BH) bude čistit stroják zpět k hodnotě obsažené v ERRSP a pak dělat RET. Obvykle to vede k chybovému hlášení objevující se ve spodní části obrazovky, ale můžeme se tomu vyhnout přes PUSH (potlačení) adresy vaši vlastní chybné manipulace do hranice a pozměníme ERRSP (možná dočasně) k udržení běžné hranice ukazatele. Když dosáhnete ovládání, ERRNR bude obsahovat číslo chyby (viz list v BASIC manuálu)

JSCRN Volba obrazovky obsažené v registru C (1-16).Obdoba po-
(0100H) velu SCREEN v BASICu.Chyba je generovaná pokud obrazov-
ka není otevřená. Tento vstup přehledu skoku není dosa-
žitelný v ROM verzi 1.0.

JSVIN Volá parametr slova zapnuté stránky systémových proměn-
(0103H) ných.HL',DE',BC', a F jsou narušeny a přerušovače umož-
ňují zavolání před adresou parametru. Všechny další re-
gistry jsou opominuty neporušné. Všechny registry na
výstupu mimo AF' mají hodnoty vlevo od rutiny jimi vy-
volané. Originální postavení HMPR je obnoveno. Např.:

```
CALL JSVIN
DW JGETINT
LD (STORE),HL      atd.
```

JHEAPROOM Reservuje bajty BC v systému hromadění. To je vždy při
(0106H) stránkování; je to v téže stránce jako hromada a systémové proměnné, stránkuje do paměti plochy B.Na vstu-
pu BC = bajty k uložení (pokud je BC pozitivní), nebo bajty k uvolnění (pokud je BC negativní). Na výstupu,
jestliže zavedený prapor je použit, volání bylo úspěš-
né a bude DE ke starému HEAP END (začátek rezervování místa), body HL k novému HEAP END (jeden bajt uplynulý
konec rezervování paměti) a BC je nezměněno. Pokud za-
vedený prapor je rezervován, málo místa bylo přístupno
a HL udržuje množství převyšující přes přístupné místo

JWKROOM Otevření BC bajtů v pracovním místě (dočasný pracovní
(0109H) prostor, čistěn před každým BASIC údajem, je rozdělen).
Na výstupu body DE k rezervování místa začínají v sekci
C paměťové mapy ve stránkování. Body HL po jednom bajtu
končí místo, poskytuje menší délku než 16 K.A a BC jsou
vyměněny.

JMKRBIG (010CH) Otevří stránky A 16k a BC bajty v HL v sekci C paměťové mapy. BC musí být 0-3FFFH. Tato rutina může být použita k otevření místa před BASIC programem, vnitřek v ploše proměnných, nebo pracovní místo. Pokud tu není dostatečné místo, chyba bude generována. Na výstupu body HL k umístění a body DE k ukončení místa (pokud je místo < 16k)

JCALLBAS Volá subrutinu BASIC v řádku HL. Když je použit RETURN (010FH) nebo se stane nějaká chyba (např.: STOP, proměnná není nalezena atd. - dokonce "OK" v tomto případě) Váš strojový kód bude přepsán. Na oznamení A obsahuje chybný kód nebo nulu, jestliže zde nebyla chyba (a nulový prapor bude použit, jestliže zde nebyla chyba). Když Váš strojový kód poté udělá RET k návratu do BASICu, údaj zavolá kód na prvním místě tah, jako CALL 54000 nebo PRINT USR 54000, bude úplný, ale jak bude program pokračovat v údaji po RETURN v subrutině BASICu nebo po bodu, kde se chyba vyskytla, protože pojmen BASICu jeho běžné řádky a údaj má být pozměněn během vykonání subrutiny. K návratu k údaji po vašem CALL nebo někde jinde v programu z toho důvodu čtení hodnot ze SUBPPC a PPC před užitím subrutin BASICu, potom INC hodnoty SUBPPC a místo v NSPPC a místo starého PPC do NEWPPC. To bude příčinou GOTO ze specifikovaného řádku a údaj, když Váš strojový kod RETuje.

JSETSTRM Použití toku specifického A registru (FBH-10H).
(011H)

FBH = panel "B" (vždy)	tiskne binární výstup
FCH = panel "*" (vždy)	výstup stringu
FDH = panel "K" (vždy)	nízší obraz I/O
FEH = panel "S" (vždy)	vyšší obraz I/O
FFH = panel "R" (vždy)	výstup editačního řádku
00H = panel "E" (obvykle)	
01H = panel "K" (obvykle)	
02H = panel "S" (obvykle)	
03H = panel "R" (obvykle)	tiskne výstupní text

JF0MSG (0115H) Výstupy zprávy číslo A ze seznamu DE. Nula dává první zprávu. Každá zpráva by měla mít 7 bitů užitých v posledním znaku. Znaky 0-31 by neměly být obsaženy. Je použit běžný výstup stream.

JEXPT1NUM (0118H) Syntax kontrola/ocenění číselné vyjádření. Během syntaxe kontroly neviditelná. 5 bajtová forma nějakých přesných čísel je vložena. Během času výsledek výrazu je vlevo uvedeného bodu kalkulační hranice.

JEXPTSTR Syntax kontrola/ocenění výraz stringu. Podívej JEXPT1NU
(011BH) M

JEXPTEXPR Syntax kontrola/ocenění výrazu. Podívej JEXPT1NUM. Tyto
(011EH) tři rutiny mohou být použity v rozšířeném interpretu
BASIC.

JGETINT Přemístění čísla z kalkulační hranice do HL. BC obsahuje
(0121H) je opis HL a A obsahuje opis L. Chyba je generována,
jestliže zaokrouhlené číslo není v rozmezí 0 - 65535.

JSTKFETCH Přemístění poslední hodnoty z kalkulační hranice k
(0124H) AEDCB. Pokud je hodnota čísla bodu v oběhu 5 bajtů je
formováno v CPC/Spectrum. Pokud je hodnota celého čísla
mezi 0 a 65535 (které mohou být negativní), může být
ve speciální formě, kde A=0 (ukazuje speciální formu)
E=SGN (0=pozitivní, FFH=negativní) a D a C jsou nižší
a vyšší významné bajty. Pokud je hodnota string, A ob-
saahuje stránku počátečního stringu textu, DE obsahuje
počáteční adresu počátku textu na stránce (8000-BFFF) a
BC obsahuje délku stringu.

JSTKSTORE Hranice registrů AEDCB na kalkulační hranici. Na výs-
(0127H) tupu HL obsahuje nový STKEND a AEDCB jsou přeměněny.

JSBUFFET Přemístění detailů stringů z kalkulační hranice a pře-
(012AH)pis textu do buferu ve stránkovacím systému v sekci B
mapy paměti. Stránkování je klidné na výstupu, kde nepod-
statný string je opsán. Generuje chybu, pokud je string
větší délky než 255 znaků nebo je nulový. Na výstupu
body DE počátku buferu a BC obsahuje délku stringu. Re-
gistr A obsahuje opis C.

JFARLDIR Kopíruje bajty ze stránky A, odděluje HL stránky C, od-
(012DH) děluje DE. Kopíruje (PAGCOUNT) stránky 16k a (MODCOUNT)
bajty (MODCOUNT=0000-3FFF), stránkování jako potřebné.
Postavení stránkování na vstupu není důležité a je pře-
měněno na výstup. Poněvadž rutina kopíruje přes bufer,
je to obvykle možné ke psaní specializovanějších rutin
rychleji, pokud požadované stránkování může být urovná-
no.

JFARLDDR DDR verze z FARLDIR
(0130H)

JPUT Obraz pouze v mode 3 a 4. Místo bloku dat na obrazovce v
(0133H) dáných koordinatech.

Na vstupu C obsahuje koordinát x (bit 0 není významný, poněvadž data mohou být jen srovnána k správnému bajtu). Rozsah x je 0-255 jakéhokoliv módu.

Registr B obsahuje koordinát y s nulou na vrcholu obrazovky a 191 v nejnižší části.

Registr bodů HL údaje k použití, první a druhý bajt údaje určité šířky bajtů a délky pixelů, obsahující všechny bajty horního obrazu z bloku, všechny bajty druhého obrazu atd.

Registr A determinuje metodu umístění údaje na obrazovce. Odpis volbou INVERSE, 1=XOR, 2=OR, 3=AND, 4=přepis (s jinou volbou než INVERSE je rychlejší), 5=maska. Volba masky požaduje aby druhý blok údajů též délky jako první (nepočítá první 2 bajty z prvního bloku) směřoval přes HL's bity použitými pro "nemaskování" a RESET pro "maskování".

To dovoluje úplnou podobu snad s dírami k umístění na pozadí s okolními efekty. Pokud v tom bude podoba umístěna do spodní části obrazovky, bude toto náležitě upraveno. Pokud data přesahují pravý okraj obrazovky, budou zaplněna doleva, ale o 1 obraz níže.

Oba dva HL a HL' (pokud jsou použity) musí směřovat k údaji, že je bude možno použít, poněvadž JPUT bude dočasně vypnut v běžné obrazovce (použitím CUSCRNP nebo nejmenšími obsahy této adresy) v 8000H. Údaje musí být umístěny na stránce obrazovky 4000-7FFFH, nebo E000-FFFFH. JPUT také potřebuje k použití systémovou proměnnou CURP (nebo bajt této adresy) jako dočasné zásoby a bude čteno pomocí INVERT k determinování postavení INVERSE. JPUT bude bezet poněkud rychleji než PUT v BASICu, poněvadž BASIC má k použití údaje z plochy proměnných.

Na rozdíl od atributů systémů barev, barevy k použití jsou zahrnuty v bloku dat, ačkoliv metody 0-3 umožňují volbu INVERSE. Omezení těchto bloků musí být položeno dole v sedmém pixelu hranic k udržení vysoké rychlosti.

Pokud si přejete umístit bloky v jedno pixelových intervalech je to možno připravit dvěma možnostmi též grafiky s daty přemístěnými po 1 pixelu v jednom z nich, v pořadí k vytvoření požadovaného efektu.

Ve verzi 1.0 ROMky je chyba v JPUT, ale můžete dostat tentýž efekt použitím: PUSH AF: CALL 3A7BH: POP AF: CALL 2DACH místo CALL 0133H. Pokud číslo verze ROM (PEEK15) není 10, musíte použít 0133H.

JGRAB Obraz v modu 3 a 4. Nakupení bloku obrazu
(0136H) data daných koordinátů buferu

Na vstupu, C obsahuje koordinát X (bit 0 není významný) a rozsah X je 0-255 jakéhokoliv obrazového módu. B obsahuje koordinát Y s nulou na vrcholu obrazovky a 191 dole. E obsahuje šířku bloku v bajtech a D obsahuje výšku v pixelech. Na výstupu, kdy DE počátku bloku v běžné stránce obrazovky (specifikován přes CUSCRNP); je první bajt vždy 0, další bajt je šířka v bajtech,

další bajt je výška v pixelech a následující bajty nakupení bloku, rozkládání řádku po řádku. BC obsahuje blok výšky zahrnující 3 hlavnější bajty. Od té doby je plocha uložena v "záloze" 8k sledující mód obrazu 3/4, výška údaje musí být menší než 8k. JGRAB požaduje dočasné uložení v proměnné CURP (nebo uložení na této adrese) a to bude číslo CUSCRNP v postupu dočasné zapnuté obrazovky. Jestliže GRABovaná plocha je položena níže na obrazovce, bude upravena podle potřeby.

JFLOT (0139H) Nakresli pixel koordinátu X v C a koordinátu Y v B, kromě toho, když mód je 3 a platí FATPIX 0 (THFATT bude 0) v každém případě použít HL jako koordinát X (0-511) a kresli "tenký" pixel. Osa Y má 0 nahoru a 191 dolů ve všech případech.

Není zde kontrola k zabránění Vašeho kreslení přes nejmíničší část obrazovky.

JDRAW (013DH) Vyvolá řádek z běžné pozice C (nebo HL, jestliže "tenké" pixely jsou použity; shodne s THFATT) pixely vodorovně a B Pixely svisle. D=1 pro dolní (nebo ne pro pohyb Y) a FF pro horní, E=1 pro pravý (nebo ne pro pohyb X) a FF pro levý. Barva, OVER a INVERSE postavení přichází z dočasných grafických proměnných. Používaná obrazovka přichází z CUSCRNP. Chyba je generovaná, pokud je řádek vyvolán přes hranu obrazovky.

JDRAWTO (013FH) Vyvolá řádek z běžné pozice bodu C,B (nebo bod HL,B) pokud "tenké" pixely jsou použity. Osa Y má 0 nahoru a 191 dolů. Barvy atd. přicházejí z dočasných grafických proměnných.

JCIRCLE (0142H) Vyvolá kružnici v C,B bodě o poloměru A. Pokud "tenké" pixely jsou užity, HL smí obsahovat počátek okolí levé strany obrazovky (v pixelech). To umožňuje umístit kružnici na obrazovku v módu 3 dokonce přes to, že koordinát X je limitovan 1 bajtem. Osa Y má nulu nahoru. Barvy atd.. přicházejí z dočasných grafických proměnných.

JFILL (0145H) Pouze pro mód 3 a 4. Používá vzor 16 * 16 pixelů v DE k vyplnění plochy začínající koordináty B,C. Osa Y má nulu nahoru. Pokud DE obsahuje 0 "jemný" vzor s běžným PEN (ze M23INKT) je vykonán. Jinak DE může mít nějakou hodnotu (ale hodnoty 0001 - 3FFFH budou použity ROM jako vzor!). Pokud A=0, potom je udělán předběžný přenos paměti obrazu k poškrábání plochy (délka 6k, hned po paměti obrazu 24k), budete potřebovat udělat A=0, použijete první nejménší vzor. Později vzory na téže obrazovce.

zovce přes tutéž barvu můžete udělat A nenulové pro větší rychlosť.

Jestliže používáte MODE 3 s "tenkými" body, HL smí obsahovat být počátku levé strany obrazovky pro bod počátku vyplňování. To umožňuje vyplňovat některou část obrazovky, navzdory tomu, že koordinát X je limitován jedním bajtem.

Vzor dat stupňovaných přes DE má 8 bajtů v horní řadě, 8 bajtů v další atd., až po 128 potřebných. Každá barva řídí ubírání po 1 pixelu v módu 4, nebo 2 pixely v módu 3.

JBLITZ (0148H) Provádí seřazení grafických povelů délky BC v DE..DE může být cokoliv ze 4000H až FFFFH a BC cokoliv od 0 do FFFFH (JBLITZ bude vyvolán, pokud to bude požadováno). Povel PLOT, DRAW, DRAW TO, CIRCLE, OVER, INK, CLS a PAUSE mohou být zahrnuty ve stringu.

Jsou ovládány ve vyšší rovině než JPLOT, JDRAW a JCIRCLE pohyblivá kalkulační čárka a speciální BASIC, proměnné mohou být použity pro výstup nebo změnu měřítka. Použití početních systémových proměnných stránuje systémové proměnné v nesnadné vyhnuti se.

Koordináty X a pohyby jsou dovoleny pouze v rozmezí 0 až 255; jen polovina obrazu je snadněji dostupná v módu 3 s použitím volby FATPIX 0. Použitý souřadnicový systém je tentýž jako při použití BASICu; bajty 01 00 00 budou pracovat stejně jako PLOT 0,0 v BASICu a upraveny pokud XOS nebo YOS byly pozměněny. Předvedení RELATIVE DRAW:

00 nebo FF dává SGN pohybu po ose X (00=kladné, FF=záporné) následuje přes pohyb X, následuje SGN pohybu po ose Y a pohyb Y. Pokud jsou pohyby záporné, jsou nahromaděny jako 256 pohybů.

PLOT:	01 následován přes x,y
DRAW TO:	02 následován přes x,y
CIRCLE:	03 následován přes x,y,r
OVER:	04 následován přes 0-3
PEN:	05 následován přes 0-17
CLS:	06 následován přes 0-1
PAUSE:	07 následován přes 0-255

Nejsnadnější cesta k přípravě dat pro JBLITZ je použitím RECORD v BASICu a pak POKE string, kde to potřebujete.

- JROLL Pouze pro módy 3 a 4. Pohyb části obrazu doleva, doprava, nahoru nebo dolů. Plocha obrazovky musí mít stejný počet pixelů. Horizontální pohyb přes 1 pixel nebo je možné jakékoliv stejné množství pixelů. Vertikální pohyb je možný přes jakékoliv množství pixelů. Cyklické přetáčení je volitelné; používá plochu okamžitě po obrazové paměti jako vyrovnávací paměť. Na vstupu, A je 00 pokud chcete SCROLL (ne obal) nebo FFH pro ROLL (s obalem). B obsahuje pixely k pohybu, C je řídící kód (1=doleva, 2=nahoru, 3=doprava, 4=dolů), D obsahuje délku bloku v pixelech (bit 0 není významný). HL obsahuje souřadnice levého horního rohu bloku, L vlastní souřadnici X (řada 0-255, jakýkoliv mód) a H vlastní souřadnici Y s 0 na vrcholu obrazovky. JROLL kontroluje systémové proměnné MODU, opisuje stálé grafické proměnné do dočasného MODu a používá M23PART jako barvu pozadí pro novou obrazovku generovanou přes SCROLL. Rutina používá CURP a TEMPB3 v systemových proměnných a nahoru k 0181H bajtům v 4D00H k "rozvinutí" kódu smyčky.
- JCLSBL Čistí celou obrazovku jestliže A=0, jinak čistí vyšší obrazovku.
- JCLSLOWER Cistí nižší obrazovku. Také vybírá panel k (nižší okno (0151H) vstup / výstup)
- JPALET Jestliže A=FFH, klade barvy BC do PALTAB jako hlavní a blýskavé barvy pro paletu barvy E. (Dělá B=C pro neblýskání). Barvy budou 0-127, vrstvy palety 0-15. Jestliže A je různé od nuly je souřadnice Y v paletě změněna PALTAB není pozměněn; místo vstupu je udělán v řádku přerušení seznamu barev.
- JOPSCR OPEN SCREEN (otevření obrazovky) čísla C v módu B. Požadované stránky jsou umístěny v Page Allocation Table (přehled rozvržení stránek) a rezervovány a vstup je udělán v SCLIST. Číslo obrazovky musí být 2-16 a nemusí být již otevřena, nebo je generována chyba.
- JMODE Udává obrazový mód v A registru (0-3 dává módy 1-4)
- JTDUMP Hromadí obraz textu (pokud je toto instalováno)
- JGDUMP Hromadí grafický obraz (pokud je toto instalováno). Systémové proměnné dokumentovány v BASIC manuálu řídí rozsah hromadění a jeho zvětšení

JRECLAIM Získání uzavření BC bajtů v HL (8000H-BFFFH) kromě toho (0163H) mohou být použity k uzavření paměti před BASIC programem nebo v ploše proměnných nebo pracovní oblasti. BC musí být menší než 16k; pokud chcete získat větší prostor volejte i bajt z počátku rutiny s A vlastního čísla k získání stránek 16k v doplňku prostoru specifikovaném v BC.

JKBFLUSH Zarovnání klávesnice vyrovnávací paměti (0166H)

JREADKEY Čte klávesnici, zarovnává ITKEYvnávací paměť (podobný (0169H) INKEY\$). Z / NC jestliže není klávesa stlačena, jinak NZ / CY a A=hodnotě klávesy (ASCII)

JWAITKEY Čte další klávesu do A z klávesnice vyrovnávací paměti. (016CH) Čeká na stlačení klávesy kterou potřebujete.

JBEEP Dělá zvuk cyklu dlouhého DE-1 v periodě postavení (016FH) HL * B T

JSAVE Uchová blok bajtů délky CDE začínající v HL (přepnut do sekce C). C obsahuje čísla stránek 16k a DE obsahuje délku MOD 16k. A registr i bajt uschová (FFH pro "blok údajů", 01 pro "hlavičku" SAM, 00 pro "hlavičku" Spectrum) Pokud SLDEVT (viz systémové proměnné) obsahuje "T" SAVE bude použit pro pásek; pokud obsahuje "N" SAVE bude použit pro síť. SLNUMT řídí rychlosť nahrávky na pásek (112 je standartní rychlosť)

JLOAD Pokud je použit přenosový příznak, vkládá LOAD CDE bajty do HL (přepíná do sekce C paměťové mapy). Pokud je přenosový příznak obnoven, dělá VERIFY. Registr A obsahuje typ údajů k rozdělení (FFH= blok dat, 01= hlavička SAM nebo Spectrum). Pokud SLDEVT obsahuje "T" bude použit pásek; pokud "N" potom síť. Kolísavé vstupy rychlosťi pásku jsou ovládány automaticky. Rutina vrací přenosovou sestavu pokud operace byla úspěšná; jinak je zde chyba.

