

SEZNAM_DOKUMENTOV_FRIROČNIKA_ "Programske_opreme_laboratorijskega_mikroračunalnika_sistema_Lakradata_1680-10"....049_340_500

	Namembnost			
	A	B	C	D
V46100010				
BEXEC01MD	A	B	C	D
V46325044	A	B	C	D
BEDIT01MD	A	B	C	D
BASMB01MD	A	B	C	D
V46325043	A	B	C	D
B683401MD	A	B	C	D
BDISS01MD	A	B	C	D
BMONI01MD	A	B	C	D
BZBIRO1MD	A	B	C	D
BBSIC01MD	A	B	C	D

	Primerak je izd	Potpis	Građiva	Odgovori se začet. mjer	Topografska obveznica	Površ. zaštitna	Pripremio	
Kazalnik		Izdaja					101680-10	
Projekt		Znak						
Pregled		St. odr.						
Števil		Datum						
Stand.		Potpis						
Kod	PRIROČNIK SW. ID 1680-10				list	stran	J X identifikacijska številka	
					1		A49340500	
Nosilcem/osebnost kopije					Arhiv	Merklo	Sobeseda	Nomero identifikacijske številke

PROGRAMSKA OPREMA LABORATORIJSKEGA MIKRORAČUNALNIŠKEGA
SISTEMA ISKRADATA 1680-10

Pričetek trajanja izdajanja in spremljača v odgovornosti menedžerja sistema Iskradata.

Izdaja	Stran	J	K	Identifikacijska številka
Št. obvestila	2			V461000010
 Iskra Elektromehanika Kranj	Arhiv			Namensko identifikacijska številka

V S E B I N A

1. FDOS ORGANIZACIJA SISTEMA
2. OPERACIJSKI SISTEM ISKRA FDOS
3. UPORABA GIBKIH DISKOV
4. IZVRŠEVALNI PROGRAM (EXECUTIVE)
5. MONITOR
6. EDITOR
7. ASMB
8. DEBUG
9. 6834
10. DISS
11. MONI
12. ZBIR

Izdaja										J	K	Identifikacijska številka
Št. izdajstva										3		V46100010
	Iskra Elektromehanika Kranj									Avtor		Namenska identifikacijska številka

1. FDOS ORGANIZACIJA SISTEMA

ISKRA FDOS PROGRAMSKI MODULI

ISKRA FDOS vsebujejo naslednje module:

- FDOS rezidenčni modul
- FDOS izvrševalec (EXECUTIVE)
- Programske datoteke

1.1. FDOS rezidenčni modul

FDOS rezidenčni modul je del FDOS - a, ki je vsebovan v EPROM spominu. V dedatku vsebuje vhodno / izhodne programe za disketo in FDOS zagonski izvrševalec. Ta rezidenčni program je na razpolago za uporabo izvrševanja disketnih bralnih in disketnih vpisovalnih operacij uporabniškega programa. Kontrolni program (DEBUGGER) je ločeni del rezidenčnega modula in je uporaben za kontroliranje in kreiranje uporabniških programov.

1.2. FDOS izvrševalec (EXECUTIVE)

Kadar programsko kontrolo prenesemo s pomočjo RESET tipke ali ob vključitvi sistema, tedaj FDOS zagonski nalagalec vsebovan v FDOS rezidenčnem modulu, prenese FDOS izvrševalec z diskete v RAM spomin centralne enote ISKRADATA 1680. Po prikazu znaka ! na zaslonu, je FDOS izvrševalec v RAM spominu in čaka na FDOS ukaz. FDOS izvrševalec izvrši interpretacijo vsake komandne linije, uporabljanje datoteke, in FDOS operacijske funkcije.

1.3. Programske datoteke

Programske datoteke vsebujejo absolutno strojni zapis v šestnajstiški (heksa decimalni) obliki. Start programske datotek vsebujejo ukazi EDIT, ASMB in RUN. Uporabnik lahko izdela svoje programske datoteke.

1.4. Položaj FDOS programov na disketi

Razen za FDOS rezidenčni modul, vsi FDOS programi so shranjeni na disketi. Spominski program diskete je razdeljen v dva ali tri ločena področja, odvisno od tega ali je disketa sistemsko ali uporabniška.

Izdeja						Ust.	Stran	J	K	Identifikacijska številka
Št. obvestila						4				V46100010
	Iskra Elektromehanika Krani					Avtor				Namensko identifikacijska številka

1.5. FDOS sistemski disketa je razdeljena v tri ločena področja: področje disketnega imenika, sistemsko področje, in uporabniško področje. Disketa je izdelana, če z FDOS programskim pakiranjem predhodno naložimo FDOS sistemski disketo. Na sistemski disketi je sled 0 rezervirana za disketni imenik., sledi 1 do 3 pa so rezervirane za zapis FDOS izvrševalca (EXECUTIVE), in ostanek je namenjen za uporabnikove datoteke.

1.6. Uporabniška disketa je razdeljena v dve ločeni področji: področje disketnega imenika in področje uporabniških datotek. Na uporabniški disketi je sled 0 rezervirana za disketni imenik in ostanek je za uporabniške datoteke. Ker uporabniška disketa ne vsebuje sistemskoga področja, je maksimalni shranjevalni prostor na voljo za shranjevanje uporabniških datotek.

1.7. FDOS DISKETNE DATOTEKE

Definicija

Izraz " datoteka " se nanaša na vsako zbirko informacij. Tipični primer so datoteke, ki vsebujejo programske strojne informacije, datoteke, ki vsebujejo programske izvirne informacije, in datoteke, ki vsebujejo uporabniške generirane podatkovne informacije.

Lokacija

Vse disketne datoteke so shranjene v uporabniškem področju diskete. Kreirani datoteki, je namenjen povezan prostor na disketi, ki je namenjen za shranitev te datoteke in je nadaljevan z začetkom naslednje datoteke itd. Mesta informacije o datoteki, ki se nahaja na disketi, je v področju disketnega imenika. Ob brisanju datoteke, je prostor v uporabniškem področju, ki je bil do sedaj zaseden s to datoteko kreiran z praznim in na voljo uporabo tehnike pakiranja v kateri FDOS vzame vse informacije o začetni lokaciji brisanja datoteke (začetna sled, začetni sektor itd.)

Izdaja							Ust	Stran	J	K	Identifikacijska številka
Št. obveznika							5				V46100010
	Iskra Elektromehanika Krani						Avtor				Namensko identifikacijsko številko

Imena datotek

Vsaka datoteka v uporabniškem področju diskete je dostopna z imenom datoteke. To datotečno ime je shranjeno skupaj z drugimi primernimi podatki te datoteke, v področju datotečnega imenika diskete, na kateri je datoteka nameščena. Datotečno ime je sestavljeno iz ASCII znakov in mora biti dolga od 1 do 5 znakov. V primeru, da vnesemo datotečno ime daljše od 5 znakov, bo FDOS vedno vzel le prvih pet znakov.

Pripone enot (DEVICE SUFFIXES)

Imenom datotek je dodano še specifikacija enote. Specifikacija enote je številka pogonskega mehanizma, ki je z dvopičjem ločena od datotečnega imena.

Primer:

JOE 3 : 1

SAM : 3

X : 0

JACK : 2

Pripone enote uporabljam za razlikovanje disket, ki se nahajajo v različnih pogonskih mehanizmih :0, 1, 2, ali 3.

V primeru da je pri imenu datoteke pripona enote izpuščena, se upošteva datoteka, ki se nahaja na disketi nameščeni v pogonskem mehanizmu 0, torej sistemski disketa.

1.8. Datotečni atributi

Datotečni atributi so karakteristike datotek in se z uporabo lahko postavljeni in spremenjeni. Ti atributi so vedno definirani v FDOS-u in so naslednji:

00 - uporabniška datoteka, ne ovirana

01 - stalna datoteka, ni možno brisanje z DELET ukazom

80 - označuje brisano datoteko *

FF - označuje konec imenika *

* Ti atributi avtomatično manipulirajo z FDOS-om in zato niso na voljo uporabniku.

V en sektor lahko zapišete 128 bytov. Ena sled pa vsebuje 26 sektorjev (16 tiško 1A), ali 3328 bytov. Na sistemski disketi je na razpolago za datoteke 73 sledi; zadnja od teh je sled 77 (16 tiško 4D). Celotna podatkovna kapaciteta je 242944 bytov.

Izdeja					Ust	Stran	J	K	Identifikacijska številka
Št. obvestila					6				V46100010
	Iskra Elektromehanika Krani				Achiv				Namensko identifikacijska številka

2. OPERACIJSKI SISTEM ISKRA FDOS

ISKRA FDOS (Floppy Disk Operating System) je paket programov z naslednjimi funkcijami:

- omogoča učinkovito in zanesljivo uporabo računalnika;
- omogoča enostavno vodenje delovanja računalnika
- omogoča programsko podporo razvoju: pisanje uporabnikovih programov (aplikacijskih).

Funkcije operacijskega sistema ISKRA FDOS:

- IZVRŠEVALNI SISTEM (EXECUTIVE SYSTEM)
- UREJEVALNIK TEKSTA (TEXT EDITOR)
- MAKRO ZBIRNIK (MACRO ASSEMBLER)
- TESTNO KONTROLNI PROGRAM (DEBUGGER)
- MONITOR

2.1. IZVRŠEVALNI SISTEM je program, ki omogoča uporabniku, da na enostaven način z uporabo specialnega upravljačkega jezika upravlja delo računalnika.

2.2. UREJEVALNIK TEKSTA (text editor) prvenstveno služi za pišanje in popravljanje (korekcijo) programov v simboličnem zbirnem jeziku ter za njihovo nalaganje na disketo.

2.3. MAKRO ZBIRNIK (macro assembler) je program, ki programe napisane v simboličnem jeziku zbirnika prevaja v strojni jezik, uporaben za mikroprocesor 6800.

2.4. TESTNO KONTROLNI PROGRAM (debugger) je program za pomoč pri iskanju napak v programih. Nudi možnost izvrševanja programov " korak za korakom ", zasledovanje vsebine in vpisovanje novih vrednosti registrov in spominskih lokacij, postavljanje točk prekinitev itd.

Izdelek	St. obvestila	Ust.	Stran	J	K	Identifikacijska številka
		7				V46100010
	Skra Elektromehanika Krani	Arhiv				Namensko identifikacijska številka

2.5. MONITOR je program z naslednjimi funkcijami:

- omogoča enostaven pristop vhodno/izhodni opremi (tastatura, video prikazovalnik, tičkalnik, gibki disk); delo z vhodno/izhodno opremo se v uporabniških programih lahko odvija z enostavnim klicanjem (pozivanjem) odgovarjajočih rutin, vsebovanih v monitorju;
- daje na razpolago uporabniku rutine splošnega pomena, npr.: rutina za prevod (konverzijo) iz ASCII heksadecimalnega v binarni zapis, rutine za pregled tabel določenih oblik itd;

■ gledano z drugega zunega kota, operacijski sistem se lahko razdeli na:

- I. IZVRŠILNI (EXECUTIVE) SISTEM
- II. REZIDENČNI MODUL
- III. PROGRAMSKE DATOTEKE

I. IZVRŠILNI (EXECUTIVE) SISTEM je program, ki se nahaja na takojmenovanih sistemskih disketah. Njegovo nalaganje v spomin centralne enote se vrši avtomatsko za RESET rutino, to je po pritisku na tipko RESET ali po vklopitvi napajanja.

II. REZIDENČNI MODUL gradita programa monitor in kontrolno testni program (debugger); vpisana sta v EPROM integrirana vezja centralne enote, ter sta dostopna v vsakem trenutku.

III. PROGRAMSKE DATOTEKE sestavljajo vsi ostali programi. Nahajajo se na disketah, v spomin centralne enote se nalagajo le tedaj, če jih koristnik s pomočjo komandnega jezika eksplicitno zahteva.

- od vseh programov ima makro zbirnik (macro assembler) največje zahteve po spominskem prostoru v centralni enoti: 16K RAM-a (naslovi od 0000 - 3FFF (HEX)).

Indija					List	Stran	J	K	Identifikacijska številka
Št. obvestila					8				V46100010
	Skra Elektromehanika Kranj				Arhiv				Namenska identifikacijska številka

Za uporabo gibkih diskov

Pri uporabi ID 1680 enote gibkega diska je prvi korak vključitve napajanja, šele za tem vložimo disketo. Podobno je pred zapustitvijo sistema zadnji korak izključitev enote gibkega diska.

Izklučitev ali vključitev napajanja, medtem ko je disketa vložena, lahko spremeni podatke zapisane na disketi, zato se tega izogibamo, če je le mogoče. Poleg tega se vratca na odprtinah ne morejo odpreti, če je prišlo do spremembe napajanja, medtem ko so bila zaprta.

Disketa je izdelana iz krhkega materiala, zato mora biti delo z njo previdno: varovati jo moramo pred toploto in vlagom, ne smemo je zvijati ali lomiti.

V vsakdanji rabi je disketa trajno zaprta v zaščitno ovojnice in je dostopna skozi dolgo meščekadar diskete se uporabljam je spravimo v papirni ovitek. Tako jo zaščitimo pred prahom in dečki rok.

Vratca odpremo tako, da dvignemo zapah nad vratci. Disketo vložimo v pogonski mehanizem gibkega diska tako, da je etiketa zgoraj na zunanji strani.

Disketo vložimo v režo pazljivo. Če je za to potrebno večja sila, je nekaj narobe s pogonskim mehanizmom gibkega diska. Vzmet v pogonskem mehanizmu gibkega diska poriva disketo nazadven. Ko vložimo disketo do konca v režo, se zaskoči ob držalo na desni strani. Za pravilno delovanje 1680 mora pogonski mehanizem 0/na levi strani/ vsebovati sistemsko disketo. Pogonski mehanizem 1 je lahko ob tem v uporabi, ali pa ne, vsebuje pa lahko sistemsko ali pa uporabniško disketo.

Izdaja							Ust	Stran	J	K	Identifikacijska številka
Št. obveznika							9				V46100010
	Skra Elektromehanika Kranj										Namensko identifikacijska številka
							Arhiv				

Obe vrsti disket/sistemskih in uporabniških se ločita med seboj po tem, da vsebuje sistemska disketa na sledih 1 do 3 FDOS IZ-WEŠILNI program /EXEC/ sled pa vsebuje disketni imenik, ki se vpiše v pomnilnik po ponovnem startu. Uporabniška disketa vsebuje na sledih 1 do 3 podatek ali kakšen drug program in je nemoremo uporabiti za ponovno startanje sistema.

Sistemski disketa je lahko neposredni posnetek (kopija) sistema diskete, ali pa je izdelana z ukazom za formiranje sistema diskete (XGEN). Uporabniška disketa je lahko rezultat ukaza za popolno brisanje wejtove diskete (INIT) ali je neposredni posnetek (kopija) takšne diskete /COPY ukaz/.

Ko smo vložili željeno disketo, zapremo vratca, počakamo nekaj sekund, nato ponovno startamo sistem. V diskovni enoti moramo saslišati tlesk, nato vidimo utripanje statusnih lučk, na zaslonu pa se prikaže sporočilo:

ISKRA FDOS VI-01

1 -

Če se ne zgodi nič takega, se prepričamo, če je vključen ID 1680 in poskusimo s ponovnim startom. Če se pojavi kontrolno testni program (DEBUGGER), potem disketa v pogonskem mehanizmu gibkega diska verjetno ni sistemski. Preverimo naš postopek in vse ponovimo snova dokler ne dobimo sporočila, na katerega čakamo.

TSKVA EPOS VI-01

Slike 3.1. VSTOP V ISKRA FDOS SISTEM

Namen tega dela je vpeljati uporabnika v ISKRA FDOS sistem. Pregled, opis in uporaba nekaterih ukazov naj bi uporabniku omogočil delo s sistemom. Podrobnejši opis vseh ukazov je podan v priročniku za - ISKRA - FDOS operacijski sistem.

Izdaža						Ust	Stran	J	K	Identifikacijska številka
Št obvestila						10				V46100010
	skra Elektromehanika Krani					Aribit				Namensko identifikacijska številka

FDOS čaka na ukaz, potem ko v prvi stolpec napiše klicaj /!. Uporabnik zaključi svoj ukaz s tipko CARRIAGE RETURN /CR/. Pred tem lahko vsak vtipkani znak zbrisemo s tipko ← . Med vpisom lahko vsak ukaz uničimo s RUBOUT tipko.

Vse ukaze zapišemo z velikimi tiskanimi črkami. Ukazi morajo biti zapisani neposredno za klicajem, brez vmesnih presledkov.

Za listanje disketnega imenika uporabimo ukaz LDIR /CR/. Slika 3.2.

ISKRA FDOS V1.0!

! LDIR

<u>NAME</u>	<u>ATTR</u>	<u>TRAK</u>	<u>SCTR</u>	<u>SIZE</u>
ASMB	01	04	01	0073
DIAGO	01	08	06	001D
EDIT	01	09	0F	0027
EXEC	01	0B	02	004B
GPIO	01	0D	19	0033
LIFE	01	0F	18	000C
P6834	01	10	0A	0011
RS001	01	11	01	000D
RSXX1	01	11	0E	000C
TRACE	01	11	1A	0016
ISKRA	01	12	16	0007

!

Slika 3.2. Listanje disketnega imenika

Če še nimamo rezervne kopije sistemske diskete, je pametno, da si jo naredimo pred nadaljnšim eksperimentiranjem.

Izdaja						Ust	Stran	J	K	Identifikacijska številka	
Št. obveznika						11				V46100010	
	skra Elektromehanika Kranj						Arhiv	Namesto identifikacijske številke			

Vstavimo novo disketo v pogonski mehanizem gibkega diska in uporabimo ukaz COPY. V roku petih minut bo pogonski mehanizem gibkega diska 1 vseboval natančno kopijo diskete v pogonskem mehanizmu gibkega diska 0. Kopiranje je zaključeno, ko FDOS izpiše naslednji klicaj.

ISKRA FDOS 01.01

! LDIR

<u>NAME</u>	<u>ATTR</u>	<u>TRAK</u>	<u>SCTR</u>	<u>SIZE</u>
ASMB	01	04	01	0073
DIAGO	01	08	0C	0010
EDIT	01	09	0F	0027
EXEC	01	08	02	004B
GPIO	01	0D	19	0033
LIFE	01	0F	18	000C
P6834	01	10	0A	0011
RS001	01	11	01	000D
RSXX1	01	11	0E	000C
TRACE	01	11	1A	0016
VIEW	01	12	16	0007
TOC	01	13	03	0007

! INIT : 1

! COPY

!

Slika 3.3. KOPIRANJE DISKETE

Sedaj zamenjamo originalno sistemsko disketo z novo in ponovno startamo in ponovimo LDIR postopek, da se prepričamo o uspešnosti kopiranja.

Izdaja						Let	Stran	J	K	Identifikacijska številka
Št. obvestila						12		LV46100010		
	Iskra Elektromehanika Kranj					Avtov		Namesto identifikacijske številke		

Z uporabo urejevalnika /EDIT/ lahko na disketi izdelamo programske datoteke katere ime je lahko dolgo 5 alfanumeričnih znakov. Če si izberemo za ime nove datoteke ISKRA, vtipkamo ukaz:

! EDIT ,ISKRA CR

Tik preden pritisnemo na CR na tastaturi prikazovalnika nastavimo na Page način. Vse nadaljnje ukaze bo sprejemal urejevalni program. Vpišimo sedaj izvirni program, ki ga izpišemo na disketo z ESC tipko. S tem se hkrati vrnemo v FDOS.

