

6S0.1
M52

Autorid: A. Amenberg, T. Arulaane, J. Freiberg, R. Haavel, V. Hanson, K. Jaaksoo, E. Jürviste, A. Kaukver, J. Lankots, K. Märtn, R. Paluoja, S. Siibak, T. Tommingas, I. Treikelder, T. Tõnspöeg

Koostanud K. Märtn
Retsenseerinud dots. L. Võhandu
Kaane kujundanud H. Puzanov

M 52 Mikroarvuti «Juku» kasutamisesjuhend / T. Arulaane, T. Tõnspöeg, R. Haavel jt.; koostanud K. Märtn. — Tln.: Valgus, 1988. — 168 lk., ill.

ISBN 5-440-00143-3

«Juku» on väike lauarvuti, mille kasutusalaad võivad olla küllalt mitmekesised. «Jukule» on jõukohased: arvutustööd, arvutiõpe ja õppetöö näitlikustamine, süsteemide juhtimine, eksperimendi automatiseerimine, andmetöötlus, tekstitöötlus ja bürooautomaatika. Eelkõige aga on «Juku» mõeldud kasutamiseks koolides arvutiõpetuse õppevahendina.

M 240500000—062 TL—7—9—86
902(15)—88

32.81

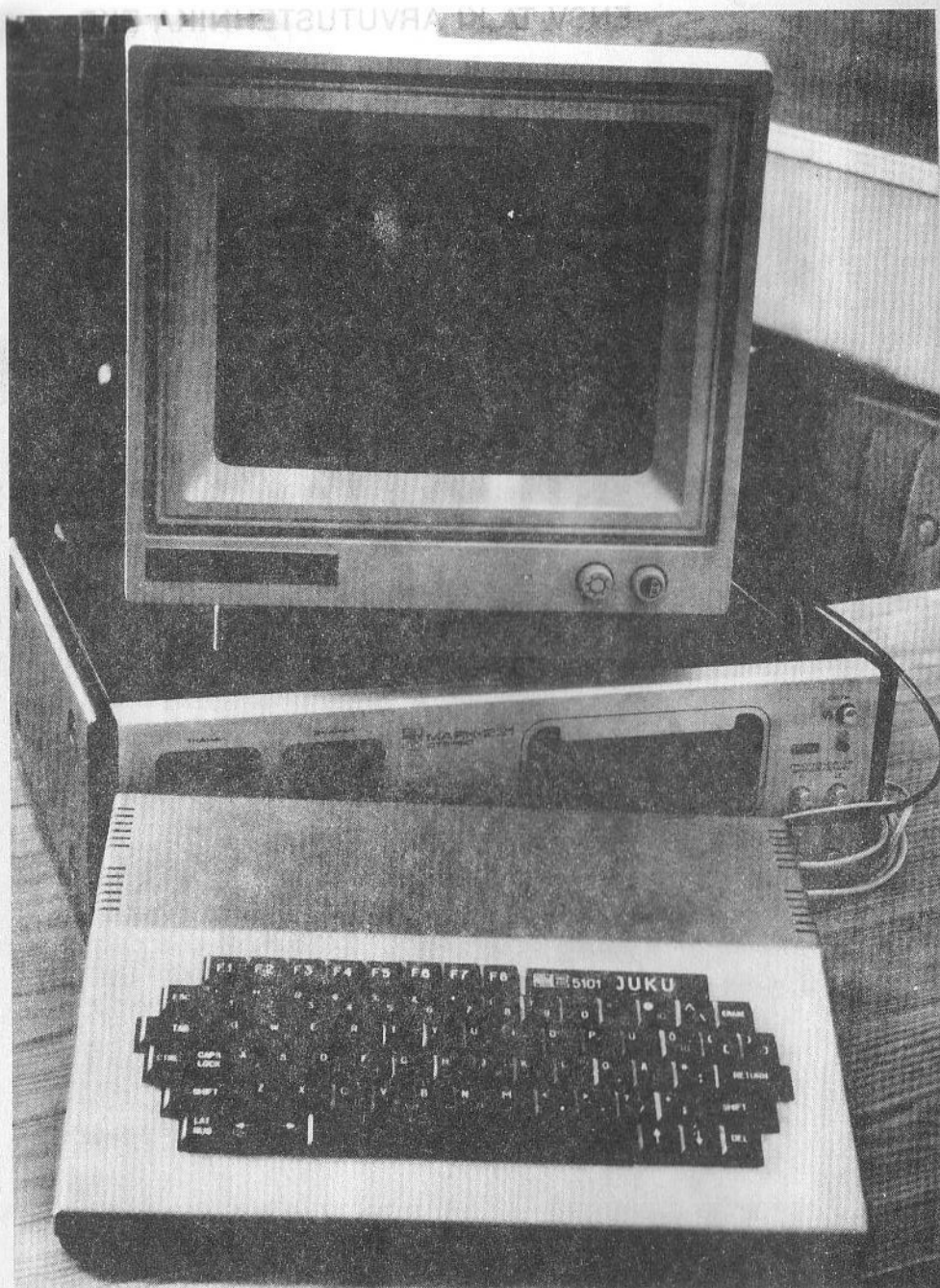
ISBN 5-440-00143-3

© Kirjastus «Valgus», 1988

ENSV TA KI ARVUTUSTEHNİKA EKB

MIKROARVUTI «JUKU» KASUTAMISJUHEKD

TALLINN «VALGUS» 1988



Magnetofoni ja teleriga ühendatud «Juku»

«Juku» on väike lauaarvuti, mille kasutusvaldkonnad võivad olla küllalt mitmekesised. «Jukule» on jõukohased: — arvutustööd, — arvutiõpe ja õppetöö näitlikustamine, — süsteemide juhtimine, — eksperimendi automatiseerimine, — andmetöötlus, — tekstitöötlus ja bürooautomaatika.

Eelkõige aga on «Juku» mõeldud kasutamiseks koolides arvutiõpetuse õppevahendina.

Kasutaja teeb end mõistetavaks tärgklaviatuuriga, mis võimaldab kasutada nii suuri kui väikesi vene ja ladina tähti, numbreid ja erimärke.

Et «Juku» saaks väljastada infot kasutajale, tuleb temaga ühendada tavaline must-valge teler, mis täidab üheaegselt nii tärg- kui ka graafilise kuvari funktsioone. Välissalvestiks on harilik magnetofon. «Juku» komplekti kuuluv kasutusjuhend sisaldab põhiteatmeid aparatuuri kohta, süsteemitarkvara (püsimonitori, operatsioonisüsteemi, translaatorite, siluri, redaktori, teenindus- ja testprogrammide, andmesidemonitoride) programmistijuhendid ning juhised arvuti ülesseadmiseks ja kasutamiseks.

Arvuti tööprintsiipide ja põhimõtteskeemide üksikasjalikum selgitus sisaldub aparatuuri tehnilise kirjelduse köites, süsteemitarkvara detailne kirjeldus aga süsteemprogrammistijuhendis.

0 PÕHITEATMEID

0.1 STRUKTUUR

«Juku» on ehituselt paindlik, kasutaja saab varieerida tema koostist vajaduste ja võimaluste kohaselt (vt. joonis 0.1).

Keskseadme põhiplokk on magistraalstruktuuriga ning sisaldab mikroprotsessori, püsi- ja muutmälu, mitmesuguseid ajastuslülitusi, välisseadmete ühendamise liideseid ja põhisiini pikendamist võimaldava puhvri.

Keskseadme koostises olev **videogeneraator** koos kõrgsagedusmodulaatoriga võimaldab kasutada kuvarina harilikku telerit. Eelistatud on mustvalget telerit tunduvalt teravam ja vähem silmi väsitava kujutise tõttu.

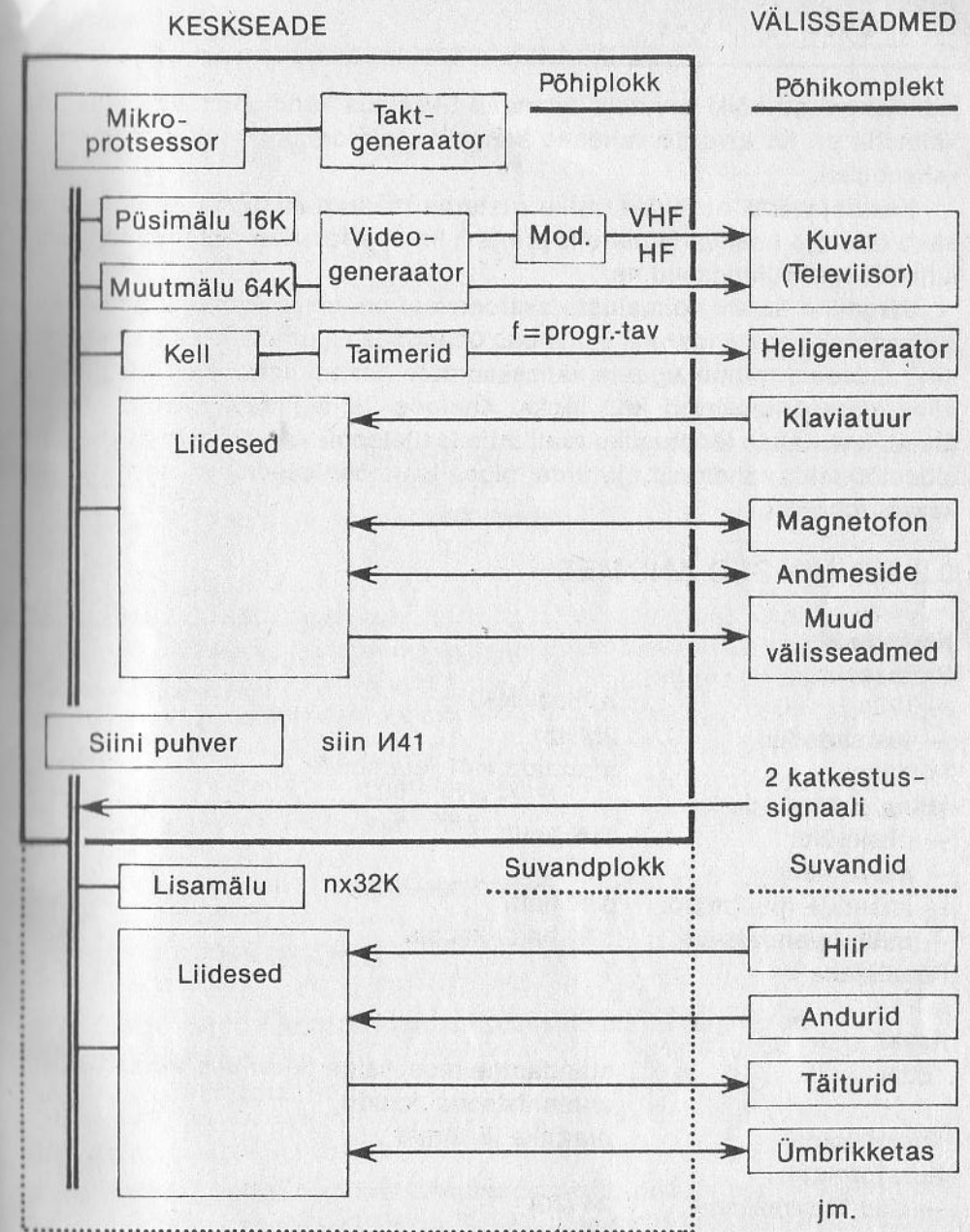
Programmeeritav **taimer** koos sisseehitatud valjuhääldiga võimaldab tekitada mitmesuguseid heliefekte ning kasutada arvutit ühehäälsel muusikariistana.

Reaalajakella baasil saab realiseerida täpse ajanäitaja, koos katkestussüsteemiga võimaldab ta aga mingi objekti raaljuhtimist reaalajas.

Välisseadmeid saab komplekteerida üsna vabalt, tingimata tuleb keskseadmega ühendada ainult **klaviatuur** ja **kuvar** (=teler). Nende seadmete vahendusel toimub dialoog arvutiga. Arvutis kasutusel olevad ladina ning vene suur- ja väiketähed võimaldavad paindlikku tekstitöötlust, graafikarežiim aga mitmesuguste kujundite esitust ja töötlust.

Välissalvestiks võib olla standardne **magnetofon**; see laiendab arvuti kasutamismõimalusi tunduvalt, eriti kui magnetofon on kaugjuhitav. Nii saab kasutada tunduvalt võimsamaid tarkvaravahendeid, luua andmebanku ja programmeerida.

Andmeside liideseid on vahend lauarvutite kohtvõrgu rajamiseks. Sellises võrgus saab õpetaja tõhusalt jälgida õpilaste suhtlemist arvutiga ja sekkuda sellesse vajaduse korral oma arvuti kaudu, sooritada kontrolltõid



Joonis 0.1. «Juku» konfiguratsioon

jne. Kohtvõrgu kõiki arvuteid ühendab üksainus kahejuhtmeline sideliin. Võimalik on ka arvutite vaheline kaugside telefoniliini kaudu, modemite vahendusel.

Kasutajavärat on liides, mille otstarbe määrab kasutaja — selle abil saab arvutiga ühendada näiteks printeri, lindiperforaatori või muu digitaaljuhtimisega väljundseadme.

Järgmine samm võimaluste avardamisel on keskseadme laiendamine **suvandplokiga**: standardne siin lubab ühendada lisamälu (32K baidi kaupa) ning liideseid mitmesuguste välisseadmete (ümbrikketta, kursorit juhtiva hiire, kõnesüntesaatori jm.) jaoks. Analoo- ja digitaalsignaali liides annab kasutajale tööstusliku raaljuhtimissüsteemi vahendid, millega saab sidustöödelda andmeid, juhtida signalisatsiooniseadmeid, makette ja kasvõi roboteid.