JLDVD Pokud je použit přenosový příznak, vkládání (LOAD) bajty CDE dat (ne hlavičku) do HL. Pokud je přenosový příznak obnoven, dělá VERIFY. Použití disku, pásku nebo síť podle obsahu SLDEVT. Chyba je generována, pokud operace schází.

JEDGE2
(017BH) Čeká na 2 přechodové signály na EA, vraci čísla 47-T v A a C postavení jednotek. Na vstupu B smí obsahovat běžné postavení EAR v bitu 6, bit 3 smí být použit a další byty smí odrážet co chcete odeslat na port BORDER, když je zde přechodový signál. Bity 2-0 budou rezervovány pro každý přechod. Registr D' bude obsahovat hodnotu určující délku mezery mezi nalezením jednoho přechodu a připravou na jiný (vyvaruje se dvojitému počítání, když signál z pásku má ostré zvuky). Pro standartní rychlosť pásků smí být D' okolo 46H, JLOAD používá D' podle rychlosti pásku. Přerušení smí být vyváženo pro správné výsledky.

Rutina vraci Z, jestliže měla čekat příliš dlouho (256 jednotek), NC, jestliže ESC bylo stisknuto a CY pokud je vše OK. Vstup 2 bajtů na začátku zbabuje účinnosti počáteční nulování registru C, který je používán jako čítač, vstup 6 bajtů na začátku čeká na 1 přechodový signál na EAR. Registr B obsahuje na výstupu novou hodnotu vhodnou pro jiné volání rutiny a A a C obsahuje hodnotu čítače.

Poznámka: ve verzi ROMky 1.0 tento vstup seznamu skoků má chybně použitou adresu bitu 15. Můžete číst adresu 017CH, obnovit bit 15 a volat tuto adresu.

JSTRS
(017EH) Vytváří verzi ASCII čísla kalkulační pohyblivé řádové čárce ve vyrovnávací paměti 5BA0H. Na výstupu DE obsahuje 5BA0H a BC obsahuje číslo znaků ve vyrovnávací paměti.

JSENDA
(0181H) Odesílá bajt v registru A na paralelní tiskárnu. Systémová proměnná 5A10H je čtena k získání užívaného portu a rutina cykluje, dokud není tiskárna připravena k odeslání a potom navrácení. Pokud je ESC stisknuto, chyba bude generována. Používá se jen pro A a BC.

Seznam skoků vstupů uvedených do ROM 1.0

JMCHAR
(0184H) Volá subrutinu (podprogram) SCREEN\$. Zkuste porovnat znak v řádku D, sloupcí E na obrazovce s jiným znakem v hlavním souboru znaků (bodování přes CHARS) nebo nějaký znak bodování přes UDG. Na výstupu je použit přenos příznaku, pokud hledání bylo úspěšné, v kterém případě A obsahuje kód znaku 20H - 7FH (32-127), jestliže znak byl nalezen v hlavní sestavě nebo 80H - ABH (128-168) pokud byl znak nalezen v sestavě UDG. Pokud byl přenos příznaku obnoven A=0 a hledání se nezdařilo.

JGRCOMP Podprogram je použit na obslužný grafický výpis
(0187H)

JGTTOK Zkuste porovnat možné hláškování mimo klíčové slovo bo-
(018AH) dované přes DE v seznamu klíčových slov v HL + 1. Seznam
by měl v krajním případě s bitem 7 dát krajní písmeno
každého klíčového slova. Registr A obsahuje číslo slov
v seznamu plus 1. Na výstupu byl nalezen nezaložen,
jestliže je dán nulový příznak. Pokud je nulový příznak
obnoven, registr A obsahuje vstup čísla v seznamu (1 ne-
bo více), body HL počátku původního slova a body DE
právě uplynulého ukončení

JCLSCR Uzávírá obrazovku čísla obsaženého v registru C.
(01BDH)

Postavení znaku

Daný znak z CHR\$ 32 až 130 je uložen ve stlačené formě v
ROM 1 a vložen do RAM ve stránce systémových proměnných, potom
přístroj je zapnut. CHR\$ 131 až 168 jsou zpočátku jako čisté,
ačkoli CHR\$ 128 až 143 jsou vyloženy jako grafický blok, ledaže
má být použit "BLOCKS 0". (Grafický blok forem je generován jako
potřebný – daný znak je neobsahuje). Znaky, nehledě na grafický
blok, zabírají 6 * 8 bodů, který je vhodný pro zobrazení v in-
tervalech 6 nebo 8 pixelů.

Znak je ohrazen 8 bajty z vrchní roviny pixelů k nejnižší
rovině. Levý a pravý sloupec pixelů tohoto bloku 8 * 8 nejsou
použity, když tisk je v intervalech 6 pixelů. Daný znak má zák-
ladní adresu 5190H a prostor je k dispozici CHR\$ 32-168 v jednom
souvislému bloku. Další systémová proměnná – HUGD – může být dána
nahoru k bodu ohrazení CHR\$ 169-254, ale standartní schema pro
znaky 128-168 je vytvořeno (soubor "font"), který obsahuje úplnou
sadu cizích znaků, který porovná znaky akceptované na nejčetněj-
ších tiskárnách. Jejich použití je doporučeno, takže softvare
může být snadněji převeden pro použití na veřejnosti.

VEKTORY

Tyto důležité rutiny ROM dovolují zastavení uživateli. Obsahují adresy, které budou volány, jestliže (a jen jestliže) nejvýznamější bajt není nulový. Dává volbu RET (pokud se nerozhodnete k pojednání podrobného návodu) nebo vyřazuje zpáteční adresu a otáčí vektor do užitečného skoku.

Různí programátoři mohli chtít použít tentýž vektor. Pokud Vaše programy prostě přepisuji vektory, jsou pravděpodobně nekompatibilní s budoucími užitečnými vektory z MGT a jinic softwarových doplňků. Doporučujeme tyto Vaše programy uložit hodnotou starého vektoru. Bohužel, automatický spojovací mechanizmus to dělá velmi snadno vynecháním z ROM, ale řešení je ještě docela jednoduché. Např: připojení kódu k přípravě též činnosti pro projev STEPu.

Inicializace CMDV, avšak zásoba prvního skutečného vstupu.

```
INIT      LD HL, (CMDV)
          LD (STORE), HL
          LD HL, MYPATCH
          LD (CMDV), HL
          RET

MYPATCH  CP STEPTOKEN
          JR Z, MYP2           ; JR pokud se zajímáme

          LD HL, (STORE)
          INC H
          DEC H
          RET Z               ; RET, jestliže adresa nebyla v CMDV
                                ; a INIT
          JP (HL)              ; nechá běžet jinou rutinu

MYP 2    ( Kód poskytující tutéž akci pro STEP )
          RET
```

NB Všechny příklady jsou částí programu, nekompletních programů a nedělá, např. obsah shora uvedeného kódu, který bude obsažen ve skutečných aplikacích.

Seznam Vektorů

DMPV Volán pokud je kličové slovo DUMP použito v reálném čase. Body CHAD k ODH nebo SAH, pokud bylo právě DUMP použito, jinak bylo použito DUMP CHR\$.

SETIYV SETIY je volán, když DRAW, PLOT a CIRCLE " rozhoduje " jaký Pixel předkládá rutinu k použití a volá SETIYV, jestliže jeho nejvýznamější bajt není nulový. Body SETIY registru IY vhodné rutiny a ustanovení vektoru je určeno k poskytnutí kreslících módů, tak jako tečkových řádů všech grafických pixelů, které byly přepsány. Na vstupu k SETIYV, A obsahuje běžný mód - 1 obrazovky.

Návrat k PLOT, DRAW a CIRCLE registru IY může bod pixelu použité rutiny a pouze A a HL může být jinak změněn. V módu 4 a módu 3, když jsou vybrány " tlusté " pixely, hodnota A na vstupu je průchodná k D' přes grafické povely a může být použita k obsahu barvy PEN, pokud si přejete.

Jestliže změníte SETIYV, jako všechny vektory, udělá Vaše rutina právě RE, připraví obvyklou ROM rutinu k provedení.

Takto Vaše kódy smí začít s ověřením módu, jestliže je to jedno, chcete rozdělit ROM a RET.

Jinak POP-zpáteční adresu tak, že rutina ROM není nikdy použita a použije své IY. Další RET bude pro DRAW, PLOT nebo CIRCLE.

Pixel používající rutinu opatříte adresou v IY smí nakreslit pixel v L,H a návrat jen výměnou AF. Použit souřadnicový systém má 0,0 v levém horním rohu na obrazovce. Můžete využít poznámky použité pro běžné OVER, INVERSE a PEN. (ve kterém případě urychlují průběhy můžete najít nahore, pokud je podobná ROM, použijete IY rozdílné rutiny pro tytéž volby) Rutina bude ukončena JP (IX) nikoli RET

PRTOKV Tisk záznamu. A = záznam kódu k rozšíření a tisku jako ASCII.

NMIV Neschopen odhalit přerušení. Obvykle obsahuje adresní rutiny, které vrací do BASICu. Když je stisknut knoflík BREAK. NMIV přepíná dočasně systém stránkování a připravuje zde jeho vlastní zásobník, jakýkoli stav stránkuje, když byl přerušovač generován. Na vstupu postavení

LMPR, když NMI je nahromaděn v NMILRP. Ukazatel zásobníku, když NMI je nahromaděn v NMISP a registry AF a HL mají již být poháněny až k použitímu zásobníku, když NMI je použit (který může být stráncován mimo).

- FRAMIV
(5AE2H) Volán přes rámec přerušení 50* za sec. Všechny hlavní registry mohou být použity.
- LINIV
(5AE4H) Volán přes řádek přerušení, který je generován, když hodnota v řádku přerušení registru se hodí k použití rozkladu. Tento vektor může být použit ke změně palety, výměně obrazovky nebo obrazového módu atd. ve střední obrazovce. K synchronizaci takových změn se zpětným rozkladem je možné čekat na výměnu portu HPEN k ukázání že číslo rozkladu bylo právě vyměněno. (to bude scházet jestliže světelné pero je zapojeno). Mohou být použity všechny hlavní registry.
- COMSV
(5AE6H) Je volán přes přerušovač COMMS
- MIPV
(5AE8H9) Je volán přes vložený přerušovač MIDI. Na vstupu má registr A být právě z portu MIDI IN.
- MOPV
(5AEAH) Je volán přes přerušovač výstupu MIDI
- EDITV
(5AECH) Je volán dříve než je použit editor pro úpravu řádku, nebo INPUT.
- RSTOBV
(5AEEH) Volán častěji při chybné manipulaci A=chybný kód. Střídání registrů má být vybráno, CHAD a CHADP mají být kopírovány k XPTR a XTRP, body DE chybného kódu, tento sleduje RST 0BH a adresa 1 bajt po tomto kódu je na zásobníku nepřehlédná adresa rutiny chybné manipulace. Odstranění zásobníků nemá být uděláno v tomto období.
- RST2BV
(5AF0H) Volán přes kalkulátor pohyblivé řadové čárky, kde A=další kód v seznamu následující instrukce RST 2BH. Body DE k ukončení zásobníku kalkulátoru pohyblivé řadové čárky. Registr IX na výstupu nemusí být vyměněn a bod DE k ukončení zásobníku kalkulátoru pohyblivé řadové čárky. (to může být rozdílné od vstupní hodnoty, jestliže polozky mají být připojeny, vyřazeny nebo kombinovány). Viz příklad níže.

RST30V
(5AF2H) Uživatel RST. Skákán (ne volány shodné jiné vektory) přes RST 30H. (ROM 0 používá RST 30H k poskytnutí snadně "CALL ROM 1", ale to se vztahuje pouze na instrukce RST použité uvnitř ROM 0). Všechny registry jsou opo- minuty nedotčené.

CMDV
(5AF4H) Volán z registru A obsahující kód znaku k přezkoušení, nebo provedení syntaxe (obvyklý povelový kód). Může být použit k připojení povelů nebo modifikace působení existujících povelů.

EVALUV
(5AF6H) Oceňovač výrazu. Volán s A = běžný znak ve výrazu; může být použit k připojení nových funkcí.

LPRTV
(5AF8H) Volán protože každý znak kódu nad 127 je ovládán ovladačem tiskárny, když panel "P" je použit. (Úplnější kontrola může být získána pozměněním panelů "P" nebo "B". Vstupní adresy - viz níže).

Na vstupu k rutině je znakový kód v registru A. Např: k přeměně CHR\$ 130 do "A":

INIT LD HL, PATCH
LD (5AF8H), HL
RET

PATCH CP 130
RET NZ

OP BC ; kousek zpáteční adresy
LD A, 65
RST 10H ; tisk znaku
LD HL, 5A70H ; tiskárna proměnného sloupce
; (SVAR &70)
INC (HL) ; 1 sloupec doprava
RET

Místo použití RST 10H k tisku nových znaků můžete dávat přednost k použití jedné ze dvou možností CALL 0181H (odesílá A paralelní tiskárně) nebo pravděpodobně lepší, volá následující kód k odeslání registru A do panelu "B":

CHBOF: LD HL, (5C4FH) ; báze kanálů adresy
LD DE, 25 ; (užívá 15 bodů kanálu "P")
ADD HL, DE
LD E, (HL)

```
INC HL
LD D, (HL)      ; DE = kanál B výstupní adresy
EX DE, HL
JP (HL)
```

Kanál "B" ovládá všechny výstupy okamžitě; panel "P" reaguje na řídící kódy, připojuje auto-LF atd. a potom je vysílá mimo panel "B". Něco podobného ESC;"R";CHR\$ O bude vysílat přes kanál "B". Kanál "B" obvykle vystupuje použitím 0181H, ale pokud použijete CHBOP raději než příkaz 0181H, bude to snadnější pro další převedení tiskárny na jinou možnost - sériový interfejs. (Přes změnu výstupní adresy na kanálu).

MTOKV (SAFAH)	Volán pokud síla klíčového slova není poznána přes ROM. Na vstupu, body DE odpovídají síle klíčového slova. Na výstupu budou podmínky rovny seznamu skoku vstupu JGTTO K.
MOUSV (SAFCH)	Volán při každém použití. Když je tento vektor použit, prohledá vlastní ROMku, pokud je myš mimoběžná.
KURV (SAFEH)	Volán před editací, kurzor je tištěn (volání adresy v MNOP). ROM 1 je zapnuta. Může být použito k doplnění "otevření" nebo další kurzor.

Další příklad: Předpokládejme, že máte množství grafických podprogramů ve strojovém kódu, který požaduje zapnutou obrazovku ve 4000H. Bude to přesná modifikace USR, takže jiné použití podprogramu zapíná požadovanou stránku dočasně, délá to rutina níže, pro strojový kód je určeno místo v horní polovině paměti. INIT je volán ke změně vektoru kalkulátoru pohyblivé řádové čárky bodu USRPTCH; kdykoliv kalkulátor ovládne USR, zadřížme to k oklamání stránek k umístění obrazu ve 4000H (dává pozor na zásobník) a volá adresu USR, rovná BC zpět na zásobník kalkulátoru pohyblivé řádové čárky jako výsledek USR a vraci DE směřující do STKEND a IX nevyměněn. (všechno očekáváno přes kalkulátor).

USRPTCH musí být v systému stránky zárukou, že je trvale uložen v paměti když je vektor volán; nejlepší místo je v systému Heap. PART2 musí být v horní polovině paměti, ale v jiném případě je to dobrý nápad k použití krátké rutiny v Heap k volání rozsáhlé rutiny na jiné stránce poněvadž drží Heap nahoře.

```
INIT    LD HL,USRPTCH
        LD (RST28V),HL
        RET
```

USRPTCH	CP 45H	; kód USR
	RET NZ	; drží jednoduché věci a ignoruje jakýkoliv jiný, jenž chce použít tento vektor
	POP BC	; část adresy RET
	IN A, (HMPR)	
	PUSH AF	; vkládá běžné postavení HMPR
	LD A,1	; OK pokud PART2 je na straně 1 nebo 2
	OUT (HMPR),A	; strana v PART2
	IN A, (LMPR)	
	AND 0BFH	
	OUT (LMPR),A	; ROM 1 vypnuta
	CALL PART2	
	IN A, (LMPR)	
	OR 40H	
	OUT (LMPR),A	; ROM 1 znova pro RET kalkulátoru
	POP AF	
	OUT (HMPR),A	; pravé postavení HMPR obnoveno
	RET	
PART2	LD (STORE),SP	; budeme vypínat zásobník, tak vložíme běžný SP
	LD SP,TSTACK	; vlož nový zásobník někam jinam >=B000H
	PUSH AF	; postavení LMPR
	PUSH IX	; kalkulátor nepotřebuje změnit IX
	CALL JGETINT	; dává adresu USR v BC
	IN A, (VPAGE)	
	AND 1FH	; A = stránka obrazovky
	DEC A	
	OUT (LMPR),A	; stránka ve 4000H
	CALL BCJUMP	; volá adresu USR
	POP IX	
	POP AF	
	OUT (LMPR),A	; počáteční postavení LMPR
	LD SP,(STORE)	; počáteční SP
	XOR A	
	LD E,A	
	LD D,C	
	LD C,B	
	CALL JSTKSTORE	
	EX DE,HL	
	RET	
STORE	DW 0	

Povelů kódů kalkulátoru pohyblivé řádové čárky

Kalkulátor pohyblivé řádové čárky vykonává na Coupé všechny matematické operace a mnoho řetězcových operací. Poskytuje jednoduchou cestu k programování celého komplexu otázek v různých jazycích, což může být velmi vhodné. Kalkulátor je volán použitím RST 28H, Řídící přes sériové kódy povelů (které mohou mít parametry). Seznam ohraničuje speciální kód. Data zpracovávaná tímto kalkulátorem jsou umístěna ve speciálním zásobníku, kde budeme odkázání na FPCS. Všechny body mají délku 5 bajtů. Přehled skoku dovoluje registry poslat do a ze zásobníku a registr B je poslán do A z kalkulátoru.

Čísla jsou obsažena ve dvou formách; celkový tvar obsahuje celá čísla mezi -65535 a +65535, jako 0, SGN (0 pro kladná, FFH pro záporná), méně důležité a více důležité bajty a poslední nulový bajt. Např.: 80H je uložen jako 00 00 80 00 00, 4001H jako 00 00 01 40 00, -1 jako 00 FF FF FF 00 a -80H jako 00 FF 80 FF 00 (Záporné hodnoty jsou uloženy v záporném tvaru (65536 minus číslo).)

Pokud jsou čísla v celkovém tvaru, kalkulátor je často schopen speciálního použití, rychlejších forem operací jako vícenásobné odečítání a sčítání. Čísla mimo -65535 až +65535 musí být obsažena v rozdílné formě, jakožto exponováno velkým množstvím volaných čísel, násobeno jiným číslem volané mantisy. První bajt je exponent + 80H a další 4 bajty jsou mantisa. Je to vždy mezi 0,5 a právě menší než 1, který znamená, že první bit je vždy 1, dovolující aktuální použití jako bit SGN(0 pro plus, 1 pro minus).

Toto schema dovoluje uložit hodnoty mezi 1E38 a (1,7E)na -39 s přesnosti na 9 nebo 10 číslic. Je důležité poznamenat, že některá dilčí čísla (asi 1/10) nemohou být uložena přesně v této binární formě (právě tak např. 1/3 nemůže být znázorněna přesně v desetinné formě). To znamená, že pokud odečítáte 0,1 od 1 desetkrát, výsledek není přesně 0. To není chybné – správné vymezení číselných systémů si budete uvědomovat.

Řetězce jsou obsaženy jako stránky 16K počátku řetězcového textu, použití počátku řetězcového textu (8000H – BFFFH, LSB/MSB) a délka řetězce (LSB/MSB)

V seznamu níže je symbolicky ukázána činnost mnoha kódů kalkulátoru. Kde povelový kód rozděluje dvě čísla (násobek), poslední číslo FPCS je poukázáno jako N2 a číslo níže je poukázáno jako N1. Pokud jsou hodnoty řetězce použity \$1 a \$2.