Izvirni program zapisan v sbirnem jeziku naj izgleda takole:

NAM SAMPL

```
*** SAMPLE TEST
ORG $100
CLR A
LLOOP INCA
NOP
INCA
NOP
BRA LLOOP
END
ESC
```

OPOMBA: Edine postavke v stolpcu 1 so lahko:

M za komentar

L - prva črka labele

Ko nam FDOS zopet izpiše vstopno sporočilo, je datoteka pod imenom ISKRA zapisan na disketi v pogonskem mehanizmu gibkega diska0. Da se o tem prepričamo, uporabimo LDIR ukaz, samo datoteko pa lahko pregledamo z ukazom:

! VIEW, ISKRA CR

Izvirni program prevedemo v strojni program s programom ASMB:

! ASMB, ISKRA,ISKRA,11 CR

Izdaja							list	Stran	J	K	Identifikacijska številka
St. obveznila							13				V46100010
	Skra Elektromehanika Kranji										Namensko identifikacijska številka

Tako dobimo na disketi novo datoteko pod imenom ISKRA, ki vsebuje strojni program, na zaslonu pa dobimo listanje prevajanja.

Z ukasom VIEW, ISKRE **CR** dobimo na zaslonu tale zapis:

```
S0 06 0000 484452 1B
S1 0A 0100 4F4C014C0120FA F1
S9 03 0000 FC
```

S0, S1 in S9 predpone pomenijo "začetek programa", "del programa", "konec programa". 0100 je naslov za strojni program, ki sledi. Primerjamo vse to z izpisom prevoda.

ISKRA	1680	SAMPL	STRAN	0001
ASMB	V1.1			
SEQ	LOC	OBJ	SOURCE	
0001			NAM	SAMPL
0002				
0003		+SAMPLE	TEST	
0004	0100		ORG	S100
0005	0100 4F		CLR	A
0006	0101 4C	LOOP	INC	A
0007	0102 01		NOP	
0008	0103 4C		INC	A
0009	0104 01		NOP	
0010	0105 20 FA		BRA	LOOP
0011				

0011 LINES ASSEMBLED, LOC = 0107, 0000 ERRORS DETECTED.
END ASMB V1.1.

Z ukasom LOAD, ISKRE **CR** naložimo program v pomnilnik in avtomatično vstopimo v kontrolno testni program (DEBUGGER).

Eksperimentirajmo s kontrolno testnim programom (DEBUGGER - jem). Lahko izvajamo program po korakih /CTR/Q/ in pri tem opazujemo

Izdaja						Ust	Stran	J	K	Identifikacijska številka
St. obvestila						14				V46100010
	Iskra Elektromehanika Kranji						Arhiv			Rametsko identifikacijska številka

spreminjanje vrednosti registrov P / sanka od naslova 0101 do naslova 0105 / in A /nenehne maraščata /.

Da damo programu ISAKRS popolne kontrole, nastavimo BRK = FFFF in pritisnemo na ETX. Čez nekaj trenutkov ponovno pritisnemo na ETX. Sedaj je vrednost registra A naključna, potem ko se je nekaj tisočkrat ali milijonkrat povečala za 1.

V tem trenutku lahko preidemo na priročnik za programske knjižnice in pričnemo z resničnim mikrorazčunalniškim razvojem.

Izdaš	Št obvestila	Identifikacijska številka
		V461000010
Arhiv	Namensko identifikacijska številka	
 Iskra Elektromehanika Kranj		

was broken because it was too i ~~large~~ ~~big~~ ~~big~~ ~~big~~ ~~big~~ ~~big~~ ~~big~~.

Primerik in imenini	Podpis	Gradivo		Odstopi netol. mer		Topljena obdelava		Površ. zaščita	Pripravost				
									ID1680-10				
Koestr.		Izdaja											
Projekt		Znak											
Pregled		Št. obr.											
Števil		Datum											
Stand		Podpis											
Naziv								List	Stran	J	K	Identifikacijska številka	
PROGRAM IZVRSEVALEC							1					BEXEC01MD	
Namembnost kopije	Arhiv							Število	Sekcija	Noveste identifikacijske številke			

4.1 IZVRŠEVALNI PROGRAM (EXECUTIVE) je del operacijskega sistema ki ima funkcijo sprejemanja in interpretiranja ukazov za delovanje sistema. Ukazi se vnašajo preko tastature, vizuelna kontrola pa se vrši s pomočjo video prikazovalnika (display-a). Za njihovo definiranje se koristi takoimenovani ukazni (komandni) jezik, o katerem bo govor v nadaljevanju.

Program se nahaja na disketi, ter je potrebno, da ga postavimo v aktivno stanje. To dosežemo na naslednji način: v pogonski mehanizem gibkega diska (drive) Ø je potrebno vstaviti sistemsko disketo, zapreti vratca pogonskega mehanizma za tem centralno enoto priključiti na izvor električnega napajanja (oziroma pritisniti na tipko RESET, če je napajanje že vključeno). Nato se prične avtomatsko nalaganje tega programa v centralni spomin. Po nekaj sekundah je proces končan in se na zaslonu prikazovalnika pojavi sporočilo.

WELCOME TO ISKRA FDOS V 1.Ø !

! -

Znak pomeni pripravljenost IZVRŠEVALCA (EXEC-a) za sprejem ukaza. Vpisovanje ukaza se lahko pričen že teden, ko traja nalaganje Izvrševalca (EXEC-a), toda vizuelna kontrola je omogočena šele po pričaku tega sporočila.

Vnos ukaza se vrši preko tastature prikazovalnika, vizuelna kontrola pa se ustvari na ta način, da se znak vsakega tipkanega znaka pojavi na zaslonu prikazovalnika, pri čemer se kazalec (kurzor) avtomatsko pomakne za eno mesto v desno.

V kolikor je potrebno popraviti vnešeni ukaz z uporabo tipke "←", vrnemo kazalec (kurzor) na mesto napačno vnešenega znaka in vnesemo pravilen znak. Tekst, ki se nahaja desno od tega mesta ne velja več in ga je potrebno ~~posodobiti~~ vneseti. Za ignoriranje vnešene ukazne linije in za iniciranje nove ukazne linije, pritisnemo na tipko "RUBOUT".

Linija ukaza ne sme zavzeti daljše dolžine kot je dolžina ene linije na zaslonu prikazovalnika (80 znakov).

Izdeja						Unit	Stren	J	K	Identifikacijska številka
St. obveznila						2				BEXEC01MD
	Iskra Elektromehanika PROGRAM IZVRŠEVALEC Kranj					Artik				Ramenska identifikacijska številka

Ukazi se po pravilu nanašajo na določene operacije z datotekami na disketah in bomo ob priliki objavne oblike (formata) in funkcije ukazov koristili naslednji zapis:

Ime vhodne datoteke v nekem ukazu mora biti enako z enim od imen datotek, ki se nahajajo na disketi. V razlagi oblike in funkcij ukazov bomo tako datoteko označevali z XXXX. Številko pogonskega mehanizma (enote) v kateri se nahaja disketa, označujemo z "u" in če to izpustimo se razume, da je to pogonski mehanizem "Ø". Ime izhodne datoteke v procesu ne sme istočasno biti enako imenu in številki pogonskega mehanizma v katerikoli od že obstoječih datotek. Izhodno datoteko bomo označevali z YYYY.

Iniciranje interpretacije izvršitve ukazne linije se vrši s pritiskom na tipko "CARRIAGE RETURN"; vsaka ukazna linija se mora končati s pritiskom na to tipko.

4.2 SPOROČILA NAPAK

Ukaz je lahko zavrnjen zaradi tega, ker ga FDOS ne bo mogel razumeti in zato ne bo vedel kač smo z ukazom zahtevali. Sporočila, ki se lahko pojavijo so:

FORMAT ERROR

Računalnik ukaza ne more prepoznati. Najpogostejša napaka je v dvočaju ali v številki neaktivnega pogonskega mehanizma.

NO SUCH FILE

Sistem ne najde zahtevane datoteke, ker datoteka s takim imenom ne obstaja. Zelo pogosto se pojavlja napaka v parametru "u", ki določa številko pogonskega mehanizma. Če parameter izpustimo, tedaj računalnik vzame pogonski mehanizem Ø, kar pa ni vedno res. Imena dveh datotek napisanih z velikimi in malimi črkami sta različna.

DUPL NAME

Z ukazom se zahteva kreiranje nove datoteke z imenom, ki je enako imenu že neke obstoječe datoteke na disketi. Najprej je potrebno, da zbrisemo staro datoteko, ali jo vpišemo na drugo disketo, ali pa spremenimo ime eni od teh dveh datotek.

NO ROOM

FDOS je osnovan, nameravana dolžina nove datoteke ne ustreza ostanku prostora na disketi. Na disketo lahko shramimo 253 datotek, ne glede na njihove dolžine.

Izdaja						Unit	Stren	J	K	Identifikacijska številka
Št. obvestila						3		BEXEC01MD		
	Iskra Elektromehanika PROGRAM IZVRŠEVALEC Krani					Arhiv				Namenska identifikacijska številka

V pogonskem mehanizmu nimamo vstavljenih disketa, vratca mehanizma so odprta in pogonski mehanizem dela z manjšo hitrostjo.

MEDIA ERROR:

Izhod z diskete je fizično neuporaben.

4.3 ŠTEVILKE NAPAK

Kadar sistem preneha z delom in se na zaslonu prikazovalnika pojavi vprašaj ali številka, tedaj je FDOS odkril med komuniciranjem (I/O) z gibkim diskom pomembno napako; po prikazanem sporočilu ~~zavrne~~ si sistem zaženemo z RESET.

?

Med nalaganjem strojne datoteke (za RESET, LOAD, RUN ali po zahtevi za izvršitev ukaza) dolžina datoteke ni enaka dolžini, ki je zapisana v imeniku.

1

Branje diskete ni mogoče.

2

Isti problem kakor z NO ROOM, s to razliko, da se pojavi za pravilnim startom izhodnega procesa. Podatki, ki so se že vpisali, lahko rešimo na ta način, da v kontrolno testnem programu (debugg-u) nastavimo programski števec na naslov EC24 in ga startamo. Kljub temu pa so vsi podatki, ki se nahajajo v spominskem vmesniku (ali pa še niso generirani) izgubljeni. Podobno se dogodi ob uporabi ukaza CREAT.

3

Podoben problem kot z DISK NOT READY, s to razliko, da se pojavi za pravilnim I/O startom gibkega diska. Rešitev problema je v ponovnem startu s pritiskom na tipko "ETX".

Izdaja	Stran	J	K	Identifikacijska številka
St. obvestila	4			BEXECO1MD
 Iskra Elektromehanika PROGRAM IZVRŠEVALEC Kranj	Avtor	Razmesta identifikacijska številka		

4.4 LISTANJE DATOTEK NA PRIKAZOVALNIKU

Za prikaz disketnega imenika uporabljamo:

LDIR : u

Informacija na zaslonu bo v kolonah in bo vsebovala: ime, atribut, začetno sled, začetni sektor in dolžino v sektorjih. Ta ukaz ne zadrži začetnih informacij seznama, če je ta daljši od snega zaslona (ekrana) ali za primerjanje dveh disket naenkrat. Za olajšavo iztiskamo imenik na papir in sicer tako, da uporabimo ukaz:

LDIR : u,L

Za pregled vsebine datotek brez spreminjanja uporabljamo:

VIEW,XXXX : u

VIEW,XXXX @ u,n,s

Parameter "n" je neobvezen in z njim določimo število vrstic na ekrani; največje dopustno število je 20. Z drugim zbirnim parametrom "s" določimo začetno vrstico znotraj datoteke. V primeru, da parameter "s" izpustimo, tedaj je začetni zapis prva vrstica.

Za pregled naslednjih n vrstic uporabimo tipko "LINE FEED". Če pa želimo pregledati predhodne n vrstice, tedaj uporabimo tipko "↑". Z uporabo tipke "←" se vrnemo na začetek datoteke. Na koncu datoteke ali kadar pritisnemo na tipko RESET, pa se konča operacija ukaza VIEW.

Izdeja	Št. obvestila	Let	Stevn	J	K	Identifikacijska številka
		5				BEXEC01MD
	skra Elektromehanika PROGRAM IZVRŠEVALEC Kranj		Avtor			Ravnost identifikacijske številke

4.5 NOVE DISKETE

Tovarniško svetje diskete in diskete s katerih smo zbrisali absolutno zbirko datotek lahko oblikujemo na dva načina, odvisno od tega ali želimo obliko (format) sistemsko ali (format) obliko uporabniške diskete.

Sistemski disketa vsebuje FDOS izvrševalce (EXECUTIVE), nato sledi imenik datotek, šele nato podatkovne datoteke. FDOS izvrševalec navzamči sledi in sicer so te zunaj meja 76-tih sledi, ki so na voljo za datoteke. Tako disketo lahko uporabimo na kateremkoli dostopnem pogonskem mehanizmu in jo lahko uporabljam za nalagalni (BOOT) proces.

Uporabniška disketa ne vsebuje FDOS izvrševalca in zato imamo na voljo za shranitev datotek 3 dodatne sledi. Zaradi tega, ker s sistemsko disketo naložimo program izvrševalce (EXECUTIVE), ki se nahaja na pogonskem mehanizmu 0, ampak je lahko na kateremkoli dostopnem pogonskem mehanizmu.

Izkušnje kažejo, da bo praviloma morala vsaka disketa biti sistemski izjemoma le tedaj ne, kadar potrebujemo 3 dodatne sledi. Kreiranje sistemski diskete se izvaja na ta način, da sistem pripravi točke, v kateri FDOS s klicajem zahteva nov ukaz. Premestimo disketo v pogonski mehanizem 0 in jo nadomestimo z novo, namenjeno za oblikovanje. (formiranje)

V pogonski mehanizem 1, namestimo disketo, ki vsebuje datoteko z imenom EXEC. Nato z naslednjim ukazom startamo oblikovanje (formiranje) diskete:

XGEN, EXEC:1

Po 30 sekundah pogonski mehanizem 0 vsebuje prazno sistemski disketo. Z uporabo MERGE ukaza vnesemo na disketo nekatere koristne datoteke (programe).

Za oblikovanje uporabniške diskete, namestimo novo disketo na pogonski mehanizem 1 in vnesemo FDOS ukaz:

INIT : 1

Indeks	Št. obveznika	Ime in priimek	Leto rojstva	Stevilo J	K	Identifikacijska številka
				6		BEXEC01MD
	Skra Elektromehanika PROGRAM IZVRŠEVALEC Kranj		Arhiv			Namenito identifikacijskih številk

Po 2 sekundah bo uporabniška disketa v pogonskem mehanizmu 1 prazna.

Ob ugotovitvi, da potrebujemo kreiranje uporabniške diskete na pogonskem mehanizmu 0, postopamo po enaki proceduri kot za XGEN. Namesto vnosa ukaza XGEN vnesemo:

INITX

Po dveh sekundah bo ne glede na to kakšen tip diskete je na pogonskem mehanizmu 0, sistem to spremenil v prazno uporabniško disketo.

Indeks	Št. obvestila	Identifikacijska številka
		7 BEXEC01MD
	Iskra Elektromehanika PROGRAM IZVRŠEVALEC Kranj	Arhiv Namesto identifikacijske številke

4.6 PREMESČANJE DATOTEK

Preslikavo datoteke iz ene diskete na drugo, izvajamo z uporabo ukaza:

MERGE, YYYY:u,XXXX:u

Nova datoteka (YYYY) je kombinacija starih (XXXX) in sicer je njihovo zaporedje določeno:

MERGE, YYYY, XXXXA:1, XXXXB:1, XXXXC:1, XXXXD:1

MERGE, YYYY:1, XXXXA, XXXXB, XXXXC

Za izdelavo popolnega posnetka (kopije) diskete, ne glede na tip ali vsebino, uporabimo ukaz:

COPY

Pri tem ukazu ni nikakršne izbire. Karkoli se nahaja na disketi, ki se nahaja v pogonskem mehanizmu 0 se presname (kopira) na disketo v pogonskem mehanizmu 1, s tem da je predhodno brisana disketa na enoti 1. Če se na disketi nahajajo slabe sledi, se bo kopiralo šele po 5 poizkusih branja glede na FDOS merila (standarde).

COPY proces traja približno 5 minut. Če prekinemo izvajanje COPY ukaza z izklopitivijo pogonskega mehanizma ali z izklopitivijo centralne enote sistema, tedaj vsebina na disketi 1 ne bo zanesljiva. Po končanem kopiranju se z ukazom LDIR:1 prepričamo, če je bilo kopiranje uspešno.

Izdeja							Lst	Strm	J	K	Identifikacijska številka
Št. obveznila							8				BEXEC01MD
	Skra Elektromehanika PROGRAM IZVRŠEVALEC Kranj						Arhiv				Rameno identifikacijska številka

4.7 SPREMINJANJE IMEN IN ATRIBUTOV

Datoteko preimenujemo tako, da določimo staro in novo ime in vnesemo ukaz:

RENAME,XXXX:u,YYYY:u

Atribute imenika spremojamo:

CHGAT,XXXX:u,n

Pri tem n lahko zavzame naslednje vrednosti:

00 nezavarovana datoteka

01 stalna datoteka (permanent) zavarovana pred DELETE ukazom

80 prazen prostor za pakiranje med naslednjim izvajanjem DELETE ukaza

Atribut 80 ne more biti nasproten. Moč tega atributa je enaka realnemu DELETE ukazu, s to razliko, da je operacija pakiranja izpuščena.

Izmišljano datoteko kreiramo z določitvijo imena in dolžine:

CREATE,YYYY:u,n

Ena od uporab CREATE ukaza je reševanje podatkov nezapuste datoteke. Torej podatkov, ki niso kriti z vnosom imena v imeniku. Ob uspešno izvršenem ukazu CREATE, bodo podatki nedotaknjeni in njihovo ime vneseno v imenik, nato pa jih lahko uporabljamo za pregled in spremenjanje podatkov z VIEW in EDIT programom.

Parameter n je šestnajstico število sektorjev, ki določuje dolžino datoteke. Datotečni atribut mora vedno biti pri tem 00, v nasprotnem primeru ga moramo spremeniti s CHGAT ukazom.

izdeja						list	stran	J	K	Identifikacijska številka
Št. obvestila						9				BEXEC01MD
	Skra Elektromehanika Krani	PROGRAM IZVRŠEVALEC				Arhiv				Namensko identifikacijsko številka

4.8 BRISANJE IN PAKIRANJE DATOTEK

Datoteke, ki se nahajajo na disketi v enem od pogonskih mehanizmov, lahko brišemo z naslednjim ukazom:

DELETE:u,XXXXA,XXXXB,XXXXC,XXXXD,itd.

DELETE:u,XXXX

DELETE: XXXX

Ob izbori enega od ukazov DELETE, brez da bi določili pogonski mehanizem, sistem avtomatsko izbere pogonski mehanizem 0. Pri tem moramo biti pazljivi, ker je na tem pogonskem mehanizmu pogostokrat sistem-ska disketa.

S pritiskom na tipko RETURN startamo ukaz in nato se izpiše na zaslonu prikazovalnika seznam vseh brisanih datotek.

FDOS avtomatično pakira disketo, to je, premakne vse datoteke, ki so ostale po brisanju (kreiranju praznin) pod izdelanimi prazninami. Čas izvajanja traja od 1 sekunde do 10 minut. Pri izvajanju ukaza DELETE moramo biti potrpežljivi, ker bi v primeru, da izključimo sistem leta izbrisal celotno vsebino diskete.

Zahtega po brisanju datoteke z atributom 01 (stalna) bo ignorirana in se bo ob imenu datoteke izpisal komentar PERMANENT.

Za brisanje katerikoli datoteke ne glede na atribut, najprej uporabimo ukaz CHGAT za postavitev teh atributov na 00. Šele nato sledi ukaz DELETE. Drugi način pa je ta, da postavimo attribute na vrednost 80, ki z uporabo ukaza ~~DELETE~~ avtomatsko izbriše imena datotek in avtomatsko povzroči datoteke praznin, ki bodo izvedene med naslednjim pakiranjem.

Vse datoteke na uporabniški disketi lahko zbrisemo s preprostim ukazom INIT.

