

M R S

MEMORY RESIDENT SYSTEM
Verzia pre ZX-SPECTRUM
Užívateľská príručka MRS

Autor: RNDr. Ivan Jedlička
Ústav aplikovanej kyber-
netiky
Hlavná 5/A
844 16 Bratislava

Rozširuje: T - klub
Vazovova č. 1
812 19 Bratislava
/Klub vedecko-technic-
kej činnosti mládeže/

MOTTO:

Operetu Broso českého skladateľa Jary Cimrmana hodnotil
nemecký muzikolog prof. dr. Kurt Gupps slovami - "Dort ist
alles". Zaujímavé je, že rovnaké slovo vyrieckol prof. Niklaus
Wirth keď prečítať túto príručku.

Kapitola 1

Úvod

MRS /Memory Resident System/ je programový prostriedok na vývoj programového vybavenia v zostavovacom jazyku mikroprocesora ZILOG 80. Poskytuje všetky bežné funkcie bez toho, aby využíval vonkajšie zariadenie typu PRUŽNÝ DISK, a preto je obzvlášt vhodný na príču na malých osobných počítačoch. Jeho použitie je však opodstatnené i na výkonnejších systémoch, pretože práca s ním je veľmi pohodlná a rýchla.

Systém sa skladá z nasledujúcich modulov:

- riadiaci modul **/MRS/** - zabezpečuje komunikáciu užívateľa so systémom a umožnuje mu prístup k ostatným časťiam systému
- EDITOR **/EDI/** - je určený na písanie a opravovanie zdrojových programov v zostavovacom jazyku mikroprocesora Z80
- zostavovací program **/LNK/** - vytvára z binárnych modulov vykonateľný program
- program na obsluhu knižnice binárnych modulov **/LIB/** poskytuje niektoré funkcie nutné pri využívaní knižnice
- ladiaci program **/DBG/** - slúži na ladenie programov
- **zpětný prekladač /DIS/** - vytvára z binárneho kódu zdrojový text, ktorý je možné ďalšími modulmi spracovať

Systém je napísaný v zostavovacom jazyku mikroprocesora ZILOG 80 a realizovaný na osobnom počítači MO8X. Prinávrhu bol však kladený veľký dôraz na prenositeľnosť systému do iného technického prostredia a už bol reálizovaný prenos na domáci počítač ZX SPECTRUM.

Ďalší popis systému sa týka tejto konkrétnej implementácie.

Kapitola 2

Popis práce s jednotlivými modulmi

2.1 Spustenie systému

Súčasná verzia systému sa dodáva na kazete a skladá sa z dvoch častí, ktoré sú na páske za sebou v tomto poradí

vlastný systém
základný modul binárnej knižnice - SYSMOD

Oba moduly sa nahrájú pomocou krátkeho programu v BASICu príkazom

LOAD "MRS"

a riadenie sa odovzdá riadiacemu modulu.

Umiestnenie modulov v pamäti Spectra je premenlivé podľa verzie, ktorú užívateľ používa. Distribuujú sa tri verzie s označením MRS.V01, MRS.V02 a MRS.V03. Líšia sa umiestnením knižnice, ktorá je od adresy #6000, #8000 a #D000. Knižnice nie sú zaseniteľné. V zavádzacom module je ako poznámka vzniku, autor udržuje len verziu, ktorá vznikla 8. 10. 1986.

Binárna knižica /SYSMOD/ # 6000 --
Systém MR; #D500-#FF00

Iriestor od adresy #D500 smerom dole sa využíva na zdrojový text užívateľa. **STACK POINTER** je pri spustení systému inicializovaný na hodnotu #FF40. Teda ak užívateľovi v jeho vlastných programoch nesledí 64 BYIOV /#FF00-FF40/ na SLACK, musí si inicializovať STACK POINTER sam. V skutočnosti je to o niečo viac, pretože MRS nesieba až po adresu #FFC0.

Startovacia adresa systému MRS je uložená v zavádzacom programe "MRS", po nahráti oboch modulov je možné systém spustiť od tejto adresy príkazom USR. V tomto prípade sa inicializuje paráter a pôvodný užívateľský text sa vymaže. Ak systém spustíme od tejto adresy+3, tak sa obsah pamäte nemení **/WARM START/**.

Na požiadanie môžu byť vytvorené i verzie umiestnené v pamäti na iných adresách ale len v oddelených prípadoch.

Na adrese #FEO0 je pracovný BUFFER systému, ktorý používa aj užívateľ a v ďalšom teste sa odvolávky na premennú BUFFER týkajú tejto adresy.

2.2 Riadiaci modul

Riadiaci modul sa obliží výpisom:

MRS>

do posledného riadku obrazovky. Tento riadok je rezervovaný pre styk užívateľa so systémom a v ďalšom teste sa nazýva DIALOGOVÝ RIADOK. Užívateľ piše príkazy do dialógového riadku bežným spôsobom, pričom na opravu preklepov využíva tieto špeciálne klávesy:

CAPS SHIFT/5 - si ťeobí presun jazdca na predchádzajúci zrak

- CAPS SHIFT/8** — spôsobí presun jazdca na nasledujúci znak
CAPS SHIFT/9 — vloží medzeru na pozícii označenú jazdcom
CAPS SHIFT/0 — vymaže znak označený jazdcom a zbytok tex-
tu posunie doprava
ENTER — vymaže znak označený jazdcom a zbytok tex-
tu posunie dolava. Ak je jazdec za posled-
ným znakom riadku, tak sa posunie dolava
— ukončí zadanie textu

Všetky znaky, ktoré systém MRS akymkolvek spôsobom spracúva, musia byť napísané **malými písmenami**. Systém však pozná aj veľké písmená, ktoré užívateľ môže použiť na ríkľad na poznamky alebo ako textové konštanty /pozri modul EDI/.

Fri písaní textu sa sledujú hranice riadku a pokus o ich prekročenie, ako aj stlačenie nedefinovanej špeciálnej klávesy oznamí zvukovým signálom.

Dialogový riadok má kapacitu 64 znakov, po napísaní viac ako 32 znakov sa text posúva vľavo a prvé znaky niznú z obrazovky.

Modul MRS rozoznáva tieto príkazy:

- EPI** — odovzdá riadenie editoru
ASM — odovzdá riadenie zostavovaciemu programu
LNK — odovzdá riadenie spôjovaciemu programu
LIE — odovzdá riadenie programu na obsluhu knižnice
DBG — odovzdá riadenie cladiacemu programu
MON — odovzdá riadenie BASICu počítača Spectrum
ALD — špeciálny príkaz, ktorý postupne vyvolá moduly ASM, LIE a DBC. Pozri popis príkazov modulu EDI.
RUN — spustí užívateľský program od adresy špecifikovanej príkazom ENT modulu LIB /pozri opis príkazov tohto modulu/

Ostatné texty sa považujú za chybne príkazy. Užívateľ je na chybu upozornený zvukovým signálom, jazdec sa nastaví na prvy znak príkazu a chybu je možné opraviť. Rovnako akcia sa vykoná, aj keď za správnym príkazom nasledujú nesprávne parametre /pozri popis jednotlivých príkazov/.

Program DIS je považovaný za zvláštny program na vytváranie zdrojových textov a ako taký je volatelný len z modulu EDI.

2.3 EDITOR

2.3.1 Příkazový režim

Editor umožňuje riadu v dvoch režimoch: příkazovom a obrazovkovom. Po vyvolení je v režime příkazovom, ktorý je indikovaný výpisom

EDI

do dialogového riadku. Možné príkazy editora sú:

INI - inicializuje pracovnú oblasť editora a pripraví ju na vstup nového textu. Obrazovka sa vymaže, jazdec sa nastaví na prvú pozíciu a editor prejde do obrazovkového režimu.

LN+

LN-

LN= - posúva ukazovateľ po teste, a sice LN-N posunie ukazovateľ o N riadkov smerom k začiatku textu, LN+Nčo N, riadkov smerom ku koncu textu a príkaz LN=N nastaví ukazovateľ na N-tý riadok. Príkaz LN-O nastaví ukazovateľ na prvý riadok, príkaz LN+O za posledný riadok a príkaz LN=O na naposledy opravovaný riadok. Príkaz LN= povoľuje ako ďalší parameter tiež retazec znakov. Ukazovateľ sa nastaví na prvý riadok, ktorý v pozname navestia obsahuje dany text. Ak taký riadok neexistuje, považuje sa to za chybu a užívateľ môže text opraviť. Vykonaním príkazu sa vypíšu na obrazovku riadky od vyhľadaného po koniec obrazovky alebo textu, jazdec sa nastaví na prvú pozíciu a editor prejde do obrazovkového režimu /pozri ďalej/.

DLB - za príkazom nasledujú dve decimálne čísla v tvare N-M. Príkaz vymaže riadky s poradovými číslami v intervale N-M včítane a editor očakáva ďalší príkaz.

CFB - za príkazom nasledujú dve decimálne čísla v tvare N-M. Príkaz skopíruje riadky s poradovými číslami N-M pred riadok, na ktorom bol jazdec pri prechode do dialogového režimu. Ak je tento riadok v intervale N-M považuje sa to za chybu. /Pozri tiež obrazovkový režim/.

SAV - slúži na uloženie zdrojového textu na magnetofón. Súčasná verzia systému využíva na obsluhu magnetofónu standarčný modul monitora SPECTRUM, ale poskytuje kne- standarčné prepojenie. Za príkazom SAV ako ďalší parameter nasleduje meno súboru. Od príkazu SAV musí byť oddelený jednou medzerou. Meno súboru je ľubovoľný retazec maximálne 10 znakov, ktorý sa na túto dĺžku v prípade potreby doplní medzerami. Ak meno súboru chýba považuje sa to za chybu. Zaznie zvukový signál a užívateľ môže chýbný príkaz opraviť.

Editor sám určí rozsah pamäte, ktorý je potrebné uložiť na pásku a vyvolá príslušný modul monitora. Zdrojový text je uložený so štandardnou hľavičkou, ktorá v posledných 2 BYTOCH má zapísané 1, podľa čoho editor rozpoznáva zdrojové texty.

Pred odoslaním príkazu klávesom ENTER je nutné zapnúť magnetofón na nahrávanie. Po nahratí textu /alebo po prerušení nahrávania klávesom BREAK/ sa opäť prihlási editor.

MER - slúži na nahranie zdrojového textu z magnetofónu. Opäť sa na vlastnú prácu využíva modul monitora. Za príkazom MER nasleduje meno požadovaného textu, ktoré tvorí retazec maximálne 10 znakov od príkazu MER oddelený me-

dzerou. Ak meno chýba, nahrá sa prvý modul, ktorý editor na páske nájde. Editor rozoznáva vlastné moduly a prípadný modul s rovnakým menom, ktorý ale nevytvoril editor, sa ignoruje.

Po odoslaní príkazu užívateľ zapne magnetofón a editor vyhľadá na páske program s daným menom. Počas vyhľadávania vypisuje editor do dialogového riadku informácie o prečítaných moduloch v tvare

MENO

kde MENO je meno najdeného modulu. Ak sa na páske nájde požadovaný modul, za jeho meno sa do dialogového riadku vypíše text

LOADING

a modul sa nahráva do pamäti. Vyhľadávanie a nahrávanie modulu je možné prerušiť klávesom BREAK. Vtedy sa riadenie vráti editoru.

Ak sa pri nahraní modulu zistí chyba nahrávania vypíše sa do dialogového riadku text

IO ERROR

a po stlačení klávesu ENTER sa riadenie vráti editoru.

Ak editor zistí, že sa hľadaný text už do pamäti nezmestí, vypíše do dialogového riadku

MEM FULL

a po stlačení klávesu ENTER sa riadenie vráti editoru.

Úspešne nahratý text sa ukladá do pamäti pred riadok, na ktorom bol nastavený jazdec vtedy, keď užívateľ prešiel z obrazovkového režimu do príťazového. Toto ukládanie môže trvať až 20 sekúnd a počas nájdenia je vypísaný v dialogovom riadku text

WAIT

po vložení nahratého textu sa opäť prihlási editor výpisom EDI.