00H MULT	N1 * N2
01H ADDN	N1 + N2
02H CONCAT	\$1 + \$2; připravuje spojování textu \$1+\$2 v pracovní oblasti a zásobník detailů dočasného řetězce

03H SUBN	N1 - N2
04H POWER	N1 na N2 (mocnina)
05H DIVN	N1/N2
06H SWOP	výměna V1,V2
07H DROP	vyřazení poslední hodnoty
08H MOD	N1 MOD N2
09H IDIV	N1 DIV N2
0AH BOR	n1 BOR N2
OBH	reservováno
0CH BAND	N1 BAND N2
ODH NUOR	N1 nebo N2
OEH NUAND	N1 a N2
OFH NOTE	N1<>N2 ;porovnává shodné <> vymazává dvě porovnané hodnoty z FPCS a nahrazuje je 1 nebo 0. (správně nebo špatně)
10H NLESE	N1 <= N2
11H NGRTE	N1 >= N2
12H NLESS	N1 < N2
13H NEQUAL	N1 = N2
14H NGRTTR	N1 > N2
15H SAND	\$1 a N2
16H SNOTE	\$1 <> \$2
17H SLESE	\$1 <= \$2
18H SGRTE	\$1 >= \$2
19H SLESS	\$1 < \$2
1AH SEQUAL	\$1 = \$2
1BH SGRTTR	\$1 > \$2
1CH SWOP13	vyměňuje poslední hodnotu a hodnotu 2. od konce
1DH SWOP23	vyměňuje hodnotu 2. od konce a hodnotu nižší

Tři povely kódů níže po DECB jsou použity pro přemístění bajtu k přeskočení v seznamu povelů dopředu nebo dozadu (podobně jako GOTO v BASICu, nebo JP v jazyce symbolických instrukcí – assembleru). Přemístění je jiné mezi přemístěním bajtu místem, kam chcete skočit; pokud je větší než 7FH, je zpracován jako skok dozadu, například FFH je dozadu přes 1.

1EH JPTRUE	Skok vede přemístění daným příštím bajtem, pokud je poslední hodnota 1. (poslední hodnota je vyřazena)
1FH JPFALSE	Skok vede přemístění daným příštím bajtem, pokud je poslední hodnota 0. (poslední hodnota je vyřazena)
20H JUMP	Skok vede přemístění daným příštím bajtem.

Kalkulačor má speciální strádač volaný přes BREG, registr B procesoru Z80 obsahuje jakýkoliv počátek, když byl kalkulačor volán

Níže jsou uvedeny čtyři kódové povely použitím BREG.

21H LDBREG	vkládá BREG s příštím bajtem
22H DECB	Decrement BREG a skok vede přemístění dané příštím bajtem, pokud BREG není nulové.
23H STKBREG	ukládá BREG na FFCS jako číslo mezi 0 - 255.
24H USEB	bere další kód kalkulátoru z BREG
25H DUP	kopie vrcholu vstupu FFCS
26H ONELIT	zásobník dalšího bajtu na FFCS (jako číslo mezi 0 a 255).
27H FIVELIT	zásobník dalších 5 bajtů na FFCS (jako jakékolič číslo).
28H SOMELIT	zásobník čísla bajtů specifikovaný přes další bajt používá (další bajt + 1) jako první bajt. Dovoluje zásobování mnoha 5-ti bajtových hodnot.
29H LKADDRB	používá další dva bajty jako adresu, zásobník získal bajt této adresy na FFCS
2AH LKADDRW	používá další dva bajty jako adresu, zásobník získal slovo (2 bajty) této adresy na FFCS
2BH REDARG	převádí argument ze SIN
2CH LESSO	zásobník 1 (TRUE) nebo 0 (FALSE), podle kterého je poslední hodnota <0. Poslední hodnota je vyřazena.
2DH LESEO	zásobník 1 nebo 0, podle kterého je poslední hodnota <=0. Poslední hodnota je vyřazena.
2EH GRTR0	zásobník 1 nebo 0, podle kterého je poslední hodnota >0. Poslední hodnota je vyřazena.
2FH GRTE0	zásobník 1 nebo 0, podle kterého je poslední hodnota >=0. Poslední hodnota je vyřazena.
30H TRUNC	usekne číslice po desetinné čárce v poslední hodnotě.
31H RESTACK	přeměňuje poslední hodnotu v plný 5 - ti bajtový tvar, jestliže není v tomto tvaru připraven.
32H POWR2	připočítává 2 ke kapacitě poslední hodnoty
33H EXIT	končí použití kalkulátoru pohyblivé řádové čárky
34H EXIT2	ukončuje použití kalkulátoru pohyblivé řádové čárky a dělá RET
35H	reservováno
36H	reservováno
37H	reservováno
38H	reservováno

Kódy povelů uvedených níže jsou použity k poskytnutí co největšího množství funkcí v BASICu.

39H SIN	4CH EOF
3AH COS	4DH PTR
3BH TAN	4EH reservováno

3CH ASN	4FH UDG
3DH ACS	50H NUMBER
3EH ATN	51H LEN
3FH LOGN	52H CODE
40H EXP	53H VAL\$
41H ABS	54H VAL
42H SGN	55H TRUNC\$
43H SQR	56H CHR\$
44H INT	57H STR\$
45H USR	58H BIN\$
46H IN	59H HEX\$
47H PEEK	5AH USR\$
48H DPEEK	5BH INKEY\$ #
49H DVAR	5CH NOT
4AH SVAR	5DH NEGATE
4BH BUTTON	

Kalkulačor má šest 5-ti bajtových pamětí k umístění dočasných výsledků. Následují kódy zásobníku a přivolání paměti zpět.

C8H ST00 kopíruje poslední hodnotu paměti 0-5 kalkulačoru,
C9H ST01 pak toto vymazává z FPCS
CAH ST02
CBH ST03
CCH ST04
CDH ST05

DOH ST00 jako předešlo, ale nemaže poslední hodnotu po kopírování
D1H ST01
D2H ST02
D3H ST03
D4H ST04
D5H ST05

DBH RCL0 zásobník hodnoty v kalkulačoru paměti 0-5 na FPCS
D9H RCL1
DAH RCL2
DBH RCL3
DCH RCL4
DDH RCL5

Kódy níže popsané poskytuji kompaktní metodu hromadění společných konstant na FPCS

E0H STKHALF zásobník 0.5
E1H STKZERO zásobník 0
E2H STK16K zásobník 16384

E6H STKFONE	zásobník 5-ti bajtů ve tvaru po 1
E9H STKONE	zásobník celého čísla ve tvaru po 1
ECH STKTEN	zásobník 10
FOH STKHALFFPI	zásobník PI/2

Systém pásku

Zde máme pět typů souborů pásky; BASIC program; Machine Code (strojový kód); Numeric Array (číselná řada); String Araay (řetězcová řada) a SCREEN\$. Stávající soubory ukládá zvolenou rychlosť, která může být stejná jako rychlosť pásku Spectra. Formát záhlaví je rozdílný od Spectra a nebude poznán u Spectra jako záhlaví. Formát bloku dat je tentýž jako u Spectra, ale může obsahovat vrch 512K. Soubory SCREEN\$ obvykle obsahují uspořádanou paletu (40 bajtů) a seznam řádků přerušení (1 až 509 bajtů) jako stejný obraz, zapíná paletu po částech dolů a obrazovka může být přehrána. Mód obrazu je použit vhodně před vložením bloku dat.

Soubory jsou nahrány automaticky ve správné rychlosti. Hlavičky CODE souboru Spectra jsou převedeny na ekvivalent SAMa a potom použity ke složení bloku dat. Soubory CODE Spectra mohou být vloženy do Coupé jako soubory CODE nebo SCREEN\$ specifikovaným tohoto po povolení LOAD. (Coupé rozeznává soubory SCREEN\$ od souborů CODE, protože mají být uloženy do obrazové paměti, která může být někde v paměťové mapě). Soubory CODE u Coupé mohou být uloženy urychlěně jako soubory SCREEN\$ jednoduchým použitím LOAD "jméno" CODE nebo SCREEN\$. Soubory CODE mohou mit provedenou adresu.

Změna systémové proměnné TPROMPTS (SVAR 50) může okamžitě uzavřít před SAVE a/nebo tisknout jména řádku jak jsou čteny z pásku.

Soubor formátu záhlaví

Pásek, disketa a síť používá společný formát záhlaví, pokud se týká BASIC popisovače. (DOS obsahuje informaci v částečně rozdílné formě pro vlastní použití, ale používá informace z a vyneschává informace ke standartním buferům záhlaví). SAVE používá jednoduchý bufer, HDR (žádané záhlaví) ve 4B00H kde detaili souboru k uložení jsou stavěny nahoře před záhlavím a potom jsou data uložena. LOAD také používá tento bufer, plus jeden zvláštní HDL (opatření záhlaví) ve 4B50H, který je použit ke vložení souboru záhlaví pro porovnání s HDR.

(Na Coupé programu "čitač záhlaví pásku" právě záleží na PEEK programu buferu HDL). (NB je zde soubor zvláštního typu v SAM-DOS - .SNP).

Formát záhlavi vyrovnávací paměti (HDR a HDL)

-
- 0 - (5) Typ 5 je Spectrum 48k snapshot (snímek), 16 je BASIC program, 17 je číselná řada, 18 je řetězec nebo řetězcová řada, 19 je soubor CODE a 20 je soubor SCREEN\$.
- 1 - 10 - (10) Jméno souboru, podloženo mezerymi
- 11 - 14 - (4) Dovoluje delší jméno souboru, jestliže DEVICE není T, např. "D2:filenamexx". SAMDOS ubere něco, protože maximální šířka jména souboru je pouze deset znaků.
- 15 - (1) Příznaky. Bit 0 je použit, pokud jméno není tisknuto na LOADING, bit 1 je použit, pokud CODE je chráněn. Příznaky mohou být vhodně použity během SAVE přes:
- SAVE CHR\$ 1 + "jméno" - používá bit 0
- SAVE CHR\$ 2 + "jméno" - používá bit 1
- SAVE CHR\$ 3 + "jméno" - používá oba byty

(Znaky <= 3 jako první znak ve jménu souboru používají právě když příznaky -nejsou uloženy jako část jména). Příznakový bit 0 pracuje v doplňku systémové proměnné TPROMPTS. Bit 1 chrání subory CODE, když je tento bit použit, automaticky běžící soubor nemůže být zastaven přes MERGE nebo přesměrován na jinou adresu opatřením npř. LOAD počátek, délka, nová adresa.

- 16 - 26 - (11) Jestliže typ je 17 nebo 18, obsahuje typ/délka bajtu a jméno řetězce
- 16 - (1) Pokud je typ 20, obsahuje MODE obrazovky
- 16 - 18 - (3) Pokud je typ 16, obsahuje délku programu vyjma proměnných
- 19 - 21 - (3) Pokud je typ 16, obsahuje délku programu plus číselné proměnné.
- 22 - 24 - (3) Pokud je typ 16, obsahuje délku programu plus číselné proměnné a délku mezery před řetězec/řada poměnných

(Zvlášť data pro programy v BASICu dovolují NVARS, NUMEND a SAVARS k použití na LOADING).

- 27 - 30 - (5) Rezervovaný
- 31 - (1) Stránka 16k v tomto souboru začíná (ignoruje LOADING, kromě souborů CODE). Bit 4-0 obsahuje stránku, byty 7-5 jsou předefinovány. Pokud je typu 7 nebo 18 a vložený řetězec neexistuje v paměti, tento bajt v HDR bude FFH.

- 32 - 33 - (2) Odkládá stránku začátku souboru (ignoruje LOAD vyjma souborů CODE). Odloženo je 8000H - BFFFH ve tvaru LSB/MSB.
- 34 - (1) Soubor stránek dlouhých 16K(v HDL nebo HDR během SAVE) nebo požadovaný začátek souborů CODE v HDR během LOAD (FFH, pokud zde nbyl požadovaný začátek). Během LOAD řetězce "délka souboru" v HDR je délka existujícího řetězce v paraméti nebo FFFFFFF, jestliže zde není řetězec.
- 35 - 36 - (2) Soubor délky MOD 16K, plus 8000H (obnova bitu 15 délky slova, abychom dostali správnou délku
- 37 - (1) Pokud soubor CODE stránka pováděně adresy nebo FF jestliže zde není. Pokud BASIC program, pak 00 jestliže je zde číslo řádku auto v dalších dvou bajtech nebo FF, pokud zde není.
- 38 - 39 - (2) Pokud soubor CODE, odkládá kromě stránky prováděné adresy (8000-BFFFH, LSB/MSB), pokud zde je. Pokud BASIC program, číslo řádku auto-run, pokud zde je.
- 40 - 79 - (40) Místo pro poznámky.Není specializováno k ničemu, ale program ukládá (SAVE) a vkládá (LOAD). Může být POKE nebo PEEK k získání zvláštních informací o souboru.
(DOS pouze ukládá (SAVE) prvních 8 bajtů v tomto prostoru).

SAM BASIC

Jména číslicových proměnných musí začínat písmenem. Čísla, jména, podtržení a mezery smí následovat první znak celku horních 32 znaků (mezery se nepočítají).(Proměnné FOR- NEXT jsou zpracovány shodně s normálními číslicovými proměnnými).

Jména řetězců a pole jsou podobné, ale limitované deseti znaky (mezery se nepočítají). Jména řetězce a řetězcového pole následují po "\$". Řetězce mohou státit na délku 65520 znaků. Pole mohou vyplňovat celou přístupnou paměť, ačkoliv nemůže být indeksováno přes 64K. Řádky programu mohou mít nejvýše 127 příkazů a může být dlouhý maximálně 16127 bajtů.

Klíčová slova jsou obvykle znázorněna úplně, ačkoliv klávesy mohou být programovány ke tvorění úplných slov. Když je klíčové slovo uznáno, je převedeno do seznamu hlavních písmen. Klíčové slovo, po kterém následuje písmeno nebude znázorněno, takže PRINT x předpokládá jméno procedury, ale print x a print i se stává PRINT x a PRINT 1.

Jestliže zmáčknete vlastní klávesu EDIT, řádek s ">" se objeví v editačním prostoru. Jestliže napišete číslo řádku a potom

stisknete EDIT, řádek, který si přejete se objeví v editačním prostoru. Kurzor může být posouván doleva, doprava, nahoru nebo dolů v editačním řádku použitím kurzorových kláves. Klávesa DELETE doleva; SHIFT + DELETE maže doprava.

Rozhraní BASIC - strojový kód

NB - hledisko hromadění ohrazením volitelných členů
SAVE "jméno" CODE začátek, délka < adresa provedení >
Soubor produkující auto-RUNing, pokud je použita adresa provedení
LOAD "jméno" CODE < začátek>< adresa provedení>
Vykoná soubor z adresy provedení, jestliže je použita. Jinak je
použita kterákoliv adresa provedení uložená se souborem. MERGE
zastaví prováděný soubor CODE, toto je chráněno připsáním bitu
v hlavičce souboru.
Počátek, délka a prováděcí hodnoty a adresy použité přes UR, USR\$
CALL, POKE, PEEK atd., mohou stačit na 528K minus 1.

USR n

Volá adresu n. Vstupuje s BC=n. Všechny registry mohou být použity. USR vrací konečnou hodnotu BC.

USR\$ n

Jako USR, ale vrací řetězec délky BC bajtů začínající v DE na straně A. (Strana A je irelevantní, jestliže DE < 8000H).

CALL n

Volá adresu n ze seznamu číslic nebo parametrů, pokud je žádán. Hodnoty čísel a délky a počátky řetězců jsou umístěny na zásobníku kalkulátoru pohyblivé řádové čárky s napsanou informací. Registr A obsahuje číslo parametrů, když adresa n je vložena. Použití zásobníku FC dovoluje číslům odstranit a zaokrouhlit 2 bajty s řadou ověření voláním JGETINT v seznamu skoků, čísla pohyblivé řádové čárky nebo řetězců. Nemusí být srovnány použitím JSTKFETCH. Řetězce kratší než 256 bajtů mohou být kopirovány do buferu (vyrovňávací paměti) v ploše systémových proměnných voláním JSBUFFET. (Řetězce mohou být kopirovány odkudkoliv v paměti, stejně, pokud jsou stránkovány mimo). JSTK STORE může být použito k umístění každého výsledku na zásobníku kalkulátoru.

CLEAR n - Pokládá nižší hranici použitím RAM BASICu

OPEN n - Rezervuje n krát více 16K stránek k použití pro BASIC

OPEN TO n - Přiděluje celkově n stránky pro BASIC
CLOSE N - Volných n 16K stránek

LENGTH - délka

Tato funkce vraci adresu a dimenzuje pole a řetězce a čísla adresy. Užitečná pro přístupy dat přes CALL.

POKE n,x	POKE 1 bajt
POKE n,1,2,3,.....	POKE mnohonásobené bajty (nejvýše 32)
DPOKE n,x	POKE 2 bajty (LSB/MSB)
POKE n,a\$	POKE úplný řetězec

PEEK n	PEEK 1 bajt
DPEEK n	PEEK 2 bajty
MEM\$ (n TO m)	PEEK řetězec, např.

LET a\$ = MEM\$ (70000 TO 71000) přiděluje a\$ 1000 bajtů
POKE 72000,a\$ pokládá tyto bajty na 72000.

MEM\$ může číst jakoukoliv stránku RAM (16K - 528K), ale nebude číst z ROMky. Použití PEEK nebo DPEEK, pokud nepotřebujete číst ROM o.

Příklady bázové proměny.

BIN 1010 = 10	HEX\$ 74 = "4A"	&FF = 255
BIN\$ 128 = "00000000"	HEX\$ 16384="4000"	&1000003 = 65539
	Hex\$ 65539="0100003"	
VAL ("+a\$)= 255 jestliže a\$ = "ff"		

Hlavní ukazatelé paměťové mapy oblasti BASICu

(všechni berou 3 bajty, ve formě: strana/přidání)

PROG	začátek programu
NVARS	začátek číselových proměnných
NUMEND	konec číselových proměnných mezera (<= 512 bajtů)
SAVARS	začátek řetězce pole proměnných
ELINE	začátek editační řádky
WORKSP	začátek pracovní oblasti zbývající prostor
RAMTOP	poslední bajt přidělen programu v BASICu

Pro úplnou informaci se podívejte do části systémové proměnné.

Přehled rozdělení stránky

Přehled 5100H - 5120H je velmi důležitý; obsahuje bajt pro každou možnou 16K RAM stránku na Coupé, plus omezovač bajtu FFH. První bajt odpovídá první stránce 16K RAM, druhý druhé stránce atd. Neexistující stránky jsou také označeny přes FFH. Nepoužité stránky jsou označeny jako 00H. Stránky použity na první BASIC Program v přístroji jsou označeny 40H, obrazy stránek jsou značeny COH a stránky použity přes DOS (obvykle jen jedna) jsou označeny 60H.

Na Coupé 256K bude přehled (ALLOC) vypadat stejně, po DOS má vloženo:

40 40 40 40 00 00 00 00 00 00 00 00 00 00 60 CO CO FF FF FF atd.

"Obslužná" stránka je označena 20H. To je rozdělováno do 16-ti 1K sekcí, které mohou být použity roztríďením krátkých obslužných programů. Posledních 16 bajtů v obslužné stránce (SLOTT) ukazuje, které "stopy" jsou rezervovány - 0 ukazuje odpovídající volnou stopu FFH, že je rezervována. (Poslední stopa je pouze 16 bajtů z 1K).

Vlastní metoda rozdělení mezer pro krátký program je podívat se zpětně skrze ALLOC do 20H. Jestliže to naleznete, vypněte ukázanou stránku a podívejte se zpětně skrze SLOTT na ušetřenou stopu. Označte to a použijte tuto stopu. (Nebo stopy - můžete rezervovat několik stop). Jestliže nenaleznete v ALLOC vstup 20H podívejte se na 00H; oznamí chybu, jestliže žádný není nalezen. Jiný, to značí 20H, vyčistí posledních 16 bajtů na této straně na nulu a potom rezervuje totéž svoje místo v nové SLOTT kterou máte právě vytvořenou.

Volné stránky mohou být použity jako dočasně pracovní oblasti, opatřeny jistotou, že stránka nepůjde přepsat. (Přerušení nedělá ale DOS můžete).

Klávesnice

Jakákoliv klávesnice může postavení klíče NORMAL, CAPS SHIFTED, SYM SHIFTED nebo CONTROL volit v kódu (0 - 255). 69 kláves a 4 změny postavení dávají 276 klíčů v "mapě" klávesnice. Tato mapa má zdánlivě náhodné seřazení, avšak odpovídá klávesnicové mozaice - viz doplněk C diagramů klávesnice a mapa poloh plus počáteční kódy vráceny ke každé pozici. Např. mapa polohy 0 má 98 jako CHR\$ 98 ("b") je provedeno, když skutečná klávesa je stisknutá, pokud odpovídá poloze v mapě bez jakýchkoliv změn. Úplná mapa může být změněna, pokud si přejete.

