

SAM Coupé - úvodní informace o softwaru

S Coupé je možné uspořádání několika pamětí s různými výhodami a nevýhodami. Způsob zobrazení a výběr paměti obrazovky jsou obvykle první faktory, které je třeba rozvážit.

Výběr paměti obrazovky

Úplná obrazovka přijímá 6,75K v modu 1 a 12K v modu 2. /Neobsahuje 2K mezeru mezi prvkem a údajem atributu/. V těchto modech pak paměť obsahující obrazovku může být zapnuta v kterékoliv 16K hranici. /Stránkovací systém má ten význam, že dalších 16K RAM stránky ve stejné polovině paměti bude stránka ve stejném čase jako stránka obsahující obrazovku/. Mody 3 a 4 požadují 24K a tudíž dvě stránky 16K. Můžeš vypustit jednu nebo druhou obrazovou paměť ve stálém stránkování nebo stránku, kterou právě chcete číst nebo psát, umožňuje obrazovce, že řídicí kód je číslován ve stejném čase. ROM zapíná na obrazovce, jak je požadováno pamětí, v sekcích C a D /8000H/

Hranice

ROM obvykle dodržuje strojovou hranici a systém proměnných v sekci B v 4000H - 5CD0H, tak že nejsou vylistovány mimo, když obrazovka, Basic program nebo oblast proměnných jsou vyčísleny v horní polovině paměti.

ROMky

ROM 0 je normálně zapnuta do sekce A mapy paměti. ROM 1 je obvykle vypnuta, dokud je to potřeba, když je použita, je umístěna v sekci D. /Skok v seznamu rutin v ROM 0, pokud bude potřeba stránka v ROM 1, bude použito automaticky a pak se obnoví jeho předchozí postavení/

Použití subrutin ROM

ROMky obvykle očekávají hranice v sekci B a mnoho rutin vyžaduje systémové proměnné v tomtéž nedotčeném prostoru. Pokud si přeješ použít tento prostor pro svůj vlastní kód, můžeš prolistovat systémové proměnné a použít ROM rutiny nepřímo přes JSVIN /uvidíš později/. Tato rutina bude přejímat systémové proměnné zpět a v krátké chvíli je ROM

rutina provedena podle tvého výběru.Obrazovka bude dočasně přepnuta do sekce C a D pokud to požadujete,ale Váš kód obvykle může být umístěn ve stejném místě.Přirozeně,ROM 0 musí být umožněno,pokud chcete použít skokový seznam,ale budete stránkovat delší dobu.

Například budete vkládat Váš strojový kód,zbaven schopnosti přerušeni,vypíná ROM 0,zapíná novou RAM do sekce A/B a používá to pro Váš program./Připravuje jakýkoliv kód k výměně odhalených přerušeni,pokud si to přejete a neodhaluje přerušeni/.Před použitím rutiny ROM učiní schopmé ROM 0 kdesi mimo sekci A a použití JSVIN k vyvolání rutiny kterou potřebuješ.

JFARLDDR LDDR verze z FARLDIR

JPUT Pouze obrazový mód 3 a 4.Prostor údajů z obrazovky k získání koordinátů /koordinát x musí být sudý/.Druhý a třetí bajt údajů určuje šířku a výšku v bodech,následující všechny bajty jsou horní uvažovaný blok,dále následují všechny bajty z druhého atd.Blok musí mít sudý počet bodů na šířku,ale mohou být jakkoliv dlouhé.Na vstupu A registru určuje metodu umístění údajů na obrazovku.

0 = nadpisy s INVERSE

1 = XOR

2 = OR

3 = AND

4 = nadpisy bez INVERSE

5 = maska

Volba masky požaduje druhý blok údajů stejné délky jako v prvním případě,s bity použitými pro "odhalení" a opětné vložení pro "maskování".To dovoluje úplný tvar možná s chybami v něm,k umístění na pozadí bez efektů okolí.Na rozdíl od atributu systémů barvy,barvy k použití jsou zahrnuty v bloku údajů,ačkoliv metody 0 - 3 dovolují volbu INVERSE.Omezení těchto bloků musí být hranicí sudých bodů,slouží k udržení vysoké rychlosti.Pokud si přejete prostor bloků v 1 - bodových intervalech,je to možné,k přípravě 2 verzí grafiky s údaji přemístěním 1 bodu v jednom z nich,k uspořádání pro vytvoření požadovaného efektu.

