Vysoká škola: **VŠST Liberec** Katedra: **technické kybernetiky**

Fakulta: **strojní** Školní rok: **1980/81**

DIPLOMOVÝ ÚKOL

pro **s. Ivana D e l e ž a l a**

obor **automatizované systémy řízení výrobních procesů ve strojírenství**

Protože jste splnil..... požadavky učebního plánu, zadává Vám vedoucí katedry ve smyslu směrnic ministerstva školství xxxxxxx o státních závěrečných zkouškách tento diplomový úkol:

Název tématu: **Přídavné moniterové programy mikroprocesoru**

Pokyny pro vypracování:

1. **Seznamte se s literaturou zaměřenou na činnost mikroprocesorů a jejich programování.**
2. **Navrhněte a vyzkoušejte přídavné moniterové programy pro malý mikroprocesorový vývojový systém s mikroprocesorem I 8080.**
3. **Navrhněte a realizujte připojení elektrické psací jednotky CONSUL 256 k mikropočítači osazeném mikroprocesorem I 8080.**
4. **V závěru zhodnoťte ekonomický přínos budování mikroprocesorových vývojových systémů.**

Rozsah grafických laboratorních prací:

Rozsah průvodní zprávy: **50 - 60 stran**

Seznam odborné literatury:

1. **Časopisy Sdělovací technika, ročníky 1978, 1979, 1980**
2. **Časopisy Automatizace, ročníky 1979, 1980**
3. **TK 80 User's Manual**
   firemní literatura firmy NEC, 1975

a další dle pokynů konzultanta a vedoucího.

Vedoucí diplomové práce: **Ing. Václav S e d l i c k ý**

Konsultanti: **Ing. Josef G r e s m a n**

Datum zahájení diplomové práce: **15.9.1980**

Datum odevzdání diplomové práce: **12.6.1981**

VYSOKÁ ŠKOLA STROJNÍ A TEXTILNÍ LIBEREC Fakulta strojního inženýrství

Doc.Ing.J.Alaxin,CSc.
Vedoucí katedry

Doc.RNDr B.Stříž,CSc.
Děkan

V **Liberci** dne **15.9.** 19**80**

Místopřísežně prohlašuji, že jsem diplomovou práci vypracoval samostatně s použitím uvedené literatury.

V Liberci dne 12. 6. 1981

Ivan Doležal

VYSOKÁ ŠKOLA STROJNÍ A TEXTILNÍ V LIBERCI

nositelka Řádu práce

Fakulta strojní

Obor 23-40-8

automatizované systémy řízení

výrobních procesů ve strojírenství

Katedra technické kybernetiky

PŘÍDAVNÉ MONITOROVÉ PROGRAMY MIKROPROCESORU

IVAN DOLEŽAL

Vedoucí práce: ing. Václav Sedlický, VŠST Liberec

Konzultant: ing. Josef Grosman, VŠST Liberec

KTK - ASRSF - 002

Rozsah práce a příloh

Počet stran 71

Počet příloh a tabulek 9

Počet obrázků 8

Počet výkresů 3

Počet modelů nebo jiných příloh . . . 0

12. 6. 1981

OBSAH

SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

MVS	mikropočítačový vývojový systém
TWS	přídavný monitor
JSA	jazyk symbolických adres
DMA	přímý přístup do paměti
I/O	vstupně-výstupní
IO	integrovaný obvod
A	akumulátor (zákl. registr) mikroprocesoru 8080
B,C,D,E,H,L	registry " "
F	registr příznaků " "
BC,DE,HL	registrové páry " "
CY,Z,S	(některé) příznaky " "
PC	programový čítač
SP	ukazatel zásobníku; klávesa a znak "mezera"
ESC	klávesa a řídící znak
CONT	klávesa "control"
NL	klávesa a znak "nový řádek"
CR	" " "návrat vozu"
LF	" " "posuv řádku"
HT	" " "vodorovná tabulace"
BS	" " "zpětný posuv"
(R), (RP)	obsah registru, registrového páru
(LABEL)	obsah symbolicky označeného místa paměti
⟨RP⟩	obsah místa paměti adresovaného registrovým párem
R↑, RP↑	inkrementace registru, registrového páru
R↓, RP↓	dekrementace registru, registrového páru

1 ÚVOD

Elektroniku a výpočetní techniku během poloviny sedmdesátých let a začátku let osmdesátých charakterizuje prudký nástup mikroprocesorů a mikropočítačů. Výpočetní technika, donedávna pro mnoho aplikací příliš drahá a rozměrná, se uplatňuje již nejen při hromadném zpracování dat, ve složitých vědeckotechnických výpočtech a při řízení komplikovaných technologických procesů, ale i v oblastech, kde bychom ji ještě před několika lety nehledali. Uvádí se přes 20 000 možných aplikací mikropočítačů v průmyslové a zemědělské výrobě, na výzkumných pracovištích, v dopravě, spojích, administrativě, službách, zdravotnictví, kultuře, ale i ve spotřebních výrobcích a při zájmové činnosti.

Státní a stranické orgány kladou v poslední době nezbytný důraz na urychlený rozvoj součástkové základny mikroelektroniky a na přípravu aplikací mikroprocesorové techniky. Pro další rozvoj národního hospodářství má zásadní význam uplatnění této techniky ve strojírenství. Nasazení mikropočítačů v NC strojích, průmyslových robotech, automatizovaných systémech řízení technologických procesů, řízení manipulace s materiálem a v systémech informačních vytváří předpoklady k zásadnímu růstu produktivity práce, která při současných omezeních zdrojů surovin, energie a pracovních sil je hlavním faktorem dalšího růstu výroby. Rovněž vybavení vyráběných strojů mikropočítači výrazně zvyšuje jejich exportní schopnost.

Aplikace mikropočítačové techniky si vynucuje nové přístupy zejména při návrhu a vývoji, vyžaduje vycházet z úzkého sepětí a silné závislosti technického (hardware) a programového (software) vybavení, což klade vysoké nároky i na vývojové pracovníky. Příprava softwaru mikropočítačů se sice v mnohém podo-

bá tvorbě softwaru velkých počítačů, avšak má i některé významné odlišnosti. K nejdůležitější patří skutečnost, že software není obvykle možno připravit na aplikačním mikropočítači, pro který je určen. Způsobuje to malý objem paměti, znemožňující použití rozsáhlejších programů (textový editor, asembler), a její rozdělení na část RAM a ROM. Rovněž periferní zařízení jsou zaměřena spíše na styk s řízeným objektem než na komunikaci s obsluhou.

Software mikropočítače lze vyvíjet prostřednictvím nepřímého softwaru na velkém počítači. Velmi často však požadujeme jeho vyzkoušení ve spolupráci s řízeným objektem a v reálném čase. Pak jsou zcela nezbytné mikropočítačové vývojové systémy (MVS).

V zahraničí jsou již řadu let prodávány bohatě hardwarově a softwarově vybavené MVS, např. INTELLEC nebo MDS 800 firmy INTEL. Podobné MVS byly v poslední době vyvinuty i v ČSSR ve Výzkumném ústavu výpočtovej techniky v Žilině (SM 50/40) a v n.p. Tesla Kolín (MVS 800), avšak zatím jsou nedostupné, patrně vzhledem ke kusovosti výroby závislé na dovozu součástek. Pravidelná výroba se připravuje na rok 1982 v návaznosti na zahájení výroby mikroprocesorových obvodů v n.p. Tesla Piešťany resp. Tesla Rožnov.

Vzhledem k tomu, že je třeba aplikace mikropočítačů připravovat s předstihem i před jejich výrobou, nezbývá než maximálně využívat i skromnější hardware, který je právě k dispozici. Jeho rozmanitost na různých pracovištích obvykle způsobuje, že již vyvinutý software nelze snadno přenést jinam, a proto nezbývá než software vytvářet konkrétně podle hardwaru daného pracoviště.

Diplomová práce se zabývá softwarem i hardwarem MVS, pou-

žívaného ve společné mikroprocesorové laboratoři Koncernového výzkumného ústavu ELITEX a textilní fakulty VŠST při vývoji mikropočítačového řízení textilních strojů. Úkol spočíval v připojení elektrického psacího stroje k mikropočítači po stránce hardwarové i softwarové a ve vytvoření softwaru, umožňujícího komunikovat s MVS prostřednictvím psacího stroje. Návrh softwaru byl zaměřen zejména na usnadnění práce při ladění uživatelských programů. Vývoj si vyžádal dořešit připojení snímače a děrovače děrné pásky do MVS a upravit nepřímý asembler používaný pro překlad vytvářených programů.

2 MIKROPOČÍTAČOVÝ VÝVOJOVÝ SYSTÉM

2.1 VYBAVENÍ MIKROPOČÍTAČOVÝCH VÝVOJOVÝCH SYSTÉMŮ

MVS slouží k přípravě softwaru, k odzkoušení vazby s okolím a jednotek styku s prostředím pro aplikační mikropočítače. Jeho jádrem je centrální jednotka mikropočítače výrazně rozšířená o RAM. Styk s obsluhou umožňuje videoterminál a mozaiková tiskárna, k archivaci programů a dat slouží buď snímač a děrovač děrné pásky nebo disketová popř. kazetopásková jednotka. MVS může obsahovat emulační a zkušební adaptor a simulátor permanentní paměti pro účinné zkoušení uživatelského systému; programátor EPROM resp. PROM je nezbytností.

MVS obsahuje rozsáhlý software. Monitor, uložený v permanentní paměti, obsluhuje periferní zařízení, provádí základní komunikaci s obsluhou, usnadňuje ladění uživatelských programů a spouští ostatní systémové programy. Prostřednictvím textového editoru uživatel zavádí a upravuje zdrojový program, který se pak překládá makroasemblerem popř. překladači z vyšších jazyků (PL/M nebo mikroprocesorové verze jazyků BASIC, FORTRAN, COBOL).

Spojovací program vytváří úplný program z modulů ve strojovém kódu získaných různými způsoby a zaváděcí program jej zavádí na požadované místo v operační paměti. Ladící program rozšiřuje možnosti monitoru při ladění cílového programu zejména při použití emulačního a zkušebního adaptoru. K softwaru MVS patří i knihovna standardních podprogramů popř. aritmetika v plovoucí řádové čárce a transcendentní funkce.

Takto vybavený MVS umožňuje provést hlavní etapy návrhu mikropočítačového řízení bez použití aplikačního mikropočítače.

2.2 MIKROPOČÍTAČ TK-80

Jádrem MVS provozovaného na katedře elektrotechniky VŠST je jednodeskový školní mikropočítač TK-80, výrobek japonské firmy Nippon Electric Co. Podrobně je popsán v /1/. Obsahuje jednočipový osmibitový unipolární mikroprocesor 8080 s podpůrnými obvody, 3 obvody 256 x 8 bit ROM a 4 obvody 256 x 4 bit RAM. Kapacitu 0,75 KB ROM lze rozšířit vložením dalších obvodů do patic na 1 KB, RAM lze rozšířit z 0,5 KB rovněž na 1 KB. Programovatelný I/O obvod 8255 slouží převážně pro provoz osmimístného sedmisegmentového displeje a klávesnice. Klávesy s hexadecimálními číslicemi umožňují vstup informace do mikropočítače. Displej hexadecimálně zobrazuje vstupující i vystupující informace ve 2 polích po 4 číslicích. Levé, adresové, slouží převážně k zobrazení adresy paměťové buňky, pravé, datové, ukazuje obsah této buňky nebo vkládané číslo. Řídící klávesy zpravidla spouštějí programy v jednoduchém monitoru, dodávaném výrobcem v ROM na adrese 0000 až 02FF.

Funkce kláves:

0,..., 9, A,...,F - vstup hexadecimální číslice a její zobrazení na 8. místě displeje (zleva), předcházející číslice se v datovém poli posunou o 1 místo vlevo;

ADRS SET - přemístění obsahu datového pole do adresového a zobrazení obsahu příslušné paměťové buňky na 7. a 8. místě;

READ INCR, READ DECR - zvětšení resp. zmenšení adresy o jednotku a zobrazení obsahu nové buňky;

WRITE INCR - zápis obsahu 7. a 8. místa displeje na zobrazovanou adresu a zvětšení adresy o jednotku;

RUN - spuštění chodu programu od zobrazované adresy;

STORE DATA - uložení obsahu paměti od adresy v adresovém poli do adresy v datovém poli na připojený magnetofon;

LOAD DATA - nahrání obsahu paměti z magnetofonu; při bezchybném přenosu se na displeji objeví počáteční a koncová adresa bloku, v opačném případě hlášení chyby;

RET - v režimu po krocích provede 1 instrukci programu;

RESET - hardwarová inicializace mikropočítače.

Přepneme-li páčkový přepínač STEP/AUTO do polohy STEP, nastavíme krokování programu. Po provedení 1 instrukce dojde k přerušení a obslužný podprogram uloží obsah všech registrů včetně F, SP a PC do vyhrazeného úseku paměti. Na displeji se zobrazí adresa následující instrukce, obsah akumulátoru a registru příznaků. Nyní lze použít monitorové funkce, m.j. k prohlédnutí nebo modifikaci obsahu registrů uložených v paměti. Při dalším kroku totiž monitor obnoví obsah registrů, provede následující instrukci uživatelského programu a opět dojde k přerušení. Zastavení uživatelského programu až po určitém počtu průchodů zvolenou adresou dosáhneme vložením obou informací na příslušná místa paměti.

Monitor obsahuje programy provádějící popsané funkce tlačítek včetně ladění a podprogramy pro obsluhu klávesnice a displeje. Některé monitorové podprogramy lze samostatně používat, např. zobrazení obsahu buněk na displeji nebo krátké zpoždění.

2.3 ROZŠÍŘENÍ TK-80

Samotný školní mikropočítač TK-80 je vhodný pouze k výuce programování mikroprocesoru 8080. K vývoji mikropočítačového řízení listového stroje a do funkce malého MVS byl rozšířen o řadu desek, zasunutých do stavebnicového rámu, na kterém je připevněn. V rámu je upevněna řada konektorů URS pro připojení periferních zařízení.

Pro připojení dalších pamětí je zesílena adresová sběrnice a zavedena úplná adresace, neboť TK-80 nevyužívá k adresování všech 16 bitů. Při provozu po krocích je v činnosti indikace příznaků. Stav každého z 5 příznaků indikuje LED. Obsah registru příznaků je sice zobrazován na displeji, avšak dešifrovat nastavení určitého příznaku z dvoumístného hexadecimálního čísla je obtížné.

Vstupní a výstupní osmibitové brány slouží k připojení periferií, dva obvody 8255 budou sloužit k simulaci kanálů aplikačního mikropočítače SBC 80/10, který bude vyráběn i v ČSSR. Vstupní brána B obvodu 8255 z TK-80 a jedna výstupní brána indikuje svůj stav prostřednictvím LED.

RAM rozšiřující paměť o 3 KB je rozdělena na dvě části. 2 KB na adrese 0800 až 0FFF simulují část permanentní paměti SBC 80/10 a 1 KB na adrese 3C00 až 3FFF představuje RAM SBC 80/10. Původní RAM TK-80 obsazuje adresu 8200 až 83FF, monitor používá 57 B od adresy 83C7.

K dispozici je 1 KB EPROM od adresy 1C00 pro uložení základních podprogramů MVS.

Monitor TK-80 využívá 3 signály přerušení. Číslicovým přepínačem lze zvolit vyslání 1 z 5 zbývajících signálů (RST 2 až RST 6) při vnějším přerušení.

Mikropočítač TK-80 je napájen ze zabudovaného zdroje. Jeho

zapnutím se samočinně sepnou 3 laboratorní zdroje napájející desky rozšíření.

3 TECHNICKÉ VYBAVENÍ SYSTÉMU

3.1 PSACÍ STROJ

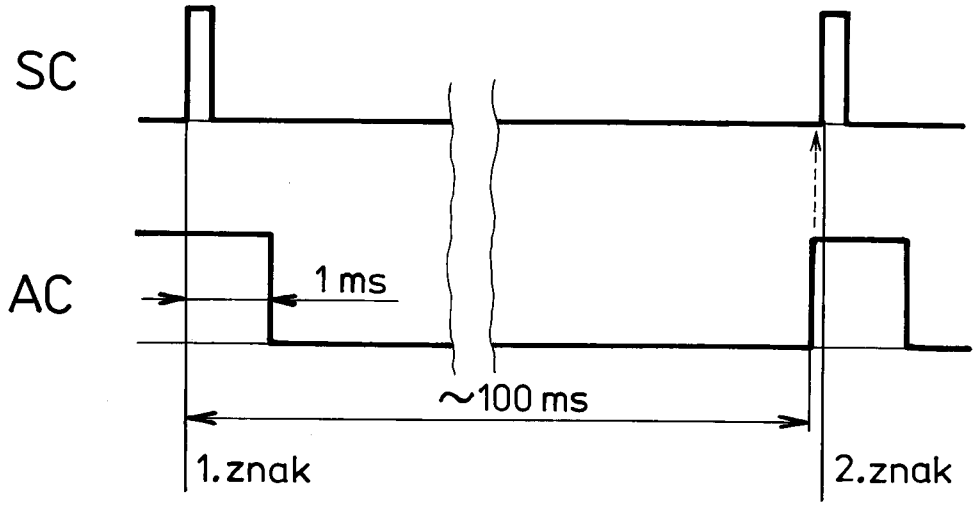
3.1.1 Charakteristika psacího stroje

Ke komunikaci obsluhy s MVS slouží elektrický psací stroj CONSUL 256 s bezkontaktní elektrickou klávesnicí.

Psací jednotka je vybavena bezkontaktními snímači funkčních stavů a elektromechanicky ovládaným vodorovným tabulátorem. Jako součást psacího stroje byla dodána i elektronika pro řízení elektromagnetů psací jednotky, která podle signálu v upraveném kódu ASCII nejprve podle potřeby vydá povel k přeřazení nebo ke změně barvy a pak vybudí elektromagnet příslušné typové páky. Rovněž provádí automatický návrat vozu z konce řádku.

Náběžnou hranou vstupního signálu "předání tisku SC" se do pracovního registru zapíše z 8 datových vodičů kód znaku, který má být psací jednotkou vytištěn (viz příl. 2). Sedmý bit znakového bytu volí černou nebo červenou barvu tisku. Asi 1 ms po přijetí kódu znaku přejde výstupní signál "žádost tisku AC" na log. 0 (obr. 1) a v činnost se uvede elektromagnet typové páky nebo zvláštní funkce. Pokud není mechanismus přeřazu nebo barvy v požadované poloze, dojde nejprve k jeho přestavení. Dokončení tisku je oznámeno přechodem signálu AC na log. 1. Při předání bytu, kterému neodpovídá v příl. 2 žádná funkce, se nastaví AC trvale na log. 0. Vstup a tisk dalšího znaku může být proveden.

Klávesnice obsahuje 74 kláves, jejichž stisk je vyhodnocován IO s Hallovými snímači. Každý IO budí 2 vodiče sběrnice zavedené do logické kombinační sítě, na jejímž výstupu se objeví



obr. 1

negace upraveného kódu ASCII příslušného znaku se 7. bitem paritním. Elektronika klávesnice sice obsahuje registr přeřazu ovládaný dvěma klávesami, ale generovaný kód kláves není modifikován ani stavem registru ani klávesou CONT, jež má při současném stisku se znakovou klávesou způsobit generování řídícího znaku dle označení na klávese.

Stisknutí kterékoliv klávesy kromě CONT a kláves přeřazu signalizuje log. 0 na signálním výstupu STROBE.

3.1.2 Překódování a propojení klávesnice s psací jednotkou

Z neznámých důvodů nesouhlasí u některých znaků kód klávesnice s kódem psací jednotky, jak vidíme porovnáním příl. 1 a příl. 2. Rozhodli jsme se zabudovat do psacího stroje desku pomocné elektroniky s následujícími funkcemi:

1. modifikace kódu klávesy stavem registru přeřazu;
2. modifikace kódu stisknutím klávesy CONT;
3. nezávislost kláves s diakritickými znaménky na stavu registru přeřazu;
4. vyčlenění klávesy pro volbu červeného tisku;
5. vyčlenění klávesy pro monostabilní ovládání registru přeřazu;
6. propojení klávesnice a psací jednotky do režimu autonomního provozu;

7. automatická repetice tisku znaku;
8. indikace stavu registru přeřazu;
9. automatické přepínání a indikace režimu provozu;
10. vytvoření zvláštního signálu pro inicializaci mikropočítače.

Režim autonomního provozu umožňuje využívat drahé zařízení bez připojení na mikropočítač jako běžný elektrický psací stroj. Propojení především invertuje negovaný kód klávesnice a generuje signál SC při výskytu signálu STROBE. Požadovaný signál stisknutí některých kláves by bylo možno získat dekódováním kódu klávesnice, ale jednodušší je vyvést tento pomocný signál přímo od klávesy. Některé signály jsou vytvořeny pomocí volných hradel na desce elektroniky (příl. 4).

Klávesy sazeče a výmazu tabulátorových zarážek budí jedním výstupem společný vodič sběrnice, který je vyveden jako signál SZVM sloužící k převodu kódu 11 a 12 na 1F a 1E.

Signál CH je aktivován stisknutím klávesy "čárka" nebo "háček" a blokuje vliv registru přeřazu na generovaný kód. Klávesa "čárka" byla zaměněna klávesou s kódem 40, která teď generuje svému označení odpovídající kód 5C resp. 7C. "Čárka" byla odpojena od vodičů generujících kód 40 a připojena přes diody na vodiče kódu háčku, generovaný signál CA však zároveň na desce překódování převádí kód háčku 5E na kód čárky 7E.

Z registru přeřazu vystupují navzájem inverzní signály MP (malá písmena) a VP (velká písmena). Je-li registr přeřazu nastaven na malá písmena (MP = 1), mění se kód sloupce 2 na 3 a sloupců 4 nebo 5 na 6 nebo 7 (příl. 1 a 2). Kód mezery 20 však nesmí být ovlivněn, proto je vyveden její signál SP provádějící blokování.

Na 7. bit byl připojen místo signálu parity, který není nutný, od sběrnice odpojený výstup klávesy USR, jež byla pře-

značena na RED. Její stisk nastaví 7. bit = 1 a vyvolá tak červený tisk.

Klávesa FSR byla odpojena a přeznačena, při stlačení nastavuje impulzy vytvořenými kondenzátory 3k9 registr přeřazu na VP = 1 a při uvolnění na MP = 1. Usnadňuje to psaní velkých písmen a interpunkčních znamének. Na klávesách s kódy 2C až 2F bylo nutno zaměnit vrchní a spodní symbol, aby tištěný znak odpovídal stavu registru přeřazu a označení klávesy.

Současné stlačení kláves ESC a CONT generuje signál INIT, který inicializuje mikropočítač a ruší blokování psací jednotky. Stiskneme-li současně klávesu CONT a klávesu s kódem ve sloupci 4 nebo 5, vystoupí kód ve sloupci 0 nebo 1. Klávesou 4C lze takto provést návrat vozu bez řádkování.

Je-li klávesa s kódem ve sloupcích 2 až 7 stlačena déle než 0,5 s, repetiční obvod zrychleně generuje signál STROBE, takže znak se tiskne opakovaně. Usnadňuje to vytváření grafické úpravy zejména mezerami a podtrháváním.

Stav registru přeřazu indikují 2 LED: MP a VP. Připojení napětí +5 V na svorku MCC způsobí odpojení psací jednotky od klávesnice a rozsvícení LED MCC (microcomputer control), což nastane při zapnutí mikropočítače a při zasunutých konektorech.

Zapojení se schematem v příl. 3 je realizováno na univerzální desce plošných spojů, zhotovené fotografickou metodou, a připojené 62-kolíkovým konektorem FRB. Devět IO je vloženo v paticích, spoje tvoří speciální vodič, umožňující rychlé pájení bez odizolování. Logická kombinační síť byla navržena metodou Karnaughových map. Signály K přicházejí od elektroniky klávesnice, signály W vstupují do elektroniky psací jednotky a signály C vedou do mikropočítače. Řízené propojení klávesnice a psací jednotky realizují oboustranné budiče sběrnice s třístavovým vý-

stupem IO 1 a 5. Propojení je aktivováno log. 0 na vývodech č. 1 obou IO, jež jsou využity pouze z poloviny. Propojení signálu tisku na signál SEW do psací jednotky je blokováno v režimu MCC hradlem 8a. Stejnosměrné oddělení výstupu desky propojení a výstupu mikropočítače vyhovuje, i když nesplňuje podmínky propojení obvodů TTL. U všech vstupů jsou proti zemi připojeny odpory, aby proudy ze vstupů přidaných IO nezvýšily úbytek na odporech elektroniky klávesnice nad povolenou hranici log. 0.

Obvod repetice tvoří hradla 4a, 4c, 7c, 8d a tranzistor T3. Monostabilní obvod z prvků 4c a T3 je překlápěn náběžnou hranou přes kondenzátor 1k z výstupu hradla 4d, které invertuje signál STROBE. Na vývodu č. 10 IO7 je log. 0, takže multivibrátor tvořený hradly 4a a 7c je zablokován. Po uplynutí 0,5 s se monostabilní obvod vrátí do stabilního stavu, multivibrátor začne překlápět s frekvencí asi 8 Hz a na hradlu 8d vytváří signál vstupu znaku do mikropočítače SEC resp. signál tisku SEW tak dlouho, dokud trvá signál STROBE. V klidovém stavu nebo při stisknutí kláves s kódem ve sloupci 0 nebo 1 je multivibrátor blokován log. 0 z hradla 8b. Při běžném psaní na psacím stroji je monostabilní obvod po uvolnění klávesy překlopen záporným impulzem přes kondenzátor 1k do stabilního stavu a časovací kondenzátor 2M se vybije přes hradlo 4c a diodu, takže obvod je okamžitě připraven k další činnosti.

Řízení repetice by bylo výhodnější provést zpětnou vazbou přes signál AC. Avšak v době realizace zapojení jsme se domnívali, že zpětná vazba nefunguje správně (viz 5.1.3), a proto jsme použili popsané řešení.

Funkce automatického návratu vozu nefungovala - při dojetí vozu na koncovou zarážku docházelo k zablokování elektroniky psací jednotky - proto jsme příslušný snímač odpojili.

Klávesnici a psací jednotku propojují s mikropočítačem ploché ohebné kabely zakončené 26-kolíkovými konektory URS. Schema propojení včetně připojení popsané desky je v příl. 6.

3.1.3 Napájení

Elektrický psací stroj CONSUL 256 vyžaduje ke své činnosti externí napájení - 16 V/1,5 A pro elektromagnety a 5 V/0,7 A pro elektroniku. Přiváděné síťové napětí slouží pouze k napájení elektromotoru.

Pro psací stroj jsme zhotovili jednoúčelový zdroj dle příl. 5. Pevně nastavené napětí 16,5 V je stabilizováno IO MAA 723 v základním zapojení s Darlingtonovou dvojicí tranzistorů jako proudovým zesilovačem. Proudová ochrana snímající velikost výstupního proudu na odporu 0,33 Ω omezuje proud na 2,5 A. Tranzistor KU 612 je připevněn s chladičem přímo na desce plošných spojů, tranzistor KD 602 je izolovaně upevněn na zadním panelu.

Zdroj 5 V/1 A napájí elektroniku psací jednotky a klávesnice a desku propojení a překódování. Integrovaný stabilizátor MA 7805 je s chladičem rovněž umístěn na desce plošných spojů. I když obsahuje proudovou a tepelnou ochranu, jsou obvody TTL na deskách elektroniky pro jistotu chráněny proti přepětí při poruše IO tyristorovou zkratovací pojistkou.

Přítomnost obou napětí kontrolují 2 LED, zapnutí zdroje pak žárovka v obvodu transformátoru. Síťový spínač zapíná i proud do síťové zásuvky na čelním panelu, takže je možné současně ovládat i elektromotor psacího stroje.

Deska plošných spojů byla vzhledem k potřebě širších vodivých drah snadno zhotovena pomocí samolepící fólie. Zdroj je zabudován ve čtyřdílné skříňce z hliníkového plechu, vývody tvoří přístrojové svorky.

3.2 SNÍMAČ DĚRNÉ PÁSKY

Vstupním periferním zařízením MVS je snímač děrné pásky FS 1503, přístroj uzpůsobený pro nepřetržitý provoz, s vysokým výkonem 1500 znaků/s.

Z ovládacích prvků je na přístroji pouze síťový spínač pro zapnutí elektroniky a prosvětlovací žárovky, ostatní funkce se ovládají signály TTL. Vnějším povelem MOTOR se spustí hnací motorek, signálem DPP se ověřuje připravenost snímače k provozu. Tento signál jsme trvale nastavili na log. 1. Signálem SO potvrzuje snímač připravenost k provozu. Log. 1 signálu SO je podmíněna splněním řady podmínek, např. povel MOTOR musí trvat déle než 4 s, páska je založena, rameno je v pracovní poloze a je přítomen signál DPP. Povelem START se páska uvádí do pohybu, povelem STOP se zastavuje.

Náběžná hrana vodící stopy synchronizuje zápis sejmutého znaku do vyrovnávací paměti. Signálem SC snímač oznamuje, že informace na výstupu je platná. Signály i ovládací povely jsou aktivní při log. 1.

Při provozu MVS je snímač stále připojen na síť, avšak motorek je s ohledem na opotřebování a hluk zapínán povelem MOTOR pouze při snímání. Při vypnutém motorku je však mimo činnost nucená ventilace, takže se žárovka a obvod elektromagnetické brzdy nadměrně zahřívají. Do snímače jsme zabudovali pomocné relé se zpožděným odpadem, které připojuje žárovku pouze při zapnutí motorku.

Schema propojení snímače s mikropočítačem zachycuje příl.7.

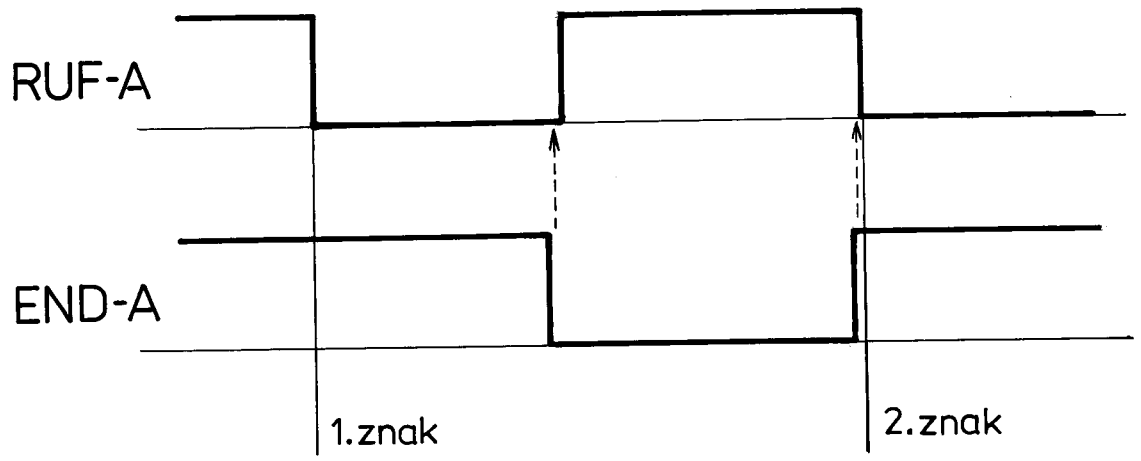
3.3 DĚROVAČ DĚRNÉ PÁSKY

K uložení informací na osmistopou děrnou pásku slouží děrovač pásky DARO 1215/1227. Skládá se ze 4 jednotek: skříň zdrojů

a elektroniky, transportní a děrovací mechanismus, zásobní cívka s odvíjecím motorem, navíjecí cívka s motorem. Navíjecí cívka není při děrování kratších pásek nutná. S připevněným nástavcem ji používáme samostatně ke svíjení pásek rozvinutých při snímání.