KEY Posn, x

KEY 23,134 - Klávesa má pozici 23 v mapě klávesy způsobi provedení CHR\$ 134, když je stlačená.

KEY 0,65 - způsobi normání "b" d CHR\$ 65 ("A")

Posn - musí být 0-255, x musí být 0-255

Když určité kódy přicházejí z klávesnice, hledání je vidět, pokud rozšířené omezení má být pro ně uděláno.

Kódy 192-254 mohou být definovány touto cestou. Pokud omezení neexistuje, CHR\$ 192-254 bude použito, jinak je použito rozšířené omezení. To dovoluje funkční klávesy definovat jako RUN (vstupy) nebo LIST (vstupy). Viz níže.

DEF KEYCODE

DEF KEYCODE 195: PRINT 123: PINT "z" způsobi jakýkoliv následující kód od 195 přicházející z klávesnice k rozšíření: PRINT 123: PRINT "z" (ENTER). To provede: 123 z se objeví na hořejší ploše obrazovky. Úplný řádek po DEF KEYCODE 195: je použit jestliže na řádku je dvojtečka automaticky (RETURN) je potlačena, např.

DEF KEYCODE 193: PRINT "asd":
 provede
PRINT "asd" v editační ploše

DEF KEYCODE může být také použit s řetězcem, spíše než zbytek řádku, například:

DEF KEYCODE 192, "TESTING" způsobi jakýkoliv následující kód po 192 přicházející z klávesnice k rozšíření "TESTING" (RETURN). (Může způsobit chybné hlášení).

DEF KEYCODE 192, "TESTING:" může potlačit (RETURN), protože poslední znak je dvojtečka.

Podrobná definice může být ozřejměna:

DEF KEYCODE 195: (nenásleduje nic)

nebo DEF KEYCODE 195, "

DEF KEYCODE ERASE odstraňuje všechna omezení.

"Příliš mnoho omezení" je oznámeno, jestliže prostor běží mimo plochu paměti použitím DEF KEYCODE.

Toky a kanály

CLOSE

OPEN

Stejně jako Spectrum, toky 0 a 1 jsou obvykle OPEN (otevřeny) kanálem "K"; PRINT #0; "test": PAUSE bude tisknout na dolním okraji obrazovky jako obvykle (použitím kanálu "S") a PRINT # 3 pracuje stejně jako LPRINT (použitím kanálu "P"). Můžete také stejně udělat použitím OPEN #6; "P": PRINT #6; "testing" k odeslání dat tiskárně přes tok 6 nebo OPEN #2, "P" tak, že normální obrazovka jde mimo tiskárnu. INPUT obvykle používá kanál "K" (nižší obrazovka), ale INPUT #2 dovoluje a používá vyšší obrazovku (kanál "S").

Např: INPUT #2; "Value:";v - Vám dovolí psát v hodnotě v na vyšší obrazovku, další "Value:" okamžitě. Rádek není čistý po stisknutí vstupu, na rozdíl od nižší obrazovky.

Povel RECORD pracuje přes tok 16 (který neexistuje na Spectrum). Tok 16 tiskne řetězcovou proměnnou specifikovanou přes RECORD a bude stále prováděna do povolu RECORD STOP. To může být užitečné některým uživatelům: např. RECORD TO a\$: DIR #16:RECORD STOP bude umístěn seznam disket v a\$. Jiné toky než 16 mohou být otevřeny použití řetězcovým proměnným, např. OPEN #5 ;"\$".

OPEN #s; "b" (nebo "B") dovolí toku S odeslat něco k tiskárně, to je užitečné pro odeslání řídících kódů. (Problém s proudem "p" je ten, že řídící kódy jsou uskutečněny a nemohou být odeslány k tiskárně - např. řídící kód TAB odešle počet mezír k tiskárně). Kanál "p" skutečně používá kanál "b" pro vstup, potom má upravena data pro jejich přijetí. Proto všichni pokročili uživatelé to potřebují k provedení, řekněme, řadiče sériové tiskárny ke změně kanálu "b" výstupní adresy.

SAVE, LOAD, VERIFY

Dovoluje programům, řetězcům, polím, strojovému kódu a obrazovkám ovládat obrazy.

SAVE "jméno" CODE začátek,délka, prováděcí adresa dovoluje totéž

LOAD "jméno" CODE bude začínat z dané adresy. MERGE "jméno" CODE bude potlačovat samospouštění kódovaných souborů.

LOAD "jméno" DATA b\$() pracuje, ale také DATA addr\$() nebo DATA abc\$ jsou dovoleny. Uvědomte si, že SAVE "jméno" DATA blit\$ je O.K. Dokonce jestliže blit\$ je jednoduchý řetězec, tak jako řetězec BLITZ.

LOAD "jméno" LINE n vkládá program a jde na řádku čísla n. Přehlídí jakékoliv číslo samospouštěcího řádku uloženého s programem.

SAVE "jméno" SCREEN\$ ukládá obraz z obrazovky a běžný mód a informace palety pro pozdější použití. (LOAD automaticky zapíná módy, jestliže jsou požadovány).

DEVICE

Tento povel vykonává SAVE, LOAD, MERGE a VERIFY, pracuje s disketou nebo sítí.

DEVICE d určuje použití disk drajvu 1

DEVICE d1 - " -

DEVICE D1 - " -

DEVICE N5 používá síť postavení 5

DEVICE T používá pásek, SAVE rychlostí podobnou Spectru

DEVICE T35 používá pásek, SAVE rychlostí 35. (Rychlosť 112 je rychlosť Spectra, 35 je mnohem rychlejsí. Vyšší rychlosťi jsou méně spolehlivé-ale trojnásobná rychlosť Spectra může být proveditelná mnoha magnetofony. LOAD automaticky přizpůsobuje rychlosť pásku rychlosťi, která byla použitá, když byl soubor ukládán na pásek

Zvuk

SOUND r,d
SOUND r,d;r,d;r,d....

Odesílá data bajtu D ke zvukovému čipu registru R.
Může být použito horních 127 dvojic čísel.
Toto vysvětlení je základem programu SAM Coupé User's Guide.
Program je zde reprodukován pro snadnější pochopení.

10 REM zvukové efekty	1030 LET L=0: LET R=16:
20 GOSUB 3000	GOSUB 1500
30 REM LOKOMOTIVA	1040 LET L=-1: LT R=0:
40 LET L=7: GOSUB 2000	GOSUB 1500
50 LET p=20: GOSUB 1000	1050 LET L=0: LET R=-16:
100 REM flying saucer	GOSUB 500
110 LET L=3: GOSUB 2000	1060 RETURN
120 LET p=20:GOSUB 1000	1500 REM output subrutina
200 REM sonic scooter	1510 SOUND 2,(a+b): PAUSE P
210 SOUND 17,4: SOUND 16,64	1520 FOR n=1 TO 15
220 LET p=5: GOSUB 1000:	1530 LET a=a+L: LET b=b+R
PAUSE 200	1540 SOUND 2,(a+b)
300 REM telefon	1550 pause P: next N
310 GOSUB 3000	1560 RETURN
320 FOR n= 0 TO 3	2000 REM output sound data
340 SOUND 2,255: PAUSE 18	2010 DATA 9,64,16,16,17,0,24,
350 SOUND 2,0: PAUSE 8	138,28,1,21,4,20,0
360 SOUND 2,255: PAUSE 18	2020 DATA 9,255,17,1,20,4
370 SOUND 2,0: PAUSE 80	2030 DATA 24,142,20,4,10,111,
380 NEXT n	17,6,9,128,16,32,28,1
400 REM cuckoo	2040 FOR N=1 TO L
410 BEEP 0.2,19: BEEP 0,4,15	2050 READ a,d: SOUND a,d:
420 STOP	NEXT n
1000 REM stereo zleva doprava	2060 RETURN
1010 LET a=0: LET b=0	3000 REM clear sound
1020 LET L=1: LET R=0:	3010 FOR N=0 TO 31: SOUND n,0:
GOSUB 1500	NEXT n
	3020 RETURN

Řádek 20 čistí registry zvukového čipu nastavením všech na nulu. Řádek 40. L poukazuje na dvojice dat na řádce 2010. Dvojice dat odkazují na nastavení různých registrů.

Řádek 50. P poukazuje na rychlosť, ve ktorej je zvuk posunovaný přes stereo pole. Pokud hrajete zvukový DEMO program, uvidíte, že se zvukový běžec posunuje zleva doprava rychleji než pára lokomotivy, nebo létající taliř. Podprogram na řádce 1000 říká

počítači, kde umístit zvuk přes stereo pole. Podprogram na řádku 1500 vypouští zvuk z obou (levého a pravého) kanálů. Létající taliř a zvukový běžec jsou podobné páře lokomotivy vyjma toho, že pára lokomotivy používá bílý zvuk.

Zvukový běžec mění běžnou oktávu tak, že registr oktávy 17 je 4 a registr oktávy 16 je dán 0 pro generátor 8 a 4 pro generátor 9. Věsiměte si také, že rychlosť pohybu běžce je vyšší než lokomotivy nebo létajícího taliře.

Telefon je umístěn ve středu stereo pole po opětovném vyčíslení zvukového čipu. Tento zvuk používá obálku registru 24 a umisťuje zvuk z tónového registru 10, použitím mixeru 20 a s použitím bitu 4. Taky používá tónový registr 9 ke kontrole rychlosti z obálky. Konečně registr 28 umožňuje zvuk. "BEEP" zvuku je řízen přepínačem hlasitosti amplitudového registru 2 mezi 0 a 255. To se opakuje 4 krát s přestávkou mezi každým zvukem.

Kukačka používá povelu BEEP. Tento povel, podobně jako na SPECTRU, nepoužívá vůbec zvukový čip, ale ASCII ke generování zvuku.

Podprogram na řádce 1000 vyvolává řídící data pro výstup podprogramu na řádce 1500. Proměnná R a L určuje směr průchodu, když "posunuje" hlasitost nahoru nebo dolu na jednom ze dvou kanálů, podprogram na řádce 1500 dělá jeho posun. Registr 2 je amplitudový registr, který ovládá zvuk. To je také zde, na řádce 1550, kde P je použito k řízení rychlosti pohybu přes stereopole.

Podprogram na řádce 2000 obsahuje registr dat a aktuálně působi na data presentovaná čipem. Tato funkce je pro výstupy jakékoli informace k registrům proměnných, stejně jako přijímá data.

Podprogram na řádce 3000 čistí zvukový čip postavením všech registrů na nulu.

Změna měřítka grafického systému

XOS, XRG
YOS, YRG

Ovládá změnu grafického měřítka a počátek. XOS a YOS udávají zvětšení koordinátů X,Y, kdy jsou použity povely pro grafiku. Dovolují posunovat začátek z normální pozice přes editační plochu doleva. RUN nebo CLEAR obnovuje XOS a YOS na nulu; LET je mění. XRG a YRG mění měřítko grafického systému. XRG je obvykle 256 nebo 512, pokud se používají "tenké" obrazové prvky. YRG je obvykle 192. Přeměnu XRG na, řekněme 1024, uděláte použitím PLOT 950, 100. První povel předělává 256/1024*950 před znázorněním pixelu na obrazovce. (Mění rozsahy kapacity 2 daných rychlejších znázornění)

Obvykle má módy 2 a 4 a mód 3, jestliže jsou použity "tlusté" pixely (obrazové prvky), grafický souřadnicový systém s 0,0 na nejnižší editační části vlevo a 255,173 nahoře vpravo.

Jestliže jsou použity "tenké" obrazové prvky, obvykle X je rozšířeno tak, že vpravo nahoře je 511,173. V módu 1 je znak vysoký 8 obrazových prvků, spíše než 9 a 2 menší prohlížeči jsou použity přes editační plochu tak, že vrch obrazovky má souřadnici Y 175.

Nejnižší dva řádky zobrazení mohou být zobrazeny použitím:

PLOT 0, -10

Jestliže plocha počátku je posunuta použitím LET XOS = -18, potom 0, 0 je velmi nízko vlevo na obrazovce a 0, 191 je nahoře vlevo.

Kódy klíčových slov

Funkce Coupé jsou nahromaděny v programu jako FFH (255) následováno po 3BH - 83H

Funkce	Dec	Hex	Funkce	Dec	Hex	Funkce	Dec	Hex
PI	59	3B	COS	84	54	VAL\$	109	6D
RND	60	3C	TAN	85	55	VAL	110	6E
POINT	61	3D	ASN	86	56	TRUNC\$	111	6F
FREE	62	3E	ACS	87	57	CHR\$	112	70
LENGTH	63	3F	ATN	88	58	STR\$	113	71
ITEM	64	40	LN	89	59	BIN\$	114	72
ATTR	6	41	EXP	90	5A	HEX\$	115	73
FN	66	42	ABS	91	5B	USR\$	116	74
BIN	67	43	SGN	92	5C	RESERVD	117	75
XMOUSE	68	44	SQR	93	5D	NOT	118	76
YMOUSE	69	45	INT	94	5E	RESERVED	119	77
XOPEN	70	46	USR	95	5F	RESERVED	120	78
YOPEN	71	47	IN	96	60	RESERVED	121	79
RAMTOP	72	48	PEEK	97	61	MOD	122	7A
RSERVED	73	49	DPEEK	98	62	DIV	123	7B
INSTR	7	4A	DVAR	99	63	BOR	124	7C
INKEY\$	75	4B	SVAR	100	64	RSERVED	125	7D
SCREEN\$	76	4C	BUTTON	101	65	BAND	126	7E
MEM\$	77	4D	EOF	102	66	OR	127	7F
RESERVED	78	4E	PT	103	67	AND	128	80
PATH\$	79	4F	RESERVED	104	68	<>	129	81
STRING\$	80	50	UDG	105	69	<=	130	82
RESERVED	81	51	RESERVED	106	6A	>=	131	83
RESERVED	82	52	LEN	107	6B	RESERVED	132	84
SIN	83	53	CODE	108	6C			

Kódy 85H až FEH nepotřebují předcházející FFH; jsou to jednoduché bajty zhuštěných forem klíčových slov. Souvislost jim dovoluje ukázat jak klíčová slova, tak znaky UDG, jak je upřesněno. Například budou ukazovat klíčová slova při listování, jestliže to podstata dovoluje, ale bude ukazovat UDG znaky, když použijete PRINT. Známky 85H až 8FH jsou bližše určeny-nemají výsledek nebo působení na jejich vlastní, ale jednoduchá modifikace působení jiných povelů. (To nebrání působení rutiny určené k nim) Např. DOS poskytuje působení rutiny na WRITE.

Známka	Dec	Hex	Známka	Dec	Hex	Známka	Dec	Hex
USING	133	85	OFF	137	89	THEN	141	8D
WRITE	134	86	WHILE	138	8A	TO	142	8E
AT	35	87	UNTIL	139	8B	STEP	143	8F
TAB	136	88	LINET	140	8C			

Známky 90H jsou povely. Seznam adres povelů (viz systémové proměnné - CMDADDR) obsahuje adresu 2 bajtu pro každou z nich začínající DIR. To je adresa, která ovládá kontrolní syntax a působení doby výpočtu pro povely.

Povel	Dec	Hex	Povel	Dec	Hex	Povel	Dec	Hex
DIR	144	90	REM	183	B7	ON	222	DE
FORMAT	145	91	READ	184	B8	GET	223	DF
ERASE	146	92	DATA	185	B9	OUT	224	E0
MOVE	147	93	RESTORE	186	BA	POKE	225	E1
SAVE	148	94	PRINT	187	BB	DPOKE	226	E2
LOAD	149	95	LPRINT	188	BC	RENAME	227	E3
MERGE	150	96	LIST	189	BD	CALL	228	E4
VERIFY	151	97	LLIST	190	BE	ROLL	229	E5
OPEN	152	98	DUMP	191	BF	SCROLL	30	E6
CLOSE	153	99	FOR	192	CO	SCREEN	231	E7
CIRCLE	154	9A	NEXT	193	C1	DISPLAY	232	E8
PLOT	155	9B	PAUSE	194	C2	BOOT	233	E9
LET	156	9C	DRAW	195	C3	LABEL	234	EA
BLITZ	157	9D	DEFAULT	196	C4	FILL	235	EB
BORDER	158	9E	DIM	197	C5	WINDOW	236	EC
CLS	159	9F	INPUT	198	C6	AUTO	237	ED
PALETTE	160	A0	RANDOMIZE	199	C7	POP	238	EE
PIEN	161	A1	DEF FN	200	C8	RECORD	239	EF
PAPER	162	A2	DEF KEYCODE	201	C9	DEVICE	240	FO
FLASH	163	A3	DEF PROC	202	CA	PROTECT	241	F1
BRIGHT	164	A4	END PROC	203	CB	HIDE	242	F2
IMVERSE	165	A5	RENUM	204	CC	ZAP	243	F3
OVER	166	A6	DELETE	205	CD	POW	244	F4

Povel	Dec	Hex	Povel	Dec	Hex	Povel	Dec	Hex
FATPIX	167	A7	REF	206	CE	BOOM	245	F5
C SIZE	168	A8	COPY	207	CF	ZOOM	246	F6
BLOCKS	169	A9	RESERVED	208	D0	RESERVED	247	F7
MODE	170	AA	KYN	209	D1	RESERVED	248	F8
GRAB	171	AB	LOCAL	210	D2	RESERVED	249	F9
PUT	72	AC	LOOP IF	211	D3	RESERVED	250	FA
BEEP	173	AD	DO	212	D4	RESERVED	251	FB
SOUND	174	AE	LOOP	213	D5	RESERVED	252	FC
NEW	175	AF	EXIT IF	214	D6	RESERVED	253	FD
RUN	176	BO	long IF	215	D7	RESERVED	254	FE
STOP	177	B1	short IF	216	D8	Not usable	255	FF
CONTINUE	178	B2	long ELSE	217	D9	(function prefix)		
CLEAR	179	B3	short ELSE	218	DA	Neschopen použití		
GOTO	180	B4	END IF	219	DB			
GO SUB	181	B5	KEY	220	DC	long = dlouhé		
RETURN	182	B6	ON ERROR	221	DD	short = krátké		

Určitá klíčová slova budou v diskové základně rozšířeného BASICu a tato slova nebudou použita v BASIC programech jako procedura nebo jména proměnných nebo následující úprava programu bude obtížná (ačkoliv ALTER je vyhledávání a nahrazování, takto můžete měnit každý použitý povel, např. procedura SORT na SORTER která bude O.K.). Prozatímní seznam nových klíčových slov následuje:

SORT, ALTER, USING\$, SHIFT\$, INARRAY, NUMBER, CHAR\$, JOIN

Systémové proměnné

Pozn. Funkce SVAR N dává výsledek 5A00H + N.

LNCUR	5A00H	Znak kurzoru pro běžný řádek (obvykle ">")
LCUR	5A01H	Znak kurzoru, když je vypnutý CAPS LOCK (obvykle CHR\$ 128);
UCUR	5A02H	Znak kurzoru, když je zapnutý CAPS LOCK (obvykle CHR\$ 129)
BIN1DIG	5A03H	Znak použitý přes BIN\$ jako "1" (obvykle "1")
BIN0DIG	5A04H	Znak použitý přes BIN\$ jako "0" (obvykle "0")
INSTHASH	5A05H	Znak použitý přes INST jako znak # (obvykle CHR\$ 35 - "#")
SLDEV	5A06H	Běžné písmeno (obvykle "T" pro systém pásku, ale může být "D" nebo "N")
SLNUM	5A07H	Běžná rychlosť ukládání na pásek nebo opomerení čísla drajvu, když je disk dray používán.

SPEEDINK	5A08H	Doba mezi změnami střídání se PEN, na 50 za sec.
LINIPTR	5A09H	(2) Běžná poloha v řádku přerušení palety seznamu změn.
XCMDP	5A0BH	(3) Strana/přeložení seznamu prvního vnějšího povetu nebo FFxxxx
PRRHS	5A0EH	Tiskárna hranice pravého sloupce. (Obvykle 79 pro 80 sloupců; Použitím 255 dostaneme neohraničený počet).
AFTERCR	5A0FH	Znakový kód odeslán k tiskárně pro CHR\$ 13, pokud je použit kanál "p". (Obvykle 10, dosťaváme případ automatické řádky. Používá nulové hodnoty, jestliže znak nebyl odeslán).
LPTPRT1	5A10H	(2) Ridici port tiskárny/0H hodnoty strobe. Port je obvykle 233, s portem 232 (1 nižší) užívaným pro data. Druhý bajt by měl být vždy 1.