LOA - tento príkaz má rovnakú funkciu ako príkaz MER, len s tým rozdielom, že text nahraný z pásky prepíše text v pamäti. Preto o ňom platí všetko, čo bolo napísané o príkaze MER, len po nahratí textu sa riadenie hned vráti editoru.

ALD - špeciálny príkaz, pomocou ktorého užívateľ spustí postupnosť volaní jednotlivých modulov systému a sice ASM, LNK, DBG. Pokial preklad alebo spájanie skončí s chybami volanie ostatných modulov sa neuskutoční a riadenie sa vráti editoru. Blížšie pozri pri popise jednotlivých modulov.

DIS - spúšta spätný prekladač. Za príkazom DIS nasleduje medzera a za ňou počiatok a koncová adresa pamäte, ktorej obsah sa má späť preložiť v tvare

XXXX-YYYY

musí byť pomlčka

kde XXXX a YYYY sú hexadecimálne konštanty /nezačínajú však znakom #/. Inštrukcia na adrese YYYY je už neprekladá. Text sa ukladá pred riadok, na ktorém boli jazdec, rovnako ako keby ho užívateľ písal sám. Výrazy

v adresnom poli sú vyjadrené ako hexadecimálna konšanta a ak hodnota na niektornej adrese neobsahuje kód platnej inštrukcie, je táto hodnota preložená pomocou pseudoinštrukcie DB /pozri popis prekladača/. Ak chceme získať aj hodnoty čítača adres, môžeme takto získaný zdrojový text normálne preložiť a na základe výpisu prekladu doplniť výrazy v poli adresy o návestia.

Friaz DIS sa využíva ak chceme pokračovať v práci na programe, ktorý sme doteraz vytvárali v strojom kóde /alebo pod iným systémom/ pod systémom MRS, alebo ak chceme do oddaného programu vložiť niektoré moduly systému MRS, aby sa program stal na systéme MRS nezávislý /pozri popis modulov LIB a LNK/.

Ak sa pri vykonávaní spätného prekladu preplní pamäť, systém vypíše MEM FULL.

a riadenie sa vráti modulu MRS.

INS - toto je špeciálna obdoba príkazu DIS a slúži na vloženie zdrojového textu, ktorý neboli vytvorený systémom MRS. Za príkazom nasledujú dve hexadecimálne konštanty rovnako ako za príkazom DIS. Prvá konštanta obsahuje adresu užívateľského podprogramu, na hodnotu druhej konštanty je inicializovaný register HL pri prvom volaní užívateľského podprogramu. Tento register je k dispozícii užívateľskému modulu a pri ďalších volaniach má hodnotu, ktorú mal pri odchode z podprogramu.

Úlohou užívateľského podprogramu je pri každom zavolení uložiť do EUFRA od adresy #FEOO ďalší riadok zdrojového textu a nasadiť CARRY FLAG = 0. Ak už nie je k dispozícii ďalší riadok, užívateľský podprogram následne CARRY FLAG = 1 a riadenie sa vráti editoru.

Riadky textu sa syntakticky kontrolujú a správne riadky sa ukladajú do textu pred riadok, na ktorom bol jazdec /rozri prikaz DIS/.

Chybné riadky sa zobrazia do prvého riadku obrazovky a v pravom dolnom rohu je ich poradové číslo. Riadky je potom možné opravovať tak ako je to popísané v odseku 2.3.3. Oprava sa ukončí klávesom ENTER. V prípade chyby zaznie zvukový signál a cyklus sa opakuje. Prikaz INS je možné prerušiť klávesami CAPS SHIFT/MEDZERA. Do diałorového riadku sa vypíše

BREAK

a po stlačení klávesy ENTER sa riadenie vráti systému MRS. Ak sa pri vkladaní textu preplní pamäť, systém reaguje rovnako ako pri príkaze DIS. Na kláves CAPS SHIFT nedzera reaguje systém krátko po ukončení opravy klávesom ENTER.

MON
RUN
ASM
LNK
LIB
DBG

tieto příkazy vyvolávají příslušný modul systému a

odovzdávajú mu riadenie.

2.3.2 Obrazovkový režim

Po vykonaní príkazov LN a INI editor prejde do obrazovkového modulu. To znamená, že na obrazovku vypíše text od riadku nastaveného príkazom LN alebo je obrazovka prázdná /po príkaze INI/, jazdec je v ľavom hornom rohu a editor očakáva ďalší text.

Pre pochopenie obrazovkového režimu je nutné si uvedomiť, že editor už pripravuje text pre zostavovací jazyk, a to tak, že ho vhodne predspracuje. Preto už pri písaní textu musíme dbať na správnu syntax a v ďalšom teste sa preto prelinajú informácie o editore s informáciami o zostavovacom jazyku.

Všeobecný tvar riadku textu je následovný:

LABEL	INS	ADR	FOZNÁMKA
-------	-----	-----	----------

Vidíme, že riadok je rozdelený na 4 polia s týmto obsahom:

LABEL - pole návestia. Začína na nultej pozícii a má dĺžku 7 znakov. Je buď vyplnené medzerami, alebo obsahuje návestie. Návestie môže byť retazec maximálne 6 písmen a číslic, ktorý začína písmenom, zbytok pola musí byť vyplnený medzerami.

INS - pole inštrukcie. Začína na siedmej pozícii a má dĺžku 5 znakov. Buď je vyplnené medzerami, alebo obsahuje platnú inštrukciu mikroprocesora ZILOG 80, prípadne platnú pseudoinštrukciu zostavovacieho jazyka. Zbytok pola musí byť vyplnený medzerami.

ADR - pole adresy. Začína na 12. pozícii a má dĺžku 20 znakov. Ak je pole INS vyplnené medzerami, musí byť aj pole ADR vyplnené medzerami. Ináč obsahuje platnú adresu pre inštrukciu alebo pseudoinštrukciu zapísanú v poli INS. Pole adresy musí byť zakončené medzerou a zbytok pola musí byť vyplnený medzerami.

FOZN - pole poznámky. Začína na 32. pozícii a má dĺžku 32 znakov. Môže obsahovať ľubovoľné znaky, keďže ho zostavovací jazyk ignoruje. Počas písania textu do pola poznámky sa riadok posúva dolava a prvé znaky sa z obrazovky strácajú. Pole poznámky nesmie byť v riadku, ktorý nemá inštrukciu. Počas písania poznámky sa sleduje dĺžka riadku.

Okrem takého riadku pozná systém riadok poznámky a riadok na riadenie prekladu. Riadok poznámky začína znakom ; a môže obsahovať ľubovoľné znaky, keďže ho prekladač ignoruje. Riadok na riadenie prekladu začína znakom *, jeho obsah sa syntakticky kontroluje. Platné príkazy sú popísané v odseku 2.4.3. a zbytok riadku musí byť vyplnený medzerami.

Tieto riadky nie sú členené na polia a stlačenie kláves CAPS SHIFT/l alebo CAPS SHIFT/4 v nich sa považuje za chybú /pozri odsek 2.3.3./.

Pole adresy môže obsahovať okrem prvkov predpísaných presne inštrukciou aj prvky, ktoré sa vyhodnocujú počas prekladu,

a ktoré nazývame výrazy. Výraz v systéme MRS môže mať jeden z týchto tvarov:

- SYMBOL
- SYMBOL + SYMBOL
- SYMBOL - SYMBOL
- SYMBOL ★ SYMBOL
- SYMBOL / SYMBOL
- SYMBOL ! SYMBOL OPERÁCIA OR
- SYMBOL & SYMBOL OPERÁCIA AND
- SYMBOL ☐ SYMBOL OPERÁCIA XOR
- - SYMBOL
- > SYMBOL /VYŠŠI BYTE 2-BYTOVEJ KONŠTANTY/
- < SYMBOL /NIŽŠI BYTE 2-BYTOVEJ KONŠTANTY/

pričom symbol môže mať jeden z týchto tvarov:

- návestie
- decimálna konštantá, čo je reťazec číslí
- hexadecimálna konštantá, čo je reťazec číslí a pís-men A-F, pred ktorými je znak #.
- znaková konštantá, čo je ľubovoľný zobraziteľný znak uzavretý v apostrofoch.
- znak \$. označuje čítač inštrukcií.

Okrem inštrukcií môžu byť v poli INS tieto **pseudoinštrukcie**:

- ORG** - v adresnej časti musí byť výraz.
- DS** - v adresnej časti musí byť výraz.
- DB** - v adresnej časti musí byt jeden alebo viac výrazov oddelených čiarkami. V prípade pseudoinštrukcie DB ako výraz môže byť tiež reťazec znakov uzavretý v apostrofoch. (jednoduchých), napr. 'ABC'
- DW** - v adresnej časti musí byt jeden alebo viac výrazov oddelených čiarkami.
- EQU** - v poli návestia musí byť návestie a v poli adresy musí byť výraz.
- ENT** - v poli adresy musí byť jedno alebo viac návestí od-delených čiarkami.
- EXT** - v poli adresy musí byť jedno alebo viac návestí od-delených čiarkami.
- END** - pole adresy musí byť prázdne.

2.3.3. Písanie riadku

Riadok sa píše bežným **zpsobom**, pričom sa využívajú tieto špeciálne klávesy:

- CAPS SHIFT/8 - spôsobí presun jazdca na nasledujúci znak. Hranice riadku sa kontrolujú.
- ← CAPS SHIFT/5 - spôsobí presun jazdca na predchádzajúci znak. Hranice riadku sa kontrolujú.
- CAPS SHIFT/9 - vloží medzeru na pozíciu označenú jazdcom a zbytok pola posunie doprava.
- CAPS SHIFT/Ø - vymaže znak označený jazdcom a zbytok pola posunie dolava ak je jazdec za posledným znakom riadku, posunie sa o jeden znak do-lava a posledný znak sa vymaže.

CAPS SHIFT/4 - posunie jazdca na začiatok ďalšieho pola, hranice riadku sa kontrolujú.

CAPS SHIFT/1 - presunie jazdca na začiatok pola alebo na začiatok predchádzajúceho pola, hranice riadku sa kontrolujú.

SYMBOL SHIFT/MEDZERA - nastaví jazdca na posledný znak v poli adresy

MEDZERA - vyplní zbytok pola medzerami a jazdca nastaví na začiatok ďalšieho pola. Táto funkcia je potlačená v poli poznámky, v riadku poznámky, alebo keď je medzera súčasťou retazca alebo znakovnej konštanty. V týchto prípadoch sa medzera normálne uloží do textu.

Ďalšie riadiace znaky spôsobujú ukončenie písania riadku. Riadok sa syntakticky skontroluje. Ak niektoré pole nevyhovuje predpisom, jazdec sa nastaví na jeho začiatok, vydá sa zvukový signál a užívateľ musí riadok opraviť.

Syntakticky správny riadok sa pomerne komplikované komprimuje a opäť sa prevedie do zdrojovej formy. Potom sa porovná s pôvodným riadkom a pokial' sa tieto dva riadky nerovnajú, vyspíše sa riadok, ktorý vytvoril systém naniesť pôvodného riadku a zaznie zvukový signál. Ak je užívateľ s novým riadkom spokojný stlačí ENTER a pokračuje v práci, ináč ho môže opraviť.

Na prvý pohľad komplikovaný systém je pre užívateľa prakticky neviditeľný. Ak sú totiž riadky rovnaké /a to sú vždy, keď je riadok správne napísaný/, pokračuje normálne v práci.

Pomocou tohto mechanizmu sa objavia chyby

- pretečenia /príliš veľká konštantu sa komprimovaním zmien/
- neformalizovaný zápis konštanty /hexadecimálna konštantu musí mať párný počet BYLOV, pred decimálnou konštantou nesmú byť nuly/
- formálne súčasne, ale neexistujúce inštrukcie /napr. inštrukcia ADD HL, D sa pretvorí na inštrukciu ADD IX, IX a tým sa zistí chyba/.