Elektronika děrovače je po zapojení MVS stále pod proudem. Motor děrovače se zapíná buď spínačem nebo úrovní log. 0 povelu LBS-ein. Pak je možno stisknutím příslušného tlačítka děrovat prázdné nebo plné znaky. Sestupnou hranou povelu RUF-A začíná děrování osmibitového znaku podle inverzních signálů na datových vodičích (obr. 2). Povel RUF-A a datové signály mohou být ukončeny až po sestupné hraně signálu připravenosti END-A. Návrat signálu END-A na log. 1 znamená konec děrování jednoho a připravenost k děrování dalšího znaku.

Motor odvíjecí cívky jsme přepojili paralelně k motoru děrovače, aby byl také ovládán povelem LBS-ein. Děrovač má možnost paritní kontroly děrovacího mechanismu s návratem a předěrováním chybného znaku, ale v MVS nebyla použita. Schema připojení je v příl. 8.



obr. 2

4 PROSTŘEDKY PRO PŘÍPRAVU PROGRAMOVÉHO VYBAVENÍ

Příprava softwaru mikropočítače vyžaduje převod algoritmu do strojového kódu mikroprocesoru. Algoritmus lze přepsat do mikroprocesorové varianty některého vyššího jazyka a program přeložit příslušným překladačem. Tyto překladače jsou však poměrně složité a nedostupné a cílový program není dostatečně efektivní, což může být u mikropočítačů pro řídící aplikace nebo s omezeným rozsahem paměti na závadu. Naopak JSA mikroprocesorů jsou ve srovnání s JSA velkých počítačů jednodušší, takže se používají běžně při programování a slouží i pro zápis programů v publikacích. Vyplývá to z jednodušší architektury mikroprocesorů, menšího počtu instrukcí i adresovacích modů atd.

Jednoduché programy lze pochopitelně převést z JSA do strojového kódu ručně, pomocí tabulky. Při tomto pracném postupu se nevyvarujeme mnoha chyb, do kterých spadají i chyby při ručním zavádění programu do mikropočítače. Navíc každá změna programu nebo jeho umístění vyvolá změny adresových konstant v příslušných instrukcích (skoky, volání podprogramu aj.). U rozsáhlejších zdrojových programů se bez asembleru neobejdeme. Nelze-li do MVS implementovat rezidentní asembler, používá se nepřímý asembler provozovaný na větším počítači.

4.1 NEPŘÍMÝ ASEMBLER NA RPP-16S

Vývoj první verze programů TWS usnadnil nepřímý asembler /3/ na počítači RPP-16S. Zdrojový program připravený na děrné pásce se nahraje na disk, kde může být z referenčního psacího stroje upravován diskovým systémem /4/ obsahujícím textový editor. Asembler tiskne požadovaný počet výpisů překladu a děruje pásku s cílovým programem. V příl. 9 je výpis překladu programu, umožňujícího použít mikropočítač jako jednoduchý a pomalý logický analyzátor. Převážná část asembleru je naprogramována v inter-

pretačním jazyku BASIC, což způsobuje neúnosnou délku výpočtu. Rychlost překladu je přibližně 8 řádků za minutu, k tomu nutno připočítat tisk výpisu na mozaikové tiskárně rychlostí asi 30 řádků za minutu.

4.2 NEPŘÍMÝ ASEMBLER ASI80A A JEHO ÚPRAVA

Konečná varianta TWS byla překládána na počítači EC-1033 upraveným nepřímým asemblerem /5/ napsaným v jazyce PL/1. Zdrojový program vstupuje do počítače z děrných štítků, což na jedné straně umožňuje poměrně snadné úpravy programu záměnou, vypuštěním, vložením nebo přeskupením štítků, avšak na druhé straně způsobuje problémy s jejich uskladněním a manipulací. Rychlost překladu dosahuje 500 řádků a tisku 300 řádků za minutu. Asembler vychází z koncepce JSA INTEL, je dvojprůchodový s pevným formátem zdrojového textu, bez možnosti zpracování makroinstrukcí. Původní varianta /5/ má zásadní nedostatky:

1. chybí výstup na děrnou pásku;
2. nelze tisknout větší počet výpisů překladu;
3. nezpracovává výrazy v poli operandů;
4. rozsah zdrojového programu je omezen na 200 řádků.

Výhodná principiální možnost zápisu konstant ve 4 číselných soustavách a použití až 8 znaků v symbolu je znehodnocena některými chybami a koncepčními omezeními.

Uvedené nedostatky jsme odstranili úpravou asembleru, která však není optimální, neboť k dispozici byl pouze hrubý vývojový diagram a výpis programu. Zdrojový program je ukládán do řádků pole STORARE (200,15) obsahujícího řetězce 8 znaků. Zvětšení první dimenze v deklaraci pole a úprava dvou podmíněných příkazů zvýšily rozsah zdrojového programu na 800 řádků. Tisk kopií výpisu překladu provádí procedura TISK, která čte řádky

pole STORARE a volá původní proceduru tisku FINPRT. Obsahuje i opis modulů ENDPAGE ON-UNIT, které zajišťují tisk hlaviček stránek. Další jednoduchá úprava umožňuje skutečně používání symbolů o délce až 8 znaků.

Při prvním průchodu zapíší doplněné příkazy během zpracování pseudoinstrukcí ORG a EJECT do numerického pole BLOTAB (50,3) číslo zpracovávaného řádku a stav čítače alokace před a po pseudoinstrukci. Uložené informace slouží proceduře PUNPREP k utvoření bloků na děrné pásce. Každý blok začíná počáteční a koncovou adresou a je oddělen 20 prázdnými znaky pro vizuální oddělení bloků. Procedura PUNPREP připravuje do řetězce PUNFIL cílový program ve znakovém formátu, tzn. každý byte je vyděrován jako 2 hexadecimální znaky v ASCII kódu. Desítková čísla na hexadecimální převádí procedura BINAHE. Znaky cílového programu jsou hledány v té oblasti pole STORARE, z které se tiskne strojový kód ve výpisu překladu. Pro děrování musela být zvlášť ošetřena pseudoinstrukce DS - do cílového programu se vkládá potřebný počet nul.

Požadavek na tisk a děrování zadáváme prvním štítkem dat, na kterém je vyděrováno:

PRINT=x, PUNCH=y;

x je počet výpisů, **y** je počet pásek. Štítek je čten příkazem GET DATA. Nalezne-li asembler v programu JSA chyby, pouze vytiskne 1 protokol bez ohledu na **x** a **y**.

U dvoubytových operandů můžeme od 28. do 30. sloupce zapsat konstantu ve tvaru +DD nebo -DD, která bude přičtena k hodnotě symbolu. D je desítková číslice, vedoucí nula může být vynechána. Výskyt nesprávně zapsané konstanty ošetří CONVERSION ON-UNIT.

Začátek komentáře byl posunut až na 33. sloupec. Obsah ští-

tku, který má na konci znak *, je tištěn s výraznou grafickou úpravou vhodnou pro nadpisy. Další grafické úpravy rovněž zvyšují přehlednost výpisu, jak lze zjistit porovnáním příkladu v /5/ s příl. 11.

Úpravy rozšířily program z 690 příkazů v 985 řádcích na 807 příkazů v 1090 řádcích.

5 PROGRAMOVÉ VYBAVENÍ SYSTÉMU

5.1 KONCEPCE PŘÍDAVNÉHO MONITORU

Možnosti softwaru jsou úzce determinovány hardwarovou konfigurací systému. Zatímco výkonný snímač a děrovač děrné pásky nejsou v malém MVS nikterak přetíženy a nejsou omezujícími prvky systému, nelze totéž říci o psacím stroji. Pro vstup vyhovuje, avšak jako výstupní zařízení je neúnosně pomalý, požadujeme-li výpis části paměti, programu nebo běžného textu. Hlavním limitujícím faktorem je však bezesporu současný rozsah operační paměti, který činí 4,5 KB včetně EPROM. Software byl navržen tak, aby při minimálním rozsahu umožňoval maximum jednoduchých funkcí a přitom zbyl prostor pro uložení uživatelského programu. Přídavný monitor je dále označován původně pracovní zkratkou TWS (typewriter system).

5.1.1 Volba systémových funkcí

K základním funkcím každého monitoru patří zápis, změna a výpis obsahu paměti. Ruční vkládání programu a dat do paměti provedeme nejrychleji klávesnicí TK-80, neboť hexadecimální klávesy jsou na ní na rozdíl od psacího stroje umístěny u sebe, při jejím sledování zůstává displej v zorném poli a řídícími klávesami lze rychle měnit adresy. Naopak při ladění je výhodné mít změny obsahu paměti zaprotokolovány na výpisu z psacího stroje.

Výpis obsahu paměti v přehledné tabulce usnadňuje kontrolu a umožňuje archivaci obsahu delších bloků. Obecnou funkcí je přesun obsahu bloku paměti na jinou adresu, s možností překrývání nového a původního bloku.

Přídavný monitor obsluhuje snímač a děrovač děrné pásky. Obsah vybraných bloků paměti děruje do pásky nebo ho z pásky zavádí do paměti, buď z celé najednou nebo jen ze zvoleného bloku.

Monitor TK-80 poskytuje poměrně dobrý způsob ladění uživatelských programů. Nevýhodou je pracné získávání a ukládání trasovacích informací do vyhrazených buněk paměti. Musíme neustále sledovat tabulku s popisem, co je kde uloženo, a zobrazené informace si poznamenávat. Obsluhu značně ulehčí výpis obsahu vyhrazených buněk uchovávajících stav registrů, nadepsaný označením registrů. Obsah vyhrazených buněk lze změnit. Manipulaci se zásobníkem zachytí výpis jeho obsahu, opět s možností modifikace. Startovací adresu uživatelského programu, adresu zastavení a počet průchodů budeme rovněž snáze zadávat z psacího stroje.

Předností psacího stroje proti mozaikové tiskárně je běžné písmo, tj. výraznější tvar znaků, malá písmena, diakritická a jiná znaménka. Psací stroj CONSUL 256 sice používá stejné háčky a čárky pro velká i malá písmena a nemá kroužkované "ů", ale přesto lze použít pro psaní běžného textu. Nabízí se použít spojení psacího stroje a mikropočítače pro uložení textu do paměti se současnou opravou překlepů. Text lze pak vypsat, i vícekrát, nebo uložit na pásku pro pozdější použití. Uvedené funkce se uplatní při vyřizování korepondence a vedení softwarové dokumentace.

Za nejpotřebnější funkci MVS můžeme bezesporu označit překlad programů z JSA do strojového kódu. Malá kapacita paměti neumožnila zavést asembler a textový editor. Překlad nutno za-

jišťovat nepřímým asemblerem, což je však zdlouhavá operace, kterou lze použít pouze při prvním překladu nového programu a pak při překladu jeho definitivní verze k získání výpisu. Problémem je provádění drobných úprav programu během ladění. Každý přesun programu, každé vsunutí instrukce vyžaduje úpravu adres skoků a volání podprogramů, protože mikroprocesor 8080 používá převážně absolutní adresování. Zaměřili jsme se tedy na rychlou úpravu uživatelských programů jako je přesun programu a tabulky proměnných či konstant v paměti, vkládání a vypouštění instrukcí, změna operandu určité hodnoty u všech instrukcí. Jednoduchý disasembler by měl po provedení úprav pozměněný program vypsat.

TWS je uložen zčásti v EPROM a zčásti na děrné pásce, ze které se po zapnutí MVS zavede do RAM. V EPROM jsou podprogramy použitelné obecně a takové, u nichž se nepředpokládá v budoucnosti změna. V permanentní paměti musí být přirozeně uložen program pro zavádění programů z děrné pásky. Rezidentní část TWS zaplňuje 0,5 KB EPROM od adresy 1D00, dala by se rozšířit na 0,75 KB.

Malá kapacita paměti si vyžádala rozdělení TWS na 3 bloky, aby zbyl prostor pro uživatelský program nebo pro uložení textu.

Jádro obsahuje řídící program s tabulkou adres příkazových programů, tabulku textu, tabulku konstant, tabulku proměnných a pracovní oblast pro ukládání 1 řádku vstupujícího textu. Dále obsahuje všechny základní funkce TWS. Je rozděleno na 2 části podle 2 variant. Prvá část obsazuje paměť od adresy 8200 až k části používané monitorem TK-80, druhá buď od 3D00 do 3FFF nebo od 0D00 do 0FFF.

Blok nazvaný "ladění" zajišťuje funkce pro snazší ladění a pro úpravu programu. Obsazuje oblast od 0E00 do 0FFF.

Blok nazvaný "text" zajišťuje práci s textem. Obsazuje pa-

měť od 3C00 do 3CFF; oblast od 0800 do 0FFF slouží pro uložení textu (2 KB, tedy necelá 1 stránka).

Každý blok je na jedné děrné pásce, aby mohl být samostatně zaveden spolu s aktuální tabulkou adres dostupných příkazových programů. Jsou možné tyto kombinace:

1. jádro (1. varianta);
2. jádro (2. varianta);
3. jádro (1. varianta) + ladění;
4. jádro (1. varianta) + text.

Programy TWS byly optimalizovány z hlediska minimální spotřeby paměti na úkor rychlosti, neboť rozsah paměti je výrazně omezen, avšak doba odezvy periferií i obsluhy je značná. Přesto i nejsložitější operace TWS, nepoužívající periferie, nad 1 KB uživatelského programu trvá pouze desetiny sekundy. Z uvedených důvodů v programech výrazně převažují jednobytové operace se zásobníkem nad tříbytovými operacemi s absolutní adresací paměti. Naopak dekompozice programů na podprogramy sledovala převážně funkční hledisko co nejširší použitelnosti podprogramů. Jestliže by vyčlenění velmi krátké části společné 2 nebo 3 programům ušetřilo pouze několik bytů, podprogram nebyl zaveden, neboť by to snížilo přehlednost a přetěžovalo zásobník.

5.1.2 Struktura příkazů a chybová hlášení

Formát příkazů zadávaných počítači může být různý. Programově nejjednodušší je vkládat startovací adresu programu, realizujícího požadovanou funkci, a seznam parametrů. Pro obsluhu nejpohodlnější, avšak dosti zdlouhavý a na kapacitu paměti náročný způsob spočívá ve výběru varianty z předloženého seznamu nebo zadání příkazového slova či jejich skupiny v běžné řeči. O potřebné parametry požádá počítač sám.

V TWS se skládá příkaz z příkazového slova a seznamu parametrů. Příkazové slovo je jedno anglické slovo, tak jak je obvykle ve výpočetní technice používáno, nebo jeho zkratka. Seznam parametrů sestává z několika hexadecimálních nebo desítkových čísel, navzájem i od příkazového slova oddělených oddělovačem. Některé příkazy seznam parametrů neobsahují nebo požadují vložení parametru i v průběhu činnosti. Parametr většinou představuje adresu. K ukončení příkazu nebo parametru a jeho provedení slouží vyhrazená klávesa ESC.

TWS je otevřený systém. Použitím systémových příkazů lze zavádět a rušit další příkazy, které mohou mít až 10 parametrů.

Syntaxi příkazů nejlépe vyjádří Backusova normální forma:

⟨příkaz⟩ ::= ⟨příkazové slovo⟩⟨ESC⟩ | ⟨příkazové slovo⟩ ⟨seznam parametrů⟩⟨ESC⟩

⟨seznam parametrů⟩ ::= ⟨oddělovač⟩⟨parametr⟩ | ⟨oddělovač⟩ ⟨parametr⟩⟨seznam parametrů⟩

⟨parametr⟩ ::= ⟨hexadecimální číslo⟩ | # ⟨desítkové číslo⟩

⟨oddělovač⟩ ::= ⟨oddělovací znak⟩ | ⟨oddělovač⟩⟨oddělovací znak⟩

⟨oddělovací znak⟩ ::= ⟨SP⟩ | ⟨NL⟩ | ⟨CR⟩ | ⟨LF⟩ | ⟨HT⟩

TWS se ohlašuje červeným výpisem "(NL)(NL)tws :" a očekává zadání příkazu.

Vždy po vypsání červené dvojtečky očekává vstup z klávesnice. Za červeným rovnítkem následuje číselná hodnota vytvořená mikropočítačem. Její význam závisí na prováděném příkazu.

Červenými uvozovkami začneme a ukončíme komentář, který TWS ignoruje. Usnadní pozdější orientaci v provedených operacích. Lze ho zapsat před, uvnitř i za příkazem, avšak nenahrazuje oddělovač.

Při zadávání příkazu nebo psaní textu lze k opravám používat klávesu BS.

U hexadecimálních parametrů akceptuje TWS poslední 4 číslice před oddělovačem, dekadický parametr může být maximálně pěticiferný, menší než 65 536. Je-li číslic méně, doplňují se zleva nulami.

TWS kontroluje správnost vkládaných příkazů. Při výskytu chyby nebo důležité okolnosti vypíše jedno z následujících chybových hlášení:

červeně:

"syntax" - nesprávně zapsané příkazové slovo nebo příkaz není definován nebo vložen nesprávný počet parametrů;

"illegal" - parametr obsahuje nepovolený znak (jiný než dekadickou resp. hexadecimální číslici);

"error" - chyba při snímání z děrné pásky (chyba parity nebo nepovolený znak);

"addr" - adresa konce bloku je menší než adresa začátku bloku;

"range" - vstupující dekadické číslo je větší než 65 535;

černě:

"out" - překročení počáteční nebo koncové hranice pracovního pole paměti při vstupu z klávesnice nebo dosažení zadané koncové hranice při ukládání textu do paměti;

"exist" - příkaz již byl definován nebo se shoduje kód vloženého příkazového slova s kódem jiného již definovaného příkazu (pouze u příkazu DEFINE);

"not exist" - příkaz není definován (pouze u příkazu CANCEL).

Syntaxe příkazů a některá chybová hlášení jsou zřejmá z příl. 10.

5.1.3 Řízení periferních zařízení

Periferní zařízení lze připojit k mikroprocesoru dvěma způsoby. Přenos se provádí buď instrukcemi pro práci s pamětí nebo instrukcemi pro vstup a výstup s jednobytovou adresou periferie. V MVS je použit druhý způsob.

Žádá-li mikroprocesor přenos, často není periferie připravena. Tato situace se obsluhuje jedním ze tří způsobů:

1. signál připravenosti periferie vyvolá přerušení a spuštění obslužného podprogramu;
2. po dobu nepřipravenosti periferie je mikroprocesor blokován signálem WAIT;
3. mikroprocesor čeká v programové smyčce na signál připravenosti periferie.

První způsob se používá ve složitějších systémech, které vyžadují plné vytížení mikroprocesoru. Z hlediska programátora je obtížné zajistit při přerušení v libovolném okamžiku správnou funkci systému. Druhý způsob vyžaduje hardwarový zásah do obvodů řízení činnosti mikroprocesoru.

Použili jsme třetí způsob - softwarové ošetření, které nevyžaduje hardwarové obvody. Signály ode všech periferií jsou přivedeny na bránu B obvodu 8255 na TK-80 - PORTB:

bit 0. magnetofon - vstup dat (nyní již nepoužíván)
1. snímač - SC (data připravena)
2. psací jednotka - AC (připravenost)
3. klávesnice - SEC (data připravena)
4. děrovač - END-A (připravenost)
5. snímač - SO (připravenost ke startu)
6.,7. volné

V programové čekací smyčce se čte stav brány B, maskováním se ponechá pouze žádaný bit, jenž se testuje buď na log. 0 nebo na log. 1, a podle výsledku se test buď opakuje nebo chod programu pokračuje. Dvě smyčky za sebou reagující na opačný stav detekují hranu signálu.

Pro výstup povelů řídících periferie je nejvhodnější obvod 8255, neboť umožňuje nastavení jednotlivých bitů. Avšak tento obvod na TK-80 má volné pouze 4 bity, které navíc mění svůj stav při inicializaci mikropočítače, a 2 obvody 8255 v rozšíření musí zůstat volné pro simulaci kanálů SBC 80/10. Nezbylo než použít jako řídící výstupní bránu obvod 8212, což přináší některé komplikace. Bylo by totiž třeba, aby každý obslužný podprogram měnil stav jen příslušného bitu a ostatní uchoval. Dvoustupňové ovládání snímače a děrovače však vyžaduje poměrně dlouhou sekvenci příkazů, a proto jsme použili jiný, jednodušší způsob, který ovšem neumožňuje paralelní chod periferií.

Obslužné podprogramy nastavují výstupní řídící byte pevně, se změnou stavu příslušného bitu buď vůči klidovému nebo vůči pracovnímu stavu. Znamená to, že např. při generování impulzu tisku se zároveň vypne motor děrovače, avšak při startu snímače zůstane zachován povel pro chod motoru snímače, jenž musí být zapínán s předstihem. Nižší nibble řídící brány je v klidovém stavu nastaven na log. 1, vyšší nibble na log. 0.

Řídící brána - CONTPORT (klidový stav OFH):

bit 0. děrovač - RUF-A (povel děrování)

1.,2. volné

3. snímač - STOP

4. snímač - START

5. snímač - MOTOR (zapnutí motoru a žárovky)

6. psací jednotka - SC (povel tisku)

7. děrovač - $\overline{\text{LBS-ein}}$ (zapnutí motoru)

Signál 7. bitu je invertován tranzistorovým invertorem. Při přímém připojení na 1. bit se motor děrovače rozbíhal ihned po zapnutí mikropočítače, neboť IO 8212 se vynuluje, což neodpovídá klidovému stavu brány CONTPORT.

Podprogram KEYINP pro obsluhu klávesnice čeká na závěrnou hranu signálu SEC a pak načítá do akumulátoru invertovaný datový byte ze vstupní brány KEYBRD.

Obsluha snímače děrné pásky spočívá v zavolání podprogramu READON, který zapne motor a žárovku a ve smyčce čeká 4 s na signál SO. Spínačem na mikropočítači je možno nastavit povel MOTOR na log. 1 trvale, takže čekání na SO odpadá. Pak READON odstartuje snímač a předá řízení nadřazenému programu. Znaky jsou z plynule běžící pásky čteny podprogramem READBYTE z brány READER po náběžné hraně signálu SC. Interval 650 μs postačí pro odbavení znaku nadřazeným programem. Zastavení se provádí posloupností instrukcí MVI A,OFH; OUT CONTPORT.

Výstup na psací jednotku provádí podprogram PRINT. Čeká na náběžnou hranu signálu AC, pak odešle na bránu TWRITER kód znaku a generuje impulz signálu SC. Tím původně podprogram končil, neboť jsme vzhledem k nedostatečné dokumentaci nevěděli, že závěrná hrana signálu AC přichází se zpožděním asi 1 ms. Stávalo se, že podprogram byl opět zavolán před uplynutím této doby a do vyrovnávací paměti psací jednotky byl okamžitě odeslán další znak, což vedlo k chybám. Nyní dojde k návratu z podprogramu až po závěrné hraně signálu AC.

Motor děrovače je před vlastním děrováním zapínán podprogramem PUNCHON s předstihem asi 0,4 s. Podprogram od návěští PUNPAR opatřuje znak sudou paritou, od návěští PUNBYTE ho ponechává beze změny. Znak vystupuje na bráně PUNCHER. Řízení pove-

lu RUF-A odpovídá vyznačené závislosti na obr. 2.

Podprogramy vstupu a výstupu jsou volány instrukcemi RST 2 a RST 3. Dovoluje to použít stejné programy s různými vstupními resp. výstupními periferiemi. Všechny 4 obslužné podprogramy jsou na adrese začínající 1D, takže stačí měnit pouze 2. byte instrukcí JMP uložených v restartové tabulce. Jelikož zejména podprogram tisku se volá v TWS mnohokrát, ušetří se zároveň několik desítek bytů.

V uvedeném režimu pracuje mikropočítač souběžně s periferií, avšak je rychlejší, a tak většinu času čeká na připravenost periferie nebo na reakci obsluhy. Propojení periferií s mikropočítačem a obslužné podprogramy, v některých případech po úpravě, byly převzaty od vedoucího diplomové práce.

5.1.4 Přerušení a režim po krocích

Rozšíření TK-80 obsahuje jednoduchý obvod, který při vnějším přerušení vyšle na datovou sběrnici instrukci RST s operandem, jenž je nastaven na číslicovém přepínači. Monitor TK-80 pak provede skok do vlastního podprogramu nebo do restartové tabulky v RAM, kam může uživatel vložit instrukce skoku do svých programů.

Při provozu TWS se stane, že potřebujeme předčasně ukončit některou operaci nebo inicializovat TWS při zběhnutí uživatelského programu, jenž ladíme. Pokud z nějakého důvodu požaduje mikropočítač tisk nepovoleného znaku, psací jednotka se na rozdíl od autonomního režimu trvale zablokuje, neboť signál AC=0 nedovolí vyslání dalšího znaku. V těchto situacích stiskneme současně klávesy CONT a ESC. Po jejich uvolnění závěrná hrana signálu INIT vyvolá přerušení a instrukce skoku v příslušném místě restartové tabulky předá řízení inicializačnímu programu

INIT, takže se TWS ohlásí. Signál INIT zároveň odblokuje psací jednotku. Uvedená funkce je možná pouze v případě, že je přerušení mikroprocesorem povoleno, nikoliv tedy po stisknutí klávesy RESET, neboť po hardwarové inicializaci mikroprocesoru je přerušení zakázáno.

TWS při ladění uživatelských programů využívá režim po krocích hardwarově i softwarově realizovaný TK-80. Spuštění uživatelského programu z TWS nečiní potíže, problém je však s návratem do TWS. Při provedení jedné instrukce uživatelského programu dojde k přerušení, které je dále zakázáno po celou dobu činnosti monitoru až do provádění další instrukce. Hardwarový obvod totiž generuje signál přerušení při každé instrukci, tzn. i monitorové, je-li přepínač v poloze STEP. Návrat do TWS by byl možný pouze vložením startovací adresy TWS a spuštěním, což je pro tento účel příliš dlouhá manipulace. Ovládání by rovněž znesnadňovalo neustálé přepínání přepínače, neboť při činnosti TWS by musel být nastaven na AUTO.

Uvedené potíže odstranil malý hardwarový zásah s patřičným softwarovým ošetřením. Přepínač je stále v poloze STEP, avšak po dobu činnosti TWS blokuje log. 0 přivedená z 1. bitu brány C obvodu 8255 TK-80 funkci IO 7474 generujícího přerušení po každé instrukci. Přerušení je povoleno, takže TWS může akceptovat signál INIT. Při normálním spuštění uživatelského programu blokování zůstává, pouze v režimu po krocích je přerušení po každé instrukci připojeno. 1. bit brány C zároveň přepíná operand restartové instrukce. Log. 1 v klidovém stavu (odpovídá nastavení monitorem TK-80) ponechává RST 4 nastavené přepínačem. Log. 0 mění RST 4 na RST 5. Nastavení brány C provádí inicializační program; zároveň nastavuje i bit řízení DMA obvodů displeje TK-80, takže po dobu činnosti TWS je displej vypnut.

Popsané zapojení je blokově znázorněno na obr. 3.

Ke skoku do uživatelského programu se používá monitorový podprogram RESRG. Podprogram BRENT nemohl být k návratu použit, neboť předává řízení monitoru. Z pozice 4 restartové tabulky se provádí skok na podprogram TWSBRENT, který je ze dvou třetin shodný s BRENT.

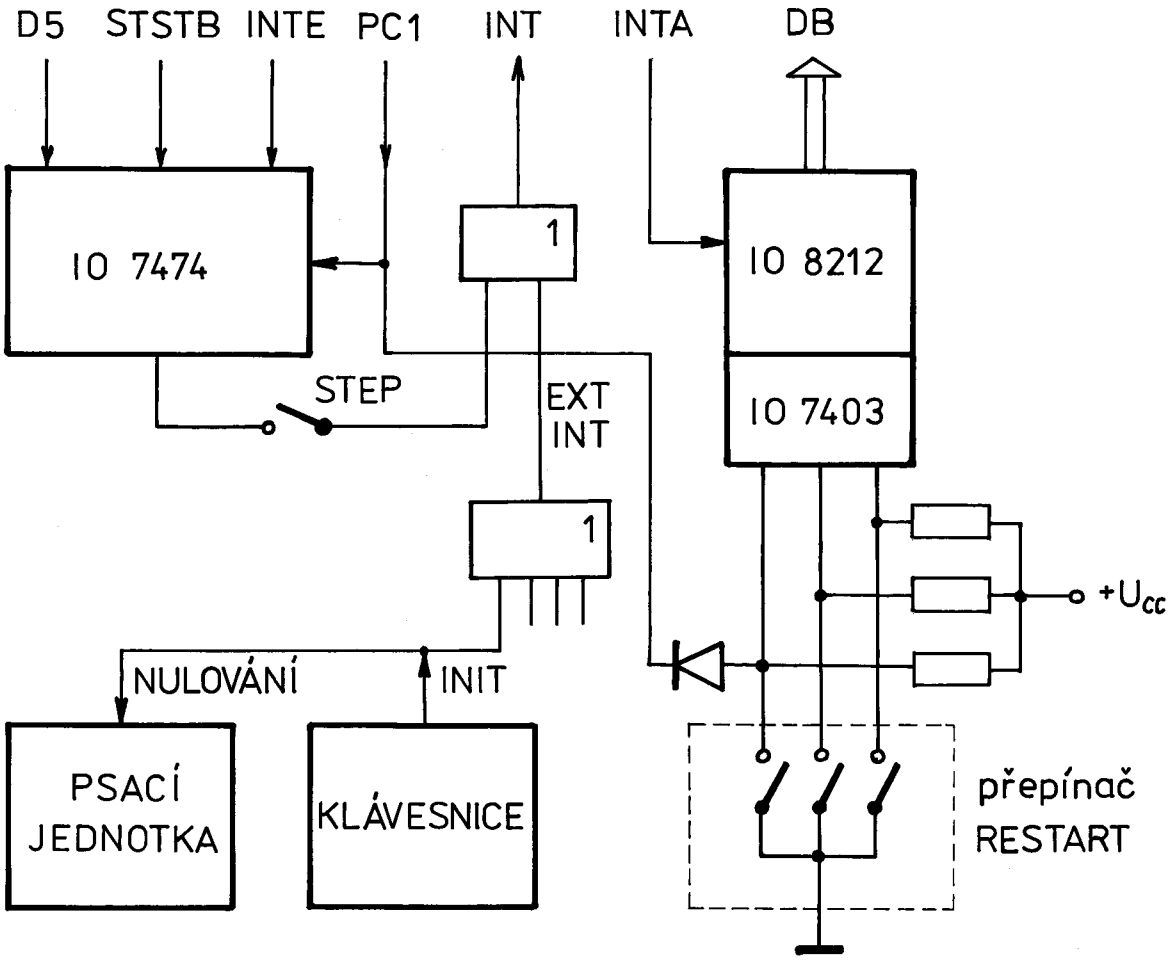
Obsah restartové tabulky:

RST2: JMP KEYINP resp. JMP READBYTE

RST3: JMP PRINT resp. JMP PUNPAR

RST4: JMP TWSBRENT

RST5: JMP INIT



obr. 3

5.2 POPIS PŘÍKAZŮ

Následující část obsahuje seznam příkazů a popis jejich funkce. Parametry jsou označeny symbolicky. Jsou-li v závorce, nemusí být v příkazu uvedeny. Za oddělovač slouží 2 mezery, prováděcí znak ESC není uveden.

5.2.1 Jádro

C p1 cell

Výpis obsahu adresy p1 s možností jeho změny. Po každém stisknutí ESC se vypíše následující adresa a její obsah beze změny obsahu předcházející buňky. Napíšeme-li před ESC číslo, uloží se na uvedenou adresu. Ukončíme současným stlačením ESC+CONT.

LIST p1 p2

Výpis obsahu paměti od adresy p1 do adresy p2 po 16 bytech na řádku do tabulky s hlavičkou.

MOV p1 p2 p3 move

Přesun obsahu bloku paměti od adresy p1 do adresy p2 na novou počáteční adresu p3. Vypíše se nová koncová adresa.

CLEAR p1 p2

Nulování bloku paměti od adresy p1 do adresy p2.

READ

Čtení (celé) děrné pásky a zavedení jejího obsahu do paměti na místo, ze kterého byla vyděrována.