Následující systémové proměnné jsou použity vždy, když obslužný výpis obrazovky má být uložen:

DMPLEN	5A12H	Délka grafického výpisu (v jednotkách 8 pixelů, obvykle 22)
DMPWID	5A13H	Šířku grafického výpisu (v jednotkách "tlustých" 8 pixelů, obvykle 32)
DMPWM	5A14H	Násobitel šířky grafického výpisu (1 pro obvyklou šířku, 2 nebo 3 pro dvojitou nebo trojnásobnou šířku)
DMPHM	5A15H	Násobitel výšky grafického výpisu (1 pro obvyklou výšku, jinak než 1 pro dvojnásobnou výšku)
GCM1	5A16H	Počáteční zpráva odeslání tiskárně před výpisem. První bajt je číslo znaků k odeslání. Obvyklé hodnoty jsou 8, 27, 108, 8, 27, 51, 24, 0, 0. (2 bajty jsou ušetřeny)
GCM2	5A1FH	Zpráva odeslána k tiskárně je vypsána před každým řádkem. První bajt je číslo znaků k odeslání. Obvyklé hodnoty jsou 5, 13, 10, 27, 42, 4, 0, 0 (2 bajty jsou ušetřeny)
GCM3	5A27H	Konečná zpráva odeslání k tiskárně po výpisu. První bajt je číslo znaků k odeslání. Obvyklé hodnoty jsou 4, 3, 10, 27, 64, 0 (1 bajt je ušetřen).
DMPTL	5A2DH	(2) Adresa pro grafický výpis na obrazovce v levém horním rohu. (Obvykle 8000H)

Konec systémových proměnných použitím obslužného výpisu.

TABVAR	5A2FH	Nula pro tisk čárky tabulky o 16 sloupcích (obvykle) nebo jakékoli jiné pro 8-sloupcovou tabulkou.
M23LSC	5A30H	(2) Mód 3/4 nižší obrazovka barev. První bajt je PAPER*16+PAPER, druhý je PEN*16+PEN, oba pokusy jsou v každém bajtu stejné.
SOFE	5A32H	Příznak pro obrazovku, vypnutí umožňuje/zne- možňuje. Pokud je nula (obvykle) obrazovka bude čistá v modech 3-4, když obrazovka nemá být použita po dobu 22 minut.
TPROMPTS	5A33H	Bit 0=1 k potlačení tisku jmen řádků během vkládání z pásu; bit 1=1 k potlačení působení během vkládání na pásek.

Je to počátek bloku proměnných uložených s obrazovkou když nejsou delší než běžná obrazovka.

BGFLG	5A34H	Blok grafických příznaků. Nula, jestliže je BLOCKS 1 a 1, jestliže je BLOCKS 0
FL60R8	5A35H	Nula, jestliže v MODE 3 je použito šířky znaků 6 pixelů, jinak jiné než nula.
CSIZEH	5A36H	Výška znaku použitím povelu CSIZE
CSIZEW	5A37H	Šířka znaku použitím povelu CSIZE
UWRHS	5A38H	Vyšší okno pravého sloupce. Počátek na 31.
UWLHS	5A39H	Vyšší okno levého sloupce. Počátek na 0.
UWTOP	5A3AH	Vyšší okno horního řádku. Počátek na 0.
UWBOT	5A3BH	Vyšší okno spodního řádku. Počátek na 18.
LWRHS	5A3CH	Nížší okno pravého sloupce. Počátek na 31.
LWLHS	5A3DH	Nížší okno levého sloupce. Počátek na 0.
LWTOP	5A3EH	Nížší okno horního řádku. Počátek na 19.
LWBOT	5A3FH	Nížší okno spodního řádku. Počátek na 20.
MODE	5A40H	MODE běžné obrazovky. 0-3 po módy 1-4
YCOORD	5A41H	Běžná grafická poloha souřadnice y s 0 v nejvyšší části obrazovky a 191 ve spodní.
XCOORD	5A42H	(2) Běžná grafická poloha souřadnice x s 0 vlevo. Rozsah je 0-255, jestliže jsou použity "tenké" pixely (v tomto případě je použit druhý bajt a rozsah je 0-511)

Následují "stálé" grafické/tiskové proměnné, použitím např; PEN 2 PAPER 4, OVER 1.

THFATP	5A44H	Nula, jestliže jsou použity "tenké" pixely, jiné než nula pro "tlusté" pixely.
ATTRP	5A45H	Atributy použity v módech 1 a 2
MASKP	5A46H	Maska použitá v módech 1 a 2. Bity, které odpovídají bitu atributu, vzaty z obrazovky, ne ATTRP

PFLAGP	5A47H	Bit 4 je dán pro papír 9, bit 6 pro pero 9.
M23PAPP	5A48H	Mód 3/4 PAPER. Ubirá (mód 4) nebo zdvojuje bity (mód 3), jestliže je požadován pruhovaný papír.
M23INKP	5A49H	Mód 3/4 PEN. Ubirá (mód 4) nebo zdvojuje bity (mód 3), jestliže je požadován pruhovaný papír.
OVERP	5A4AH	Postavení OVER pro tisk. 0 pro OVER 0, 1 pro OVER 1 (XOR).
INVERP	5A4BH	0 pro INVERSE 0, 255 pro INVERSE 1
GOVERP	5A4CH	Postavení OVER pro grafiku. 0-3 pro normální XOR, OR, AND.

Dočasná grafika/tisk proměnných např: PRINT PEN 3; PAPER 5;
To jsou dočasné verze nalistovaných proměnných.

THFATT	5A4DH	Kopirován z thfatp, když je mód=2, jinak nedává nulu (fat)
ATTRT	5A4EH	
MASKT	5A4FH	
PFLAGT	5A50H	
M23PART	5A51H	
M23INKT	5A52H	
OVERT	5A53H	
INVERT	5A54H	
GOVERT	5A55H	

Běžné okno proměnných, kopirován z UWRHS atd. nebo LWRHS atd souhlasný s oknem, které je použito.

WINDRHS	5A56H
WINDLHS	5A57H
WINDTOP	5A58H
WINDBOT	5A59H

Další tři proměnné jsou 5AB1H, 5B70H a 5B71H v ROMce verze 1.0.

WINDMAX	5A5AH	(2) Vyšší okno nejnižšího spodního řádku/máximálně pravý sloupec. Použití hranice povelom WINDOW. Obnovení povelom MÓDE.
ORGOFF	5A5CH	Grafický počátek v pixelech, udán ze spodní části obrazovky.
LSOFF	5A5DH	Velikost mezery mezi vyšším a nižším oknem v prohlížených řádcích MOD CSIZEH.
RESERVED	5A5EH	(14)
SPOSNU	5A6CH	(2) Vyšší poloha okna sloupec/řádek
SPOSNL	5A6EH	(2) Nižší poloha okna sloupec/řádek

(Konec vkládaného bloku z předvolené obrazovky)

PRPOSN	5A70H	Běžný sloupec tiskárny
RESERVED	5A71H	
OPCHAR	5A72H	Použití LPRINT - znak je tištěn
DEVICE	5A73H	0 = vyšší okno, 1 = nižší okno, 2 = tiskárna 3 = jiné
CLET	5A74H	Běžný sloupec písmen K/S/P/B/T/\$ atd.
IFTYPE	5A75H	Dlouhé / krátké postavení IF
REFFLG	5A76H	Nula, pokud proměnná REF pracuje
CURDISP	5A77H	Běžné zobrazení dánou povelom DISPLAY
CUSCRNP	5A78H	Běžná stránka obrazovky. Bit 7=0, bity 6 a 5 = MODE (0-3) a bity 4-0 = čísla stránky. Dánou použitím povelu SCREEN
CURF	5A79H	Běžný vyšší port RAM. (Dočasné uložení během stránkování)
CLRP	5A7AH	Běžný nižší port RAM. (Dočasné uložení během stránkování)
CSA	5A7BH	(2) Běžný údaj adresy. Použitím DDS.
FIRST	5A7DH	(2)
LAST	5A7FH	(2) Čísla řádků, např. LIST 20 TO 100.

Následují proměnné hlavních částí rozlohy paměti BASICu. Jsou seřazeny, když je udělána nebo obnovena mezera, pak je potřeba. V každé sestavě tří bajtů je první bajt číslo stránky RAM (0-31) a další dva bajty jsou doplnky stránky plus 8000H. (V některých případech doplněk dovoluje být větší než 16K, ale vždy nižší než 32K).

SAVARS	5A81H	
SAVARS	5A82H	(2) Řetězec a pole počátku proměnných
NUMENDP	5A84H	
NUMEND	5A85H	(2) Konec číselných proměnných
NVARSP	5A87H	
NVARS	5A88H	(2) Počátek číselných proměnných
DATADD'	5A8AH	
DATADD	5A8BH	(2) Adresa dat použitím povelu READ
WKENDP	5A8DH	
WKEND	5A8EH	(2) Konec pracovní oblasti. (poslední použitý bajtu před RAMTOP)
WORKSPP	5A90H	
WORKSP	5A91H	(2) Počátek pracovní oblasti
ELINEP	5A93H	
ELINE	5A94H	(2) Počátek editačního řádku
CHADP	5A96H	
CHAD	5A97H	(2) Běžná adresa znaku

KCURP	5A99H	
KCUR	5A9AH	(2) Adresa kurzoru v editačním řádku
NXTLINEP	5A9CH	
NXTLINE	5A9DH	(2) Adresa dalšího řádku v BASIC programu
PROGP	5A9FH	
PROG	5AA0H	(2) Počátek programu (adresa čísla řádku první řádky)
XPTRP	5AA2H	
XPTR	5AA3H	(2) Adresa chybné syntaxe v editační řádce
DESTP	5AA5H	
DEST	5AA6H	(2) Použití v přidělování proměnné
PRPTRP	5AA8H	
PRPTR	5AA9H	(2) Adresa běžné volané procedury
DPPTRP	5AABH	
DPPTR	5AACH	(2) Adresa běžného údaje DEF PROC
CLAPG	5AAEH	
CLA	5AAFH	(2) Adresa počátku běžné řádky
RESERVED	5AB1H	
STRNO	5AB2H	Běžné číslo proudu (streamu). Použití přes DOS
LDCD	5AB3H	Využít když je vkládán strojový kód Spectra (stránky 16K)
RESERVED	5AB4H	
OPSTORE	5AB5H	Dočasné uložení pro tisk výstupních adres
DMPFLG	5AB7H	Pokud není nula, tištěné znaky jsou "hozeny prýč"
LISTFLG	5AB8H	0/1/2 pro LIST FORMAT 0/1/2
LSTFT	5AB9H	Dočasná verze LISTFLG, použitým přes kanál "R"
INQUFG	5ABAH	Příznak "v úvozovkách". Bit 0=1, jestliže tištěný znak je uvnitř úvozovek. LIST nuluje tentot bit, tedy počáteční stav je "mimo" a známky jsou tištěny, mimo vnitřní obrazovky. PRINT dává 1, tedy UDG jsou tištěny mimo známky.
SPROMPT	5ABBH	Pokud není nula, nejsou dány náznaky "scroll?"

Příštích šest proměnných je použito odsazením listované rutiny.

OLDSPCS	5ABCH	Odsazení postavení předešlé stránky
INDOPFG	5ABDH	Příznak
NXTSPCS	5ABEH	
CURSPCS	5ABFH	
NXTHSPCS	5AC0H	
CURTHSPCS	5AC1H	
KPOS	5AC2H	(2) Poloha editačního kurzoru na obrazovce

SOFFCT	5AC4H	Cítač používán vypnutím auto-screenu. Vyměňuje každých 256 snímků, pokud není klávesnice použita
SOLF	5AC5H	Příznak pro "obrazovka má být vypnuta" (pokud není nula).
SPEEDIC	5AC6H	Čítač pro blýskající inkousty
PALFLAG	5AC7H	Bit 0 ukazuje, který seznam palety je použit hlavní nebo vedlejší.

Dočasná uložení:

TEMPW1	5AC8H	(2)
TEMPW2	5ACAH	(2)
TEMPW3	5ACCH	(2)
TEMPPB1	5ACEH	
TEMPPB2	5ACFH	
TEMPPB3	5AD0H	
LASTSTAT	5AD1H	Postavení hodnoty portu na poslední přerušení
SRSTORE	5AD2H	(2) Zásoba ukazatele uložení použitím přerušení schopné maskování
JVSP	5AD5H	(2) Zásoba ukazatele uložení JSVIN
NMISF	5AD7H	(2) Neschopno , maskování přerušení zásoby ukazatele uložení
NMILRP	5AD9H	(1) Hodnota LMPR, když se přihodí NMI

Vektory (viz detailní popis jinde)

DMPV	5ADAH	(2) Výpis obrazovky
SETIYV	5ADCH	(2) Nastavení registru IY pixelu výstupní rutiny
PRTOKV	5ADEH	(2) Tisk známky
NMIV	5AE0H	(2) Přerušení neschopné maskování
FRAMIV	5AE2H	(2) Přerušení snímku
LINIV	5AE4H	(2) Přerušení řádku
COMSV	5AE6H	(2) Přerušení úvozovek
MIPV	5AE8H	(2) Přerušení MIDI vstupu
MOPV	5AEAH	(2) Přerušení MIDI výstupu
EDITV	5AECH	(2) Editor
RST8V	5AEEH	(2) RST 0BH (chybný ovladač)
RST2BV	5AF0H	(2) RST 2BH (kalkulátor pohyblivé řádové čárky)
RST3OV	5AF2H	(2) RST 30H (uživatel RST)
CMDV	5AF4H	(2) Povel
EVALUV	5AF6H	(2) Vyjádření vyhodnocovače
LPRTV	5AF8H	(2) Tiskárna
MTOKV	5AFAH	(2) Protější známka

MOUSV	5AFCH	(2) Myš
KURV	5AFEH	(2) Editační kurzor

Seznamy použity přes MODE 3 a 4 k rozšíření znaku vzoru dat (jako např. uložení v UDG) má formu vhodnou k umístění na obrazovce. V MODE 3 každý seznam obsahuje 16 bajtů (rozšiřován verzemi každého možného pokusu), zatímco v MODE 4 každý seznam obsahuje 16 slov (rozšiřováno verzemi každého pokusu).

CEXTAB	5B00H	(32) Barevný seznam rozšíření - (běžné PEN, PAPER a INVERSE použity jsou k úpravě těchto dat.
EXTAB	5B20H	(32) Nebarevný seznam rozšíření. Pokus 1010b v MODE 3 nebo 4 je rozšířen na 11001100b, nebo 1111000011110000b.
COMPFLG	5B40H	Příznak bitů použitých přes kompilátor LABEL/FN/PROC
BREAKDI	5B41H	Nenulový, jestliže ESC mezi údaji není možno použít.
ERRSTAT	5B42H	Údaj jde na ON ERROR
ERRLN	5B43H	(2) Rádek jde na ON ERROR
ONERRFLG	5B45H	Bit 7=dočasně aktivuje příznak ON ERROR, bit 0 je trvalý
ONSTORE	5B46H	Číslo zprávy povelu ON
BCSTORE	5B47H	(2) Použít k obsahu registru BC přes RST 30H
M3PAPP	5B49H	(2) Použít k obsahu MODE 4 PAPER a PEN když je použit MODE 3
M3LSC	5B4BH	(2) Použít MODE 4 barev nižšího okna, když je použit MODE 3
TEMPW4	5B4DH	(2) Použít přes rutinu ADJUST POINTERS
TEMPW5	5B4FH	(2) Použít přes rutinu ADJUST POINTERS
RESERVED	5B51H	

Seznam použit přes editor ukazuje, které řádky obrazovky mají čísla řádků.

LPT	5B52H	(30)
RESERVD	5B70H	(2)
RNSTKE	5B72H	(2) Přejmenování ukazatele zásobníku použitím přes procedury REF
CURCMD	5B74H	Běžný povel v BASICu
LTDFF	5B75H	Příznak LET/DEFAULT
STRM16NM	5B76H	(11) Kódování typ/délka a jméno proměnné, že stream 16 tiskne. Použití přes povel RECORD TO.
GRARF	5B81H	Grafický záznam příznaku (0=vypnuto)
DHADJ	5B82H	Úprava dvojnásobné výšky. Nula, jestliže je tištěn spodní znak dvojnásobné výšky.

PAGCOUNT	5B8EH	Čítač stránky použitý přes FARLDIR a FARLDDR
MODCOUNT	5B84H	(2) Čítač MOD 16K použitý přes FARLDIR a FARLDDR
BCREG	5B86H	(2) BC registr kalkulátoru
AUTOFLG	5B88H	Nula, jestliže AUTO je vypnuto, jinak 1
AUTOSTEP	5B89H	(2) Krok hodnoty povolení AUTO
LSPRT	5B8BH	(2) Ukazatel rozkládaného řádku použitím přes editor
HNPTR	5B8DH	(1) Použití přes editor
MSEDP	5B8EH	(8) Použití přes myš jako uložitě dat
BUTSTAT	5B8FH	Postavení knoflíku myši
MXCRD	5B96H	(2) Souřadnice X myši
MYCRD	5B98H	(2) Souřadnice Y myši

Použití rutiny PRINT a NUMBER

FRACLM	5B9AH	
NPRPOS	5B9BH	(2)
DIGITS	5B9DH	
EPOWER	5B9EH	
DECPLNTED	5B9FH	
PRNBUFF	5BA0H	(16) Tvar čísla ASCII
BCDBUFF	5BB0H	(5)
OTHER	5BB5H	Určuje postavení čísla
RESERVED	5BB6H	
SLDEVT	5BB7H	Dočasný záměr písmene (obvykle kopirován z SLDEV).
SLNUMT	5BB8H	Dočasný záměr čísla (obvykle kopirováno z SLNUM).
OVERF	5BB9H	Príznak "SAVE OVER" použitým přes DOS. Nula, pokud SAVE OVER, jinak nenulová.
INSLV	5BBAH	(2) Blok pohybu vektoru
STRLOCN	5BBCH	(2) Použití rutiny LOOK FOR A STRING / ARRAY VARIABLE (proměnnou/ketezec pole)
TVDATA	5BBDH	(2) Použití v obsluze řidicích kódů a jejich parametrů.
DOSER	5BC0H	(2) Jestliže adresa není nulová, DOS skáče na výstup
DOSFLG	5BC2H	Nula, pokud není DOS vložen, jinak číslo stánky obsahuje DOS
DOSCNT	5BC3H	Bit 0 je dan, jestliže DOS je v systému řízení
BSTKEND	5BC4H	(2) Konec zásobníku BASICu (použití přes procedury DO, GO SUB)
BASSTK	5BC6H	(2) Počátek zásobníku BASICu
HEAPEND	5BC8H	(2) Konec systému hromadění
HPST	5BCAH	(2) Počátek systému hromadění

FPSBOT	5B0CH	(2) Počátek zásobníku kalkulačoru pohyblivé řádové čárky
DKDEF	5BCEH	(2) Počátek ohrazení KEYCODE
DKLIM	5BD0H	(2) Hranice adresy buferu KEYCODE
PATOUT	5BD2H	(2) Adresa výstupní rutiny umožňující tisk znaků
ERRMSG5	5BD4H	(2) Počátek přehledu chybových hlášení
UMSGS	5BD6H	(2) Počátek přehledu užitečných hlášení
KBTAB	5BD8H	(2) Počátek přehledu klávesnice
CMDADDR5	5BDAH	(2) Počátek přehledu adresy povolení
MNOP	5BDCH	(2) Adresa hlavní výstupní rutiny
MNIP	5BDEH	(2) Adresa hlavní vstupní rutiny
PAGER	5BE0H	(14) Reservováno pro podprogram stránkování
KBUFF	5BEEH	(18) Přehled klávesnice - 2 přehledy, když 72 bitů
LHM1	5C00H	Použití přes prohlídku klávesy
LAST	5C01H	Uhození poslední klávesy. Nula, jestliže klávesa nebyla stisknuta. Zastavuje změnu, pokud nejsou klávesy čteny, když se bufer naplňuje RES 5 (FLAGS) je viděno jako čtení.
KDATA	5C02H	Použití přes prohlídku klávesy
LKPB	5C03H	(2)
REPCT	5C05H	Použití přes prohlídku klávesy
LASTKV	5C06H	(2) Posuv a kody klávesy stisknuté naposledy
KBHEAD	5C08H	Klávesa z buferu hlavní řady. Má hodnotu poslední klávesy. Potřebuje periodické hodnoty bitu 5 (FLAGS) nebo se bufer naplňuje
REPDEL	5C09H	Zpoždění před opakoványmi klávesami (50 x za sec), obvykle 33
REPSPD	5C0AH	Zpoždění mezi klávesou opakuje (50 x za sec) obvykle 3
RESERVED	5C0BH	
STREAMS	5C0CH	(42) Pro streamy 5-15 slovo dává přemístění ze začátku plochy kanálů k uvedení kanálu. Pokud je slovo nula, proud je uzavřen. Proud 16 je rozdelen jako proud = 4
CHARS	5C36H	(2) Adresa 256 o bajtu nižší než počátek hlavního postavení znaku
ERRSOUND	5C38H	Délka chyby zvuku = 50 za sec. Obvykle 60
CLICK	5C39H	Délka cvaknutí klávesnice (obvykle nula)
ERRNR	5C3AH	Chybnné číslo
FLAGS	5C3BH	Hlavní bajt příznaků
DFLAG	5C3CH	Příznaky zobrazení
ERRSP	5C3DH	(2) Hodnota SP k použití, když se stane chyba
LISTSP	5C3FH	(2) Hodnota SP k použití, když automaticky seznam vyplňuje obrazovku
RESERVED	5C41H	