0.2 TEHNILISED ANDMED

Keskseade

Protsessor:	
— tüüp	KP580IK80
— taktsagedus	2MHz
Põhisiin:	standard И41 (alamhulk)
Mälu, põhimoodulis:	
— püsिमälu:	16K baiti
— muutmälu:	64K baiti
— kasutaja muutmälu:	53K baiti
— mälu laiendatavus:	32K baidi kaupa
Reaalajakell:	1

Kuvar

Tüüp:	standardne must-valge televiisor, video- või antennisisendi kaudu
Kuvatüübid:	graafika ja tärgid
Kuvaformaad:	
— ridu tärgirežiimis:	24 (20)
— tärke reas:	40 (64)
— graafikaraster:	320x240 (384x200) punkti
Tähestikud:	ladina ja vene suur- ja väiketähed
Kuvamälu:	10K baiti, muutmälu ühiseväljal

Klaviatuur

Klahve:	73
Sisestuse kviteerimine:	akustiline

Liidestus

Magnetofoni liides:	
— edastuskiirus:	2400 boodi
— magnetofoni programme juhtimine koos failide kiirrotsingu süsteemiga	
Kohtvõrgu liides:	
— sideliini tüüp:	juhtmepaar
— kohtsiini pikkus:	kuni 50 m
— edastuskiirus:	kuni 9600 boodi
Modemiliides:	C2
Kasutajaväratid:	
— sisendid ja väljundid:	24
— väratid koormatavus:	1 TTL-sisend
Kasutajakatkestusi:	2

Toide

Pinge:	220V-10%, 50Hz
Võimsustarve:	kuni 20W

Gabaridid:	350x300x70mm
------------	--------------

0.3 KONSTRUKTSIOON

Mikroarvuti «Juku» konstruktsiooni põhisõlmed on järgmised:

- 1) arvutimoodul,
- 2) klaviatuurimoodul,
- 3) toitemoodul,
- 4) kest.

Arvutimoodul kujutab endast üheplaadiarvutit, mis on realiseeritud trükkplaadil mõõtmetega 300x250 mm. Sellele plaadile on monteeritud sisuliselt kõik arvuti põhielemendid ning pistmikud väliste seadmete ühendamiseks.

Klaviatuurimoodul on trükkplaat mõõtmetega 140x310 mm sellele monteeritud 73 klahviga, mis moodustavad arvuti klaviatuuri.

Toitemoodul formeerib arvuti tööks vajalikke toitepingeid (+5; -12; +12 volti). Toitemooduli elemendid on monteeritud trükkplaadile, mis on paigutatud suletud metallkarpi mõõtmetega 220x115x40 mm.

Kõik kolm kirjeldatud moodulit on monteeritud **plastmasskesta** mõõtmetega 350x300x70 mm.

0.4 MÄLUJAOTUS. ADRESSEERIMISMOODUSED

«Juku» mälu koosneb 16K (laiendatav kuni 64K) baidist püsimalust ja 64K baidist muutmälust. 10K baiti muutmälust moodustab nn. videomälu, kuhu salvestatud info põhjal formeeritakse teleri ekraanil videokujutis.

Et kasutatav mikroprotsessor (KP580IK80) võimaldab otseselt adresseerida mälu ainult 64K baidi ulatuses, siis on aadressruumi efektiivse kasutamise ja ka mälu edasise laiendamise huvides võetud kasutusele 4 erinevat mälu adresseerimise moodust. Vajaliku mooduse valik toimub süsteemivärati (aadress 02H) kahe madalaima bitiga, vastavalt järgnevale tabelile:

D1	D0	Moodus
0	0	0
0	1	1
1	0	2
1	1	3

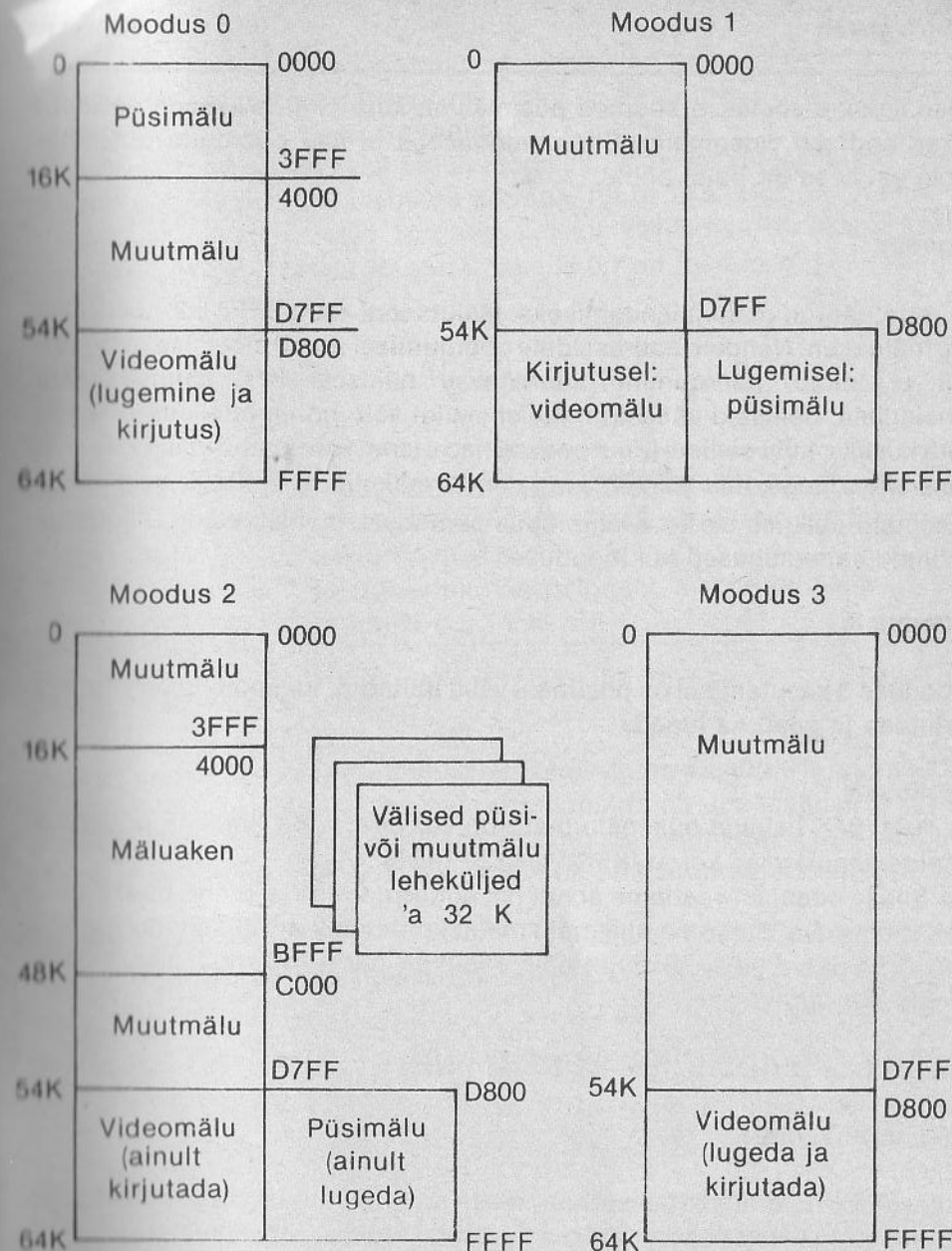
Joonisel 0.2 on toodud mälujaotuse skeem iga adresseerimismooduse jaoks.

Moodus 0

Rakendub arvuti sisselülitamisel kohe pärast RESET-signaali. Aadressruumi esimese 16K baidi ulatuses paiknevad paralleelselt muut- ja püsimalu. Lugemiskäsuga loetakse andmed püsimalust, kirjutamiskäsuga salvestatakse andmed muutmällu. Muutmälu viimased 10K baiti (õigemini 640 baiti vähem) hõlmab videomälu.

Moodus 1

Selle moodusega algab muutmälu aadressist 0000. Aadressruumi viimase 10K baidi korral sõltub mälu liik pöörduoperatsiooni tüübist. Mälust luge-



Joon.0.2 Mälujaotuse plaanid

mise käsuga võetakse andmed püsimalust, kuid kirjutuskäsuga salvestatakse andmed videomällu. Selle moodusega ei saa pöörduda püsimaluploki esimese 6K baidi poole.

Moodus 2

On ette nähtud mälu laiendamiseks. Mälutsooni 4000-BFFF formeeritakse nn. mäluaken. Nendele aadressidele pöördumisel arvuti sisemise mälu valikut ei toimu, pöördumine suunatakse näilisele 32K baidise mälu-leheküljele. Selliseid väliseid mälulehekülgi võib põhimõtteliselt olla rohkem kui üks, kuid sellisel juhul peavad nad olema varustatud vastavate riistvaravahenditega, mis võimaldavad nende valikulist avamist ja sulgemist. Püsimalu paikneb aadressvälja lõpus paralleelselt videomäluga. Pöördusrežiimid samasugused kui mooduses 1.

Moodus 3

Mooduse 3 kasutamisel on püsimalu välja lülitatud. Videomällu on võimalik kirjutada ja sealt ka lugeda.

Märkused:

Kuni 64K baidine püsimalu maht on saavutatav 8K baidiste püsimalu-kiipide kasutamise korral (K573PΦ4, K573PΦ6).

Siin ja edaspidi vaatleme arvuti nn. põhivarianti — s. o. 16K püsimalu ja 64K muutmälu. Suurema püsimalu mahu korral nihkub «allapoole» mooduses 0 näidatud püsi- ja muutmälu vaheline piir. Ülejäänud mälujaotuse plokid jäävad samaks.

0.5 VÄLISSEADMETE ÜHENDAMINE

0.5.1 VIDEOLIIDES

Videoliides on ette nähtud standardse teleri (soovit. must-valge) ühendamiseks arvutiga. Telerit kasutatakse kuvarina, kuhu arvuti väljastab tarbijale vajaliku informatsiooni.

Teleri ühendamiseks on 2 võimalust. Pistmikusse X6 on toodud kompleksne madalsageduslik videosignaali (VIDEO), mis tuleb anda teleri videovõimendi sisendisse. Pistmikusse X7 on toodud moduleeritud kõrgsage-

dussignaali (HF), kandesagedusega $f = 215$ MHz, mis tuleb anda teleri antennisisendisse ja häälestada teler 11. kanalile.

Parema kujutise ekraanil tagab madalsagedusliku videosignaali kasutamine. Väljundeid võib kasutada korraga, nii et üks teler on ühendatud madalsagedusliku ja teine kõrgsagedusliku väljundiga. Koaksiaalkaabli lainetakistus peab olema 75 oomi. Tabelis 0.1 on toodud videosignaali spetsifikatsioonid:

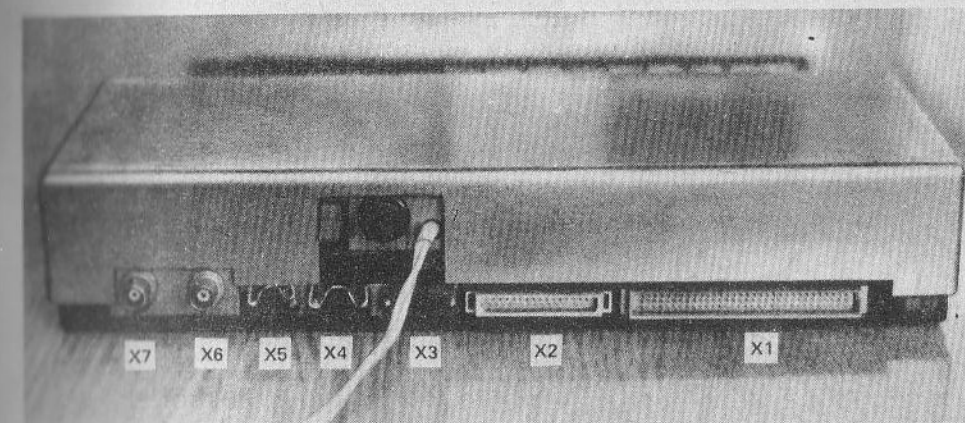
Videoliidese väljundid

Tabel 0.1

Pistmik	Signaal	Kirjeldus
X6	VIDEO	Kompleksne videosignaali, amplituud 2V. Sünkronivoo on madal
X7	HF	Moduleeritud videosignaali, amplituud 50 mV, $f = 215$ MHz (11. TV kanal)

0.5.2 MAGNETOFONI LIIDES

Magnetofoni liides on ette nähtud välissalvestina kasutatava standardse tarbimagnetofoni (eeldatavalt kassetmagnetofoni) ühendamiseks arvutiga. Andmevahetus toimub kiirusega 2400 boodi. **Magnetofon võib olla juhitud või mittejuhitav.** Juhitaval magnetofonil on distantsjuhtimisliides. See



«Juku» tagakülje pistmikumid

võimaldab tarbijal kasutada lindioperatsioonisüsteemi (LOS) kõiki võimalusi, mis tunduvalt hõlbustavad arvuti ning magnetlindi kui välise andmekandja kasutamist.

Magnetofon ühendatakse arvutiga kaabli abil pistmike X4 ja X5 kaudu.

Magnetofonisignaali spetsifikatsioon on tabelites 0.2 ja 0.3.

Mittejuhitava magnetofoni kasutamisel ühendatakse magnetofon ainult pistmiku X5 abil.

Magnetofoni juhtsignaalid

Tabel 0.2

X4 kontakt	Signaal	Funktsioon
1	FF	Edasikerimine
4	RW	Tagasikerimine
3	PLAY	Taasesitamine
2	REC	Salvestamine
5	STOP	Lindi peatamine
8	TAPE RUN	Impulsid lindi liikumise andurilt
7	CNTR CHECK	Magnetofoni juhitavuse tunnus
6	GND	Maa

Signaalid FF, RW, PLAY, REC, STOP on TTL-avakollektorväljundid ($U_1 = 15V$; $I = 40mA$). Aktiivne seisund vastab madalale nivoole. Impulsid TAPE RUN saadakse magnetofoni lindi liikumise andurilt (Autostopi formeerimise skeem). Kui arvutiga ühendatakse juhitud magnetofon, siis tuleb pistmikus X4 signaal CNTR CHECK ühendada maa siiniga (6. kontakt). Arvuti mõistab, et tegemist on juhitud magnetofoniga.

Magnetofoni andmesignaalid

Tabel 0.3

X5 kontakt	Signaal	Kirjeldus
2	REC.DATA	Salvestatav signaal $U = 300mV$
4	PLAY DATA	Loetav signaal $U = 300mV$
3	GND	Maa
1	SYNC	Sünkrosignaal, jadasignaali bititakt

0.5.3 SIDELIIDES

Sideliides kujutab endast standardse liidese C2 allvarianti ja on ette nähtud samasugust liidest omavate seadmete ühendamiseks arvutiga. Kõne alla tulevad modemid, printerid, teised arvutid, mitmesugused mõõteseadmed jne.

Lisaks standardsetele C2-signaalidele kuulub sideliidese koosseisu veel signaal OC SOUT. Viimane on avatud kollektoriga jadaväljund ja on mõeldud eelkõige suhteliselt lähedal paiknevate arvutite (näiteks õppeklassis) ühendamiseks lihtsasse arvutite kohtvõrku.

Sideliidese järjestikkanal on ette nähtud tööks asünkroonses režiimis. Andmeedastuskiirus on programmiliselt valitav vahemikus 200..–9600 boodi.