READ1

Čtení 1 bloku děrné pásky a výpis počáteční a koncové adresy bloku.

PUNCH

Děrování obsahu paměti do děrné pásky. Po vyzvání vložíme počáteční a koncovou adresu bloku a stiskneme ESC. Na jednu pásku

lze vyděrovat několik bloků. Ukončíme stisknutím ESC bez vložení adres. Děrování zaváděcí a výběhové části pásky je zajištěno. Nechceme-li děrovat výběhovou část, ukončíme stisknutím ESC+CONT.

PUN

Totéž jako PUNCH, ale neděruje zaváděcí část pásky.

CAL calculate

Provádí sčítání a odčítání hexadecimálních a dekadických čísel modulo 65 536 resp. jejich obousměrný převod. Každému číslu kromě prvního předchází oddělovač a operátor + nebo - . Dekadické číslo začíná znakem #. Po stisknutí ESC se vypíše výsledek hexadecimálně i dekadicky a TWS očekává další příklad. Ukončíme stisknutím ESC+CONT.

R (pl) run

Start uživatelského programu od adresy pl. Při opakovaném startu již nemusí být pl uveden.

DEFINE pl

Definování příkazového slova k označení programu na adrese pl. Program dále spouštíme zapsáním jeho příkazového slova a popř. seznamu parametrů. Příkazové slovo po vyzvání zapíšeme (bez oddělovačů) a stiskneme ESC. Jestliže se vypíše "exist", musíme příkazové slovo mírně upravit a zadat znova.

CANCEL

Zrušení příkazu, jehož příkazové slovo po vyzvání zapíšeme.

5.2.2 Ladění

G pl go

Start uživatelského programu od adresy pl v režimu po krocích. Další kroky provádí po stisknutí ESC se současným zobrazením na displeji (viz 2.2). TWS se ohlásí po stisknutí ESC+CONT.

CE continue

Dokončení uživatelského programu v plynulém režimu.

S (p1) (p2) step

Nejsou-li uvedeny parametry, pokračuje v krokování programu po jedné instrukci. Je-li uvedena adresa zastavení p1, provede část programu až za instrukci na této adrese. Je-li uveden i počet průchodů p2, chod programu se zastaví až po p2 průchodech adresou p1.

RL register list

Výpis obsahu registrů mikroprocesoru včetně F, SP a PC, jaký byl po posledním provedeném kroku uživatelského programu.

RC register change

Totéž jako RL s možností změny obsahu registrů. K dalšímu páru registrů přejdeme stisknutím ESC. Jestliže před ESC zapíšeme číslo, bude následující krok uživatelského programu proveden s novým obsahem registrů. Předčasně můžeme ukončit stisknutím ESC+CONT.

SL stack list

Výpis obsahu zásobníku uživatelského programu počínaje vrcholem.

SC stack change

Totéž jako SL s možností změny obsahu zásobníku analogicky k RC.

REL p1 p2 p3 relocate

Přesun programu od adresy p1 do adresy p2 na novou počáteční adresu p3. Vypíše se nová koncová adresa a jsou definovány hranice přemístěného programu. Všechny operandy tříbytových instrukcí v intervalu ⟨p1,p2⟩ jsou upraveny, takže skokové instrukce přemístěného programu směřují na původní návěští. Totéž platí i pro následující instrukce.

program

PRGM

Vypíše uživatelem definované hranice programu s možností jejich změny. BP (begin of program) znamená začátek, EP (end of program) konec programu. Zapíšeme-li před ESC číslo, hranice se změní, jinak nikoliv. Tento příkaz musíme použít vždy před úpravami uživatelského programu.

insert NOP

INOP p1 p2

Vsune před instrukci na adrese p1 počet p2 instrukcí NOP. Konec programu se příslušně posune.

insert

INS p1

Vkládání instrukcí před instrukci na původní adrese p1. Vypíše se adresa a TWS očekává vložení 1, 2 nebo 3 bytů (oddělených oddělovačem) podle délky vkládané instrukce. Po stisknutí ESC se část programu za místem vložení posune, upraví se koncová hranice programu a vypíše se další adresa. Ukončíme stisknutím ESC bez vložení čísla.

delete

DEL p1 p2

Vypuštění části programu od adresy p1 do adresy p2. Část programu za adresou p2 se přisune, upraví se koncová hranice programu a vynuluje se část paměti do původní koncové adresy.

shift

SHF p1 p2 p3

Totéž jako MOV, avšak v programu se zároveň upraví odkazy na přesouvaný blok, aby směřovaly na jeho nové umístění.

change

CHG p1 p2

V programu hledá tříbytové instrukce s operandem p1, vypisuje jejich adresy a operand mění na p2.

5.2.3 Text

KEY p1 (p2)

Uložení textu vstupujícího z klávesnice do paměti od adresy p1. Je-li uveden p2, znamená maximální povolenou adresu uložení textu. Ukončíme klávesou ESC, vypíše se adresa koncového znaku ESC uloženého textu. Text se ukládá ve zhuštěné formě.

PRINT p1 (p2)
Výpis textu z paměti od adresy p1 do koncového znaku ESC resp. pouze do adresy p2.

STORE
Uložení textu na děrnou pásku. Text se ukládá do vyrovnávací paměti, po jejím zaplnění se samočinně vyděruje včetně 20 oddělovacích blanků. Pak můžeme pokračovat v psaní textu. Kratší blok vytvoříme stisknutím ESC. Stisknutím ESC bezprostředně po děrování provádění příkazu ukončíme. Děrování zaváděcí a výběhové části pásky je zajištěno. Nechceme-li děrovat výběhovou část, ukončíme stisknutím ESC+CONT.

STOR
Totéž jako STORE, ale neděruje zaváděcí část pásky.

TEXT
Výpis textu z děrné pásky. Po výpisu každého bloku čeká na stisknutí ESC. Před koncem pásky ukončíme stisknutím ESC+CONT.

5.3 POPIS PROGRAMŮ

Výpis překladu TWS je v příl. 11. Výsledný překlad byl pořízen až po odladění konečné verze TWS a nemohl už být z časových důvodů opatřen poznámkami. Názvy příkazových programů až na výjimky přibližně odpovídají příkazovým slovům.

5.3.1 Zpracování příkazu

Inicializační a řídící program INIT vždy nastavuje SP na začátek vyhrazené oblasti zásobníku TWS. SP totiž mohl být na-

staven na uživatelský zásobník nebo mohlo předcházet chybové hlášení, které je obvykle doprovázeno mimořádným ukončením podprogramů a tedy porušením kontinuity zásobníku. Dále dojde k nastavení brány C obvodu 8255 a k nastavení adres v restartové tabulce tak, aby vstupní periferií byla klávesnice a výstupní periferií psací jednotka. Po výpisu ohlášení čeká TWS v programu WORKLINE na vstup příkazu.

Program WORKLINE ukládá vstupující text do pracovního pole paměti, jehož počáteční adresa WORKADR musí mít nižší byte nulový. Vyhrazeno je 128 B, které pojmou celý řádek textu i na psacím stroji s dlouhým vozem. Vlastní ukládání provádí podprogram WLIN. K návratu dojde, jestliže je překročena hranice pracovního pole (S=1), jestliže bylo stisknutoESC (Z=1, CY=0) nebo jestliže vstoupil znak shodný s obsahem D (Z=1, CY=1) - WORKLINE však nastavuje (D)=0, takže k tomu v tomto případě nemůže dojít. Vstoupí-li červená uvozovka, nejsou následující znaky ukládány do paměti, dokud nevstoupí druhá červená uvozovka.

INIT pokračuje vyhodnocováním uloženého příkazu. BC slouží jako ukazatel v pracovním poli paměti. Podprogram POINTER posouvá ukazatel přes oddělovací znaky, návrat provede při nalezení buď ESC (Z=1) nebo znaku ze sloupce 2 až 7 (Z=0). Podprogram CODESUM kóduje příkazové slovo do jednoho bytu. Postupně od každého znaku příkazového slova odečte bázi (kód písmene A), přičte ho k mezivýsledku, s nímž provede rotaci doleva o 1 bit. Po nalezení oddělovače se kódování ukončí, výsledek zůstane v E.

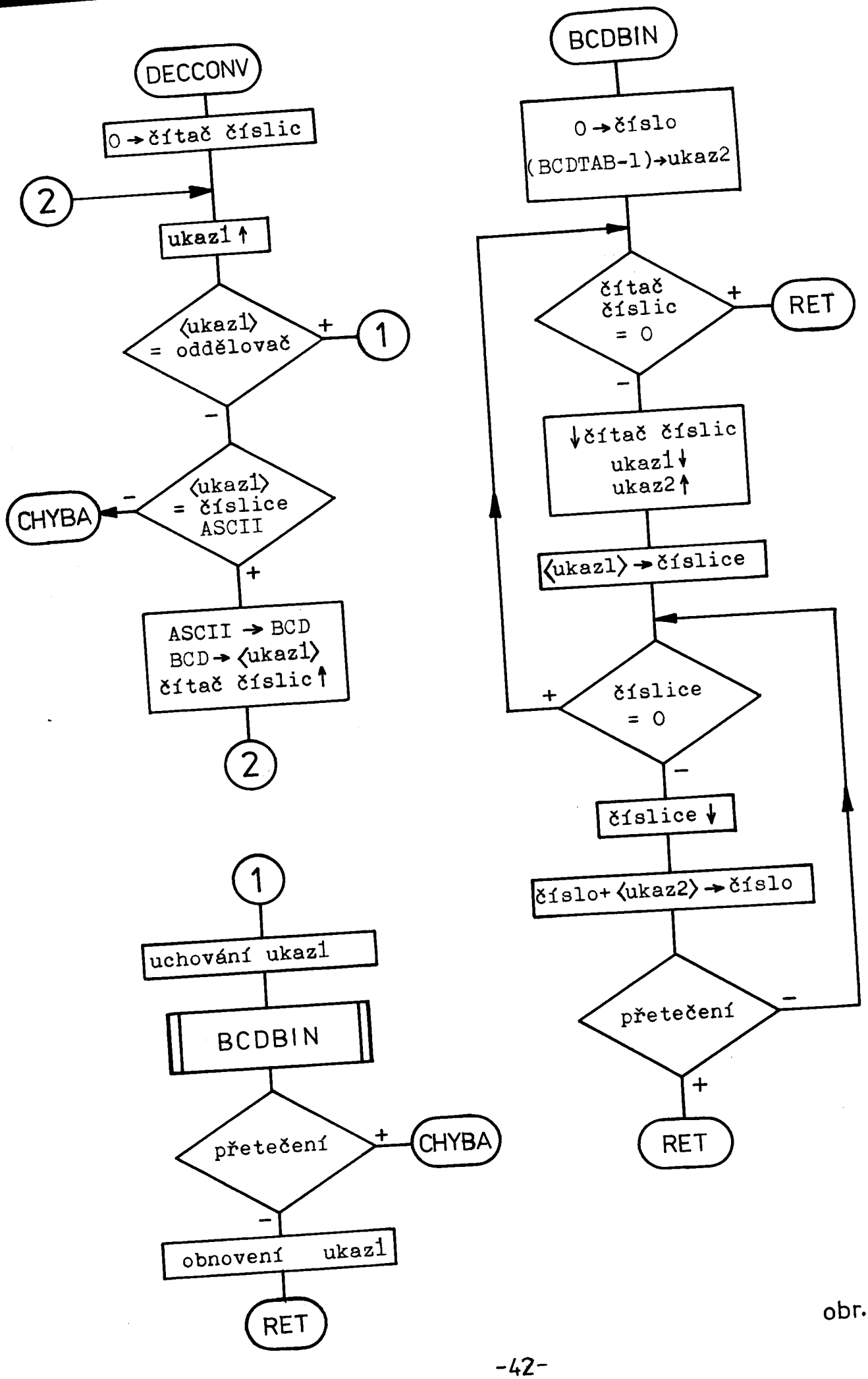
INIT dále ve smyčce čte uložené parametry programem INPUTPAR a ukládá je do zásobníku. D je čítačem parametrů. K návratu z INPUTPAR dojde buď při nalezení ESC (CY=1) nebo po převedení jednoho parametru z pracovního pole do HL (CY=0). Je-li prvním znakem po oddělovači znak #, volá se podprogram DECCONV, jinak

HEXACONV.

DECCONV převádí jedno- až pěticiferné dekadické číslo v ASCII znacích z pracovního pole na dvoubytové binární číslo v HL (obr. 4). D je čítač číslic. DECCONV nejprve určí počet číslic a převede je v pracovním poli z ASCII na BCD. Nalezne-li jiný znak, vrací se s CY=1. Po nalezení oddělovače volá podprogram BCDBIN, který provádí vlastní převod. DE ukazuje na tabulku BCDTAB, ve které jsou uloženy dvoubytové binární ekvivalenty mocnin deseti. Do HL, na počátku vynulovaného, se přičte tolikrát položka tabulky BCDTAB, kolik ukazuje BCD číslice v pracovním poli. Ukazatel BC se pak posune o jednotku zpět, ukazatel DE naopak o 2 vpřed a zpracovává se další číslice, dokud se nevynuluje čítač číslic A. Dojde-li k přetečení HL, podprogram se ukončí s CY=1. DECCONV na závěr nastaví ukazatel BC za zpracovaný parametr.

HEXACONV převádí čtyřciferné hexadecimální číslo v ASCII znacích z pracovního pole na dvoubytové binární číslo v HL. Nalezne-li oddělovač, ukončí převod s CY=0. Každý znak je zpracovávám podprogramem ASCHEXA, který původně nulový obsah HL posouvá o 4 bity vlevo a na nejnižší nibble ukládá hexadecimální číslici, převedenou z ASCII podprogramem ASCHEX1. Objeví-li se nepovolený znak, ukončí se postupně všechny podprogramy s CY=1.

Smyčka zpracování parametrů se ukončí při nalezení ESC na konci řádku (INPUTPAR vrátí CY=1). Podprogram TABLE1 pak hledá v tabulce adres příkazových programů JMPTAB klíč shodný s kódem příkazového slova v E. HL je ukazatelem v tabulce. Každá položka tabulky obsahuje klíč a dvoubytovou adresu. Konec tabulky je označen klíčem 00. Neobsahuje-li tabulka hledaný klíč, podprogram vrací CY=0. Je-li položka nalezena, uloží se adresa do DE, HL ukazuje na druhý (vyšší) byte adresy a CY=1. Adresa příkazového


DECCONV
0 → čítač číslic
2
ukaz1 ↑
⟨ukaz1⟩ = oddělovač
+
1
−
⟨ukaz1⟩ = číslice ASCII
CHYBA
−
+
ASCII → BCD
BCD → ⟨ukaz1⟩
čítač číslic ↑
2
1
uchování ukaz1
BCDBIN
přetečení
+
CHYBA
−
obnovení ukaz1
RET
BCDBIN
0 → číslo
(BCDTAB-1) → ukaz2
čítač číslic = 0
+
RET
−
↓čítač číslic
ukaz1↓
ukaz2↑
⟨ukaz1⟩ → číslice
číslice = 0
+
−
číslice ↓
číslo+⟨ukaz2⟩ → číslo
přetečení
−
+
RET

obr. 4

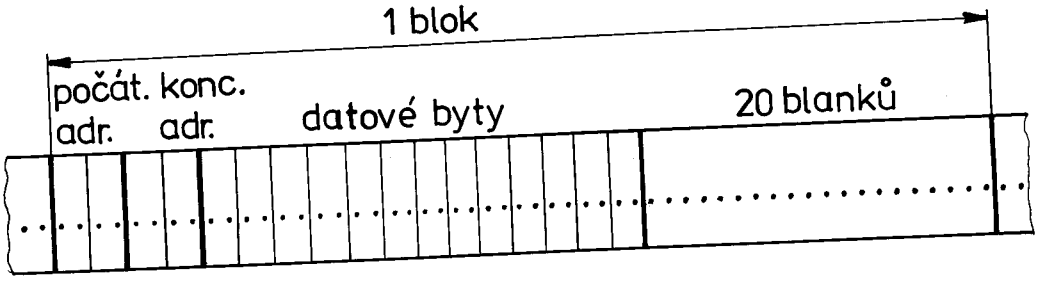
programu je uložena do zásobníku, C obsahuje počet parametrů. Je-li CY=0, provede se chybové hlášení. V opačném případě se do A uloží počet parametrů zmenšený o jednotku s nastavením příznaků a provede se skok na adresu uloženou na vrcholu zásobníku. Řízení se tak předá příslušnému příkazovému programu, který nejprve zjišťuje, zda byl vložen správný počet parametrů, jež pak vyzvedne ze zásobníku a uloží do registrů. V některých případech počet parametrů určuje použitou variantu programu. Zjišťuje se podle příznaků nebo v podprogramu PARAM2 (požadavek na 2 parametry) resp. PARAM3 (3 parametry). Tento podprogram ukládá parametry do HL, DE resp. i BC, ovšem musí přeorganizovat zásobník narušený uložením návratové adresy. Ke skoku na chybové hlášení dojde nejen při nesprávném počtu parametrů, ale i v případě, že je druhý parametr menší než první.

Program MESSAGE zajišťuje tisk chybových hlášení. Vstupující obsah E po vynulování sedmého bitu tvoří nižší byte adresy v tabulce textu MESTAB. Chybové hlášení z MESTAB vypíše podprogram TEXTOUT. Je-li podle obsahu E příznak S=1, vrátí se řízení nadřazenému programu, jinak následuje skok na INIT. Obsah E se nastavuje obvykle již na začátku programu, avšak pouze při výskytu chyby dojde k podmíněným návratům z podprogramů a k podmíněnému skoku na MESSAGE.

5.3.2 Snímání děrné pásky

V EPROM jsou uloženy všechny potřebné programy pro snímání děrné pásky. Program MONREAD se spouští od adresy 1D00 monitorem TK-80. Volá podprogram READBLOK, který zapíná podprogramem READ-ON snímač. BLANK propouští prázdnou pásku až do počtu (B) blanků, pak se vrací se Z=1, čehož se využívá ke zjištění konce pásky. Na začátku snímání se do B vkládá maximální hodnota. Narazí-li

BLANK na znak, vrací Z=0. Není-li konec pásky, volá READBLOK vlastní snímání bloku RBLOCK. Podprogram READWORD snímá z pásky, kontroluje a převádí do HL 1 až 4 znaky podle (B). Využívá k tomu podprogram ASCHEXA. RBLOCK sejme 4 znaky nejprve počáteční adresy bloku, kterou uloží na buňku ADDR, a pak koncové adresy. Počáteční adresa je v HL, koncová v DE. Následuje ve smyčce snímání a převod 2 znaků a ukládání bytu do paměti adresované HL. Smyčka se ukončí při rovnosti (HL) a (DE), kterou vyhodnotí RPEQU příznakem Z=1. Formát dat na pásce znázorňuje obr. 5.



obr. 5

RBLOCK při návratu vrací v HL koncovou adresu bloku. Nastavení příznaků určuje chybu, konec pásky nebo bloku. Podle toho READBLOK snímač buď zastaví nebo i vypne a MONREAD buď skočí na monitorový podprogram ERROR, který na displeji zobrazí "E.....", nebo předá řízení monitoru nebo pokračuje zobrazením počáteční a koncové adresy bloku na displeji monitorovým podprogramem RGDSP. Následuje čekání v dalším monitorovém podprogramu KEYIN na stisknutí libovolné klávesy TK-80 a snímání dalšího bloku.

Oba příkazové programy READ a READ1 využívají rovněž podprogram READBLOK, při chybě však tisknou chybové hlášení. READ čte celou pásku najednou, READ1 pouze 1 blok, ale současně podprogramem OUTADR2 vytiskne počáteční a koncovou adresu.

5.3.3 Vstup a výstup čísel

V 5.3.1 byly popsány základní podprogramy pro vstup čísel. Jim nadřazené jsou programy INADR1 a INPARAM. INADR1 tiskne výz-

vu a ukládá parametry do pracovního pole, INPARAM navíc převádí 1 parametr do HL a do A (nižší byte).

Podprogram HEXASCII zajišťuje výstup čtyřciferného hexadecimálního čísla uloženého do HL. Volá 2x HEXASC2, který pokaždé podprogramem HEXASC1 převádí postupně oba nibble v A z hexadecimálního kódu na ASCII.

Programy OUTADR1 a OUTADR2 zajišťují číslu tištěnému podprogramem HEXASCII grafickou úpravu požadovanou v TWS.

Výstup dekadického čísla provádí program BCDOUT. Nejprve zajišťuje grafickou úpravu, pak volá BINCON, který uchovává obsahy registrů. Vlastní převod binárního čísla z HL na dekadické provádí podprogram BINBCD. Pěticiferné dekadické číslo ukládá v BCD do paměti, počínaje číslicí nejvyššího řádu od návěští DIGITS. BINBCD postupně odečítá v proceduře DECNO od čísla v HL mocniny deseti. Každou mocninu deseti odečte tolikrát, aby (HL) bylo právě ještě větší než 0. Počet provedených rozdílů se uloží z A jako číslice příslušného řádu do buňky paměti. BCDOUT pak v cyklu převádí číslice BCD na ASCII a tiskne je podprogramem HEXASC2+10, ovšem vyjma vedoucích nul. (B) slouží jako vlajka - určuje, zda již byla nějaká číslice tištěna. Číslice nejnižšího řádu je tištěna vždy, tedy i 0.

Převodní podprogramy ASCHEX1, HEXASC1 a BINBCD byly převzaty z /2/ a /14/.

5.3.4 Děrování děrné pásky

Příkazový modul PUNCH začíná zápisem počáteční a koncové adresy bloku a převodem počáteční adresy do DE v programu INPARAM. Podprogram PUNCHON zapíná motor děrovače a zavoláním DELAY ponechává čas pro jeho rozběh. DELAY realizuje zpoždění, jehož délka závisí na hodnotě vložené do A podle vztahu

$T = 0{,}06 \cdot (A)^2$ /ms/. Nejdelší zpoždění je tedy asi 4 s.

Pokud nebyly vloženy adresy (INPARAM vrátil CY=1), následuje děrování výběhové části pásky podprogramem BLANKOUT, který děruje (B) blanků, jinak následuje skok na PCH1. INPUTPAR převede do registru HL koncovou adresu, zamění se (HL) a (DE) a provede se porovnání obou adres v PARAM2+7. Jestliže program PUNCH začal s CY=1, což je při spuštění řídícím programem INIT splněno vždy, vyděruje se zaváděcí část pásky. Teprve pak následuje vlastní děrování bloku podprogramem PUNBLOK.

PUNBLOK nastaví změnou v restartové tabulce jako výstupní periferii děrovač a dvojím provedením HEXASCII vyděruje počáteční a koncovou adresu. Pak vybírá v cyklu z paměti byte za bytem a děruje je podprogramem HEXASC2, dokud se ukazatel HL neztotožní s koncovou adresou bloku v DE. Porovnání provádí RPEQU. Za účelem oddělení bloků je pak vyděrováno 20 blanků, na výstup nastavena psací jednotka a řízení předáno nadřazenému programu.

PUNCH probíhá znova, ovšem s CY=0, takže stejně jako u příkazu PUN, který skáče na návěští PUNCH1, nedojde již k děrování zaváděcí pásky.

5.3.5 Vstup a výstup textu

Text se v paměti mikropočítače ukládá v ASCII, bez parity, 7. bit = 1 znamená červenou barvu a pro výstup do psací jednotky musí být v podprogramu PRINT invertován.

Následuje-li bezprostředně za sebou dva a více stejných znaků, jsou v paměti uloženy zhuštěně ve zvláštním kódu na 2 bytech. První byte obsahuje příslušný znak, druhý byte s hodnotou v intervalu ⟨81,9F⟩ nese informaci o počtu opakování znaku. Tento byte odpovídá "červeně tištěným" zvláštním funkcím psací jednotky (příl. 2), a pokud nebudeme tisknout klávesu RED při

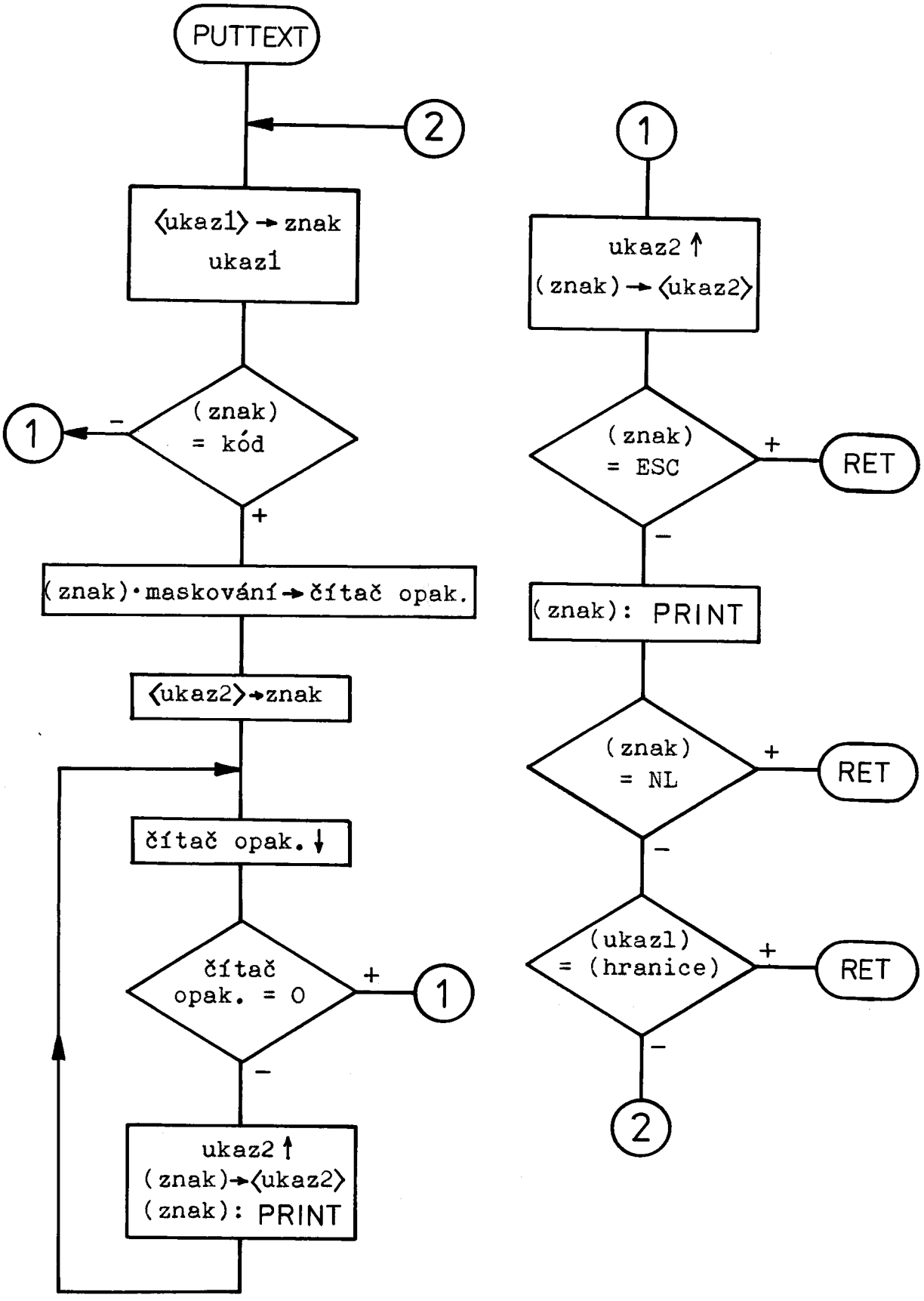
provádění nového řádku nebo jiné funkce ze sloupců 0 a 1, nemůže se v textu objevit. Počet opakování je (kód - 80H); jednou je znak tištěn ještě před identifikací kódu, takže znak může být vytištěn 32x. Přirozeně může následovat několik kódů za sebou nebo naopak několik stejných znaků může být za sebou uloženo bez kódování.

Zhuštění textu šetří objem paměti při ukládání grafických úprav (zejména podtrhávání) a při ukládání informace o sázení tabulátorových zarážek (vůz je na potřebné pozice dopravován mezerníkem).

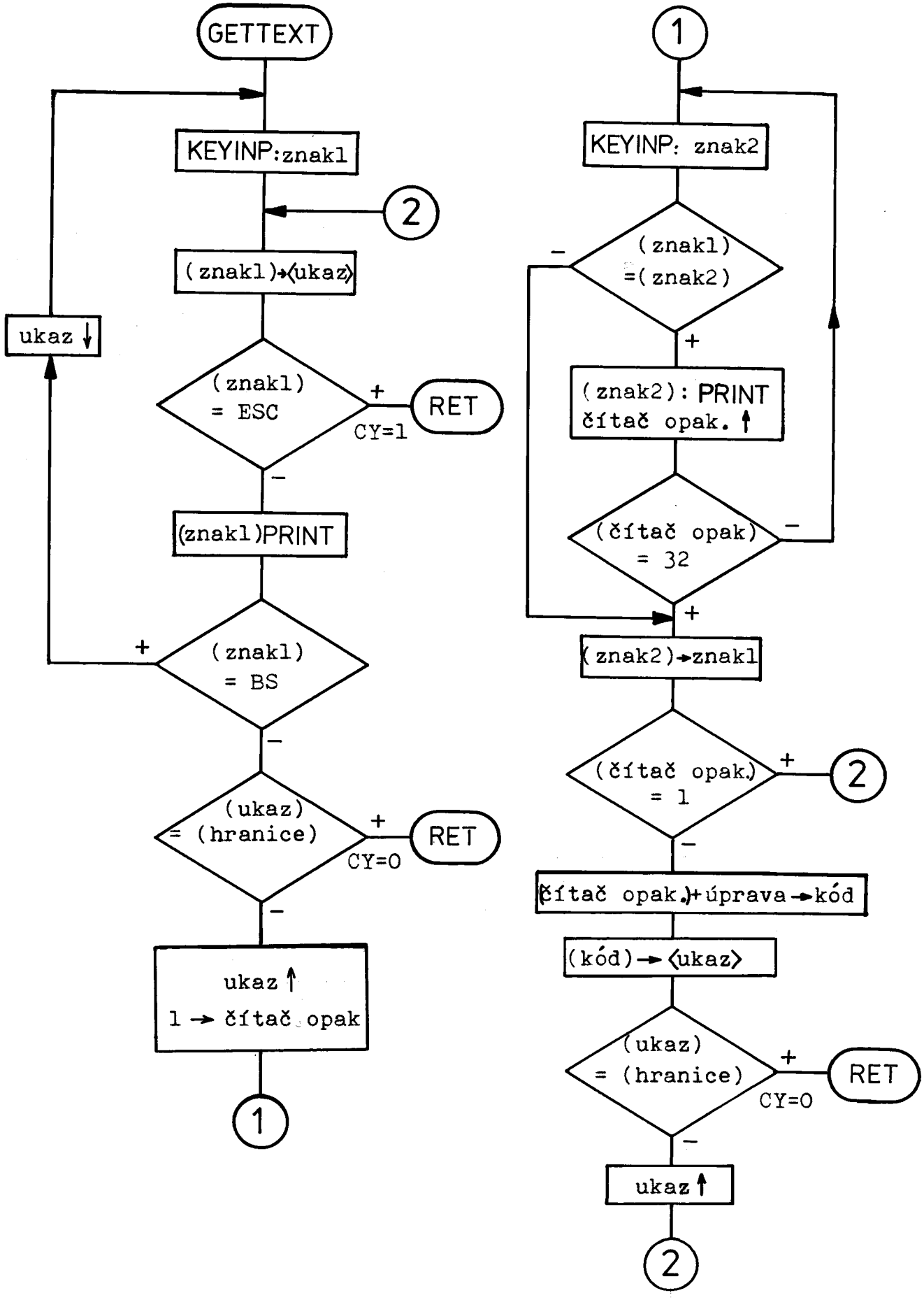
Program TEXTOUT uchovává a nastavuje registry pro vlastní podprogram výpisu textu PUTTEXT. V HL je počáteční a v DE maximální koncová adresu textu.