NEWPPC	5C42H	(2) Nový řádek skoku
NSPPC	5C44H	Nový údaj ke skoku nebo FFH
PPC	5C45H	(2) Běžné číslo řádku během provedení programu
SUBPPC	5C47H	Běžné číslo údaje
BORDCR	5C48H	Atributy pro nižší obrazovku v MODE 1 a 2
EPPC	5C49H	(2) Číslo řádku s kurzorem
BORDCOL	5C4BH	Hodnota k oděslání na port okolí (border)
CHANS	5C4FH	(2) Počátek plochy kanálů
CURCL	5C51H	(2) Počátek běžného kanálu
DEFADDP	5C53H	Adresa DEF FN (strana)
DEFADD	5C54H	(2) Adresa DEF FN (doplňek)
ESERVED	5C56H	(15)
STKEND	5C65H	(2) konec zásobníku kalkulátoru pohyblivé řádové čárky
KPFLG	5C67H	Funkční klávesy, jestliže je sudý, číslo jestliže je liché
MEM	5C68H	(2) Počátek plochy paměti kalkulátoru
KLFLAG	5C6AH	8, jestliže je stisknutý CAPS LOCK, jinak 0
SDTOP	5C6CH	Číslo horního řádku v automatickém listování
COPPC	5C6EH	Číslo řádku následující po CONTINUE
COSPPC	5C70H	Číslo údaje následující po CONTINUE
FLAGE	5C71H	Příznaky použity přes povel INPUT a editor
STRIL	5C72H	Použití, když proměnné jsou vyhrazeny
SEED	5C76H	Náhodné číslo. Dáno přes RANDOMIZE
FRAMES	5C78H	(3) Snímky, ježto přístroj byl zapnut (první LSB)
UDG	5C7BH	(2) Adresa CHR\$ 144
HUDG	5C7DH	(2) Adresa CHR\$ 169 (počátek neohraničen)
FRAMES34	5C7FH	(2) 2 další bajty čítače FRAMES
OLDPOS	5C82H	Použit přes editor na odstranění nižšího okna
SCRCT	5C8CH	Čítač použit, abyhom dostali náznak "scroll"
KBQB	5C8DH	(8) Klávesnice, ve verzi 1.0 ROM 5A5AH
KBQP	5C95H	(2) Ukazatelé klávesnice konec / vrchol. Ve verzi ROM 5A6AH
RESERVED	5C97H	(6)
SCPTR	5C9DH	(2) Adresa vstupu běžné obrazovky ve SCLIST
FISCRNP	5C9FH	Stránka obrazovky 1
SCLIST	5CA0H	(16) Seznam obrazovek. MODE / strana obrazovky 1-16 nebo FFH, jestliže je obrazovka zavřená. Bity 6 a 5 = MODE, 4-0 = stránka
LASTPAGE	5CBOH	Poslední stránka rezervovaná po BASIC. Dána použitím OPEN, OPEN TO nebo CLOSE.
RAMTOPP	5CB1H	Stránka RAMTOP
RAMTOP	5CB2H	(2) Doplňek RAMTOP

PRAMTP

SCB4H Poslední stránka 16k fyzicky přítomná v přístroji

Formy proměnné

Coupé má 5 hlavních druhů proměnných.

- číslicové
- proměnné FOR-NEXT
- řetězce
- řetězcová pole
- číslicová pole

Číslicové proměnné a proměnné FOR-NEXT jsou drženy v oblasti NVARS a NVARSP. Tato oblast začíná dvacetisešti 2-bajtovými vstupy, které obsahují doplněk první proměnné začínající daným písmenem (A-Z – odtud 26 vstupů). Jestliže je doplněk FFFFH, proměnné nezačínají tímto písmenem. Začátek proměnné v každé číslicové poměnné má tuto formu:

Typ / jméno délka bajtu, LSB, MSB, znak 1, znak 2..... atd. Bajt typ/délka ukládá jméno délka-1 v bitech 4-0 (jména mohou být dlouhá nejvíce 32 znaků). Bit 5 je dán, jestliže se nepoužívá delší proměnná, bit 6 je dán pro proměnnou FOR - NEXT, bit 7 je dán, jestliže proměnná je "eskrytá" (je použita procedura k provedení proměnných LOCAL). Je daná vzdálenost pohybu z MSB běžného doplnku bajtu typ/délka příští proměnné začínající stejným písmenem nebo FFFFH, jestliže zde není více takových proměnných. Smí mit 0 až 31 znaků po předání, působici na vnitřní počet, jestliže jsou použita písmena a vyloučeny mezery.

Potom přichází 5 bajtů, které obsahují hodnotu proměnné. (Podívej se do části kalkulátoru pohyblivé řádkové čárky pro informaci uspořádání těchto bajtů). Pokud to je obyčejná číslicová proměnná, je zde správná hodnota, ale proměnná FOR - NEXT má dalších 14 bajtů, úplný tvar je:

VALUE (5):LIMIT (5):STEP (5):LOP ADDRESS (3):STATEMENT (1)

Adresa cyklu je stránka a dána (8000 – BFFFFH) počátkem řádku ke zpětnému cyklu a údaj číslo údaje bez této řádky ke zpětnému cyklu (cykly k prvnímu údaji jsou tedy nejrychlejší).

Řetězce a pole jsou drženy v oblasti dostupné přes SAVARS a SAVARSP. Zde je zóna vyrovnávací paměti (buferu) mezi číslicovými proměnnými a proměnnými řetězec / pole tak, že tvorba nových číslicových proměnných obvykle nepožaduje pohyb řetězců a polí.

Proměnné v této oblasti jsou v pořadí jejich tvorby (cestou DIM nebo LET; LET A\$ = A\$ posune A\$ na konec oblasti, jestliže A\$ není pole). Pro každou proměnnou je následující tvar:

Typ / délka bajtu, znak 1 nebo znak 10, délka (3)

Bajt typ / délka obsahuje délku jména (1-10) v bitech 4-0. Bit 6 je dán pro řetězcová pole, bit 5 pro číslicová pole a oba

byty jsou obnoveny pro běžné řetězce. Bit 7 je dán, pokud je proměnná "skrytá". Deset bajtů dovoluje prostoru obsáhnout jméno proměnné; nepoužité bajty nejsou definovány. Délku dostanete jako číslo stránek 16k, potom délku MOD 16k (LSB/MSB) pro zbytek pole plus 3 délky bajtů. Pro řetězce text řetězce následuje; pro pole je další bajt číslo rozměrů, následován délkou rozměru 1, rozměru 2 atd. (2 bajty pro každý rozměr). Data pole potom následují, všechna data s prvním popisem 1, které jsou první, potom data s prvním popisem 2 atd. FFH ohraňuje přehled řetězcových proměnných a proměnných pole.

Forma programu v BASICu

První řádek programu je zpracován přes POG a PROGP (dáno 8000 -BFFFH a stránky) Každý řádek má tvar:

LINE NUMBER (MSB/LSB):	LINE LENGTH (LSB/MSB):	TEXT: ODH
číslo řádku	délka řádku	

Poznámka: číslo řádku je neobvyklé na uložení nejdůležitějšího prvního bajtu. Délka řádku je délka textu a omezovače ODH. Řádek textu obsahuje stlačená klíčová slova (viz seznam kódu klíčového slova) a neviditelné formy všech přesných čísel (desítkové, dvojkové a šestnáctkové soustavy). Neviditelné formy jsou OEH následován 5-ti bajty hodnoty uložení. Řádek může také obsahovat řídicí kódy a jejich parametry.

Závěrečný řádek v programu následuje přes FFH (max.dovolené číslo řádku je FFFFH).

Diskový operační systém

Úvod

SAMDOS byl vytvořen speciálně pro SAM Coupé. Je podobný G+DOS který MGT využila pro Plus D Spectrum disk interfejs.

Disketová jednotka

Vnitřní disketová jednotka SAMa je snížený typ Citizen 3,5". Každá jednotka je zkompletována společně s diskovým řadičem MGT, který využívá řadič pružných disků VL-1772-02. Diskety jsou formátovány oboustranně, 80 stop na stranu, 10 sektorů na stopu, ve standardu IBM 3740.

Formát diskety

Používáme 80 stop na každou stranu diskety, to dává dohromady 160 stop na disketu. Stopa je rozdělena na 10 sektorů, na každý z nich je možné uložit 512 byteů. První čtyři stopy diskety jsou určeny pro řízení SAMDOSu, zbývajících 156 stop je k dispozici pro ukládání (dat). K dispozici je tedy 1560 datových sektorů po 512 bytech (to je 798 720 byteů).

I když každý datový sektor může obsahovat 512 byteů, jen 510 z nich je k dispozici pro uložení (dat). Poslední dva byty každého datového sektoru jsou využívány DOSem k určení místa, kde je uložena další část souboru. Byte 511 určuje další stopu a byte 512 další sektor, kde se nachází další část souboru.

Hlavička diskového souboru

Na začátku každé diskové nahrávky souboru je jeho hlavička. Hlavička souboru je dlouhá 9 byteů.

byt	SAMDOS	PLUS D
0	typ souboru	typ souboru
1-2	modul délky	délka souboru
3-4	OFFSET START	startovací adresa
5-6	nevyužité	
7	Počet stránek	
8	číslo počáteční stránky	

Podrobnosti o hlavičce Plus D je možné nalézt v technických informacích pro Plus D.

Typ souboru

Každý typ souboru je v SAMDOSu určen příslušnými číselnými označením, i.

5	- ZX nahrávka fotografického snímku	SNP 48k
16	- Program v SAM BASICu	BAS
17	- Číselná data	D ARRAY
18	- Stringová data	\$ ARRAY
19	- Soubor ve strojovém kódu	C
20	- Nahrávka obrazovky	SCREEN\$

Modul délky a počet stránek

V hlavičce SAMDOSu je délka souboru počítána násobením počtu stránek (byte 7) číslem 16384 a přičtením modulu délky (slovo (t.j. byte) 1-2), LSB/MSB, t.j. délka MOD 16k.

OFFSET START + číslo počáteční stránky

Přečtěte si číslo počáteční stránky (byte): a to společně s 1FH dá číslo stránky v rozsahu 0 - 31. Abyste našli počátek, vynásobte číslo stránky číslem 16384, přičtěte OFFSET a odečtěte 4000H. (Protože ROM zabírá 0 - 3FFFH).

SAMDOS je uložen v oblasti nad 4000H a ROM 0 je rozmístěna v oblasti 0 - 3FFFH.

Rizení SAMDOSu

První čtyři stopy diskety jsou přiděleny pro řízení disku, počátek je na stopě 0, sektor 1. Tyto čtyři stopy nám dívají 40 sektorů, z nichž každý je rozdělen na dvě části po 256 bytech. Každá z těchto částí bude identifikovat jednu nahrávku (soubor), takže dovoluje až 80 záznamů pro řízení.

Formát každého řídícího záznamu je následující: (uživatelský informační soubor UIFA bude popsán později)

Byte	UIFA	Popis
0	0	STATUS / TYP SOUBORU Tento byte je přidělen pro jeden z typů souboru, jak již bylo uvedeno dříve, ale je také využit jako Status souboru. Je-li tento byte roven 0, je soubor smazán. Je-li soubor souborný je zapnut bit č.7.Je-li soubor chráněn proti přepisu, je zapnut bit č.6.
1-10	1 - 10	Jméno souboru Toto jméno může mít délku až 10 znaků.
11		MSB počtu sektorů využitých souborem
12		LSB počtu sektorů využitých souborem
13		Číslo stopy, na které soubor začíná
14		Číslo sektoru, na kterém soubor začíná
15-209		Adresovaná mapa sektorů (195 bytů) (Podrobnosti viz dále)
210-219		Budoucnost a minulost MGT (10 bytů).Byly využity v řízení PLUS D,ale není využito v SAMDOSu.Jsou přiděleny MGT pro další využití
220	15	Návěšti (využívá pouze MGT)
221-231		Informace o typu souboru
16-26		Je-li soubor typu 17 nebo 18,pak tyto byty obsahují typ souboru/délku a jméno
16		Je-li soubor typu 20,pak tyto byty obsahují mód obrazovky
16-18		Je-li soubor typu 16,pak tyto byty obsahují délku programu bez proměnných
19-21		Je-li soubor typu 16,pak tyto byty obsahují délku programu + číselné proměnné

Byte	UIFA	Popis
	22-24	(Je-li soubor typu 16, pak tyto byty obsahují délku programu+číselných proměnných + délku mezery před stringy a indexovanými proměnnými)
232-235	27-30	Záložní 4 byty (rezervováno)
236	31	(číslo počáteční stránky v bitech 4-0, byty 7-5 nejsou definovány)
237-238	32-33	OFFSET stránky (8000-BFFFH). Je to jakási pomocná hlavička souboru.
239	34	(Počet stránek v délce(jako pomocná hlavička))
240-241	35-36	(Modul délky od 0 do 16383, t.j. délka nahrávky MOD 16384 (jako pomocná hlavička))
242-244	37-39	(Výkonná adresa (Prováděcí) Výkonná adresa v případě souboru ve strojovém kódu, nebo číslo řádku v případě samospoustěcího programu v BASICu.)
245-253	40-47	Záloha 8 bitů
254-255		(Pro další využití, pouze pro MGT)

Adresová mapa sektorů

SAMDOS přiděluje adresové mapy sektorů 195 bytů, t.j. 1560 bitů, což je počet sektorů na disku, přístupných pro soubory.

Adresová mapa sektorů je počítána pro každý řídící záznam. Poté, co je vytvořen soubor, je zpracován řídící záznam pro tento soubor. Adresová mapa sektorů je vytyčena zapnutím určitých bitů, které korespondují se sektory přidělenými souboru. (Bit 0 prvního bytu je přidělen stopě č.4, sektoru 1). Pro příklad: Zabírá-li soubor 5 sektorů, potom je 5 příslušných bitů (vztahujících se k těmto sektorům) adresové mapy zapnuto (1) a uloženo jako součást řídícího záznamu.

Adresová mapa bitů (BAM)

Adresová mapa bitů není SAMDOSem na disk ukládána. Tato je vytvářena prostřednictvím funkce OR adresové mapy sektorů každého souboru. Toto pak pro SAMDOS vytváří uživatelskou mapu disku.

Když je vytvořen soubor, první věcí, kterou SAMDOS udělá, je výpočet BAM, a po zjištění přístupných sektorů (t.j. jejich bity nejsou zapnuty), je-li pro soubor prostor, může pokračovat dále. Jestliže je pro soubor dostatek prostoru, je vytvořen řídící záznam včetně adresové mapy sektorů, příslušící souboru a soubor je uložen do sektorů, které byly specifikovány v adresové mapě sektorů tohoto souboru.

Interfacing SAMDOSu

Strojový kód SAMDOSu poskytuje ukazatele pro specifické části DOSu, které jsou nazývány hook kódy (hook kódy) (pomocné rutiny). Tyto ukazatele provádí funkce DOSu např.: SAVE, LOAD, VERIFY atd. Pro použití těchto hook kódů musí být různé registry procesoru Z80B nastaveny v závislosti na volených podprogramech k provedení požadovaných úloh. Každý hook kod je v tomto manualu později popsán společně s nezbytnými informacemi oregistrech.

Je-li použít hook kod, ROM přesune SAMDOS přechodně do sekce B čtyřiašedesátikilové adresovatelné paměti (t.j. od 4000H výše).

RST 08H	;volá ROM RST 8
DEFBX	;kde x je určitý požadovaný hook kod

Vyskytne-li se chyba, bude normálně zachycena ROMkou a zároveň o ni bude ve spodní části obrazovky zobrazeno chybové hlášení. Je-li systémová promenná DOSER (SBCOH) nahrána s adresou Vašeho vlastního podprogramu "návratu s DOSu", potom bude tato rutina spuštěna až SAMDOS vykoná nějaký hook kod, nebo porovná-li některý z chybových kódů BASICu s obsahem akumulátoru, kde může být 0 (t.j. žádná chyba) nebo číslo nějaké chyby. Toto číslo pak může být buď jeden z chybových kódů BASICu nebo chybový kod SAMDOSu od 108 výše.

Uživatelský informační soubor (UIFA)

Při použití hook kódů SAMDOSu musí volací program nastavit oblast v paměti, určenou registrém IX, který upřesňuje informace pro ovládání souboru, např.: SAVE nebo LOAD. Tento soubor se nazývá UIFA.

byte	popis
0	STATUS / typ souboru
1-14	Jméno souboru. Povolen je 14-ti znakový identifikační název, např. D1:filenamexx. Protože ale SAMDOS bude oddělovat identifikátor záznamové mechaniky, je maximální délka jména souboru 10 znaků.
15	Návěsti
16-26	Je-li nahrávka typu 17 nebo 18, pak tyto bajty obsahují typ nahrávky / délku a jméno
16	Je-li nahrávka typu 20, pak tyto bajty obsahují mód obrázovky
16-18	Je-li nahrávka typu 16, pak tyto bajty obsahují délku programu bez proměnných
19-21	Je-li nahrávka typu 16, pak tyto bajty obsahují délku programu plus číselných proměnných
22-24	Je-li nahrávka typu 16, pak tyto bajty obsahují délku programu plus číselných proměnných a délku mezer před stringy a řadou proměnných
27-30	Záložní 4 bajty (reservováno)
31	Cílovo počáteční 16k stránky
32-33	OFFSET stránky (8000-BFFFH) LSB/MSB
34	Počet stránek v délce
35-36	Modul délky od 0 do 16383, t.j. délka souboru MOD 16384
37	Cílovo prováděcí (výkonné) stránky, je-li vhodná
38-39	prováděcí OFFSET (8000-BFFFH) LSB/MSB, je-li vhodný
40-47	Záloha 8 bytů (poznámkové pole)

Když DOS použil UIFA a chce ji dostat zpět, např. pro potvrzení natazení (programu) volaným programem, DOS vytvoří informační záznamové pole disku (DIFA). Jak UIFA, tak DIFA mají délku 48 bytů. DIFA je zapsána do oblasti paměti 80 bytů nad uživatelem specifikovanou UIFA.

HOOK kody SAMDOSu

SAMDOS poskytuje uživateli hook kody, které umožňují programátorům ve strojovém kódě využívat možnosti SAMDOSu bez navratu do BASICu nebo jeho volání.

Dojde-li k chybe, SAMDOS nastaví návěsti a uloží číslo chyby do akumulatoru. Nedojde-li k chybě, v akumulatoru bude 0.

Bezne dostupné hook kody jsou:

INIT	128 dec	inicializuje a očekává soubor auto
HGTHD	129 dec	dá hlavičku souboru
HLOAD	130 dec	natahne soubor
HVERY	131 dec	zkontroluje soubor
HSAVE	132 dec	uloží soubor
HVAR	139 dec	dodá adresu proměnné DVAR
HFLE	147 dec	otevře soubor
SBYT	148 dec	uloží byte
HWSAD	149 dec	zapíše sektor na disk
HSVBK	150 dec	uloží blok dat
CFSM	152 dec	uzavře sektrovou mapu souboru
HGFLE	158 dec	dodá soubor z disku
LBYT	159 dec	natahne byte
HRSAD	160 dec	prečte sektor z disku
HLDBK	161 dec	natahne blok dat z disku
REST	164 dec	přesune čtecí hlavičku nad stopu 0
FCAT	165 dec	provede listing adresare
HERAZ	166 dec	vymaže soubor z disku

Záloha byla zabezpečena pro připadné další hook kody, které by byly přidány k SAMDOSu.