Tabelis 0.4 on toodud sideliidese signaalide täielik spetsifikatsioon. Signaalide koodid vastavad ISO soovitusel V24. «1» = $-3 \dots 12V$, «0» = $3 \dots -12V$.

Sideliidese signaalid

Tabel 0.4

X3 kontakt	Signaal	Kirjeldus	Kood, ISO V24(C2)
02,12	OC SOUT	Jadaväljund, avatud kollektor $U_{max} = 30V$, $I_{o,max} = 100mA$	—
08	SOUT	Jadaväljund	103
07	GND	Signaali maa	102
4	SIN	Jadasisend	104
10	RTS	Saate nõue (väljund)	105
11	DTR	Arvuti on sideks valmis (väljund)	108.2
05	CTS	Valmisolek arvutist väljastatavate andmete vastuvõtuks (sisend)	106
06	DSR	Andmeedastusseadme valmisolek (sisend)	107
01	POLL UP	Liini sobitustakisti	—

0.5.4 KASUTAJAVÄRAT

Arvutimooduli koosseisu kuulub üks **programmeeritav sisendväljundkiip KP580IK55**, mille A-, B- ja C-kanali kõik väljaviigud on toodud pistmikule X2. See kiip moodustabki kasutajavärati, mis on mõeldud mitmesuguste mittestandardsete sisend-väljundseadmete ühendamiseks arvutiga, kusjuures kasutaja käsutuses on kiibi KP580IK55 kõik funktsionaalsed võimalused. **Tabelis 0.5 on esitatud pistmikule X2 toodud signaalide loetelu.** Tuleb meeles pidada, et kiibi sisendid ja väljundid on võimendamata, s.t. iga väljundit võib koormata ainult ühe standardse TTL-sisendiga.

Kasutajavärat

Tabel 0.5

X2 kontakt	Signaal	Kirjeldus (täidab kasutaja)
1	2	3
08	A0	
06	A1	
01	A2	
02	A3	
04	A4	
03	A5	
05	A6	
07	A7	
21	B0	
23	B1	
25	B2	
26	B3	
24	B4	
22	B5	
20	B6	
18	B7	
13	C0	
15	C1	
17	C2	
19	C3	

1	2	3
11	C4	
12	C5	
09	C6	
10	C7	
16	SOUT2	Magnetofoniliidese võimendatud jadaväljund ($\pm 10V$).
29,30	Maa	
27	+5V	Koormatav kuni 0,3 A

0.6 ARVUTI LAIENDAMINE

Arvuti laiendamiseks täiendavate sisend-väljund- ja mälumoodulitega on pistmikule X1 toodud arvuti võimendatud aadressi-, andme- ja juhtsiin. Pistmiku kommutatsioon ja signaalide spetsifikatsioonid vastavad standardsele liidesele I41 (Multibus 1).

Pistmikusse X1 on peale loogikasignaale toodud ka toitepinged +5V, +12V ja -12V. Nii saab lihtsamat laiendusmoodulit toita otse arvutist. Pistmiku X1 andmed on esitatud tabelis 0.6.

Arvuti põhisiin

Tabel 0.6

X1 kontakt	Signaal	Kirjeldus
24C	-ADR0	ADR0 ... ADRF moodustavad arvuti aadressisiini
24B	-ADR1	
23C	-ADR2	
23B	-ADR3	
22C	-ADR4	ADR0 ... ADRF on väljundid
22B	-ADR5	
21C	-ADR6	
21B	-ADR7	
20C	-ADR8	
20B	-ADR9	
19C	-ADRA	
19B	-ADRB	
18C	-ADRC	
18B	-ADRD	
17C	-ADRE	
17B	-ADRF	

X1 kontakt	Signaal	Kirjeldus
32C	– DAT0	DAT0 ... DAT7 moodustavad arvuti kahesuunalise andmesiini, üldiselt väljundid. Muutuvad sisenditeks, kui toimub lugemine välisest salvestist või sisend-väljundseadmest
32B	– DAT1	
31C	– DAT2	
30C	– DAT4	
30B	– DAT5	
29C	– DAT6	
29B	– DAT7	
04C 02B 04B 05C 05B	– MRDC – AMWC – MWRC – I0RC – I0WC	Arvuti juhtsiin, väljundid
09B	– I0/M	Juhtsignaal. Näitab, kas antud tsüklis toimub pöördumine mälu või sisend-väljundseadme poole. Kui väärtus on «1», siis on tegemist mäluseadmega
01B	– INIT	Toite lülitumisele järgneva üldlähtestuse signaal. Väljund.
07B	– BLOCK	Blokeeriv signaal, mille toimel blokeeritakse arvuti sisemiste mälu- ja sisend-väljundseadmete valimine, antud tsüklis loetav andmebait loetakse väliselt andmesiinilt. Sisend. Mõeldud kasutamiseks arvuti diagnostikas ja programmide silumisel
06C	– XACK	Pöörduse kviteerimissignaal välisseadmeilt. Sisend.
06B 11C	– INHB CC LCK	Keelab välise mäluseadme. Väljund. taktisignaal 2 MHz
01,02,03, 04A	GND	Maa siin
06,07,08A 08B	+5V	Toide +5V, koormatav kuni 0,3 A
31, 32 A	+12V	Toide +12V, koormata 50 mA
10, 11 A	–12V	Toide –12V, koormata 50 mA
08C		
13C	– INT6	Kasutaja katkestussignaal
13B	– INT7	Kasutaja katkestussignaal

Märkus: Kõik tabelis toodud signaalid on madala aktiivse nivooga, millele viitab märk «-» signaali nimetuse ees.

KASUTAMISJUHISED

1 ARVUTI ÜLESSEADMINE

«Juku» jõuab kasutajani komplektina, kuhu kuuluvad arvuti ise ning juhtmistik tema ühendamiseks välisseadmetega (teleri ja magnetofoniga).

Kui arvuti on lahti pakitud, võib alustada tema ülesseadmist. «Juku» tagaküljel on 7 tähistatud pistmikku:

- X1 — arvuti siin
- X2 — kasutajavärat
- X3 — andmeside pistikupesa
- X4 — magnetofoni juhtimise pistikupesa
- X5 — magnetofoni andmeside pistikupesa
- X6 — kõrgsagedusväljund
- X7 — videoväljund

Peale selle on arvuti tagaküljel veel võrgulüli ja RESET-klahv. Tagaküljelt väljub ka «Juku» toitejuhe.

Ülalnimetatud pistmikke on lähemalt kirjeldatud arvuti tehnilises üldiseloostuses. Ülesseadmisel tuleb neid kasutada alljärgnevalt.

NB! Enne arvuti ühendamist televiisori ja magnetofoniga tuleb veenduda, et kõik ühendatavate seadmete võrgujuhtmed on vooluvõrgust väljas.

1.1 TELERI ÜHENDAMINE

Teler ühendatakse «Jukuga» arvuti komplekti kuuluva koaksiaalkaabliga. See, milliseid pistmikuid ja pesi ühendamiseks kasutada, oleneb olemasoleva teleri tehnilistest võimalustest.

1. Kui teleril on olemas videosisend, ühendatakse kaabli abil «Juku» videoväljund X7 (VIDEO) teleri videosisendiga.

2. Kui teleril videosisend puudub, ühendatakse sama kaabli abil «Juku» kõrgsagedusväljund X6 (HF) teleri antennisisendiga.

Eelistada tuleb videosisendit saadava kujutise parema kvaliteedi tõttu. Pärast seda, kui arvuti on ühendatud ka magnetofoniga vastavalt jaotisele 1.2, tuleb teler sisse lülitada ning häälestada kanalile 11. Kui arvuti on sisse lülitatud, nagu on kirjeldatud jaotises 1.3, võib teleri pilti paremaks reguleerida kõigi kasutaja käsutuses olevate teleri vahenditega.

1.2 MAGNETOFONI ÜHENDAMINE

Magnetofoni ühendamiseks arvutiga on «Juku» komplektis vastav kaabel. Juhitava magnetofoni olemasolul kasutatakse arvuti pistikupesasid X5 ja X4 vastavalt andmesideks magnetofoniga ja magnetofoni juhtimiseks.

«Juku» ühendamisel magnetofoniga tuleb jälgida kaabli pistikutel olevaid mnemoonilisi sümboleid.

1.3 ARVUTI SISSELÜLITAMINE

Nüüd on «Juku» ühendatud teleri ja magnetofoniga (viimase olemasolul), teler sisse lülitatud, häälestatud kanalile 11 ja piisavalt soojenenud.

Järgmiseks ühendatakse «Juku» tagaküljelt väljuv toitejuhe voluvõrgu pistikku. Seejärel lülitatakse arvuti sisse tagapaneelil oleva tumblersi **CETb** abil. «Juku» esiküljel peab süttima valgusdiod, mis näitab, et arvuti toide on sisse lülitatud; kuvarile peab ilmuma püsimonitori teade **MONITOR**.

Kui valgusdiod küll süttis, aga püsimonitor midagi ei kuva, võib vajutada arvuti tagaküljel olevale klahvile **RESET**.

2 ETTEVALMISTUS TÖÖKS

Mikroarvutit «Juku» on võimalik kasutada nii juhitava kassettmagnetofoniga kui ka ilma selleta. Arvuti ettevalmistus tööks oleneb kasutatavast konfiguratsioonist; magnetofoniga võib tööle hakata alles seejärel, kui «Juku» ilma magnetofonita on töökorda seatud.

2.1 ETTEVALMISTUS TÖÖKS PÜSIMONITORIGA

Arvuti kasutamisel püsimonitoriga on meie käsutuses monitor oma direktiividega ja püsimalu-BASIC, mis võimaldavad küll juhtida arvuti tööd, kuid puudub võimalus programmide salvestamiseks ja valmisprogrammide kasutamiseks.

«Juku» sisselülitamisel, nagu kirjeldatud peatükis 1, ilmub kuvarile püsimonitori teade **MONITOR**, monitori versiooni number ja viip *. **Viip** on programmi poolt väljastatav teade, mis näitab, et programm ootab kasutaja edasisi direktiive. Eri programmidel on ka erinevad viibid, mis on kirjeldatud nende kasutamisesjuhistes.

Kohta kuvaril, kuhu ilmuvad klaviatuurilt sisestatud sümboolid, märgib ristkülikukujuline plokk — **kursor**. «Juku» kursor võib olla kas vilkuv või mittevilkuv. Vilkuv kursor näitab, et klaviatuur on ladina tähestiku režiimis, mittevilkuv kursor aga, et ta on vene tähestiku režiimis. Lähemalt on ladinavene ja suur-/väiketähtede kasutamisest juttu jaotises 7.4.

Väiketähtede režiimis saab sisestada suurtähti, kui hoida all klahvi **SHIFT** ja vajutada soovitud klahvile. Tärgikoodide arvu piiratuse tõttu saab osa tärke sisestada ainult väiketähtedena. Nendeks on eesti tähestiku tähed ö, õ ja ü.

2.2 ETTEVALMISTUS TÖÖKS LOS-iga

Kui «Juku» on varustatud kassettmagnetofoniga, võib arvuti tööd juhtida lindiooperatsioonisüsteem LOS. LOS-i kasutamine võimaldab tarvitada valmisprogramme ning salvestada programme ja andmeid kassetidele.

Et alustada tööd LOS-iga, tuleb arvuti koos magnetofoniga üles seada vastavalt peatükile 1 ning seejärel seada ta töösse püsimonitoriga, nagu on kirjeldatud jaotises 2.1.

Enne magnetofoniga töö alustamist tuleb reguleerida õige salvestusnivoo (-3dB) ja jälgida, et magnetofon ei oleks ajutise peatamise režiimis.

Magnetofoni esmakordsel kasutamisel koos arvutiga tuleb salvestussüsteemi tingimata kontrollida ja reguleerida programmi **TTEST** abil (vt. 15.6). Edasise töö hõlbustamiseks võib selle programmiga saadud õige salvestusnivoo märgistada reguleerimisnuppudel, kasutades hiljem neid märgiseid. Aeg-ajalt on siiski soovitatav kontrollida salvestusnivood **TTEST**-iga.

NB! Väära salvestusnivoo korral võib lindi sulgemine (operatsioon CLOSE) rikkuda kogu lindi.

«Jukuga» on kaasas LOS-i süsteemne kassett, millele on salvestatud lindioperatsioonisüsteem ise ning mõningad arvuti tööks ja programmvara väljatöötamiseks vajalikud valmisprogrammid. Enne tööle asumist tuleks süsteemsest kassetist teha koopia (süsteemne osa programmiga SYSGEN, vt. 8.3). Soovitatav on seda koopiat kasutada igapäevases töös süsteemse kassetina ja originaali säilitada kindlas kohas juhuks, kui kasutatava süsteemse kassetiga midagi peaks juhtuma.

Kui magnetofon on üles seatud ja sisse lülitatud ning «Juku» töötab püsimonitoriga ja ootab korraldusi, tuleb magnetofoni asetada süsteemne kassett ja anda monitorile käsk T. Selle tulemusel loetakse LOS lindilt mälusse ja antakse talle juhtimine üle. LOS-i eduka käivitumise tunnuseks on teade kuvaril:

ENSV TA KÜB.I AT EKB

LINDIOPERATSIOONISÜSTEEM

ja LOS-i viip

A>

Viip näitab, et LOS ootab kasutaja direktiive. Nendeks võivad olla kas LOS-i residentsete direktiivide nimed (vt. peatükki 8) või lindil olevate valmisprogrammide nimed.

Enne kui lindilt võib midagi lugeda või sinna kirjutada, tuleb lint **avada**, s.t. lugeda lindil olevate failide kataloog mällu. Lint avatakse LOS-i direktiiviga **OPEN**. Seejärel võib lindil olevate failide nimesid kuvada direktiiviga DIR, sooritada nendega teisi operatsioone, mis on kirjeldatud peatükis 8 ning käivitada valmisprogramme. Valmisprogrammide tunnuseks on failinime laiend (vt. 8.1) **COM**.

Kui töö käigus on lindile midagi kirjutatud, tuleb lint töö lõpetamisel **sulgeda** direktiiviga **CLOSE**. Lindi sulgemisel kirjutatakse uuendatud kataloog jälle mälust lindile. Kui töö käigus on linti midagi kirjutatud, lint aga eemaldatakse teda eelnevalt sulgemata ja alustatakse tööd uue lindiga või lõpetatakse töö LOS-iga, **läheb kõik töö käigus linti kirjutatu kaotsi**.