PUTTEXT vybírá byty z paměti a zjišťuje, zda se nejedná o kód. Znaky rozvinutého textu jsou na návěští LOOPMEM ukládány též do pracovního pole paměti, aby bylo možno při dalším rozvoji TWS snáze aplikovat textový editor. Je-li znak ESC nebo NL, je podprogram ukončen s CY=0 resp. 1. Po vytištění znaku je proveden test ukazatele HL vzhledem ke koncové adrese s možností návratu a následuje zpracování dalšího znaku. Jestliže se jedná o kód, maskováním se získá počet opakování, který v D určuje počet cyklů od návěští RPEAT, v nichž se předcházející znak tiskne a ukládá do pracovního pole. Hrubý vývojový diagram je na obr. 6.

Program GETTEXT (obr. 7) začíná vstupem znaku z klávesnice. HL je ukazatelem, který se nastavuje nadřazeným programem na počáteční adresu, DE obsahuje maximální koncovou adresu. Znak je uložen do paměti. Je-li ESC, dojde k návratu s CY=1. Po vytištění se srovnává s BS a popř. se provede posuv ukazatele zpět a vstup nového znaku. Jinak následuje test ukazatele ke koncové adrese a jeho posuv. Znak je uchován v C, čítač opakování B je


PUTTEXT
2
⟨ukaz1⟩ → znak
ukaz1
(znak)
= kód
1
−
+
(znak)·maskování→čítač opak.
⟨ukaz2⟩→znak
čítač opak. ↓
čítač
opak. = 0
+
1
−
ukaz2 ↑
(znak)→⟨ukaz2⟩
(znak): PRINT
1
ukaz2 ↑
(znak) → ⟨ukaz2⟩
(znak)
= ESC
+
RET
−
(znak): PRINT
(znak)
= NL
+
RET
−
(ukaz1)
= (hranice)
+
RET
−
2

obr. 6


GETTEXT
KEYINP:znak1
2
(znak1)→⟨ukaz⟩
ukaz ↓
(znak1) = ESC
+
RET
CY=1
−
(znak1)PRINT
+
(znak1) = BS
−
(ukaz) = (hranice)
+
RET
CY=0
−
ukaz ↑
1 → čítač opak
1
1
KEYINP: znak2
(znak1) =(znak2)
−
+
(znak2): PRINT
čítač opak. ↑
(čítač opak) = 32
−
+
(znak2)→znak1
(čítač opak.) = 1
+
2
−
čítač opak.)+úprava→kód
(kód)→⟨ukaz⟩
(ukaz) = (hranice)
+
RET
CY=0
−
ukaz ↑
2

obr. 7

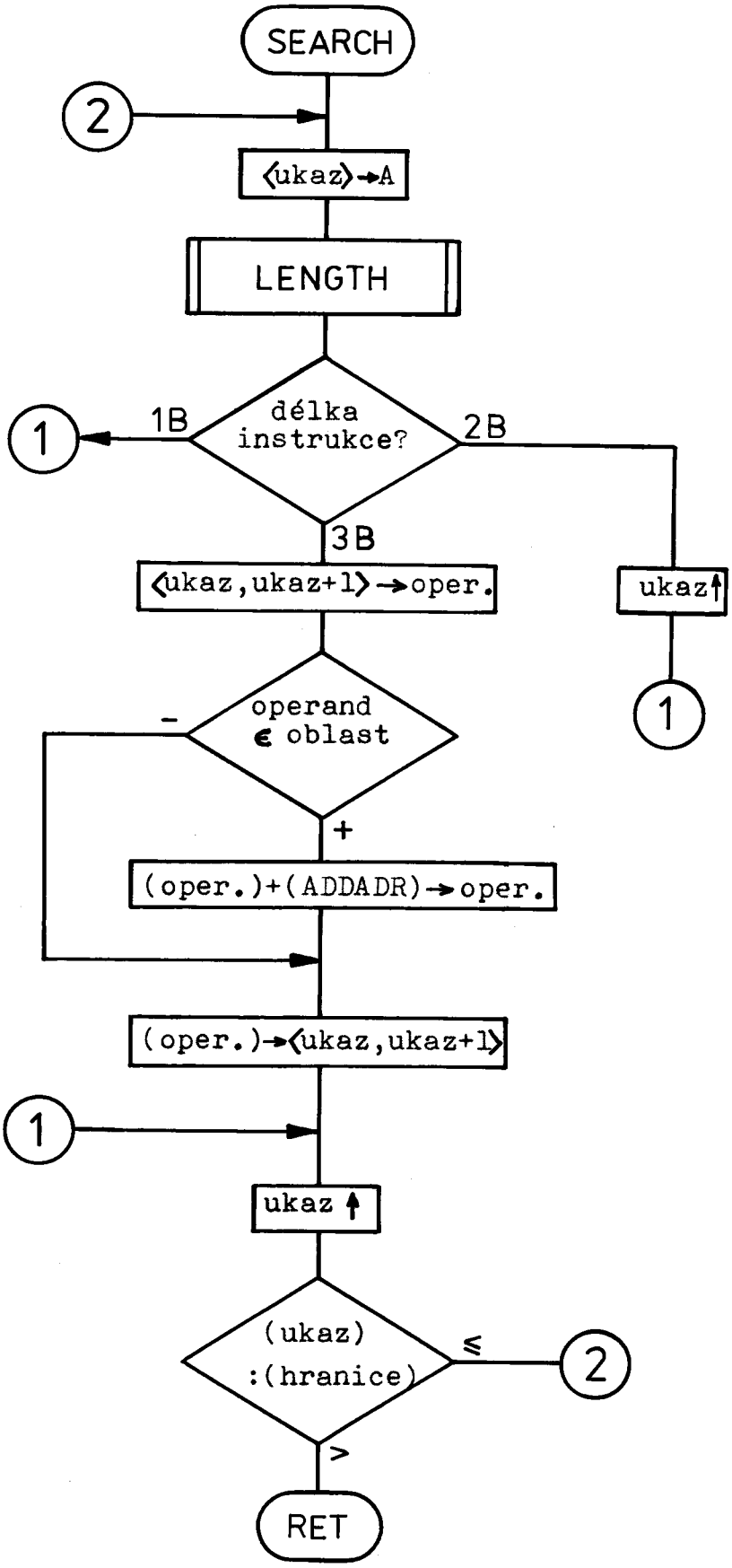
nastaven na 1. Nový znak, který vstoupil na návěští CHCOUNT, je porovnán se starým. V případě shody je vytištěn, B je inkrementován a smyčka se může uzavírat přes CHCOUNT tak dlouho, dokud B nenačítá 32. Pak, stejně jako při neshodě dvou po sobě následujících znaků, pokračuje podprogram od GT1 testem B. Je-li (B)=1, pokračuje se uložením znaku na návěští GETTEXT+1, v opačném případě slouží (B) k utvoření kódu, který je uložen do paměti. Ukazatel je opět testován na koncovou adresu a na GETTEXT+1 je do paměti uložen znak, který vstoupil naposled.

5.3.6 Relokace uživatelských programů

Při relokaci programů nebo jejich částí je třeba upravit adresy skoků. Pro jednoduchost upravujeme všechny operandy tříbytových instrukcí, což přináší některé výhody i nevýhody (viz 5.4), avšak umožňuje použít podprogram LENGTH jak k posuvu ukazatele na začátek instrukce tak i k určení, zda se jedná o instrukci podléhající případné úpravě.

Do akumulátoru se uloží 1. byte instrukce a LENGTH vrátí Z=0, jedná-li se o jednobytovou instrukci, Z=1, CY=1 při dvoubytové a Z=1, CY=0 při tříbytové instrukci. Používá k tomu maskování a porovnávání.

Program SEARCH (obr. 8) provádí úpravu programu od adresy (P1) do (P2). Je-li dvoubytový operand v intervalu ⟨(BC),(DE)⟩ , přičte k němu (ADDADR). HL je ukazatelem v programu, P1, P2 a ADDADR jsou proměnné uložené v paměti. SEARCH začíná zkoumáním instrukce podprogramem LENGTH. Jedná-li se o jedno- nebo dvoubytovou instrukci, provede se od návěští SH1 pouze posuv a test ukazatele na koncovou adresu (P2). Není-li ještě konec programu, přejde se na jeho další instrukci. U tříbytové instrukce se operand převede do HL a testuje se vůči (BC) a (DE) v podprogramu


SEARCH
2
⟨ukaz⟩→A
LENGTH
délka instrukce?
1B
2B
3B
1
⟨ukaz,ukaz+1⟩→oper.
ukaz↑
1
operand ∈ oblast
−
+
(oper.)+(ADDADR)→oper.
(oper.)→⟨ukaz,ukaz+1⟩
1
ukaz ↑
(ukaz) :(hranice)
≤
2
>
RET

obr. 8

RPCOMP.

RPCOMP vrací CY=0, je-li (HL) v intervalu ⟨(BC),(DE)⟩, jinak CY=1. Využívá k tomu RPSUB, který odečítá (HL) - (DE) → HL a zároveň při (HL)<(DE) nastavuje CY=1, jinka CY=0. Od návěští RPCOMPDH je testován podprogramem RPSUB (HL) vůči (DE), ovšem stejně jako v RPCOMP je původní obsah registrů zachován.

Podle výsledku testu se buď od návěští SH2 zpět do paměti uloží původní operand nebo se k němu nejprve přičte (ADDADR). Pak SEARCH pokračuje stejně jako u kratších instrukcí.

Program PLACE vloží do uživatelského programu (DE) instrukcí NOP před instrukci na původní adrese (HL). Část programu od adresy (HL) se zřejmě musí posunout, přičemž všechna její návěští se zvětší o (DE). (DE) se uloží na ADDADR, adresa vložení v (HL) se přesune do BC, (P2) do DE a (P1) do HL. Program SEARCH pak upraví v celém uživatelském programu přičtením počtu NOP adresy skoků na návěští, jež jsou v intervalu ⟨adresa vložení, konec programu⟩. Vlastní posunutí upravené části programu zajistí podprogram MOVEBLOK. P2 se nastaví na novou koncovou adresu.

MOVEBLOK přesouvá blok od adresy (HL) do adresy (DE) na novou počáteční adresu (BC) a vrací v (HL) novou koncovou adresu. Jestliže nový blok překrývá začátek starého bloku, přesouvá se blok od začátku (návěští DIRLOOP), jestliže nový blok překrývá konec starého, přesouvá se od konce (BACKLOOP). Nejprve se podprogramem RPSUB určí z počáteční a koncové adresy počet přesouvaných bytů a pak se porovnáním staré a nové počáteční adresy v RPEQU zvolí použitá varianta. Je-li (DE)>(HL), vrací RPEQU CY=1, jinak CY=0. Počet přesouvaných bytů se uloží do BC a podprogramem BCTEST se v obou variantách testuje konec přesunu. BCTEST dekrementuje BC, při nulovém obsahu vrací Z=1.

5.3.7 Operace s tabulkou adres

Tabulku adres příkazových programů JMPTAB upravují příkazové programy DEFINE a SCRATCH. Program CODETAB provádí vstup a zakódování příkazového slova, jehož kód pak porovnává s klíči JMPTAB. K návratu dojde za stejných podmínek jako u podprogramu TABLE1.

DEFINE podle výsledku CODETAB buď vytiskne hlášení nebo uloží na konec JMPTAB kód, parametr a koncovou značku.

SCRATCH rovněž podle výsledku CODETAB buď vytiskne hlášení nebo část tabulky za nalezeným klíčem přisune podprogramem MOVEBLOK tak, že se klíč a odpovídající adresa přemaže a příkaz je tak zrušen.

Upozorňujeme na to, že není sledováno přeplnění JMPTAB (maximálně 32 položek), a proto v případě definování uživatelských příkazů je třeba podle použité kombinace bloků TWS omezit počet příkazů. Při používání příkazu CANCEL je třeba dbát na správné vložení příkazového slova - mohlo by se totiž stát, že nesprávné příkazové slovo bude mít kód shodný s kódem příkazu TWS, takže by došlo k jeho zrušení.

5.3.8 Ostatní příkazové programy jádra

Funkce CELL a MOVE je podle příl. 11 vzhledem k předchozímu výkladu zřejmá.

Program CLEAR používá obecný podprogram CLR k nulování paměti od adresy (HL) do (DE). Ukončení cyklu při shodě ukazatele HL s (DE) testuje RPEQU.

Program RUN nastavuje buňku obsahující aktuální stav SP při zpracování uživatelského programu na uživatelský zásobník USESP. Byl-li vložen parametr (Z=1), uloží ho na registr adres ADDR (ADRES podle /1/). GOTO je návěští spouštěcího programu v moni-

toru TK-80.

LIST začíná tiskem hlavičky - nejprve z řetězce STRING2 uloženého v paměti, pak v cyklu. Vypsat 16 hexadecimálních číslic s oddělujícími mezerami v programu je úspornější než je tisknout z paměti. A obsahuje číslici, která se v cyklu od návěští TITLOOP převádí na ASCII a tiskne podprogramem HEXASC2+10.

Na začátku každého řádku se po úpravě tiskne z HL podprogramem OUTADR1 adresa s poslední číslicí 0. Končí-li počáteční adresa bloku nulou, začne se podprogramem HEXASC2 tisknout obsah paměti v cyklu od návěští LLOOP2. Ukazatel HL se testuje vůči konci bloku (DE), (B) je parametr cyklu - zajišťuje tisk 16 bytů na řádek a pak skok na tisk adresy dalšího řádku. Jestliže počáteční adresa bloku nekončí nulou, je nutno zajistit tisk prvního bytu od správné pozice tabulky. Maskováním počáteční adresy se určí počet pozic, které je třeba přeskočit, což se provede v cyklu od návěští SPLOOP posuvem vozu mezerami. (C) je parametr cyklu. Dále následuje již popsaná činnost. Po vytištění posledního bytu bloku se program ukončí z libovolné pozice tabulky.

CAL nejprve zapíše programem INPARAM do pracovního pole celý příklad a první operand převede do HL. Nalezne-li POINTER znak ESC, provede se tisk výsledku z HL hexadecimálně programem OUTADR2, dekadicky programem BCDOUT a následuje vstup dalšího příkladu. Nalezne-li POINTER jiný znak, uchová se v zásobníku. Následující operand se převede programem INPUTPAR, uloží se do DE a odečte se podprogramem RPSUB od mezivýsledku. Původní znak (operátor) se vybere ze zásobníku. Je-li - , čte se od návěští CALNEXT další operand, jinak se operand 2x přičte k mezivýsledku, takže se provedlo sčítání. Je-li operátor +, pokračuje se rovněž od CALNEXT, jinak se provede chybové hlášení. Tato situace na-

stane, např. zapomeneme-li vložit mezi operandy operátor.

5.3.9 Příkazové programy bloku "ladění"

Příkazy RL a RC používají společný program REG. (D) určuje hodnotou OF, že bude proveden pouze výpis, nebo hodnotou FO, že bude možnost změny obsahu. (E) určuje počet vypisovaných položek.

REG nejprve nastavuje ukazatel HL na oblast paměti REGAREA vyhrazenou pro uložení obsahu registrů a ukazatel BC na textový řetězec REGNAME, ve kterém jsou uloženy zkratky registrů. Následuje tisk 2 znaků z REGNAME, grafické úpravy podprogramem OUTADR1+6 a první dvoubytové položky oblasti REGAREA. Z určitých důvodů se k tomu 2x používá HEXASC2 místo HEXASCII. Ukazatele jsou průběžně posouvány; HL vzad, neboť položky oblasti REGAREA jsou uspořádány v paměti sestupně. Podle (D) se provede buď ihned skok na RSKIP, kde se testuje parametr cyklu (E) na ukončení programu, nebo se provede programem INPARAM vstup z klávesnice. Nebylo-li vloženo číslo, pokračuje se od RSKIP, jinak se číslo vloží místo původního obsahu do oblasti REGAREA. Program pokračuje ve smyčce až do konce oblasti REGAREA.

Příkazy SL a SC používají společný program STACK analogicky k REG. Variantu programu určuje (C). Je-li OF, provede se výpis, je-li FO, provede se i změna obsahu.

STACK podprogramem HEXASCII vypisuje obsah zásobníku. Ukazatelem je HL, na počátku se obsadí stavem SP uloženým v oblasti REGAREA. Programové řešení změny obsahu odpovídá programu REG, ukazatel HL je však posouván vpřed. Cyklus programu je ukončen při shodě ukazatele s počátkem uživatelského zásobníku v DE.

PRGM využívá program REG, nastavuje ovšem ukazatel HL na oblast uložení P1 a P2, BC na řetězec BOUNDNAM obsahující označe-

ní hranic programu a DE tak, aby byl proveden výpis 2 položek s možností změny.

REL přesune programový blok podprogramem MOVEBLOK a vypíše koncovou adresu programem OUTADR2. Úpravu skoků směřujících do původního programu provede SEARCH. Posunutí (ADDADR) se určí z rozdílu nové a staré koncové hranice podprogramem RPSUB.

INOP obsahuje pouze volání programu PLACE.

INS začíná výpisem adresy vložení podprogramem HEXASCII. Vkládané byty vstupují programem INPARAM. Bylo-li stisknuto pouze ESC, program se ukončí, jinak se ve smyčce od návěští IS1 čtou byty, které se zatím neukládají. Pouze do E se načítá počet bytů, podle kterého se programem PLACE vytvoří v uživatelském programu mezera o příslušné délce. Od návěští IS2 se ve smyčce opět čtou byty z pracovního pole a nyní se již ukládají do vytvořené mezery. Program pokračuje výpisem následující adresy a vložením další instrukce.

DEL nejprve určí z rozdílu koncové a počáteční adresy vypouštěné části programu posunutí (ADDADR) podprogramem RPSUB. Následuje přisunutí druhé části programu, takže se vypouštěná část přemaže. Posune se i koncová hranice programu (P2) a část paměti mezi současnou a původní hranicí je vynulována podprogramem CLR. Konec programu je společný s koncem programu SHIFT.

SHIFT přesouvá obsah bloku paměti stejně jako program MOVE. Následuje úprava adres skoků v uživatelském programu mezi hranicemi (P1) a (P2) programem SEARCH. Posunutí (ADDADR) se vypočte jako rozdíl mezi novou a starou koncovou adresou.

CHANGE prohledává uživatelský program. Podle výsledku podprogramu LENGTH se posouvá ukazatel HL na začátek instrukcí. Operand tříbytové instrukce se načte do HL a porovnává se s (DE), kde je uložen 1. parametr příkazu. Neshodují-li se, od návěští

CH3 následuje test konce uživatelského programu podprogramem RPCOMPDH a zpracování další instrukce. Shoduje-li se operand a 1. parametr, vytiskne se podprogramem HEXASCII adresa 1. bytu instrukce a 2. parametr (nový operand) se z BC uloží do paměti na místo původního operandu.

Podprogramy TWSBRENT a BRKSTOP vycházejí z odpovídajících podprogramů monitoru TK-80. TWSBRENT s výhodou používá podprogram RPEQU. Pro urychlení chodu uživatelského programu do adresy zastavení nejsou trasovací informace podprogramem ADDSP zobrazovány na displeji průběžně, ale až po zastavení. BRKSTOP také zablokuje signál přerušení od prováděné instrukce, povolí přerušení a očekává stisknutí ESC. Následuje skok na konec programu STEP, který zakáže přerušení, odblokuje signál přerušení a pokračuje monitorovým podprogramem RESRG, který provede další instrukci uživatelského programu.

GO nastavuje SP na uživatelský zásobník a startuje program.

STEP podle příznaků nastavených počtem parametrů v A provádí 1 ze 3 variant nastavení adresy zastavení a počtu průchodů.

5.3.10 Příkazové programy bloku "text"

Program TEXTIN nastavuje registry pro podprogram GETTEXT. Není-li zadán 2. parametr příkazu KEY, DE se nastaví na maximální hodnotu, aby nedošlo k nežádoucímu návratu. Je-li maximální koncová adresa zadána, vloží se do její buňky znak ESC. Vrátí-li GETTEXT CY=0, bylo dosaženo koncové adresy a provede se hlášení "out". V opačném případě vytiskne OUTADR2 v HL vrácenou adresu posledního znaku.

TEXTPRN nastavuje registry pro podprogram PUTTEXT. Protože PUTTEXT provádí návrat také po provedení nového řádku, obsahuje program smyčku od návěští TP2.

STORE začíná děrováním zaváděcí pásky, nastavením registrů pro podprogram GETTEXT a uložením znaku ESC na konec vyrovnávací paměti. Stiskneme-li ESC, aniž bychom vložili text, provede se skok na děrování výběhové části pásky (PUNCH+10), neboť ESC je první znak ve vyrovnávací paměti. Po ukončení textu klávesou ESC nebo zaplnění vyrovnávací paměti se nastaví jako výstupní periferie děrovač a programem TEXTOUT se text vyděruje. Koncový znak ESC se musí vyděrovat samostatně. Psací jednotka je opět definována jako výstupní periferie. Program vyděruje 20 oddělovacích blanků a v cyklu pokračuje.

TEXT propouští blanky děrné pásky, čte znaky textu, kontroluje paritu a ukládá je do vyrovnávací paměti. Dojde-li k zaplnění vyrovnávací paměti (testuje RPEQU), doplní se ESC a stejně jako při nalezení ESC na pásce se uložený text vypíše programem TEXTOUT. TEXT pak čeká na stisknutí ESC, aby vypsal z pásky další blok. Zjistí-li BLANK výběhovou část pásky, program se ukončí.

5.4 POZNÁMKY K PROVOZU TWS

Chceme-li spustit TWS, zavedeme po zapnutí MVS děrné pásky zvolené kombinace (viz 5.1.1) programem MONREAD (adresa 1DOO). Přepínač STEP/AUTO nastavíme na STEP, číslicový přepínač na č.4 a odstartujeme TWS od adresy 3DOO resp. ODOO. TWS se ohlásí a můžeme vložit příkaz.

Při úpravách uživatelských programů se mění rovněž operandy instrukcí STA, LDA, SHLD, LHLD a LXI. Na jedné straně to znamená, že se upravují odkazy i na tabulky konstant a proměnných, avšak na druhé straně může být úprava nežádoucí. Představuje-li totiž operand proměnnou nebo konstantu a nikoliv adresu, může se stát, že náhodou (byť s malou pravděpodobností) jeho hodnota padne me-

zi hranice programu (P1) a (P2) a dojde tak ke změně.

Použijeme-li několik příkazů INS za sebou, nesmíme zapomenout, že každý tento příkaz odsouvá následnou část programu na vyšší adresy. Musíme s tím počítat při zadávání 1. parametru dalších příkazů INS. Nebezpečí omylu snadno obejdeme, budeme-li vkládat instrukce odzadu, od konce programu.

Chceme-li vložit posloupnost instrukcí před instrukci označenou návěštím a toto návěští přemístit na začátek vkládané posloupnosti, uplatníme kromě INS také příkaz CHG. Příkaz CHG s oběma parametry shodnými zjišťuje výskyt instrukcí s určitým operandem.

Příkaz SHF slouží nejen pro přesun tabulek konstant a proměnných, ale i pro složitější úpravy. K ilustraci slouží následující složitější příklad. Mějme v jedné oblasti paměti podprogram od adresy a1 do a2, který je vzájemně a oboustranně vázán s hlavním programem v druhé oblasti paměti na adresách b1 až b2. Podprogram chceme vložit do hlavního programu od adresy c. Provedeme následující sled příkazů: PRGM - BP: b1, EP: b2;
INOP c a2-a1+1; SHF a1 a2 c; REL a1 a2 c.

Při vkládání hranic programu zásadně uvádíme adresu posledního (nikoliv prvního) bytu poslední instrukce. Parametry příkazů, které představují adresy uvnitř upravovaného programu, musí mít skutečně hodnotu mezi (P1) a (P2), jinak může dojít k poškození TWS. V příkazu INS je třeba zadávat najednou právě tolik bytů, kolik má vkládaná instrukce.

Před zavedením uživatelských programů se doporučuje vynulovat příkazem CLEAR celou oblast paměti. Můžeme pak definovat program přes celou oblast, i když uživatelské podprogramy na sebe nenavazují, aniž bychom se museli obávat poškození programu, způsobeného tím, že vzhledem k náhodnému obsahu paměti mezi pro-

gramovými bloky přeskočil ukazatel začátek některého bloku a za kód instrukce považoval operand. Uvedené opatření rovněž snižuje nebezpečí přemazání paměti při zběhnutí uživatelského programu.

Při práci s textem si musíme uvědomit, že použití klávesy BS bezprostředně za několika shodnými znaky vymaže z paměti kód zhuštěného zápisu, takže při výpisu se znak vytiskne pouze jednou.

Relokaci TWS, např. při změnách kapacity a adresace paměti, je sice možno provést příkazy bloku "ladění", ale nepřímý asembler dle 4.2 vytvoří také výpis překladu s novými adresami. Je třeba vyměnit nejen všechny štítky ORG, ale i EQU s následujícími symboly: TWSSP, WORKADR, BUFFER, BUFEND, JMPTAB, ENDJMPTB, MESTABHI, STRING1, STRING2. Adresa WORKADR a adresa začátku tabulky MESTAB musí končit 00. Po zavedení relokovaného TWS musíme vložit obsah MESTAB, vynulovat oblast pro uložení JMPTAB a na její začátek vložit: 0E; nižší byte; vyšší byte adresy DEFINE. Na závěr příkazem DEFINE, který byl právě zaveden, definujeme podle tabulky symbolů výpisu překladu všechny příkazy daného bloku. V příl. 12 je výpis tabulek MESTAB a JMPTAB. V JMPTAB jsou definovány všechny příkazy TWS v pořadí dle 5.2, pouze příkazy DEFINE a CANCEL jsou uloženy jako první.

5.5 SEZNAM OBECNÝCH PODPROGRAMŮ

V této části jsou uvedeny nejdůležitější atributy těch podprogramů TWS, které jsou obecné a mohou být používány i v uživatelských programech. Podprogramy nerezidentní části TWS doporučujeme relokovat k uživatelskému programu, aby mohl být celý program uložen na děrné pásce nezávisle na TWS.

Za jménem podprogramu následuje startovací adresa. Je-li

v závorkách, je podprogram součástí nerezidentní části 1. varianty jádra TWS a podléhá tak případné relokaci. Následuje délka podprogramu v bytech. Počáteční adresa uložení podprogramu není obecně shodná se startovací adresou. Jestliže atribut není nadepsán, pak neexistuje. Podprogramy jsou uvedeny jen do 1. úrovně.

Funkce je popsána stručně, podrobněji viz 5.3.

READON 1D2D 16 B

Zapnutí a start snímače děrné pásky.

změna: A, F

BLANK 1D3D 9 B

Průchod blanků na děrné pásce.

vstup: max. počet blanků do B

výstup: 1. první platný znak v A, Z=0
2. (A)=0, Z=1 - překročen vložený počet blanků

změna: A, F, B podprog.: READBYTE

KEYINP 1D7A 18 B

Vstup znaku z klávesnice.

výstup: znak v A

změna: A, F

PRINT 1D8C 30 B

Tisk znaku psací jednotkou.

vstup: znak ASCII do A

READBYTE 1DAA 18 B

Vstup bytu z běžícího snímače děrné pásky.

výstup: byte v A, nastaveny příznaky

změna: A, F

RPEQU 1DBC 6 B

Porovnání obsahu registrových párů HL a DE.

vstup: 2 čísla do HL a DE

výstup: 1. Z=1, CY=0 při (HL) = (DE)

2. Z=0, CY=0 při (HL) > (DE)

3. Z=0, CY=1 při (HL) < (DE)

změna: A, F

ASCHEXA 1DC2 12 B

Převod ASCII-HEX/BIN 1 znaku a zařazení do čísla.

vstup: hexadecimální ASCII znak do A, číslo do HL

výstup: 1. číslo se zařazenou číslicí v HL, CY=0

2. číslo nezměněno, CY=1 - v A je nepovolený znak

změna: A, F, HL podprog.: ASCHEX1

ASCHEX1 1DCE 23 B

Převod ASCII-HEX/BIN 1 znaku.

vstup: hexadecimální ASCII znak do A

výstup: 1. převedený znak v nižším nibble A, S=1

2. znak v A nezměněn, S=0 - nepovolený znak

změna: A, F

PUNPAR 1DE5 35 B

Děrování znaku opatřeného paritou na děrovači pásky.

vstup: děrovaný znak do A

změna: A, F

PUNBYTE 1DEC 28 B

Děrování bytu děrovačem pásky.

vstup: děrovaný byte do A

HEXASCII 1E08 9 B

Výstup čtyřciferného hexadecimálního čísla v ASCII.

vstup: číslo do HL

změna: A, F podprog.: HEXASC2

HEXASC2 1E11 15 B

Výstup dvouciferného hexadecimálního čísla v ASCII.

vstup: číslo do A

změna: A, F podprog.: HEXASC1

HEXASC2+10 1E1B 5 B

Výstup hexadecimální číslice v ASCII.

vstup: číslice do nižšího nibble A

změna: A, F podprog.: HEXASC1

HEXASC1 1E20 12 B

Převod BIN/HEX-ASCII 4 bitů.

vstup: 4 bity do nižšího nibble A

výstup: hexadecimální ASCII číslice v A

změna: A, F

RPSUB 1E3C 7 B

Odečtení obsahu DE od obsahu HL.

vstup: 2 čísla do HL a DE

výstup: rozdíl v HL, 1. Z=1, CY=0 při (HL) = 0

2. Z=0, CY=0 při (HL) > 0

3. Z=0, CY=1 při (HL) < 0

změna: A, F, HL

RPCOMP 1E43 18 B

Porovnání obsahu HL vůči BC a DE.

vstup: 3 čísla do HL, DE a BC

výstup: 1. CY=0 při (HL) $\in$ $\langle$(BC),(DE)$\rangle$

2. CY=1 při (HL) vně intervalu

změna: A, F podprog.: RPSUB

RPCOMPDH 1E4D 8 B

Porovnání obsahu HL a DE.

vstup: 2 čísla do HL a DE

výstup: 1. CY=0 při (HL) ≥ (DE)

2. CY=1 při (HL) < (DE)

změna: A, F podprog.: RPSUB

BCTEST 1E55 6 B

Dekrementace BC a test obsahu na 0.

vstup: číslo do BC

výstup: 1. číslo zmenšené o 1 v BC, Z=0

2. (BC)=0, Z=1

změna: A, F, BC

TABLE1 1E5D 15 B

Vybrání položky z tabulky podle klíče.

vstup: klíč do E, adresa 1. klíče tabulky do HL

výstup: 1. nalezená položka v DE, adresa 2. bytu položky v HL, CY=1

2. adresa koncové značky v HL, CY=0 - klíč nenalezen

změna: A, F, DE, HL

BLANKOUT 1EA6 9 B

Děrování blanků na zapnutém děrovači.

vstup: počet blanků do B

změna: A, F, B podprog.: PUNBYTE

MOVEBLOK 1EAF 44 B

Přesun bloku dat v paměti.

vstup: počáteční adresa bloku do HL, koncová do DE, nová počáteční adresa do BC

výstup: nová koncová adresa v HL

změna: všechny registry podprog.: RPEQU, RPSUB, BCTEST

DELAY 1EF5 11 B

Zpoždění délky T /ms/.

vstup: $4.\sqrt{T}$ do A

změna: A, F podprog.: TIMER z monitoru

INPARAM (3D99) 11 B

Vstup hexadecimálního nebo dekadického čísla z psacího stroje.

výstup: 1. číslo v DE, nižší byte také v A, CY=0
2. CY=1 - stisknuto pouze ESC

změna: A, F, BC, DE podprog.: INADR1

DECCONV (3DB0) 29 B

Převod 1- až 5-ciferného dekadického čísla v ASCII na binární.

vstup: adresa číslice nejvyššího řádu snížená o 1 do BC, další číslice následují na vyšších adresách, zakončeno oddělovačem

výstup: 1. číslo v HL, adresa oddělovače v BC, v paměti převedeny číslice ASCII na BCD, CY=0
2. CY=1 - při nepovoleném znaku

změna: všechny registry podprog.: BCDBIN

BCDBIN (3DCD) 34 B

Převod 1- až 5-ciferného dekadického čísla v BCD na binární.

vstup: počet číslic do A, adresa číslice nejnižšího řádu zvětšená o 1 do BC, další číslice uloženy na nižších adresách

výstup: 1. číslo v HL, adresa číslice nejvyššího řádu v BC, CY=0
2. CY=1 - při nepovoleném znaku

změna: všechny registry

CLR (3E66) 10 B

Nulování bloku paměti.

vstup: počáteční adresa bloku do HL, koncová adresa do DE

změna: A, F, HL podprog.: RPEQU

PUNCHON (3F19) 12 B

Zapnutí děrovače.

podprog.: DELAY

TEXTOUT (3F25) 17 B

Výstup textu uloženého v paměti.

vstup: adresa 1. znaku textového řetězce do HL

výstup: adresa koncového znaku ESC zvětšená o 1 v HL

změna: A, F, HL podprog.: PUTTEXT

BCDOUT+9 (3F69) 27 B

Výstup 1- až 5-ciferného dekadického čísla v ASCII.

vstup: číslo do HL

změna: A, F, BC, DE podprog.: BINCON

BINBCD (3F91) 43 B

Převod binárního čísla na dekadické.

vstup: číslo do HL, do DE adresa, na kterou uložit číslici nejvyššího řádu

výstup: v DE adresa číslice nejnižšího řádu, v paměti 5 číslic BCD uloženo vzestupně

změna: všechny registry podprog.: DECNO

6 ZÁVĚR

Realizovaný přídavný monitor TWS splňuje požadavky kladené na jednoduché programové vybavení, umožňující ovládat z psacího stroje základní funkce malého MVS při ladění uživatelských programů. Při obsazení 1,7 KB RAM ponechává 1,75 KB pro uživatelský program. Délka celého TWS včetně EPROM činí 2,45 KB.