Systém MRS /Moduly EDI, ASM a DEC/ správne spracúva aj tzv. **skryté** inštrukcie. Sú to jednak inštrukcie **SLL** /operačný znak **CB30** až **CB37**/ a inštrukcie ekvivalentné inštrukciám, ktoré pracujú s regístrami H a L, ale s prefixom DD resp. FD/. Tieto potom pracujú s vyšším alebo nižším BYTOM indexregistra IX resp. IX. Tieto inštrukcie sa zapisujú tak, že na mieste registraHsa píše XH alebo YH a na mieste registra L sa píše XL alebo YL. Nie každá nahradá napr. registra H registrom XH je možná /napr. ak už inštrukcia má prefix CB tak H sa na XH ne-dá nahradit/. Bližšie o skrytých inštrukciách pozri v odbornej literatúre.

Ináč jediná odchýlka od štandardného zápisu inštrukcii je, že inštrukcia EX AF,AF' sa píše bez apostrofu, teda ako EX AF,AF

V prípade správneho riadku ďalšia akcia závisí od použitého riadiaceho znaku.

CAPS SHIFT/7 ↑

- jazdec sa nastaví na začiatok predchádzajúceho riadku. Ak je jazdec na prvom riadku obrazovky, celá obrazovka sa posunie o riadok dole. Ak je jazdec na prvom riadku textu zaznie zvukový signál a nič sa nevykoná.

CAPS SHIFT/6 ↓

- jazdec sa nastaví na začiatok nasledujúceho riadku. Ak je jazdec na poslednom riadku obrazovky posunie sa celá obrazovka o riadok hore. Ak je jazdec za posledným riadkom textu, ozve sa zvukový signál a nič sa nevykoná.

CAPS SHIFT/3

- vypíše sa predchádzajúca obrazovka textu a jazdec je na začiatku prvého riadku obrazovky.

CAPS SHIFT/2

- vypíše sa nasledujúca obrazovka textu a jazdec je na začiatku prvého riadku obrazovky.

SYMBOL SHIFT/ENTER

- vypíše sa nová obrazovka textu tak, že riadok, na ktorom bol jazdec, sa stane prvým riadkom obrazovky

CAPS SHIFT/SYMBOL SHIFT

- vymaže sa riadok, na ktorom bol jazdec a zbytok obrazovky sa posunie hore. V tomto prípade sa neanalyzuje syntaktická správnosť riadku. Ak je jazdec za posledným riadkom textu ozve sa zvukový signál a nič sa nevykoná.
- za riadok, na ktorom je nastavený jazdec sa vytvorí jeho kópia.

CAPS SHIFT/ENTER

- text na obrazovke počnúc riadkom, na ktorom je jazdec, sa posunie smerom dole, čím sa v tvorí miesto pre vloženie nového riadku. Ak je jazdec na poslednom riadku obrazovky, toto miesto sa vytvorí posunom celého textu o riadok hore. Ak je jazdec za posledným riadkom textu, ozve sa zvukový signál a rič sa nevykoná.

CAPS SHIFT/MEDZERA

- editor vypíše text EDI do dialogového riadku a prejde do príkazového režimu.

Odoslanie iných riadiacich znakov má za následok zvukový signál a nič sa nevykoná. Každý kláves má opakovaciu schopnosť, to znamená, že jeho podržanie má rovnaký výsledok, ako keby sa opäťovne stláčal.

2.3.4. Chybové hlásenia

Na veľkú väčšinu chybných akcií reaguje editor zvukovým signálom a prípadne nastavením jazdca na chybnú položku, existujú však dva chybové stavby, následkom ktorých editor vypíše chybové hlásenie do dialogového riadku a po stlačení klávesu ENTER sa riaderie vráti systému MRS.

MEM FULL - tento výpis známená, že pamäť, určená na zápis textu sa zaplnila. Nutné je text skrátit, prípadne vymazať moduly z binárnej knižnice /pozri program na obsluhu knižnice/. Prakticky je ale tento stav zriedkavý, pretože veľkosť voľnej pamäti stačí na cca 6000 riadkov zdrojového textu. Ak došlo k preplneniu pamäti užívateľ vyvolá opäť editor príkazom LN=0 prejde do obrazovkového režimu, klávesom CS/SS vymaže riadok, pri ktorom došlo k preplneniu a klávesom SS/ENTER otnoví obrazovku.

Iný dôvod na výpis tohto oznamu je, keď sa zaplní tabuľka symbolov. Táto tabuľka je uložená za textom a dynamicky sa rozširuje. Jej maximálna kapacita je však 255 návestí. Návestia, ktoré sa nepoužijú pri preklade sa automaticky vymažu. Po skončení prekladu, /pozri popis prekladača zostavovacieho jazyka/.

2.4. Prekladač zostavovacieho jazyka

2.4.1. Spustenie prekladu

Prekladač zostavovacieho jazyka /v ďalšom texte tiež ASSEMBLER/ sa vyvolá príkazom **ASM**, za ktorým ako ďalší parameter nasleduje meno prekladaného modulu. ASSEMBLER vyhľadá daný modul v binárnej knižnici a zdrojový text preloží na miesto pôvodného modulu. Ak taký modul v knižnici nie je, ASSEMBLER preloží zdrojový text ako nový modul a pripojí ho na koniec knižnice.

Ak v príkaze ASM meno modulu chýba, vytvorí sa modul s prázdnym menom na konci knižnice. Takýto modul je v knižnici len dočasne. Každý ďalší preklad ho vymaže.

Inými slovami, užívateľ by mal preložiť a odladiť modul najprv bez mena a až keď je všetko v poriadku, zaradiť modul do knižnice príkazom ASM MENO.

ASSEMBLER v každom prípade sleduje hranice pamäti, ktorá je modulu pripravená už veľkosťou miesta, ktoré zaberá pôvodný modul alebo veľkosťou miesta od konca knižnice po koniec voľnej pamäti.

ASSEMBLER prekladá text pripravený editorom do príslušnej pamäti a na obrazovke vytvára výpis prekladu. Riadok výpisu sa vypíše do dvoch až troch riadkov obrazovky nasledovne:

1'. až 5. znak - poriadové číslo riadku

7'. znak - obsahuje medzeru alebo kód chyby. Chybové kódy sú popísané ďalej.

9'. až 12. znak - hexadecimálny výpis hodnoty čítača adres.

14. až 21. znak - hecadecimálny výpis kódu

Druhý a v prípade poznámky aj tretí riadok obrazovky obsahuje výpis zdrojového textu.

Výpis je možné keďkoľvek zastaviť klávesom **CS/S** a potom opäťovne spustiť klávesom **CS/Q**. Po skončení prekladu vypíše

ASSEMBLER informáciu o počte chýb

ERRORS : N

kde N je počet zistených chýb.

Pokiaľ sa počas prekladu vyskytli chyby, modul sa do knižnice nezaradí. Treba si uvedomiť, že keď sme chceli preložiť text cez modul, ktorý už v knižnici bol, je tento pôvodný modul chybným prekladom zničený. Pravda, funkcie ostatných modúlov sú zachované.

Pokiaľ sa preložil celý text /to znamená, že pseudoinštrukcia END je posledným riadkom textu/, prezrie sa tabuľka symbolov a v preklade nepoužité symboly sa vymažú.

Na záver prekladu, ak bol požadovaný výstup všetkých riadkov príkazom *A /pozri ďalej odsek 2.4.2/, sa vypíše tabuľka použitých návestí. Návestia sú usporiadané podľa abecedy a za menom návestia nesleduje jeho hodnota ako hexadecimálne číslo. Návestia popísané pseudoinštrukciou EXT majú na mieste hodnoty, ktorá môže byť ešte neznáma, znaky XXXX.

2.4.2 Vyhodnotenie výrazov

Výraz v poli adresy sa vyhodnotí bežným spôsobom. To znamená, že decimálna a hexadecimálna konštanta má hodnotu danú svojím zápisom. Znaková konštanta má hodnotu danú hodnotou ASCII znaku, ktorý ju reprezentuje a hodnota návestia je daná veľkosťou čítača adres v okamihu, keď sa návestie vyskytlo v poli návestia.

Špeciálne unárne výrazy sú >Symbol, ktorý definuje hodnotu vyšších 8 BITOV zo 16-BITOVEJ hodnoty a <Symbol, ktorý definuje hodnotu nižších 8 BITOV 16-BITOVEJ hodnoty. Napr. ak chceme zistit, či L register sa rovná nižším 8 BITOM návestia adresy, možno to vykonáť nasledovne:

LD A,L
CF <ADRESA

Iná možnosť, ako priradiť návestiu hodnotu je pseudoinštrukcia EQU /pozri ďalej/.

Symbol \$ nadobúda počas prekladu hodnotu čítača inštrukcií.

2.4.3 Riadenie prekladu pseudoinštrukciami a direktívami

Pseudoinštrukcie riadia preklad zdrojového textu a direktívy riadia vlastnú činnosť prekladača. Pseudoinštrukcie už boli spomenuté v časti venovanej editoru. Tu rozoberieme ich funkciu vzhľadom k ASSEMBLERU.

ORG - riadi veľkosť čítača adres. Výraz v poli adresy sa vyhodnotí a jeho hodnota sa priradí čítaču adres. Prvky výrazu musia byť známe už pri prvom prechode. To znamená, že súčasťou výrazu nesmie byť návestie, ktoré sa predtým nevyskytlo v poli návestia.

DS - riadi veľkosť čítača adres. Čítač adres sa zvýši o hodnotu danú výrazom v poli adresy, čím sa vlastne

v programe rezervuje pamäť. Todenie ako v prípade pseudoinstrukcie ORG musí byť hodnota výrazu známa už v prvom prechode.

- DB** - ukladá do pamäti hodnoty určené výrazmi v poli adresy. Tieto hodnoty sa musia zmestíť do 1 BYTE. Ak je v adresnej časti retazec znakov v apostrofoch, tak sa ¹⁷²³ do pamäti postupne uložia ich ASCII hodnoty. *prednachystaný* /mpa. ATAOJ/
- DW** - ukladá do pamäti hodnoty určené výrazmi v poli adresy. Každá hodnota zabera dva BYTY. *Nejde referenciu znaku?*
- EQU** - pripadí návestiu v poli návestia hodnotu danú výrazom v poli adresy. Aj tato hodnota musí byť známa už v prvom prechode.
- ENT** - dopisuje návestia v poli adresy ako bod vstupu /ENTRY POINT/. Význam tejto pseudoinstrukcie je objasnený v časti spojovací program.
- EXT** - popisuje návestia v poli adresy ako vonkajší symbol /EXTERNAL POINT/. Význam tejto pseudoinstrukcie je popísany v časti Spojovací program.
- END** - tato pseudoinstrukcia ukončuje prekladanú časť textu, teda text sa prekladá od začiatku po pseudoinstrukciu END.

Pseudoinstrukcie ENT a EXT sa môžu použiť, len ak je v pamäti prítomná knižnica binárnych modulov /aspoň základný modul SYSMOD/. V opačnom prípade ich použitie má nedefinovaný výsledok.

Na riadenie práce prekladača sa využívajú špeciálne riadky a sice také, ktoré začínajú znakom *. V tomto prípade ASSEMBLER predpokladá, že za znakom * nasleduje niektorý z týchto príkazov:

- A** - znamená, že sa má na výstupnom zariadení vytvárať výpis všetkých riadkov. Ak chce užívateľ získať len výpis tabuľky symbolov /pozri odsek 2.4.1/, musí vložiť pred riadok s pseudoinstrukciou END direktívou A.
- E** - znamená, že sa má na výstupnom zariadení vytvárať len výpis chybných riadkov. Táto hodnota je preddefinovaná na začiatku prekladu.
- L** - znamená, že počas prekladu sa vytvára výpis na tlačiareň /pozri bližšie v odseku 2.4.5/.
- P** - má zmysel len v spojení s direktívou L. Pomocou direktívy P vyšleme na tlačiareň riadiaci znak nová strana /#OC/. Ak za znakom F nasleduje decimálna konštantá, tak sa znak nová strana vyšle vždy po vytlačení daného počtu riadkov.
- T** - výpis sa bude vytvárať na terminál. Táto hodnota je nastavená na začiatku prekladu.