Kui püsimonitori direktiiviga T küll mündi linti lugema, kuid kuvarile ilmub jällegi monitori viip *, tähendab see LOS-i sisselugemise ebaõnnestumist ja seda toimingut tuleb korrata, s.t. anda monitorile veelkord direktiiv T. Kui pärast mitmekordset LOS-i sisselugemise üritust see ikkagi ei õnnestu, on kõige tõenäolisemalt tegu vigase süsteemse lindiga.

Kui LOS-iga töötamise käigus hakkab suhteliselt tihti esinema veateateid (CHECKSUM ERROR, BLOCK NOT FOUND jt.), siis on see märgiks, et on viimane aeg magnetofoni helipead puhastada või vahetada kasutusel olev lint uuema vastu.

Kui lindi avamisel esineb veateateid (TIME OUT ja DIRECTORY BLOCK NOT FOUND), tuleb kohe, kui õnnestub linti siiski avada, vajalikud failid kopeerida uutele lintidele ja vormindada käesolev lint uuesti.

LOS-ile direktiivi andmisel võib seda eelnevalt kuvaril redigeerida ja LOS-ile saata ta alles siis, kui ta on õige. Seejuures kasutatakse järgmisi klahve:

RETURN	Saadab korralduse LOS-ile täitmiseks
← või CTRL H	Viib kursori ühe märgi võrra tagasi ja kustutab selle
CTRL X	Viib kursori rea algusse ja kustutab kogu sisestatava rea

Linte, mida «Juku» andmete ja programmide salvestamiseks kasutab, tuleb eelnevalt **vormindada**. Vormindamise käigus luuakse lindile kataloog, kirjutatakse lindile andmeplokid ja kontrollitakse pärastise ülelugemisega kirjutatud plokkide õigsust, s.t. lindi erinevate osade kõlbulikkust sinna salvestamiseks. Vormindamine garanteerib, et andmeid ei kirjutata lindi füüsiliselt kahjustatud osadesse, kust neid oleks hiljem võimatu õigesti kätte saada. Et vormindamisel kirjutatakse kõik lindi plokid täis, siis eelnevalt kasutatud lindi vormindamisel kustutatakse sealt kogu kasulik info. Täpsemalt on vormindamist programmiga **FORMAT** kirjeldatud jaotises 7.3.

Soovitatav on iga lindi algusesse kirjutada ka opsüsteem ise. See võimaldab kasutada LOS-i lugemiseks mällu suvalist linti ja kasutada enamikel juhtudel ainult üht linti korraga. LOS-i salvestamiseks lindile on olemas süsteemne programm **SYSGEN**, mille kasutamise täpsem kirjeldus on jaotises 7.3.

Vene tähestiku kasutamiseks tuleb käivitada programm RUS.

2.3 KORRASOLEKU KONTROLL

Kui «Juku» koos kassetmagnetofoniga on töökorda seatud ja ta töötab LOS-iga, on õige aeg kontrollida tema põhisõlmede töökorda. Selleks on LOS-i komplektis olemas testprogrammide pakett «Diagnostics», mis testib arvuti protsessori, mälu, kuvari ja (nende olemasolul) ümbrikettaseadmete korrasolekut.

Neid teste on kaht tüüpi: kiirtest, mida kasutatakse ekspluatatsiooni käigus leitud tõrgete uurimiseks, ja põhjalik, kogu arvutit hõlmav test. Et põhjalik testprogramm töötab väga pikka aega (üle 13 tunni), pole teda mõtet eriti sageli kasutada. Küll on aga selle testprogrammi käivitamine kasulik «Juku» esialgse ülesseadmise järel, sest ta võimaldab avastada ka sääraseid arvuti elektroonika tõrkeid, mis jäävad kiirtesti eest varjule. Testimise puhul tuleb jälgida peatükis 10 toodud järjekorda (1. mälu, 2. protsessor, 3. kuvar) ja testprogrammide kasutamishuiseid.

3. VALMISPROGRAMMIDE KASUTAMINE

Valmisprogrammid on kirjutatud assembler- või kõrgkeeles ja transleeritud seejärel protsessori K580IK80 poolt täidetavasse masinakoodi. Programmi käivitamisel nime järgi otsitakse lindilt fail antud nimega ja laiendiga COM.

Näide:

A> PROG

käivitab programmi, mis asub antud lindil ja mille nimeks on PROG.COM. Programmifail laaditakse mällu alates aadressist 100H ning juhtimine antakse üle sellele aadressile, s.t. programm alustab tööd.

Harilikult on programmist väljumine programmi töö lõppedes automaatselt või ette nähtud erilise käsuga, mis on kirjeldatud programmi kasutamishuistes. Kui aga mingil põhjusel (näiteks vigase programmi puhul või kasutamishuiste puudumisel) ei õnnestu programmi tööd lõpetada, on selleks 4 moodust.

1. CTRL C võib katkestada mõningate programmide töö juhul, kui seal on vastav kontroll olemas. Juhtimine antakse sel juhul üle LOS-ile.
2. CTRL ESC katkestab iga programmi täitmise ja annab juhtimise üle LOS-ile või, töös püsimalu-BASIC-uga, BASIC-u interpretaatorile.
3. CTRL SHIFT ESC katkestab iga programmi täitmise ja annab juhtimise üle püsimonitorile. Kui enne katkestamist oli lindile midagi salvestatud ja lint ei olnud suletud, lähevad kõik need failid kaotsi. Programmi töö katkestamist selle klahvide kombinatsiooniga tuleks seetõttu kasutada ainult **erandjuhul**, kui LOS on mälus rikutud ja ei suuda arvuti tööd juhtida.
4. RESET-klahv katkestab iga programmi täitmise ja annab juhtimise üle monitorile. Töö jooksul loodud failid sulgemata lindil lähevad kaotsi. Seetõttu tuleks ka RESET-klahvi kasutada ainult erandjuhul.

4 PROGRAMMIDE KIRJUTAMINE

4.1 PROGRAMMEERIMISKEELE VALIMINE

Mikroarvuti «Juku» on varustatud vahenditega programmeerimiseks kolmes programmeerimiskeeles:

- BASIC-us
- PL/M-is
- assembleris

Programmeerimiskeele valikul tuleb lähtuda kolmest põhilisest parameetrist

- programmi poolt vajatav mälumaht
- programmi töökiirus
- programmeerimiseks kuluv aeg

4.1.1 ASSEMBLERKEEL

Kõige efektiivsema programmi saab kirjutada assemblerkeeles. Assembleri puuduseks on aga programmi kirjutamiseks ja silumiseks kuluv aeg, mis on palju pikem kõrgkeelte kasutamisel kuluvast ajast. Seetõttu tuleks enamiku programmeerimisülesannete lahendamisel eelistada kõrgkeeli. Programmilõikude jaoks, mis vajavad eriti suurt töökiirust, võib assembler- ja kõrgkeeli omavahel siduda.

4.1.2 PL/M

PL/M-keele translaator annab suhteliselt efektiivse masinakeelse programmi. PL/M-i eelisteks on moodul- ja plokkstruktuur, mis võimaldavad kasutada moodsat struktuurprogrammeerimise meetodit. Ka PL/M annab võimaluse juhtida arvuti tööd madalal tasemel, on võimalik juurde pääseda protsessori registritele ja aadresside järgi arvuti mälule, samuti välja kutsuda assemblerkeeles kirjutatud alamprogramme. PL/M-i tuleks eelistada «Juku» põhilise süsteemprogrammeerimiskeelena ning samuti rakendusprogrammeerimisel suurema praktilise tähtsusega programmide puhul.

4.1.3 BASIC

Püsimalu-BASIC on interpreteeriv keel. Tema puuduseks on madal töökiirus transleeritud keeltega võrreldes, eelisteks aga kasutamise mugavus ning programmist suure tööviljakus programmide kirjutamisel ja silumisel. BASIC-u teisteks eelisteks on kõrgekeele tasemel olemasolevad vahendid tööks stringidega ja graafikakäsud. Samuti on olemas vahendid arvuti sisend-väljundväraati poole pöördumiseks, mälu poole pöördumiseks aadressi järgi ja assemblerkeeles kirjutatud alamprogrammide väljakutsumiseks.

BASIC-ut tuleks eelistada suhteliselt lihtsamate programmide kirjutamisel, mille puhul töökiirus ei ole kriitiline, samuti programmeerimise õppimisel.

4.2 ALGTEKSTI SISESTUS

Enne programmi transleerimist ja käivitamist on vaja sisestada tema algtekst ja säilitada see lindil. Sellel etapil kasutatavad vahendid sõltuvad tarvitavast programmeerimiskeelest.

4.2.1 BASIC-KEELSE ALGTEKSTI SISESTUS

Et püsimalu-BASIC-us on võimalus programmiridade sisestamiseks ja kustutamiseks interpretaatoriga töötades, pole BASIC-u puhul otseselt vaja kasutada teist tekstiredaktorit. BASIC-programm sisestatakse ridadena, mille ees on reanumbrid. Programm säilitatakse mälus reanumbrite kasvamise järjekorras; selles järjekorras teda tsükli ja siirete puudumisel ka täidetakse.

Soovitav on, et algteksti rea numbrid kasvaksid 10-kaupa. See võimaldab hiljem programmi vajalikesse kohtadesse ridu vahele lisada.

Ühe programmirea asendamiseks teisega tuleb sisestada uus rida sama reanumbriga, mis oli vanal real.

Rea kustutamiseks tuleb sisestada sama reanumbriga tühi rida.

Kui programmi tekst on sisestatud, tuleb ta salvestada lindile direktiiviga **CSAVE «< failinimi >»**, sealjuures ei või failile määrata failinime laiendit. BASIC ise annab programmifailile nime laiendi **BAS**.

BASIC-programmi algteksti võib sisestada ka näiteks tekstiredaktoriga

EDIT, nagu on kirjeldatud jaotises 4.2.2. Sealjuures tuleb jälgida, et iga rida algaks reanumbriga, sest BASIC ei luba sisendfailis kasutada otserežiimi käsku.

4.2.2 PL/M- JA ASSEMBLERKEELE ALGTEKSTI SISESTUS

PL/M ja assemblerkeelse algteksti sisestuseks on olemas tekstiredaktor EDIT, mis lubab klaviatuurilt sisestatavat teksti redigeerida ja pärastiseks kasutamiseks lindile kirjutada.

Redaktori väljakutsumisel tuleb talle anda redigeeritava faili nimi. Näiteks soovides redigeerida PL/M-keelset programmi MATEM.PLM tuleb redaktor välja kutsuda järgmiselt:

A> EDIT MATEM.PLM

Kui näidatud nimega faili enne lindil ei olnud, ilmub kuvarile teade

NEW FILE

EDIT-i viibaks on teade

* EDIT COMMAND:

millele vastuseks tuleb algteksti sisestusel anda direktiiv I. Selle tulemusel läheb EDIT sisestusrežiimi ja algteksti sisestus võib alata. EDIT näitab sisestamise käigus iga sisestatava rea ees selle järjenumbrit. Lindile salvestatavasse faili neid numbreid ei kirjutata.

Viimase sisestatud märgi ja kogu sisestatava rea kustutamiseks võib kasutada vastavalt sõrmiseid <- ja **DEL**.

Sisestusrežiimi lõpetab **CTRL Z**, misjärel sisestatud faili võib täiendavalt redigeerida, või kirjutada lindile direktiiviga **W**.

Kui fail on lindile kirjutatud, võib direktiiviga **X** redaktorist väljuda.

4.3 PROGRAMMEERIMISVAHENDID

4.3.1 BASIC-u INTERPRETAATOR

BASIC-programmi interpreteeritakse töö käigus ning seetõttu paikneb interpretaator mälus BASIC-programmiga samaaegselt. See loob väga hea võimaluse programmide kirjutamiseks ja silumiseks, puuduseks on aga

programmide suhteliselt väike töökiirus. BASIC-u interpretaator kutsutakse välja püsimonitorist direktiiviga **B**, LOS-ist aga direktiiviga **BASIC**. Kui BASIC-ut kasutada koos LOS-iga, võib BASIC-programme lugeda magnetlindilt ja neid sinna kirjutada; kasutamisel ainult püsimonitoriga aga see võimalus puudub. Muus suhtes BASIC-u käitumine püsimonitori või LOS-i all käivitatusena ei erine.

BASIC-u väljakutsumisel ilmub kuvarile teade

BASIC

ENSV TA KÜB.I AT EKB

ja BASIC-u viip

READY

mis näitab, et interpretaator ootab kasutaja edasisi direktiive, milleks võivad olla otserežiimi käsud või sisestatava BASIC-programmi read.

Direktiiviga RUN käivitatud programmi võib seisata klahvidega **CTRL C**, silumiseks on võimalik aga programmi kahtlastesse kohtadesse lisada direktiive **STOP**. Kui programmi täitmine on jõudnud STOP-ini või on katkestatud klahviga **CTRL C**, ilmub kuvarile teade

BREAK IN nnnn

kus nnnn on selle rea number, millel programmi täitmine seiskus.

Nüüd saab direktiiviga **PRINT** väljastada meid huvitavate muutujate väärtusi.

«Juku» graafikakäsundite kohta (vt.7.3.2), mis on realiseeritud ka BASIC-u tasemel, on oluline teada järgmist:

— kuvaril saab adresseerida horisontaalsuunas punkte 0..319 ja vertikaalsuunas punkte 0..239. Punkt (0,0) asub kuvari vasakus ülanurgas, punkt (319,239) aga paremas alanurgas

— enne teiste graafikadirektiivide kasutamist tuleb täita direktiiv **HGR**, mis viib kujuteldava graafikakursori punkti (0,0). HGR ei kustuta kuvari ekraani

— direktiiv **CLS** kustutab kuvari ekraani, kuid ei muuda graafikakursori asukohta.

4.3.2 PL/M-KOMPILAATOR

PL/M-keelse programmi transleerimine masinakoodi toimub translaatoriga **PLM.COM**, mille direktiive ja veateateid on lähemalt kirjeldatud punktis 12.

Kompilaatori väljakutsumiseks tuleb sisestada

A > PLM

misjärel kompilaator laetakse lindilt mällu. Et kompilaator on küllalt pikk, võtab ka see toiming mitu minutit. Kui PL/M käivitub, ilmub kuvarile viip:

INPUT COMMAND?

millele tuleb vastata transleeritava faili nimega ja vajadusel lisada käuslausesse ka kompilaatori võtmeid (vt. 12.2). Kui jooksev programm on transleeritud, küsib kompilaator, kas kasutaja soovib veel PL/M-programme transleerida:

MORE INPUT (Y/N)?

millele tuleb olenevalt soovist vastata Y (soovin veel üht programmi transleerida) või N (lõpetan töö kompilaatoriga).