Vysvětlení HOOK kódů

Pokud se vysvětlení hook kódů odvolává k " RPT ", týká se to určitého ukazatele uvnitř SAMDOŠU.

INIT Tato rutina očekává soubor AUTO na běžném (aktivním) disku a inicializuje proměnnou DVAR\$.

HGTHD Dodá hlavičku souboru. Tato rutina by byla volána stavem registru IX do UIFA, která by obsahovala typ souboru a požadovaný název souboru. Po sestavení bude celá hlavička převědena do formy UIFA na bajty IX + 80.

HLOAD Natáhne soubor, určený v UIFA registrém IX. Registr C obsahuje počet 16-ti kilových stránek využitých souborem, registrový pár DE musí obsahovat modul délky 16k. Registrový pár HL musí označit cíl mezi 8000-BFFFH, zatímco cílová stránka musí být stránkována využitím registru HMPR. Tyto hodnoty lze získat z hlavičky nahrané příkazem HGTHD.

HVERY Kontrola paměti s nahrávkou uloženou na disku. Registr IX musí být opět ukazatelem souboru v UIFA. Užívá se jako HLOAD, ale častěji.

HSAVE Uložení souboru jenž je v UIFA určen registrém IX. UIFA musí být kompletní, včetně délky souboru, atd.

HVAR Tato rutina umístí nesetříděné DVAR vstupně do registru BC. Rutina potom dodává adresu DVAR jejím přidáním nad BASIC.

HOFLE Otevírá soubor na disku. Registr IX je ukazatelem pro UIFA. Rutina bude vytvářet adresovou mapu sektorů, uloží hlavičku na disk a zresetuje ukazatel RPT.

SBYT Ukládá byte do akumulátoru RAM určené ukazatelem RPT. Jeli sektor zaplněn, data budou uložena do následujícího sektoru určeného adresovou mapou sektorů.

HWSAD D obsahuje číslo stopy, E číslo sektoru. Akumulátor obsahuje číslo disketové jednotky (1 nebo 2). Registrový pár DE určuje sektor k zápisu. Akumulátor obsahuje číslo drážky, registrový pár HL určuje zdrojová data, která musí být uložena v 64k adresovatelné oblasti.

- HSVBK Uloží blok dat na disk, registrový pár D ukazuje začátek dat a registrový pár BC obsahuje čitací bajt (byte).
- CFSM Uzavírá soubor sektorové mapy. Tato rutina vyprázdní RAM a kopíruje záhlavi oblasti za adresář, uzavře soubor, potom obnoví adresář.
- HGFLE Dodá soubor z disku. Registr IX musí určovat UIFA. Návrat je zabezpečen prvním sektorem souboru nahraného do RAM a RTP určujícím první byte.
- LBYT Natáhne bajt určený RTP z RAM, umístí ho do akumulátoru a připojí RTP. Když je sektor celý přečten, je z disku natažen další a je upraven ukazatel.
- HRSAD Registr D obsahuje číslo stopy, registr E obsahuje číslo sektoru. Akumulátor obsahuje číslo diskové mechaniky (1 nebo 2). Je čten sektor určený registrovým párem DE. Akumulátor obsahuje číslo drajvu, registrový pár HL určuje cíl.
- HLDBK Natáhne blok dat z disku do oblasti paměti určené registrovým párem DE. Počítadlo bloků je v registrovém páru BC.
- REST Nastaví mechaniku disku na stopu 0. Akumulátor obsahuje číslo drajvu, t.j. 1 nebo 2.
- PCAT Provede průběžný listing adresáře.
- HERAZ Vymaže soubor z disku. Registr IX musí ukazovat na UIFA toho souboru, který má být smazán.

Umístění SAMDOSu

Když je natahován SAMDOS, ROM prohlédne dostupnou paměť a natáhne SAMDOS do poslední volné 16k stránky. ROM využívá poslední dvě 16k stránky pro SCREEN 1, takže SAMDOS je obvykle natažen do třetí 16k stránky odzadu, ale může to být i stránka jiná, jestliže byly před natažením SAMDOSu otevřeny dodatečné obrazovky. Na adrese 5BC2H (SVAR 450), je uloženo číslo stránky použité pro SAMDOS, nebo nula, jestliže SAMDOS nebyl natažen.

PEEK SVAR 450 * 16384 + 16384 dává startovací adresu SAMDOSu. Je-li vyslán povel DOSu, pak je DOS natažen do sekce B (4000H) 64k adresovatelného místa, povel je proveden a potom je DOS vystránkován.

Chybové kódy SAMDOSu

- 81 Nesmysl v SAMDOSu
- 82 Nesmysl v SNOSu (síťový OS SAMa)
- 83 Chybný koncový kód
- 84 Vyžádané opuštění
- 85 Chyba TRK nnn SCT nnn
- 86 Ztracený formát TRK nnn
- 87 Zastavený disk v drajvu
- 88 Chybi BOOT soubor
- 89 Chybné jméno souboru
- 90 Chybné postavení
- 91 Chybné zařízení
- 92 Proměnná nebyla nalezena
- 93 Neúspěšná kontrola
- 94 Špatný typ souboru
- 95 Chyba setřídění (sjednocení)
- 96 Chybný kód
- 97 Zákovská sestava ?
- 98 Špatný kód
- 99 Čtení psaného souboru
- 100 Psaní čteného souboru
- 101 Není AUTO * soubor
- 102 Síť vypnuta
- 103 Není takový disk
- 104 Disk chráněn proti zápisu
- 105 Není dostatek místa
- 106 Plný adresář
- 107 Soubor nebyl nalezen
- 108 Konec souboru
- 109 Jméno souboru je již použito
- 110 Není natažen SAMDOS
- 111 Použitý stream
- 112 Použitý kanál

Zvukový stereo generátor

Všechna vysoko náročná použití softwaru na světě nemůže nahradit domácí počítač, který se stýká s uživatelem nepřitažlivou, nebo zmatenou cestou. Ke zvýšení existující úrovni vizuálního a zvukového přenosu z počítače k uživateli mohou být vyvolávány generátorem stereo zvuku SAA 1099 a 64barevným kódovačem TEA 2000 projektanty každého daného softwaru a hardwaru ke zvýšení jejich nejrealističtějších produktů.

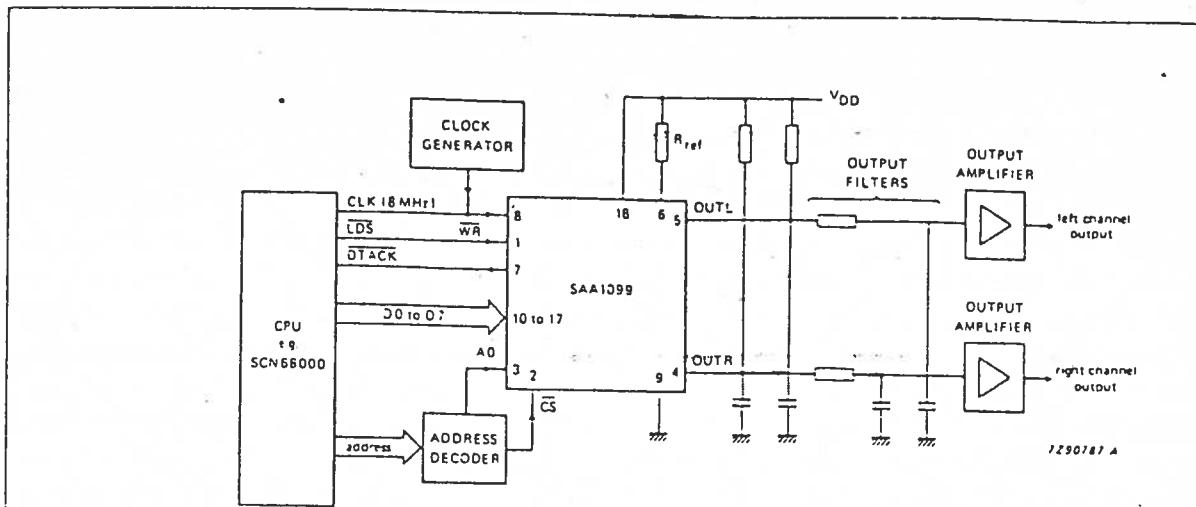
Zvukový generátor SAA 1099 je popsán v tomto článku. Může předvést širokou škálu zvukových efektů včetně napodobování hudebních nástrojů a zvuků, které jsou potřeba pro použití her.

Zatím co nejintegrovanější celky mají pouze 3 frekvenční generátory, SAA 1099 jich má šest a tóny z každého generátoru mohou být mixovány s několika druhy zvuku. Protože je zde šest frekvenčních generátorů, plně hudební akordy (včetně základních tónů tóniny) mohou být produkovány a dva akordy (mimo tóniny) mohou přesahovat. Všechny hudební tóny mohou být produkovány v 8-oktávě. Stereo efekt může rozšířit dekorace video her, je produkován ke zdvojení šesti zvukových komponentů ke tvarování totožného levého a pravého panelu signálů. Oprava každého signálu SA 1099 je prováděna interfejsy s nejvýše 8-mi bitovými mikroovládači a požaduje pouze jednoduchý filtr k potlačení jakýchkoliv vysokých frekvenčních složek na audio výstupu. To má být provedeno tak, že je požadováno minimum vnějších komponentů.

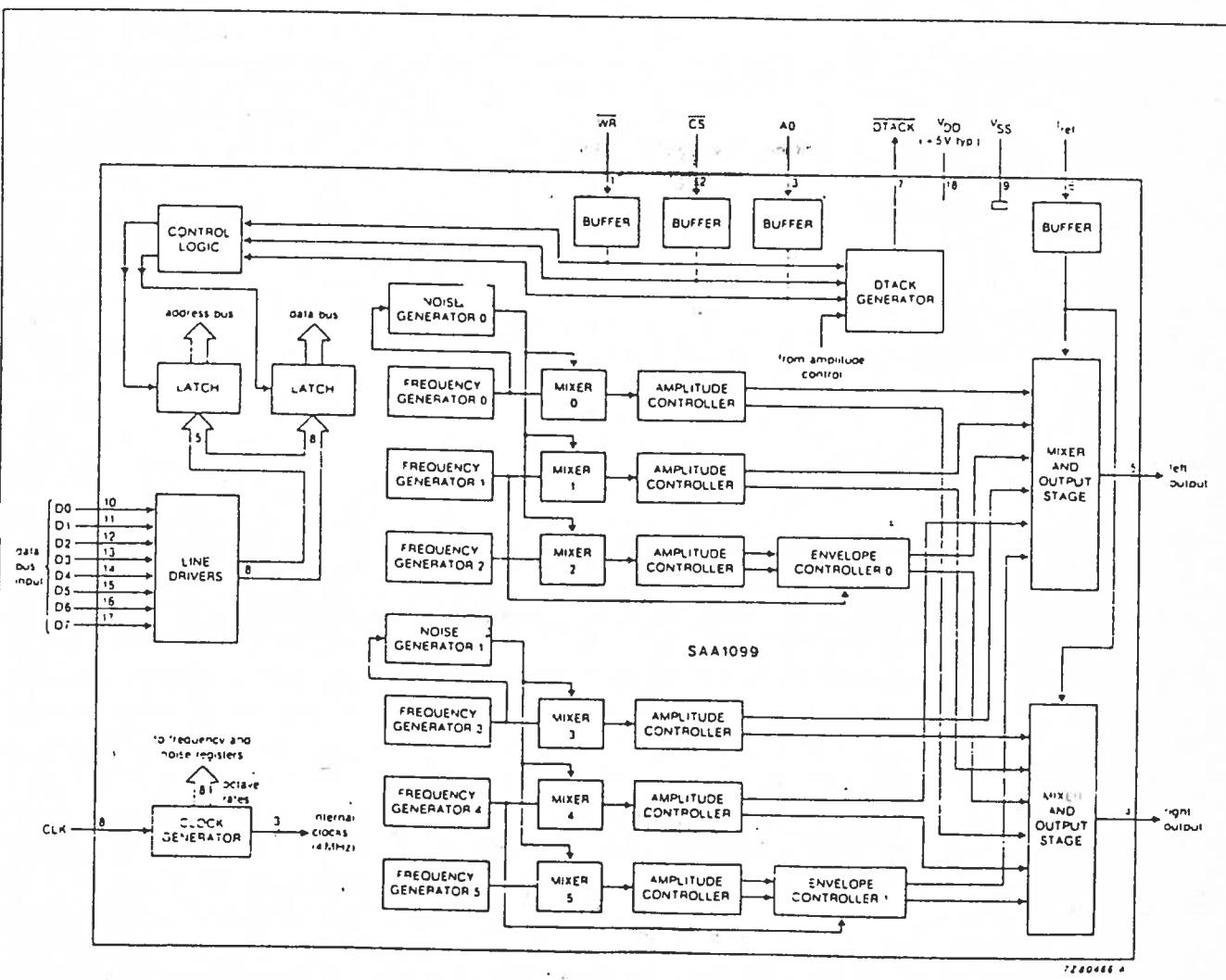
Seznam 1 ukazuje doplňkové údaje SAA 1099

Tab. 1: Skutečné údaje SAA 1099 (všechny hodnoty jsou charakteristické)

dodávané napěti	- 5 V
dodávaný proud	- 70 mA
potřebný proud	- 250 mA
ztráty rozptylem	- 500 mW
vnější kmitočet	- 8 MHz
vstupní data	- 8-mi bitový paralelní (TTL-kompatibilní)
výstupy frekvenční řady	31 Hz - 7,81 kHz (8 oktáv)



Zvukový generační systém pro domácí počítače a vybavení video her



Generátor stereo zvuku SAA 1099

Kompletní zvukový generátor

Obrázek 1 ukazuje kompletní zvukový systém. Z 8-mi bitových vstupních dat mikroprocesoru SAA 1099 generuje proměnnou stereo amplitudu, obdobný signál, rozkouskován v poměru 62,5 kHz. Základní vnější propouštěcí filtr potlačuje vysoko frekvenční složky výstupního signálu. Nastávající data, která upravují spektrum audio výstupu násobí ke zjednodušenému přispůsobení, signál AO, který se používá k indikaci, zda data jsou registrém adresy, nebo data registru. Signál AO je používán se signály CS a WR ke kontrole přeložených dat z mikroprocesoru do SAA 1099. Tyto kontrolní signály jsou slučitelné se širokou řadou mikroprocesorů. V připojení, pro optimální přispůsobení se sériovým mikroprocesorem SCN 68000, má SAA 1099 výstupy DTACK. Všechny vnitřní regulace rychlosti jsou získány z vnější 8 MHz.

Tab. 2: Funkce vstupu AO

data vstupu								funkce		
AO	D7	D6	D5	D4	D3	D2	D1	D0		
0	D7	D6	D5	D4	D3	D2	D1	D0	data pro vnitřní registry (viz tab.3)	
1	X	X	X	X	A3	A2	A1	A0	adresa vnitřního registru A3 je MSB	

X = nedbá

Tab.2 ukazuje funkci vstupu AO. Když AO=1, data označují adresu řídícího registru v SAA 1099 a tato adresa je vložena do kontrolního registru. Další data, která obsahují řídící informace registru, že byla adresována pokud AO=0. Jedenou adresován řídící registr může být doplněn o nejnovější informaci bez dalšího adresování.

Výstupy kmitočtů generátorů 0 a 3 mohou každý chvíli ovládat zvukový generátor těchto generátorů. Výstup 1 a 4, každý může ovládat obal generátoru pro vytvoření speciálních efektů.

Tabulka 3 udává adresy a bit přidělení SAA 1099 vnitřního registru používaného k ovládání frekvenčních generátorů a jiných soustav zvukového generátoru.

Registr	VSTUP DAT								POPIS - poznámky
adres	D7	D6	D5	D4	D3	D2	D1	DO	
00	pravý panel	levý panel			ovladač 0				
01	pravý panel	levý panel			ovladač 1				
02	pravý panel	levý panel			ovladač 2				
03	pravý panel	levý panel			ovladač 3				
04	pravý panel	levý panel			ovladač 4				
05	pravý panel	levý panel			ovladač 5				
06	X X X X	X X X X			rezervováno pro možné				
07	X X X X	X X X X			zvětšení rozsahu				
08	číslo tónu frekv.gen.	0							
09	číslo tónu frekv.gen.	1							15625*2↑ číslo okta
0A	číslo tónu frekv.gen.	2							frekvence tónu = -----
0B	číslo tónu frekv.gen.	3							511 = číslo tónu
0C	číslo tónu frekv.gen.	4							2 = číslo oktávy
0D	číslo tónu frekv.gen.	5							511 = číslo tónu
0E	X X X X	X X X X			rezervováno pro možné				
0F	X X X X	X X X X			zvětšení rozsahu				
10	X čís.oktávy frekv.gen.-oktaava 0(000)	31Hz až 61Hz							
	pro gen.1	pro gen.0-			1(001) 61Hz až 122Hz				
11	pro gen.3	pro gen.2-			2(010) 122Hz až 244Hz				
12	pro gen.5	pro gen.4-			3(011) 245Hz až 488Hz				
					4(100) 489Hz až 977Hz				
					5(101) 978Hz až 1,95kHz				
					6(110) 1,95kHz až 3,91kHz				
					7(111) 3,91kHz až 7,81kHz				
13	X X X X	X X X X			rezervováno pro možné				
					zvětšení rozsahu				
14	X X 5 4	3 2 1 0			umožňuje frekvenci 0-5				
15	X X 5 4	3 2 1 0			umožňuje zvuk 0-5, vztahu -				
					je se ke zvukovým mixérům				
16	X X gen.1	X X gen.0			zvukový gener.frekvenče				
	0 0	0 0			31,25 kHz				
	0 1	0 1			15,6 kHz				
	1 0	1 0			7,8 kHz				
	1 1	1 1			1 1-61Hz-15,6kHz.Frekv.gen.0 ne-				
					Ibo 3 ovládá zvuk gen.jednotlivě 0 nebo 1				
17	X X X X	X X X X							
18	obal generátoru 0				viz.tab.4				
19	obal generátoru 1				viz.tab.4				
1A	X X X X	X X X X							
1B	X X X X	X X X X							
1C	X X X X	X X RST SE			reset všech frekv.gen.				
1D	X X X X	X X X X			SE umožňuje zvuk panelům				
1E	X X X X	X X X X							
1F	X X X X	X X X X							

Zvukové generátory

Každý ze dvou zvukových generátorů má programovatelný výstup ovládaný obsahem registru 16, který určuje, zda výstup je:

- ovládající software cestou frekvence generátoru 0 a 3 (který nevyvolává tón). "Barva" zvukového generátoru je získána z dvojnásobné frekvence výstupu frekvenčního generátoru 61 Hz = 15,6 kHz
 - jeden ze tří předem definovaných zvuků založených na frekvenčích 7,8 kHz; 15,6 kHz, nebo 31,25 kHz. V tomto případě výstupy zvukového generátoru 0 mohou být smíšeny s výstupy frekvenční generátorů 0; 1 a 2 a výstup zvukového generátoru 1 může být smíšen s výstupy frekvence generátorů 3; 4 a 5 (Viz. obr. 2)
- Pro smíšení amplitudy tónu je zvětšován úměrně k tomuto zvuku.

Zvukové frekvenční mixéry

SAA 1099 má 6 mixérů, jedna část frekvenčního generátoru pro směšování tónů se zvukem. Je závislý na postavení bitů D0 až D5 registrů 14 a 15. Každý mixér může být určen:

- ke směšování zvuku a tónu
- k rozšíření jediného tónu
- k rozšíření jediného zvuku
- k nerozšíření tónu ani zvuku

Tento blok 32 registrů je opakování 8* mezi adresami 00 a FF v plné mapě vnitřní paměti. Všechny neobsazené (X) budou psány jako 0 (nuly). Čísla tónů 1 až 256 jsou platná.

Ke zvýšení možnosti zvuku, bit je umístěn jako 0 (všechny panely vyřazeny z provozu). Pokud je bit frekvenčního resetu použit všechny frekvenční generátory jsou nově vloženy a synchronizovány.

Obal ovladačů

Dva obaly ovladačů umožňují levé a pravé součásti dvou stereo panelů k určení pro:

- jednoduché nasazení
- jednoduchý rozklad
- jednoduché nasazení a rozklad (trojúhelní)
- největší amplitudu
- opakovné nasazení
- opakový rozklad
- opakovné nasazení a rozklad
- nulovou amplitudu

Obr.3 ukazuje obaly, které jsou vybrány z bitů D1 až D3 obalových registrů 18 a 19, viz.tab.4. Opakovací frekvence obalů mohou ovládat software k napsání obalu adresy registru (psaná data jsou bezvýznamná), nebo zaznamenává vnitřek na výstupní frekvenční generátoru (1 nebo 4). Obal bude vždy kompletován předtím než nový obal bude realizován. V případě opakování obalů se obal vraci k počáteční úrovni, která nutně nesmí být nulová. Když je obal plynule použit a největší amplituda uloží do obsahu registrů 00 až 05, amplituda je velikosti 7/8 obvykle přistupná.