Platnosť direktív začína ďalším riadkom a trvá po vysielte ďalšieho príkazu včítane.

CNNNN - /kde NNNN je hexadecimálna konštantá/ znamená, že hoci preklad pokračuje normálnym spôsobom, výsled-

ný kód sa začne ukladať v pamäti od adresy NNNN. Týmto spôsobom možno vytvoriť v pamäti kód, ktorý je schopný práce až po presunutí na patričné miesto v pamäti. Ak NNNN=0, binárny kód sa v pamäti nevytvára vôbec. Tak je možné získať výpis programu bez narušenia pamäti. Platnosť príkazu CNNNN sa zruší pri výskypke pseudoinstrukcie ORG.

2.4.4 Uloženie modulu do knižnice - pseudoinstrukcia ORG

Preložený modul sa spravidla prekladá na koniec knižnice a okrem vlastného binárneho kódu sa do knižnice uložia aj informácie o mene a dĺžke modulu /tieto sa uložia pred začiatím prekladu/ a informácie o vstupných /ENT/ a výstupných /EXT/ návestiach. Táto akcia je korektná, pokial' sa počas prekladu nevykyla pseudoinstrukcia ORG alebo direktíva CNNNN. V opačnom prípade sa nedá zaručiť integrita knižnice. Pri bežnom spôsobe práce teda užívateľ pseudoinstrukciu ORG nepoužíva.

Napriek tomu bola do prekladača zaradená a s jej pomocou je možné riadiť preklad tak, aby sa binárny kód uložil na presne stanovené miesto pamäti a aby sa nič iné neprepísalo. To je možné dosiahnuť aj vhodným použitím direktívy CNNNN. V oboch prípadoch je nutné aby

- preklad bol bez mena, teda za príkazom ASM už nič ne-nasledovalo
- pseudoinstrukciou OPC sa nastavil čítač adries na požadovanú hodnotu
- v programe sa nesmú vyskytovať pseudoinstrukcie ENT a EXT, aby sa za program nezačali ukladať príslušné návestia

Ešte raz zdôrazňujeme, že použitie pseudoinstrukcie ORG je v systéme MRS neštandardné. Užívateľ spravidla nemusí vedieť, kde sa jeho kód uložil, pretože sa v každom prípade môže obracať na návestia a vstupné body.

2.4.5 Použitie tlačiarne

Ako bolo spomenuté, výskyt riadku *L spôsobí, že sa začne vytvárať výpis na pripojenej tlačiarne. Systém je dodávaný bez obslužného programu pre tlačiareň, pretože spôsoby pripojenia tlačiarne k počítaču ZX SPECTRUM sú veľmi rozmanité. V knižnici SYSMOD je však zaradený vstupný bod FNTCHR, kde je instrukcia skoku /C3/. Normálne je v jej adresnej časti skok na instrukciu RET. Ak si užívateľ napíše vlastný obslužný podprogram pre vytlačenie jedného znaku /a prípadne aj pre inicializáciu tlačiarne/, zaradí ho do knižnice a zodpovedajúcim spôsobom modifikuje adresnú časť instrukcie skoku na adrese FNTCHR, získa možnosť tlačiť výstup z prekladača. Užívateľský obslužný podprogram preberá kód znaku v registri A, regisetre môže meniť. Ako riadiace znaky sa vyskytujú kódy #OA /LF/ a # OD /C../ # OC FF /'preč' cd na novú stranu/.

2.4.6 Chybové oznamy

Chybové oznamy možno rozdeliť na dve skupiny. Jednak sú

to vážne chyby, ktoré okamžite ukončia preklad a sice:

NO END - prekladač nenašiel pseudoinstrukciu END, ktorá musí ukončiť prekladaný zdrojový text

NEM FULL - pokus o uloženie kódu mimo pamäť vyhradenú pren. Bud j preložený binárny kód príliš dlhý, alebo sa nesprávne použila pseudoinstrukcia ORG, prípadne direktíva CNNNN. Táto chyba vzniká aj pri skutočnom pokuse o zápis do pamäti. Vhodným použitím direktív CNNNN a následným posunutím binárneho kódu na správne miesto je možné preložiť kód na ľubovoľné miesto pamäti.

Prerušenie prekladu - ak užívateľ počas prekladu stlačí kláves CS/MEDZERA, preklad sa preruší, do diaľkového riadku sa vypíše text

BREAK

a po stlačení klávesy ENTER sa riadenie vráti systému MRS.

Druhú skupinu tvoria chyby zistené počas prekladu. Tieto chyby sú indikované kódom chyby v znaku pred polom návestia. Chybnej riadky sa zobrazia na obrazovke vždy, nie je možné to potlačiť.

U - v adresnej časti sa vyskytuje návestie, ktoré nebolo definované, teda nevyskytlo sa v poli návestia ani nebolo definované pseudoinstrukciou EXIT. Návestia v adresnej časti pseudoinstrukcii musia byť definované prv, ako sa vyskytnú v adresnej časti, teda musia byť definované v prvom prechode prekladu. Návestie, ktoré bolo raz označené ako nedefinované, je tak označené pri každom výskytu, aj keď bolo prípadne neskôr v teste nájdene v poli návestia.

M - v poli návestia sa vyskytuje návestie, ktoré už bolo definované, teda už sa vyskytlo v poli návestia. Týmto chybovým kódom je označený jeho každý výskyt. Táto chyba sa vyskytuje aj vtedy, keď ako vstupné návestie bolo popísané návestie, ktoré sa ako vstupný bod v knižnici už vyskytuje, alebo je ako vonkajšie návestie popísané niektoré návestie programu.

D - v adresnej časti sa vyskytlo návestie, ktoré bolo viackrát definované, teda jeho výskyt v poli návestia bol označený chybovým kódom D. Tento oznam má v podstate informatívny charakter. Užahčuje užívateľovi opravu chyby D.

R - výraz v adresnom poli inštrukcie, ktorá pripúšta len 8-BITOVÚ hodnotu prekročil rozsah 1 BYTU /0-255/.

E - návestie popísané ako externál sa vyskytlo ako súčasť výrazu alebo je v adresnej časti inštrukcie, ktorá pripúšta len 8-BITOVÚ hodnotu v adresnej časti.

Z - pri použíti operácie delenia došlo k pokusu o deleenie Ø.

Výraz, pri spracovaní ktorého bola zistená chyba má hodnotu 0. Ak sa v adresnej časti inštrukcie v skytuje návestie popísané ako vonkajšie, tak v binárnom kóde sa počas prekladu ukladajú informácie, ktoré prekladač na záver použije na prípravu tabuľky potrebnej pre prácu spojovacieho programu. Vo výpise prekladu sa teda objavia hodnoty, ktoré pre užívateľa nemajú zmysel.

2.4.7 Použitie relatívnych skokov

Fokial sa v adresnej časti inštrukcie relatívneho skoku vyskytuje výraz obsahujúci návestie, tak výsledná hodnota sa automaticky upraví. Ak sa vo výraze návestie nevyskytuje, výsledná hodnota výrazu sa priamo zapíše do adresnej časti inštrukcie. Teda napr. inštrukcia **JR LABEL+4** má v adresnom poli hodnotu **LABEL+4-\$-2**, kde \$ je hodnota čítača inštrukcie v čase prekladu inštrukcie. Inštrukcia **JR #10** má po preklade v adresnej časti hodnotu **#10**.

2.5 Spojovací program

Spojovací program sa vyvoláva príkazom **LNK**, za ktorým nasleduje parameter. Úloha spojovacieho programu je veľmi jednoduchá, keďže veľkú väčšinu jeho práce vykoná už prekladač. Spojovací program len prehľadá zoznam vonkajších návestí v celej binárnej knižici, ku každému vonkajšiemu návestiu nájdzie zodpovedajúce vstupné návestie a jeho hodnotu uloží na všetky miesta binárneho kódu, kde sa toto návestie vyskytuje v adresnej časti. Vonkajšie návestie, ktoré sa neziví vstupnými návestiami knižnice nenájde, sa vypíše na obrazovku pod hlavičku

UNDEFINED EXTERNALS:

a na záver sa vypíše informácia o počte takýchto návestí výpisom

ERRORS:N

Ak bol spojovací program spustený príkazom LNK, riadenie sa vráti modulu MRS. Ak bol spojovací program spustený príkazom ALD, riadenie sa vráti editoru /ak sa vyskytli nedefinované vonkajšie návestia/, alebo ladiaceru programu.

Inými slovami, návestia, v programe, ktoré majú byť volané z iných samostatne prekladaných modulov, sa popíšu v module, v ktorom sa vyskytujú pseudoinstrukciou ENT a v ostatných moduloch pseudoinstrukciou EXT. Návestia definované pomocou EXT musia reprezentovať 16-BITOVÚ hodnotu a nesmú byť súčasťou výrazu. Spojovací program správne doplní adresné časti, v ktorých sa vyskytujú návestia popísané pomocou EXT a tak vlastne prepojí viac modulov do jedného celku.

Upozorňujeme, že sa nesmie spustiť program, ktorý obsahuje vonkajšie návestia, ale neboli spojený spojovacím programom, alebo ktorý obsahuje nedefinované vonkajšie body, pretože adresná časť príslušnej inštrukcie po preklade obsahuje nedefinovanú hodnotu.

2.6. Program na obsluhu binárnej knižnice

2.6.1 Ovládanie programu na obsluhu knižníc

Tento program sa spúšta príkazom **LIB** bez parametra. Po spustení sa prihlási výpisom

LIB

do dialogového riadku a očakáva jeden z týchto príkazov

LST - vypíše zoznam modulov uložených v knižnici. Pre každý modul tiež vypíše zoznam vstupných návestí s ich hodnotami v hexadecimálnom tvare. Rovnako ako pri výpise, ktorý robí prekladač je možné tento výpis zastavovať a spúštať klávesami CS/S a CS/Q alebo prerušiť klávesom CS/MEDZERA. Vtedy systém vypíše do dialogového riadku text

BREAK

a po stlačení klávesy ENTER sa vráti riadenie riadiacemu modulu.

DEL - vymaže z knižnice posledný modul. Moduly z vnútra knižnice sa nedají vymazať tak, aby sa ušetrilo miesto /aby sa urobila kompresia knižnice/, keďže sú preložené v absolutných adresách. Vymazanie vnútorného modulu bez ušetrenia pamäte /napr. ak chceme použiť vstupné návestia, ktoré sú v nom definované, v inom module/ je možné, ak preložíme prázdny modul /pozostávajúci z pseudoinštrukcie LND/ s rovnakým menom. Posledný modul bez mena sa automaticky vymaže na začiatku každého prekladu. Základný modul knižnice SYSMOD nie je možné vymazat.

SAV - rovnako, ako pri ukladaní zdrojového textu tento príkaz uloží binárnu knižnicu na magnetofónovú pásku. Pred jeho odoslaním užívateľ zapne magnetofón a po uložení knižnice sa prihlási opäť modul LIB. V posledných 2 BYTOCH hlavičky sú zapísané Ø, čím systém r zozná moduly knižnice od iných modulov s rovnakým menom. Pred uložením knižnice sa vymaže prípadný modul bez mena.

LOA - slúži na nahranie binárnej knižnice z magnetofónu. Nahranie sa vykoná rovnakým spôsobom ako v prípade zdrojového textu /pozri tam/. Program kontroluje, či sa nahraním knižnice neprepíše prípadný zdrojový text. V takom prípade vypíše chybové hľasenie

MEM FULL

a po stlačení klávesy ENTER sa riadenie vráti modulu MRS.

Nová knižnica vždy nahradí starú.