Kompilaator väljastab **objektfaali**, mille failinime laiendiks on OBJ. Mitmest moodulist koosneva programmi transleerimisel tuleb need moodulid transleerida järjekorras. Iga mooduli transleerimisel saab ette määrata koodile ja andmetele eraldatava mälu piirkonna algusaadressid, kompilaator aga väljastab vastavad lõpuaadressid. Iga järgmine moodul tuleb transleerida eelmisest suuremal aadressil, et vältida nende kattumist mälus.

Eraldi transleeritud moodulite mällu laadimiseks on olemas programm **LOADT.COM**, mis laadib objektmooduli mällu alates transleerimisel ette antud aadressist. Kui moodul on mällu laaditud, annab **LOADT** juhtimise tagasi LOS-ile ja me võime mällu laadida järgmise mooduli. Pärast seda, kui kogu programm on mällu laaditud, tuleb ta lindile salvestada LOS-i residentse käsundi **SAVE** abil. Selleks on vaja viimase mooduli lõpuaadressi järgi välja arvutada programmi poolt kasutatavate mälu lehekülgede (2K-baidiste plokkide) arv.

PL/M-programmi silumiseks tuleb tarvitada silurit **SID** (vt. ptk. 5).

4.3.3 ASSEMBLER

Assembler on mõeldud kogenumate programmistide poolt kirjutatud assemblerkeelsete programmide transleerimiseks. LOS-i komplekti kuuluv assembler käivitub, kui kutsuda välja lindil olev programm ASM.COM. Samal käsureal tuleb ära näidata ka transleeritava programmi nimi. Assemblerkeelse programmi laiendiks on samuti ASM.

Näiteks programmi PROGRAMM.ASM transleerimisel on käsurida järgmine:

A> ASM PROGRAMM

Nagu PL/M, nii ka assembler väljastab töö tulemusena objektifaili. Sealjuures on selle objektifaili struktuur (täpsemalt on seda kirjeldatud peatükis 9) sama, mis PL/M korral. Ka assemblerkeelsete programmide mällu laadimine toimub laaduriga LOADT.COM, nagu on kirjeldatud jaotises 15.3.2.

5 PROGRAMMIDE SILUMINE

Programmide silumine moodustab ühe kõige aeganõudvama osa programmeerimisest. Et seda tööd hõlbustada, on kasutusele võetud **silurid** — programmid, mis võimaldavad jälgida silutava programmi täitmise käiku ja vajadusel seda muuta.

BASIC-programmide silumine toimub töös interpretaatoriga, STOP- ja PRINT-direktiive kasutades (vt. jaotis 4.3.1). PL/M- ja assemblerkeelsete programmide silumiseks on LOS-i kompleksis olemas silur **SID**.

5.1 SILUR SID

SID on mõeldud nende programmide silumiseks, mis on kirjutatud assembler- või kõrgkeeles ja seejärel transleeritud masinakoodi. SID-i abil saab jälgida masinakoodiprogrammi täitmist üksikute protsessori käskude kaupa ning protsessori registreid ja mälu sisu muutumist programmi töö käigus. Samuti on võimalik muuta protsessori registreid ja mälu sisu, uurida meid huvitavate programmi osade tööd jms. Silur on lindil programmifailina **SID.COM**, mis silumiseks tulebki käivitada, näidates käsulauses ära ka silutava programmifaili nime, milleks harilikult on assembleri või PL/M-i

kompilaatori poolt väljastatav objektifail (nime laiendiks on OBJ) või täidetav programmifail (nime laiendiks on COM). SID-i direktiivid, mida silumisel kasutatakse, on kirjeldatud jaotises 13.2.

SID-iga on võimalik ka sisse viia muudatusi transleeritud programmi. Selleks tuleb mälus programmi muuta direktiividega A või S, väljuda SID-ist ja salvestada programm LOS-i direktiivi SAVE abil lindile. Sealjuures tuleb ära näidata salvestatavate mälulehekülgede arv, s. t. et eelnevalt tuleb üles märkida SID-i väljakutsumisel kuvatud kasutatud mälu hulk.

5.2 VEATEATED

Programmide transleerimisel väljastab translaator veateateid, mis näitavad vigase süntaksiga programmi elementide asukohta ja samuti leitud vea tüüpi. Et veateadete arv on piiratud, võib teate sisu mitte vastata tegelikule veale. Seetõttu tuleb programmi silumisel suhtuda veateadetes loomingukselt.

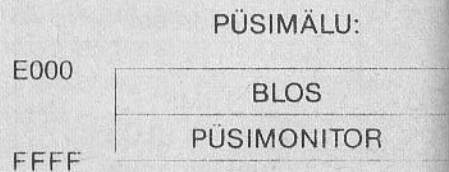
BASIC väljastab veateateid töö käigus. Seetõttu moodustavad enamiku tema kuvatavatest vigadest mitte süntaksivead, vaid programmi täitmisel tekkinud erisituatsioonid (näiteks ületäitumine jms.). Vea tekkimisel tuleks programmi parandamiseks uurida listingult direktiive, mille täitmine viis veani.

5.3 ARVUTI MÄLUJAOTUS

Programmide silumisel ja katsetamisel on kasulik tunda süsteemi mälujaotust. Mälutsoonide üldskeem on esitatud järgmisel leheküljel, üksikasjalikumaid seletusi võib leida jaotisest 8.1 ja muudest süsteemitarvara kirjeldustest.

0000	0001	0002	0005	0006	0007
MUUTMÄLU:					
JMP					
Käsitöötlusprotsessori aadress					
JMP					
Aadress BLOS-i funktsioonide poole pöördumiseks					

005C	FAILI JUHTPLOKK
006B	
006C	FAILI JUHTPLOKK
007B	
0080	OTSEMÄLLUPÖÖRDUSE PUHVER, 128 baiti
00FF	
0100	< - Programmide laadimise algusaadress LOS-i kasutamisel
BEFF	TARBIJATSOON
BFC0	OTSEMÄLLUPÖÖRDUSE PUHVER, 2K baiti
C7BF	
CA00	KÄSUPROTSSESSOR
D1FF	
D200	LINDIFAILIDE KATALOOG
D5FF	
D600	SÜSTEEMIPARAMEETRID
D7FF	
D800	VIDEOMÄLU
FD7F	
FD80	SÜSTEEMIPARAMEETRID
FFFF	



Joonis 5.1. Arvuti mälujaotus

6. TARKVARA ÜLDKIRJELDUS

Kirjeldatav programmvara põhineb kassetmagnetofonile orienteeritud lindioperatsioonisüsteemil. **Lindioperatsioonisüsteem LOS** on ülesehituselt ja kasutajakäsuandite (-direktiivide) poolest võimalikult sarnane mikroarvuti te levinud operatsioonisüsteemiga CP/M, mis baseerub ümbrikettaseadmetel. Muidugi teeb lint terve rea kitsendusi (näiteks ei tasu organiseerida juhupöördust), seetõttu saab ainult teatavat hulka CP/M-süsteemi programme muudatusteta või väikese kohandamisega realiseerida LOS-il.

Programmide väljakutse (laadimine ja käivitus) ning residentsed süsteemsed funktsioonid — kataloogi väljastus (**DIR**), teksti faili väljastus (**TYPE**), faili ümbernimetamine (**REN**), mälusisu salvestamine lindile programmina (**SAVE**) ning failide kustutamine (**ERA**) — on väliselt realiseeritud nagu CP/M-is.

Kogu arvuti programmvara võib jagada **püsiprogrammvaraks** (on püsिमälus) ja **lindil asuvaks programmvaraks**. Püsिमälus on residentne **püsimonitor**, millega saab uurida ja muuta arvuti mälusisu, protsessori tööregistreid jne., lisaks veel LOS-i alati vajalik osa, vajaduse korral ka **BASIC-u interpreteraator** või **miniassemblersilur**. Viimaste olemasolu sõltub arvuti kasutamise eesmärgist ja kassetmagnetofoni kasutamise vajadusest ning võimalusest. Kooliarvutite ühendamisel kohtvõrguks on püsिमälus ka **andmeside programmvara**.

Kasutamiseks opsüsteemiga LOS on mitu **teenindusprogrammi**. Eraldi grupi moodustab **testimise tarkvara**: keskseadme test, mälu testid, kuvari ja klaviatuuri test, linditest. Neid saab tarbija käivitada kui tavalisi programme LOS-i raames.

Programmitekstide sisestamiseks ja korrigeerimiseks ning teksti redigeerimiseks on määratud **redaktor EDIT**. Korrigeeritav või sisestatav tekstifail peab tervikuna mahtuma muutmällu.

Peale miniassembleri kuulub opsüsteemi juurde **ASSEMBLER**, mille väljundiks on objektifail (ettenähtud aadressidele laaditav programmifail) ja/või listingufail. Viimase võib väljastada ka kuvarile.

Süsteemprogrammeerimise hõlbustamiseks kasutatakse plokkstruktuuriga masinoriendatud keelt PL/M. Ühekäiguline kompilaator lubab väljundis saada lindile kas objektifaili või listingufaili ühe transleerimisseansi jooksul. Kompilaatoril on vahendid erinevate PL/M-programmide ning ka PL/M- ja assemblerkeelsete programmide linkimiseks.

Lindioperatsioonisüsteemis on realiseeritud ka BASIC-u ulatuslikum variant, millel on suuremad võimalused kui püsimalu-BASIC-ul; see erineb väikeste kitsenduste poolest standardsest «Microsofti» BASIC-ust.

Programmide silumiseks masinakoodi tasemel on silur SID, mis võimaldab ühtlasi programme lindilt laadida ja lindile salvestada. SID kergendab assembler- ja PL/M-keelsete programmide silumist.

Graafika kasutamise võimalused on muidugi olemas nii PL/M-keele (vastavate monitoridirektiivide näol) kui ka BASIC-u tasemel.

Tarkvarakomponendid on varustatud eestikeelsete kasutamisjuhenditega.

7 PÜSIMONITOR

7.1 MONITORI DIREKTIIVID

Pärast vajutamist klahvile RESET või juhtimise andmist aadressile FFC4H tuleb kuvarile tekst MONITOR < versiooni nr. > ning «*». Seejärel on kasutajal võimalik pöörduda monitori poole järgmiste direktiividega.

D XXXX/YYYY — Mälutõmmis

Väljastada ekraanile mälu sisu alates aadressist XXXX kuni aadressini YYYY ridadena kujul < aadress >: < bait 1 > < bait 2 > .. < bait 8 >

Väljastuse peatab CTRL S vajutamine ja jätkab suvalisele klahvile vajutamine. Väljastuse lõpetamiseks vajutada CTRL S ja kohe seejärel CTRL C.

M XXXX/YYYY > ZZZZ — Välja nihe

Kopeerida mälu sisu aadressidelt XXXX kuni YYYY uuele algusaadressile ZZZZ.

C XXXX/YYYY > ZZZZ — Välja kontroll

Võrrelda mälu sisu aadressidel XXXX kuni YYYY mälu sisuga alates aadressist ZZZZ. Erinevused väljastatakse kujul < aadress >: < bait x > < bait z >

Väljastuse peatab CTRL S vajutamine ja jätkab suvalisele klahvile vajutamine. Väljastuse lõpetamiseks vajutada CTRL S ja seejärel CTRL C.

K XXXX/YYYY BB — Baidi otsing

Võrrelda mälu sisu alates aadressist XXXX kuni aadressini YYYY väärtusega BB. Erinevused väljastatakse kujul < aadress >: < bait x > < BB >

Väljastuse peatab CTRL S vajutamine ja jätkab suvalisele klahvile vajutamine. Väljastuse lõpetamiseks vajutada CTRL S ja kohe seejärel CTRL C.

F XXXX/YYYY BB — Baidi laussalvestus

Täida mälu alates aadressist XXXX kuni aadressini YYYY väärtusega BB. Lubatav on ka väärtus YYYY=XXXX.

S XXXX BB-CC BB-CC .. RETURN — Salvestise muutmine

Aadressil XXXX asuva baidi väärtuse BB väljastus ja asendamine uue väärtusega CC. Dialogi sammu «BB-» sooritab süsteem. Kui CC on tühik, siis jääb olemasolev väärtus muutmata ning väljastatakse järgmise baidi väärtus. Käsu täitmise lõpetab RETURN.

Q .. — Programmi käivitus

Käivitada programm algusest, aadressist XXXX või katkestuspunktist ja täita teda lõpuni või aadressini YYYY.

Q RETURN — algusest või katkestuspunktist, lõpuni

Q XXXX RETURN — aadressist XXXX, lõpuni

Q XXXX,YYYY — aadressist XXXX, aadressini YYYY

Q YYYY — algusest või katkestuspunktist, aadressini YYYY

E CTRL C — Vallasrežiim/Sidusrežiim

E-ga minnakse kviteerivasse vallasrežiimi (OFF-LINE), kus kõikidele klahvivaljastustele vastavad koodid (kaasa arvatud ESCAPE-liitkoodid) väljastatakse ekraanile. Kood CTRL C toob monitori tagasi sidusrežiimi (ON-LINE).

B — Transleerimine

Käivitada BASIC või MINIASSEMBLER.

T — Opsüsteem

Laadida ja käivitada lindioperatsioonisüsteem.

7.2 MONITORI FUNKTSIOONID

Järgnevas kirjelduses tähistavad A,B,C,D,E,H,L arvuti vastavaid registreid, AC,CY,P,S,Z aga protsessori lippe (kahendindikaatoreid).

AC=1: dekaadülekanne (madalamast poolbaidist)

CY=1: ületäitumine

P =1: paarisarvuline tulem

S =1: negatiivne tulem

Z =1: nulltulem

«Parameeter» tähendab sisendparameetrit.

«H» arvu järel tähistab kuueteistkümnendsüsteemi.

Monitori funktsioonid on kasutatavad vaid mälu adresseerimise mooduste 1 ja 2 puhul. Erandiks on funktsioonid MEMMD ja MEMMDRST, mida võib kasutada kõikides moodustes.