JGRAB

Používá se pouze pro obrazovku v modu 3 a 4. Uložení bloku údajů na obrazovce k získání koordinátů buferu.

D = délka

E = šířka /v bodech/

L,H = koordináty x a y horního levého rohu

BC = adresa buferu

JPLOT

Zakreslení bodu C,B pokud je vstup NC. Jinak zakreslení HL,B používá "tenký" bod. /Možný v modu 3 v 512x192 bodů/

JDRAW

Vytažení řady z běžné pozice C /nebo HL pokud jsou použity "tenké" body/ bodů vodorovně a B bodů svisle.

pro D = 1 nahoru /nebo pro zastavení pohybu Y / a FF dolů,

E = 1 doprava /nebo pro zastavení pohybu X/ a FF pro doleva

JDRAWTO

Vytažení řady z běžné pozice k bodu C,B /nebo bod HL,B pokud jsou použity "tenké" body/

JCIRCLE

Vytažení kruhu v C,B o poloměru A. Pokud jsou použity "tenké" body HL bude obsahovat výstupy z levého okraje obrazovky / v bajtech to je 4 pixly/. To dovoluje kruh umístit kamkoliv na obrazovku v modu 3, koordinát x je limitován 1 bajtem.

JFILL

Pracuje pouze v modech 3 a 4. Používá modelu 16x16 bodů v HL ve spodní polovině paměti k vyplnění plochy v C,B jestliže NC nebo v HL,B jestliže CY.

Jestliže A = 0, pak je vytvořen předběžný přenos ke kontrole obrazovky, později vyplňuje tutéž obrazovku přes tutéž barvu může zhotovit A NZ pro větší rychlost. Model údajů má 8 bajtů pro hlavní řadu, 8 bajtů pro další atd. Tak je potřeba 128 bajtů. Každý ubírá řízení barvy pro jeden bod v modu 4 nebo dva body v modu 3.

JBLITZ

Provádí seřazení grafických povelů délky BC v DE. Povelů PLOT, DRAW, DRAW TO, CIRCLE, OVER, INK, CLS a PAUSE mohou být obsaženy v řadě. Jsou ovládány ve vyšší rovině než JPLOT, JDRAW a JCIRCLE, měnivý bodový počítač a speciální proměnné BASIC mohou být použity pro počátek a plátkování. Pohyb koordinátů je dovolen pouze v rozsahu 0 - 255

Relativní DRAW

00 nebo FF udělí SGN pohyb po ose X sledující pohyb x, sledován SGN pohybu po ose Y a pohyb Y.

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

JROLL

Pracuje pouze v modech 3 a 4. Pohybuje částí obrazovky vlevo, vpravo, nahoru nebo dolů. Plocha obrazovky musí být stejná jako je počet bodů. Vodorovný pohyb jednoho bodu nebo jakéhokoliv počtu stejných bodů je možný. Svislý pohyb jakéhokoliv počtu bodů je možný. Úplné svinutí je nezávazné; to používá rozsah bezprostředně po obrazovce paměti jako bufer.

JCLS

Pokud A = 0, čistí celou obrazovku, jinak čistí nejhornější obrazovku.

JCLSL

čistí nižší obrazovku

JPALET

Pokud A = FFH, umístí BC barvy do PALTAB jako hlavní a třetí barvy pro paletu barvy E. /udělej B = C, aby se netřpily/. Barvy mohou být 0 - 127, vstup z palety 0 - 15.

Pokud je A různé od 0, je koordinát Y ve změně palety. PAL-TAB není změněn, místo toho je vstup udělán v řádku přerušení seznamu barev.

JOPSCR

OPEN SCREEN C v modu B. Společná /více programům/ obrazovka je otevřena pokud A = 1, nebo rezervována obrazovka pokud A = 0. Požadované stránky jsou umístěny v PAGE ALLOCATION / technický manuál // seznam přidělení stránky a určení/. Číslo obrazovky musí být 2 - 15

JCLSCR

CLOSE SCREEN C. Rozvržení stránek obrazovky.