Základní funkce TWS (jádro) se osvědčily během ladění ostatních částí, rovněž funkce bloku "ladění" byly používány ihned po svém odzkoušení. Druhá varianta jádra byla získána relokací první varianty rovněž prostřednictvím bloku "ladění". Úpravy uživatelských programů by ještě více usnadnil disasembler. Z časových důvodů však již naprogramovaný disasembler nemohl být přeložen a odladěn.

Software vytvořený pro práci s textem se uplatňuje v programech, které zakládají na děrné pásce adresář a umožňují pak podle čísla vyhledat a na obálku napsat adresu.

Mnoho podprogramů TWS nalezne uplatnění v uživatelských programech, zejména při převodu čísel a při výstupu na psací jednotku nebo děrovač pásky, např. při sběru dat ze snímačů instalovaných na zkoumaném zařízení. Po rozšíření kapacity paměti MVS se počítá se zavedením textového editoru a asembleru. I tyto programy budou moci vycházet z podprogramů TWS, nejen z obecných, ale i z několika specielních, v jejichž funkci na to bylo pamatováno. Koncepce TWS umožňuje jeho další rozvoj, neboť lze zavádět nové příkazy. Řídícím programem TWS bude možno ovládat i textový editor a asembler.

Jistě by bylo účelné realizovat řadu dalších funkcí, např.:

1. Kopírování děrné pásky s libovolným obsahem.
2. Test paměti a test připojení periferií.

3. Trasování programu, tj. automatický výpis obsahu registrů po každé instrukci (bohužel velmi zdlouhavé).
4. Počítání strojových cyklů resp. času při chodu programu.
5. Řazení slov podle abecedy.

Zhodnotit ekonomický přínos budování MVS je obtížné, neboť používání prostředků pro vývoj softwaru je při návrhu mikropočítačových systémů zcela nezbytné. Je totiž téměř vyloučeno, aby se vývojovému pracovníkovi napoprvé podařilo sestavit program, který by po překladu a uložení do permanentní paměti bezchybně zabezpečil požadovanou funkci aplikačního mikropočítače.

Je-li třeba odlaďovat software v reálném čase a ve spolupráci se řízeným objektem nebo jeho modelem, bez MVS se neobejdeme. V ostatních případech je alternativou k MVS nepřímý software na velkém počítači, zejména asembler a simulátor mikroprocesoru. Na velkém počítači lze k činnosti nepřímého softwaru používat operační systém, bohaté periferní vybavení a velkokapacitní vnější paměti. Bohužel práce na velkém počítači je málo operativní. I když pomineme vzdálenost mezi mikropočítačovým pracovištěm a výpočetním střediskem, způsob zpracování "přes přepážku" téměř vylučuje odladění programu na simulátoru v únosně dlouhé době. Budeme-li uvažovat zatím málo rozšířená výpočetní střediska vybavená terminály a umožňující konverzační způsob zpracování, zjistíme ekonomickou neefektivnost tohoto řešení. Vyčerpáme-li denně půl hodiny strojního času, což není tak mnoho vzhledem k nutnosti mnohokrát spustit asembler a simulátor, zaplatíme za 1 rok tolik, kolik by stál např. MVS 800 (asi 300 tis. Kčs; 1 h strojního času moderního počítače asi 3 tis. Kčs).

Jinou alternativou k MVS je nepřímý software na stolním minipočítači (rozšířené typy firmy Hewlett-Packard). Ekonomicky vyznívá příznivěji než MVS, neboť stolní minipočítač má cenu

srovnatelnou s MVS, avšak provádí také vědeckotechnické výpočty. Problém může činit menší kapacita paměti, neumožňující současné zavedení textového editoru, asembleru a simulátoru, a až o 3 řády nižší rychlost simulace proti reálnému času v důsledku použití interpretačního jazyka BASIC.

Avšak ani tato alternativa neumožňuje připojení řízeného objektu v reálném čase, a tak nezbývá než podporovat budování MVS, které umožní, budou-li efektivně využívány schopnými pracovníky, navrhovat a připravovat aplikace mikroprocesorů a mikropočítačů ve všech odvětvích národního hospodářství.

Na závěr bych chtěl poděkovat
vedoucímu diplomové práce ing. Václavu Sedlickému za všestrannou pomoc, cenné rady a důvěru, kterou projevil navrženým hardwarovým úpravám a softwarové koncepci;
jeho spolupracovníkovi ing. Petru Vlkovi za užitečné připomínky k funkcím TWS a za pomoc při uvádění psacího stroje do provozu včetně mechanické stavby zdroje;
konzultantovi ing. Josefu Grosmanovi zejména za úpravu a pomoc při užívání nepřímého asembleru na počítači RPP-16S;
programátorovi Václavu Hanouskovi za rady k programování v jazyce PL/1 a za realizaci děrování pásky na počítači EC-1033.

LITERATURA

/1/ NIPPON ELECTRIC CO., Japonsko: TK-80 User´s Manual. 1977.

/2/ NIPPON ELECTRIC CO., Japonsko: The μCOM-8 software manual. 1975.

/3/ ŠEFCOVÁ,H. - DUŠEK,M.: Křížový asembler. /Soutěžní práce SVOČ./ Liberec, VŠST, 1980.

/4/ JANÍK,M.: Popis diskového systému. ÚVT TESLA Žilina, B.r.

/5/ BERKA,Z. - ŽILKA,Z.: Nevlastní asembler ASI80A pro mikroprocesory typu 8080. /Výzkumná zpráva./ Praha, ÚRE ČSAV, 1979.

/6/ Mikroprocesorový systém 8080. ZP ČSVTS TESLA Piešťany,1978.

/7/ ZBROJOVKA Brno n.p.: CONSUL 256. Technický popis. 1978.

/8/ ZBROJOVKA Brno n.p.: CONSUL 256-ISO. Návod k obsluze. B.r.

/9/ ZPA Košíře n.p.: Fotoelektrický snímač děrné pásky FS 1503. Návod pro montáž, obsluhu a údržbu. 1976.

/10/ VEB ROBOTRON-ELEKTRONIK, Zella-Mehlis: Steuerelektronik 1215-1111/12/13. Technische Dokumentation. 1980.

/11/ VEB ROBOTRON-ELEKTRONIK, Zella-Mehlis: Lochbandstanzer und Auf- und Abspulgerät 1215/1227. Techn. Dokumentation. 1978.

/12/ TESLA Rožnov n.p.: Polovodičové součástky. 1978.

/13/ SEDLICKÝ,V.: Studium možností použití mikroprocesoru pro řízení textilních strojů. /Dílčí zpráva fakultního výzkumného úkolu./ Liberec, VŠST, 1979.

/14/ Elektronik Sonderheft II: Mikroprozessoren software. 1978.

/15/ Automatizace (přílohy), 1979 - 1980.

/16/ SOBOTKA,Z., Sdělovací technika, 1979, č. 5, s. 177.

/17/ OLEHLA,M. et al.: Programování, programovací jazyky a operační systémy. /Skriptum./ Liberec, 1980.

SEZNAM PŘÍLOH

vyšší nibble / nižší	0	1	2	3	4	5	6	7
0	NUL		␣	0	@	DLE		
1		+	!	1	SOH	Q		
2		−	"	2	STX	R		
3			#	3	ETX	DC3		
4			$	4	EOT	STOP		
5			%	5	ENQ	NAK		
6			&	6	ACK	SYN		
7			'	7	BEL	ETB		
8	←		(	8	H	CAN		
9	→		)	9	I	EM		
A	↓		*	:	J	SUB		
B		[ESC	+	;	VT K	[{		
C		< FSR	<	,	FF L	' '		
D	↵	= GSR	=	-	M	] }		
E		> RSR	>	.	SO N	ˇ ˇ		
F		? USR	?	/	SI O	— 0		DEL

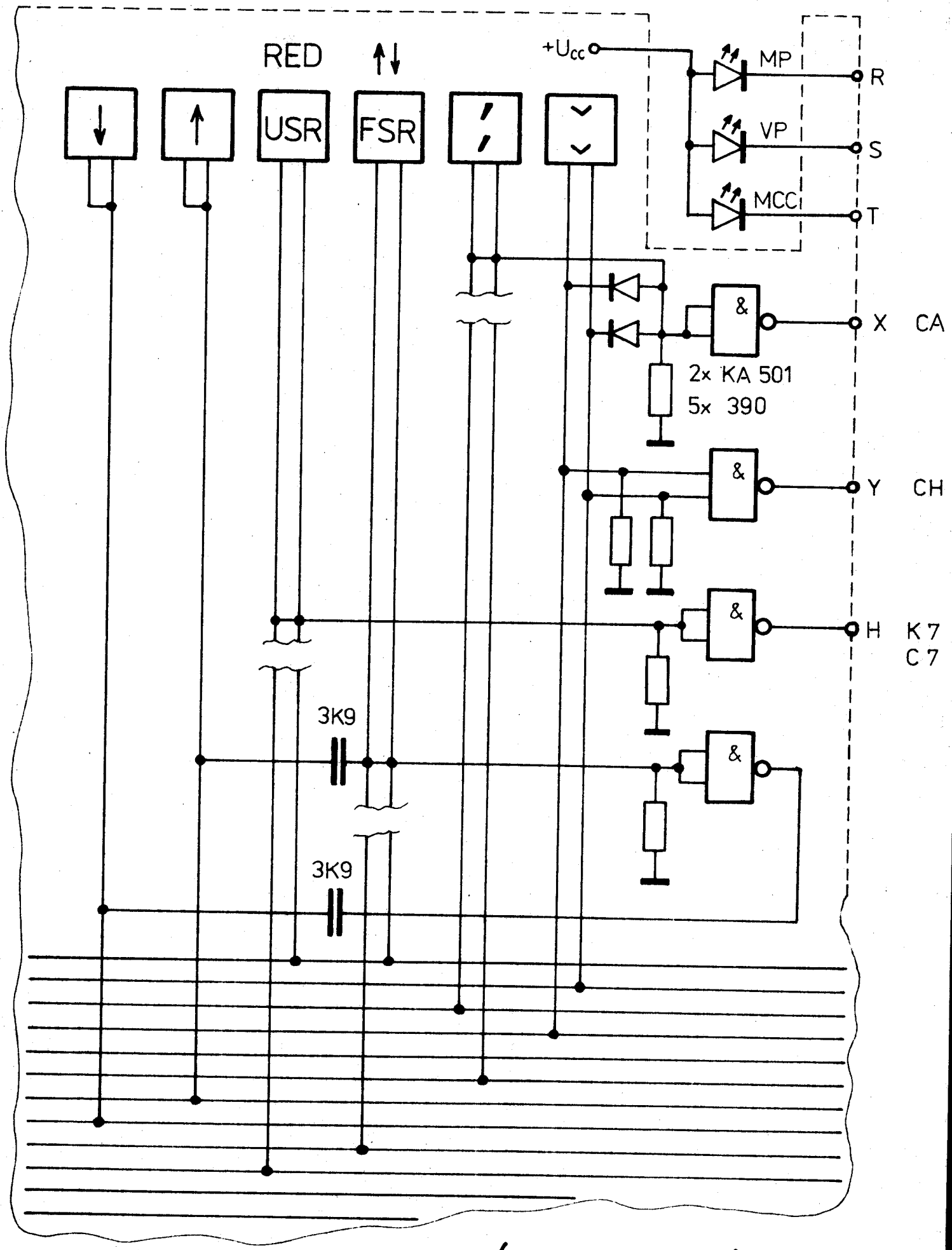
příl. 1

KÓD KLÁVESNICE

	0 8	1 9	2 A	3 B	4 C	5 D	6 E	7 F
0			␣ SP	0		P		p
1			!	1	A	Q	a	q
2			"	2	B	R	b	r
3			#	3	C	S	c	s
4			Kčs	4	D	T	d	t
5			%	5	E	U	e	u
6			&	6	F	V	f	v
7			'	7	G	W	g	w
8	← BS		(	8	H	X	h	x
9	→ TAB		)	9	I	Y	i	y
A	↓ LF		*	:	J	Z	j	z
B			+	;	K	[	k	{
C	↤ CR		,	<	L	\	l	@
D	↵ NL		-	=	M	]	m	}
E	— VÝMAZ		.	>	N	ˇ	n	´
F	+ SAZEČ		/	?	O	_	o	

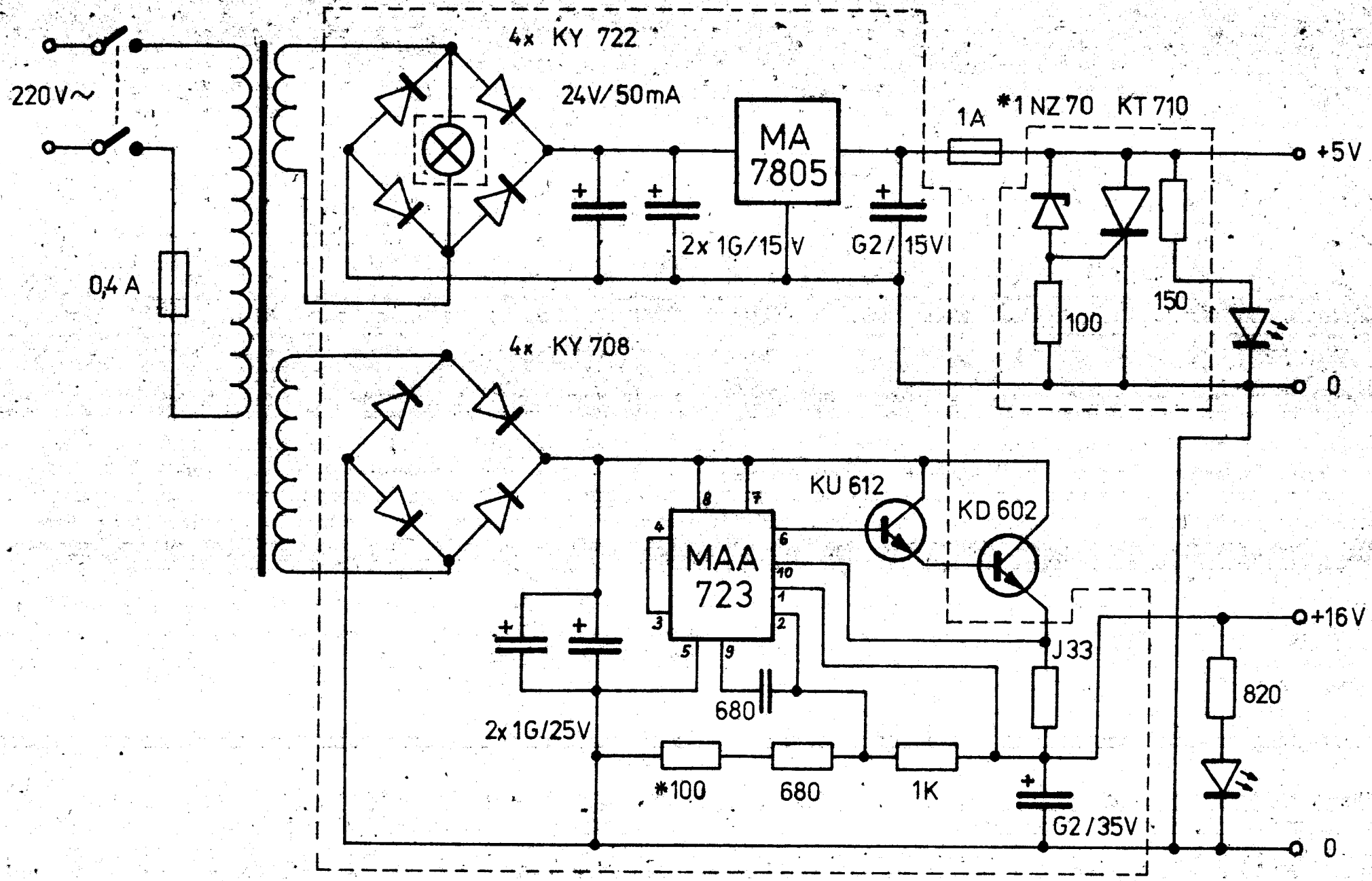
přil. 2

KÓD PSACÍ JEDNOTKY



VYVEDENÍ POMOCNÝCH SIGNÁLŮ Z KLÁVESNICE

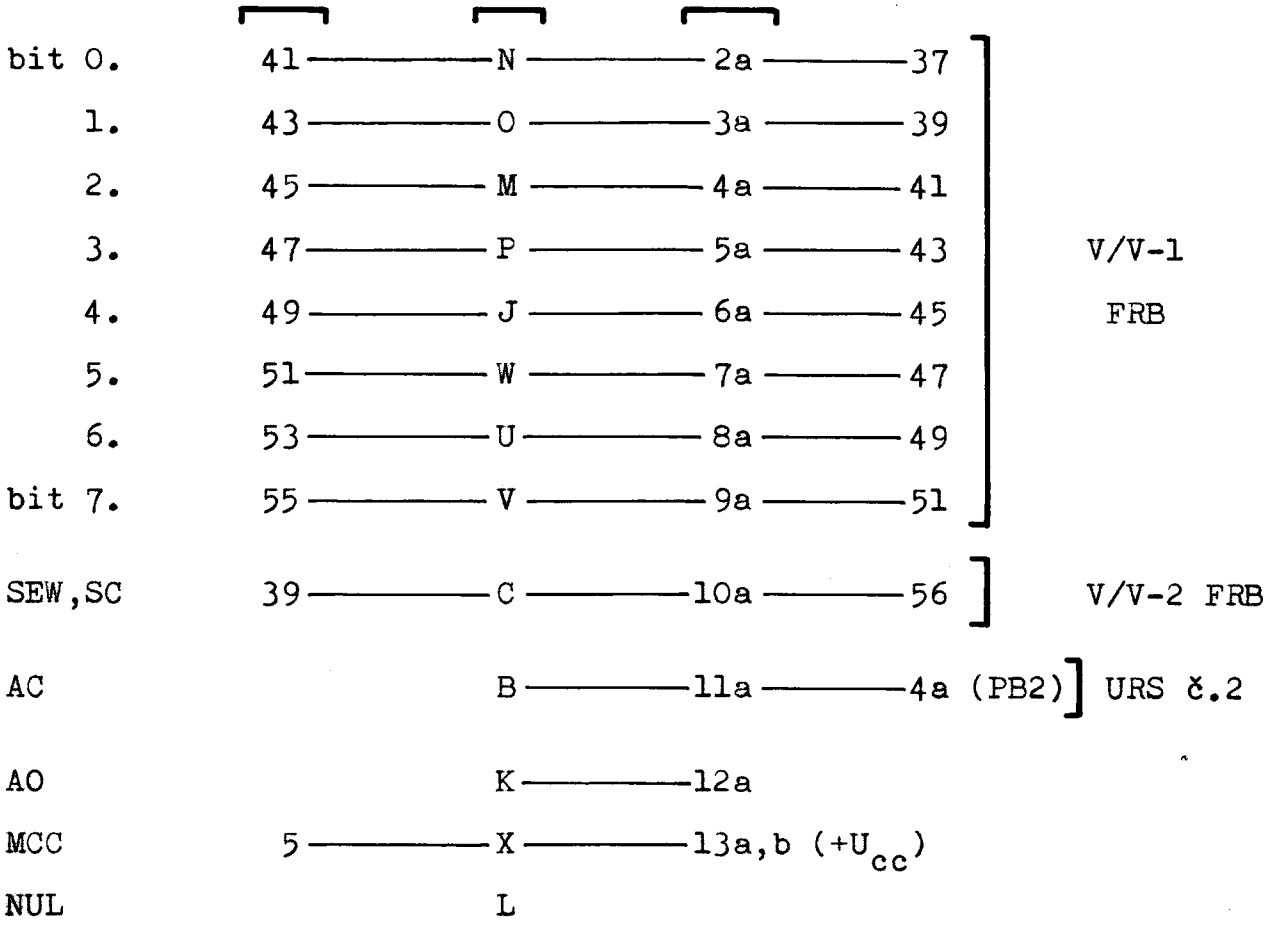
příl.4

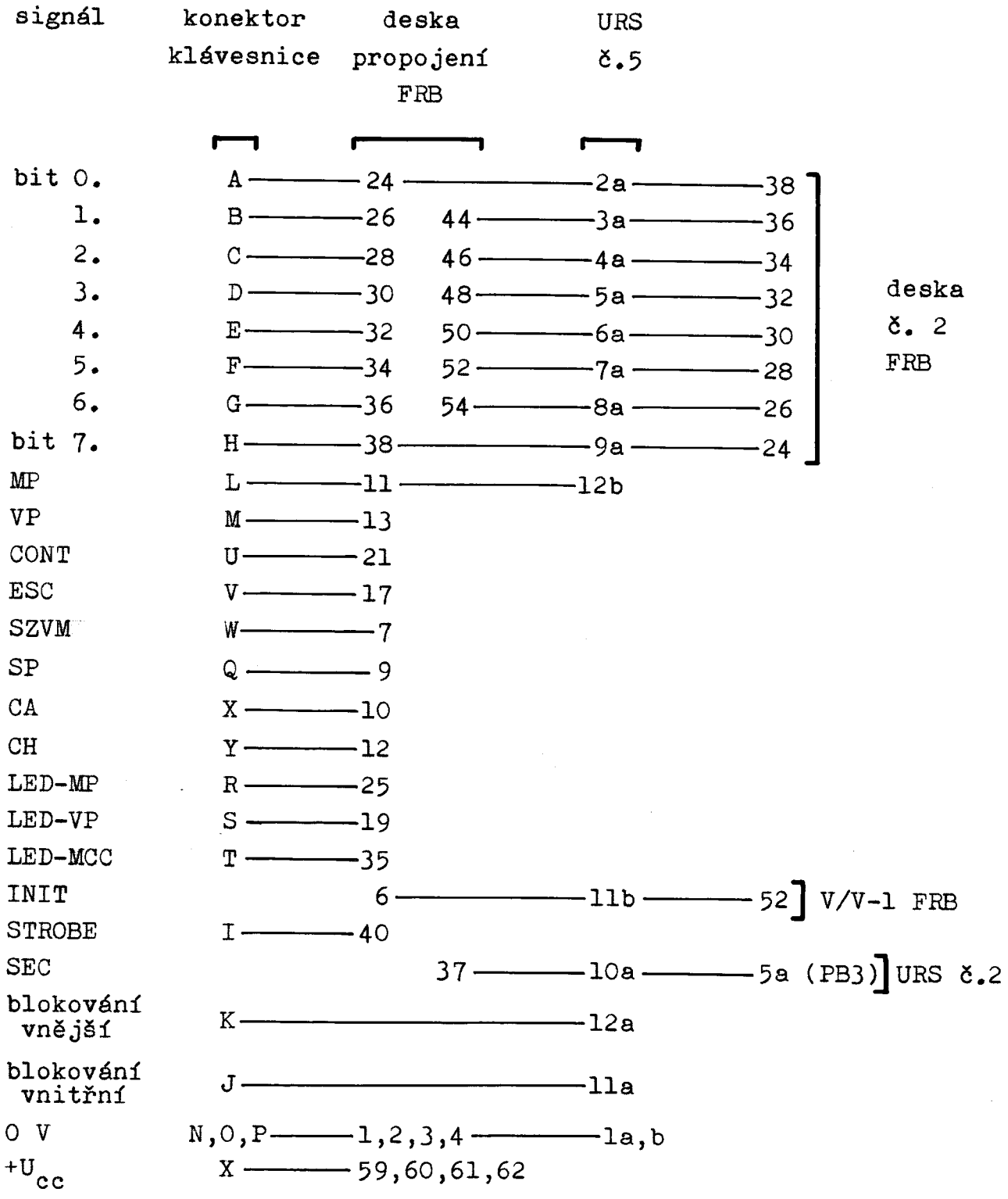


příl. 5

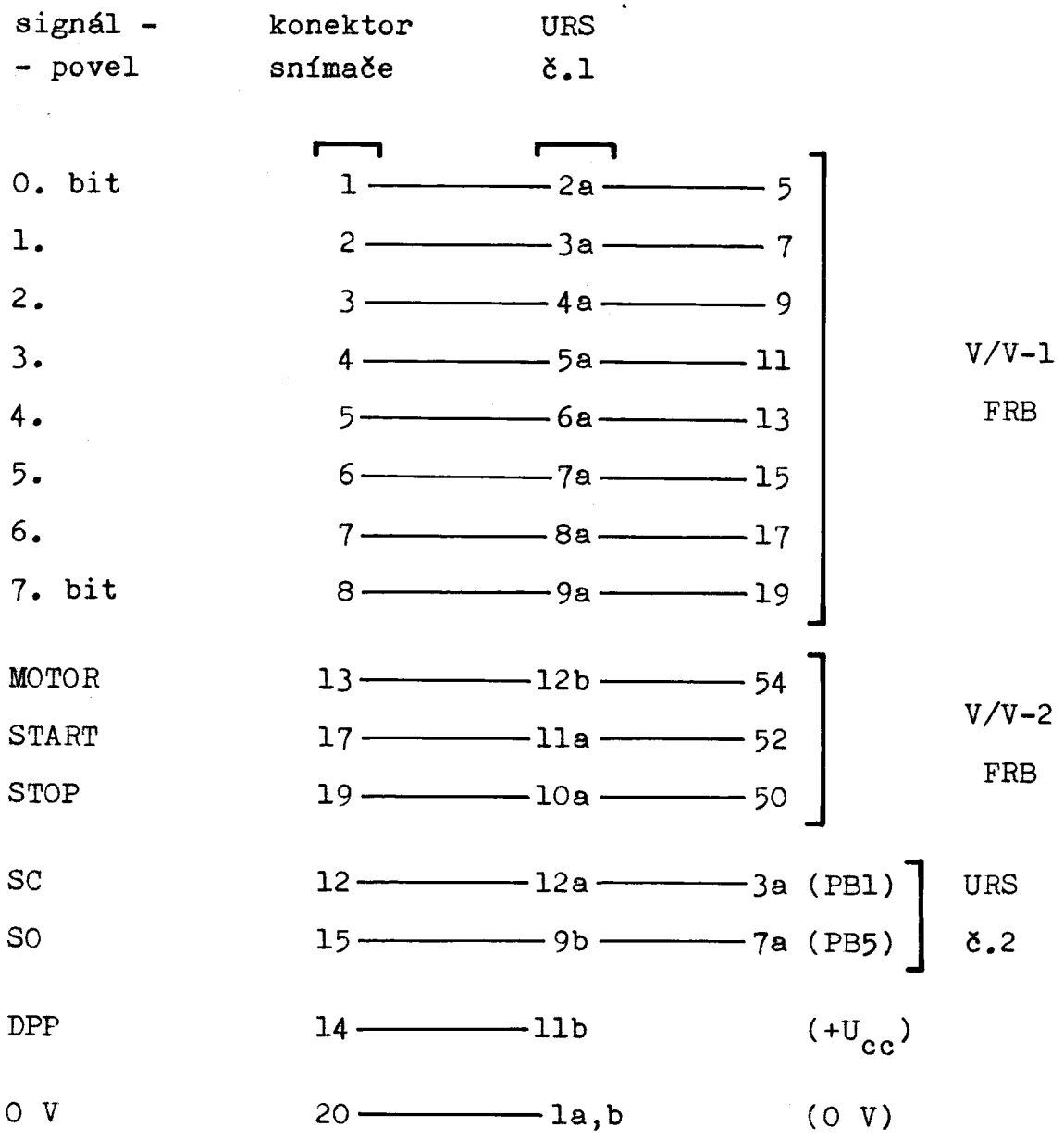
NAPÁJECÍ ZDROJ PRO PSACÍ STROJ

Příl. 6 PROPOJENÍ PSACÍHO STROJE S MIKROPOČÍTAČEM

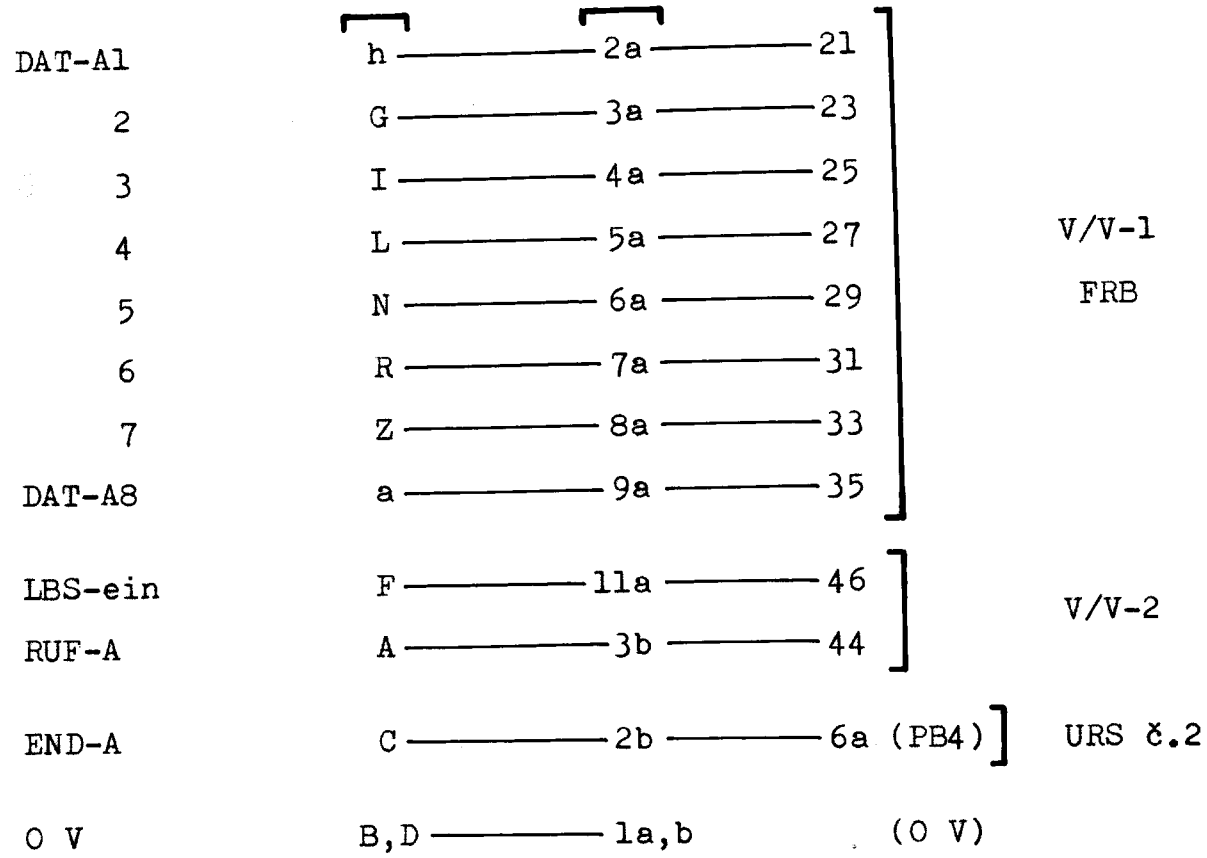



signál
konektor klávesnice
deska propojení FRB
URS č.5
bit 0. A 24 2a 38
1. B 26 44 3a 36
2. C 28 46 4a 34
3. D 30 48 5a 32
4. E 32 50 6a 30
5. F 34 52 7a 28
6. G 36 54 8a 26
bit 7. H 38 9a 24
deska č. 2 FRB
MP L 11 12b
VP M 13
CONT U 21
ESC V 17
SZVM W 7
SP Q 9
CA X 10
CH Y 12
LED-MP R 25
LED-VP S 19
LED-MCC T 35
INIT 6 11b 52 V/V-1 FRB
STROBE I 40
SEC 37 10a 5a (PB3) URS č.2
blokování vnější K 12a
blokování vnitřní J 11a
0 V N,O,P 1,2,3,4 1a,b
+Ucc X 59,60,61,62