MONITOR — Monitori sisendpunkt

Sisendpunkt: FFC4H

TTSTAT* — Klaviatuuri oleku lugemine

Sisendpunkt: FFC7H

Väljund: Kui mingile klahvile on vajutatud, siis Z=0 ja A=sisestatud sümbool, kui ei, siis Z=1

CRLF* — Reavahetus ekraanil

Sisendpunkt: FFCAH

TTCON* — Teksti väljastus ekraanile

Sisendpunkt: FFCDH

Parameeter: BC = teksti algusaadress; teksti lõpu tunnuseks on bait väärtusega 00H või «\$»

TTCLF* — Teksti väljastus ekraanile, reavahetusega teksti lõpus

Sisendpunkt: FFDOH

Parameeter: BC = teksti algusaadress; teksti lõpu tunnuseks on viimane bait väärtusega 00H või «\$»

TTI* — Tärgi sisestus väljastuseta ekraanile

Sisendpunkt: FFD3H

Väljund: A = sisestatud märk

TTIO* — Tärgi sisestus väljastusega ekraanile

Sisendpunkt: FFD6H

Väljund: A = sisestatud märk

TTO* — Tärgi väljastus ekraanile

Sisendpunkt: FFD9H

Parameeter: A = väljastatav märk

OUTHX* — Baidise 16-ndarvu väljastus ekraanile

Sisendpunkt: FFDCH

Parameeter: A = väljastatav 16-ndarv

OUTH2* — Kahebaidise 16-ndarvu väljastus ekraanile

Sisendpunkt: FFDFH

Parameeter: BC = väljastatav 16-ndarv

INHEX* — Baidise 16-ndarvu sisestus

Sisendpunkt: FFE2H

Väljund: Kui sisestati veatu 16-ndarv, siis CY=0 ja A = sisestatud arv, kui ei, siis CY=1

INHX2* — Kahebaidise 16-ndarvu sisestus

Sisendpunkt: FFE5H

Väljund: Kui sisestati veatu 16-ndarv, siis CY=0 ja BC = sisestatud arv, kui ei, siis CY=1

NIBBLE* — 16-ndnumbri kontroll

Sisendpunkt: FFE8H

Parameeter: A = märk(16-ndnumber)

Väljund: Kui märk on 16-ndnumber, siis CY=0, kui ei, siis CY=1

DMEM* — Mälutõmmise väljastus ekraanile.

Peatamine: CTRL S

Jätkamine: suvaline klahv

Lõpetamine: CTRL S ja kohe seejärel CTRL C

Sisendpunkt: FFF1H

Parameetrid: BC = mälupiirkonna algusaadress

HL = mälupiirkonna lõpuaadress

MOVE* — Salvestise nihe teise mälupiirkonda

Sisendpunkt: FFF4H

Parameetrid: BC = nihutatava salvestise algusaadress

HL = nihutatava salvestise lõpuaadress

DE = salvestise uus algusaadress

COMP* — Kahe salvestise võrdlus erinevuste väljastusega ekraanile

Peatamine: CTRL S

Jätkamine: suvaline klahv

Lõpetamine: CTRL S ja kohe seejärel CTRL C

Sisendpunkt: FFF7H

Parameetrid: BC = esimese salvestise algusaadress

HL = esimese salvestise lõpuaadress

DE = teise salvestise algusaadress

FILL* — Mälupiirkonna täitmine konstandiga

Sisendpunkt: FFFAH

Parameetrid: BC = mälupiirkonna algusaadress

HL = mälupiirkonna lõpuaadress

A = salvestatav väärtus

CNST* — Salvestise võrdlus konstandiga, erinevuste väljastusega ekraanile.

Peatamine: CTRL S

Jätkamine: suvaline klahv

Lõpetamine: CTRL S ja kohe seejärel CTRL C

Sisendpunkt: FFFDH

Parameetrid: BC = mälupiirkonna algusaadress

HL = mälupiirkonna lõpuaadress

A = võrreldav konstant

PRINTCHAR — Tärki väljastus printerile

Sisendpunkt: FFEEH

Parameetrid: A = väljastatav märk

Süsteemsete vahenditega on realiseeritud printeri D-100 (Poola RV) tüüpi. Teistsuguse printeri kasutamiseks tuleb aadressile, mis loetakse aadressilt FFEF, liita 1 ja salvestada sinna vastava draiveri kahebaadne aadress (sisendpunkt).

BLEEP — Helisignaali väljastus

Sisendpunkt: FF86H

Parameetrid: A = (signaali kestus, ms)/100

DE = 2000000/(signaali sagedus, Hz) või

DE = 0000H — signaali väljalülitamine

HGR — Siirdumine graafikarežiimi

Sisendpunkt: FF8CH

Parameetrid: E = 00H — väljastus vastavalt valitud värvile, või

E = 01H — väljastus pöördkujutisena valge värvuse korral (valge värviga valgele taustale väljastamisel tekib must kujutis)

HCOLOR — Värvuse valimine graafikarežiimis

Sisendpunkt: FF8FH

Parameeter: E = värvuse indeks

0: must

muu: (1..7) valge

HPLOT — Punkti väljastus ekraanile

Sisendpunkt: FF92H

Parameetrid: DE = x-koordinaadi väärtus (0..0140H)

B = y-koordinaadi väärtus (0..FOH)

HPLOTTO — Sirglõigu väljastus ekraanile

Sirglõik kulgeb viimasena valitud punktist punkti (x, y)

Sisendpunkt: FF95H

Parameetrid: DE = x-koordinaadi väärtus (0..0140H)

B = y-koordinaadi väärtus (0..FOH)

INTCNT — Programmeeritav taimer

Etteantud aja möödumist inditseerib tunnus, mis nullitakse selle lugemisel

Sisendpunkt: FF9EH

Parameeter: DE = (aeg, ms)/100 — aja etteandmine
DE = 0 — tunnuse lugemine

Väljund: A = FFH — aeg on täis

A = OOH — aeg jookseb

TIMCNT — Programmeeritav stopper

Sisendpunkt: OFFA1H

Parameeter: D = OOH — stopperi nullimine
D = FFH — stopperi lugemine

Väljund: DE = (aeg, ms)/100

INTRSV — Katkestuse teenindamine

Sisendpunkt: FF89H

Parameetrid: A = katkestuse number
DE = katkestust teenindava programmi aadress ja katkestuse lubamine
DE = OOOH — katkestuse keelamine

Kasutajale on eraldatud välised katkestused 6 ja 7 ning katkestus 8, mis toimub iga 20 ms järel. Katkestuse puhul antakse juhtimine vastavale teenindavale programmile ja süsteemsete vahenditega tagatakse registre ja pinumälu säilimine.

MEMMD — Mälu adresseerimise mooduse valik

Sisendpunkt: FFB9H

Parameeter: A = 0..3 mooduse number

Funktsiooni poole pöördumisel peab arvestama, et pinumälu ei tohi satuda mälu ümberlülitamisel püsikäsu alasse.

MEMMDRST — Siirdumine mälu adresseerimise moodusele 1

Sisendpunkt: FFBCH

Märkus: *märgitud funktsioonid säilitavad registre sisu.

7.3 KUVARI TÖÖREŽIIMID

Kuvar võib töötada graafilises või tärgirežiimis.

Ekraanile saab väljastada horisontaalsihis (x-telg) 320 ja vertikaalsihis (y-telg) 240 punkti.

7.3.1 TÄRGIREŽIIM

Tärgirežiimis jaguneb ekraan 24 reaks pikkusega 40 tärki. Tärgimaatriksi formaat on 8x10 punkti. Tärgipositsioonide koordinaadid on püstsihis 0..23 ja rõhtsihis 0..39.

Kõlmstardi korral ekraan kustutatakse ja positsiooni (0,0) viiakse ladina tähestikule vastav vilkuv kursor (vahelduvad tausta ja märgi värvus). Vene tähestikule siirdumisel muutub kursor staatiliseks (valgel taustal must).

Väljastus toimub monitori (või LOS-i) direktiividega. Kuvari juhtkoodid on kirjeldatud tabelites 7.1 ja 7.2, tärgikoodid tabelis 7.3.

Kuvari juhtkoodid

Tabel 7.1

Tähis	16-kood	Koodi väljastuse tulemus
HELL	07	Helisignaal
AS	08	Kursori tagasilüke positsiooni võrra vasakule (kuni positsioonini (0,0))
AB	09	Kursori viimine järgmisele tabuleerimisveerule (8 positsiooni kaupa), vajadusel ekraani kerimine
F	0A	Kursori nihutamine positsiooni võrra alla, vajadusel ekraani kerimine
T	0B	Sama, mis LF
F	0C	Sama, mis LF
R	0D	Kursori viimine rea algusesse
SC	1B	ESCAPE-liitkoodi töötluse alustamine
stejäänud	00..1F	Reaktsioon puudub

ESCAPE-liitkoodid

Tabel 7.2

Funktsioon	ESCAPE-jada *)	16-nd-kood
Kursori viimine positsiooni (0,0) (ekraani vasakusse ülanurka)	ESC H	1B 48
Ekraani kustutus ja kursori viimine positsiooni (0,0)	ESC L	1B 4C
Kursori viimine positsioonile (r,v) rr=rea nr. +20H, vv=veeru nr. +20H	ESC=rrvv või ESC Y rrvv	1B 3D rr vv 1B 59 rr vv
Ekraaniosa kustutus alates kursorist kuni ekraani lõpuni	ESC J	1B 4A
Realõigu kustutus alates kursorist kuni rea lõpuni	ESC K	1B 4B
Kursor tagasi positsiooni võrra	ESC D	1B 44
Kursor edasi positsiooni võrra	ESC C	1B 43
Kursori langetus positsiooni võrra	ESC B	1B 42
Kursori tõste positsiooni võrra	ESC A	1B 41
Kursori positsiooni küsimine; järgnema peab 2 klaviatuuri lugemise direktiivi; lugemise tulemusteks on vastavalt r ja v	ESC R	1B 52
Kviteeriva helisignaali keelamine	ESC 0	1B 30
Kviteeriva helisignaali lubamine	ESC 1	1B 31
Ekraani kerimise keelamine	ESC 2	1B 32
Ekraani kerimise lubamine	ESC 3	1B 33
Kursori kustutamine ning edasise väljastamise keelamine	ESC 4	1B 34
Kursori väljastamise lubamine	ESC 5	1B 35
Ekraani sujuva kerimise režiim	ESC :	1B 3A
Ekraani hüppelise kerimise režiim	ESC ;	1B 3B
Põrdkujutise algus	ESC ')	1B 27
Põrdkujutise lõpp	ESC (1B 28

*) Lubatud on ka nurksulg ESC koodi järel, näiteks ESC [H.

7.3.2. GRAAFIKAREŽIIM

Graafikarežiimis saab kujutise koostada suvalistest ekraanipunktidest. Punktide koordinaadid on järgmised:

— x-telg : 0 .. 319,
— y-telg : 0 .. 239.

Punkt koordinaatidega (0,0) paikneb ekraani vasakus ülanurgas.

Graafikarežiimi saab siirduda monitori (või LOS-i) funktsiooniga HGR. Võimalik on graafikakäskude ja tärgiväljastuse vaheldumine. Iga graafikakäsu täitmisel kustutatakse ekraanilt tärgireiimi kursor. Tärkide väljastamisel kursor taastub.

Graafikakursori all mõistetakse viimasena valitud punkti, sirge puhul sama otspunkti. Graafikakursorit ekraanil ei inditseerita.

Käsund(direktiiv) CLS kustutab kuvari ekraani, kuid ei muuda graafikakursori asukohta. Funktsiooni HGR täitmisel viiakse valge graafikakursor punkti (0,0). Seejärel on võimalik täita järgmisi graafikafunktsioone:

COLOUR n — värvuse valimine, indeks n kolme madalaima biti järgi (0:must, 1..7:valge)

PLOT x,y — graafikakursori viimine punkti koordinaatidega (x,y), millele on valitud värvus

PLOT TO x,y — sirglõigu joonestamine viimasena määratud punktist (kus paikneb graafikakursor) punkti koordinaatidega (x,y); lõigu punktid esitatakse valitud värvusega, lõigu alguspunkti indikatsiooni ei uuendata

7.4. KLAHVIKOODID

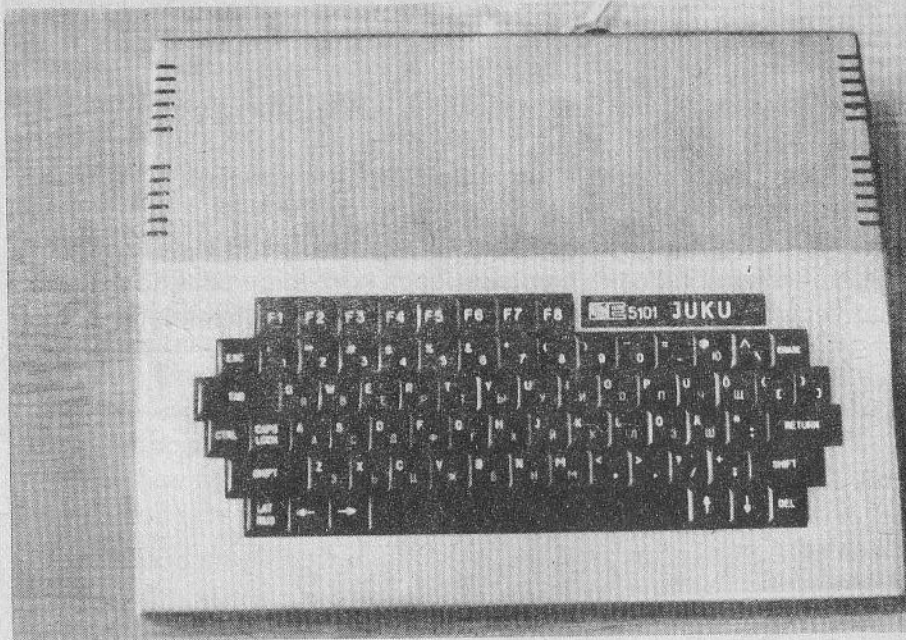
Klaviatuur koosneb kolme tüüpi klahvidest:

- tärgiklahvid,
- juhtklahvid,
- režiimiklahvid.

Tärgi- ja juhtklahvide väljastatavad koodid on esitatud tabelis 7.3.

Režiimiklahvidel on järgmised funktsioonid:

- CTRL** — vajutamisel koos tärgiklahviga väljastatakse talitlu kood (tärgikoodi viis nooremat bitti)
- SHIFT** — vajutamisel koos tärgiklahviga valitakse suurtähed kahe märgiga klahvi puhul ülemine märk
- CAPS LOCK** — suur- ja väiketähtede vahetus; valitud tähed on kasutusel järgmise vajutuseni
- RUS/LAT** — ladina ja vene tähestiku vahetus; valitud tähestik on kasutusel järgmise vajutuseni
- CTRL ESC** — siire aadressile 0000, sealt antakse juhtimine edas monitorile või lindioperatsioonisüsteemile, või püsimalu-BASIC-ule
- CTRL SHIFT ESC** — juhtimine antakse monitorile, kui taimerikatkestus töötab
- SHIFT ERASE** — ekraani kustutus (kui ei toimu andmevahetust magnetofoniga)



«Juku» klaviatuur

Tabel 7.3

0	0	F5	F1	F7	F4	F8	ESC	0	Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О	тühik*)
1	1							1	Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О	
2	2							2	Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О	
3	3							3	Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О	
4	4							4	Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О	
5	5							5	Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О	
6	6							6	Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О	
7	7							7	Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О	
8	8							8	Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О	
9	9							9	Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О	
A	A							A	Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О	
B	B							B	Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О	
C	C							C	Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О	
D	D							D	Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О	
E	E							E	Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О	
F	F							F	Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О	

Sulgudesse paigutatud märk või talitlustähis vastab standardsele KOI-8 kooditabelile (GOST 19768-74) ning on kujutatud ka standardklahvidel, ekraanile toovad aga nende klahvide koodid eesti tähe.