Tab:4-Rozdělení bitů obsahu registrů generátoru (adresa 18 a 19)

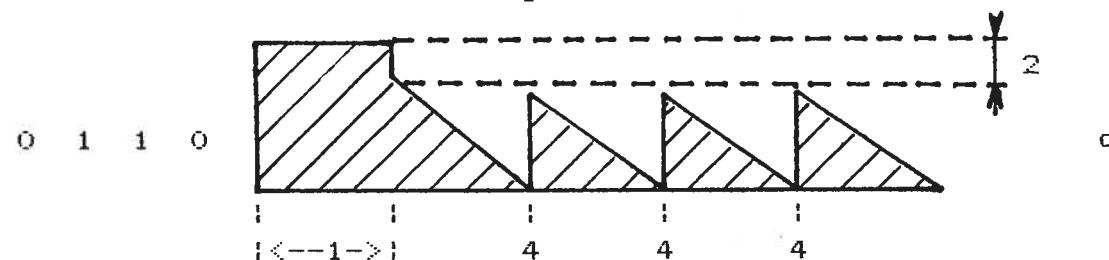
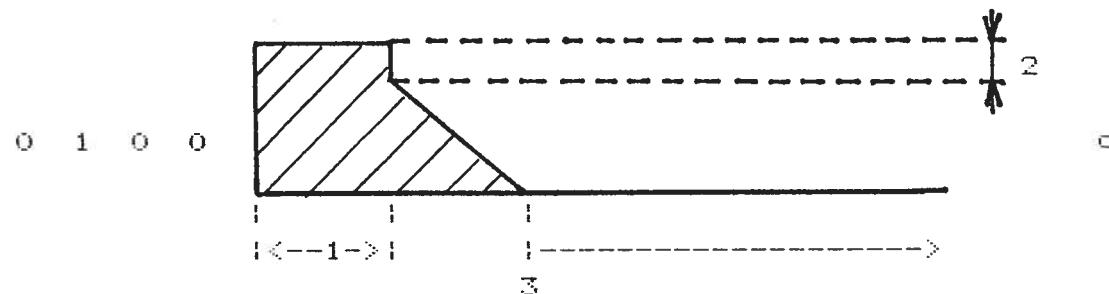
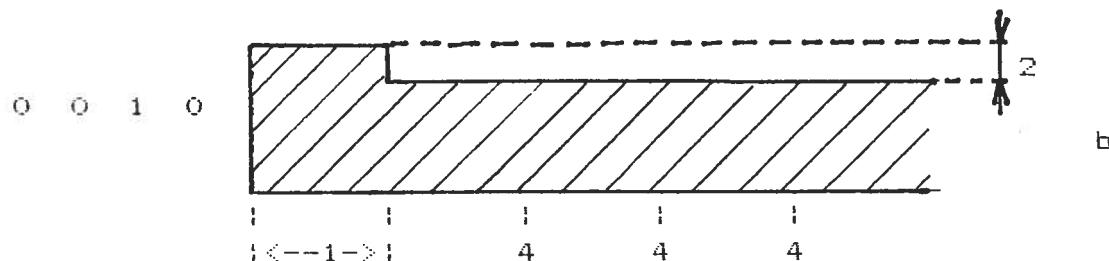
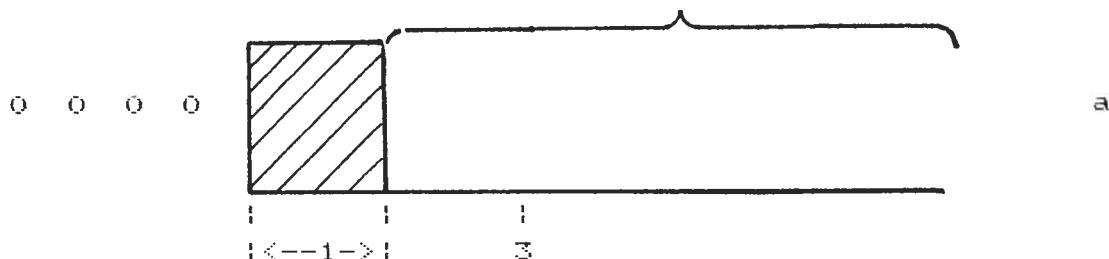
bit	funkce
D0	
0	totožné obaly pro levý a pravý panel komponentů
1	opačné obaly pro levý a pravý panel komponentů
D1 D2 D3	
0 0 0	nulová amplituda
0 0 1	největší amplituda
0 1 0	jednoduchý rozklad
0 1 1	opakováný rozklad
1 0 0	jednoduchý trojúhelník
1 0 1	opakováný trojúhelník
1 1 0	jednoduché nasazení
1 1 1	opakováne nasazení
D4	
0	4 bity rozlišení obsahu
1	3 bity rozlišení obsahu
D5	
0	vnitřní obalový záznam frekv.generátoru 1 nebo 4
1	vnější obalový záznam (adresa píše impuls AO)
D6	nedělá starost
D7	
0	reset (neovládá obsah)
1	umožňuje ovládání obsahu

Obraz 3. Obsah tvarů vlny. Tvary vln (a) až (h) z jednoho panelu umožňují vyjádřit obal generátoru.

- | | |
|---|------------------------|
| a - nulová amplituda | b - největší amplituda |
| c - jednoduchý rozklad | d - opakováný rozklad |
| g - jednoduché nasazení | h - opakovane nasazení |
| e - jednoduché nasazení a rozklad | |
| f - opakovane nasazení a rozklad | |
| i - jako h ale pro pravý panel ukazuje operaci opačného obsahu bitu D0, viz tab.4 | |

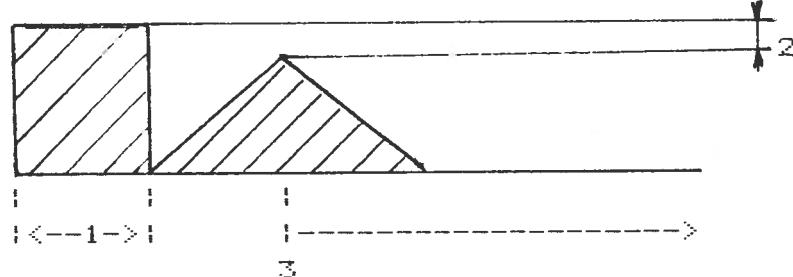
D3 D2 D1 D0 D7 = 0

D7 = 1

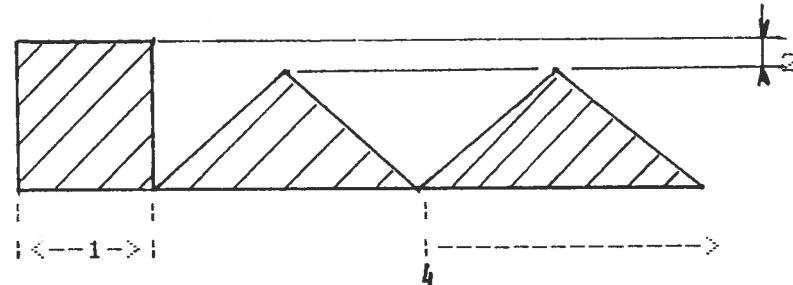


D3 D2 D1 D0 D7 = 0

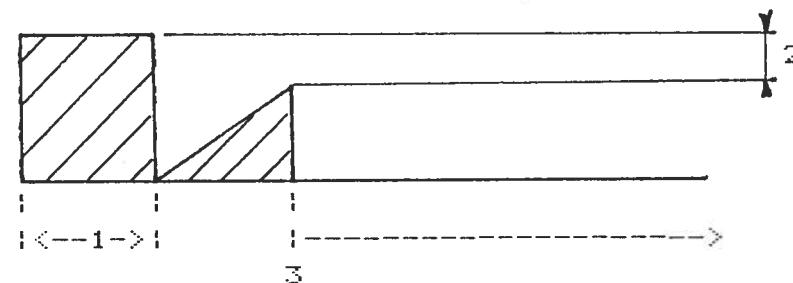
1 0 0 0



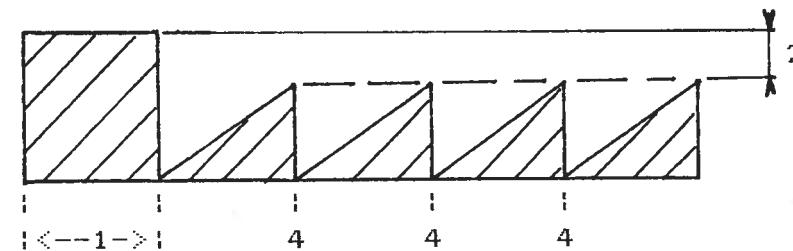
1 0 1 0



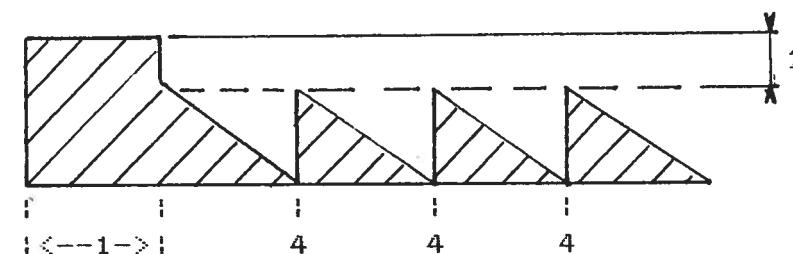
1 1 0 0



1 1 1 0



1 1 1 1



Dva typy obalu ovládání jsou rezervovány v obalu registrů pro otevřené ovládání a ovládání buferu. Otevřené ovládání vždy bere bezprostřední efekt a jsou:

- obal umožňuje (reset bit D7)
- obal rozlišuje: 16 rovin nahoru k obalu, opakovací frekvence 977 Hz, 8 rovin výše 977 Hz - bit D4

Ovládání buferu je uskutečněno jen v časech ukázaných na obr.3 a udává:

- obal tvaru - bity D1 až D3
- typ dosaženého obalu - bit D5
- zda je pravý a levý panel převrácený - bit D0

Pokud je vybrán vnější obal, je obal vytvořen jen když je adresa 18 nebo 19. (To platí, pokud A0 = 1 a je zde použit povel "write 18 or 19")

Šesti panelové mixéry - výstupní uspořádání

Šest komponentů levého panelu je kombinováno v mixéru. Výstupní uspořádání mixéru obsahuje šest stejně upravených běžných průniků, které zajistí výstupy PWM, z kterého analogický výstup je zajistěn průnikem přes užší průchod. Tentýž mixér je užíván ke kombinování komponentů z pravého panelu.

Synchronizace

Ke zdjednodušení práce pisatele softwaru je několik synchronizačních funkcí zahrnuto v SAA 1099 a způsobuje:

- počátek frekvenčních generátorů
- výměnu frekvencí a oktáv
- výměnu obalů

Synchronizování frekvenčních výměn

Vnitřní skladba SAA 1099 je taková, že pokud je nový požadovaný tón v oktávě rozdílný od běžného výběru, frekvenční registr bude psán před registr oktavy. Opomenutí napsání registru je toto seřazení se zvukem umožněno, smí způsobit cvaknutí v audio výstupu. Frekvence a oktáva registrů může být jakýmkoliv způsobem psána v jakýmkoliv čase, ale data mohou být uskutečněna přes SAA 1099 na přechod ze sdruženého generátoru; to zn., že data nebudou uskutečněna do poloviny periody běžné frekvence.

To způsobuje, že:

- když jsou frekvenční a oktávové registry dány pro nejnižší frekvenci 31 Hz, nové údaje o frekvenci, nebo oktávě nesmí být

- uskutečněny pro horních asi 17 ms (1/2 periody). Proto k zajištění tohoto intervalu má přesunout odpovídající půl periodu existujícího tónu a bude napsán v software mezi napsanou novou frekvencí nebo oktávou a změněným výstupem.
- ve vyšší části spektra, pokud je to požadováno k výměně frekvenčních a oktávových registrů současně probíhajících, obě nové hodnoty musí být napsány (frekvence první,oktáva jako druhá), uvnitř půl periody běžné frekvence.

Všechny frekvenční generátory mohou být smazány umístěním bitu D1(RST) z registru do adresy 1C. V tomto stavu mohou být údaje frekvence a oktávy psány SAA 1099, ale nebudou uskutečněny. Proto tak, jak dlouho působí RST hodnota registru může být přepsána s novým údajem. Avšak jakékoli nové údaje v registru nebudou uskutečněny, až do poloviny periody frekvence jejíž hodnota je udržována v registru, když RST bylo použito. To proto, že RST pouze nepřipravuje všechny generátory ke známému stavu, synchronizuje jejich začátek.

APLIKACE

Pravděpodobně nejjasnější aplikace SAA 1099 je ve video hrách kde širší zaměření dostupných zvukových efektů může být použito k vytvoření širší dekorace a vytvoření pohybu objektů. Zajímavější je možnost popisu obou panelů amplitud a Dopplerova změna frekvence k pozici vzájemného obsahu uživatelů. Např.: zvuk pojíždějících aut, nebo přelet kosmických lodí mohou být udělány realisticky. Mnoho zvuků v počítačových hrách je založeno na "barevném" zvuku. (např. letadlo, dělostřelecká palba a automobilové motory). Dva zvukové generátory SAA 1099 s úplným ovládáním barvy zvuku a schopnosti míchání zvuku s tóny umožňuje dva samostatné "barevné" zvuky k produkci ve stereo pro zvětšení realistického pocitu.

Nejdříve zmínka: SAA 1099 může vytvořit všechny hudební noty přes 8 oktáv od 31Hz do 7,81 kHz. Použitelnost šesti frekvenčních generátorů umožňuje podkovovat plný hudební akord (včetně rozlišování výšky tónu) a dovoluje překrytí dvou akordů (mimo rozlišování výšky tónu). Moderní obal vytváření příslušenství a softwérkové ovládání amplitudy a frekvence, dává možnost napodobení hudebních nástrojů, vibrato a tremolo efektů.

Softwérkové moduly vyrábějí určité zvuky např. piano a trumpetu, právě tak jako mohou být změněny tyto moduly určitých zvuků, mohou být vytvořeny ke změně situace ve video hrách, např. laserové dělo, sirény a chybová upozornění.

Následující části naznačují, jak může být vyrobeno několik zvukových efektů. Detaily programování nejsou dány (jsou publikovány v 3 části), pouze některé význačné body a údaje.

Kosmické lodi

K vytvoření dvou létajících kosmických lodí, dvou totožných obalů dosažených v rozdílných poměrech kde každý obrácený bit je užit se dvěma tóny jejichž frekvence jsou měněny mezi přítomnými body v rozdílných poměrech.

Parní lokomotiva

K simulování zvuku parní lokomotivy jsou použity barvy zvuku (15,6 kHz) a opakování trojúhelníkových obalů. Se zvukem znemožňujícím obal je dosažen skrze jeho vrchol, tak i se zvukem umožňujícím dosáhnout panelu skrze zvuk jemný sykot páry. Po přestávce obal bude dosažen nepřetržitě (začínající např. v 3,5 registrů za sec. a stoupající ke 28 registrům za sec. k vytvoření rychlosti lokomotivy). Zvuk pískání lokomotivy může být produkován použitím tónu 480 Hz (tón HFF v oktavé 3) smíšený se zvukem.

Zvonění telefonu

Zvuk telefonního zvonění přispůsobený Trimphone je simulován použitím vnitřně dosažitelným pilovým obalem smíšeným s vhodným tónem (3,3 kHz; tón H6F v 6 oktavě). Rytmus je produkován umožněním a znemožněním zvuku (0,46 sec umožněn; 0,23 sec znemožněn).

Ohodnocení obvodu

Schema 4 ukazuje obvod, který je pro tebe k dispozici k ohodnocení SAA 1099 při jeho použití. Obvod, který dosahuje komplex zdroje stereo zesilovače, může být uspořádán užitím části čísla OM 1099. Popis obvodů hardwerů a softverů je ukázán na schematu 4 a 5 navzájem.

Údaj bajtů obsahuje EEPROM (mikroprocesor MAB 8035 HL nemá čip ROM) jsou rozeslány přes mikroprocesor ke vložení údajů z SAA 1099 spolu s AO a signály WR.

Předprogramované zvukové moduly mohou být vybrány použitím "selekt programu" (výběrovým programem) klávesou, která přeruší obvod zvukového modulu během hry. displej indikuje vybraný program. Klávesa "play" je užívána ke hře vybraného programu. Daný softwáre je k dispozici na přání. Pro načasování detailů je čtenář upozorněn předem přehledem údajů z SAA 1099 (schema 2). Standardní MAB 8035 potřebuje pásmo 1-6 MHz. Je vhodný pro použití krystalového oscilátoru 8 MHz pro SAA 1099 s děličem, k poskytnutí 4 MHz pro mikroprocesor.

Od té doby může SAA 1099 přijmout údaje spínače, potom zaměřit, ve které MAB 8035 může vytvořit obvod z obr.4. Výstupy DTACK nemohou být použity. Ačkoliv když procesor spínače, tak jako SON

68000, je použit; DTACK musí být použit k zabránění chybných údajů SAA 1099.

Při nižším průchodu pronikání (17 kHz, -3 dB) výstupy z SAA 1099 je rozšiřován dvěma TDA 1011 tvořící komplex obvodů zesilovače. Protože výstup z SAA 1099 je okolo 1,3 V pouze zdroj zesilovacích částí z TDA 1011 je použit a poskytuje okolo 23 dB zisku, který je ekvivalentní audio výstupu, asi 6W ze středu ve 4 systému s doplněním napětí na 16 V. Dokonce ačkoliv předzesilovače nejsou použity, některé komponenty (R18, R21, C11 a C21) jsou nutné k zajištění opravy spádu výstupu. Další průnik má být přidán ke snížení nějaké vysoké frekvence spojující zesilovače a SAA 1099 přes nahrazené linky SAA 1099 jako periferie domácího počítače.

Obr.6 ukazuje jak SAA 1099 může být připojen k typickému domácímu počítači - BBC Model B. Ačkoliv následující popis je obecný na toto pojednání, je vhodný pro velký počet domácích počítačů.

Audio zesilovač je TDA 7050 tvořící komplex stereo zesilovače poskytující okolo 140 mW na výstupu zdroje v 16 ohmech systému se 4,5 V zásuvkou. TDA 7050 požaduje malo vnějších komponentů a je tedy vhodný pro nejlevnější aplikace.

Operace na vstupu do SAA 1099 je zcela softwarově kontrolována. Tedy software musí připravit vhodné načasování, například pro trvání psaní a jeho relativní načasování údajů a výměnu z AD

DTACK

Protože signál DTACK je příliš rychlý k použití do domácích BBC počítačů, DTACK pin je připojen k jednomu z uživatelských portů (CB1), který je uspořádán v neprůchodném stavu (HIGH) tak, jako když DTACK je LOW, klesající proud je limitován okolo 3 mA.

Volba čipu

CS linka je uzemněna přes 22 kilohmový rezistor a proto SAA 1099 je normálně přístupný. Příkon CS je také uzemněn k CB2 na počítači a tento port, nebo ekvivalent bude držen dolů (LOW) pro normální operaci a vysoko (HIGH) k ochraně SAA 1099.

Lineární část

Provedení části audio zesilovače z obr.6 je horší než na obr. 4, ale obvod je levnější. Komplet obvod stereozesilovače TDA 7050 poskytuje 26 dB zisku po kanálu odpovídá asi 50 mW na panelu výstupu zdroje s 50 ohmovými reproduktory. Nízká frekvenční omez

va zesilovače (70 Hz; -3 dB s 50 ohmovými reproduktory), je hlavně nutná pro hodnotu spojení C4 nebo C8 (47 mikro faradů). Před rozšířením každý signál je filtrován k potlačení nízkofrekvenčního signálu (62,5kHz a více) příslušného dílu amplitudy a obalu řízení SAA 1099. Jestliže jsou použity malé reproduktory, jejich nedostatek nízko frekvenčních odpovědí zdůrazňuje vyšší harmonické tóny na výstupu z SAA 1099. Produkují spíše hrubý zvuk který může být odstraněn dalším vysokofrekvenčním oslabením audio výstupu.