ENT - za príkazom nasleduje návestie, ktoré je v knižnici popísané ako vstupný bod. Od tohto návestia sa potom spustí program príkazom **RUN**. Fokial' uží-

vateľ ešte nepoužil príkaz ENT, alebo za ním neuviedol žiadne návestie. Príkaz RUN vráti riadenie okamžite systému MRS.

RUN - týmto príkazom sa spustí užívateľský program od návestia popísaného príkazom ENT. Posledná inštrukcia RET v užívateľskom programe vráti riadenie systému MRS.

MON
EDI
ASM
LNK
ALD
DBG

ukončia prácu programu LIB a riadenie sa vráti príslušnému modulu

2.6.2 Základný modul knižnice

Tento modul sa volá **SYSMOD** a musí byť v knižnici vždy prítomný. Obsahuje rôzne podprogramy systému **MRS** popísané ako vstupné návěstí. Užívateľ teda nemusí vediať na ktorých absolútnych adresách sú moduly nachádzajú, ale ich označuje menami pomocou pseudoinštrukcie **EXT**. Modul **SYSMOD** obsahuje tieto vstupné návestia:

- **MEMORY** - obsahuje adresu prvej voľnej pamäťovej buňky za knižnicou. Táto hodnota sa inicializuje až pri vstupe do ladiaceho programu.
- **MEMTOP** - obsahuje adresu prvej obsadenej pamäťovej bunky. **MEMORY** a **MEMTOP** umožňujú prácu s dynamickou pamäťou. Inými slovami, ak užívateľ v programe vykoná inštrukcie

```
EXT MEMORY, MEMTOP
LD DE, /MEMORY/
LD HL, /MEMTOP/
```

tak získa v registroch **<DE>** a **<HL>** hranice použitnej pamäti - bežiaci program má k dispozícii pamäť od **<DE>** do **<HL>** - 1

- | | |
|-------------|---|
| DCBN | <ul style="list-style-type: none"> - konverzia retázca <u>decimálnych čísl</u> na 16-BITOVÚ binárnu hodnotu
 vstup: <HL> = adresa retázca
 výstup: <HL> = adresa konca retázca /řetězec
 kód je lubovoľný symbolom mimo čísla 0-9.
 <DE> = konvertovaná binárna hodnota (hexadecimálne)
 <A> = ukončovací znak (ASCII kód hexadecimálne)
 miení sa: <AF>, <DE>, <HL> |
| HXBN | <ul style="list-style-type: none"> - konverzia řetězce hexadecimálnych čísl na 16-BITOVÚ binárnu hodnotu
 vstup: <HL> = adresa řetězce
 výstup: <HL> = adresa konca řetězce /řetězec končí lubovoľným symbolom mimo čísla 0-9 a písmen A-F.
 <DE> = konvertovaná binárna hodnota
 miení sa: <AF><DE><HL> |

priemysla

- BNHX** - konverze 8-BITOVÉ hodnoty na dvě hexadecimální číslice
 vstup: <A> = konvertované číslo
 <HL> = adresa paměti, kde se má výsledek
 konverze uložit
 výstup: <HL> = adresa paměti za uloženou dvojicí znaků
 mění se: <AF>, <HL>
- BNDC** - konverze 16-BITOVÉ hodnoty na řetězec decimálních číslic.
 vstup: <BC> = konvertované číslo
 <HL> = adresa paměti, kde se má výsledek uložit
 výstup: <HL> = adresa paměti, za řetězem decimálních číslic číslo je uložené v paměti jako řetězec ASCII znaků, předřazené Ø se vypustí
 mění se: <AF>, <BC>, <DE>, <HL>
- INKEY** - čítanie jedného znaku z klávesnice s autoratickým opakováním. Modul čaká, kým nie je stlačený kláves. Po opäťovnom volaní vykonáva automatické opakovanie, teda podržanie klávesu má rovnaký účinok ako jej opäťovné stláčanie. Stlačenie klávesu je indikované zvukom.
 vstup:
 výstup: <A> = ASCII kód stlačenej klávesy
 mění se: <AF>
- SCAN** - ohmatanie klávesnice. Modul zistí, či je niektorý kláves stlačený
 vstup:
 výstup: <A> = Ø ak nie je nič stlačené, alebo ak je stlačená nelegálna kombinácia kláves /napr. SS alebo SS/A/
 mění se: <AF>
- CLEAR** → vymazanie obrazovky
 vstup:
 výstup:
 mění se: všetko
- OUTCHR** → vypisuje znak na pozícii označenú jazdcom a jazda posunie doprava. Na konci riadku prejde na nový riadok, na konci obrazovky posunie obrazovku hore. OUTCHR správne ošetrí tieto riadiace znaky:
 #19 - posun jazdca vľavo
 #18 - posun jazdca vpravo
 #OD - ukončenie riadku a prechod na nový riadok
 Na obrazovke sa jazdec nezobrazí, /zabezpečuje to modul SETCRS/ modifikuje sa len premenná CURSOR.
 vstup: <A> = kód znaku
 výstup:
 mění se: <AF>
- CURSOR** → adresa paměti, která obsahuje poziciu jazdca na obrazovke pre modul OUTCHR. Sú to dva BYTY, v prvom sa uchováva číslo stĺpca /Ø-31/, v druhom číslo riadku /Ø-23/.

SETCRS - na pôvodnom mieste sa jazdec /blikajúci štvorček/ zmaže a na pozícii, ktorú udáva premená CURSOR sa objaví.

vstup: CURSOR = nová pozícia jazdca

výstup:

mení sa: <AF>

MRS - praca s čialógovým riadkom. Modul vypíše požadovaný text do dialógového riadku, načíta vstupný text a podľa požiadavky ho analyzuje na trojznakové príkazy. Vyvolaním tohto modulu má užívateľ k dispozícii všetky funkcie, ktoré sú popísané v opise práce s modulom MRS. To znamená, že na editovanie textu je možné využiť klávesy CAPS SHIFT/5, 8, 9, 0. Vstup textu vždy ukončuje klávesa ENTER

vstup: <A> = číslo pozicie v dialogovom riadku, od ktorej sa má vypísať požadovaný text /0-31/

<HL>= adresa vypisovaného textu

 dĺžka vypisovaného textu. Ak B = 0 text sa nevypisuje

<C> = 0 nič sa nečíta. Ináč sa číta text z klávesnice a ukladá sa do BUFRA od adresy #FE00.

<DE>= adresa tabuľky trojznaakových príkazov. Ak <DE> = 0, načítaný text sa neanalyzuje. V tabuľke nasledujú za sebou vždy trojznaaková skratka príkazu a adresa obslužného programu. Tabuľka je zakončená binárnu 0. Ak je požadovaná táto analýza, modul nevráti riadenie, kym užívateľ nenapíše riadok, ktorý začína prílustrným príkazom.

výstup: = počet načítaných znakov bez ENTER. Načítaný reťazec je ukončený binárnu 0.

<DE>= adresa načítaného reťazca. Pokiaľ bola požadovaná analýza trojznaakových príkazov, ukazuje DE na prvý znak za príkazom.

<HL>= ak bola požadovaná analýza trojznaakových príkazov, obsahuje adresu príslušného podprogramu.

mení sa: <AF> , <BC> , <DE> , <HL>

MRS2 - umožní opraviť text načítaný modulom MRS. Musímu predchádzať volanie modulu MRS s čítaním reťazca.

vstup: použije infomácie získané pri praci modulu MRS. Modul MRS2 sa !!!!! musí volať inštrukciou JP MRS2 !!!!!

výstup: rovnaký ako pri module MRS
mení sa: <AF> , <BC> , <DE> , <HL>

SCHENT - vyhľadá vstupný bod v knižnici. Dá sa využiť ak napr. jeden program predukuje dátu, ktoré má spracovať iný modul knižnice. Pomocou SCHENT môže daný program nájsť tabuľku v inom module, kam má uložiť výstupné dátu.

vstup: <HL>= adresa reťazca, ktorý udáva požadované vstupné návestie. Reťazec musí končiť binárnou nulou.

výstup: Z FLAG = 1 - chybná syntax alebo v knižnici nie je také meno. Ináč

<HL> = adresa meno v knižnici. Na tejto adrese je 6 znakov meno a ďalšie dva znaky adresa, ktorú toto meno označuje /napr. adresa tabuľky, ktorej meno je popísané ako ENT/. mení sa: **<AF>**, **<BC>**, **<DE>**, **<HL>**

PNTCHR - vytlačenie jedného znaku na tlačiareň. Vysvetlenie tohto modulu pozri v odseku 2.4.5 práca s tlačiarňou

výstup: **<A>** - kód tlačeného znaku

výstup:

mení sa: **<AF>**

INVPIX → invertuje znak obrazovky.

výstup: **<H>** - riadok Ø-23

<L> - stĺpec Ø-31

výstup:

mení sa: **<AF>**

II. - zoznam ilegálnych inštrukcií. Pozri odsek 2.7.4

Využitie modulu **SYSMOD** môže spočiatku robíť ťažkosti, keďže v obdobných systémoch sa nevyskytuje. Je to však veľmi pochoplný a silný prostriedok ako dokumentuje nasledujúci príklad:

Chceme vymazať obrazovku a v cykle načítať znak, zobrazit znak na obrazovku, ktorým nie je načítaný znak !!.

	EXT	CURSOR, OUTCHR, INKEY, CLEAR
START	CALL	CLEAR
	LD	HL, Ø
	LD	/CURSOR/, HL
LOOP	CALL	INKEY
	PUSH	AF
	CALL	OUTCHR
	POP	AF
	CP	'!'
	JR	NZ, LOOP
	RET	
	END	

Po priečložení /príkazom **ASM**/ a spojení /príkazom **LNK**/ programu, tento je pripravený k práci. Vidíme, že sa nemusíme starat o adresy daných modulov a program nemusíme modifikovať, aj keď sa systém **MRS** zmení. Zmení sa totiž zodpovejúcim spôsobom aj modul **SYSMOD**.

Ak neskôr chceme odstrániť závislosť programu na systém MRS, môžeme príkazom **DIS** modulu EDI vložiť príslušné časti kódu programu **MRS**, ako to bolo prv popísané. Preto je súčasťou tejto príručky aj výpis zdrojového programu niektorých modulov.

2.7. Ladící program DEBUGGER

2.7.1. Úvod

Ladiaci program sa spúšťa príkazem **DBG** bez parametrov. Návrhu modulu **DBG** bola věnovaná veľká starostlivosť, protože lade-

debug - ladiť (program); odstraňovať závady

debugger - ladiaci program

debugging - ladienie (programu)

bua [bag] - 1. štěnice 2. hmyz

nie programov je najčastejšia činnosť programátora. Po spustení modul DBC vypíše základnú stavovú informáciu do spodných dvoch riadkov v tvare

X PCPC inštrukcia SZAPC
AA BBCC LDEE HHLL XHXL YHYL SPSF

X - status. Status je jeden znak, ktorým je užívateľ informovaný o stave ladiaceho programu, spravidla X je medzera
PCPC - hexadecimálna hodnota registra PC

Inštrukcia vytvorí výpis inštrukcie, ktorú sa má vykonať, v symbolickom tvare. Hodn ta adresnej časti je vyjadrená ako hexadecimálne konštantu. Toto pole je na obrazovke vypísané inverzné.

SZAPC - indikátory /SIGN, ZERO, AUXILIARI CARY, PARITY, CARRY/. Příslušné písmeno znamená, že zodpovedajúci indikátor je rovný 1, ináč je na danej pozícii znak -.

AA - hexadecimálna hodnota registra A
BBCC - hexadecimálna hodnota registrov BC /v tomto poradí/
DDEE - hexadecimálna hodnota registrov DE /v tomto poradí/
HHLL - hexadecimálna hodnota registrov HL /v tomto poradí/
XHXL - hexadecimálna hodnota registra IX
YHYL - hexadecimálna hodnota registra IY
SPSP - hexadecimálna hodnota registra SF

Základná informácia sa vypisuje vždy do dialógového riadku. Užívateľ však môže zabezpečiť opis dialógového riadku do obrazovky pomocou klávesy L tak, ako je to popísané ďalej.