Indeks	Št. obveznika	Ust.	Vnos	J	K	Identifikacijska številka
				10		BEXEC01MD
	Iskra Elektromehanika PROGRAM IZVRŠEVALEC Kranj	Atrib				Nemotna identifikacijska številka

4.9 TISKANJE DATOTEK IN INFORMACIJ NA Matričnem tiskalniku

Vedno ne zadostuje le datoteka shranjena na disketi, temveč potrebujemo datoteko iztiskano na papirju, katera nam služi za dokumentiranje ali za boljši pregled. V takih primerih uporabljamo naslednji ukaz:

PRINT,XXXX:u,L

Če izpustimo neobvezen parameter L, tedaj sledi tiskanje datoteke na zaslonu prikazovalnika in je rezultat podoben kot pri uporabi ukaza VIEW.

Za tiskanje disketnega imenika uporabljamo ukaz:

LDIR:u,L

4.10 NALAGANJE PROGRAMOV BREZ IZVAJANJA

Za nalaganje programa v spomin uporabljamo ukaz:

LOAD,PROG:0

LOAD,PROG, premik (offset)

Izbirno šestnajstico vrednsot premika (offset-a) moramo uporabljati pri nalaganju podatkovnih tabel ali pri nalaganju samopremakljivih (relokativnih) programov. Vrednost premika je dodana lokacijskim naslovom polj zapisanega programa v strojenem zapisu med izvajanjem nalaganja. Učinek negativnega premika (offset-a) pri uporabi visoke vrednsoti, če se približa 65K, je ta, da vrne vrednost na 0. Na koncu nalagalnega procesa je startan testno kontrolni program (DEBUG). Le-ta pa zahteva, da je prikazovalnik v PAGE stanju.

izdeka						list	stran	J	K	Identifikacijska številka
št. obveznika						11				'BEXEC01MD'
	Skra Elektromehanika PROGRAM IZVRŠEVALEC Kranj					Arhiv				Namesto identifikacijske številke

4.11 STARTANJE PROGRAMA

Za start izvrševanja programa preprosto vnesemo njegovo ime (če je potrebno tudi številko pogonskega mehanizma), podobno kot pri vnosu FDOS ukazov:

```
LDIR:u  
EDIT,,YYYY  
ASMB,XXXX,YYYY,n
```

Če iz kakršnegakoli vzroka izdelamo program z natančno istim imenom kot ga ima katerikoli FDOS ukaz, tedaj za start izvrševanja uporabljamo RUN ukaz, pri čemer je prvi parameter ime programa.

```
RUN,LDIR:u  
RUN;MERGE,YYYY,XXXX:u
```

V obeh primerih bo FDOS naložil program katerikoli določene disketne datoteke in prevedel neobvezni šestnajstiški parameter v binarnega.

Vhodni kazalci za XXXX bodo postavljeni v polja FDOS-a: ISIZE,ITRK,ISCTR,ICNTR. Če vhodna datoteka ni bila določena, tedaj se uporabljene vrednosti 1,0,0,0.

Polja, ki se postavijo za izhodne kazalce YYYY so OSIZE,OTRK,OSCTR in OCNTR. V primeru, da izhodna datoteka ni bila določena, tedaj je polje OSIZE enako 1, druga polja pa bodo kazala na naslednji prosti sektor na disketi.

Parameter pogonskega mehanizma(ali izpuščena vrednost 0) se hrani v lokacijo PASS.

FDOS nato preide v kontrolo programa na absolutni lokaciji 0100.

Izdaš	Št obvestila	Identifikacijska številka
		12 BEXEC01MD
Arhiv	Namensko identifikacijsko številko	
Iskra Elektromehanika Kranj		

ZA OSVEŽITEV PONOVNO NAVAJAMO VSE UKAZE:

CHGAT,XXXX:u, atribut
CREATE,YYYY:u dolžina v sektorjih
DELETE:u,XXXXA,XXXXB; itd.
INIT:u
LDIR:u
LOAD,PROGRAM:u, premik (offset)
MERGE,YYYY:u,XXXXA:u,XXXXB:u, itd.
PRINT,XXXX:u, L
RENAME,XXXX:u, YYYY
RUN,PROGRAM:u,XXXX:u; YYYY:u, parameter
VIEW,XXXX:u, število vrstic zaslona, začetna
vrstica

Če je parameter pogonskega mehanizma izpuščen (:0 tali :1 ali :2 ali :3), tedaj se uporablja pogonski mehanizem 0.

Vsek ukaz mora biti zaključen s tipko RETURN.

Izdaja	St. obveznila	Let	Stran	J	K	Identifikacijska številka
		13				BEXECD1MD
	Iskra Elektromehanika PROGRAM IZVRŠEVALEC Kranj	Avtor				Naravnost identifikacijske številke

THERMODYNAMIC PROPERTIES OF POLY(1,3-PHENYLENE TEREPHTHALIC ACID)

	Primer in ime	Podpis	Gradivo	Odstopni metri, mjer	Topljena obdelava	Površ. zascita	Pripadnost
Konstr.			Izjava				101680
Projekt			Znak				
Pregled			St. obv.				
Števil			Datum				
Stand			Podpis				
Naziv				Iskra Elektromehanika Kranj 1.3			
PROGRAM FDOS MONITOR				List	Stran	K	Identifikacijska številka
				1			V46325044
Namembrovanost kopije				Ažurir	Menilo	Sekcija	Namesto identifikacijske številke

5 MONITOR

Za izvrševanje programov v sistemu 1680 obstaja na razpolago velik adresni prostor (max 52 K), od adrese 0 do DFFF (Hex). Preostali adresni prostor, začenši od E000 (Hex) je rezerviran za sistemske programe, podatke in vhodno/izhodne enote.

Poznavanje teh " magičnih " adres je često neobhodno za razvoj programov, za katere je potrebno, da se izvršujejo na sistemu 1680, za razliko od programov, ki se bodo izvrševali na nekem drugem mikrorračunalniškem sistemu.

GLAVNI DODELJENI ADRESNI PROSTOR

0000 - DFFF	Naslovni prostor, ki je na voljo uporabniku
E000 - FFFF	1680 operacijski sistemski ROM (EC00 - EFFF) je rezerviran za rezidenčni rokovalevnik (handler) gibkega diska
F800 - FBFF	Rezervirane lokacije vhodno/izhodne enote
FC00 - FFFD	1680 operacijski sistemski RAM
FFFE - FFFF	Strojno na E000

APARATURNE

NASLOVI ELEKTRONSKE OPREME (HARDWARE - a)

Izhod na printer : ACIA

FBCC Kontrolni register

FBCD Podatkovni register

Vhod s tastature in izhod na prikazovalnik : ACIA

FBCE Kontrolni register

FBCF Podatkovni register

Izhod/vhod na gibki disk : PIA

FBDO Vhodni podatkovno statusni register

FBD1 Vhodni kontrolno statusni register

Izdaja						List	Stran	J	K	Identifikacijska številka
Št. obvestila						2				V46325044
	Iskra Elektromehanika MONITOR FDOS Krani					Arih				Namensko identifikacijska številka

FBD2 Izhodni komandno podatkovni register
 FBD3 Izhodni komandno kontrolni register
 FBD4 Naslov je še neizkoriščen
 FBD5 Naslov je še neizkoriščen
 FBD6 Izhodni podatkovni register
 FBD7 Izhodni kontrolni register

FBFA, FBFB Register strojne prekinitvene točke kontrolno testnega programa

E000 RESET Naslov rutine za ponovni start in zagon
 FFF8-FFF9 IRQVEC Naslov vektorja za maskirane prekinitve
 FFFA-FFFB SWIVEC Naslov vektorja za programske prekinitve
 FFFC-FFFD NMIVEC Naslov vektorja za nemaskirane prekinitve
 FFFE-FFFF Naslov RESET vektorja (vektorja za ponovni start Strojno na E000).

SISTEM PROGRAMSKIH NASLOVOV (SOFTWARE ADDRESSES)

Naslednji naslovi (adrese) ustrezajo programskim podrutinam in parametrom za ISKRADATA 1680; Parametri so opisani na naslednji način :

Ime : tip
pri tem je tip lahko

zlog	8 bitni zlog
znak	ASCII znak
bočl	bulova (logična) vrednost nepravilno če = Ø pravilno če ≠ Ø
dvojni zlog	16 - bitna beseda
naslov	16 - bitna naslov (adresa)

Podrutine so klicane z JSR, in spremenijo vsebine vseh registrov razen, če je drugače določeno. Parametri so opisani pod rutino za katero se uporabljamjo.

Znak # pred imenom rutine označuje uporabno rutino, ki se zelo enostavno uporablja ter je zelo težko narediti napako pri uporabi.

Izdaja						List	Stran	J	K	Identifikacijska številka
Št. obvestila						3				V46325044
	Iskra Elektromehanika MONITOR FDOS Kranj					Arhiv				Namensko identifikacijska številka

Ostale rutine so vključene zaradi opisov namena.

V glavnem sistemski programi (software) uporabljajo RAM spomin od FC00 do FFFD za začasna shranjevanje, za statusne informacije in za obdelavo parametrov. Uporabnik ne sme uporabljati lokacij, ki so rezervirane za parametre. Nekontrolirane spremembe v teh lokacijah lahko povzročijo nepredvidene rezultate.

V. HOUSKO/T.ZHODNE BUTINE

Vhodno/izhodne rutine ohramijo B in X register. Izhodne rutine vrnejo znak iz A registra in vsebino A registra tudi ohramijo. Vhodne rutine vnesajo spremeniti znak v A register. Če vhodno/izhodne rutine ne morejo biti izvršene, je vrnilitev iz teh z postavitvijo prenosnega (carry) bita na 1. Ta ustreza izgubi nosilca podatkov v ACIA, kjer so datoteke na disketti. Če ustrezana zahteva po vhodno/izhodni rutini ne more biti trenutno izvršena, (na primer: izhodni vmesni register je poln ali vhodni vmesni register je prazen) tedaj bodo vhodno/izhodne rutine ţakale.

Vnos s tastature je izvršen z uporabo prekinitev. KBDIN ne more biti uporabljen z maskiranimi prekinitvami. Če uporabnik želi klicati DEFOUT ali KYSTAT z maskiranimi prekinitvami mora isdelati

INC CLEVEL

SEI

za onemogočanje (maskiranje) prekinitev

DEC CLEVEL.

BPL : 1

CLJ

EQU 指令

za omogočanje prekinitve, sicer pa prekinitve morajo biti omogočene z DFYOUT ali KYSTAT. DEBIN bo operiral z postavljenimi prekinitvami.

NASLOV	IME	FUNKCIJA
E003	*KBDDIN	Sprejem znaka z tastature
E006	*DPFFOUT	Prikaz znaka na zaslonu prikazovalnika
E009	*RDRTN	Vhod čitalca (<u>ni izkoriščen</u>) (Enota čitalca je normalno komunikacijska linija ACIA-e)
E00G	*VRDOUT	Izhod tiskalnika (<u>ni izkoriščen</u>) (enota tiskalnika je normalno komunikacijska linija ACIA-e)

Ldt	Stran	J	K	Mentovácká čísla
4				V46325044
Ačk				Neplatí identifikací čísla

E00F ~~m~~DSKIN Branje znaka z diskete
 E012 DSKOUT Vpis znaka na disketo
 E015 ~~m~~PRINT Izpis znaka na matričnem tiskalniku
 E018 DEBIN Sprejem znaka s tastature v kontrolno testni
 program in makro razširitev
 EC24 XCLOSE Zapiranje vsake izhodne datoteke in ponovno na-
 laganje FDOS-a.
 FC5F DKSTR:nas-
 lov Naslov makro niza za DEBIN
 FC61 DKLEN:
 zlog Dolžina makro niza. Če je DKLEN=0, bo DEBIN
 sprejel znak z tastature. V primeru, da so pre-
 kinitve maskirane, bo DEBIN sprejel znak direk-
 tno z tastature. Sicer pa bo uporabljen rutina
 KBDIN.
 EO51 ~~m~~KYSTAT Stanje tastature. Vrnjeni A<0, pomeni, da s
 tastature ni poslan noben znak.

VHODNE TOČKE KONTROLNO TESTNEGA PROGRAMA

EO1B TDEBUG Brezpogojen prenos kontrole kontrolno testnemu
 programu.
 EO1E ~~m~~BRKPT Vstop v kontrolno testni program zaradi prekin-
 vene točke.
 EO21 ~~m~~SNAP Vstop v kontrolno testni program za prikaz trenut-
 nega stanja programa. Kontrola se neposredno vra-
 ča.
 EO24 ERROR Vstop v kontrolni program na neko napako. X mora
 biti postavljen na niz zaključen z NULL(0),
 ki se bo pojavila v E vrsti.
 EO27 LOG Nov zapis v zapisu dogajanja, ki se prikaže v L
 vrsti. B bo prikazan kot ASCII znak, sledila pa
 mu bo heksadecimalna vrednost A-ja. Najnovejši
 vnos bo na levi strani, ohrani & in B.

Izdaš	St. obrestila	Identifikacijska številka
		5 V46325044
	Iskra Elektromehanika MONITOR FDOS Kranj	Namensko identifikacijska številka

PREVAJALNE RUTINE

Vse rutine za prevajanje ohranljajo X registre in vse se vrnejo z postavljenim prenosnim bitom (carry) v primeru preiskusa nepravilnega prevoda.

E03F	CA2U	Pretvorba ASCII znaka male črke v A v ASCII znak velike črke (B). Če A ni črka, se vrne, B=A tedaj pa se počavi "overflow" bit.
E042	CA2H	Pretvorba ASCII heksa števila v A v binarni ekvivalent v B
E045	CH2B	Pretvorba dveh ASCII heksadecimalnih znakov v A in B (višji v A) v binarni ekvivalent v A.
E048	CA2D	Pretvorba ASCII decimalnega števila v A v binarnega v B
E04B	CB2H	Pretvorba binarnega v A v dva ASCII heksadecimalna števila v A (višji) in B
E04E	CH2A	Pretvorba binarnega števila v A, v mejah od 0 do F v ASCII heksadecimalno število v A.

TABELA PREGLEDA UPORABE

E054 FIND Iskanje po tabeli, ki jo kaže X za zlog enak A.
Tabela ima obliko:

Zlog	N
Zlog	Vnos 1
Beseda	Vrednost 1
Zlog	Vnos 2
Beseda	Vrednost 2
.	
.	
.	
Zlog	Vnos N
Beseda	Vrednost M
Beseda	Vrednost N+1

Izjava						List	Stran	J	K	Identifikacijska številka
Št. doverstila						6				V46325044
	Iskra Elektromehanika MONITOR FDOS Kranj					Arhiv				Namenska identifikacijska številka

Vrnitev, X kaže na vrednost I, pri tem je vnos I=A ali vrednost N+1, če A ni najden. Vrnitev B=(N-1)+1, če je A najden. V nasprotnem je B=0.

E057 FINDB Iskanje po tabeli tro-zlogovnih vhodov, na vhodu čigar je prvi slog enak A li B. Vrne X -- drugi slog najdenega vhoda. Izvod iz rutine je nepredviden, če niti eden vhod ne vsebuje niti A, niti B.

E05A CASE Klic: JSR CASE
Zlog N
Zlog Vnos 1
Beseda Naslov 1
.
.
.
Zlog Vnos N
Beseda Naslov N

Iskanje po tabeli sledi JSR-nu zaradi takega vnosa I, da je Vnos I=A. Kontrola je nato prenešena na Naslov I. Če ne najde takega I-ja, preide kontrola na instrukcijo, ki sledi Naslovu N. Skladovni kazalec je v času prenosa kontrole enak skladovnemu kazalcu, ki je bil pred izvršitvijo JSR.

DRUGE MOŽNE UPORABE

E075 MXPSHX Shrani X register v sklad. ohrani A,B,X.

MONITORSKI PARAMETRI

FC2D CLEVEL:slog Sekcija kritičnega nivoja. Monitorske rutine ki morajo teči z maskiranimi prekinitvami nastoejo pa joc

INC CLEVEL

SEI

na vhodu,in

Indaja						List	Stran	J	K	Identifikacijska številka
Št. obvestila						7				V46325044
	Iskra Elektromehanika MONITOR FDOS Kranj					Arhiv				Nomerica identifikacijske številke

DEC CLEVEL

BPL :1

CLI

:1 EQU #

Na ishodu. Če uporabnik želi, da njegov program teče z maskiranimi prekinitvami mora izvršiti :

INC CLEVEL

SEI

FC30 FNKEYS :naslov Naslov tabele, ki ima obliko :

Zlog tipka 1

Beseda Naslov I

.

.

Zlog 0

Beseda Z

Kjer je naslov I naslov rutine, kadar je tipka 1 pritisknjena. Z ali 0 ali pa naslov neke druge tabele.

FC32 IOPORT:naslov Naslov tabele vhodne/ishodnih rutin

FC47 RC:sleg C register shranjen na RESET

FC48 RB:sleg B register shranjen na RESET

FC49 RA:sleg A register shranjen na RESET

FC4A RX:dvojni sleg X register shranjen na RESET

FC4C RSP:dvojni sleg SP register shranjen na RESET

FC4E RIRQ:dvoj.sleg IRQVEC shranjen na RESET

Izjava						List	Stran	J	K	Identifikacijska številka
Št obvestila						8				V46325044
 Skra Elektromehanika MONITOR FDOS Kranj						Arhiv				Namensko identifikacijska številka

FC50 RSWI: dvejni slog
 SWIVEC shranjen na RESET
 FC52 RMWI: dvejni slog
 RMWVEC shranjen na reset

PARAMETRI KONTROLNO TESTNEGA PROGRAMA (DEBUGGER - ja)

FC57 DKSTR:naslov
 Naslov makre niza za branje z DEBIN
 FC61 DKLEN: slog
 Dolžina niza namenjena za DKSTR. Če 0, tedaj DEBIN
 prejme svoj vhod s tastature.

SISTEM ZAČASNIH LOKACIJ

Med prekinitvami so te začasne lokacije shranjene v sklad. Uporabljajo se v sistemu za operacije prenosa vsebine registrov. Na voljo so uporabniku, teda lahko se popolnoma spremenjene (vsebině teh lokacij) med pozivanjem katerokoli sistemsko rutine.

FC28 TA:slog Začasna lokacija namenjena za register A.
 FC29 TB:slog Začasna lokacija namenjena za register B.
 FC2A TX:slog Začasna lokacija namenjena za indeks register.

PARAMETRI FDOS II

FC00 PASS:slog Informacija prehoda sbirnika in parameter RUN
 ukaza.
 FC01 OFILE: slog Številka izhodne datoteke.
 FC02 OUNIT: slog Enota izhodne datoteke.
 FC03 IUNIT: slog Enota vhodne datoteke.
 FC04 ISIZE: dvejni
 slog Dolžina vhodne datoteke v sektorjih.
 FC06 ITRK: slog Sled vhodne datoteke.
 FCD7 ISCTR: slog Sektor vhodne datoteke.
 FC08 ICNTR: slog Števec vmesnikov vhodne.datoteke.
 FC09 OSIZE: dvejni
 slog Delžina izhodne datoteke v sektorjih.

Izdelava						Ust	Stres	J	K	Identifikacijska številka
St. obveznila								9		V46325044
	Iskra Elektromehanika MONITOR FDOS Kranj					Avtor				Namenski identifikacijski številki

PCOB	OTRK :sleg	Sledi ishodne datoteke.
POOC	OSCTR :sleg	Sektor ishodne datoteke.
POOD	OCNTR :sleg	Števec vmesnika ishodne datoteke.
POOE	TITRK :sleg	Začetna sled vhodne datoteke.
POOF	TISIZE:čvoj.sleg	Začetna dolžina vhodne datoteke.