JSCRN

Je zvolena SCREEN C /obrazovka C/. Každá obrazovka má svoji vlastní tiskovou pozici, barvy, grafickou pozici a paletu. Výběr obrazovky bude vidět, jestliže proměnná CURDISP je 0; pokud není 0, obrazovky to specifikuje, že bude viditelná vždy když pozměníte běžnou obrazovku /ale produkt na obrazovce stále běží jakoukoliv obrazovkou výběru JSCRN/

JMODE

Umístění obrazovky modu A / 0 = mod 1 3 = mod 4 /

JTDUMP

Udělejte text složený z obrazovky k toku 3 /který bude odchylen k další řadě/. Pracujte v kterémkoliv modu obrazovky. Znaky 32 - 168 jsou známy. Přejmenování znaků lze provést.

JGDUMP

Udělejte grafiku složenou z obrazovky k tisku přes kanál "B". Levý horní roh tohoto obrázku a jeho šířka, výška a x-ové a y-ové zvětšení může být použito ke změně systémových proměnných. Sledy řídicích kódů jsou posílány k tisku před složením, před každým řádkem údaje bitu a po složení. Rozdílné sledy jsou vhodné pro kompatibilní tiskárnu EPSON, ale jsou vzaty ze systémových proměnných a mohou být pozměněny. Každý řádek údaje bitu je upřednostněn LSB než MSB bajty/řádek.

JKBFLUSH

Bohatý bufer klávesnice

JTSTAK

Vrací Z, pokud není klávesa ihned stisknuta, jinak NZ.

JREADKEY

Čte klávesnici, bohatý bufer /podobný INKEY\$/ .Z/NC pokud není klávesa stisknuta, jinak NZ/CY a A = hodnota klávesy /ASCII/

JGETKEY

Čte další klávesu do A přes klávesnici buferu. Z, pokud nebyla jedna. /a A = 0/

JBEEP

Vytvoří BEEP dlouhý DE - 1 cyklů v periodě HLx8 T stavů.

JSAVE

Nahraje blok dlouhý CDE bajtů začínající v HL /přepínán do sekce C/. Vytvořte regostr A, kde je vložen první bajt /FFH pro "data blok", 01 pro hlavičku/

JLOAD

LOAD bajtů CDE k HL /přepínán do sekce C/. A = typy /FFH = blok dat, 01 = hlavička/

JEDGE2

Čekej 2 přechody signálu na EAR, vracejí v A číslo 47-T bere postavení jednotek. Vrací Z, pokud čeká příliš dlouho /256 jednotek/, NC pokud je stisknut ESC, CY pokud je vše O.K.

Rozměr pásku

Zde jsou 4 typy možností; BASIC program, číselná řada, pole řetězců a SCREEN\$. Řádek poskytuje schopnost zvolit rychlost která může být použita jako protějšek rychlostí pásku Spectrum. Hlavní formát je rozdílný od Spectra a neuznává Spectrum jako hlavní. Blok dat je jiný než u Spectra, ale může ho obsahovat 512K. Řádky SCREEN\$ obvykle obsahují paletu nastavení /32 bajtů dovoluje 16 svítících barev/ a řádek přerušeni seznamu stejně jako obrazovka zapíná paletu částečně dolů, obrazovka může být znovu obnovena. Obrazový mód je vhodně uspořádán před vložením bloku dat.

Řádky jsou vloženy automaticky ve správné rychlosti. CODE řádek Spectra je převoditelný do SAMa a pak použit k vložení bloku dat.

Řádky SCREEN\$ Spectra mohou být vloženy jako řádky SAM CODE v modu 1 ke specifikování adresy obrazovky při povělu LOAD. /SAM rozlišuje řádky SCREEN\$ od řádků CODE, protože musí být vloženy do paměti obrazovky, které mohou být někde v paměti/. Řádky CODE mohou mít adresu provedení. Změnu důležitých systémových proměnných může přerušit rychlé použití před SAVE a nebo tištění jmen řádku, pak jsou čteny z pásku.

Strojový kód

SAVE "jméno" CODE začátek, délka, adresa provedení
Provádí samospuštění řádku, pokud je použita adresa provedení.

LOAD "jméno" CODE začátek, délka, adresa provedení. Vykoná řádek z adresy provedení, pokud je použita. Jinak je použita jakákoliv adresa provedení zachovaná s řádkem. Merge bude zastaveno řádek CODE vykonán ledaže to je chráněno při umístění bitu v hlavním řádku.