Příl. 7 PROPOJENÍ SNÍMAČE DĚRNÉ PÁSKY S MIKROPOČÍTAČEM



Příl. 8 PROPOJENÍ DĚROVAČE DĚRNÉ PÁSKY S MIKROPOČÍTAČEM



Příl. 10 VÝPIS ČINNOSTI TWS

```
tws  :"Uložení a výpis textu"

KEY 0C00

TO BE OR NOT TO BE THAT IS THE QUESTION
---------------------------------------
Vysoká škola strojní a textilní v Liberci
 =0C5A

tws  :PRINT 0C00 0C50

TO BE OR NOT TO BE THAT IS THE QUESTION
---------------------------------------
Vysoká škola strojní a textilní

tws  :"Ukázka hlášení chyb v příkazech"
      CLAEP 0BB0 0BFF
syntax
tws  :CLEAR 0BB0
syntax
tws  :CLEAR 0BG0 0BFF
illegal
tws  :CLEAR 0BFF 0BB0
addr
tws  :CLEAR 0BB0 0BFF

tws  :
"Přesun podprogramu MOVEBLOK z EPROM (viz příl. 11) do RAM"

REL 1EAF 1EDA 0CAF =0CDA

tws  :
"Do uvedeného programu vsuneme instrukce, které provedou
 tisk "Z" při přesunu bloku od začátku a "K" při přesunu
 od konce. RET změníme na RST 7."

C 0CDA
0CDA: C9 :FF
0CDB: 43 :

tws  :INS 0CCE
0CCE :3E 4B
0CD0 :DF
0CD1 :

tws  :INS 0CC0
0CC0 :3E 5A
0CC2 :DF
0CC3 :

tws  :
```

```
tws  :"Jaké jsou nové hranice upraveného programu ?"

PRGM
BP: 0CAF :
EP: 0CE0 :

tws  :
"Program vypíšeme. Porovnáním s příl. 11 lze vysledovat
změnu adres skoků provedenou příkazy REL a INS."

LIST 0CAF 0CE0

addr   0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F

0CA0:                                                 D5
0CB0:  EB CD 3C 1E E5 C5 E1 C1 CD BC 1D D2 CE 0C F1 03
0CC0:  3E 5A DF 1A 77 CD 55 1E C8 13 23 C3 C3 0C D1 09
0CD0:  03 3E 4B DF E5 1A 77 CD 55 1E 1B 2B C2 D5 0C E1
0CE0:  FF

tws  :
"Nastavíme počáteční stav registrů. Chceme upraveným
 programem přesunout text z adresy 0C00 na 0C40."

RC
AF: 0046 :
BC: 00E9 :0C40
DE: 0000 :0C5A
HL: 0855 :0C00
SP: 83C7 :
PC: 0D33 :

tws  :"Start programu"
R 0CAF
K

tws  :
"Přesun byl proveden od konce. Definujme pro spouštění
 programu příkazové slovo."

DEFINE 0CAF
 :BLOK-Z/K

tws  :"Zjištění výskytu návěští BCTEST v programu."

CHG 1E55 1E55
0CC5
0CD7

tws  :
"Počítání a převod čísel."
CAL
 :30 +50 -#32 =0060 =#96
 :800 =0800 =#2048
 :#1000 =03E8 =#1000
 :#67000
range
tws  :
```

```
LINE  ADDR  CODE            SYMBOL    MNEM   OPERAND        COMMENT                              IDENTIF.
-------------------------------------------------------------------------------------------------------
  1.                                  TITLE  I,DOLEZAL! PRIDAVNY MONITOR 'TWS' PRO TK=80

     ;=================================================================================
            BLOK 'JADRO 1, CAST' - ULOZENO V EPROM
     ;=================================================================================

  3.                                  ORG    1D00H

     ;=================================================================================
            ZAVADENI PROGRAMU Z DERNE PASKY BEZ TWS, S INDIKACI NA DISPLEJI
     ;=================================================================================

  5.  1D00  06 FF           MONREAD   MVI    B,0FFH
  6.  1D02  CD 19 1D                  CALL   READBLOK
  7.  1D05  C2 CB 02                  JNZ    ERROR
  8.  1D08  CA 08 00                  JC     MONITOR
  9.  1D0B  22 EC 83                  SHLD   DATA
 10.  1D0E  CD A1 01                  CALL   RGDSP
 11.  1D11  CD 16 02                  CALL   KEYIN
 12.  1D14  06 17                     MVI    B,23
 13.  1D16  C3 02 1D                  JMP    MONREAD+2

     ;=================================================================================
            SOUBOR PODPROGRAMU PRO ZAVADENI PROGRAMU Z DERNE PASKY
     ;=================================================================================

 15.                                         SNIMANI 1 BLOKU A RIZENI SNIMACE
 16.  1D19  CD 2D 1D        READBLOK  CALL   READON
 17.  1D1C  CD 3D 1D                  CALL   BLANK
 18.  1D1F  37                        STC
 19.  1D20  C4 46 1D                  CNZ    RBLOCK
 20.  1D23  3E 2F                     MVI    A,2FH
 21.  1D25  D3 F4                     OUT    CONTPORT
 22.  1D27  C0                        RNC
```

```
ADDR  CODE          SYMBOL     MNEM   OPERAND         COMMENT            IDENTIF.
----------------------------------------------------------------------------------
1D28  3E 0F                    MVI    A,0FH
1D2A  D3 F4                    OUT    CONTPORT
1D2C  C9                       RET
                                      ZAPNUTI A START SNIMACE
1D2D  3E 2F         READON     MVI    A,2FH
1D2F  D3 F4                    OUT    CONTPORT
1D31  DB F9         RN         IN     PORTB
1D33  E6 20                    ANI    20H
1D35  CA 31 1D                 JZ     RN
1D38  3E 37                    MVI    A,37H
1D3A  D3 F4                    OUT    CONTPORT
1D3C  C9                       RET
                                      PRUCHOD BLANKU
1D3D  CD AA 1D      BLANK      CALL   READBYTE
1D40  C0                       RNZ
1D41  05                       DCR    B
1D42  C2 3D 1D                 JNZ    BLANK
1D45  C9                       RET
                                      VLASTNI SNIMANI 1 BLOKU
1D46  06 83         RBLOCK     MVI    B,83H
1D48  CD 6D 1D                 CALL   READWORD+3
1D4B  D8                       RC
1D4C  22 EE 83                 SHLD   ADDR
1D4F  EB                       XCHG
1D50  06 83                    MVI    B,83H
1D52  CD 6A 1D                 CALL   READWORD
1D55  D8                       RC
1D56  EB                       XCHG
1D57  E5            LOOPBYTE   PUSH   H
1D58  06 81                    MVI    B,81H
1D5A  CD 6A 1D                 CALL   READWORD
1D5D  7D                       MOV    A,L
1D5E  E1                       POP    H
1D5F  D8                       RC
1D60  77                       MOV    M,A
1D61  CD BC 1D                 CALL   RPEQU
1D64  23                       INX    H
1D65  C2 57 1D                 JNZ    LOOPBYTE
```

```
LINE  ADDR  CODE              SYMBOL    MNEM   OPERAND          COMMENT          IDENTIF.
-----------------------------------------------------------------------------------------
 61.  1D68  2B                          DCX    H
 62.  1D69  C9                          RET
 63.                                           SNIMANI A PREVOD ASCII/BIN
 64.  1D6A  CD AA 1D          READWORD  CALL   READBYTE
 65.  1D6D  37                          STC
 66.  1D6E  E0                          RPO
 67.  1D6F  E6 7F                       ANI    7FH
 68.  1D71  CD C2 1D                    CALL   ASCHEXA
 69.  1D74  05                          DCR    B
 70.  1D75  D8                          RC
 71.  1D76  FA 6A 1D                    JM     READWORD
 72.  1D79  C9                          RET

     ;===================================================================
          VSTUP ZNAKU Z KLAVESNICE PSACIHO STROJE
     ;===================================================================

 74.  1D7A  DB F9             KEYINP    IN     PORTB
 75.  1D7C  E6 08                       ANI    08H
 76.  1D7E  CA 7A 1D                    JZ     KEYINP
 77.  1D81  DB F9             KN        IN     PORTB
 78.  1D83  E6 08                       ANI    08H
 79.  1D85  C2 81 1D                    JNZ    KN
 80.  1D88  DB F6                       IN     KEYBRD
 81.  1D8A  2F                          CMA
 82.  1D8B  C9                          RET

     ;===================================================================
          TISK ZNAKU PSACIM STROJEM
     ;===================================================================

 84.  1D8C  F5                PRINT     PUSH   PSW
 85.  1D8D  17                          RAL
 86.  1D8E  3F                          CMC
 87.  1D8F  1F                          RAR
 88.  1D90  D3 F6                       OUT    TWRITER
```

```
 LINE  ADDR  CODE                  SYMBOL    MNEM    OPERAND           COMMENT                        IDENTIF.
-----------------------------------------------------------------------------------------------------------------
  89.  1D92  DB F9                 PT1       IN      PORTB
  90.  1D94  E6 04                           ANI     04H
  91.  1D96  CA 92 1D                        JZ      PT1
  92.  1D99  3E 4F                           MVI     A,4FH
  93.  1D9B  D3 F4                           OUT     CONTPORT
  94.  1D9D  3E 0F                           MVI     A,0FH
  95.  1D9F  D3 F4                           OUT     CONTPORT
  96.  1DA1  DB F9                 PT2       IN      PORTB
  97.  1DA3  E6 04                           ANI     04H
  98.  1DA5  C2 A1 1D                        JNZ     PT2
  99.  1DA8  F1                              POP     PSW
 100.  1DA9  C9                              RET

     ================================================================================================
             CTENI ZNAKU ZE SNIMACE DERNE PASKY
     ================================================================================================

 102.  1DAA  DB F9                 READBYTE  IN      PORTB
 103.  1DAC  E6 02                           ANI     02H
 104.  1DAE  C2 AA 1D                        JNZ     READBYTE
 105.  1DB1  DB F9                 RE        IN      PORTB
 106.  1DB3  E6 02                           ANI     02H
 107.  1DB5  CA B1 1D                        JZ      RE
 108.  1DB8  DB F7                           IN      READER
 109.  1DBA  B7                              ORA     A
 110.  1DBB  C9                              RET

     ================================================================================================
             SROVNANI OBSAHU REGISTRU DE A HL
     ================================================================================================

 112.  1DBC  7C                    RPEQU     MOV     A,H
 113.  1DBD  BA                              CMP     D
 114.  1DBE  C0                              RNZ
 115.  1DBF  7D                              MOV     A,L
 116.  1DC0  BB                              CMP     E
```

```
 LINE  ADDR  CODE                  SYMBOL     MNEM     OPERAND          COMMENT                                IDENTIF.
-------------------------------------------------------------------------------------------------------------------------
 117.  1DC1  C9                               RET

     ====================================================================================================
               PREVOD HEXADECIMALNIHO CISLA Z ASCII ZNAKU DO BINARNIHO TVARU
     ====================================================================================================

 119.                                                             PREVOD 1 ZNAKU Z AKUMULATORU DO REG. HL
 120.  1DC2  CD CE 1D              ASCHEXA    CALL     ASCHEX1
 121.  1DC5  37                               STC
 122.  1DC6  F0                               RP
 123.  1DC7  29                               DAD      H
 124.  1DC8  29                               DAD      H
 125.  1DC9  29                               DAD      H
 126.  1DCA  29                               DAD      H
 127.  1DCB  B5                               ORA      L
 128.  1DCC  6F                               MOV      L,A
 129.  1DCD  C9                               RET
 130.                                                             PREVOD 1 ZNAKU V AKUMULATORU
 131.  1DCE  D6 30                 ASCHEX1    SUI      30H
 132.  1DD0  FA E2 1D                         JM       ADD30
 133.  1DD3  FE 0A                            CPI      0AH
 134.  1DD5  F8                               RM
 135.  1DD6  D6 07                            SUI      7
 136.  1DD8  FE 0A                            CPI      0AH
 137.  1DDA  FA E0 1D                         JM       ADD7
 138.  1DDD  FE 10                            CPI      10H
 139.  1DDF  F8                               RM
 140.  1DE0  C6 07                 ADD7       ADI      7
 141.  1DE2  C6 30                 ADD30      ADI      30H
 142.  1DE4  C9                               RET

     ====================================================================================================
               DEROVANI ZNAKU DEROVACEM PASKY
     ====================================================================================================

 144.  1DE5  E6 7F                 PUNPAR     ANI      7FH
```

```
LINE  ADDR  CODE         SYMBOL    MNEM   OPERAND        COMMENT                IDENTIF.
-----------------------------------------------------------------------------------------
145.  1DE7  EA EC 1D               JPE    PUNBYTE
146.  1DEA  F6 80                  ORI    80H
147.  1DEC  F5           PUNBYTE   PUSH   PSW
148.  1DED  2F                     CMA
149.  1DEE  D3 F7                  OUT    PUNCHER
150.  1DF0  DB F9        PE1       IN     PORTB
151.  1DF2  E6 10                  ANI    10H
152.  1DF4  CA F0 1D               JZ     PE1
153.  1DF7  3E 8E                  MVI    A,8EH
154.  1DF9  D3 F4                  OUT    CONTPORT
155.  1DFB  DB F9        PE2       IN     PORTB
156.  1DFD  E6 10                  ANI    10H
157.  1DFF  C2 FB 1D               JNZ    PE2
158.  1E02  3E 8F                  MVI    A,8FH
159.  1E04  D3 F4                  OUT    CONTPORT
160.  1E06  F1                     POP    PSW
161.  1E07  C9                     RET

     ==========================================================================
          PREVOD BINARNIHO CISLA NA HEXADECIMALNI V ASCII ZNACICH A VYSTUP NA PERIFERII
     ==========================================================================

163.                                      2 BYTE = 4 ZNAKY
164.  1E08  7C           HEXASCII  MOV    A,H
165.  1E09  CD 11 1E               CALL   HEXASC2
166.  1E0C  7D                     MOV    A,L
167.  1E0D  CD 11 1E               CALL   HEXASC2
168.  1E10  C9                     RET
169.                                      1 BYTE = 2 ZNAKY
170.  1E11  F5           HEXASC2   PUSH   PSW
171.  1E12  0F                     RRC
172.  1E13  0F                     RRC
173.  1E14  0F                     RRC
174.  1E15  0F                     RRC
175.  1E16  CD 20 1E               CALL   HEXASC1
176.  1E19  CF                     RST    3
177.  1E1A  F1                     POP    PSW
```

```
 LINE  ADDR  CODE            SYMBOL    MNEM    OPERAND          COMMENT                    IDENTIF.
 178.  1E1B  CD 20 1E                  CALL    HEXASC1
 179.  1E1E  DF                        RST     3
 180.  1E1F  C9                        RET
 181.                                  PREVOD 1 NIBBLE = 1 ZNAK
 182.  1E20  E6 0F           HEXASC1  ANI     0FH
 183.  1E22  FE 0A                     CPI     0AH
 184.  1E24  FA 29 1E                  JM      CNO
 185.  1E27  C6 07                     ADI     7
 186.  1E29  C6 30           CNO      ADI     30H
 187.  1E2B  C9                        RET

;=================================================================================
          PREVOD ASCII/BIN Z PRACOVNIHO POLE PAMETI DO REGISTRU HL
;=================================================================================

 189.  1E2C  21 00 00        HEXACONV LXI     H,0
 190.  1E2F  0A                        LDAX    B
 191.  1E30  FE 21                     CPI     21H
 192.  1E32  3F                        CMC
 193.  1E33  D0                        RNC
 194.  1E34  CD C2 1D                  CALL    ASCHEXA
 195.  1E37  D8                        RC
 196.  1E38  03                        INX     B
 197.  1E39  C3 2F 1E                  JMP     HEXACONV+3

;=================================================================================
          VYPOCET  (HL) - (DE) --> (HL)
;=================================================================================

 199.  1E3C  7D              RPSUB    MOV     A,L
 200.  1E3D  93                       SUB     E
 201.  1E3E  6F                       MOV     L,A
 202.  1E3F  7C                       MOV     A,H
 203.  1E40  9A                       SBB     D
 204.  1E41  67                       MOV     H,A
 205.  1E42  C9                       RET
```

```
 LINE  ADDR  CODE            SYMBOL    MNEM    OPERAND        COMMENT                                  IDENTIF.
-------------------------------------------------------------------------------------------------------------
      ;=============================================================================
                POROVNANI OBSAHU REGISTRU: JE (HL) V INTERVALU [(BC),(DE)] ?
      ;=============================================================================

  207.  1E43  E5              RPCOMP    PUSH    H
  208.  1E44  D5                        PUSH    D
  209.  1E45  50                        MOV     D,B
  210.  1E46  59                        MOV     E,C
  211.  1E47  CD 3C 1E                  CALL    RPSUB
  212.  1E4A  D1                        POP     D
  213.  1E4B  E1                        POP     H
  214.  1E4C  C8                        RC
  215.  1E4D  D5              RPCOMPDH  PUSH    D
  216.  1E4E  EB                        XCHG
  217.  1E4F  CD 3C 1E                  CALL    RPSUB
  218.  1E52  EB                        XCHG
  219.  1E53  D1                        POP     D
  220.  1E54  C9                        RET
  221.                                  DEKREMENTACE REG, BC A TEST NA 0
  222.  1E55  0B              BCTEST    DCX     B
  223.  1E56  79                        MOV     A,C
  224.  1E57  B7                        ORA     A
  225.  1E58  C0                        RNZ
  226.  1E59  B0                        ORA     B
  227.  1E5A  C9                        RET


      ;=============================================================================
                PODPROGRAMY RIDICIHO PROGRAMU
      ;=============================================================================

  229.                                  HLEDANI ADRESY SKOKU V TABULCE SKOKU PODLE KODU PRIKAZU
  230.  1E5B  23                        INX     H
  231.  1E5C  23                        INX     H
  232.  1E5D  7E              TABLE1    MOV     A,M
  233.  1E5E  B7                        ORA     A
  234.  1E5F  C8                        RZ
  235.  1E60  B8                        CMP     E
```

```
 LINE  ADDR  CODE            SYMBOL    MNEM    OPERAND        COMMENT                                  IDENTIF.
 236.  1E61  23                        INX     H
 237.  1E62  C2 5B 1E                  JNZ     TABLE1-2
 238.  1E65  5E                        MOV     E,M
 239.  1E66  23                        INX     H
 240.  1E67  56                        MOV     D,M
 241.  1E68  37                        STC
 242.  1E69  C9                        RET
 243.                                          VYTVARENI KODU PRIKAZOVEHO SLOVA
 244.  1E6A  AF              CODESUM   XRA     A
 245.  1E6B  5F                        MOV     E,A
 246.  1E6C  0A                        LDAX    B
 247.  1E6D  FE 21                     CPI     21H
 248.  1E6F  D8                        RC
 249.  1E70  D6 41                     SUI     41H
 250.  1E72  83                        ADD     E
 251.  1E73  07                        RLC
 252.  1E74  03                        INX     B
 253.  1E75  C3 6B 1E                  JMP     CODESUM+1
 254.                                          PRUCHOD ODDELOVACI, HLEDANI PRVNIHO PLATNEHO ZNAKU
 255.  1E78  03                        INX     B
 256.  1E79  0A              POINTER   LDAX    B
 257.  1E7A  FE 21                     CPI     21H
 258.  1E7C  D0                        RNC
 259.  1E7D  FE 1B                     CPI     1BH
 260.  1E7F  C2 78 1E                  JNZ     POINTER-1
 261.  1E82  C9                        RET

    ;=============================================================================================
                DEROVANI BLOKU PROGRAMU
    ;=============================================================================================

 263.  1E83  3E E5           PUNBLOK   MVI     A,PUNPARLO
 264.  1E85  32 D5 83                  STA     RST3+1
 265.  1E88  CD 88 1E                  CALL    HEXASCII
 266.  1E8B  EB                        XCHG
 267.  1E8C  CD 88 1E                  CALL    HEXASCII
 268.  1E8F  EB                        XCHG
```

```
 LINE  ADDR  CODE            SYMBOL     MNEM    OPERAND        COMMENT                    IDENTIF.
 269.  1E90  7E              PKLOOP     MOV     A,M
 270.  1E91  CD 11 1E                   CALL    HEXASC2
 271.  1E94  CD BC 1D                   CALL    RPEQU
 272.  1E97  23                         INX     H
 273.  1E98  C2 90 1E                   JNZ     PKLOOP
 274.  1E9B  06 14                      MVI     B,20
 275.  1E9D  CD A6 1E                   CALL    BLANKOUT
 276.  1EA0  3E 8C                      MVI     A,PRINTLO
 277.  1EA2  32 D5 83                   STA     RST3+1
 278.  1EA5  C9                         RET
 279.                                   DEROVANI BLANKU
 280.  1EA6  AF              BLANKOUT   XRA     A
 281.  1EA7  CD EC 1D                   CALL    PUNBYTE
 282.  1EAA  05                         DCR     B
 283.  1EAB  C2 A6 1E                   JNZ     BLANKOUT
 284.  1EAE  C9                         RET

      ;=============================================================================
                PRESUN BLOKU DAT V PAMETI
      ;=============================================================================

 286.  1EAF  C5              MOVEBLOK   PUSH    D
 287.  1EB0  EB                         XCHG
 288.  1EB1  CD 3C 1E                   CALL    RPSUB
 289.  1EB4  E5                         PUSH    H
 290.  1EB5  C5                         PUSH    B
 291.  1EB6  E1                         POP     H
 292.  1EB7  C1                         POP     B
 293.  1EB8  CD BC 1D                   CALL    RPEQU
 294.  1EBB  D2 CB 1E                   JNC     BACKMOVE
 295.  1EBE  F1                         POP     PSW
 296.  1EBF  03                         INX     B
 297.  1EC0  1A              DIRLOOP    LDAX    D
 298.  1EC1  77                         MOV     M,A
 299.  1EC2  CD 55 1E                   CALL    BCTEST
 300.  1EC5  C8                         RZ
 301.  1EC6  13                         INX     D
```

```
 LINE  ADDR  CODE            SYMBOL     MNEM    OPERAND        COMMENT                          IDENTIF.
------------------------------------------------------------------------------------------------------
 302.  1EC7  23                         INX     H
 303.  1EC8  C3 C0 1E                   JMP     DIRLOOP
 304.  1ECB  D1              BACKMOVE   POP     D
 305.  1ECC  09                         DAD     B
 306.  1ECD  03                         INX     B
 307.  1ECE  E5                         PUSH    H
 308.  1ECF  1A              BACKLOOP   LDAX    D
 309.  1ED0  77                         MOV     M,A
 310.  1ED1  CD 55 1E                   CALL    BCTEST
 311.  1ED4  1B                         DCX     D
 312.  1ED5  2B                         DCX     H
 313.  1ED6  C2 CF 1E                   JNZ     BACKLOOP
 314.  1ED9  E1                         POP     H
 315.  1EDA  C9                         RET

      ;=================================================================================================
                VYSTUP CISLA Z REGISTRU HL NA PERIFERII S GRAFICKOU UPRAVOU
      ;=================================================================================================

 317.                                                TISK  [NL](HL):[SP]
 318.  1EDB  3E 0D              OUTADR1    MVI     A,NL
 319.  1EDD  CF                         RST     3
 320.  1EDE  CD 08 1E                   CALL    HEXASCII
 321.  1EE1  3E 3A                      MVI     A,':'
 322.  1EE3  CF                         RST     3
 323.  1EE4  3E 20                      MVI     A,' '
 324.  1EE6  CF                         RST     3
 325.  1EE7  C9                         RET
 326.                                                TISK  [NL][SP]*(HL)
 327.  1EE8  3E 0D              OUTADR2    MVI     A,NL
 328.  1EEA  CF                         RST     3
 329.  1EEB  3E 20                      MVI     A,' '
 330.  1EED  CF                         RST     3
 331.  1EEE  3E BD                      MVI     A,0BDH
 332.  1EF0  CF                         RST     3
 333.  1EF1  CD 08 1E                   CALL    HEXASCII
 334.  1EF4  C9                         RET
```

```
 LINE  ADDR  CODE            SYMBOL    MNEM    OPERAND         COMMENT                    IDENTIF.
-------------------------------------------------------------------------------------------------
      ;*************************************************************************************
      ;*            ZPOZDENI URCENE OBSAHEM AKUMULATORU                                    *
      ;*************************************************************************************

  336. 1EF5  C5              DELAY     PUSH    D
  337. 1EF6  57                       MOV     D,A
  338. 1EF7  CD DF 02                  CALL    TIMER
  339. 1EFA  3D                       DCR     A
  340. 1EFB  C2 F6 1E                 JNZ     DELAY+1
  341. 1EFE  C1                       POP     D
  342. 1EFF  C9                       RET
```

```
LINE  ADDR  CODE              SYMBOL    MNEM    OPERAND          COMMENT                IDENTIF.
-----------------------------------------------------------------------------------------------

343.                                    EJECT

      ==========================================================================================
              BLOK 'JADRO 3, CAST'
      ==========================================================================================

345.                                    ORG     3D00H

      ==========================================================================================
              RIDICI A INICIALIZACNI PROGRAM
      ==========================================================================================

347.  3D00  F3                INIT      DI
348.  3D01  21 00 82                    LXI     H,STRING1
349.  3D04  31 9F 82                    LXI     SP,TMSSP
350.  3D07  3E 0F                       MVI     A,0FH
351.  3D09  D3 F4                       OUT     CONTPORT
352.  3D0B  3E 7A                       MVI     A,KEYINPLO
353.  3D0D  32 D2 83                    STA     RST2+1
354.  3D10  3E 8C                       MVI     A,PRINTLO
355.  3D12  32 D5 83                    STA     RST3+1
356.  3D15  AF                          XRA     A
357.  3D16  D3 FA                       OUT     PORTC
358.  3D18  FB                          EI
359.  3D19  CD 25 3F                    CALL    TEXTOUT
360.  3D1C  CD 06 3E                    CALL    WORKLINE
361.  3D1F  CD 79 1E                    CALL    POINTER
362.  3D22  CA 00 3D                    JZ      INIT
363.  3D25  CD 6A 1E                    CALL    CODESUM
364.  3D28  E5                LOOPPAR   PUSH    H
365.  3D29  CD 6B 82                    CALL    INPUTPAR
366.  3D2C  14                          INR     D
367.  3D2D  D2 28 3D                    JNC     LOOPPAR
368.  3D30  4A                          MOV     C,D
369.  3D31  21 9F 82                    LXI     H,JMPTAB
```

```
 LINE  ADDR  CODE            SYMBOL    MNEM    OPERAND        COMMENT
-------------------------------------------------------------------------------
 370.  3D34  CD 5D 1E                  CALL    TABLE1
 371.  3D37  D5                        PUSH    D
 372.  3D38  1E 09                     MVI     E,SYNMES
 373.  3D3A  C2 7F 3D                  JNC     MESSAGE
 374.  3D3D  0D                        DCR     C
 375.  3D3E  79                        MOV     A,C
 376.  3D3F  3D                        DCR     A
 377.  3D40  C9                        RET

;===============================================================================
;               PRIKAZ 'DEFINE'
;===============================================================================

 379.  3D41  C2 7F 3D        DEFINE    JNZ     MESSAGE
 380.  3D44  CD 72 3D                  CALL    CODETAB
 381.  3D47  DA 53 3D                  JC      EXIST
 382.  3D4A  73                        MOV     M,E
 383.  3D4B  D1                        POP     D
 384.  3D4C  23                        INX     H
 385.  3D4D  73                        MOV     M,E
 386.  3D4E  23                        INX     H
 387.  3D4F  72                        MOV     M,D
 388.  3D50  23                        INX     H
 389.  3D51  77                        MOV     M,A
 390.  3D52  EF                        RST     5
 391.  3D53  1E B1           EXIST     MVI     E,EXIMES
 392.  3D55  CD 7F 3D                  CALL    MESSAGE
 393.  3D58  C3 44 3D                  JMP     DEFINE+3

;===============================================================================
;               PRIKAZ 'CANCEL'
;===============================================================================

 395.  3D5B  1E AD                     MVI     E,NEXMES
 396.  3D5D  CD 7F 3D                  CALL    MESSAGE
 397.  3D60  CD 72 3D        SCRATCH   CALL    CODETAB
```

```
LINE  ADDR  CODE              SYMBOL     MNEM     OPERAND            COMMENT                                IDENTIF.
-----------------------------------------------------------------------------------------------------------------------
398.  3D63  C2 5B 3D                     JNC      SCRATCH-5
399.  3D66  44                           MOV      B,H
400.  3D67  4D                           MOV      C,L
401.  3D68  0B                           DCX      B
402.  3D69  0B                           DCX      B
403.  3D6A  23                           INX      H
404.  3D6B  11 FF 82                     LXI      D,ENDJMPTB
405.  3D6E  CD AF 1E                     CALL     MOVEBLOK
406.  3D71  EF                           RST      5

     ;======================================================================================
             PODPROGRAMY RIDICIHO PROGRAMU
     ;======================================================================================

408.                                              VSTUP PRIKAZOVEHO SLOVA A HLEDANI JEHO KODU V TABULCE
409.  3D72  CD A4 3D          CODETAB    CALL     INADR1
410.  3D75  CD 6A 1E                     CALL     CODESUM
411.  3D78  21 9F 82                     LXI      H,JMPTAB
412.  3D7B  CD 5D 1E                     CALL     TABLE1
413.  3D7E  C9                           RET
414.                                              TISK CHYBOVEHO HLASENI
415.  3D7F  7B                MESSAGE    MOV      A,E
416.  3D80  E6 7F                        ANI      7FH
417.  3D82  6F                           MOV      L,A
418.  3D83  26 82                        MVI      H,MESTABHI
419.  3D85  3E 0D                        MVI      A,NL
420.  3D87  CF                           RST      3
421.  3D88  CD 25 3F                     CALL     TEXTOUT
422.  3D8B  3E 0D                        MVI      A,NL
423.  3D8D  CF                           RST      3
424.  3D8E  1C                           INR      E
425.  3D8F  F8                           RM
426.  3D90  21 02 82                     LXI      H,STRING1+2
427.  3D93  C3 04 3D                     JMP      INIT+4
428.                                              ZAPIS A VSTUP PARAMETRU
429.  3D96  3E 0D                        MVI      A,NL
430.  3D98  CF                           RST      3
```

```
LINE  ADDR  CODE              SYMBOL    MNEM   OPERAND          COMMENT                IDENTIF.
------------------------------------------------------------------------------------------------
431.  3D99  E5                INPARAM   PUSH   H
432.  3D9A  CD A7 3D                    CALL   INADR1+3
433.  3D9D  CD 6B 02                    CALL   INPUTPAR
434.  3DA0  7D                          MOV    A,L
435.  3DA1  EB                          XCHG
436.  3DA2  E1                          POP    H
437.  3DA3  C9                          RET
438.                                           ZAPIS PARAMETRU
439.  3DA4  3E 0D             INADR1    MVI    A,NL
440.  3DA6  CF                          RST    3
441.  3DA7  3E 20                       MVI    A,' '
442.  3DA9  CF                          RST    3
443.  3DAA  3E BA                       MVI    A,0BAH
444.  3DAC  CF                          RST    3
445.  3DAD  CD 06 3E                    CALL   WORKLINE
446.  3DB0  C9                          RET
447.                                           PREVOD ASCII-DEC/BIN Z PRAC. POLE PAMETI DO REG. HL
448.  3DB1  16 00             DECCONV   MVI    D,0
449.  3DB3  03                          INX    B
450.  3DB4  0A                          LDAX   B
451.  3DB5  C6 5F                       ADI    95
452.  3DB7  F2 C5 3D                    JP     CONV
453.  3DBA  C6 67                       ADI    103
454.  3DBC  D8                          RC
455.  3DBD  D6 F6                       SUI    246
456.  3DBF  D8                          RC
457.  3DC0  02                          STAX   B
458.  3DC1  14                          INR    D
459.  3DC2  C3 B3 3D                    JMP    DECCONV+2
460.  3DC5  C5                CONV      PUSH   B
461.  3DC6  7A                          MOV    A,D
462.  3DC7  CD CE 3D                    CALL   BCDBIN
463.  3DCA  C1                          POP    B
464.  3DCB  1E 23                       MVI    E,RANMES
465.  3DCD  C9                          RET
466.                                           PREVOD DEC/BIN
467.  3DCE  21 00 00          BCDBIN    LXI    H,0
468.  3DD1  11 50 02                    LXI    D,BCDTAB-1
```