TABELA VHODNO/IZHODNIH VRAT IN MOŽNOSTI UPORABE

Standardna garnitura vhodno/izhodnih rutin (KBDIN, DPYOUT, itd.) zadostuje za najboljše namene, teda ne omogoča :

1. Testiranje stanja enote.
2. Izvršitve kontrolnih funkcij.
3. Preusmerjanje vhodno/izhodne aktivnosti na alternativne enote.

Za ta namen so pripravljene prilagodljive vhodno/izhodne raktine, h katerim se pristopa indirektno preko tabele, imenovane Tabele vhodno/izhodnih vrat.

K tej tabeli se ponovno pristopa indirektno preko sistemske variable IOPORT. Na ta način je omogočeno, da bo prvotna tabela nameščena v ROM spomin.

Vhodna tabela je niz naslovov vhodno/izhodnih programov. Premiki (offsets) (številke vrat) vhodno/izhodnih programov so v tabeli v naslednjem zaporedju :

<u>ENOTA</u>	<u>VRATA</u>	<u>UPORABLJEN Z</u>
TASTATURA	0	KBDIN
ZASLON PRIKAZOVALNIKA	2	DPYOUT
VHODNA KOMUNIKACIJA Z ACIA-o	4	RDRIN (ne iskoriščena)
IZHODNA KOMUNIKACIJA Z ACIA-e	6	VRTOUT (")
BRANJE ZLOGA Z DISKETE	8	DSKIN
VPIS ZLOGA NA DISKETO	10	DSKOUT
TISK ZNAKA NA Matričnem tiskalniku	12	PRINT
Tastatura kontrolno testnega programa (DEBUGGER-ja)	14	DEBIN

Indeks						Let	Stran	J	K	Identifikacijska številka
St. obvestila						10				V46325044
Arhiv										Namesto identifikacijske številke
	skra Elektromehanika Kranj	MONITOR FDOS								

Vsaki od zgoraj navedeni vhodno/izhodnih programov so različni, vendar im izhod ACIA-je sta programirana z različnimi rutinami. Vhodno/izhodni programi imajo tri vhodne tečke. Če je naslov vhodno/izhodnega programa HADDR, potem:

- | | |
|---------|--------------------------------|
| HADDR+0 | Vhodna tečka standardne rutine |
| HADDR+2 | Vhodna tečka statusne rutine |
| HADDR+4 | Vhodna tečka kontrole rutine |

Standardna vhodna rutina prinese slog v register A.

Standardna izhodna rutina povzroči, da se slog, ki se nahaja v registru A prenese na izhod, pri čemer se vsebina registra A ne spremeni. Vsi ostali registri se ohranjeni. Kakorkoli pa standardne vhodno/izhodne rutine ohranijo vsebino registra B in registra X.

Standardne rutine vrnejo pestavljen bit prenosa (carry) na koncu datoteke. Statusne rutine vrnejo naslednje infirmacijo:

- | | |
|-------|--|
| BIT 7 | Enota je zasedena. Vhodno/izhodni vhod bo čakal. |
| BIT 2 | Napaka paritete. (potek ni uporabljen). |
| BIT 1 | Prehitevanje (znak izgubljen). |
| BIT 0 | Konec datoteke ali izgubljen mesilec na komunikacijski liniji. |

TABELA : ENOTE IN NINIJOVA STANJA

<u>ENOTA</u>	<u>ZAPIS STANJA</u>	
TASTATURA	BUSY	Še ni bile sprejetih znakov.
	OVERRUN	Znak sprejet v času, ko je vmesni register poln.
VHODNA KOMUNIKACIJSKA	BUSY	Še ni bilo sprejetih znakov, ali pa je EOF.
	OVERRUN	Znak sprejet v času, ko je vmesni register poln.
	EOF	Mesilec je izgubljen.

IZHODNA KOMUNIKACIJA	BUSY	Vmesni register je poln ali EOF.
Z DISKETE	EOF	Nosilec je izgubljen.
	BUSY	Konec datoteke.
NA DISKETO	EOF	Konec datoteke.
	EOF	Vedno postavljen.

Kontrolni vhod vnedne/izhodnega programa opravlja kontrolne funkcije, odvisne od kontrolnega zapisa v A registru. Ne vsebujejo vsi vnedne/izhodni programi vseh navedenih kontrolnih funkcij:

- 0 Inicializacija enote. Brisanje vseh vmesnih registrov, čiščenje zaslona prikazovalnika.
- 1 Prevrtitev (Rewind).
- 2 Prekinitev (Break).
- 3 Konec datoteke (Zapiranje).

TABELA : ENOTE IN VNEDNA KONTROLA

<u>ENOTA</u>	<u>KONTR.KODA</u>	<u>POMEN</u>
Tastatura	INIT	Brisanje vmesnega registra, inicializacija ACIA-e.
Zaslon prikazovalnika	INIT	Brisanje zaslona.
Vhodna komunikacija	INIT	Brisanje vmesnega registra, inicializacija ACIA-e.
Izhodna komunikacija	INIT	Brisanje vmesnega registra , inicializacija ACIA-e.
	BREAK	Prenos prekinitev.
Z diskete	INIT	(Ne deluje. Inicializiran z RESET).
	REWIND	Postavi na začetek datoteke in sačne s branjem na ponovnem začetku.
Na disketo	INIT	(Enake kot pri Z disketu).
	EOF	Konec datoteke in prenos podatkov v imenik ter vrnilje kontrole FDOS.
Debin	INIT	Brisanje makro niza.

Indija	St. obvestila	Skra Elektromehanika Kranj	MONITOR FDOS	Unit	Stran	J	K	Identifikacijska številka
				12				V46325044

V primeru, da kontrelna koda ni bila preposnana kontrolna rutina vrne postavljen prenosni (carry) bit.

Primer: za delovanje stanja vhoda ACIA-e.

LDX	IOPORT	X → vhodna tabela
LDX	4, X	X → komunikacijska linija vhodnega prog.
JSR	2, X	Elie stanja vhoda
TST	A	Testiranje, če je enota zasedena.
BMI	ZASEDNA	Zasedena
.		
.		
.		
Enota ni zasedena.		

Zaseden v tem primeru upomni, da se za zadnjim pozivom RDRIN, ni sprejet noben znak).

PROGRAMIRANJE VIDEO PRIKAZOVALNIKA S TASTATURO

Video prikazovalnik sprejme kontrelne kode, ki so prikazane v tabeli 5.1.

Kontrelni znaki so lahke poslani iz uporabnikovega programa do zasedenega prikazovalnika s DPYOUT rutino.

X - Y posicirvanje znakov (po ESC, 'l zaporedju) je vedno učinkovito le tedaj, če je prikazovalnik v PAGE načinu.

Index						List	Stran	J	K	Identifikacijska številka
Št. obvestila						13				V46325044
	Iskra Elektromehanika Kranj		MONITOR FDOS			Arhiv				Namenska identifikacijska številka

TABELA 6.1. KONTROLNE KODE PRIKAZOVALNIKA S TASTATURO

ASCII MNEMONIK	HEKSA ZAPIS	FUNKCIJA
BEL	07	250 ms dolg zvočni alarm.
BS	08	Vrnitev kazalca za eno mesto nazaj brez sprememb prikazanih podatkov.
CR	0D	Premik kazalca na začetek tekče vrstice ali naslednje vrstice, odvisno od položaja notranjega "AUTO LINE FEED" stikala.
LF	0A	Premik kazalca za 1 vrstice navzdol od sedanega položaja brez sprememb prikazanih podatkov (PAGE način) ali premik saslonca podatkov za vrstice navzgor v SCROLL načinu.
NAK	15	Premik kazalca sa 1 mesto naprej brez sprememb prikazanih podatkov.
SYN	16	Brisanje vrstic v kateri je kazalec in sicer ed mesta kazalca do konca vrstic, kazalec ostane na istem mestu.
CAN	18	Brisanje saslonca prikazovalnika in ponevna postavitev kazalca na izhodiščno mesto.
EM	19	RESET kazalec na začetne mesto brez sprememb prikazanih podatkov.
ESC	1B	Sledi nesmislen znak ali sledi X - Y pozicioniranje znakov, odvisno od položaja notranjega stikala za X - Y pozicijemiranje.
SO	0E	Pomakne kazalec za 1 mesto naprej in vpelje polovični ton polj.
SI	0F	Določa, da bodo polja desne od kazalca do konca vrstic prikazana normalno (prekine prikaz polj z polovično itensiteto tona ali prekine utripajoča polja). Pomakne kazalec za eno polje .
ETB	17	Pomakne kazalec za 1 mesto naprej in vpelje utripajoča polja.

Izdaš						List	Stran	J	K	Identifikacijska številka
St. obvestila						14				V46325044
	Skra Elektromehanika MONITOR FDOS Kranj					Arhiv				Namensko identifikacijska številka

FDOS - REZIDENČNI MODUL

Naslednje informacije so prikazane z namenom prikazati uporabo sredstev naprednih sistemov baziranih na ISKRADATA 1680.

DISKEVNI VHOD/IZHOD

Pri FDOS rezidenčnem modelu je postavljen pogoj, omogočanje programerju razvijati uporabniško orientirane programe, ki koristijo disketo kot periferni spominski medij izven FDOS-a. Znotraj FDOS modula se nahajata rutina za branje z diskete DSKIN in rutina za vpisovanje na disketo DSKOUT, ki zagotavlja zlogovno orientiran izhod in vhod.

Vrsta uporabe vhodnih in izhodnih rutin DSKIN in DSKOUT je naslednja. Na prvem mestu je ureditev kazalcev dostopnega prostora na disketi. To je poznano kot edpiranje disketnih datotek. Brž ko bo disketna datoteka odprta, lahko kadarkoli kličemo rutini DSKIN in DSKOUT, podobne kot bi uporabnik klical v mikroričunalniškem kentrelne testnem programu ali v monitorskem programu rutine za konzolni izhod. Program za krmiljenje pogonskega mehanizma nudi celotno podporo datotečnim kazalecem po odprtju datoteke. V nekem trenutku odpremo le ene vhodne in ene izhodnede datoteko. Odpiranje vhodni in izhodnih datotek za potrebe nekega programa, lahko opravimo s pomočjo FDOS Izvrševalca (HEEC-a), z koriščenjem ukaza RUN.

V kolikor je potrebne da imamo več vhodnih in izhodnih datotek istočasne odprtih, te ne moremo izvršiti z enostavnim koriščenjem sistemskih programov, temveč moramo sami izdelati v našem programu.

Uporablja se naslednje RAM spominske lokacije:

<u>LOKACIJA</u>		<u>OPIŠ</u>
ISIZE	FC04	Dolžina vhodne datoteke v sektorjih (2zloga)
ITRK	FC06	Naslov začetne sledi vhodne datoteke.
ISCTR	FC07	Naslov začetnega sektorja vhodne datoteke in pogonska enota.
ICNTR	FC08	Števec bralnega vmesnega registra.
OSIZE	FC09	Dolžina izhodne datoteke v sektorjih (2zloga)
OTRK	FC0B	Naslov začetne sledi izhodne datoteke.
OSCTR	FC0C	Naslov začetnega sektorja izhodne datoteke in pogonska enota.

Izdaja						List	Stran	J	K	Identifikacijska številka
Št. obvestila						15				V46325044
	Iskra Elektromehanika MONITOR FDOS Kranj					Arhiv		Namesto identifikacijske številke		

OCNTR	FOOD	Števec nadsornika (controller-ja) pisalnega vmesnega registra.
IUNIT	FOO3	Enota pogonskega mehanizma vhodne datoteke.
OUNIT	FOO2	Enota pogonskega mehanizma izhodne datoteke.

DISKETNI IMENIK

Mesto disketnega imenika

Vsaka disketa vsebuje imenik datotek, ki se nahaja na njej. Imenik je shranjen na sledi 0, na sekterjih 4 do 26. Sektorji 1 do 3 so rezervirani za razširitev FDOS-a. Imenik se začne na 4 sektorju, vsak sekter vsebuje 11 datotečnih kontrolnih vhodnih blokov (FCB-File Control Block) vsak FCB pa je 11 zlogov dolg. Disketni imenik lahko sprejme 253 unikatnih datotečnih imen.

Sled 0 sekter 1 - rezerviran

2 - rezerviran

3 - rezerviran

4 - datotečni kontrolni blok (FCB) 1 do 11

5 - FCB 12 do 22

.

.

.

26 - FCB 243 do 253

Vsebina imenika:

Zahtevane informacije o datotekah so shranjene v imeniku datotek. Določene informacije o datoteki so vsebovane znotraj teh datotečnih 11 zlogovnih kontrolnih blokov (FCB). Vsebovana informacija znotraj datotečnih FCB-jev vključuje ime datoteke, datotečne atribute, začetne sledi, začetni sekter in dolžino datoteke v sekterjih. 11 zlogov dolga informacija izgleda:

ZLOGI 1-5 ime datoteke dopolnjeno z prazninami
(heksa 20)

ZLOG 6 datotečni atributi

ZLOG 7 naslev začetne sledi datoteke

ZLOG 8 naslev začetnega sekterja datoteke

ZLOGA 9-10 dolžina datoteke v sekterjih +1

ZLOG 11 (reserviran za kasnejšo razširitev kontrolnega bloka)

Tedaja						list	Stran	J	K	Identifikacijska številka
St. obvestila						16				V46325044
	skra Elektromehanika Kranj	MONITOR FDOS				Arhiv				Namensko identifikacijsko številko

Ker se vsa datotečna imena nahajajo v le enem imeniku na sačetku diskete, mora biti vsako datotečno ime unikat. V primeru, da v imeniku že obstaja enako ime datoteke, kot je ime datoteke, ki jo želimo dodati, tedaj nam računalnik javi napako.

Datoteka je lahko dolga od enega do maksimalno 1975 sektorjev (252 800 slogov)

Branje s diskete

Uporabnik vhodno datoteko odpre na ta način, da na lokacije ISIZE, ITRK, ISCTR in ICNTR shrani informacije o vhodni datoteki. Nato vsak klic DSKIN rutine prebere podatkovni slog z diskete in ga naloži v register A.

Če ne obstaja več podatkov (v primeru, da je dolžina vhodne datoteke ISIZE dosegla vrednost 0) se postavi prenosni (carry) bit sicer pa je prenosni (carry) bit brisan. V lokacije ISIZE je potrebno vpisati število sektorjev + 1, da bi se lahko brala še predno je DSKIN rutina prispela na znak, da je konec datoteke (prenosni bit pestavljen). Če programer želi sam skrbeti o koncu datoteke, tedaj je dolžina datoteke lahko malo razsežnejše pozitivno števile, kot na primer ?FFF.

Vsebina v ITRK mora vsebovati število sledi (00-4C) na kateri bo računalnik začel brati vhodno datoteko. ISCTR mora vsebovati v šestem in sedmem bitu število pogonskega mehanizma (00-11) in sekter (00-19 heksa) vhodne datoteke. ICNTR mora pri tem biti postavljen na 00. Vsak klic DSKIN bo prinesel naslednji zaporedni podatkovni slog z diskete. Ko je prebran sektor podatkov (128 zlogov). DSKIN poveča (inkrementira) vsebino lokacij ITRK in ISCTR ter zmanjša (dekrementira) vhodne dolžine (isize). Sektorji, ki vsebujejo DD, so ignorirani, kljub temu pa se vsteti v vhodni dolžini (ISIZE).

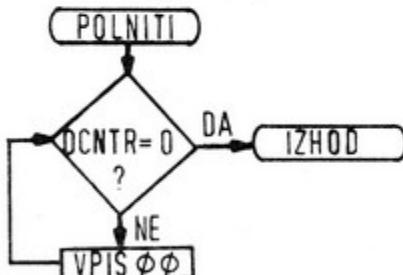
Vpisovanje na disketo

Uporabnik ishodno datoteke odpre na ta način, da na deležene lokacije ISIZE, ITRK in ICNTR shrani informacije o izhodni datoteki. V lokacijo OSIZE se shrani dolžina v sektorjih, če pa se ta dolžina prekeraci, pa DSKOUT posreduje izpis sporočila napake 3. Če programer želi sam skrbeti o koncu datoteke, tedaj je dolžina neko

Izdeja						Ust	Stran	J	K	Identifikacijska številka
Št obvestila						17				V46325044
	Iskra Elektromehanika Krani	MONITOR FDOS				Arhiv				Namensko identifikacijska številka

veliko pozitivno število, kot na primer 7FFF. OTRK vsebuje številko sledi (00-4C) na kateri se bodo začeli vpisovati izhodni podatki. Lokacija OSCTR vsebuje v bitih 7 in 6 številko pogonskega mehanizma (00-11) in začetni sektor (01-1A) na katerem se začno vpisovati izhodni podatki. Ravno tako kot pri branju z diskete, mora tudi tu lokacija OCNTR vsebovati 00. Vsak klic DSKOUT rutine vpiše en podatkovni zlog na disketo. Ko je 128 zlogov poslanih na disk, DSKOUT vpiše te podatke na disketo in poveča (inkrementira) vsebine naslovov OTRK in OSCTR ter zmanjša (dekrementira) dolžino izhodne datoteke OSIZE, DSKOUT po vpisu poizkuša ali je bil vpisan ali ne in po petih neuspeilih poizkusih vpisa, se v ta sektor vpiše znak DD in napreduje na naslednji stični sektor na disketi in ponovno poizkuša vpisati. V primeru, da se v sektor vpiše DD znak, se ravno tako kot ob normalnem vpisu sektorja zmanjša (dekrementira) vrednost lokacije OSIZE.

Vsebina vmesnega vpisnega registra nadzornika (kontrolerja) se vpiše v en sektor na disketi šele tedaj, ko je ta polno zaseden. Ker pa se dogodi, da je uporabnik že vpisal vse svoje podatke, vendar še ni zaseden kontrolno vmesni vpisni register in podatki se ne vpišejo v sektor na disketi. To programer lahko reši na ta način, da dodaja dopolnilne znake (na primer 00) vse dotlej, da ta kontrolno vmesni vpisni register ni poln. Šele nato rutina DSKOUT vpiše podatke na disketo. Diagram poteka takšne polnilne tutine je naslednji:



DELJENJE DISKET

Iz opisa je razvidno da DSKIN in DSQOUT rutimi uporabljata logično/fizične tehnike disketnega naslavljjanja. Sektorji so fizično priležni in stični od 1-26 (hexadecimalne). Po dostopu k fizičnemu 1, se v primeru, da fizični sektor 2 ne more biti trenutno dostopen, se mora disketa zavrteti za 360 stopinj. Ker pa je tak način zelo zamuden, se uporablja za naslavljjanje drug način. DSKIN in DSQOUT prenese zahtevo sekternega nasleva (logični sektor) v nek drug sektorski naslev (fizični sektor), ki ga nato uporabljata DSKIN in DSQOUT rutini.

Primer: zahtevani sektor (logični) je 02, dejansko dostopen pa je fizični sektor OA. Če pa zahtevamo 14 sektor, pa se nato uporablja fizični sektor 10 in take dalje.

Zato da bi spremenili odnos med logičnimi in fizičnimi sektorji, mora program predhodno narediti svoj pripravljalni preved (konverzijo) in s tem "ukani" FDOS pri oskrbi sektorja, prilagojenega z pravkar opisanim prevedom. Prevajalna tabela je shranjena v ROM-u in je na voljo med disketnim nalagalnim (bootstrap) procesom.