Počátek, délka a prováděcí hodnoty a adresy použití USR, USR\$, CALL, POKE, PEEK atd. mohou být do 512K minus 1

USR_n

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

USR\$_n

Jako USR, ale vrací string BC bajtů uvedení do chodu v DE

na straně A. /Stránka je nepodstatná, pokud DE < 8000H/

CALL_n

Volá adresu n přes seznam čísel nebo řetězec parametrů, pokud si přežete. Hodnoty čísel, délek a počátků řetězců jsou umístěny na nestálém bodě hranice počítače s informací na typu. Regisr A obsahuje číslo parametrů, když je vložena adresa n. Použití hranice FPC, dovoluje číslům, aby byla odstraněna a zaokrouhlena na 2 bajty s řadou zastavení voláním GETINT v seznamu skoků, čísla nestálého bodu nebo řetězců mohou být rozmístěny použitím STKFETCH. Řetězce kratší než 256 bajtů mohou být překlenuty k buferu o rozsahu 4000 - 7FFFH k vyvolání STRTOHL. /Řetězce mohou být překlenuty z odkudkoliv v paměti, stejně tak pokud jsou stránkovány mimo/

CLEAR_n

Umístění vyšší hranice použití RAM Basic

OPEN_n

Vyhrazení n 16K stránek při použití Basic

CLOSE_n

n volných stránek 16K

LENGTH

Tato funkce vrací adresu a rozsahy řad a řetězců a adresu čísel. Užitečný pro přechod dat cestou CALL.

POKE n,x - POKE 1 bajt

DPOKE n,x- POKE 2 bajty /LSB/MSB/

TPOKE n,x- POKE 3 bajty /stránka,offset+8000H/

POKE n,a\$- POKE úplný řetězec

PEEK n - PEEK 1 bajt

DPEEK n - PEEK 2 bajty

TPEEK n - PEEK 3 bajty /stránka,offset+8000H/

MEM\$(n TO m)- PEEK řetězec

Např: LET A\$ = MEM\$(70 000 TO 71 000) přiděluje 1000 bajtů
A\$
POKE 72000,A\$ klade tyto bajty do 72000

Podklad příkladů přeměny

BIN 1010 = 10
BIN\$ 128 = "10000000"
HEX\$ 74 = "4A"
HEX\$ 16384 = "4000"
hex\$ 65539 = "010003"
%FF = 255
& 10003 = 65539
VAL("&" + a\$) = 255 jestli-že a\$ = "ff"

Přepínání palety

ROM přerušuje rutinu umístěním horní palety registrů(CLUT) na začátku každé stovky, používá jeden ze dvou seznamů 16 bajtů v PALTAB a přepíná mezi nimi, pokud chceme třpitivé barvy. Je to dobrá možnost pro zachování těchto seznamů s běžnými barvami, stejně pokud nechcete použít subrutiny ROM, protože registry palety nemohou být čteny.

Software vstupující přes NMI bude opět přesně získán použitím užitečného PALTAB, když vytvořenou obrazovku uloží, nebo paletu střídá.

ROM také ovládá přepínání palety po částech dolů obrazovky v nějaké uvažované řádce. Horních 127 změn může být uděláno po zarámování, ale pokud jsou udělány více než 2 po uvažované řádce, nesmí se stát podstatou okolí, snad způsobuje "jags" / zuby/. Barva okolí může být vyměněna s jakýmkoliv "jags" pokud je to první výměna stanovena v uvažované řádce./Paleta bude změněna během běhu řádky nazpátek/. Pokud si přejete opatřit Vaši vlastní rutinu, to je snadné připojením použití přerušování řádku řídicí adresy /LINIV/ nebo přepnutím mimo ROM.

System stránkování

System stránkování umožňuje v jakékoliv adrese, že jsou v nejméně 16K bytech před FFFFH vynechány. To znamená že když rozdělujeme složení dat kratší než 16K, počátek dat je strán-

kován tam, kde není potřeba speciální opatrnosti k pojištění. Ukazatele nezmenší konec paměti. ROM obvykle používá paměť sekcí C a D jako otáčivé okno až k paměti a chrání systémové proměnné a hranici v sekci B.