2.7.2 Zadanie hodnoty veličiny pre ladiaci program.

Vo všetkých prípadoch keď užívateľ zadáva hodnotu nejakej veličiny má k dispozícii tieto možnosti

- decimálna konštantna
- hexadecimálna konštantna
- návestie
- ľubovoľná kombinácia uvedených tr磘och. Prvkov spájaných operáciami + alebo - s výnimkou, keď návestie označuje vstupný bod v knižnici. V tomto prípade musí byť návestie posledný člen výrazu.

Fritom na syntax zápisu sú kladené rovnaké požiadavky ako pri písaní symbolu v module EDI. Syntakticky nesprávny zápis ladiaci program odmieta zvukovým signálom a dá možnosť ho opraviť.

Ak užívateľ uadal hodn tu ako návestie, ladiaci program najprv prezrie tabuľku symbolov posledne prekladaného zdrojového textu a ak tam daný symbol nenájde, tak prehľadá tabuľku vstupných bodov v binárnej knižnici. Ak sa ani tam daný symbol nevyskytuje, zaznie zvukový signál a je m žné chybný symbol opraviť. Daná veličinazíska hodno-

tu príslušného symbolu, inými slovami napr. register FC môžeme mastaviť na adresu ľubovoľného návestia v po siedne prekladanom module alebo na adresu ľubovoľného vstupného bodu pomocou jeho symbolického mena a bez toho, aby sme sa museli staráť o zloženie programu v pamäti.

Na editovanie vstupu možno využiť rovnaké prostriedky ako pri práci s dialogovým riadkom.

Ak sa odošle prázdny riadok príslušná veličina sa nemení /pozri napr. príkaz pre modifikovanie pamäte M/.

2.7.3 Příkazy pre ladící program

Příkazy pro ladící program jsou jednopísmeňové. Po napsání příslušného znaku se příkaz hned vykoná, než teda možné opravit chybný příkaz. Odoslání znaku, kterému není přiřazený žádoucí příkaz nevykoná nič.

Následuje popis příkazov. Způsobů ich použití pozri v odseku 2.7.4.

R - příkaz pro nastavenie hodnôt registrov. Po jeho napsání systém očakáva meno registra, teda jeden zo znakov **P, A, B, D, X, Y, S**. Ak napiše iný znak, ladiaci program reaguje zvukovým signálom a očakáva ďalší příkaz. Po napsání správneho mena registra sa dialogový riadok vymaže, vypíše sa

RN:

kde **N** je meno registra, ktorý sa bude modifikovať. Modifikovať možno len **16-BITOVÉ** registre /napr. nie len C register, ale vždy celú dvojicu BC/. Ak sa odošle prázdny riadok, hodnota príslušného registra sa nezmiení.

Fokial užívateľ inicializuje register **P**, automaticky sa inicializuje aj register **SP** na hodnotu, ktorú mal pri spustení ladiaceho programu.

I - příkaz pre nastavenie bodu prerušenia. Ladiaci program očakáva hodnotu zadanú podľa bodu 2.7.2 a na adresu danú touto hodnotou nasadí bod prerušenia. Potom do dialogového riadku vypíše

N:

a očakáva decimálne číslo, ktoré udáva kolkokrát musí program prejsť cez bod prerušenia, kým sa jeho vykonávanie zastaví. Ak sa hodnota neudá, program sa preruší hned, keď narazí na bod prerušenia. Ak sa udá napr. 1000 tak sa program preruší keď narazí na bod prerušenia 10001 krát. Okrem toho sa pri každom prechode bodom prerušenia testuje kláves CS/MĽDZERA a ak je stlačený vylonávanie programu sa preruší.

Pri prerušení sa riadenie odovzdá ladiacemu programu, ktorý pokračuje výpisom záladného stavu a čí-

taním príkazu. Pri výpise základného stavu je na prvej pozícii znak '**B**' /**BREAK**/, ktorý signalizuje, že program narazil na bod prerušenia. Vo vykonávaní programu možno pokračovať buď po odstránení bodu prerušenia príkazom **O** alebo po vykonaní jednej inštrukcie príkazom **S** /pozri ďalej/. V druhom prípade bod prerušenia v programe ostal a ak program pri ďalšom vykonávaní naň narazi, opäť sa preruší jeho vykonávanie. Počet prechodov bodom prerušenia musíme opäť špecifikovať, pretože ostal nastavený na hodnotu **O**.

Užívateľ môže špecifikovať najviac jeden bod prerušenia. Pri zadaní nového sa starý automaticky vymaže.

Na bod prerušenia program reaguje bez ohľadu na spôsob, akým bol spustený, „pravda pri príkaze **G** len pre programy, ktoré nie sú v pamäti **ROM**.“

Bod prerušenia je realizovaný pomocou inštrukcie **RST 10** a využíva systémovú premennú **CURCHL** na adrese #**5C51**. Ire užívateľa je teda táto pamätná pri práci s ladiacim programom nepoužiteľná.

- O** - príkaz pre odstranie bodu prerušenia.
- G** - príkaz na spustenie programu. Registre sa inicializujú na hodnoty, ktoré sú vidno pri výpise základného stavu a program sa odštartuje. Pokial program úspešne prebehne až po poslednú inštrukciu návratu, riadenie sa vráti ladiacemu programu. Register **PC** je nastavený opäť na začiatok programu, čiže program možno opäť spustiť príkazom **G**.
- CS/G** - príkaz má rovnakú funkciu ako príkaz **G**, ale vykonávanie programu sa preruší pri prvom prechode bodom prerušenia bez ohľadu na počet prechodov.

Systém MRS beží so zakázanými prerušeniami a prácu s klaviatúrou vykonáva vlastným modulom. Ireto po príkaze **C** musí užívateľ sam vykonať inštrukciu **FI**, pokial je to potrebné. Pri návrate do ladiaceho programu sa opäť preruší zákaz. Upozorňujeme, že pri povolených prerušeniach musí byť register **IX** rovný # **5C3A** a na tento hodnotu ho ladiaci program inicializuje.

-
- S** - príkaz na výkonanie jednej inštrukcie. Vykoná sa jedna inštrukcia s hodnotami registrov aké sú vidno pri výpise základného stavu. Po jej vykonaní sa riadenie vráti ladiacemu programu. Výpis základného stavu začína znakom '**S**' /**STEP**/, ktorý signalizuje krokovací režim. Ako užívateľ zabezpečí opis dialogového riadku do obrazovky /klávesom **L**/, získa podrobnejší prehľad o vykonávaní svojho programu.

Pred vykonaním sa inštrukcia analyzuje, či neporušuje niektorú, užívateľom stanovenú ochranu /pozri odsek 2.7.4/.

Analyzovanie inštrukcie typu **LDIR** môže trvať až **20** sekúnd, je to možné odstrániť krokováním pomocou klá-

vesu CAPS SHIFT/S /pozri ľalej/.

Krokovať možnò aj programy v pamäati ROM

CS/S - krokovanie programu s ignorovaním ochrán /pozri odsek 2.7.4/.

C - príkaz má rovnakú funkciu ako príkaz S, ale krokovanie programu sa vykoná, že za inštrukciu sa vloží bod prerušenia a inštrukcia sa vykoná normálne. Ten-to interný bod prerušenia nesúvisí s bodom prerušenia, ktorý zadal užívateľ okrem prípadu, že bod prerušenia zadaný užívateľom je v podprograme, ktorý krokuje príkazom C. V tomto prípade sa vykonávanie preruší v podprograme, ale už nie po výstupe z neho.

Význam príkazu je predovšetkým v tom, že odladené podprogramy volené inštrukciou CALL sa vykonajú v normálnom rýchlorežime. Podobne ako pri príkaze I / INSERT BREAK/ sa funkcia nedá použiť na programy v pamäti ROM. Rovnako ako pri krokovanií príkazom S sa inštrukcia pred vykonaním analyzuje.

CS/C - má rovnakú funkciu ako príkaz C, ale inštrukcia sa vykoná aj keď analýza zistila narušenie niektornej ochrany.

CS/J - týmto príkazom sa inštrukcie vypisujú, ale **nevýkonávajú se** /v tomto prípade sa samozrejme ani nič nekontroluje/. Register PC sa nastaví na ďalšiu inštrukciu. Príkaz CS/J slúži na preskočenie inštrukcii, ktoré sa nemajú vykonáť /napr. chceme ukončiť prečasné príkaz cyklu, teda nevykonáme inštrukciu DJNZ, ale ju preskočíme príkazom CS/J/.

T - program sa vykonáva v sledovacom režime. Jedná sa vlastne o opakovanie krokovací režim. Ladiaci program každú inštrukciu pred vykonaním interpretuje a zistí či je prípustná. Prípustné inštrukcie vykoná, vypíše základný stav do dialogového riadku, ale bez výpisu inštrukcie v symbolickom stave, pretože túto informáciu v sledovacom režime nestihne užívateľ v dialogovom riadku sledovať. Pravda ak tieto riadky '**T**' /TRACE/, ktorý signalizuje sledovací režim a obsahuju aj výpis z inštrukcie v symbolickom tvare. Popis možnosti, ktoré poskytuje sledovací režim pozri v odseku 2.7.4.

N - program sa vykonáva v zrychlenom sledovacom režime. Možnosti tohto režimu sú rovnaké ako pri príkaze T, len sa po vykonaní inštrukcie nevypisuje základný stav, čím sa značne urýchli vykonávanie sledovaného programu.

W - príkaz umožňuje nastaviť pamäťové okná a okná pre register PC. Po zadaní príkazu ladiaci program očakáva znak 'M' /MEMORY/, ak sa budú definovať pamäťové okná alebo znak 'P' /program COUNTER/, ak sa budú definovať okná pre register PC. Ak užívateľ napiše iný znak, ladiaci program prejde do základného stavu, ináč vypíše hodnoty príslušných okien - štyri dvojice hexadecimálnych čísel, kde každá dvo-

jica definuje jedno okno. Ak užívateľ zmení niektoré z týchto ôsmich čísel, zadá jeho poradové číslo pomocou číslice v rozsahu 1-8. Po napišaní iného znaku prejde ladiaci program do základného stavu, ináč očakáva hodnotu príslušnej hranice, ktorá sa zada podľa odseku 2.7.2. Po jej napišaní prejde ladiaci program do základného stavu.

Funkcia pamäťových okien je vysvetlená v odseku 2.7.4.

P - príkaz na zobrazenie pamäťového okna. Ladiaci program vypíše do dialogového riadku text

P:

a čaká hodnotu podľa zásad popísaných v odseku 2.7.2.
Potom vypíše

L:

a očakáva desiatkové číslo. Adresa definuje začiatok pamäťového okna a číslo L udáva jeho dĺžku v štvoriciach BYTOV /inýmislovami počte riadkov výpisu pamäte/. Ak sa v ľubovoľnom sledovacom režime zmení obsah tohto pamäťového okna zobrazí sa na obrazovku výpis pamäte od danej adresy L riadkov /pozri tiež príkaz D/.

Ak sa namiesto čísla stlačí len ENTER ruší sa funkcia zobrazovania pamäťového okna.

M - príkaz na modifikovanie obsahu pamäti. Ladiaci program vypíše do dialogového riadku text

M:

a čaká hodnotu podľa zásad popísaných v odseku 2.7.2.
Potom do dialogového riadku vypíše adresu a obsah štyroch BYTOV od tejto adresy vrátane. Výpis je jednak hexadecimálny a jednak znakový. Užívateľ môže modifikovať pamäť zadávaním hexadecimálnych konštánt a využíva nasledujúce riadiace klávesy:

- CAPS SHIFT/5 - prechod na predchádzajúci znak. Medzery, ktoré oddelujú BYTY sa automaticky skáču, ak je jazdec na prvom znaku posunie sa výpis o jeden znak smerom k nižšej adrese.
- CAPS SHIFT/8 - prechod na nasledujúci znak. Medzery, ktoré oddelujú BYTY sa automaticky skáču. Ak je jazdec na poslednom znaku, posunie sa výpis o jeden znak smerom k vyšším adresám.
- CAPS SHIFT/1 - zobrazia a modifikujú sa predchádzajúce 4 BYTY.
- CAPS SHIFT/4 - zobrazia a modifikujú sa nasledujúce 4 BYTY.
- CAPS SHIFT/SYMBOL SHIFT - BYTY modifikované na obrazovke sa vrátia do pôvodného stavu. Všeobecne do pamäti sa zapíše modifikovaná štvoricu znakov až vtedy, keď sa posunú po

obrazovke tak, že niektorý z nich zmizne z obrazovky.