Tu so logični in fizični sektorji prikazani v šestnajstškem zapisu:

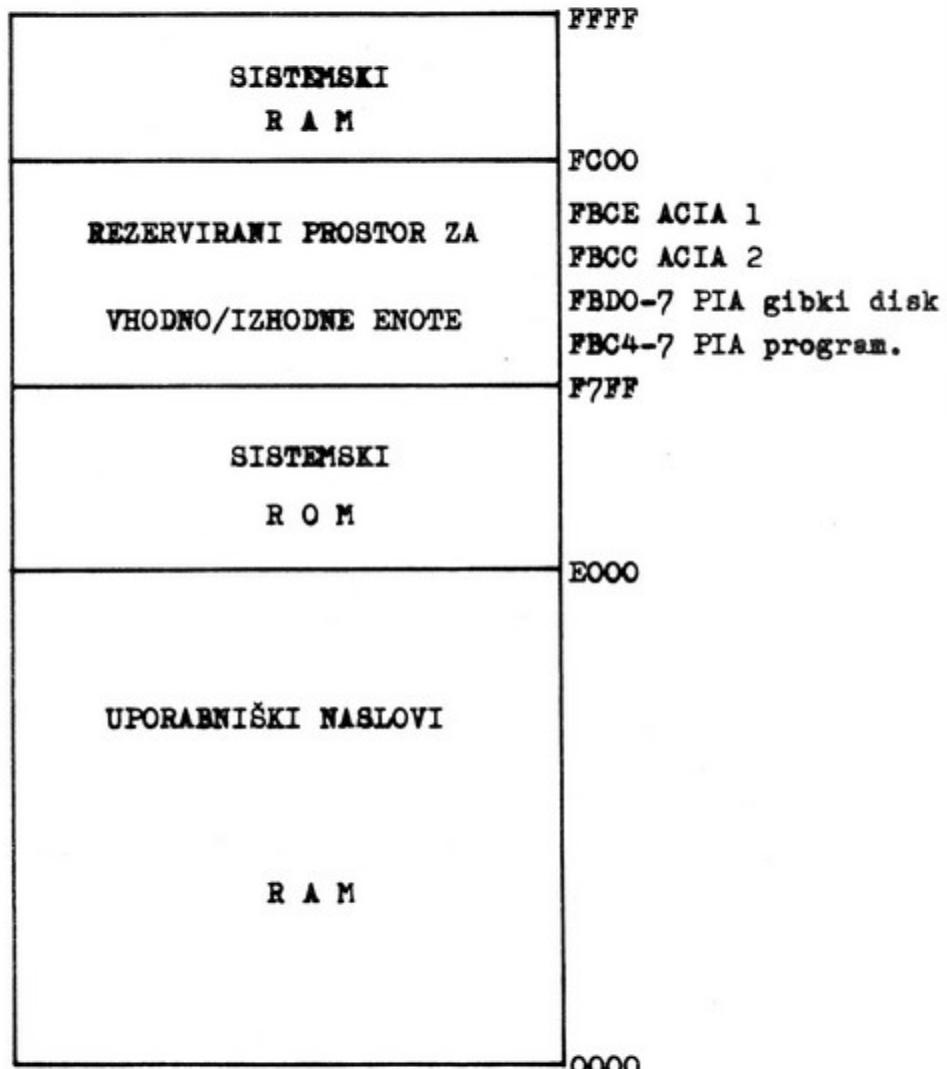
LOGIČNI	FIZIČNI
01	01
02	OA
03	13
04	02
05	OB
06	14
07	03
08	OC
09	15
OA	04
OB	OD
OC	16
OD	05
OE	OE
OF	17
10	06

Izdeja	Let	Stran	S J	K	Identifikacijska številka
Št. obvestila	19				V46325044
	Iskra Elektromehanika Kranj	MONITOR FDOS	Avtor		Namensko identifikacijska številka

LOGIČNI	FIZIČNI
11	0F
12	18
13	07
14	10
15	19
16	08
17	11
18	1A
19	09
1A	12

Izdeja						List	Stran	J	K	Identifikacijska številka	
Št. obvestila						20				V46325044	
 Iskra Elektromehanika MONITOR FDOS Kranj						Arhiv	Nenamenska identifikacijska številka				

RAZDELITEV NASLOVNega PODROČJA :



Prični brojčni nizki so v uporabi v nadzornih posameznih delih dokumenta.

Izdaš							List	Stran	J	K	Identifikacijska številka
Št obvestila							21				V46325044
	skra Elektromehanika MONITOR FDOS Kranj						Arhiv				Namensko identifikacijska številka

MANNING / THE POLITICAL ECONOMY OF THE BOUNDARY 11

	Prinik in ime	Podpis	Gradivo	Odstoje netol. mer	Toplotna obdelava	Površ. zasečita	Pripravnost
Konslr.			Izdaja				101680-10
Projekt			Znak				
Pregled			Sl. obv				
Števil			Datum				
Stand			Podpis				
Naziv	PROGRAM ZBIRNIK				List	Stran	J K Identifikacijska številka
					1		BASMB01MD
Naknadnost kopije				Arhiv	Merilo	Sekcija	Namesto identifikacijske številke

7. ASMB

6800 MAKRO ZBIRNIK

ASMB je program prirejen za sistem ISKRADATA 1680. Njegov namen je prevesti izvirni program v strojnega. To pomeni, da prevede program, ki je napisan v zbirnem jeziku 6800 v drugo, računalniku razumljivo obliko.

Opis delovanja ASMB programa in zahteve zbirnega jezika (zahteva po točno določenem načinu pisanja, po postavljanju ločil v stavkih izvirnega programa, pomenoslovje in uporaba posameznih ukazov, itd.) so razloženi v dokumentaciji, ki je poleg vsakega dobavljenega računalniškega sistema ISKRADATA 1680.

IZVIRNE, STROJNE IN LISTING DATOTEKE

Osnova za nadaljno obdelavo je datoteka izvirnega programa, napisana na gibkem disku. Le-ta mora vsebovati vse stavke, ki bodo kasneje s pomočjo ASMB programa prevedeni. Vsak stavek je niz 0 - 80 znakov napisanih v vrsti, ki jim sledi CARRIAGE RETURN (CR) ali LINE FEED pred ali za CARRIAGE RETURN. Za kreiranje in popravljanje takih stavkov uporabimo standardni program sistema ISKRADATA 1680, ki ga imenujemo EDIT. Za združevanje več datotek v enoto uporabimo FDOS ukaz MERGE. Za tem pa lahko to na novo kreirano datoteko prevajamo kot nov program.

Na posebno zahtevo bo ASMB program kreiral novo datoteko na gibkem disku v standardni 6800 obliki strojnega programa ("S0/S1/S9"), ki jo zahteva 1680 nalagalnik (Loader) in tovarniški postopek 1680 za ustvarjanje ROM-a.

Zbirniški listing (določena predstavitev izvirnega programa, ustreznega strojnega programa in vseh najdenih sintaktičnih napak) lahko prikažemo na ekranu, ga shranimo na gibki disk ali pa ga izpišemo na matričnem tiskalniku.

IZVAJANJE IN KONČNICE

Izdaja					Let	Stran	J	K	Identifikacijska številka
Št. obvezila					2				BASMB01MD
	Iskra Elektromehanika PROGRAM ZBIRNIK Kranj				Arlin				Namensko identifikacijska številka

ASMB je dvostopenjski zbirnik. Za vsak prevod mu je potreben dvakratni prehod preko izvirnega programa. Pri prvem prehodu se prevedejo simboli operacijskih kod in ostale implicitno določene postavke ter izračunajo vrednosti drugih simbolov. Pri drugem prehodu se ustvari strojni program in kreira vse zahteve izhodne datoteke. Vse čas prevajanja je del ekrana rezerviran za prikaz poteka prevajanja. To nam služi kot zavarovanje pri dolgo trajajočih prevajanjih. Na konec prevajanja nas opozori zvočni signal. Kot vhod nam služi izvirni program, medtem ko obliko izhodne datoteke lahko izbiramo poljubno. Zato bo imela zahteva po izvajanju programa ASMB eno od naslednjih štirih oblik:

ASMB, ime vhoda

ASMB, ime vhoda, ime izhoda

ASMB, ime vhoda, ime izhoda, parameter

ASMB, ime vhoda,, parameter

S parametrom določimo kje in v kakšni obliki želimo imeti izhodno datoteko. Določi se z eno ali dvema heksadecimalnima številkama.

ASMB, TRT :1, TST:1, 12

S prvo številko povemo, če zahtevamo na gibkem disku strojni program ali ne.

0X - datoteka strojnega programa se ne prenese na gibki disk

1X - zahteva se datoteka v strojnem programu na gibkem disku

Druga številka nam pove, na kakšno napravo želimo prenesti zbirni listing.

X0 - gibki disk

X1 - ekran

X2 - matrični tiskalnik

X8 - ne ustvari listinga na gibkem disku

S kombinacijo številk lahko dosežemo različne načine prikazovanja, tiskanja in zapisovanja zbirnega listinga. V nobenem primeru pa se

Izdaja						Let	Stren	J	K	Identifikacijska številka
Št. obvestila						3				BASMB01MD
	Iskra Elektromehanika PROGRAM ZBIRNIK Kranj					Aktiv				Namensko identifikacijska številka

zbirni listing ne bo prenesel na gibki disk, če se zahteva datoteka strojnega programa. Če ne napišemo prve številke v parametru, se smatra, da je vrednost prvega znaka nič (0). V celeti ispuščen parameter, pa da enak rezultat kot kombinacija številk 00. Ta kombinacija je najčeščje uporabljena v času razvijanja programa.

Pri uporabi ASMB programa se nam nudi na prvem mestu parametra še ena možnost :

8X

Pomen uporabe bo razložen kasneje.

OBLIKA ZBIRNEGA LISTINGA

ASMB program običajno služi programerju kot orodje za izdelavo zbirnega listinga. Strojni program je navadno mišljen kot stranski produkt listing obdelave in je redkeje prikazan na ekrantu ali tiskan. Najpogosteje pišemo zahteve izvirnega programa izhajajoč iz oblike listinga.

Glava zbirnega listinga sestavljata pet vrstic :

- dve vrsti predstavitve ASMB programa in programa, ki ga bomo prevajali,
- prazna vrstica,
- vrstica za razložitev pomena stolpcev,
- in še ena prazna vrstica.

V SOURCE stolpcu se ponovno prikaže vsaka vrstica izvirnega programa. Zaradi večje preglednosti in lajšega branja je izvirni program izpisani v obliki medsebojno ločenih stolpcev (polj).

S pomočjo SEQ stolpca (Sequence) določimo mesto vsake vrstice snotraj datoteke izvirnega programa. Dodatne vrstice, ki jih ustvari makro ukaz, se označene s kombinacijo znaka plus in številke. Številka v osnaki pomeni nivo vrinjenja.

Popolnoma prazna vrsta snotraj izvirne datoteke se šteje kot normalna vrsta. Rezultat take vrste v listingu je prazna vrsta označena s saperednim številom v SEQ stolpcu.

Izdaja						list	Stran	J	K	Identifikacijska številka
Št. dovestila						4				BASMB01MD
	skra Elektromehanika PROGRAM ZBIRNIK Kranj					Aribit				Namensko identifikacijska številka

LOC ("Location") stolpec prikaže vrednost ASMB lokacijskega števca v trenutku pričetka obdelave ustrezne vrste izvirnega programa. Lokacijski števec predstavlja "poskusni naslov", ki se lahko spremeni v času nalaganja strojnega programa v pomnilni medij.

OBJ ("Object") stolpec vsebuje en, dva ali tri bajte, za odgovarjajočo vrstico izvirnega programa, ustvarjene kode. Če stavek izvirnega programa ne ustvari strojne kode, se na ustrezačem mestu OBJ stolpca pojavi vrednost določena s stavkom. Na koncu listinga programa so napisane ne glede na ostale parametre tri postavke:

- vsota vrstic izvirnega programa
- zadnji heksadecimalni naslov lokacijskega števca
- in število vseh najdenih sintaktičnih napak

Vsaka napaka je posebej opisana v sporočilu, ki sledi listingu.

Izdaža							Ust	Stran	J	K	Identifikacijska številka
St. obvestila							5				BASMB01MD
	Skra Elektromehanika PROGRAM ZBIRNIK Kranj						Avtor				Namensko identifikacijsko številko

KOMENTAR

Zvezdica na začetku vrste v izvirnem programu označuje, da je ta vrsta komentar. Če pa se zvezdica ali katerikoli drug znak pojavi v nadaljevanju te vrste, se le-ta pojavi nespremenjen in na svojem mestu tudi v listingu. Sprejemljiva je tudi popolnoma prazna vrsta. Z njo dosežemo večjo pregljednost listinga.

POLJA V STAVKU

Vsak stavek izvirnega programa, ki ni komentar je sestavljen iz veš delov imenovanih polja. Katera polja mora stavek vsebovati in katera niso obvezna je določeno s stavčnim tipom (6800 ukaz, zbirniške direktive in podobno).

V zbirnem jeziku poznamo štiri polja:

- polje labele
- operacijsko polje
- polje operanda
- polje komentarja

POLJE LABELE

Na prvem mestu v stavku se zašne polje labele. Če je prvo mesto prazno, potem labele v stavku ni, pač pa se stavek začne z operacijskim poljem.

Na prvem mestu v stavku se lahko pojavi le globalni simbol, lokalni simbol ali zvezdica, ki označuje takojšen začetek polja komentarja. Maksimalna dolžina globalnega simbola je šest znakov.

Prvi znak v simbolu je ena od velikih črk A do Z. Znaki v nadaljevanju simbola pa so lahko velike črke ali številke.

Vrednost globalnega simbola je enaka vrednosti ~~izhajajočega~~ števca v trenutku ko se simbol pojavi v polju labele. Z izjemo simbolov, ~~ki~~ ~~so~~ ~~so~~ definirani z direktivo SET, je na ta način vrednost globalnega simbola stalna kjerkoli v izvirnem programu. Če uporabiš isti globalni simbol kot labelo večkrat se smatra kot napaka,

Izdaja	St. obvestila	Let	Štorm	J	K	Identifikacijska številka
		6				BASMB01MD
	Iskra Elektromehanika PROGRAM ZBIRNIK Kranj	Artik				Nomero identifikacijske številke

ker je s tem simbol večkratno določen (multiply defined).

Na prvem mestu lokalnega simbola je vedno dvopičje, ki mu lahko sledi ena do šest velikih črk ali številk.

: Q

: ABCDEF

Vrednost lokalnega simbola je določena ravnotako kot vrednost globalnega simbola, z njegovo pojavom v polju labele. V trenutku ko se v operacijskem polju v določeni točki izvirnega programa pojavi zbirniška direktiva LOCAL izgubijo svojo identiteto in vrednost vsej lokalni simboli, ki so nad to točko. Ta lastnost dopušča večkratno uporabo istega lokalnega simbola znotraj nesorodnih delov programa, kot so določila kratkih razvezanj ali skokov ali znotraj makro definicij.

SAMODOLOČLJIVE VREDNOSTI (SELF - DEFINING VALUES)

Vsek operand v stavku je lahko enostaven člen ali pa zložen izraz, ki vsebuje več členov in operatorjev.

"Samodoločjivi člen", kot nasprotje simboličnega člena, ima takoj razpoznavno vrednost. Vrednost je poznana takoj ko zbirnik prvič predstavlja člen.

Za samodoločljive člene lahko uporabimo štiri sisteme simbolov:

- desetiška števila
- heksadecimalna števila
- binarna števila
- znakovne vrednosti

Desetiška števila so sestavljena iz ene do pet (desetiških) številk, ki predstavljajo katerokoli vrednost od 0 do 65535.

Heksadecimalno število sestoji iz ene do štirih heksadecimalnih številk. (Številk 0 do 9 in velikih črk A do F). Da je to število heksadecimalno, označimo tako, da pred število napišemo oznako za dolar (\$). Sicer ne bi mogli razlikovati simbola kot naprimer FAD od heksadecimalnega števila \$FAD.

Binarno število je sestavljeno iz 1 do 16 binarnih številk. (bitov 0 in 1). Da omogočimo razlikovanje binarnega števila od desetiškega napišemo pred številko znak za procent (%).

Indra						Ust	Stran	J	K	Identifikacijska številka
Št. obveznika										
	skra Elektromehanika PROGRAM ZBIRNIK Kranj					7				BASMB01MD
						Artik				Namensko identifikacijska številka

Apostrof napisan pred členom pa pomeni, da je vrednost člena znakovna. Črka, številka ali lečilo, ki apostrofu neposredno sledi je ASCII znak (the American Standard Code for Information Interchange). Tako naprimer znakovna vrednost 'A ustreza desetiški 65, heksadecimalni \$41 in binarni 01000001.

REFERENCA LOKACIJSKEGA ŠTEVCA

Vrednost lokacijskega števca uporabljamo kot operand. Označen je z zvezdico, in je enak vrednosti lokacijskega števca. V trenutku začetka prevajanja ustrezne vrstice izvirnega programa.

ZLOŽENI IZRAZI (COMPLEX EXPRESSIONS)

Člene sestavljamo v sležene izraze s pomočjo operatejjev. Ti nakazujejo na aritmetične ali logično odvijanje procesa. Pravila za delovanje vrednosti izraza so povsem podobna algebraičnim in tudi pravilom, ki veljajo za večino programskih jezikov.

Če delečevanje vrednosti poteka od leve proti desni je delni rezultat delečevanja vedno 16-bitna vrednost. Postopek se popolnoma sklada z aritmetike 6800 registrov - overflow-a ne upoštevamo in vsake vrednost s enico (1) v levem bitu lahko vzamemo za predstavitev vrednosti dvojnega komplementa.

UNARY OPERATORJI

Vrednosti kakršnihkeli členov (samodoločljivih ali simboličnih), lahke urejame z Unary operatorji. V naspremju s binarnimi operatorji, ki služijo za povezavo dveh členov v složen izraz, se Unary operator lahko pojavi na samem začetku izraza ali šele po binarnem operatorju.

Minus snak pevsreči, da člen pekače negacije ali bolje rečeno dvojno komplementacijo binarne snakovrednih členov.

Number:

- T
- \$7F

Plus znak nima vpliva na vrednost člena. Uporabimo ga če hočemo podatiti, da člen nima predpisane minus Unary znaka. Lahko pa se pojavi tudi med makso postopkom.

- + T
- + \$7F

Znak \wedge pred členi, podobnim temu dvema:

- $\wedge T$
 $\wedge \$7F$
- T - l povzroči enojni komplement

ARITMETIČNI OPERATORJI

Za sestevanje in odštevanje uporabljamo plus in minus znak.

T + U
T - U
T + U - l
T - U + l

Za množenje dveh ali več členov uporabljamo zvezdico.

T * U
T * 2
256 * T * U

Operacijo deljenja označimo s / znakom.

T/U
T/2

Ostanek pri deljenju ne upoštevamo. Če je divizor 0 bo rezultat

Izdele						List	Stran	J	K	Identifikacija številka
Št. obvestilo						9				BASMB01MD
	skra Elektromehanika PROGRAM ZBIRNIK Kranj	Artik								Domestno identifikacijsko številka

binarno število s samimi enicami, heksadecimalno \$FFFF in (desetiško) - 1 ali + 65535.

LOGIČNI OPERATORJI

Znak "&" označuje logični IN vrednosti bitov dveh členov.

T & U

T & \$7FFF

Podobno klicaj označuje inkluzivni ALI.

T ! U

T ! \$7FFF

Eksluzivni ALI pa označimo z znakom #

T # U

T # \$7FFF

TRDITVE (ASSERTIONS)

Operacijsko polje lahko vsebuje trditve o relativni vrednosti dveh členov ali izrazov. Napriemr trdišev, da ima simbol ABC večjo vrednost od simbola OPR plus 1 napišemo na ta način:

ABC > OPR + 1

Vsaka trditev ima "vrednost resnice", (truth value), ki jo lahko uporabimo kot katerikoli drug člen v izrazu. Pravilne trditve ustvarijo vrednost šestnajstih (16) binarnih enic (\$FFFF), napačne pa šestnajst binarnih ničel (0000).

Kot bomo videli pozneje se "vrednost resnice" v glavnem uporablja za kontrolo pogojnega zbirnika (conditional assembly).

Za tvorjenje trditve uporabljamo ~~zapisane~~ matematične simbole.