Například, když další Basic řádek je vhodný k připojení čísla k HL /připojuje běžný řádek na počátek běžného řádku/ kódu, podobně níže zabraňuje sledování pokud se HL má pohybovat ze sekce C do sekce D. Jestliže tímto způsobem hodnota stránky vzrůstá a HL je přesto přispůsoben určení současného bajtu na adrese 4000H níže, v sekci C. Celý řádek pak může být jednoduše pěkně rozdělen poněvadž kromě toho nezastaví, HL potřebuje být udělán dokud je další řádek sledován.

BIT 6,H

JR Z,LAB1

IN A,/HMPAGE/

INC A

OUT /HMPAGE/

RES 6,H

LAB1:atd.

Zde není nebezpečí 5 nižších bajtů z HMPAGE, narůstá po 1 FH k narušení nejvyšších bitů, poněvadž minulý řádek programu má zakončovat. To bude proces s nejvyšším jiným počtem dat, jsou vhodné k použití, pokud chcete/

Vektory

Dovolují uživateli přerušit důležité rutiny ROM. Obsahují adresy, které budou volány, jestliže nejvýznamější bajt není nulový. Měly by být pozměněny přímo, ale volány JPVECTOR / ve skokovém seznamu /.

A = vektor číslo a BC = adresa volání.

Tento mechanismus dovoluje více rutinám připojení jednoduchého vektoru./nejnovější připojení rutiny je vždy vyvoláno jako první/.

Částečný seznam vektorů

RST08V volá chybovou manipulaci, A = chybný kód
RST28V volá letmý bod počítače s A = další kód v seznamu, dovoluje instrukci RST 28H
RST30V volá RST 30H/nepoužívá Basic; ROM = RET/
FRAMIV volá přerušeni konstrukce 50 Hz
LINIV volí přerušeni řádku
MOUSV volá přerušeni myši
MIDI volá přerušeni MIDI
EDVECT volá editor

SETIYV

Je volán když DRAW, PLOT a CIRCLE "rozhodnout se" jaký bod použije rutina a samo volá SETIYV jestli že jeho nejvýznamější bajt není roven nule.

SETIY body IY registru vhodné rutiny a jenom A a HL budou změněny jinak.

V modu 4 a v modu 3, když jsou vybrány "tlusté" body, hodnota A na výstupu přejde na D' v grafických příkazech a může být použito k přechodu barvy PEN, pokud si to přejete. Jestliže změňte SETIYV, jako všechny vektory, udělá-li Vaše rutina RET způsobí provedení normální ROM rutiny. Takto Váš kód může přerušit mód, který vidíte, jestliže je to ten, který chcete přidělit ROM a RET takto. Jinak POP vrací adresu tak, že ROM rutina není použita a umístíte Vaše IY. Další RET bude DRAW, PLOT nebo CIRCLE.

Bod umístění rutiny určíte adresou v IY, zmapujete bod v L, H a vrací pouze výměnou AF.

System koordinátů užívá 0,0 v levém horním rohu obrazovky. Můžete si všimnout běžných příkazů OVER, INVERSE a PEN. Rutina bude končit v JP /IX/, nikoliv v RET.

POSTOKV

Posice symboliky. Volá, potom řádek Basic má být symbolický s HL ukázán na počátečním řádku. Schválení návratu končí řádek

CMDCDV

Volá A-čkem registr obsahující kódokolního znaku k seřazení přerušeni nebo provedení /Obvykle A obsahuje povelový kód/.Může být použito k doplnění povelů.

EVALUV

Provádí vyjádření zhodnocení situace.Volá s A = běžný znak ve výrazu,může být použito k doplnění funkcí.

PRTOKV

Tiskne doklad.A = znak k rozšíření a tisku jako ASCII

Umístění znaku

Daný znak z CHR\$ 32 - 168 je uskladněn ve zhuštěné formě v ROM 1 a vložen do RAM v sekci B,když je počítač zapnut.CHR\$ 128 - 168 jsou nejprve definovány jako neznámé znaky ačkoliv CHR\$ 128 - 143 jsou ukázány jako grafický blok,pokud není " BLOCKS 0 ",měly být dány změnou systémových proměnných./Upravený blok grafiky je generován jako potřebný - toto umístění znaku není v něm obsaženo/.Znaky stranou grafického bloku zabírají matrix 6x8,vhodný pro ukázání v intervalech 6 nebo 8 bodů.Znak vymezuje prostor 8 bajtů z horní řady bodů k dolnímu řádku.Levý a pravý sloupec bodů tohoto bloku 8x8 nejsou použity pokud je tisk v intervalu 6 bodů.