*) Koodi FF väljastab klahvikombinatsioon RUS DEL, ekraanile toob kood tühiku.

8 OPERATSIOONISÜSTEEM

8.1 ÜLDANDMEID

Lindioperatsioonisüsteem hõlbustab informatsiooni säilitamist lindil ja pakub arvuti kasutajale muidki teenuseid.

Lindioperatsioonisüsteemi alglaadimiseks tuleb

- asetada magnetofonile süsteemne lint operatsioonisüsteemiga
- sisestada monitori direktiiv «T»

Alglaadimise käigus loetakse käsutöötlusprotsessor lindilt muutmällu. Eduka laadimise lõppedes ilmub ekraanile tekst

*** ENSV TA KÜB.I AT EKB ***
* LINDIOPERATSIOONISÜSTEEM *

ja süsteemi valmisoleku tähis (viip) «A>». Edasiseks tööks tuleb lindi kataloog avada käsuga OPEN. Kui kataloogi avamine toimub edukalt, väljastatakse teade.

Pärast operatsioonisüsteemi alglaadimist ja kataloogi avamist on süsteemi mälujaotus järgmine:

0000H - 00FFH	Süsteemi parameetrite tsoon (ST)
0100H - BFFFH	Tarbijaprogrammi tsoon (TT)
BFC0H - C7BFH	2K baidine otsemällupöörduse (OMP) puhver
CA00H - D1FFH	Käsuprotsessor (KP)
D200H - D5FFH	Lindifailide kataloog (LK)

Need operatsioonisüsteemi osad asuvad muutmällus. Püsिमällus asub Baas-LindiOperatsioonisüsteem (BLOS).

Süsteemi parameetrite tsooni jaotus:

00H	siirdekäsu JMP 1.bait
01H - 02H	käsuprotsessori sisendpunkti aadress
05H	siirdekäsu JMP 1.bait
06H - 07H	BLOS-i funktsioonide poole pöördumise aadress
5CH - 6BH	faili juhtplokk(FJP), järgmiste andmetega:

0 1 2 8 9 10 11 12 15

0 01 02 ... 08 09 10 11 12 13 14 ... 15

01-08 failinimi

09-15 failinime laiend

00H - 7BH

teine faili juhtplokk

00H - FFH

128-baidine otsemällupöörduse puhver

Tarbijatsoonis asuvad kasutajaprogrammid, mis on lindilt laaditud mälu. Näiteks teksti redigeerimise ajal sisaldab TT tekstiredaktori ja redigeeritava teksti.

2K baidine otsemällupöörduse puhver on mäluväli, mille kaudu toimuvad kõik failioperatsioonid, kui ei ole spetsiaalselt näidatud muud.

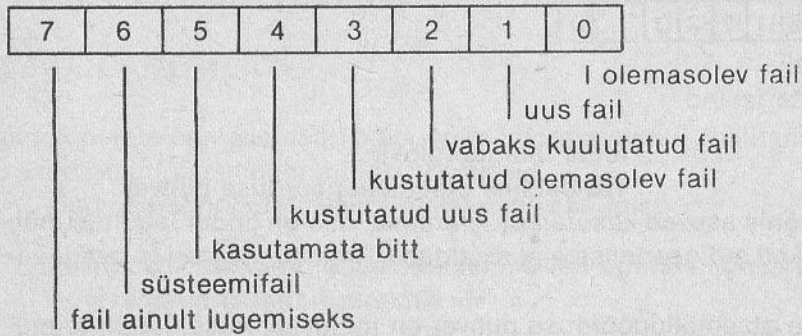
Lindifailide kataloog sisaldab lindi nime, andmed kuni 63 lindifaili kohta, samuti andmed vigaste, vabade ning lugemisel vigu andnud plokkide kohta. Kataloogi pikkus on 1K bait. Kataloogi kirjed on järgmised:

Lindi nimi		11 baiti
Kataloogi ploki number		1 "
Fail 1		14 "
Fail 63		14 "
Otsingumärgised iga 16 ploki järel		34 "
Vabade plokkide massiiv		32 "
Lugemisel vigu andnud plokkide massiiv		32 "
Vigaste plokkide massiiv		32 "

Andmed faili kohta kataloogis:

Faili staatus		1 "
Faili nimi		11 "
Faili algus: ploki number		1 "
Faili lõpp: ploki number		1 "

Faili staatus kataloogis:



Käsuprotsessor (KP) vahetab infot kasutaja ja operatsioonisüsteemi vahel. KP loeb ja töötleb klaviatuurilt sisestatud käsuriidu. KP valmisoleku käsu sisestuseks näitab teade «A>». KP sisaldab ka rea sisefunktsioone.

DIR	— mittesüsteemsete failide kataloogi esitus
DIRS	— süsteemsete failide kataloogi esitus
REN	— failide ümbernimetamine
ERA	— failide kustutamine
REST	— kustutatud failide taastamine
MEM	— üldinfo lindi kohta
TYPE	— tekstifaili väljastus ekraanile
DUMP	— faili sisu väljastus 16-ndkoodis
SAVE	— mälu sisu salvestamine faili
OPEN	— lindi avamine
CLOSE	— lindi sulgemine
MONID	— väljumine monitori
BASIC	— püsimalus oleva BASIC-u käivitamine
LOAD	— faili laadimine lindilt muutmällu
RUN	— laaditud programmi käivitamine

Lindifaile tähistatakse järqmisel:

FAILINIMI.LAIEND

Failinimi sisaldab kuni 8 ja laiend 3 tärki ning neid eraldab üksteisest punkt. Laiend võib ka puududa. Failinimes ja laiendis ei tohi esineda jär-

reid märgid: koma(,), semikoolon(;), koolon(:), küsimärk(?), tärn(*), nool (< või >). Mõningad kasutatavamad laiendid:

ASM MAC	— assemblerkeele lähtefail
PRN LST	— listingufail
TXT	— tekstifail
HEX	— masinakood 16-ndkujul
!!!	— ajutine fail
COM	— laadefail

Sisefunktsioonide ERA, REST, DIR, DIRS kasutamisel võib failinime ja laiendi sisestada kas üheselt või mitmeselt määratuna. Mitmeselt määratuleks kasutatakse tähiseid «*» ja «?» :

- asendab failinimes või laiendis ühte märki, tähenduses «mis tahes märk sellel kohal»
 - asendab failinime või laiendit, tähenduses «mis tahes nimi (laiend)»; tärn nime (laiendi) algusosa järel asendab järgnevat lõpuosa, tähenduses «mis tahes lõpuosaga nimi(laiend)»
- Vormingud *.* ja ????????????? on sarnased. Järgnevates peatükkides täpsustatakse mõiste «failinimi» all failinimest ja laiendist koosnevat lindifaili identifikaatorit.

Näide:

- ERA A??.*** — kustutamisele kuuluvad kõik failid, mille nimi on 3 märki pikk ja algab sõltumata laiendist tähega «A»
- ERA A*.COM** — kustutamisele kuuluvad kõik failid, mille nimi algab tähega «A» ja mille laiendiks on «COM»

Vastuseks süsteemi valmidusteatele (viibale) sisestatakse käsuriida (3 võimalikku kuju):

- programminimi
- programminimi parameeter1
- programminimi parameeter1 parameeter2

Programminimi on sisefunktsiooni nimi või kasutaja programmi nimi. Kui käsureas on sisefunktsioon, siis see täidetakse. Vastasel korral otsitakse kataloogist laadefaili

programminimi.COM

Sellise nimega faili leidmisel laaditakse see alates TT algusest (aadressist 100H) mällu ja käivitatakse. Olematu laadefaili puhul väljastatakse ekraanil järgmisele reale märk «?» ning programminimi.

Programminime järel saab sisestada ühe või kaks parameetrit (tavali- selt on parameetriks failinimi). KP moodustab nendest parameetritest ST-sse ühe või kaks faili juhtplokki; parameetrite puudumisel täidetakse FJP-d tühikutega. Käsurea maksimaalne pikkus on 128 märki. Pärast sisestatud käsurea analüüsi salvestatakse 128-baidisesse OMP puhvrissse programminimele järgnevast märgist algav käsurea osa. KP puhvri esimeses baidis (aadressil 80H) on sisestatud sümbolite arv.

Käsurea sisestamisel saab kasutada järgmisi juhtkoode (klahv CTRL ja täht):

CTRL J	— (=reavahetus) lõpetab sisestuse
CTRL M	— (=tagastus) lõpetab sisestuse
CTRL X	— rea kustutus ja kursor rea algusesse
CTRL H	— kursori tagasilüke, märgi kustutusega
<RETURN>	— tagastusklahv lõpetab sisestuse

8.2 KÄSUPROTSessori RESIDENTSED FUNKTSIOONID

Funktsioonide kirjeldamisel tähendab «fnimi» täielikku lindifaili nime. Nurksulgudes olev parameeter võib jääda ka sisestamata, s.t. saab kasutada vaikimisi-väärtust.

DIR	[fnimi]	Failide nimekirja väljastus. Vaikimisi väljastatakse kogu kataloogi mittesüsteemsete failide loetelu
DIRS	[[fnimi]]	Nagu DIR, kuid väljastatakse ainult süsteemi failid
REN	fnimi1 fnimi2	Faili ümbernimetamine, fnimi1 on uus ja fnimi2 on vana failinimi
ERA	fnimi	Failide kustutamine kataloogist. Kui nimeks on sisestatud *.* , küsib süsteem: ALL FILES(Y/N)? (Kas kõik failid?) Jaatava vastuse (Y) korral kustutatakse kataloogist kõik staatusega READ/WRITE failide nimed, eitava vastuse korral kustutusi ei toimu.

REST	fnimi	Käsuga ERA kustutatud failinimede taastamine. Kui etteantud nimega fail on olemas, jääb taastamine ära. Kui kustutatud faili kuulunud plokkidele on kirjutatud teine fail, siis failinime ei taastata.
MEM		Väljastatakse üldinformatsioon lindi mäluhõive kohta: nnK FILES xxxK MEMORY USED yyyK MAXIMUM FILE SPACE nnK - failide arv lindil xxxK - kasutatud mälu baitides yyyK - maksimaalne järjestikune vaba ruum lindil baitides. See näitab, kui suure faili võime antud lindile kirjutada.
TYPE	fnimi	Faili teksti väljastus ekraanile. Väljastatav fail peab olema tekstifail.
DUMP	fnimi [aaaa]	Faili sisu väljastus ekraanile 16-süsteemis, 8 baiti reas, iga rea ees aadress. Algusaadressi võib ise sisestada 16-arvuna, vaikimisi on algusaadress 0000H.
LOAD	fnimi [aaaa]	Lindifaili laadimine muutmällu. Vaikimisi laaditakse fail alates aadressist 100H. Algusaadress sisestatakse 16-ndarvuna.
RUN	[paam1] [paam2]	Mällu laaditud programmi käivitus. Selle käsu toimel antakse juhtimine üle aadressile 100H. Parameetrid salvestatakse FJP-sse.
SAVE	nn fnimi [aaaa]	nn plokki mälusisu alates aadressist aaaa salvestatakse lindifaili. Kui etteantud nimega fail on juba olemas, kustutatakse see eelnevalt. Algusaadressi võib jätta ka sisestamata, vaikimisi on algusaadressiks 0100H. Aadressid tuleb sisestada 16-ndarvudena.
CLOSE		Kataloogi kirjutus mälust lindile ja lindi sulgumine.

OPEN	Kataloogi lugemine lindilt mällu ja lindi avamine.
MONID	Väljumine operatsioonisüsteemist monitori. Lindisüsteem lähestatakse, lint keritakse algusse.

8.3 TEENINDUSPROGRAMMID

FORMAT [Inimi]	Lindi vormindamine. Lindile luuakse ploki-struktuur. Lint kirjutatakse täis 2K baidiseid plokkide, seejärel plokkid loetakse. Kui mõne ploki lugemine ei õnnestu, kuulutatakse see plokk vigaseks. Lõpuks kirjutatakse lindile uus kataloog. Parameetrikse võib olla lindi nimi, mis on samasuguse struktuuriga kui failinimi
-----------------------	---

DIALOG: (8-märgine nimi ja 3-märgine laiend).

(1) FORMAT(Y/N)? Süsteem küsib: kas vormindada? Jaatavale vastusele «Y» järgneb samm (2). Eitavale vastusele järgneb samm (3).

(2) PLACE NEW TAPE, THEN PRESS <RETURN> Asetada magnetofonile lint, mida soovitakse vormindada, seejärel vajutada klahvile RETURN. Peale vormindamise lõppu teatab süsteem kirjutatud plokkide ja veaga plokkide arvu ning siirdub tagasi sammule (1)

(3) PLACE OLD TAPE, THEN PRESS <RETURN> Asetada magnetofonile esialgne lint tagasi ja vajutada klahvile RETURN. Sellega väljutakse programmist FORMAT.

SYSGEN See programm võimaldab kopeerida süsteemi ühelt lindilt teisele. Parameetreid ei ole.

Dialog:

PLACE NEW TAPE, THEN PRESS <RETURN> Asetada magnetofonile lint, millele soovime süsteemi kirjutada, ja vajutada RETURN.
FUNCTION COMPLETED Süsteem on kopeeritud uuele lindile.
PLACE OLD TAPE, THEN PRESS <RETURN> Asetada magnetofonile esi-

algne lint tagasi ja vajutada RETURN. Seejärel toimub programmist väljumine.

STAT Väljastab informatsiooni kataloogis olevate failide kohta ja võimaldab muuta failide staatusatribuute. Faili atribuudid on:

R/O (READ/ONLY) — fail ainult lugemiseks
R/W (READ/WRITE) — fail lugemiseks ja kirjutamiseks
SYS (SYSTEM) — süsteemifail

Käsu vormingud:

— STAT failinimi (1)
 — STAT failinimi atribuut (2)

Käsu (1) täimisel väljastatakse etteantud failide kohta järgmised andmed: failinimi, pikkus K baitides ja faili atribuut. Käsu (2) täitmisel kinnistatakse etteantud failile uus atribuut.