Indraž	Št. obveznika	Ust	Stran	J	K	Identifikacijska številka
		10				BASMBD1MD
	Iskra Elektromehanika PROGRAM ZBIRNIK Kranj	Arhiv				Nenamerno identifikacijska številka

- < manjši od
- = enak
- > večji od

Ločila imenujemo relašijske operatorje. Lahko jih tudi sestavljamo.

- \leq manjši ali enak
- \geq večji ali enak
- \neq ni enak

Uporaba logičnih operatorjev nudi možnost tvorjenja zloženih trditev.

$$A = B \& B + 1 \leq C \& C \neq D - 2$$

Ta zložena trditev je enakovredna naslednjim trem, ki so istočasno tudi vsaka zase pravilne.

$$\begin{aligned} A &= B \\ B + 1 &\leq C \\ C \neq &D - 2 \end{aligned}$$

VRSTNI RED IZVRŠEVANJA OPERACIJ

Vrstni red izvrševanja operacij poteka od leve proti desni, toda nekateri operatorji imajo prednost pred drugimi. Tako ima množenje prednost pred seštevanjem. Zato je vrednost izraza:

$$2 + 3 \times 5 = 17 \quad (2 \text{ plus produkt } 3 \text{ in } 5)$$

in ne 25 kakršna bi bila v primeru poteke operacij od leve proti desni. Vrstni red izvrševanja operacij lahko spremenimo z uporabo oklepajev.

$$(2 + 3) \times 5
((A + B) \times (D - E)) - (F/3)$$

Oklepaje lahko uporabimo tudi v primeru ko z njimi ne želimo spremeniati vrstni red izvrševanja operacij, ampak z njimi dosežemo boljje branje in večjo preglednost sestavljenih izrazov.

Izdaja						Lst	Stvar	J	K	Naselj/Načrtna številka
Št. obveznila						11				BASME01MD
	Skra Elektromehanika PROGRAM ZBIRIK Kranji					Avtor				Bameto identifikacijska številka

$$(A = B) \& (B = C) \& (C > D)$$

Press freedom remains in jeopardy & independent media rights are under assault.

TABELA ZA DOLOČITEV PRIORITETE OPERATORJEV

OPERATOR	STOPNJA PRIORITETE	POMEN
+	6	unary plus
-	6	unary minus
Λ	6	logični NE
×	5	množenje
/	5	deljenje
+	4	seštevanje
-	4	odštevanje
<	3	manjši od
=	3	enak
>	3	večji od
<*	3	manjši ali enak
>*	3	večji ali enak
<>	3	ni enak
&	2	logični IN
I	1	logični inkluzivni ALI
#	1	logični ekskluzivni ALI

Alguns treballs en els quals es troben l'etnografia i l'antropologia d'una forma molt estreta.



Iskra
Elektromehanika PROGRAM ZBIRNIK
Kranj

List	Št. stran	J.	K.	Identifikacijska številka
13		BASMB01MD		
Arhiv		Mestno identifikacijska številka		

6800 UKAZI

Ko v operacijskem polju stavka zbirnik spozna standardno 6800 operacijsko kodo, jo skupno z njo pripadajočo vrednostjo prevede. V listingu se ta prevod prikaže v OBJ stolpcu.

06	TAP		
39	RTS		; KOMENTAR
4C	INC A		; KOMENTAR
5C	INC B		
7C 1234	INC	\$1234	; KOMENTAR
7C 2468	INC	\$1234 # 2	
9E 00	LDS	0	; DIREKTNO
BE 1234	LDS	\$1234	; RAZŠIRJENO
AE 00	LDS	0,X #	; INDEKSNO
8E 0000	LDS	# 0	; TRENUTNO

V zadnjih štirih vrstah vidimo, da mora zbirnik omogočiti razlikovanje načina naslavljjanja posameznega ukaza.

Direktni in razširjeni način naslavljjanja nimata posebnega pokazatelja. Če ne zahtevamo drugače, bo zbirnik, težec h krajšim ukazom vedno uporabil direktni način naslavljjanja, čeprav je naslov operanda navadno večji od \$FF, ali pa še ni določen ko gre izraz skozi prvo stopnjo prevajanja.

Indeksni način naslavljjanja označimo z izhodiščnim naslovom povečanim za ustrezno premaknitev. Sledi vejica in velika črka X. Trenutni način označimo z znakom # na začetku polja operanda. Znak je napisan pred predvideno trenutno vrednostjo izraza.

V operacijskem kodu akumulatorskega ukaza sta črki A in B lahko napisani neposredno za operacijskim kodom ali pa sta od njega ločeni z enim praznim mestom. Sistem 1680 navadno uporablja ločeni način pisanja.

Izdelek							Ust.	Stran	J	K	Identifikacijska številka
Št. dovestila							14				BASMB01MD
	Iskra Elektromehanika PROGRAM ZBIRNIK Krani						Avtor				Namensko identifikacijska številka

B6 1234	LDA A \$1234
C6 7B	LDA B 'Z + 1
96 FE	LDAA \$FE
E6 66	LDAB \$33 # 2,X

DODELJEVANJE PODATKOV IN NJIHOVO SHRANJEVANJE

Za dodeljevanje podatkov in določevanje njihove vsebine lahko uporabljamo štiri zbirniške direktive:

FCB	formiraj konstantni zlog
FDB	formiraj dvojni zlog
FCC	formiraj konstantni znak
RMB	rezerviraj spominski zlog

Polje operanda FCB ali FDB stavka vsebuje izraz ali več izrazov ločenih z vejicami. Vsak izraz je nameščen v posameznem zlogu (pri FCB direktivi) ali v 16-bitnem dvojnem zlogu,(pri FDB direktivi) primeren za uporabo z LDX ukazom.

FCB	\$FF
CAT	9CB - 1, +0, ALFA + 4 , \$FF
	FDB BETA
DOG	FDB \$FFFF, DELTA - L, (ABC - DEF)/2

Polje operanda FCC stavka vsebuje niz - poljubno zaporedje znakov (string). Navadno se uporablja za izpisovanje sporočil na zaslonu prikazovalnika ~~ali se tiskanje na matičnem tiskalniku.~~ Ne posredno pred in po sporočilu moramo postaviti oklepaj (/). Če pa se oklepaj pojavi v sporočilu samem, pa lahko uporabimo za označitev začetka in konca sporočila narekovaj.

FCC	/VNESI NASLEDNJI UKAZ/
MSG3	FCC "NAPAKA V I/O VRSTI"

Polje operanda v RMB stavku predstavlja neko število zlogov zaporednega spomina. Zbirnik samo poveča lokacijski števec za ustrezeno število.

Izdaja					Ust	Stavm	J	K	Identifikacijska številka
Št. obvestila					15				
	Iskra Elektromehanika PROGRAM ZBIRNIK Krani								
									Namensko identifikacijska številka

no število zlogov brez ustvarjanja podatkov za rezervirana polja. S tem dosežemo da ostane vsebina nespremenjena tudi pri morebitnem nalaganju strojnega prgorama.

RMB 4
FOX RMB SIGMA + 1

Če ena od teh direktiv ustvari več kot tri zloge strojnega programa se odvečni zlogi pojavijo v listingu v delu predvidenem za heksadecimalne vrednosti. Tem zlogom pa sledi ustreznna vršta izvirnega programa. Da to preprečimo, lahko uporabimo listinag kontrolno direktivo LDATA. V tem primeru se ti zlogi izpišejo kasneje.

VODENJE ZBIRNIKA

Za spremembo lokacijskega števca brez ustvarjanja strojnega zapisa lahko uporabimo direktivo ORG.

```
ORG      $FFFA ; ustvarjanje zapisa od $FFFA naprej
        BDB      SOFT ; 6800 prekinitveni prvi
        ORG      $0100 ; ustvarjanje zapisa od $0100 naprej
        LDS      #FFFO ; prvi programski ukaz
```

Primer pokaže kako lahko namestimo naslov lokacije prekinitvenega vektorja med nalaganjem programa, neglede na to, da bo ostanek strojnega programa najdešen v drug del spomina.

Za pripis vrednosti simbola uporabimo direktivo EQU, s simbolom v polju labele in izrazom v polju operanda.

```
START    EQU      *
ALFA     EQU      * + 256
BETA     EQU      ALFA/2 - 1
```

Vsek člen v operandu ORG in EQU stavka mora biti predhodno definiran. Nasprotno pa je dopuščeno, da pri ostalih direktivah (SET direktiva) izraz operanda definiramo pozneje. Prav tako je tudi mogo, da v polju labele uporabimo enake simbole v več SET stavkih.

Izdeja	St. obveznila	Let	Stran	J	K	Identifikacijska številka
		16				BASMB01MD
	skra Elektromehanika PROGRAM ZBIRALNIK Kranj	Arhiv				Nomero identifikacijske številke

```

FLAG      SET    1
MARK     SET    (A - B) & (C   D): FLAG
FLAG      SET    0

```

Tak način uporabe simbolov v pelju labela je dovoljen le v SET stavkih. SET direktivo prvenstveno uporabljamo v kontroli pogojnega sbirnika.

Zadnji stavek v programu mora biti END direktiva. S tem dosežemo, da sbirnik po prvi stopnji prevajanja začne tudi z drugo. V nasprotnem primeru bi se sporečilo o napakah pojavilo ko bi sbirnik naletel na logični konec izvirnega programa.

POGOJNO PREVAJANJE

Z vstavljanjem deleženih direktiv v izvirni program je mogoče sbirnik vedeti skozi proces imenovan "pogojno prevajanje". To pomeni, da bo nekaj skupin stavkov prevedenih v strojni zapis ali obdelanih kot direktive, medtem ko ostale stavke enostavno ignoriramo.

Pogojno prevajanje prvenstveno uporabljamo v dva namena:

- za kreiranje rahle različnih verzij istega programa
- za možnost razširjanja makrejev.

Začetek pogojnega prevajanja označimo z IF, konec pa z IEND. IF stavek zahteva člen operanda ali izraz, ki vsebuje le prediktivno deležene vrednosti. Operand služi za deležitev "resnične vrednosti" (truth value). Pri tem se ravnamo po pravilu:

"Napačno" je vsako sedež število

"Pravilno" je vsako like število

Tako je glede na te pravile \$0000 narobe in \$FFFF pravilno. V primeru ko IF stavek specificira "napačno" trditev, se ne bo posredoval noben stavek izvirnega programa vse do IEND direktive. Če pa bo trditev resnična bo prevajanje potekalo normalno.

IF/IEND par lahko vrinemo snetratj drugega para. Tako strukturo lahko izdelamo maksimalno v 16-tih nivojih.

```

IF      ALFA = BETA      ; zmanjši IF
PCB     4

```

Izdaš	St. obveznila	Identifikacijska številka
	skra Elektromehanika PROGRAM ZBIRNIK Kranj	17 BASMBO1MD Namesto identifikacijske številke

FDB	OR,UV,WX,YZ	
IF	DELTA & SIGMA & GAMA	; vrinjeni IF
FCC	/NAPAKA/	
FCC	/NAPAKA NA/	
FAT	RMB 4	
	IEND	; vrinjeni IEND
	FCB O	
	IEND	; zunanji IEND

MAKROJI

Makro je skupina stavkov, ki ima svoje ime in lastnost, da lahko v izvirnem programu večkrat uporabimo celo skupino stavkov, pri tem pa nam je potrebno označevanje treh med seboj sorodnih gledanj zbirnega jezika:

- makro definicije
- makro klice
- makro razširjenja

Makro definicija je izjava, ki določi ime skupini stavkov. Izjavi sledijo stavki, ki jih zbirnik shrani za nadaljne napotke.

Makro klic je stavek, ki dejansko uporabi v makro definiciji določeno ime. Ime se pojavi v operacijskem polju, le če je to 6800 ukaz ali standardna zbirniška direktiva in je operacijsko polje specifirano kot potrebovano. Dopustno je da stavek makro klica vsebuje tudi polje labele in komentarja.

Makro razširjanje je rezultat zbirniške predelave makro klica. Njegova naloga je, da v listingu primerja skupine stavkov, ki bodo vneseni v vhodno datoteko takoj po makro klicu.

Če z makro klicem le reproduciramo določeno skupino stavkov, ne bomo polno izkoristili možnosti, ki nam jih daje makro obdelava. Prava vrednost le-te je v tem, da lahko v makro definiciji vključimo tudi pogojno prevajanje stavkov. Na ta način lahko ovrednotimo vrednost operanda in določimo izvor ustvarjenega zapisa. Vsak operand se lahko fizično pojavi v katerikoli točki makro

Izdeja					Ust	Stren	J	K	Identifikacijska številka
Št obveznika					18				BASMB001MD
Iskra Elektromehanika PROGRAM ZBIRNIK Kranj					Avtobus				Danesko identifikacijska številka

razširjanja. Postopek imenujemo "substitucija".

Makro definicija se začne z MACRO, konča pa z MEND direktivo. Polje labele makro stavka mora vsebovati ime, s katerim bo makro določen. Začetni znak imena mora biti velika črka, v nadaljevanju pa so lahko črke in številke. Ime je lahko dolgo 1 do 6 znakov in enako kot kakšen že prej v druge namene uporabljen simbol. To je dopuščeno zaradi lastnosti, da je zbirnik zmožen razpozнати res-like v namezu uporabe.

Število stavkov, ki jih napišemo med MACRO in MEND stavkoma ni omejeno. Ti stavki predstavljajo "model" iz katerih bo zbirnik ustvaril makro razširjanje.

Polje operanda makro stavka lahko vsebuje seznam "oblikovnih parametrov" (Formal Parameters), ki kontrolirajo substitucijo in pogojno prevajanje. Vsak oblikovni parameter primerja simbol, razen v primeru, ko je pred njim napisan znak "@".

MACK @ FIR, @ ELM, @ PINE

Med postopkom makro razširjanja se vsak oblikovni parameter nadomesti z operandom makro klica ali izrazom, ki zavzame ustrezeno mesto v operandu.

Če se pojavi makro klic:

MACK SIGMA,DELTA,TAU

se bo vsak simbol @ ELM v makro definiciji nadomestil s simbolom DELTA. Šele potem bo prevedena ustrezena vrsta izvirnega programa.

Primer makro definicije:

```
NOVA MACRO @ RED, @ BLUE, @ GREEN  
      FCB @ GREEN + 1  
      :CRED FCC /@TIKALO NA @ BLUE, IN STOP/  
      IF @ GREEN 3
```

Izdeja							Let	Stran	J	K	Identifikacijska številka
St. obvestila								19			BASME01MD
	skra Elektromehanika PROGRAM ZBIRNIK Krani						Avtor				Namensko identifikacijska številka

RMB	2
IEND	
FCB	0
MEND	

Z makro klicem

XYZ NOVA N7, DELTA, OFF

se bo v listingu pojavilo makro razširjanje

FCB \$FF + 1
:H7 FCC /STIKALO NA DELTA IN STOP/
IF \$FF 3
RMB 2
FCB 9

MAKRO KLICI

Makro klici se lahko pojavijo na vseh tistih mestih kot običajen ukaz. V posebnih prisparih se lahko pojavi tudi znotraj definicije drugega makroja.

Vsek makro klic za dani makro lahko vsebuje operande, ki se ujemajo z nekaterimi ali vsemi oblikovnimi parametri makro definicije. Če je operand izpuščen, bodo izpuščeni tudi oblikovni parametri. V razširjanju moramo paziti, da bo stavek še vedno obdržal veljavno obliko. Tukaj je primer makro klicev za makroje z nekaterimi izpuščenimi operandi.

NOVA ,R,S ; komentar
NOVA , ,S ; komentar
NOVA Q , R ; komentar

Če želimo izpustiti končne operande jih enostavno ne napišemo. Medtem, ko moramo pri ostalih označiti kateri operand želimo izpustiti s praznim mestom in vejico ali z dvema vejicama.

Izdejca						List	Stran	J	K	Identifikacijska številka
Št. obveznika						20				BASMBQ1MD
	Iskra Elektromehanika PROGRAM ZBIRNIK Kranj					Avtor				Namensko identifikacijska številka

ČRKOVANJE OPERANDA IN POGOJNO PREVAJANJE

Če moramo preveritit definicijo je bolje da operand makro klica črkujemo kot da bo pregledovali njegovo vrednost. V ta namen uporabim direktivo za kontrolo pogojnega prevajanja IFC. Ta direktiva da "pravilno vrednost" ko sta v operandu dva identična niza ločena z oklepajem na začetku in koncu vsakega od njih.

IFC /@FIR/DELTA/

Med razširjanjem, so lahko makro definicije podobne tem:

IFC /DELTA/DELTA/
IFC /SIGMA/DELTA/

Po substituciji pa je prva "pravilna" druga pa "napačna".

POMEMBNOST SPOROČILA NAPAK

Če hočemo, da se nam pojavijo v listingu (in na ekranu) izpisane le napačne vrste, lahko uporabimo direktivo ERROR. V tem primeru se bo izpisala le direktiva z vsemi njenimi polji.

```
IF      ( ELM < 1 ) ! ( ELM > 9 )
ERROR   VREDNOST @ ELM NI SPREJEMljIVA
ELSE
PCB     @ ELM
IMMO
```

KONTROLNE DIREKTIVE LISTINGA

Direktiva TITLE povzroči, da se listing nadaljuje na naslednji strani (čeprav prejšnja še ni polna). V polju operanda te direktive je napisan tekst, ki bo napisan v sredini prve vrste naslednje strani.

TITLE CONTROLA ROM-a, LIST REVISION 2

Izdaja	Št. obvestila	Let	Stran	J	K	Identifikacijska številka
		21				BASMB01MD
	skra Elektromehanika PROGRAM ZBIRNIK Kranj	Avtor				Namensko identifikacijska številka

V zbirniku je dopustna tudi druga oblika te direktive - NAM.

Direktiva PAGE povzroči skok na naslednjo stran. V polju operanda te direktive je ~~tekst~~, ki bo izpisana v sredini druge vrste naslednje strani.

PAGE IZHODNE PODRUTINE

Če PAGE direktiva nima polja operanda, se v drugi vrsti izpiše, enak tekst kot na predhodni strani.

Vse naslednje direktive so pogojne z "resnično" ali "napačno" vrednostjo operanda.

Direktivo LIST lahko uporabljamo, kadar želimo imeti možnost izbire ali se bodo vrste od te direktive dalje pojavile tudi v listingu, ali pa bodo predelane ne da bi se v njem prikazale.

LDATA direktiva služi za kontrolo prikazovanja posebnih vrstic. Te vrste so vzrok FCB, EDB in FCC direktiv, ki ustvarijo več kot tri zloge strojnega zapisa.

Direktiva LSKIP služi za preskok vrstic, za katere pokaže pogojno prevajanje, da jih ni treba prevesti.

LMDEF preprečuje izpisovanje makro definicij.

LMCAL pa preprečuje makro razširjanje nadre makro klic vrsto.

LSYMB direktiva, pomeni da bo listingu sledila stran z vrednostimi globalnih simbolov v zahtevanih definicijah.

V primeru, ko ne napišemo nobene od kontrolnih direktiv bo potekalo prevajanje (listanje) programa tako, kot da bi bile direktive LIST, LSKIP, LMDEF in LMCAL zahtevane ("resnica"), direktivi LSYMB in LDATA pa izpuščeni ("napačni").

Vrednost za vsako od teh direktiv je shranjena v ločenem 8-nivojskem polju. Z vnašanjem nove vrednosti se polje razširja, edino v primeru, da je direktiva brez kakršnegakoli operanda se polje briše. Ta lastnost dopušča makro definicijam obdržati vrednosti dobljene pred razširitvijo.

Izdaja									Let	Sistem	J	K	Identifikacijska številka
St. obveznila									22				BASMB01MD
	Iskra Elektromehanika PROGRAM ZBIRNIK Kranj								Arhiv				Namensko identifikacijska številka

OMEJEVANJE UPORABE SPOMINA

ASMB program uporablja način naslavljanja od najvišjega naslova v RAM-u navzdol. Če hočemo ohraniti višji RAM, da bi ohranili določeno rutino, moramo uporabiti parameter 80 ali višji. To bo povzročilo, da bo ASMB program deloval kakor, da bi bil najvišji naslov 35FF.