Zdroje ROM

Tato část dokumentace nedovoluje dosti detailně použití všech rutin,ale dostanete dobrý návod dostuoných zdrojů.ROM bude použita přez prokázání stálých poloh nebo skoků v seznamu k vyhnutí se problémům s výstupy nové ROM.

0000 - počátek stroje

0004 - POP HL Volázde k návratu vaší vlastní adresy v HL

0005 - JP/HL/

0006 - JP/IY/

0008 - chybná obsluha

000F - Verse čísla ROM

- 0010 - Tisk v "A" registru běžného sledu. Kanály K'a 'S' reagují na kontrolní kódy PEN, PAPER, OVER, INVERSE, FLASH, DELETE, doleva, doprava, nahoru, dolů, AT, TAB a prosazení návratu
 - 0013 - Tisk BC bajtů začíná z DE použitím běžného sledu.
 - 0016 - JP/BC/
 - 0018 - Dostává běžný znak řádku Basic
 - 0020 - Dostává další znak řádku Basic
 - 0028 - Volá pohyblivý bod počítače. Příklad násobení nejvyšších dvou čísel pohyblivého bodu hranice počítače, potom dělí výsledek deseti a vyměňuje nejvyšší a třetí posici
- | | |
|-----------|-------------|
| 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 | |

Používané operační kódy ze Spectra jsou rozdílné a četnější. Adresa vektoru umožňuje překlad jakéhokoliv operačního kódu do jiného, než je operace provedena.

- 002B - DJNZ 002B Užitečný, pokud nebojujeme s časem
 - 002D - JP/IX/
 - 0030 - USER RST. Vhodný pro odstranění chyb atd. JP k adresování v RST30V pokud je různé od nuly.
 - 0033 - JP/DE/
 - 0035 -
 - 0038 - přerušuje mód 1
 - 0066 - přerušuje neschopnost maskování
 - 018E - CLAHL. Volá rutinu na straně A, doplněk HL
 - 0190 - FARCL. Volá rutinu na straně/doplněk sledující CALL
- | |
|---------------------|
| CALL FARCL |
| DB stránka |
| DW doplněk |
| / vypnutí programu/ |

01AF - FARREAD.Čte bajt v 19-bitové adresy v AHL do C
01BA - FARPGRD.Čte bajt strany A,doplňk HL /8000 - BF
FF / k C
01C9 - FARWRITE.Píše C 19 - bitové adresy v AHL
01D4 - FARPGWR.Píše C straně A,doplňk HL / 8000 - BFFF /
01E4 - PC RELATIVE CALL.Nahrazení nabídky CALL.Příklad
níže může být umístěn kdekoliv bez potřeby přes-
kupení,ačkoliv používá subrutinu v LABEL 2.Všech-
ny registry na vstupu vyjma IX jsou chráněny.

CALL PCRELAT

LABEL1:DW LABEL2 - LABEL1
LD A,B
RET

LABEL2:LD B,99
RET

RAM tvrzení znamená,že ROM rutiny obvykle běží nepatrně rychleji než stejné rutiny RAM.Jestli-že uskutečníte za-
blokování instrukcí Z80 ve větších blocích a rychlost je
důležitá,vyplatí se volat rutiny viz níže:

01F4 LDIR:RET
01F7 LDDR:RET /blok se v ROM pohybuje rychlostí o 8%
vyšší/
01FA CPIR:RET
01FD CPDR:RET

JUMP TABLE AT 0200H /Seznam skoků na 0200H/

Občas potřebujeme podmínky pro vstup několika málo bajtů
,poslední začátek rutiny/kde je to uděláno/ můžete číst
adresu rutiny ze seznamu skoků,připojit posun a vstou-
pit částečně skrz / užitím např.CALL HL JUMP /.
Prokázání vstupní podmínky zůstane konstantní.

JSWIN

Volá parametr slova zapojených systémových proměnných. Návrat původních stránek v sekci A/B na výstupy. Prochází hlavní registry a volá rutinu.