LINE	ADDR	CODE	SYMBOL	MNEM	OPERAND	COMMENT	IDENTIF.
469.	3DD4	F5		PUSH	PSW		
470.	3DD5	D5		PUSH	D		
471.	3DD6	D1	LOOPDIG	POP	D		
472.	3DD7	F1		POP	PSW		
473.	3DD8	3D		DCR	A		
474.	3DD9	F8		RM			
475.	3DDA	F5		PUSH	PSW.		
476.	3DDB	EB		XCHG			
477.	3DDC	23		INX	H		
478.	3DDD	7E		MOV	A,M		
479.	3DDE	23		INX	H		
480.	3DDF	E5		PUSH	H		
481.	3DE0	66		MOV	H,M		
482.	3DE1	6F		MOV	L,A		
483.	3DE2	EB		XCHG			
484.	3DE3	0B		DCX	B		
485.	3DE4	0A		LDAX	B		
486.	3DE5	3D	ADDLOOP	DCR	A		
487.	3DE6	FA D6 3D		JM	LOOPDIG		
488.	3DE9	19		DAD	D		
489.	3DEA	C2 E5 3D		JNC	ADDLOOP		
490.	3DED	D1		POP	D		
491.	3DEE	C1		POP	D		
492.	3DEF	C9		RET			
493.						KONTROLA VSTUPNICH PARAMETRU	
494.	3DF0	E1	PARAM3	POP	H		
495.	3DF1	C1		POP	B		
496.	3DF2	3D		DCR	A		
497.	3DF3	C3 F7 3D		JMP	PARAM2+1		
498.	3DF6	E1	PARAM2	POP	H		
499.	3DF7	3D		DCR	A		
500.	3DF8	C2 7F 3D		JNZ	MESSAGE		
501.	3DFB	D1		POP	D		
502.	3DFC	E3		XTHL			
503.	3DFD	CD 4D 1E		CALL	RPCOMPCH		
504.	3E00	D0		RNC			
505.	3E01	1E 1E		MVI	E,ADRMES		
506.	3E03	C3 7F 3D		JMP	MESSAGE		

```
 LINE  ADDR  CODE            SYMBOL    MNEM    OPERAND          COMMENT                    IDENTIF.
-------------------------------------------------------------------------------------------------
 507.                                            ZAPIS DO PRAC. POLE PAMETI
 508.  3E06  16 00            WORKLINE  MVI     D,0
 509.  3E08  01 00 83                    LXI     B,WORKADR
 510.  3E0B  C5                         PUSH    B
 511.  3E0C  CD 19 3E                   CALL    WLIN
 512.  3E0F  C1                         POP     B
 513.  3E10  F0                         RP
 514.  3E11  1E A9                      MVI     E,OUTMES
 515.  3E13  CD 7F 3D                   CALL    MESSAGE
 516.  3E16  C9                         RET
 517.  3E17  0D                         DCR     C
 518.  3E18  F8                         RM
 519.  3E19  C7            WLIN        RST     2
 520.  3E1A  FE A2                      CPI     0A2H
 521.  3E1C  C2 2A 3E                   JNZ     WN1
 522.  3E1F  DF            WN2         RST     3
 523.  3E20  D7                         RST     2
 524.  3E21  FE A2                      CPI     0A2H
 525.  3E23  C2 1F 3E                   JNZ     WN2
 526.  3E26  DF                         RST     3
 527.  3E27  C3 19 3E                   JMP     WLIN
 528.  3E2A  02            WN1         STAX    B
 529.  3E2B  FE 18                      CPI     18H
 530.  3E2D  C8                         RZ
 531.  3E2E  DF                         RST     3
 532.  3E2F  E6 7F                      ANI     7FH
 533.  3E31  FE 08                      CPI     08H
 534.  3E33  CA 17 3E                   JZ      WLIN-2
 535.  3E36  BA                         CMP     D
 536.  3E37  37                         STC
 537.  3E38  C8                         RZ
 538.  3E39  0C                         INR     C
 539.  3E3A  F8                         RM
 540.  3E3B  C3 19 3E                   JMP     WLIN
```

```
 LINE  ADDR  CODE            SYMBOL    MNEM    OPERAND        COMMENT                    IDENTIF.
-------------------------------------------------------------------------------------------------

 541.                                   EJECT

    ;*************************************************************************************
                PRIKAZ 'C'
    ;*************************************************************************************

 543.  3E3E  C2 7F 3D       CELL      JNZ     MESSAGE
 544.  3E41  E1                       POP     H
 545.  3E42  2B                       DCX     H
 546.  3E43  23             CLOOP     INX     H
 547.  3E44  CD DB 1E                 CALL    OUTADR1
 548.  3E47  7E                       MOV     A,M
 549.  3E48  CD 11 1E                 CALL    HEXASC2
 550.  3E4B  CD 99 3D                 CALL    INPARAM
 551.  3E4E  DA 43 3E                 JC      CLOOP
 552.  3E51  77                       MOV     M,A
 553.  3E52  C3 43 3E                 JMP     CLOOP

    ;*************************************************************************************
                PRIKAZ 'MOV'
    ;*************************************************************************************

 555.  3E55  CD F0 3D       MOVE      CALL    PARAM3
 556.  3E58  CD AF 1E                 CALL    MOVEBLOK
 557.  3E5B  CD EB 1E                 CALL    OUTADR2+3
 558.  3E5E  EF                       RST     5

    ;*************************************************************************************
                PRIKAZ 'CLEAR'
    ;*************************************************************************************

 560.  3E5F  CD F6 3D       CLEAR     CALL    PARAM2
 561.  3E62  CD 66 3E                 CALL    CLR
 562.  3E65  EF                       RST     5
```

```
 LINE  ADDR  CODE            SYMBOL    MNEM    OPERAND        COMMENT                    IDENTIF.
-------------------------------------------------------------------------------------------------
 563.  3E66  AF              CLR       XRA     A
 564.  3E67  77                        MOV     M,A
 565.  3E68  CD BC 1D                  CALL    RPEQU
 566.  3E6B  23                        INX     H
 567.  3E6C  C2 66 3E                  JNZ     CLR
 568.  3E6F  C9                        RET

    ;=============================================================================================
                PRIKAZ 'R'
    ;=============================================================================================

 570.  3E70  21 C7 83        RUN       LXI     H,USESP
 571.  3E73  22 E2 83                  SHLD    SSAVE
 572.  3E76  C2 CC 00                  JNZ     GOTO
 573.  3E79  E1                        POP     H
 574.  3E7A  22 EE 83                  SHLD    ADDR
 575.  3E7D  C3 CC 00                  JMP     GOTO

    ;=============================================================================================
                PRIKAZ 'LIST'
    ;=============================================================================================

 577.  3E80  CD F6 3D        LIST      CALL    PARAM2
 578.  3E83  E5                        PUSH    H
 579.  3E84  21 37 82                  LXI     H,STRING2
 580.  3E87  CD 25 3F                  CALL    TEXTOUT
 581.  3E8A  AF                        XRA     A
 582.  3E8B  F5              TITLOOP   PUSH    PSW
 583.  3E8C  CD 1B 1E                  CALL    HEXASC2+10
 584.  3E8F  3E 20                     MVI     A,' '
 585.  3E91  CF                        RST     3
 586.  3E92  CF                        RST     3
 587.  3E93  F1                        POP     PSW
 588.  3E94  3C                        INR     A
 589.  3E95  FE 10                     CPI     10H
 590.  3E97  C2 8B 3E                  JNZ     TITLOOP
```

```
 LINE  ADDR  CODE            SYMBOL    MNEM    OPERAND          COMMENT                    IDENTIF.
 591.  3E9A  3E 0D                     MVI     A,NL
 592.  3E9C  CF                        RST     3
 593.  3E9D  E1                        POP     H
 594.  3E9E  7D              LLOOP1    MOV     A,L
 595.  3E9F  F5                        PUSH    PSW
 596.  3EA0  E6 F0                     ANI     0F0H
 597.  3EA2  6F                        MOV     L,A
 598.  3EA3  CD DB 1E                  CALL    OUTADR1
 599.  3EA6  F1                        POP     PSW
 600.  3EA7  6F                        MOV     L,A
 601.  3EA8  06 11                     MVI     B,17
 602.  3EAA  E6 0F                     ANI     0FH
 603.  3EAC  CA BA 3E                  JZ      LLOOP2
 604.  3EAF  4F                        MOV     C,A
 605.  3EB0  3E 20                     MVI     A,' '
 606.  3EB2  CF              SPLOOP    RST     3
 607.  3EB3  CF                        RST     3
 608.  3EB4  CF                        RST     3
 609.  3EB5  05                        DCR     B
 610.  3EB6  0D                        DCR     C
 611.  3EB7  C2 B2 3E                  JNZ     SPLOOP
 612.  3EBA  05              LLOOP2    DCR     B
 613.  3EBB  CA 9E 3E                  JZ      LLOOP1
 614.  3EBE  3E 20                     MVI     A,' '
 615.  3EC0  CF                        RST     3
 616.  3EC1  7E                        MOV     A,M
 617.  3EC2  CD 11 1E                  CALL    HEXASC2
 618.  3EC5  CD BC 1D                  CALL    RPEQU
 619.  3EC8  23                        INX     H
 620.  3EC9  C2 BA 3E                  JNZ     LLOOP2
 621.  3ECC  EF                        RST     5

      ;*****************************************************************************************
      ;               PRIKAZ 'READ'
      ;*****************************************************************************************

 623.  3ECD  06 FF           READ      MVI     B,0FFH
```

```
 LINE  ADDR  CODE            SYMBOL   MNEM   OPERAND        COMMENT                    IDENTIF.
 624.  3ECF  CD 19 1D                 CALL   READBLOK
 625.  3ED2  1E 18                    MVI    E,ERRMES
 626.  3ED4  C2 7F 3D                 JNZ    MESSAGE
 627.  3ED7  06 17                    MVI    B,23
 628.  3ED9  D2 CF 3E                 JNC    READ+2
 629.  3EDC  EF                       RST    5

     ;=================================================================================
               PRIKAZ 'READ1'
     ;=================================================================================

 631.  3EDD  06 17          READ1     MVI    B,23
 632.  3EDF  CD 19 1D                 CALL   READBLOK
 633.  3EE2  1E 18                    MVI    E,ERRMES
 634.  3EE4  C2 7F 3D                 JNZ    MESSAGE
 635.  3EE7  DA 00 3D                 JC     INIT
 636.  3EEA  EB                       XCHG
 637.  3EEB  2A EE 03                 LHLD   ADDR
 638.  3EEE  CD E8 1E                 CALL   OUTADR2
 639.  3EF1  EB                       XCHG
 640.  3EF2  CD EB 1E                 CALL   OUTADR2+3
 641.  3EF5  EF                       RST    5

     ;=================================================================================
               PRIKAZY 'PUNCH' A 'PUN'
     ;=================================================================================

 643.  3EF6  C5              PCH1      PUSH   D
 644.  3EF7  CD 6B 02                 CALL   INPUTPAR
 645.  3EFA  D1                       POP    D
 646.  3EFB  EB                       XCHG
 647.  3EFC  CD FD 3D                 CALL   PARAM2+7
 648.  3EFF  06 20                    MVI    B,20H
 649.  3F01  F1                       POP    PSW
 650.  3F02  CC A6 1E                 CC     BLANKOUT
 651.  3F05  CD 03 1E                 CALL   PUNBLOK
```

```
 LINE  ADDR  CODE            SYMBOL    MNEM    OPERAND          COMMENT                    IDENTIF,
-------------------------------------------------------------------------------------------------
 652.  3F08  AF              PUN       XRA     A
 653.  3F09  F5              PUNCH     PUSH    PSW
 654.  3F0A  CD 96 3D                  CALL    INPARAM-3
 655.  3F0D  CD 19 3F                  CALL    PUNCHON
 656.  3F10  D2 F6 3E                  JNC     PCH1
 657.  3F13  06 60                     MVI     B,60H
 658.  3F15  CD A6 1E                  CALL    BLANKOUT
 659.  3F18  EF                        RST     5
 660.                                          ZAPNUTI DEROVACE
 661.  3F19  F5              PUNCHON   PUSH    PSW
 662.  3F1A  3E 8F                    MVI     A,8FH
 663.  3F1C  D3 F4                    OUT     CONTPORT
 664.  3F1E  3E 50                    MVI     A,80
 665.  3F20  CD F5 1E                  CALL    DELAY
 666.  3F23  F1                        POP     PSW
 667.  3F24  C9                        RET

      ;=============================================================================================
                  TISK TEXTU ULOZENEHO V PAMETI
      ;=============================================================================================

 669.  3F25  C5              TEXTOUT   PUSH    B
 670.  3F26  D5                        PUSH    D
 671.  3F27  11 FF FF                  LXI     D,0FFFFH
 672.  3F2A  01 FF 82                  LXI     B,WORKADR-1
 673.  3F2D  CD 45 3F                  CALL    PUTTEXT
 674.  3F30  DA 2A 3F                  JC      TEXTOUT+5
 675.  3F33  D1                        POP     D
 676.  3F34  C1                        POP     B
 677.  3F35  C9                        RET
 678.                                          PODPROGRAM TISKU
 679.  3F36  03              LOOPMEM   INX     B
 680.  3F37  02                        STAX    B
 681.  3F38  D1                        POP     D
 682.  3F39  FE 1B                     CPI     1BH
 683.  3F3B  C8                        RZ
 684.  3F3C  CF                        RST     3
```

```
ADDR  CODE        SYMBOL    MNEM    OPERAND         COMMENT         IDENTIF.
-----------------------------------------------------------------------------
3F3D  FE 0D                 CPI     0DH
3F3F  37                    STC
3F40  C8                    RZ
3F41  CD BC 1D              CALL    RPEQU
3F44  C8                    RZ
3F45  D5          PUTTEXT   PUSH    D
3F46  7E                    MOV     A,M
3F47  23                    INX     H
3F48  FE 80                 CPI     80H
3F4A  DA 36 3F              JC      LOOPMEM
3F4D  FE A0                 CPI     0A0H
3F4F  D2 36 3F              JNC     LOOPMEM
3F52  E6 7F                 ANI     7FH
3F54  57                    MOV     D,A
3F55  0A                    LDAX    B
3F56  15          RPEAT     DCR     D
3F57  CA 36 3F              JZ      LOOPMEM
3F5A  03                    INX     B
3F5B  02                    STAX    B
3F5C  CF                    RST     3
3F5D  C3 56 3F              JMP     RPEAT

:==========================================================================
      PODPROGRAMY PRO VYSTUP DEKADICKEHO CISLA
:==========================================================================

                                    TISK CISLA
3F60  3E 20       BCDOUT    MVI     A,' '
3F62  CF                    RST     3
3F63  3E BD                 MVI     A,0BDH
3F65  CF                    RST     3
3F66  3E 23                 MVI     A,23H
3F68  CF                    RST     3
3F69  CD 84 3F              CALL    BINCON
3F6C  0E 04                 MVI     C,4
3F6E  06 00                 MVI     B,0
3F70  1A          BT1       LDAX    D
```

```
 LINE  ADDR  CODE            SYMBOL    MNEM    OPERAND          COMMENT                    IDENTIF.
 718.  3F71  13                        INX     D
 719.  3F72  B7                        ORA     A
 720.  3F73  CA 78 3F                  JZ      BT2
 721.  3F76  06 80                     MVI     B,80H
 722.  3F78  04                BT2     INR     B
 723.  3F79  FC 1B 1E                  CM      HEXASC2+10
 724.  3F7C  0D                        DCR     C
 725.  3F7D  C2 70 3F                  JNZ     BT1
 726.  3F80  1A                        LDAX    D
 727.  3F81  C3 1B 1E                  JMP     HEXASC2+10
 728.                                  PREVOD CISLA
 729.  3F84  11 61 82          BINCON  LXI     D,DIGITS
 730.  3F87  C5                        PUSH    B
 731.  3F88  D5                        PUSH    D
 732.  3F89  E5                        PUSH    H
 733.  3F8A  CD 91 3F                  CALL    BINBCD
 734.  3F8D  E1                        POP     H
 735.  3F8E  D1                        POP     D
 736.  3F8F  C1                        POP     B
 737.  3F90  C9                        RET
 738.                                  PREVOD BIN/DEC
 739.  3F91  01 F0 D8          BINBCD  LXI     B,0D8F0H
 740.  3F94  CD AC 3F                  CALL    DECNO
 741.  3F97  01 18 FC                  LXI     B,0FC18H
 742.  3F9A  CD AC 3F                  CALL    DECNO
 743.  3F9D  01 9C FF                  LXI     B,0FF9CH
 744.  3FA0  CD AC 3F                  CALL    DECNO
 745.  3FA3  01 F6 FF                  LXI     B,0FFF6H
 746.  3FA6  CD AC 3F                  CALL    DECNO
 747.  3FA9  7D                        MOV     A,L
 748.  3FAA  12                        STAX    D
 749.  3FAB  C9                        RET
 750.  3FAC  AF                DECNO   XRA     A
 751.  3FAD  C5                        PUSH    D
 752.  3FAE  5D                        MOV     E,L
 753.  3FAF  54                        MOV     D,H
 754.  3FB0  3C                        INR     A
 755.  3FB1  09                        DAD     B
```

```
 LINE  ADDR  CODE                SYMBOL     MNEM     OPERAND          COMMENT                         IDENTIF.
---------------------------------------------------------------------------------------------------------------
 756.  3FB2  DA AE 3F                       JC       DECNO+2
 757.  3FB5  3D                             DCR      A
 758.  3FB6  6B                             MOV      L,E
 759.  3FB7  62                             MOV      H,D
 760.  3FB8  D1                             POP      D
 761.  3FB9  12                             STAX     D
 762.  3FBA  13                             INX      D
 763.  3FBB  C9                             RET

      ;*************************************************************************************
              PODPROGRAMY POUZIVANE BLOKEM 'LADENI'
      ;*************************************************************************************

 765.                                                ULOZENI OBSAHU REGISTRU DO PAMETI
 766.  3FBC  E3                  TWSBRENT   XTHL
 767.  3FBD  22 E0 83                       SHLD     PSAVE
 768.  3FC0  F5                             PUSH     PSW
 769.  3FC1  21 04 00                       LXI      H,4
 770.  3FC4  39                             DAD      SP
 771.  3FC5  F1                             POP      PSW
 772.  3FC6  22 E2 83                       SHLD     SSAVE
 773.  3FC9  E1                             POP      H
 774.  3FCA  31 EC 83                       LXI      SP,DATA
 775.  3FCD  F5                             PUSH     PSW
 776.  3FCE  C5                             PUSH     B
 777.  3FCF  D5                             PUSH     D
 778.  3FD0  E5                             PUSH     H
 779.  3FD1  31 9F 82                       LXI      SP,TWSSP
 780.  3FD4  3A F2 83                       LDA      BRKCT
 781.  3FD7  B7                             ORA      A
 782.  3FD8  CA EF 3F                       JZ       BRKSTOP
 783.  3FDB  2A F0 83                       LHLD     BRKAD
 784.  3FDE  EB                             XCHG
 785.  3FDF  2A E0 83                       LHLD     PSAVE
 786.  3FE2  CD BC 1D                       CALL     RPEQU
 787.  3FE5  C2 F9 01                       JNZ      RESRG
 788.  3FE8  21 F2 83                       LXI      H,BRKCT
```

```
 LINE  ADDR  CODE          SYMBOL    MNEM    OPERAND          COMMENT                    IDENTIF.
-------------------------------------------------------------------------------------------------
 789.  3FEB  35                      DCR     M
 790.  3FEC  C3 F9 01                JMP     RESRG
 791.  3FEF  3E F0           BRKSTOP MVI     A,0F0H
 792.  3FF1  D3 FA                   OUT     PORTC
 793.  3FF3  FB                      EI
 794.  3FF4  CD 91 01                CALL    ADDSP
 795.  3FF7  D7                      RST     2
 796.  3FF8  FE 1B                   CPI     1BH
 797.  3FFA  C2 F7 3F                JNZ     BRKSTOP+8
 798.  3FFD  C3 E4 0F                JMP     SP2
```

```
 LINE  ADDR  CODE            SYMBOL    MNEM    OPERAND        COMMENT                    IDENTIF.
-------------------------------------------------------------------------------------------------

 799.                                   EJECT

    ;=============================================================================================
              BLOK 'TEXT'
    ;=============================================================================================

 801.                                   ORG     3C00H

    ;=============================================================================================
              ZAPIS TEXTU DO PAMETI
    ;=============================================================================================

 803.  3C00  2B                         DCX     H
 804.  3C01  D7             GETTEXT     RST     2
 805.  3C02  77                         MOV     M,A
 806.  3C03  FE 1B                      CPI     1BH
 807.  3C05  37                         STC
 808.  3C06  C8                         RZ
 809.  3C07  DF                         RST     3
 810.  3C08  FE 08                      CPI     08H
 811.  3C0A  CA 00 3C                   JZ      GETTEXT-1
 812.  3C0D  4F                         MOV     C,A
 813.  3C0E  CD BC 1D                   CALL    RPEQU
 814.  3C11  C8                         RZ
 815.  3C12  23                         INX     H
 816.  3C13  06 01                      MVI     B,1
 817.  3C15  D5                         PUSH    D
 818.  3C16  D7             CHCOUNT     RST     2
 819.  3C17  57                         MOV     D,A
 820.  3C18  B9                         CMP     C
 821.  3C19  C2 24 3C                   JNZ     GT1
 822.  3C1C  DF                         RST     3
 823.  3C1D  04                         INR     B
 824.  3C1E  78                         MOV     A,B
 825.  3C1F  FE 20                      CPI     32
```

```
ASSEMBLER AS180A/ID        I.DOLEZAL! PRIDAVNY MONITOR 'TMS' PRO TK-80                IDENTIF.
LINE  ADDR  CODE          SYMBOL     MNEM    OPERAND          COMMENT
----------------------------------------------------------------------------------------------
                                     JNZ     CHCOUNT
826.  3C21  C2 16 3C      GT1        MOV     A,D
827.  3C24  7A                       POP     D
828.  3C25  D1                       DCR     B
829.  3C26  05                       JZ      GETTEXT+1
830.  3C27  CA 02 3C                 MOV     C,A
831.  3C2A  4F                       MOV     A,B
832.  3C2B  78                       ORI     80H
833.  3C2C  F6 80                    MOV     M,A
834.  3C2E  77                       CALL    RPEQU
835.  3C2F  CD BC 1D                 RZ
836.  3C32  C8                       INX     H
837.  3C33  23                       MOV     A,C
838.  3C34  79                       JMP     GETTEXT+1
839.  3C35  C3 02 3C

        ========================================================================================
                PRIKAZ 'KEY'
        ========================================================================================

                          TEXTIN     LXI     D,0FFFFH
841.  3C38  11 FF FF                 JZ      TN1
842.  3C3B  CA 49 3C                 DCR     A
843.  3C3E  3D                       MVI     E,SYNMES
844.  3C3F  1E 09                    JNZ     MESSAGE
845.  3C41  C2 7F 3D                 POP     D
846.  3C44  D1                       MVI     A,1BH
847.  3C45  3E 1B                    STAX    D
848.  3C47  12                       DCX     D
849.  3C48  1B            TN1        POP     H
850.  3C49  E1                       CALL    PARAM2+7
851.  3C4A  CD FD 3D                 MVI     A,NL
852.  3C4D  3E 0D                    RST     3
853.  3C4F  CF                       CALL    GETTEXT
854.  3C50  CD 01 3C                 MVI     E,29H
855.  3C53  1E 29                    JNC     MESSAGE
856.  3C55  D2 7F 3D                 CALL    OUTADR2
857.  3C58  CD E8 1E                 RST     5
858.  3C5B  EF
```

```
LINE  ADDR  CODE             SYMBOL    MNEM    OPERAND         COMMENT                IDENTIF.
-----------------------------------------------------------------------------------------------
     ================================================================================
            PRIKAZ 'PRINT'
     ================================================================================

860.  3C5C  11 FF FF         TEXTPRN   LXI     D,0FFFFH
861.  3C5F  CA 68 3C                   JZ      TP1
862.  3C62  3D                         DCR     A
863.  3C63  C2 7F 3D                   JNZ     MESSAGE
864.  3C66  C1                         POP     D
865.  3C67  13                         INX     D
866.  3C68  E1               TP1       POP     H
867.  3C69  CD FD 3D                   CALL    PARAM2+7
868.  3C6C  3E 0D                      MVI     A,NL
869.  3C6E  DF                         RST     3
870.  3C6F  01 FF 02         TP2       LXI     B,WORKADR-1
871.  3C72  CD 45 3F                   CALL    PUTTEXT
872.  3C75  CA 6F 3C                   JC      TP2
873.  3C78  EF                         RST     5

     ================================================================================
            PRIKAZY 'STORE' A 'STOR'
     ================================================================================

875.  3C79  3E 0D            STORE     MVI     A,NL
876.  3C7B  DF                         RST     3
877.  3C7C  CD 19 3F                   CALL    PUNCHON
878.  3C7F  06 20                      MVI     B,20H
879.  3C81  CD A6 1E                   CALL    BLANKOUT
880.  3C84  21 00 08         STOR      LXI     H,BUFFER
881.  3C87  11 FF 0F                   LXI     D,BUFEND
882.  3C8A  3E 1B                      MVI     A,1BH
883.  3C8C  12                         STAX    D
884.  3C8D  1B                         DCX     D
885.  3C8E  E5                         PUSH    H
886.  3C8F  CD 01 3C                   CALL    GETTEXT
887.  3C92  E1                         POP     H
888.  3C93  CD 19 3F                   CALL    PUNCHON
```

```
 LINE  ADDR  CODE            SYMBOL   MNEM   OPERAND        COMMENT                    IDENTIF.
-------------------------------------------------------------------------------------------
  889. 3C96  7E                        MOV    A,M
  890. 3C97  FE 1B                     CPI    1BH
  891. 3C99  CA 13 3F                  JZ     PUNCH+10
  892. 3C9C  3E E5                     MVI    A,PUNPARLO
  893. 3C9E  32 D5 83                  STA    RST3+1
  894. 3CA1  CD 25 3F                  CALL   TEXTOUT
  895. 3CA4  3E 1B                     MVI    A,1BH
  896. 3CA6  DF                        RST    3
  897. 3CA7  3E 8C                     MVI    A,PRINTLO
  898. 3CA9  32 D5 83                  STA    RST3+1
  899. 3CAC  06 14                     MVI    B,20
  900. 3CAE  C3 81 3C                  JMP    STORE+8

;=============================================================================================
                PRIKAZ 'TEXT'
;=============================================================================================

  902. 3CB1  06 FF            TEXT     MVI    B,0FFH
  903. 3CB3  21 00 08                  LXI    H,BUFFER
  904. 3CB6  E5                        PUSH   H
  905. 3CB7  CD 2D 1D                  CALL   READON
  906. 3CBA  CD 3D 1D                  CALL   BLANK
  907. 3CBD  CA 00 3D                  JZ     INIT
  908. 3CC0  B7              TT1      ORA    A
  909. 3CC1  1E 18                     MVI    E,ERRMES
  910. 3CC3  E2 7F 3D                  JPO    MESSAGE
  911. 3CC6  E6 7F                     ANI    7FH
  912. 3CC8  77                        MOV    M,A
  913. 3CC9  23                        INX    H
  914. 3CCA  11 FF 0F                  LXI    D,BUFEND
  915. 3CCD  CD BC 1D                  CALL   RPEQU
  916. 3CD0  3E 1B                     MVI    A,1BH
  917. 3CD2  CA DD 3C                  JZ     TT2-5
  918. 3CD5  CD AA 1D                  CALL   READBYTE
  919. 3CD8  FE 1B                     CPI    1BH
  920. 3CDA  C2 C0 3C                  JNZ    TT1
  921. 3CDD  77                        MOV    M,A
```

```
 LINE  ADDR  CODE            SYMBOL   MNEM   OPERAND        COMMENT                    IDENTIF.
-----------------------------------------------------------------------------------------------
  922. 3CDE  E1                        POP    H
  923. 3CDF  CD 25 3F                  CALL   TEXTOUT
  924. 3CE2  C7                TT2     RST    2
  925. 3CE3  FE 18                     CPI    18H
  926. 3CE5  C2 E2 3C                  JNZ    TT2
  927. 3CE8  06 17                     MVI    B,23
  928. 3CEA  C3 83 3C                  JMP    TEXT+2
```

```
LINE  ADDR  CODE        SYMBOL   MNEM    OPERAND        COMMENT                         IDENTIF.
-----------------------------------------------------------------------------------------------
                                 EJECT
929.
        ;==========================================================================
        ;       BLOK 'LADENI'
        ;==========================================================================
                                 ORG     0E00H
931.
        ;==========================================================================
        ;       PRIKAZY 'RL' A 'RC'
        ;==========================================================================
933.  0E00  11 06 0F    RL       LXI     D,0F06H
934.  0E03  C3 09 0E             JMP     REG
935.  0E06  11 06 F0    RC       LXI     D,0F006H
936.  0E09  21 EC 83    REG      LXI     H,REGAREA+1
937.  0E0C  01 41 82             LXI     B,REGNAME
938.  0E0F  C5                   PUSH    D
939.  0E10  3E 0D                MVI     A,NL
940.  0E12  DF                   RST     3
941.  0E13  0A                   LDAX    B
942.  0E14  DF                   RST     3
943.  0E15  03                   INX     B
944.  0E16  0A                   LDAX    B
945.  0E17  DF                   RST     3
946.  0E18  03                   INX     B
947.  0E19  CD E1 1E             CALL    OUTADR1+6
948.  0E1C  2B                   DCX     H
949.  0E1D  7E                   MOV     A,M
950.  0E1E  CD 11 1E             CALL    HEXASC2
951.  0E21  2B                   DCX     H
952.  0E22  7E                   MOV     A,M
953.  0E23  CD 11 1E             CALL    HEXASC2
954.  0E26  15                   DCR     D
955.  0E27  F2 36 0E             JP      RSKIP
```

```
 LINE  ADDR  CODE            SYMBOL    MNEM    OPERAND          COMMENT                                  IDENTIF,
-------------------------------------------------------------------------------------------------------------------
 956.  0E2A  C5                        PUSH    B
 957.  0E2B  CD 99 3D                  CALL    INPARAM
 958.  0E2E  C1                        POP     B
 959.  0E2F  CA 36 0E                  JC      RSKIP
 960.  0E32  23                        INX     H
 961.  0E33  72                        MOV     M,D
 962.  0E34  2B                        DCX     H
 963.  0E35  73                        MOV     M,E
 964.  0E36  D1              RSKIP     POP     D
 965.  0E37  1D                        DCR     E
 966.  0E38  C2 0F 0E                  JNZ     REG+6
 967.  0E3B  EF                        RST     5

      ;*************************************************************************************************
                PRIKAZY 'SL' A 'SC'
      ;*************************************************************************************************

 969.  0E3C  0E F0             SC      MVI     C,0F0H
 970.  0E3E  C3 43 0E                  JMP     STACK
 971.  0E41  0E 0F             SL      MVI     C,0FH
 972.  0E43  11 C7 83          STACK   LXI     D,USESP
 973.  0E46  2A E2 83                  LHLD    SSAVE
 974.  0E49  CD BC 1D                  CALL    RPEQU
 975.  0E4C  CA 00 3D                  JZ      INIT
 976.  0E4F  C5                        PUSH    D
 977.  0E50  3E 0D                     MVI     A,NL
 978.  0E52  DF                        RST     3
 979.  0E53  E5                        PUSH    H
 980.  0E54  7E                        MOV     A,M
 981.  0E55  23                        INX     H
 982.  0E56  66                        MOV     H,M
 983.  0E57  6F                        MOV     L,A
 984.  0E58  CD 08 1E                  CALL    HEXASCII
 985.  0E5B  E1                        POP     H
 986.  0E5C  0D                        DCR     C
 987.  0E5D  F2 6C 0E                  JP      STSKIP
 988.  0E60  C5                        PUSH    B
```

```
 LINE  ADDR  CODE            SYMBOL    MNEM    OPERAND        COMMENT                    IDENTIF.
 989.  0E61  CD 99 3D                  CALL    INPARAM
 990.  0E64  C1                        POP     B
 991.  0E65  DA 6C 0E                  JC      STSKIP
 992.  0E68  73                        MOV     M,E
 993.  0E69  23                        INX     H
 994.  0E6A  72                        MOV     M,D
 995.  0E6B  2B                        DCX     H
 996.  0E6C  D1              STSKIP    POP     D
 997.  0E6D  23                        INX     H
 998.  0E6E  23                        INX     H
 999.  0E6F  C3 49 0E                  JMP     STACK+6

      ;=================================================================================
                PRIKAZ 'REL'
      ;=================================================================================

1001.  0E72  CD F0 3D       REL       CALL    PARAM3
1002.  0E75  C5                       PUSH    B
1003.  0E76  D5                       PUSH    D
1004.  0E77  E5                       PUSH    H
1005.  0E78  CD AF 1E                 CALL    MOVEBLOK
1006.  0E7B  CD EB 1E                 CALL    OUTADR2+3
1007.  0E7E  22 5B 82                 SHLD    P2
1008.  0E81  C1                       POP     B
1009.  0E82  D1                       POP     D
1010.  0E83  CD 3C 1E                 CALL    RPSUB
1011.  0E86  22 5F 82                 SHLD    ADDADR
1012.  0E89  E1                       POP     H
1013.  0E8A  22 5D 82                 SHLD    P1
1014.  0E8D  CD 79 0F                 CALL    SEARCH
1015.  0E90  EF                       RST     5
```

```
 LINE  ADDR  CODE            SYMBOL    MNEM    OPERAND         COMMENT                    IDENTIF.
-------------------------------------------------------------------------------------------------

1016.                                   EJECT

     ;=================================================================================
                PRIKAZ 'PRGM'
     ;=================================================================================

1018.  0E91  21 5F 82       PRGM      LXI     H,P1+2
1019.  0E94  01 4D 82                 LXI     B,BOUNDNAM
1020.  0E97  11 02 F0                 LXI     D,0F002H
1021.  0E9A  C3 0F 0E                 JMP     REC+6

     ;=================================================================================
                PRIKAZ 'INOP'
     ;=================================================================================

1023.  0E9D  3D             INOP      DCR     A
1024.  0E9E  C2 7F 3D                 JNZ     MESSAGE
1025.  0EA1  D1                       POP     D
1026.  0EA2  E1                       POP     H
1027.  0EA3  CD 53 0F                 CALL    PLACE
1028.  0EA6  EF                       RST     5

     ;=================================================================================
                PRIKAZ 'INS'
     ;=================================================================================

1030.  0EA7  C2 7F 3D       INS       JNZ     MESSAGE
1031.  0EAA  E1                       POP     H
1032.  0EAB  3E 0D                    MVI     A,NL
1033.  0EAD  CF                       RST     1
1034.  0EAE  CD 08 1E                 CALL    HEXASCII
1035.  0EB1  E5                       PUSH    H
1036.  0EB2  E5                       PUSH    H
1037.  0EB3  CD 99 3D                 CALL    INPARAM
```

```
 LINE  ADDR  CODE               SYMBOL    MNEM    OPERAND          COMMENT                    IDENTIF.
------------------------------------------------------------------------------------------------------
1038.  0EB6  CA 00 3D                     JC      INIT
1039.  0EB9  11 00 00                     LXI     D,0
1040.  0EBC  1C                 IS1       INR     E
1041.  0EBD  CD 6B 82                     CALL    INPUTPAR
1042.  0EC0  D2 BC 0E                     JNC     IS1
1043.  0EC3  E1                           POP     H
1044.  0EC4  CD 53 0F                     CALL    PLACE
1045.  0EC7  E1                           POP     H
1046.  0EC8  01 00 83                     LXI     B,WORKADR
1047.  0ECB  E5                 IS2       PUSH    H
1048.  0ECC  CD 6B 82                     CALL    INPUTPAR
1049.  0ECF  7D                           MOV     A,L
1050.  0ED0  E1                           POP     H
1051.  0ED1  DA AB 0E                     JC      INS+4
1052.  0ED4  77                           MOV     M,A
1053.  0ED5  23                           INX     H
1054.  0ED6  C3 CB 0E                     JMP     IS2

     ;=================================================================================
            PRIKAZ 'DEL'
     ;=================================================================================

1056.  0ED9  CD F6 3D           DEL       CALL    PARAM2
1057.  0EDC  44                           MOV     B,H
1058.  0EDD  4D                           MOV     C,L
1059.  0EDE  2B                           DCX     H
1060.  0EDF  CD 3C 1E                     CALL    RPSUB
1061.  0EE2  22 5F 82                     SHLD    ADDADR
1062.  0EE5  2A 5B 82                     LHLD    P2
1063.  0EE8  EB                           XCHG
1064.  0EE9  23                           INX     H
1065.  0EEA  E5                           PUSH    H
1066.  0EEB  D5                           PUSH    D
1067.  0EEC  CD AF 1E                     CALL    MOVEBLOK
1068.  0EEF  22 5B 82                     SHLD    P2
1069.  0EF2  D1                           POP     D
1070.  0EF3  23                           INX     H
```

```
 LINE  ADDR  CODE            SYMBOL    MNEM    OPERAND        COMMENT                    IDENTIF,
-------------------------------------------------------------------------------------------------
1071.  0EF4  CD 66 3E                   CALL    CLR
1072.  0EF7  C3 0C 0F                   JMP     SHIFT+18

      ;=============================================================================
                PRIKAZ 'SHF'
      ;=============================================================================

1074.  0EFA  CD F0 3D       SHIFT       CALL    PARAM3
1075.  0EFD  E5                         PUSH    H
1076.  0EFE  D5                         PUSH    D
1077.  0EFF  CD AF 1E                   CALL    MOVEBLOK
1078.  0F02  CD EB 1E                   CALL    OUTADR2+3
1079.  0F05  D1                         POP     D
1080.  0F06  CD 3C 1E                   CALL    RPSUB
1081.  0F09  22 5F 82                   SHLD    ADDADR
1082.  0F0C  C1                         POP     B
1083.  0F0D  2A 5D 82                   LHLD    P1
1084.  0F10  CD 79 0F                   CALL    SEARCH
1085.  0F13  EF                         RST     5

      ;=============================================================================
                PRIKAZ 'CHG'
      ;=============================================================================

1087.  0F14  3D             CHANGE      DCR     A
1088.  0F15  C2 7F 3D                   JNZ     MESSAGE
1089.  0F18  C1                         POP     B
1090.  0F19  D1                         POP     D
1091.  0F1A  2A 5D 82                   LHLD    P1
1092.  0F1D  7E             CH1         MOV     A,M
1093.  0F1E  C5                         PUSH    B
1094.  0F1F  CD AA 0F                   CALL    LENGTH
1095.  0F22  C1                         POP     B
1096.  0F23  C2 44 0F                   JNZ     CH2
1097.  0F26  CA 43 0F                   JC      CH3
1098.  0F29  E5                         PUSH    H
```

```
 LINE  ADDR  CODE           SYMBOL   MNEM    OPERAND        COMMENT                              IDENTIF.
---------------------------------------------------------------------------------------------------------
1099.  0F2A  23                      INX     H
1100.  0F2B  7E                      MOV     A,M
1101.  0F2C  23                      INX     H
1102.  0F2D  66                      MOV     H,M
1103.  0F2E  6F                      MOV     L,A
1104.  0F2F  CD BC 1D                CALL    RPEQU
1105.  0F32  E1                      POP     H
1106.  0F33  C2 42 0F                JNZ     CH3-1
1107.  0F36  3E 0D                   MVI     A,NL
1108.  0F38  DF                      RST     3
1109.  0F39  E5                      PUSH    H
1110.  0F3A  CD 08 1E                CALL    HEXASCII
1111.  0F3D  23                      INX     H
1112.  0F3E  71                      MOV     M,C
1113.  0F3F  23                      INX     H
1114.  0F40  70                      MOV     M,B
1115.  0F41  E1                      POP     H
1116.  0F42  23                      INX     H
1117.  0F43  23             CH3      INX     H
1118.  0F44  23             CH2      INX     H
1119.  0F45  D5                      PUSH    D
1120.  0F46  EB                      XCHG
1121.  0F47  2A 5B 82                LHLD    P2
1122.  0F4A  CD 4D 1E                CALL    RPCOMPDH
1123.  0F4D  EB                      XCHG
1124.  0F4E  C1                      POP     D
1125.  0F4F  DA 1D 0F                JC      CH1
1126.  0F52  EF                      RST     5

      :=========================================================================================
                 PODPROGRAMY PRO UPRAVU PROGRAMU
      :=========================================================================================

1128.                                        VLOZENI PRAZDNYCH INSTRUKCI NOP
1129.  0F53  E5             PLACE    PUSH    H
1130.  0F54  E5                      PUSH    H
1131.  0F55  44                      MOV     B,H
```

```
 LINE  ADDR  CODE            SYMBOL    MNEM    OPERAND        COMMENT                    IDENTIF.
1132.  0F56  4D                         MOV     C,L
1133.  0F57  EB                         XCHG
1134.  0F58  22 5F 82                   SHLD    ADDADR
1135.  0F5B  E5                         PUSH    H
1136.  0F5C  2A 5B 82                   LHLD    P2
1137.  0F5F  EB                         XCHG
1138.  0F60  2A 5D 82                   LHLD    P1
1139.  0F63  CD 79 0F                   CALL    SEARCH
1140.  0F66  E1                         POP     H
1141.  0F67  09                         DAD     B
1142.  0F68  44                         MOV     B,H
1143.  0F69  4D                         MOV     C,L
1144.  0F6A  E1                         POP     H
1145.  0F6B  C5                         PUSH    B
1146.  0F6C  CD AF 1E                   CALL    MOVEBLOK
1147.  0F6F  22 5B 82                   SHLD    P2
1148.  0F72  D1                         POP     D
1149.  0F73  1B                         DCX     D
1150.  0F74  E1                         POP     H
1151.  0F75  CD 66 3E                   CALL    CLR
1152.  0F78  C9                         RET
1153.                                   UPRAVA ADRES SKOKU
1154.  0F79  D5            SEARCH       PUSH    D
1155.  0F7A  C5                         PUSH    B
1156.  0F7B  7E                         MOV     A,M
1157.  0F7C  CD AA 0F                   CALL    LENGTH
1158.  0F7F  C1                         POP     B
1159.  0F80  C2 9C 0F                   JNZ     SH1
1160.  0F83  23                         INX     H
1161.  0F84  CA 9C 0F                   JC      SH1
1162.  0F87  E5                         PUSH    H
1163.  0F88  7E                         MOV     A,M
1164.  0F89  23                         INX     H
1165.  0F8A  66                         MOV     H,M
1166.  0F8B  6F                         MOV     L,A
1167.  0F8C  CD 43 1E                   CALL    RPCOMP
1168.  0F8F  CA 97 0F                   JC      SH2
1169.  0F92  EB                         XCHG
```

```
 LINE   ADDR  CODE            SYMBOL    MNEM    OPERAND          COMMENT                    IDENTIF.
1170.  0F93  2A 5F 82                   LHLD    ADDADR
1171.  0F96  19                         DAD     D
1172.  0F97  EB               SH2       XCHG
1173.  0F98  E1                         POP     H
1174.  0F99  73                         MOV     M,E
1175.  0F9A  23                         INX     H
1176.  0F9B  72                         MOV     M,D
1177.  0F9C  23               SH1       INX     H
1178.  0F9D  EB                         XCHG
1179.  0F9E  2A 5B 82                   LHLD    P2
1180.  0FA1  CD 4D 1E                   CALL    RPCOMPDH
1181.  0FA4  EB                         XCHG
1182.  0FA5  C1                         POP     D
1183.  0FA6  DA 79 0F                   JC      SEARCH
1184.  0FA9  C9                         RET
1185.                                           URCENI DELKY INSTRUKCE
1186.  0FAA  47               LENGTH    MOV     B,A
1187.  0FAB  FE C3                     CPI     0C3H
1188.  0FAD  C8                         RZ
1189.  0FAE  FE CD                     CPI     0CDH
1190.  0FB0  C8                         RZ
1191.  0FB1  E6 F7                     ANI     0F7H
1192.  0FB3  FE D3                     CPI     0D3H
1193.  0FB5  37                         STC
1194.  0FB6  C8                         RZ
1195.  0FB7  E6 E7                     ANI     0E7H
1196.  0FB9  FE 22                     CPI     22H
1197.  0FBB  C8                         RZ
1198.  0FBC  E6 C7                     ANI     0C7H
1199.  0FBE  FE 06                     CPI     6
1200.  0FC0  37                         STC
1201.  0FC1  C8                         RZ
1202.  0FC2  FE C6                     CPI     0C6H
1203.  0FC4  37                         STC
1204.  0FC5  C8                         RZ
1205.  0FC6  FE C2                     CPI     0C2H
1206.  0FC8  C8                         RZ
1207.  0FC9  FE C4                     CPI     0C4H
```

```
 LINE  ADDR  CODE               SYMBOL     MNEM    OPERAND          COMMENT                    IDENTIF.
---------------------------------------------------------------------------------------------------------
1208.  0FCB  C8                            RZ
1209.  0FCC  78                            MOV     A,B
1210.  0FCD  E6 CF                         ANI     0CFH
1211.  0FCF  FE 01                         CPI     1
1212.  0FD1  C9                            RET

      ;==========================================================================
             PRIKAZY 'S' A 'CE'
      ;==========================================================================

1214.  0FD2  0E 01              STEP       MVI     C,1
1215.  0FD4  CA DC 0F                      JZ      SP1
1216.  0FD7  3D                            DCR     A
1217.  0FD8  C2 E4 0F                      JNZ     SP2
1218.  0FDB  C1                            POP     B
1219.  0FDC  79                  SP1        MOV     A,C
1220.  0FDD  32 F2 83                      STA     BRKCT
1221.  0FE0  E1                            POP     H
1222.  0FE1  22 F0 83                      SHLD    BRKAD
1223.  0FE4  F3                  SP2        DI
1224.  0FE5  3E FF                         MVI     A,0FFH
1225.  0FE7  C3 FA                         OUT     PORTC
1226.  0FE9  C3 F9 01            CONTINUE   JMP     RESRG

      ;==========================================================================
             PRIKAZ 'G'
      ;==========================================================================

1228.  0FEC  C2 7F 3D            GO         JNZ     MESSAGE
1229.  0FEF  21 C7 83                      LXI     H,USESP
1230.  0FF2  22 E2 83                      SHLD    SSAVE
1231.  0FF5  E1                            POP     H
1232.  0FF6  22 E0 83                      SHLD    PSAVE
1233.  0FF9  C3 E4 0F                      JMP     SP2
```

```
 LINE  ADDR  CODE              SYMBOL    MNEM    OPERAND          COMMENT                    IDENTIF.
-----------------------------------------------------------------------------------------------------

1234.                                    EJECT


    ;*************************************************************************************
            BLOK 'JADRO 2. CAST'
    ;*************************************************************************************

1236.                                    ORG     8241H


    ;*************************************************************************************
            TABULKY
    ;*************************************************************************************

1238.  8241  41464243          REGNAME   DB      'AFBC'
1239.  8245  4445484C                    DB      'DEHL'
1240.  8249  53505043                    DB      'SPPC'
1241.  824D  42504550          BOUNDNAM  DB      'BPEP'
1242.  8251  01 00             BCDTAB    DW      1
1243.  8253  0A 00                       DW      10
1244.  8255  64 00                       DW      100
1245.  8257  E8 03                       DW      1000
1246.  8259  10 27                       DW      10000
1247.  825B                    P2        DS      2
1248.  825D                    P1        DS      2
1249.  825F                    ADDADR    DS      2
1250.  8261                    DIGITS    DS      5
1251.                                    ORG     826BH
1252.                                            PODPROGRAM PRO VSTUP PARAMETRU Z PRAC. POLE PAMETI
1253.  826B  CD 79 1E          INPUTPAR  CALL    POINTER
1254.  826E  37                          STC
1255.  826F  C8                          RZ
1256.  8270  C5                          PUSH    D
1257.  8271  1E 10                       MVI     E,ILLMES
1258.  8273  FE 23                       CPI     23H
1259.  8275  C4 2C 1E                    CNZ     HEXACONV
1260.  8278  CC B1 3D                    CZ      DECCONV
```

```
 LINE  ADDR  CODE            SYMBOL    MNEM    OPERAND        COMMENT              IDENTIF.
-------------------------------------------------------------------------------------------
1261.  827B  CA 7F 3D                  JC      MESSAGE
1262.  827E  C1                        POP     D
1263.  827F  C9                        RET
1264.                                  ORG     8380H
1265.  8380  1B                        DB      1BH

    ======================================================================
            PRIKAZ 'CAL'
    ======================================================================

1267.  8381  CD EB 1E                  CALL    OUTADR2+3
1268.  8384  CD 60 3F                  CALL    BCDOUT
1269.  8387  CD 96 3D        CAL       CALL    INPARAM-3
1270.  838A  EB                        XCHG
1271.  838B  CD 79 1E        CALNEXT   CALL    POINTER
1272.  838E  CA 81 83                  JZ      CAL-6
1273.  8391  F5                        PUSH    PSW
1274.  8392  03                        INX     B
1275.  8393  E5                        PUSH    H
1276.  8394  CD 6B 82                  CALL    INPUTPAR
1277.  8397  EB                        XCHG
1278.  8398  E1                        POP     H
1279.  8399  CD 3C 1E                  CALL    RPSUB
1280.  839C  F1                        POP     PSW
1281.  839D  FE 2D                     CPI     '-'
1282.  839F  CA 8B 83                  JZ      CALNEXT
1283.  83A2  19                        DAD     D
1284.  83A3  19                        DAD     D
1285.  83A4  FE 2B                     CPI     '+'
1286.  83A6  CA 8B 83                  JZ      CALNEXT
1287.  83A9  1E 09                     MVI     E,SYNMES
1288.  83AB  C3 7F 3D                  JMP     MESSAGE
```

```
 LINE  ADDR  CODE            SYMBOL    MNEM    OPERAND        COMMENT                          IDENTIF.
-------------------------------------------------------------------------------------------------------

1289.                                   EJECT
1290.                                   ORG     83D1H


      ;=============================================================================================
                TABULKA RESTART
      ;=============================================================================================

1292.  83D1  C3 7A 1D       RST2      JMP     KEYINP
1293.  83D4  C3 8C 1D       RST3      JMP     PRINT
1294.  83D7  C3 BC 3F       RST4      JMP     TWSBRENT
1295.  83DA  C3 00 3D       RST5      JMP     INIT
```

```
 LINE  ADDR  CODE              SYMBOL     MNEM     OPERAND          COMMENT                        IDENTIF.
------------------------------------------------------------------------------------------------------------

1296.                                     EJECT

     ;====================================================================================
              SYMBOLICKE ADRESY
     ;====================================================================================

1298.                          PORTB      EQU      0F9H
1299.                          PORTC      EQU      0FAH
1300.                          CONTPORT   EQU      0F4H
1301.                          KEYBRD     EQU      0F6H
1302.                          READER     EQU      0F7H
1303.                          TWRITER    EQU      0F6H
1304.                          PUNCHER    EQU      0F7H
1305.                          MONITOR    EQU      8
1306.                          KEYIN      EQU      0216H
1307.                          TWSSP      EQU      829FH
1308.                          WORKADR    EQU      8300H
1309.                          BUFFER     EQU      0800H
1310.                          BUFEND     EQU      0FFFH
1311.                          JMPTAB     EQU      829FH
1312.                          ENDJMPTB   EQU      82FFH
1313.                          MESTABHI   EQU      02H
1314.                          SYNMES     EQU      09H
1315.                          ILLMES     EQU      10H
1316.                          RANMES     EQU      23H
1317.                          ADRMES     EQU      1EH
1318.                          ERRMES     EQU      18H
1319.                          EXIMES     EQU      0B1H
1320.                          NEXMES     EQU      0ADH
1321.                          OUTMES     EQU      0A9H
1322.                          NL         EQU      0DH
1323.                          STRING1    EQU      8200H
1324.                          STRING2    EQU      8237H
1325.                          KEYINPLO   EQU      07AH
1326.                          PRINTLO    EQU      08CH
1327.                          PUNPARLO   EQU      0E5H
```

```
 LINE  ADDR  CODE            SYMBOL     MNEM     OPERAND          COMMENT                                  IDENTIF.
-------------------------------------------------------------------------------------------------------------------
1328.                                             SYMBOLY MONITORU TK-80
1329.                       RGDSP      EQU      01A1H
1330.                       ERROR      EQU      02CBH
1331.                       TIMER      EQU      02DFH
1332.                       GOTO       EQU      00CCH
1333.                       ADDSP      EQU      0191H
1334.                       RESRG      EQU      01F9H
1335.                       DATA       EQU      03ECH
1336.                       ADDR       EQU      03EEH
1337.                       PSAVE      EQU      03E0H
1338.                       SSAVE      EQU      03E2H
1339.                       BRKAD      EQU      03F0H
1340.                       BRKCT      EQU      03F2H
1341.                       REGAREA    EQU      03E8H
1342.                       USESP      EQU      03C7H
1343.                                  END
```