DELOVANJE ZBIRNIKA

Za delovanje zbirnika je vseeno, če je terminal priklopljen v PAGE ali SCROLL način delovanja. Vendar je zaradi preglednejšega testiranja napak na ekranu, priporočljivo da ga preklopimo v SCROLL položaj.

Izdaš					List	Stran	J	K	Identifikacijska številka
Št obvestila					23				BASMB01MD
	Iskra Elektromehanika PROGRAM ZBIRNIK Kranj				Arhiv		Namesta identifikacijska številka		

LIST	STRAN	IZDAJA	LIST	STRAN	IZDAJA	LIST	STRAN	IZDAJA
LIST	STRANA	IZDANJE	LIST	STRANA	IZDANJE	LIST	STRANA	IZDANJE
SHEET	PAGE	ISSUE	SHEET	PAGE	ISSUE	SHEET	PAGE	ISSUE

PROGRAM ZA VPIS V EPROM INTEG. VEZJE - S 6834

Program za vpis v EPROM S 6834 (S6834) omogoča vpisovanje, čitanje primerjanje in premikanje EPROM integ. vezja, ki ga vstavimo v programsko podnožje na čelni plošči mikrorodenalnika Iskradata 1680 -10. Uporabnik definira začetni in končni naslov pomnilnika in naslov EPROM integ. vezja (0000 - 01FF). Pri vpisu in primerjanju program javlja napake, če se pojavi.

UPORABA: ko pokličemo program S 6834 z diskete se ta naloži v pomnilniške lokacije 0000 - 04FF, zato lahko vpisujemo, čitamo, itd. samo iz ostalih pomnilniških lokacij, kamor predhodno z "LOAD" komando naložimo program, ki ga želimo obdelovati. Na prikazovalniku dobimo:

I S 6834 (CR)

"B" - VPIS V EPROM INTEG. VEZJE
"I" - BERI EPROM INTEG. VEZJE
"V" - PRIMERJAJ Z EPROM INTEG. VEZJEM
"M" - PREMIK V POMNILNIKU PROSTORU
"E" - IZHOD V IDOS

ADL ADH PROM

%

Sedaj lahko z vtipkavanjem ustrezne šrke in naslovov starta program.

Primer vpisovanja: Vpiši z pomnilniških lokacij 2000 do 21FF v EPROM integ. vezje z začetnim naslovom 0000:

B2000-21FF-0000 (CR)

Primer čitanja: v pomnilne lokacije 3000 do 31FF naloži vsebino EPROM integr. vezja z naslovom 0000

T 3000-31FF, 00000(CR)

Izdsja							List	Stran	J	K	Identifikacijska številka
Št obvestila							2				B683401MD
	skra Elektromehanika	PROGRAM	S	6834	Kranj		Avtor				Razpolaga identifikacijsko številko

Primer primerjanja: primerjaj pomnilne lokacije 3500 do 36FF z vsebino EPROM integ. vezja z naslova 0000. Napake se izpišejo! V3500, 36FF, 0000 (CR)

Primer premika: Premakni vsebino pomnilnih lokacij od 2500 do 3000 na pomnilno lokacijo 1000.
M2500, 3000, 1000(CR)

Primer za izhod iz programa:
E (CR).

Izdaja	St. obvestila	Unit	Stran	J	K	Identifikacijska številka
		3		B	683401MD	
	Iskra Elektromehanika Kranji	PROGRAM	S 6834	Avtov	Razmesta identifikacijske številke	

PROGRAM INVERZNI ZBIRNIK DISS

Program INVERZNI ZBIRNIK (DISS) prevaja ukaze iz strojne kode v ukaze zbirnega jezika. Ukaze bere iz določenih lokacij v pomnilniku, katerih začetni naslov definira uporabnik na začetku. Uporabnik definira tudi končni naslov in enoto kamor naj se izpisujejo rezultati prevajanja.

V izpisu je v prvi koloni naslov, kjer je shranjena strojna koda, v drugi koloni je izpisana ta strojna koda, v tretji koloni je njen prevod v ukaze zbirnega jezika. Če strojna koda ne odgovarja nobenemu ukazu zbirnega jezika, je v tretji koloni ponovno izpisana.

V primeru vejitev program izračuna naslov, kamor kaže vejitev.

Uporaba: ko pokličemo program DISS z diskete se ta naloži v pomnilške lokacije O100-OCOO, zato lahko s njim prevajamo ukaze, ki so na ostalih pomnilnih lokacijah.

Če želimo prevesti v zbirnik določen program z diskete (napr. TEST), ga prej naložimo v spomin više od lokacije OCOO (napr. na 2000).

LOAD, TEST, 2000 in potem pokličemo program DISS.

Na zaslonu dobimo:

XX
ISKRADATA 1680 INVERZNI ZBIRNIK DISS

ZA IZPIS NA EKRANU PRITISNI C,
ZA IZPIS NA PRINTERJU PRITISNI L,
VTIPKAJ ZAČETNO IN KONČNO ADRESO
ZA START PRITISNI CR.

XX
Uporabnik vtipka: C 2000 3000 CR (CARRIAGE RETURN)

Na zaslonu se izpiše:

2000	BD	JSR	1E35
2003	OC	CLC	
2004	39	RTS	
2005	EE	LDX	00,X
2007	AD	JSR	35,X
2009	BD	JSR	016C
200C	D4	AND B	F7
200E	D2	SBC B	C6 *

Če izpisujemo na ekran, se izpiše najprej 20 vrstic, nato računalnik čaka na znak LF (LINE FEED), da izpiše novih 20 vrstic. Če pritisknemo tipko RESET se vrnemo v monitorski program.

	Primerik in log	Pedpis	Gredine	Odstopni nivoj, nmr	Tiskalna obdelava	Površ. zadržita	Priporavnost		
Konstr.		Izdaja					1D1680 1.0		
Projekt.		Znak							
Pragliš.		Št. str.							
Števil.		Datum							
Stand.		Pedpis							
Naziv	ZBIRNIK INVERZNI - DISS					List	Stran	J K Identifikacijska številica	
Namenskočnost kopije						1		BDI1SS01MD	
						Arhiv	Morila	Sekcija	Spominska identifikacijska številica

		Priskit. in imo		Podpis		Građevina		Oznakuj sečni. niz		Topografski odabir		Površ. zadržka		Projektno																																																			
																ID 16680-10																																																	
Kodstr.						Tajna																																																											
Projekt.						Znak																																																											
Pregrđ.						St. obr.																																																											
Števil.						Datum																																																											
Stand.						Podpis																																																											
<table border="1"> <tr> <td colspan="8">Naziv</td> <td colspan="8">Identifikacija številke</td> </tr> <tr> <td colspan="8">PROGRAM UREJEVALNIK HH</td> <td colspan="8">BEDNITO1MD</td> </tr> <tr> <td colspan="8">Namenskost kopije</td> <td colspan="2">Atrib</td> <td colspan="2">Skrbita</td> <td colspan="2">Skrbita</td> <td colspan="2">Skrbita</td> <td colspan="2">Skrbita</td> </tr> </table>																Naziv								Identifikacija številke								PROGRAM UREJEVALNIK HH								BEDNITO1MD								Namenskost kopije								Atrib		Skrbita		Skrbita		Skrbita		Skrbita	
Naziv								Identifikacija številke																																																									
PROGRAM UREJEVALNIK HH								BEDNITO1MD																																																									
Namenskost kopije								Atrib		Skrbita		Skrbita		Skrbita		Skrbita																																																	

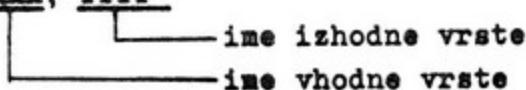
6. PROGRAM UREJEVALNIK EDIT

6.1 Uvod

Program Urejevalnik (EDIT) je namenjen urejanju tekstov (predvsem izvornih programov), ki jih pomnimo na disketi mikroracunalniškega sistema Iskradata 1680. Sistem mora poleg osnovnega mikroracunalnika vsebovati še prikazovalnik s tastaturo, gibki disk in disketni operacijski sistem Iskradata 1680 (FDOS). Urejevalnik sprejema (morebitno) staro besedilo iz vhodne vrste (input file), omogoča vprašanja različnih sprememb, ter generira novo, popravljeno izhodno vrsto (output file).

Urejevalnik kličemo z monitorским ukazom naslednje oblike:

EDIT, XXXX, YYYY



To velja v primeru popravljanja starega besedišča. Pri pisanju novega besedila (programa) pa uporabljamo ukaz oblike EDIT, ,YYYY. Predhodno moramo prikazovalnik s tastaturo preklopiti na "PAGE" način. Na prikazovalniku s tastaturo se pajavi slika, sestavljena iz dveh polj. V zgornjem so pomožne informacije, v spodnjem pa besedilo, ki ga obdelujemo. Istočasno vidimo na zaslonu do 20 vrstic besedila.

Poleg tega vidimo na prikazovalniku s tastaturo utripajoč kazalec (kurzor), ki kaže mesto v besedilu, kjer lahko vnašamo spremembe.

6.2 Urejevalni ukazi

Operater ima na voljo popolno zbirkzo znakov ASCII ter nekaj posebnih tipk in tipkovnih kombinacij.

Pri vtipkavanju znaka ASCII se bo le-ta vrinil v besedilo pred znak pod kazalcem (kurzorjem).

Seznam in pomen posebnih tipk pa je naslednji:

Uredja						Leta	Sistem	J	K	Identifikacijska številka
Št. obveznila						2				BEDIT01MD
	Iskra Elektromehanika PROGRAM UREJEVALNIK Kranj					Atrib				Novejša identifikacijska številka

←	pomik kazalca v desno
→	pomik kazalca v levo
↑	pomik kazalca navzgor
LF	pomik kazalca navzdol
cntrl/B	pomik kazalca na začetek teksta (BEGIN) v pomnilniku
cntrl/E	pomik kazalca na konec bloka
cntrl/K	brisanje vrstice od kazalca dalje (KILL)
cntrl/S	definiranje niza znakov (STRING), ki ga z ukazom cntrl/F iščemo po besedilu. Ta ukaz bomo podrobneje opisali.
cntrl/F	iskanje definiranega niza v besedilu (FIND)
RETURN	vstavitev nove vrstice (vrinjene kombinacije CR/LF)
RUBOUT	brisanje znaka pod kazalcem (ne briše vrstice)
cntrl/N	čitanje naslednjega tekstovnega bloka (NEXT) iz vhodne vrste (input file). Pomen tega ukaza bo podrobneje opisan.
ESC	zaključek urejanja
cntrl/R	Ponavljanje ukazov (←, →, ↑, LF, cntrl/K, RUBOUT).

6.3 Način dela

Najbolj preprosto rokovanje urejevalnikom predstavlja pomikanje kazalca po besedilu s tipkami → ← ↑ in LF. Na iskanem mestu nato v vrstico vrivamo nove znake, ali jih brišemo (RUBOUT).

Nove vrstice generiramo s tipko "RETURN", brisanje vrstic povzroča kombinacija CONTROL/K (NULL).

Pri daljših tekstih je posamezno pomikanje kazalca zamudno. Pomagamo si tako, da določimo niz znakov (string), ki naj ga urejevalnik poišče ter tako lociramo željeno mesto v tekstu. Po pritisku kombinacije CONTROL/S (string) se kazalec preseli v zgornje polje na zaslonu. Operater lahko vtipka do 20 znakov dolg niz, ki ga zaključimo s tipko RETURN. Če se med tipkanjem niza zmotimo, lahko napako odpravimo z zaporednimi vtipkavanji RUBOUT.

Ukaze ponavljamo z kombinacijo cntrl/R, nato sledi vnos števila ponovitev in ukaz, ki ga želimo ponavljati.

Izdelek						Let	Sistem	J	K	Identifikacijska številka
Št. obveznika						3		B	E	D
	Skra Elektromehanika PROGRAM UREJEVALNIK Krani					Atrib				Ramensko identifikacijsko številko

Z ukazom CONTROL/F (find) sprožimo iskanje definiranega niza od kazalca dalje. Če urejevalnik najde enak niz v besedilu, postavi na to mesto kazalec. Tu lahko nato nadaljujemo obdelavo teksta. Če urejevalnik niza ne najde, zapiska in postavi kazalec spet na začetek besedila.

Ukaz CONTROL/F lahko uporabljamo večkratno, vendar moramo vsakokrat kazalec odmakniti od niza v besedilu.

Pri daljših besedilih pride do izraza omejenost tekstovnega delovnega polja v hitrem pomnilniku. Editor rešuje ta problem tako, da ima v tem primeru le po en blok teksta. Ko končamo z urejanjem tega teksta, lahko s tipko CONTROL/N (NEXT) sprožimo zapis bloka v izhodno vrsto ter včitanje naslednjega bloka po izvajanju naslednjih blokov ne moremo več vrniti na absolutni začetek. (1. blok) besedila.

V tem primeru moramo editor zaključiti s tipko ESC in postopek ponoviti.

Z ukazom CONTROL/E sproži iskanje konca bloka in postavi kazalec na to mesto. Kombinacije uporabljamo tedaj, če želimo hitro priti do konca teksta (v drugem primeru kombiniramo CONTROL/E in CONTROL/N).

6.4 Obvestila

V zgornjem polju na zaslonu se poleg niza pojavljajo še občasna obvestila operaterju. To so:

ČITAM Urejevalnik bere z diskete blok podatkov
NAPAKA Vtipkali smo neregularno tipkano kombinacijo

NA ROBU Kazalec želimo pomakniti izven besedila

KONČUJEM Obvestilo po vtipkanju ESC

PIŠEM Po pritisku na kombinacijo CONTROL/N urejevalnik vpisuje na disketo popravljeni blok podatkov

Uredaja	St. obvestila	Ust	Stran	J	K	Identifikacijska številka
	skra Elektromehanika PROGRAM UREJEVALNIK Kranj	4				BEDITD01MD

Namenske identifikacijske številke

4	1											
3	1											
2	1											
1	1											

LIST	STRAN	IZDAJA	LIST	STRAN	IZDAJA	LIST	STRAN	IZDAJA
LIST	STRANA	IZDANJE	LIST	STRANA	IZDANJE	LIST	STRANA	IZDANJE
SHEET	PAGE	ISSUE	SHEET	PAGE	ISSUE	SHEET	PAGE	ISSUE

Pravni učinkovitost izdaje je omejena na vrednost navedenih podatkov.

X	Primak in ime	Podpis	Gradivo		Odstopi netol. mer		Toplotna obdelava		Površ. zaščita	Prilagodljivost			
Konstr.		Izdaja								ID1680			
Projekt		Znak											
Pregled		Št. obr.											
Stevil.		Datum											
Stand		Podpis											
Naziv	MONITOR - MONI							List	Stran	J	K	Identifikacijska številka	
								1				BMON101MD	
Namembnost kopije								Arhiv	Meno	Sekcija	Namesto identifikacijske številke		



Iskra
Elektromehanika
Kranj

13

PROGRAM MONITOR - MONI

Program MONITOR (MONI) omogoča uporabniku opazovanje pomnilnika (pokaži, postavi) startanje programa, izvajanje programa korak za korakom, izpisovanje procesorjevih registrov, itd. S pomočjo ukaza (programska prekinitve) kličemo podprograme: izhod, vhod, prekinitve itd.

UPORABA: Ko pokličemo program MONI z diskete se ta naloži v pomnilniške lokacije 000-0650, zato lahko z njim izvajamo programe, ki smo jih predhodno naložili z komando LOAD na višje pomnilniške lokacije.

Na prikazovalniku dobimo:

! MONI

(Dejanski start programa je na naslovu 0100)

Primer prikaza: Pokaži podatke od naslova 2000 do naslova 201F:

D2000_201F(CR) (SP)
2000 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20
2010 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 30

Primer vpisa: Vpiši na naslov 1000 podatek 77, na naslov 1001 podatek 88 in na naslov 1002 podatek 99. Naslov se sam poveča za eno po vpisu podatka. V eno vrsto lahko vpišemo šestnajst podatkov. Če se po vpisu podatek ne vpiše pravilno na naslov se pojavi vprašaj ob podatku.

S10000_77_88_99(CR)

Primer start: Program na naslovu 1500 startamo

G1500(CR)

Primer podprogramskega starta: Podprogram (ki se zaključuje z instrukcijo RTS (39)) na naslovu 1FO0 startamo
T1FO0(CR)

Po izvajanju programa na naslovu 1FO0 se vrnemo v monitorski program.

Izdača						Let	Stran	J	K	Identifikacijska številka
St. obveznika						2				BMONI 01MD
	Iskra Elektromehanika Kranj	MONITOR - MONI				Arhiv		Ramesta identifikacijske številke		

Primer izpisa registrrov: Izpiši vse registre mikroprocesorja
R (CR)
R C4 FF FF 1000 1000 FFFF kjer so registri

C B A IX PC SP C pogojni register
A, B akumulatorja
13x indeksni register
PC programski števec
SP skladovni kazalev

Primer izvajanja korak za korakom: Program na pomnilniški lokaciji
2FOO izvajamo korak za korakom
K2FOO(CR)

C B A IX PC SP
C4 00 3F 3000 2001 FFBF

S pritiskom na tipke na tastaturi nadaljujemo izvajanje. Pritisik
tipke "K" na tastaturi nas vrne v monitorski program.
Korak za korakom moremo izvajat samo v RAM pomnilniku (podprogra-
mi, ki navadno v EPROM se izvršijo JSR). Vektorjev IRQ, SWI ne
smemo spremenijat v programu.

Primer starta s prednastavljivo: Vpišemo vsebino v akumulatorja
A in B, indeksni register in nastavimo vrednost programskemu štev-
cu s tipko "G" na tastaturi startamo program kot glavni program,
s tipko "T" na tastaturi startamo program kot podprogram (zaklju-
čen podprogram z RTS). Nastavimo akumulator B na 00, akumulator A
na 10, indeks register na 1000 in programski števec na 1100 dobimo:

J (CR)
B A IX PC ? " G" ? "T"
00_10_1000_1100G ali T, za start

Primer povratka v Iskra FDOS:

E (CR)

Monitorski podprogrami:

- \$11 - vsebina akumulatorja A se prenese na prikazovalnik
- \$12 - izpiši tekst kamor kaže indeks register do kode ETX (04)
- \$14 - vsebina s tastature prikazovalnika je v akumulatorju A

Izdaja						List	Stran	J	K	Identifikacijska številka
Št. obvestila						3		B	MONI	01MD
	Iskra Elektromehanika Kranj	MONITOR	MONI							Namensko identifikacijska številka

- \$80 - izpiši mikroprocesorjeve registre (C,B,A,IX,PC,SP) in se vrni v monitorski program
- \$81 - izpiši mikroprocesorjeve registre (C,B,A,IX,PC,SP) in nadaljuj program

Vsi podprogrami se izvajajo s pomočjo instrukcije SWI (programska prekinitve - \$3F).

Primer stalnega izpisa registrov na naslovu 1000:

S1000 3F 81 7E 10 000 (CR)

SWI \$81 JMP 10 000

G1000 (CR)

Indaja						Leto	Stran	J	K	Identifikacijska številka
St. ovesnila						4				BMON101MD
	Iskra Elektromehanika Kranj	MONITOR	MONI				Arhiv	Namenjate identifikacijske številke		

Instituto acelham se aponta como o maior centro de ensino nista do país.

8. RAČUNALNIKOV TESTNO KONTROLNI PROGRAM - DEBUG

8.1. Kontrolna plošča, ki je sestavni del sistema ID 1680, je izvrsten pripomoček za pepravljanje (čiščenje) in kontrolo programov. Vsebuje vse funkcije, ki so navadno na računalnikovi kontrolni plošči:

prikaz in spreminjanje vrednosti registerov in pomnilnika ter prekinitve na določenih naslovih.