Příklad:

```
CALL JSWIN
DW JGETINT
LD /STORE/,HL atd.
```

- | | |
|-----------|--|
| JPVECTOR | Připojení adresy vektoru / viz.VECTORS / |
| JHEAPROOM | Uchování BC bajtů v paměti plochy B /dlouhá doba /.Vystupuje s DE a ukazuje prostor. |
| JWKROOM | Otevírá BC bajty v pracovním prostoru /dočasně /.Vystupuje s DE a ukazuje místo začínajícího stránkování v sekci C |
| JCALLBAS | Volá subrutinu Basic v řádku HL.Když je použit RETURN,pak strojový kód bude znovu zaveden. |
| JSETSTRM | Umisťuje tok specifikovaný v registru A /FBH -
- 10H /. |
| | FBH = kanál "B" vždy tiskne dvojitou hodnotu |
| | FCH = kanál "\$" vždy hodnota stringu |
| | FDH = kanál "K" vždy Nižší obrazovka I/O |
| | FEH = kanál "S" vždy vyšší obrazovka I/O |
| | FFH = kanál "R" vždy Výstupní hodnota k sestavení řádku |
| | 00H = kanál "K" obvykle |
| | 01H = kanál "K" obvykle |
| | 02H = kanál "S" obvykle |
| | 03H = kanál "P" obvykle - výstup tisku textu |
| JPOMSG | Výstupní hodnota zprávy číslo A ze seznamu v DE.Nula dává první zprávu.Seznam tačne s modelem 7-mi bitového znaku;každá zpráva bude mít 7 bitů na posledním znaku. |

JEXPT1NUM	Skladba přezkoušení/zhodnocení číselného výrazu
JEXPTSTR	Skladba přezkoušení/zhodnocení řetězcového výrazu
JEXPTEXPR	Skladba přezkoušení/zhodnocení výrazu. Tyto tři rutiny mohou být užitečné při rozšiřování Basicu.
JGETINT	Vezmi číslo z hranice počítače do HL/ a BC /. Chybná hodnota, pokud není v rozsahu 0 - 65535
JSTKFETCH	Vezmi poslední hodnotu z hranice počítače k AEDCB. Pokud je bod nepohyblivý, 5 bajtů je ve struktuře CPC/Spectrum. Jestli-že řetězec A = stránka počátku textu, DE = doplněk, BC = délka
JSTKSTORE	Hranice AEDCB na hranici počítače
JSTRTOHL	Vezměte řetězec z hranice počítače a opište z odkudkoliv do buferu v HL /4000 - 7FFF /, jestli-že délka $\zeta = 256$; jinak udává chybné hlášení
JFARLDIR	Opisuje bajty ze strany A, doplněk HL na straně C, doplněk DE. Opisuje / PAGCOUNT / 16K stránky a / MODCOUNT / bajty / MODCOUNT = 0000 - 3FFF /, listuje jak je třeba. Listování stran na vstupu není důležité a je nezměnný na výstupu.

Jestli-že změníte stránku, váš kód je průběžně realizován, pamatuj, že další instrukce přijde z nové stránky!

Použití obrazových módů 3 a 4

Jednoduchá úprava paměti obrazovky v módech 3 a 4 provádí zmapování bodu velmi jednoduchým. Podobná metoda tohoto použití ROM módu 4 / 256 x 192 , 16 barev / je ukázána níže.

Na vstupu, L a H jsou koordináty X a Y / 0,0 vlevo nahore / a D má barvu k použití v obou jeho kouscích. /Např: D = 77H pro barvu 7 /.H a L se vracejí nezměněny./Vhodné například během kreslení čáry /

```
PLOTLH:      SCF
              RR H
              RR L
              LD A,/HL/
              JR C,PLOTODD ; jr pokud je použit lichý bod

              XOR D
              AND OFH
              XOR D      ; mísí levý kousek D s pravým kouskem
                        ?;čte z obrazovky

              LD /HL/,A
              ADD HL,HL  ; obnovuje vstup HL
              RET

PLOTODD:     XOR D
              AND OF0H
              XOR D
              LD /HL/,A
              ADD HL,HL
              INC L      ; obnovuje vstupy HL
              RET
```

Adresa dalšího zkoumaného řádku může být vždy získána přidáním 128 k běžné adrese obrazovky.

Obrazovka se snadněji pohybuje doleva nebo doprava po 1 bodu použitím instrukcí RLD a RRD, což Z80 umožňuje. LDIR a LDDR může být použit pro 2,4,6 atd. bodů. Subrutiny ROM poskytují nejvíce možností.