SYMBOL TABLE

NO.	SYMBOL	DEC	HEX
1	MONREAD	7424	1D00
2	READBLOK	7449	1D19
3	READON	7469	1D2D
4	RN	7473	1D31
5	BLANK	7485	1D3D
6	RBLOCK	7494	1D46
7	LOOPBYTE	7511	1D57
8	READWORD	7530	1D6A
9	KEYINP	7546	1D7A
10	KN	7553	1D81
11	PRINT	7564	1D8C
12	PT1	7570	1D92
13	PT2	7585	1DA1
14	READBYTE	7594	1DAA
15	RE	7601	1DB1
16	RPEQU	7612	1DBC
17	ASCHEXA	7618	1DC2
18	ASCHEX1	7630	1DCE
19	ADD7	7648	1DE0
20	ADD30	7650	1DE2
21	PUNPAR	7653	1DE5
22	PUNBYTE	7660	1DEC
23	PE1	7664	1DF0
24	PE2	7675	1DFB
25	HEXASCII	7688	1E08
26	HEXASC2	7697	1E11
27	HEXASC1	7712	1E20
28	CNO	7721	1E29
29	HEXACONV	7724	1E2C
30	RPSUB	7740	1E3C
31	RPCOMP	7747	1E43
32	RPCOMPOH	7757	1E4D
33	BCTEST	7765	1E55
34	TABLE1	7773	1E5D

SYMBOL TABLE

NO.	SYMBOL	DEC	HEX
35	CODESUM	7786	1E6A
36	POINTER	7801	1E79
37	PUNBLOK	7811	1E83
38	PKLOOP	7824	1E90
39	BLANKOUT	7846	1EA6
40	MOVEBLOK	7855	1EAF
41	DIRLOOP	7872	1EC0
42	BACKMOVE	7883	1ECB
43	BACKLOOP	7887	1ECF
44	OUTADR1	7899	1EDB
45	OUTADR2	7912	1EE8
46	DELAY	7925	1EF5
47	INIT	15616	3D00
48	LOOPPAR	15656	3D28
49	DEFINE	15681	3D41
50	EXIST	15699	3D53
51	SCRATCH	15712	3D60
52	CODETAB	15730	3D72
53	MESSAGE	15743	3D7F
54	INPARAM	15769	3D99
55	INADR1	15780	3DA4
56	DECCONV	15793	3DB1
57	CONV	15813	3DC5
58	BCDBIN	15822	3DCE
59	LOOPDIG	15830	3DD6
60	ADDLOOP	15845	3DE5
61	PARAM3	15856	3DF0
62	PARAM2	15862	3DF6
63	WORKLINE	15878	3E06
64	WLIN	15897	3E19
65	WN2	15903	3E1F
66	WN1	15914	3E2A
67	CELL	15934	3E3E
68	CLOOP	15939	3E43
69	MOVE	15957	3E55

SYMBOL TABLE

NO.	SYMBOL	DEC	HEX
70	CLEAR	15967	3E5F
71	CLR	15974	3E66
72	RUN	15984	3E70
73	LIST	16000	3E80
74	TITLOOP	16011	3E8B
75	LLOOP1	16030	3E9E
76	SPLOOP	16050	3EB2
77	LLOOP2	16058	3EBA
78	READ	16077	3ECD
79	READ1	16093	3EDD
80	PCH1	16118	3EF6
81	PUN	16136	3F08
82	PUNCH	16137	3F09
83	PUNCHON	16153	3F19
84	TEXTOUT	16165	3F25
85	LOOPMEM	16182	3F36
86	PUTTEXT	16197	3F45
87	RPEAT	16214	3F56
88	BCDOUT	16224	3F60
89	BT1	16240	3F70
90	BT2	16248	3F78
91	BINCON	16260	3F84
92	BINBCD	16273	3F91
93	DECNO	16300	3FAC
94	TWSBRENT	16316	3FBC
95	BRKSTOP	16367	3FEF
96	GETTEXT	15361	3C01
97	CHCOUNT	15382	3C16
98	GT1	15396	3C24
99	TEXTIN	15416	3C38
100	TN1	15433	3C49
101	TEXTPRN	15452	3C5C
102	TP1	15464	3C68
103	TP2	15471	3C6F
104	STORE	15481	3C79

SYMBOL TABLE

NO.	SYMBOL	DEC	HEX
105	STOR	15492	3C84
106	TEXT	15537	3CB1
107	TT1	15552	3CC0
108	TT2	15586	3CE2
109	RL	3584	0E00
110	RC	3590	0E06
111	REG	3593	0E09
112	RSKIP	3638	0E36
113	SC	3644	0E3C
114	SL	3649	0E41
115	STACK	3651	0E43
116	STSKIP	3692	0E6C
117	REL	3698	0E72
118	PRGM	3729	0E91
119	INOP	3741	0E9D
120	INS	3751	0EA7
121	IS1	3772	0EBC
122	IS2	3787	0ECB
123	DEL	3801	0ED9
124	SHIFT	3834	0EFA
125	CHANGE	3860	0F14
126	CH1	3869	0F1D
127	CH3	3907	0F43
128	CH2	3908	0F44
129	PLACE	3923	0F53
130	SEARCH	3961	0F79
131	SH2	3991	0F97
132	SH1	3996	0F9C
133	LENGTH	4010	0FAA
134	STEP	4050	0FD2
135	SP1	4060	0FDC
136	SP2	4068	0FE4
137	CONTINUE	4073	0FE9
138	GO	4076	0FEC
139	REGNAME	33345	8241

SYMBOL TABLE

NO.	SYMBOL	DEC	HEX
140	BOUNDNAM	33357	824D
141	BCDTAB	33361	8251
142	P2	33371	825B
143	P1	33373	825D
144	ADDADR	33375	825F
145	DIGITS	33377	8261
146	INPUTPAR	33387	826B
147	CAL	33671	8387
148	CALNEXT	33675	838B
149	RST2	33745	83D1
150	RST3	33748	83D4
151	RST4	33751	83D7
152	RST5	33754	83DA
153	PORTB	249	00F9
154	PORTC	250	00FA
155	CONTPORT	244	00F4
156	KEYBRD	246	00F6
157	READER	247	00F7
158	TWRITER	246	00F6
159	PUNCHER	247	00F7
160	MONITOR	8	0008
161	KEYIN	534	0216
162	TWSSP	33439	829F
163	WORKADR	33536	8300
164	BUFFER	2048	0800
165	BUFEND	4095	0FFF
166	JMPTAB	33439	829F
167	ENDJMPTB	33535	82FF
168	MESTABHI	130	0082
169	SYNMES	9	0009
170	ILLMES	16	0010
171	RANMES	35	0023
172	ADRMES	30	001E
173	ERRMES	24	0018
174	EXIMES	177	00B1

SYMBOL TABLE

NO.	SYMBOL	DEC	HEX
175	NEXMES	173	00AD
176	OUTMES	169	00A9
177	NL	13	000D
178	STRING1	33280	8200
179	STRING2	33335	8237
180	KEYINPLO	122	007A
181	PRINTLO	140	008C
182	PUNPARLO	229	00E5
183	RGDSP	417	01A1
184	ERROR	715	02CB
185	TIMER	735	02DF
186	GOTO	204	00CC
187	ADDSP	401	0191
188	RESRG	505	01F9
189	DATA	33772	83EC
190	ADDR	33774	83EE
191	PSAVE	33760	83E0
192	SSAVE	33762	83E2
193	BRKAD	33776	83F0
194	BRKOT	33778	83F2
195	REGAREA	33771	83EB
196	USESP	33735	83C7

Příl. 12 VÝPIS NEPŘEKLÁDANÝCH ČÁSTÍ TWS

tws :
"Tabulka textu MESTAB"

LIST 8200 8240

```
addr   0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F

8200:  0D 0D F4 F7 F3 20 20 BA 1B F3 F9 EE F4 E1 F8 1B
8210:  E9 EC EC E5 E7 E1 EC 1B E5 F2 F2 EF F2 1B E1 E4
8220:  E4 F2 1B F2 E1 EE E7 E5 1B 6F 75 74 1B 6E 6F 74
8230:  20 65 78 69 73 74 1B 0D 0D 61 64 64 72 20 20 20
8240:  1B
```

tws :
"Tabulka adres JMPTAB"

LIST 829F 82FF

```
addr   0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F

8290:                                                0E
82A0:  41 3D 87 60 3D 04 3E 3E 5F 80 3E C2 55 3E 33 5F
82B0:  3E 37 CD 3E 4E DD 3E A1 09 3F E2 08 3F 26 87 83
82C0:  22 70 3E 0C EC 0F 10 E9 0F 24 D2 0F 5A 00 0E 48
82D0:  06 0E 5E 41 0E 4C 3C 0E AE 72 0E A9 91 0E 3F 9D
82E0:  0E 98 A7 0E 3E D9 0E B6 FA 0E 38 14 0F 90 38 3C
82F0:  8D 5C 3C 2C 79 3C 12 84 3C D3 B1 3C 00 00 00 00
```