Izhodna enota je prikazovalnik s tastaturo nastavljen na "PAGE" način. Vsaka vrsta ima v prvem stolpcu svoj identifikator - kontrolni znak, ki označuje tip vrste, ki mu sledi. Za testiranje kontrolne plošče (DEBUGGER-ja) je dovolj če pritisnemo "RESET" ali CTRL/NUL. Ob vstopu v testno kontrolni program (DEBUGGER) se na zaslonu prikaže osnovni položaj, ki izgleda takole:

```
DEBUG BRK=FFFF-00 P=F500- 36 A600 A=4B B=0C X=FFEC S=FFEB C=C0
M 0000 D0 D7 D8 D9 DA DB DC DD DE DF EO E1 E2 E3 E4 E5
P F500 36 A6 00 36 B6 FC 2B A7 01 B6 FC 2A A7 00 B6 FC 6 6
X FFEC F0 4B OC EA 57 E1 3D 97 81 F0 BE F2 F0 E8 EO FF
S FFEB D0 F0 4B OC EA 57 E1 30 97 81 FD BE F2 F0 F8 EO
```

Sl. 8.1. Vstop v testno kontrolni program (DEBUGGER).

1. VRSTA

DEBUG Vidimo, da je kontrolni znak D. Temu sledi prikaz stanja testno kontrolnega programa.

Prikazane so vrednosti vseh registerov procesorja 6800. Na začetku vrste se namesto besede DEBUG med delom kontrolnega programa DEBUGGER-ja pojavlja tudi INTRPT (kadar sledi vstop v testno kontrolni program (DEBUGGER) za NMI, ali pri ustavitvi na prekinitveni točki "BREAK - POINT"), ali SNAP (kadar je uporabnikov program še vedno v teku, prikazan pa je izgled testno kontrolnega programa (DEBUGGER-jev) največkrat po SNAPSHOT zahteyi).

Indeks							List	Stran	J	K	Identifikacijska številka
Št. davčnika							2		V	46325043	
	skra Elektromehanika Krani	PROGRAM TESTNO KONTR.					Avtor				Namensko identifikacijska številka

BRK Naslednja postavka v vrsti prikazuje trenutno vrednost prekinitvenega registra - 16 bitni naslov kateremu sledi vsebina pomnilniške lokacije, ki jo ta register naslavlja. Vse vrednosti testne kontrolnega programa so prikazane v heksadecimalnem zapisu. BRK = FFFF pomeni, da uporabnik ni definiral vrednosti za prekinitveno točko(breakpoint). Kadarkoli se med izvajanjem programa pojavi prekinitvena točka, vrednost prekinitvenega registra predstavlja naslov instrukcije ali podatka) se tekoča instrukcija izvrši do konca, nato se generira NMI. Če NMI vektor še vedno naslavljaja testno kontrolni program (DEBUGGER), ta prikaže svojo vsebino in čaka na uporabnikove ukaze.

P Programski števec: vsebini P sledi vsebina treh memorijskih lokacij, ki jih P naslavlja. Te vrednosti predstavljajo naslednjo instrukcijo in operanda /kadar sta/. Če vrednost P spremenimo, bo nadaljni program startal z nove lokacije.

A,B,X, Ostale postavke v 1.vrsti so registri A,B,X,S in C.

S,C Vsak register sovpada z registerom v mikroprocesorju. C register je prikazan s heksadecimalno vrednostjo in posamezni znaki, ki mu sledijo in označujejo imena bitov, ki so v stanju 1 /H,I,N,Z,V,C/. Gl.sl. 8.2.

```
DEBUG    BRK=FFFF- 00 P=F500- 36 A600 A=00 B=0C X=FFEC S=FFEE C=F
M 0000 00 D7 D8 D9 DA DB DC DD DE DF E0 E1 E2 E3 E4 E5      HINZVC
P F5D0 36 A6 00 36 B6 FC 2B A7 01 B6 FC 2A A7 00 B6 FC
X FFEC F0 4B 0C EA 57 E1 3D 97 81 F0 BE F2 F0 E8 E0 EF
S FFEB D0 F0 4B 0C EA 57 E1 30 97 81 FD BE F2 F0 E8 E0
```

Slika 8.2. prikazuje modificirana reg. A in C

VRSTE 2 - 5 Ostale vrste /2-5/ predstavljajo spominske lokacije . Prvi stolpec /M,P,X in S/ je kontrolni znak za tip vrste /opisan kasneje/. Temu sledi 16-bitni naslov za njim pa je 16 polj podatkov, ki prikazujejo vrednosti 16 zaporednih spominskih lokacij/ začenši z lokacijo, ki jo naslavljaja začetni naslov v vrstici. Na skrajni desni je 16 ASCII znakov - znakovni zapis vsebine spominskih lokacij. Izjema so kontrolni znaki /\$00-\$1F/ in neASCII

Indeks						Ust	Stres	J	K	Identifikacijska številka
Št. obveznika						3				V46325043
	Iskra Elektromehanika Kranj	PROGRAM TESTNO KONTR.								Namensko identifikacijsko številko

znaki /\$80-\$FF). ki se ne ispišejo.

8.2. DELO S TESTNO KONTROLNIM PROGRAMOM (DEBUGGER-jem)

Delo s testno kontrolnim programom je preprosto. Kazalec kaže na mesto, kjer uporabnik komunicira s testno kontrolnim programom(DEBUGGER-jem). Za spremjanje katerekoli pomnilniške lokacije ali register premikamo kazalec s kontrolnimi puščicami in LF do lokacije, ki jo želimo spremeniti in vtipkamo novo vrednost. Na primer: vrednost registra A spremenimo tako, da vnesemo novo heksadecimalno vrednost za reg.A. Na enak način lahko uredimo katerikoli del prikaza testno kontrolnega programa na zaslonu (DEBUGGER - jevega display-a).

Osnovne vrste kontr. test. programa (DEBUGGER-ja) so prikazane na sliki 6.1. Ostale 16-besedne bloke pomnilnika lahko pregledamo, če v prvi vrsti stolpec naslednje vrste vpišemo znak Min začetni naslov bloka, ki ga pregledujemo. Na začetku druge vrste vidimo M 0000, kar pomeni, da je pomnilniška lokacija 0000 prikazana v naslednjih dveh stolpcih. Tej sledi prikaz nadaljnih 15-tih lokacij. To pomeni, da ta vrsta prikazuje lokacije od \$0000 do \$000F.

Prvi znaki v ostalih treh vrstah so imena 16-bitnih registrov, njim sledijo njihove vrednosti, ki podobno kot v drugi označujejo začetek 16-besednega bloka v pomnilniku.

8.2.1. PRIMER MODIFIKACIJE REGISTROV

Za boljše razumevanje, kako ti registri sledijo pomnilniku, premaknimo kazalec k vrednosti, ki predstavlja X register in vpišemo novo vrednost za indeks register. Ko je nova vrednost vpisana spremenijo tudi vrednost v poljih za prikaz pomnilniških lokacij, ki sedaj prikazujejo nove lokacije. Uporabnik lahko modificira svoj lastni prikaz z dodajanjem novih vrst ali s spremjanjem starih vrst prvega znaka v že obstoječih vrstah, ki pa jih lahko izbršemo z zaslonom da v 1. stolpec vtipkamo kontrolni znak RUBOUT".

Izdeja						Let	Stran	J	K	Identifikacijska številka
Št. obveznila						4		V	46325043	
	skra Elektromehanika	PROGRAM TESTNO KONTR.				Arhiv				Namenska identifikacijska številka

8.2.2. DODAJANJE NOVIH VRSTIC

Poleg že znanih lahko dodajamo tudi nove tipove vrst, od katerih sta največ uporabljeni "plus" in "minus" vrsti /"+" in "-"/. Ti dve vrsti sta podobni M-vrstam, vendar predstavlja lokacije za 16 večje kot prejšnja vrsta /+vrsta/ oziroma za 16 manjše kot naslednja vrsta /-vrsta/. Na spodnji sliki imamo 5 vrst, ki prikazuje spomin, vse pa so kontrolirane z naslovom, ki je v eni sami M-vrsti.

DEBUG BRK-FFFF- 00 P-F5D0- 36 A600 A=4B B=0C X=FFEC S=FFEB C=00

M 0000	DO D7 D8 D9 DA DB DC DD DE DF EO E1 E2 E3 E4 E5
P F5D0	36 A6 00 36 B6 FC 2B A7 01 B6 FC 2A A7 00 B6 FC
X FFEC	F0 4B 0C EA 57 E1 3D 97 81 F0 BE F2 F0 E8 EO FF
S FFEB	DO F0 4B 0C EA 57 E1 30 97 81 FD BE F2 F0 E8 EO
- 00FO	41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 ABCDEFGH
M 0100	41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 ABCDEFGH
+ 0110	51 52 53 54 55 56 41 42 43 44 45 46 47 48 49 4A QRSTUVAB

Slika 8.3. "M", "+" in "-" dodane vrste

8.2.3. SPREMINJANJE VREDNOSTI POMNILNIŠKIH LOKACIJ

Da lahko spremenimo vsebino spomina, moramo imeti M-vrsto, ki prikazuje željeno področje spomina. Kazalec premaknemo k heksadecimalnim ali ASCII vrednostim in vtipkamo novo vrednost. Poskus spremembe ROM lokacije je ignoriran.

8.2.4. ZAPUSTITEV KONTROLNEGA PROGRAMA (DEBUGGER-ja)

Za povratek v program, ki ga popravljamo, na tekočo "P" lokacijo, uporabimo tipko ETX. Pred tem pa lahko sprememimo vrednost P registra kot tudi katerikoli drug podatek.

a.) POSAMEZNI KORAK

Da zapustimo kontrolni program (DEBUGGER) samo za toliko, da

Izdaja						List	Stran	J	K	Identifikacijska številka
št. obvestila						5				V46325043
	Skra Elektromehanika Kranj	PROGRAM TESTNO KONTR.				Arhiv				Namensko identifikacijska številka

program, ki ga čistimo, opravi eno samo instrukcijo, uporabimo CONTROL/Q; če želimo da vsaka BSR ali JSR instrukcija opravi rutino, ki jo kliče do konca RTS in še eno instrukcijo/, vse ostale instrukcije pa naredijo le posamezni korak, uporabijo CONTROL/W.

b.) "SNAPSHOT"

Med izvajanjem lastnega programa lahko sprožimo testno kontrolni program DEBUGGER samo za toliko, da nam prikaže svoj izgled. To možnost nam daje "SNAPSHOT" v trenutku, ko pritisnemo tipko ETX/terminal mora biti v "PAGE" načinu/.

c.) PROGRAMSKE PREKINITVENE TOČKE (SOFTWARE-ske)

V strojni zapis RAM programa lahko vstavimo dodatne prekinitvene točke. Za to uporabimo novi tip vrste: B - vrsto. V vsaki B-vrsti lahko ustvarimo sedem programskih naslovov ali pa jih pustimo na FFFF; s tem povemo, da jih ne potrebujemo. Kadar program med svojim potekom pride do katere teh lokacij, se pojavi SWI in kontrolo dobi testno kontrolni program (DEBUGGER). Resnične SWI instrukcije/in njim pripadajoči vektorji/ se s tem ne spremenijo, toda po potrebi jih lahko obidemo z uporabo programskih prekinitvenih točk.

8.2.5. VRSTE S KOMENTARJEM

Da si ustvarimo vrste s komentarjem, za kontrolni znak uporabimo zvezdice (*). To je posebno uporabno za "zamrznitev" vrste s predhodnimi vrednostmi testno kontrolnega programa /DEBUGGER-ja/. S H-vrsto si ustvarimo "glavo". Tako imamo boljši pregled nad spominskimi lokacijami, ki jih pregledujemo.

```
DEBUG BRK*FFFF- 00 P=F508- 36 A600 A=00 B=0C X=10000 S=FFEB C=FF
                                                HINZVC
```

```
B      FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
```

Slika 8.4. B-vrsta brez SWI prekinitvenih točk

Izdaja					Let	Stran	J	K	Identifikacijska številka
St. obvestila					6				V46325043
	Skra Elektromehanika PROGRAM TESTNO KONTR. Kranj	Arhiv							Namensko identifikacijska številka

DEBUG BRK-FFFF- 00 P-F 5D0- 36 A600 A=01 B=0C X=2000 S=FFEB C=00
 ■ vrsta s komentarjem lahko vsebuje uporabnikov komentar ali sta-
 ro vrednost DEBUG-a.

 ■ BUG BRK-FFFF- 00 P-F5D0 36 A600 A=00 B=0C X=10000 S=FFEB C=FF

 ■ 0100 20 0E 55 50 00 03 54 52 41 43 45 20 56 31 2E 30 UPDJTRACE

 ■ - "H" vrsto uporabljam za izdelavo glave spomina
 H ADDR 0 1 2 3 4 5 6 7 8 9 A B C D E F ... ASCII
 M 01F0 09 31 43 4B 50 59 20 0C 54 52 45 4E 41 40 08 A8...COPY..

Slika 8.5. "■" in "H" vrste

8.3. PREGLED UKAZOV TESTNO KONTROLNEGA PROGRAMA

I VSTOP V DEBUGGER

CRT/NULL - Aktiviranje testno kontrolnega programa
 (DEBUGGER-ja).

ETX - SNAPSHOT

II. PROGRAMSKA KONTROLA

ETX - Povratek v program

CONTROL/Q - Posamezni korak

CONTROL/W - Posamezni korak-dovoljuje kompletiranje pod-
 rutin

III. UREJANJE VSEBINE TESTNO KONTROLNEGA PROGRAMA

(DEBUGGER- ja)

LINE FEED - Premik kazalca eno vrstico navzdol

CONTROL/Z - Premik kazalca eno vrstico navzgor

← - Premik kazalca en znak levo

CONTROL/U - Premik kazalca en znak desno

CONTROL/B - Premik kazalca v najnižjo vrsto

CONTROL/T - Premik kazalca v najvišjo vrsto

CONTROL/L - Premik kazalca v skrajni levi stolpec

CONTROL/R - Premik kazalca v skrajni desni stolpec

Izdaja						list	Stran	J	K	Identifikacijska številka
St. obvestila						7		V	46325043	
	Iskra Elektromehanika	PROGRAM TESTNO KONTR.	Kranj			Avtiv				Namensko identifikacijska številka

CARRIAGE RETURN - Premik kazalca na začetek naslednje vrste
 CONTROL/E - Premik kazalca na začetek prejšnje vrste
 CONTROL/Y - Premik kazalca na "home" položaj /levo
 zgoraj/
 CONTROL/H ali
 ENTER+ - Premik kazalca na začetek naslednjega
 polja
 CONTROL/M ali
 ENTER- - Premik kazalca na začetek prejšnjega pol-
 ja
 RUB OUT - Brisanje vrste ali spremembe
 Kadar ga uporabimo na začetku vrste, bri-
 šemo s tem celo vrsto ali pa izničimo
 zadnjo spremembo / preden začnemo spremen-
 jati drugo polje/ .

8.4. TIPI VRST

- B - Programske prekinitvene točke
- D - Status kontrolnega programa(DEBUGGER-ja)
■ namesto D pomeni, da komunikacija z
testno kontrolnim programom ni mogoča
- F - Podrutine funkcij tipk (ni še razvito)
- H - Vrsta z "glavo"
- M - Pommilnik
- P - Programski števec
- S - Skladovni kazalec
- "-" - Sledi naslednji vrsti
- "+" - Sledi prejšnji vrsti
- "■" - Komentar ali "zamrznjena" vrsta
- E - Napaka v kontrolnem programu (DEBUGGER-ja)
rezervirano/
- L - Prikaz LOG vmesnega registra

8.5. POMOŽNA PRIPOROČILA

Na zaslonu prikazana informacija sovpada s trenutnim stanjem računalnika, razen med delnimi spremembami polj: če želimo spremeniti besedo v pomnilniku s heks. 01 na heks. 23 medtem ko je kazalec pod "1" in je na zaslonu prikazana vrednost

Leta	Stran	J	K	Identifikacijska številka
Si obvezila	8			V46325043
Iskra Elektromehanika PROGRAM TESTNO KONTR Kranj	Arhiv	Nemesto identifikacijske številke		

21 je dejanska vrednost te lokacije še vedno 01. Zamenjava se izvrši, ko kazalec zavzame drugo polje, bodisi zaradi končane zamenjave v celiem polju, ali zaradi premika kazalca s kontrolnimi tipkami → , CR, itd.

Določene ID 1680 strojni naslovi ne smejo biti prikazani s testno kontrolnimi programi(DEBUGGER -jem), to so lokacije PIA in ACIA ali spomin, ki ga naslavljajo strojni register preko prekinitev /FBFA,FBFB/. Testno kontrolni program navadno upravlja strojni register preko prekinitev, tako da ga drži v monitor ROM naslovu in ga spremeni na željo strojne prekinitvene točke tik pred izhodom.

CONTROL/W dovoljuje izvršitev cele podrutine, če je naslednja instrukcijo, ki sledi klicu podrutine. Če se rutina ne vrne /manjka RTS/, ponovnega vstopa v testno kontrolni program ne bo. Če se rutina vrne, se bo izvršila tudi instrukcija, ki neposredno sledi klicu podrutine, preden bo dobil kontrolo nazaj testno kontrolni program. Njegov naslov bo strojna lokacija prekinitvene točke. Ker prvotne izvedbe MPU 6800 niso mogle hkrati rešiti NMI in SWI, se NMI ne sme pojaviti med izvajanjem SWI instrukcije. Da to lahko omogočimo, ne smemo nastaviti strojne prekinitvene točke na mesto SWI ali eno besedo naprej. Posamezni korak mimo SWI je mogoč, ker kontrolni program nastavi register prekinitrov na lokacijo, ki jo naslavlja SWI vektor. Prav tako je možna uporaba programskih točk na pravih SWI instrukcijah. Uporabnikova modifikacija instrukcije v SWI ali direktno prevajanje le-te ne bo povzročila vloge prekinitvene točke. Vendar pa nam pregled pomnilnika brez uporabe testno kontrolnega programa pokaže "3F" /SWI/ na mestu, kjer smo napravili prekinitveno točko.

Ko pregledujemo program po korakih, IRQ prekinitve ne bodo upoštevane. Testno kontrolni program med izvajanjem po korakih onemogoči IRQ prekinitev, potek po korakih pa je kontroliran z NMI, ki ima višjo prioriteto kot IRQ. Kadar spustimo program po korakih, je strojna prekinitvena točka še vedno na instrukciji, ki smo jo nasadnje izvedli po korakih, kot je označeno na zaslonu. Če na tak način pregledujemo zanko, se pri polni hitrosti prekinitev pojavi zelo hitro, če med tem nismo spremenili vrednost BRK.

Izdeja					Let	Stran	J	K	Identifikacijska številka
Št. obveznika					9				V46325043
	Skra Elektromehanika PROGRAM TESTNO KONTR. Kranj				Avtor				Namenska identifikacijska številka

Napačen vstop v testno kontrolni program bo sledil, če se pojavi IRQ od enote za katero to ni pričakovano ali kadar je strojna prekinitvena točka nastavljena na SWI instrukcijo.

Navodilo: prikazovalnik s tastaturo CDC 752, ki je uporabljen v sistemu 1680, lahko med drugim dela v PAGE ali SCROLL načinu. Prednost nadaljnega izpisa teksta na zaslonu, ko so že vse vrste izpisane, je boljša če je prikazovalnik s tastaturo v SCROLL načinu. Samo dva programa : testno kontrolni program in urejevalnik zahteva PAGE način, sicer je vsebina zaslona popolnoma nepregledna in za operatorja ne-uporabna.

Izdeja							Ust	Stran	J	K	Identifikacijska številka
Št. obveznila							10				V46325043
 Iskra Elektromehanika PROGRAM TESTNO KONTR. Kranj							Avtor				Namesto identifikacijske številke