ENTER modifikované BYTY sa uložia do pamäti a riadenie sa vráti ladiacemu programu, ktorý vypíše základný stav

ASCII znak reprezentujúci hexadecimálnu číslicu (0-9 A-F) prepíše číslicu, na ktorej je jazdec. Jazdec sa posunie o jeden pozíciu pričom sa medzery skáču.

MEDZERA tento znak je akceptovaný len keď je jazdec na prvom znaku dvojice. Systém po ňom očakáva ďalší znak. Jeho kód sa uloží na miesto označené jazdcom.

A stlačenie inej klávesy je chyba. Zaznie zvukový signál a nič sa nevykoná.

Príkaz M /a príkaz D/ najčastejšie využíva tú vlastnosť vstupu, že odoslanie prázdnego riadku nezmiení čítanú hodnotu. Inými slovami ak napr. príkazom M:#4000 prezeráme pamäť od adresy #4000, tak ak najbližšie príkaz M: ukončíme hned klávesom ENTER, opäť sa zobrazí pamäť od #4000.

D - príkaz na zobrazenie pamäte. Systém vypíše do dialogového riadku

D:

a zobrazí na obrazovku hexadecimálny a znakový výpis 64 BYTOV od adresy, ktorú zadal užívateľ. Ak odoskal prázdný riadok, tak sa použije naposledy zadaná adresa. Výpis riadi užívateľ týmito klávesami

Klávesom CAPS SHIFT/6 sa zobrazí nasledujúcich 64 BYTOV.

Klávesom CAPS SHIFT/7 sa zobrazí pre chádzajúcich 64 BYTOV

Klávesom **A** špecifikuje znakový retazec (pozri ďalej)
Klávesom **H** špecifikuje retazec hexadecimálnych konštánt (pozri ďalej)

Klávesom **ENTER** sa riadenie vráti ladiacemu programu.

Iné klávesy sú považované za chybné.

Ak užívateľ použije príkaz A (resp. H) systém vypíše do dialogového riadku

A: (resp. **H**)

bez uvozovek a malými písmenami
a očakáva retazec znakov (resp. hexadecimálnych konštánt).
Pri ich zadávaní je možné opravovať chyby klávesom CS/Ø,
retazec sa ukončí klávesom ENTER. Pri chybnej zadanej retazci zaznie zvukový signál.

Editor vyhľadá retazec v pamäti a urobí jej výpis na obrazovku. Ak sa daný retazec v pamäti nenájde zaznie zvukový signál. Ak sa pri hľadaní retazca zistí koniec pamäte systém automaticky pokračuje od začiatku.

Ak sa zadá prázdný retazec systém vyhľadá nový výskyt prv zadaného retazca. Ak ešte neboli retazec zadany hľadá s výskyt prvého BYTU výpisu pamäte.

A - príkaz na spätný preklad s výpisom na obrazovku. Do dialogového riadku sa vypíše text

A:

a na obrazovku sa zobrazí spätný preklad ²² riadkov od adresy, ktorú zadá užívateľ. Opäťovným stláčaním klávesu **A** sa vypíše vždy ďalší riadok spätného prekladu. Stláčením klávesu ENTER sa systém vráti do základného stavu.

X - tento klaves slúži ako prípínač. Jeho opäťovným stláčaním vymieňame sadu registrov za alternatívne registre. Inými slovami stlačením klávesu X vykonáme inštrukcie EXX aj EX AF, AF'.

L - týmto príkazom určí užívateľ kolko riadkov sa má zobraziť na obrazovke. Systém do dialogového riadku vypíše

N: L:

a očakáva počet riadkov. Pred výpisom základného stavu sa potom posúva obrazovka o 2 riadky hore, čím sa prechádzajúci stavový riadok opíše do obrazovky pričom sa na to využíjte N dvoj riadkov obrazovky. Ak N=0 používa sa len diaľkový riadok, ale v režime T sa vypisuje aj inštrukcia v zdrojovom tvare. Ak sa namiesto čísla stlačí ENTER tak sa táto vlastnosť zruší.

Q - *návrh do mre*

2.7.4 Ladenie programu

Sledovací režim (príkazy **T** alebo **N**) je veľmi silný prostriedok na ladenie programov. Pri jeho návrhu bol sledovaný hlavný cieľ - užívateľ v sledovacom režime nesmie stratiťvládu nad svojim programom. Kontrolu nad programom môže stratiť troma spôsobmi:

- prepísaním pamäti
- vykonaním zakázanej inštrukcie
- zacyklením programu

Ladiaci program teda musí poskytnúť užívateľovi možnosť chrániť lubovoľnú časť pamäte pred prepísaním, filtrovať niektoré inštrukcie a umožniť prerušenie programu. Toto je dosiahnuté nasledovne:

- v sledovacom režime nie je možné prepísať adresy pamäte, ktoré sa nachádzajú v niektorom pamäťovom okne. Ak teda máme niektoré pamäťové okno nastavené ako interval AAAA BBBB, tak adresy od AAAA do BBBB včítane nie je možné v sledovacom režime prepísať. Pri pokuse o ich modifikovanie sa riadenie vráti ladiacemu programu, ktorý vypíše základný stav. Tento výpis začína znakom '**M**', /MEMORY/, ktorý signalizuje že došlo k počusu modifikovať chránenú pamäť.

Ak užívateľ spustil program pomocou príkazu editora ALD, je prvé pamäťové okno inicializované tak, že je chránený zdrojový text užívateľa.

- v základnom modulu knižnice je 16 BYTOV, pomocou ktorých užívateľ môže definovať ilegálne inštrukcie. Ak pri vykonávaní programu príkazom S, C, T alebo N ladiaci program zistí, že ná vykonáť inštrukciu, ktorá je v tomto zozname, vykonávanie programu sa preruší a do diaľkového riadku sa vypíše základný stav, pričom sa nastaví status = '**I /ILLEGAL/**,

Uvedených 16 BYTOV je pod vstupným bodom II, teda užívateľ definuje ilegálne inštrukcie normálne príkazom MODIFY (M:II). Zoznam ilegálnych inštrukcií musí byť zakončený inštrukciou NOP (00) ktorá teda nemôže byť ilegálna.

Inštrukcie sú zadávajú jedným BYTOM (operačným znakom) okrem inštrukcií s prefixom #ED, ktoré sa zadávajú dvoma BYTMI (teda #ED a ďalej jeden BYTE). Teda ako ilegálnu inštrukciu možno definovať celú triedu inštrukcií s prefixom #CB (rotácie, posuny BYTOVÉ operácie), inštrukcie, ktoré pracujú s regiszrom IX (PREFIX #DD), ale aj napr. konkrétnu inštrukciu LDIR (BYTY #ED, #DO) alebo inštrukciu EXX (BYTE #DE9)

V základnom module SYSMOD je ako ilegálna definovaná inštrukcia **HALT (#76)**.

Inštrukcie definované ako ilegálne možno vykonať príkazmi CS/S alebo CS/C.

Okrem ilegálnych inštrukcií definovaných užívateľom sa rovnakým spôsobom využívajú BYTY, ktorých kombinácia nemá pre procesor Z80 význam (napr. BYTY #DD, #FD). V takomto prípade je tiež status vo výpise ladiaceho programu '**I /ILLEGAL/**', ale krokováním s vypustením kontroly (príkazmi CS/S alebo CS/C) sa takáto ilegálna inštrukcia nevykoná, ale sa preskočí jej BYTE.

- pred vykonaním každej inštrukcie v sledovacom režime zistuje ladiaci program, či neboli stlačený kláves **CS/MEZERA**. Ak áno vykonávanie programu sa preruší a ladiaci program prejde do základného stavu. Týmto klávesom je teda možné prerušiť program počas vykonávania a pokračovať ďalej krokováním alebo spustiť normálne program príkazom **G**.
- obdobnú funkciu ako pamäťové okná majú aj okná pre register PC. Ich využitie rie je také kritické ako využitie pamäťových okien, keďže ladiaci program vie sledovať alebo krokovat aj programy v pamäti ROM. To znamená, že aj keď má register PC akúkolvek hodnotu, neznamená to stratu kontroly nad programom. Aj tak je použitie okien pre PC užitočné, užívateľ pomocou nich môže realizovať ďalšie body ba dokonca celé oblasti prerušenia. Na rozdiel od pamäťových okien, ktoré zakazujú vykonanie inštrukcie, okná pre register PC ich naopak povolujú. To znamená, že inštrukcia sa vykoná len vtedy, ak hodnota regisra PC padne aspoň do jedného okna. Pri spustení programu je prvé okno pre register inicializované na interval 0000-FFFF, teda register PC môže nadobúdať akékolvek hodnoty.

Kapitola 3

Závěr

Systém MRS bol vyvinutý na Ústave aplikovanej kybernetiky. Pre ZX SPECTRUM je volne prístupný ale distribuovať je povolené len verziu oficiálne schválenú ažen za podmienok uvedených v copyrighte (program zadarmo). Táto príručka aj vlastný program sú záujemcom prístupné v T-klube SZM Bratislava.

Rovnako je zakázané prenášať program na iné systémy bez súhlasu Ústavu aplikovanej kybernetiky. Autori MRS sú ochotní preniesť programové vybavenie do iného technického prostredia za predpokladu, že daný počítač bude zapožičaný do Ústavu aplikovanej kybernetiky a že je možnosť prenosu súborov do tohto počítača či už cez sériový alebo paralelný interface alebo nejaké nosné médium.

Ešte mi padá milá povinnosť spomenúť všetkých, ktorí sa na tvorbe systému podielali (predovšetkým členom Elámu klubu FV SZM EF SVŠT).

Ivan Jedlička navrhol systém a realizoval moduly EDI, LNK a LJB.

Igor Titka realizoval modul DBG.

Milan Tvarožek realizoval modul ASM a tie časti modulu EDI, ktoré súvia sia s prekladačom (vytváranie zhosteného formátu zdrojového textu a ich prevod do textového tvaru, spätný preklad).

Ivan Fedorko realizoval modul práce s klaviatúrou.

Peter Ondriš a Marián Přikryl realizovali paralelný interface pre ZX SPECTRUM čím vlastne umožnili, aby sa MRS do SPECTRA vôbec dostalo.

Miroslav Adámy našiel v pracovnej verzii viac chýb ako hodnotil iný.

Róbert Vašíček dal užívateľskej príručke jej pekný tvar.

Poďakovanie patrí aj vedeniu ústavu aplikovanej kybernetiky, ktoré celú prácu umožnilo a T-klubu SZM, ktorý sa stará o propagáciu a distribúciu programu.

Prípadné ďalšie informácie o systéme ako aj o ďalších reálizovaných programoch môže podať

RNDr. Ivan Jedlička
Ústav aplikovanej kybernetiky
Danulova 5/A
844 16 Bratislava

a
T-klub SZM
ul. Vazovova 1
812 19 Bratislava

Příloha A
příloha

B N D C

přeměna 16 bitové hodnoty na řetězec
decimálních číslic

BC = 16 bitová hodnota

HL = adresa řady které se má uložit výsledek

BNDC	LD	A,1
BDO:	LD	D,B
	LD	E,C
	LD	C,A
	EX	DE, HL
	PUSH	BC
BD1:	XOR	A
	LD	B,16
BD2:	ADD	HL,1L
	ADC	A,A
	CP	10
	JR	C, BD3
	SUB	10
	INC	L
BD3:	DEC	B
	JR	NZ, BD2
	OR	30H
	PUSH	AF
	LD	A,L
	OR	H
	JR	NZ, BD1
	EX	DE, HL
	INC	A
	CP	C
	SBC	A,A
	LD	B,A
BD4:	POP	AF
	RET	C
	LD	(HL), A
	ADD	HL, BC
	JR	BD4
BNHX2:	LD	A,D
	CALL	BNHX
	LD	A,E

B N H X

přeměna 8 bitové hodnoty na číslo
hexadecimálního řetězce

A = 8 bitová hodnota

HL = adresa, na kterou se má uložit výsledek

BNHX:	PUSH AF	
	RRCA	
	CALL BNHX1	
	POP AF	
BNHX1:	AND #FF	maska - vymazuje bity DH až DF
	ADD A, #90	#9X
	DAA	9X, [x < A]
	ADC A, #40	→ D(X+6) [X > 9]
	DAA	→ 4 (X+6+1)
	OR #20	#6 (X+6+4)
DNUM:	LD (HL), A	→ 6 (x+6+1), → ASCII číslice → ASCII písmene

INC HL
RET

H X B N

#F328 HXBN: LD DE, Ø
LD A, (HL)
CALL HXBN1 #F33B
RET C
EX DE, HL
ADD HL, HL
ADD HL, HL
ADD HL, HL
ADD HL, HL
EX DE, HL
OR E
LD E, A
INC HL
JR HXBN+3
#F33B HXBN1: ADD A, #C6
JR C, HXBN2
SUB #F6
RET
#F342 HXBN2: AND #DF
SUB 7
RET C
ADD A, #FA
RET C
SUB #FO
RET

#F34C-----

#F34D DCBN: LD DE, ØØØØ
LD A, (HL)
CB #3A
RET NC
SUB #3Ø
RET C
EX DE, HL
PUSH BC
ADD HL, HL
LD B, H
LD C, L
ADD HL, HL
ADD HL, HL
ADD HL, BC
LD C, A
LD B, Ø
ADD HL, BC
POP BC
EX DE, HL
INC HL
JR DCBN+3 t.j. #F35Ø

D C B N

Konverze řetezce decimalních čísel na 16 bitovou hexadecimální hodnotu

HL = adresa začátku řetezce

ML = adresa konca řetezce (ukončov. znak)

DE = konvertovaná hodnota (hexadecimálně

návrat, když ASCII kod neje kodem

čísla

v reg. A je dekadické číslo (Ø-9)

: Konstanty a data pre ďalšie moduly

KPORT	EQU	#FE	pont kľúčnice na adresu #FE (254)
SHF1	EQU	1	
SHF2	EQU	37	
MAXC	EQU	64	
MAXCS	EQU	31	
MAXR	EQU	23	
CHLEFT	EQU	#19	
CHRHT	EQU	#18	
CRLF	EQU	#OD	
PE	EQU	#OC	
LF	EQU	#OA	
SYSBEP	EQU	#3B5	
KREPT:	DB	0	
LASTK:	DB	0	
KTABO:	DB	0, 'ZXCVASDFGQWERT12345'	
	DB	'09876POIUY', ODH, 'LKJH'	
	DB	', , 0, 'MNB'	
KTAB1:	DB	0, 'ZXCVASDFGQWERT'	
	DB	#1E, #1F, #1C, #1D, #19	
	DB	#13, #12, #18, #17, #1A	
	DB	'POIUY', 5, 'LKJH'	
	DB	#11, #07, 'MNB'	
KTAB2:	DB	0, #3A, 0, #3F, #2F; 0, 0, 0, 0, 0	
	DB	0, 0, 0, #3C, #3E, #%	
	DB	,) (, #27, , , , , 0, 0, 0	
	DB	3, , =+-, , #14, 0, , , 0	
POSCUR:	DW	#5800	
CURSOR			
ROW:	DS	1	
COLUMN:	DS	1	

Rôzne pomocné pod programy

#D5BEABSCRS:	LD	A, H	program na vloženie čísla 12345678 do pamäti
	RRCA		prvá číslica je zadaná v registri D5BE, posledné tri sú zadané v registriach D5BF, D5C0, D5C1
	RRCA		
	RRCA		
	AND	#E0	vloženie čísla 00000000
	OR	L	zoplnenie čísla očkovacou
	LD	L,A	
	LD	A,H	
	AND	#18	zloženie čísla (0-7) príklad 3 bloky čísel 00000000
	OR	#40	zloženie čísla (8-9) príklad 3 bloky čísel 00000000
	LD	H,A	
#D5CC	RET		
*#B5CDSETCHR:	LD	HL, CURSOR (# FE42)	
#D5D0SETCHO:	PUSH	AF	
	CALL	ABS CRS (# D5BE)	
	POP	AF	
X#D5D5PIXEL	PUSH	HL	

	PUSH	DE
	PUSH	BC
	EX	DE, HL
	LD	L, A
	LD	H, O
	ADD	HL, HL
	ADD	HL, HL
	ADD	HL, HL
	LD	BC, #3C00
	ADD	HL, BC
	LD	B, 8
PX1:	LD	A, (HL)
	LD	(DE), A
	INC	D.
	INC	HL
	DJNZ	PX1
	POP	BC
	POP	DE
	POP	HL
	RET	
:		
IID:	PUSH	HL
	PUSH	PC
	LD	A, 8
IID1:	LD	BC, 32
	PUSH	DE
	PUSH	HL
	LDIR	
	POP	HL
	POP	DE
	INC	H
	INC	D
	DEC	A
	JR	NZ, IID1
	POP	BC
	POP	HL
	RET	
:		
IDELLN:	LD	HL, (CURSOR)
	SUB	H
	RET	Z
	LD	B, A
	LD	L, O
	CALL	ABSCRS
IIO:	LD	D, H
	LD	A, L
	ADD	A, 32
	LD	E, L
	LD	E, A
	JR	NZ, III
	LD	A, 8
	ADD	A, H
	LD	H, A
III:	CALL	IID
	DJNZ	IIO

```

        RET
:
IINSJN:    LD      A, MAXR-1
             LD      HL, (CURSOR)
             SUB    H
             RET    Z
             RET    C
             LD      HL, #50A0
             LD      DE, #50C0
             LD      B, A
IDO:       CALL   IID
             EX      DE, HL
             LD      A, E
             SUB    32
             LD      L, A
             LD      H, D
             JR      NC, ID1
             LD      A, D
             SUB    8
             LD      H, A
ID1:      DJNZ   IDO
             RET

:
SCROLL:   LD      HL, (CURSOR)
SCROL1:  LD      A, MAXR
          PUSH   HL
          PUSH   DE
          LD      HL, Ø
          LD      (CURSOR), HL
          CALL   IDELLN
          POP    DE
          POP    HL
          LD      (CURSOR), HL
          RET

:
:
:
#D73DSCAN: PUSH   BC
             LD      BC, KPORT #FFFE
             IN      A, (C) Čtení portu. Stisk je reprezent. log. Ø
             POP    BC
             CPL    komplementace reg. A
             AND    #1F
             RET
             PUSH   Z
             PUSH   BC
             PUSH   DE
             PUSH   HL
             LD      BC, KPORT+#FE00 # FFFE
             LD      HL, Ø
             LD      E, 1
KS3:       IN      A, (C) Čtení portu
             LD      D, 5
KS4:       RRCA
             JR      C, KS1 # D760 SKOK, je-li stiskn. CS
             INC    H
             DEC    h

```

S C A N

D760KS1:

JR	NZ, KERR
LD	H, L
LD	L, E
INC	E
DEC	D
JR	NZ, KS4
RLC	B
JR	C, KS3
EX	DE, HL
LD	HL, KTAB0-1
INC	D
DEC	D
JR	Z, KTE
INC	E
DEC	E
JR	Z, ADDD
LD	HL, KTAB1-1
LD	A, SHF1
CP	D
JR	Z, ADDE
CP	E
JR	Z, ADDD
LD	HL, KTAB2-1
LD	A, SHF2
CP	D
JR	Z, ADDE
CP	E
JR	Z, ADDD
KERR:	XOR A
- KRET:	POP HL
	POP DE
	POP BC
	RET
ADDD:	LD E, D
ADDE:	LD D, G
	ADD HL, DE
	LD A, (HL)
	JR KRET
KTE:	INC E
	DEC E
	JR Z, KERR
	JR ADDE

:

:

:

INKEY:	PUSH BC
INKEY1:	CALL SCAN
	OR A ----- je A = Ø ?
	JR NZ, IN3
	LD (LASTK), A
	JR INKEY1
IN3:	LD BC, #1000
WAIT1:	DEC C
	JR NZ, WAIT1
	DJNZ WAIT1
	LD C, A
	CALL SCAN
	CP C

INKEY d79c

```

JR      NZ, INKEY1
LD      A, (LASTK)
CP      C
JR      Z, REPEAT
LD      A, 250
JR      SIN1
REPEAT: LD      A, (KREPT)
LD      B, A
REP1   XOR    A
REP2:  DEC    A
JR      NZ, REP2
CALL   SCAN
CP      C
JR      NZ, INKEY1
DJNZ   REP1
LD      A, 25
SIN1:  LD      (KREPT), A
LD      A, C
LD      (LASTK), A
PUSH   HL
PUSHI  DE
PUSH   AF
LD      HL, 80
LD      DE, 20
CALL   SYSBEP
POP    AF
POP    DE
POP    HL
POP    BC
RET
:
:
:
:
```

O U T C H R

```

# D9FC OUTCHR: CP      LF      CP #04
                  RET    Z
                  CP     PE
                  RET    Z
                  PUSH   HL
                  LD     HL, (CURSOR)
                  CP     #20
                  JP     C, SPEC
                  PUSH   HL
                  CALL   SETVHR
                  POP    HL
                  INC    L
                  LD     A, MAXCS+1
                  SUB    L
                  JR     NZ, NEWC
NEWL:   LD     L, A
                  LD     A, H
                  SUB    MAXR-1
                  ADC    A, MAXR-1
                  LD     H, A
                  CALL   NC, SCROL1
NEWX:   LD     (CURSOR), HL

```

	POP	HL
	RET	
SPEC:	SUB	CRLF
	JR	Z, NEWL
	LD	DE, NEWC
	PUSH	DE
	SUB	CHLEFT-CRLF
	JR	Z, LVLAVO
	SUB	CHRGHT-CHLEFT
	RET	NZ
	LD	A, MAXC-2
	CP	L
	RET	C
	INC	L
	RET	
LVLAVO:	DEC	L
	RET	P
	INC	L
	RET	

INVPC:	PUSH	HL
	PUSH	BC
	CALL	ABSCRS
	LD	B, 8
IPX1	LD	A, (HL)
	CPL	
	LD	(HL), A
	INC	H
	DJNZ	IPX1
	POP	BC
	POP	HL
	RET	

SETCRS:	PUSH	HL
	LD	HL, (POSCUR)
	RES	7, (HL)
	POP	HL
	PUSH	HL
	PUSH	DE
	LD	A, L
	CP	MAXCS+1
	JR	C, SCR1
	LD	A, MAXCS
SCR1:	LD	D, #58
	LD	E, A
	LD	L, H
	LD	H, O
	ADD	HL, HL
	ADD	HL, DE
	POP	DE
	LD	(POSCUR), HL

I N V P I X

Invertuje tabuľku s kódmi obrazovky
H = Farbačka 0-33
L = Farbačka 0-34

S E T C R S

SET 7, (HL)
POP HL
LD (CURSOR), HL
RET

#d6d2 CLEAR:

LD HL, #4000
LD DE, #4001
LD BC, #1800
LD (HL), 0
LD LDIR
LD (POSCUR), HL (#FE44)
LD (HL), #30
LD BC, #300-1 #D2FF
LD IR
RET

#D6E9 -----

C L E A R

ADDRESS Addressovat b22
adresat 1 byte naceti
pojet b22