8.4 OPERATSIOONISÜSTEEMI LIIDESTUS

8.4.1 BLOS-I FUNKTSIOONIDE KASUTAMINE

Pöördumine funktsioonide poole toimub viienda mälupea kaudu. Süsteemitooni viiendas baidis on siirdekäsu JMP kood C3H, järgmises kahes baidis aga BLOS-i sisendpunkti aadress. Süsteemsete funktsioonide kasutamiseks tuleb arvestada järgmisi liidestusreegleid:

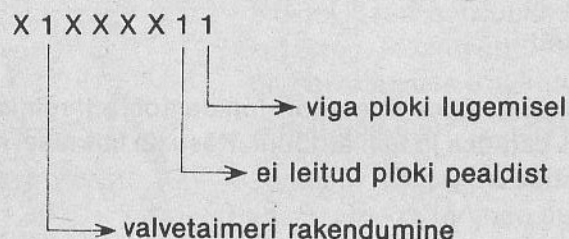
1. Funktsioonid on nummerdatud ja pöördumisel tuleb eelnevalt C-registrisse salvestada funktsiooni number.
2. Sisendinfo aadress antakse üle registripaaris DE. Mõne funktsiooni korral on kogu sisendinfo registripaaris DE ja registris A. Graafikafunktsioonide kasutamisel antakse sisendinfo registripaaris DE ja registris B.
3. Ühebaidine sisendinfo antakse üle registris E.
4. Väljundinfo antakse tagasi registris A. Kui A=0, täideti funktsioon edukalt, A=FFH korral oli funktsiooni täitmisel viga. Registris C väljastatakse veakood.

5. Failifunktsioonide kasutamisel peab arvestama, et faili tohib töödelda ainult funktsioonidega, millel on ühesugune ploki pikkus. Näiteks kui fail avati funktsiooniga 15, võib tööks failiga kasutada ainult funktsioone 16, 20 ja 21. Kui avamine toimus funktsiooniga 41, võib kasutada funktsioone 41,42 ja 43.

8.4.2 BLOS-i FUNKTSIOONIDE LOETELU

FUNKTSIOON 0: SYSTEM RESET — Operatsioonisüsteemi lähtestus

Sisendparameetreid ei ole
Väljund: C = funktsiooni olek



Operatsioonisüsteem loetakse lindilt mällu ja käivitatakse. Lindisüsteem on nn. algseisus.

FUNKTSIOON 1: CONSOLE INPUT — Klaviatuurilt lugemine

Sisendparameetreid ei ole
Väljund: A = märgi kood (KOI-8)
Klaviatuurilt loetakse märgi kood registrisse A.

FUNKTSIOON 2: CONSOLE OUTPUT — Väljastus ekraanile

Sisendparameetrid E = märk koodis KOI-8
Väljundandmeid ei ole
Märgi kood (KOI-8) väljastatakse registrist E ekraanile. Arvestatakse kontrollkoodi CTRL S (väljastuse katkemine) ja CTRL P (väljastuse jätkamine koos prindiga).

FUNKTSIOON 5: LIST OUTPUT — Väljastus printerile

Sisendparameetrid: E = märgi kood (KOI-8)
Väljund puudub
Registrist E väljastatakse märgi kood printerile.

FUNKTSIOON 6: DIRECT CONSOLE I/O — Otsene andmevahetus kuvariga

Sisendparameetrid: sisestusel E = 0FFH
väljastusel E = märgikood
Väljund: sisestusel A = 0 (klahvile ei vajutatud)
või A = sisestatud märgi kood
See funktsioon ei arvesta juhtkoode (CTRL S ja CTRL P)

FUNKTSIOON 9: PRINT STRING — Stringi väljastus

Sisendparameetrid: DE = stringi aadress
Väljund: ei ole
Märgistring saadetakse kuvarile. Stringi lõputunnuseks on «\$».

FUNKTSIOON 10: READ CONSOLE BUFFER — Lugemine klaviatuurilt puhvrisse

Sisendparameeter: DE = puhvri algusaadress
Väljund: märkide koodid puhvris

0	1	2	3		n
mx	nc	c1	c2		

mx — maksimaalne koodide arv, mida on võimalik sisse lugeda (1..255)
nc — sisseloetud koodide arv

Puhvrisse lugemisel võib kasutada järgmisi juhtkoode:

CTRL H — kursori tagasilüke
CTRL J — reavahetus lõpetab sisestuse
CTRL M — kursori tagastus lõpetab sisestuse
CTRL X — kursori tagastus rea algusesse

FUNKTSIOON 11: GET CONSOLE STATUS — Klaviatuuri oleku lugemine

Sisendparameetreid ei ole

Väljund: A = klaviatuuri oleku kood

See funktsioon kontrollib, kas märk on sisestatud. Kui on, siis A=0FFH, muidu A=00H

FUNKTSIOON 13: RESET TAPE SYSTEM — Lindisüsteemi lähestus

Sisendparameetreid ei ole

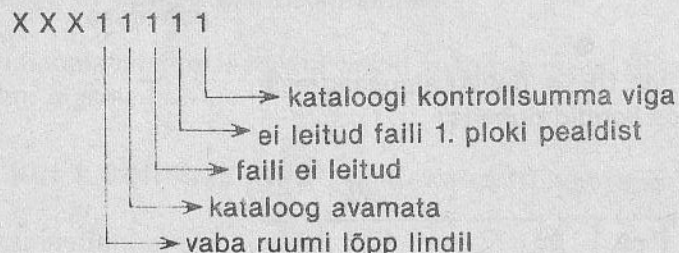
Väljund puudub

Lindisüsteem viiakse algseisu. Selleks suletakse nii kataloog kui avatud fail ning otsemällupöörduse puhvrite viidad saavad vaikimisväärtused.

FUNKTSIOON 15: OPEN FILE — Faili avamine tööks 128- baidiste plokkidega

Sisendparameetrid: DE=FJP aadress

Väljund: C = funktsiooni olek:

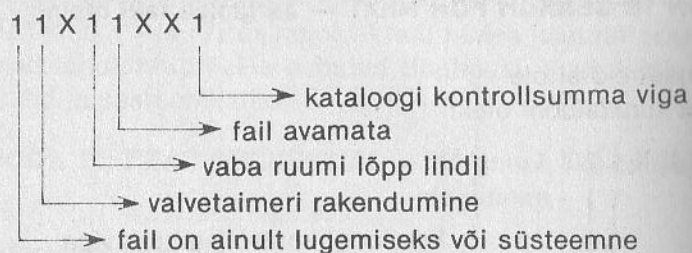


Valmistatakse ette töö 128-baidiste blokkidega, käsutab sisemiselt faili avamist funktsiooniga 40 (Open file).

FUNKTSIOON 16: CLOSE FILE — 128-baidiste plokkidega faili sulgemine

Sisendparameetreid ei ole

Väljund: C = funktsiooni olek:



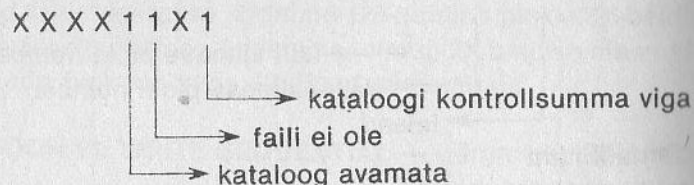
Funktsiooni täitmisel avatud fail suletakse funktsiooniga 41 (Close file). Kui 2K baidine otsemällupöörduse puhver ei olnud veel lõpuni täidetud 128-baidiste plokkidega ja lindile ära kirjutatud, siis tehakse seda nüüd enne faili sulgemist funktsiooniga 43(Write segmental).

FUNKTSIOON 17: SEARCH FOR FIRST — Esimese faili otsing

Sisendparameetrid: DE = FJP aadress

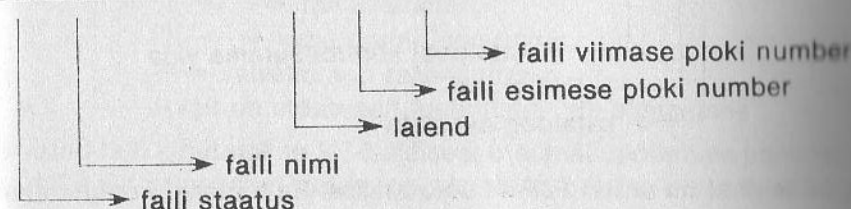
A = mask otsinguks faili staatuse järgi

Väljund: C = funktsiooni olek:



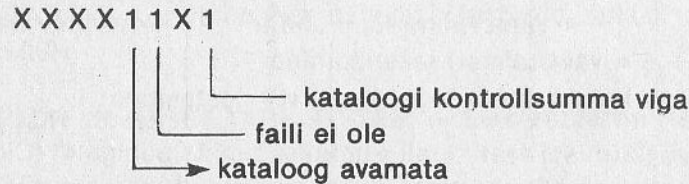
Kataloogist otsitakse faili, mille nimi on antud FKP-s. Faili leidmisel kantakse 128-baidisesse otsemällupöörduse puhvrisse faili kataloogikirje:

0	1		9	12	13	14
S	NNNNNNNN	LLL	A	B	X...	

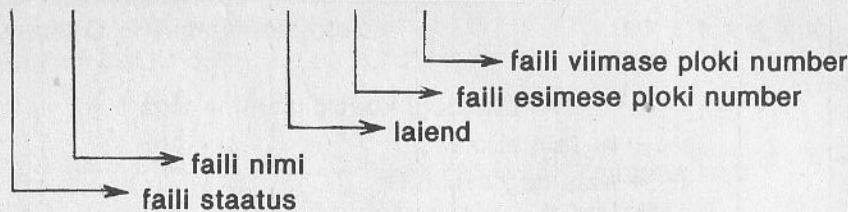
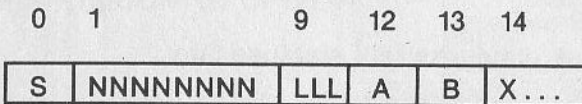


FUNKTSIOON 18: SEARCH FOR NEXT — Järgmise faili otsing

Sisendparameetreid ei ole
 Väljund: C = funktsiooni olek:

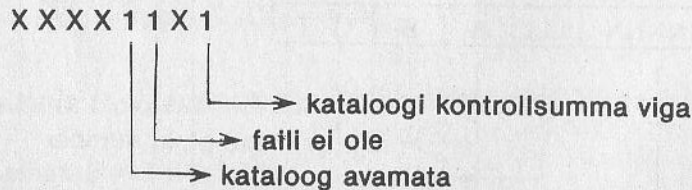


Kataloogist otsitakse järgmist faili, mille nimi rahuldab viimase funktsiooniga 17 (Search for first) FJP-s antud tingimusi. Faili nimes on lubatud kasutada märke «*» ja «?». Faili leidmisel kantakse 128-baidisesse otsemällupöörduse puhvrise faili kataloogikirje:



FUNKTSIOON 19: DELETE FILE — Faili kustutus

Sisendparameetrid: DE = FJP aadress
 Väljund: C = funktsiooni olek:

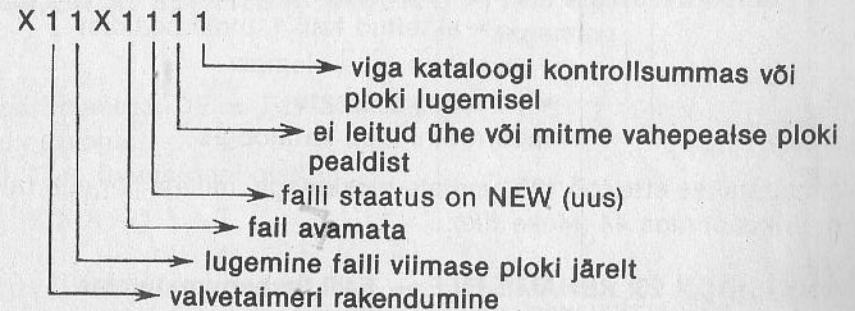


Fail, mille nimi on antud FJP-s kustutatakse. Faili plokid kuulutatakse

vabaks. Märkide «*» ja «?» esinemisel faili nimtes kustutatakse kõik failid, mille nimed rahuldavad FJP-s esitatud tingimusi. Kustutamisele ei kuulu lugemisfailid ja süsteemifailid.

FUNKTSIOON 20: READ SEQUENTIAL — Järgmise 128-baidise ploki lugemine

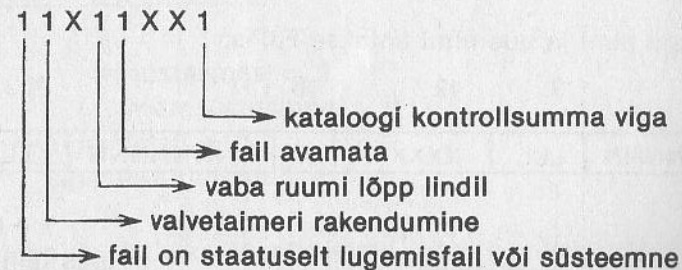
Sisendparameetreid ei ole
 Väljund: C = funktsiooni olek:



Avatud failist loetakse järgmine 128-baidine plokk 128-baidisesse otsemällupöörduse puhvrise. Lugemine toimub 2K baidise otsemällupöörduse puhvri kaudu funktsiooniga 42 (Read sequential).

FUNKTSIOON 21: WRITE SEQUENTIAL — Järgmise 128-baidise bloki kirjutus

Sisendparameetreid ei ole
 Väljund: C = funktsiooni olek:

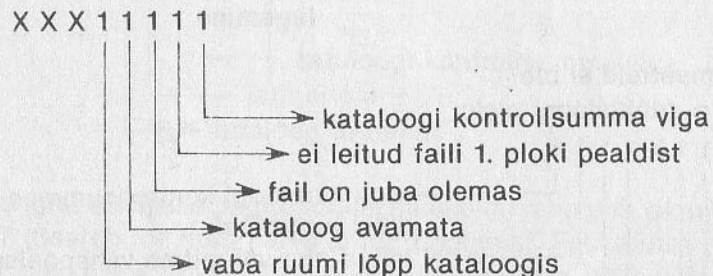


Avatud faili kirjutatakse 128-baidisest otsemällupöörduse puhvrilt järgmine 128-baidine plokk. Kirjutamine toimub 2K baidise otsemällupöörduse puhvri kaudu funktsiooniga 43 (Write sequential).

FUNKTSIOON 22: MAKE FILE — 128-baidiste plokkidega faili loomine

Sisendparameetrid: DE = FJP address

Väljund: C = funktsiooni olek:

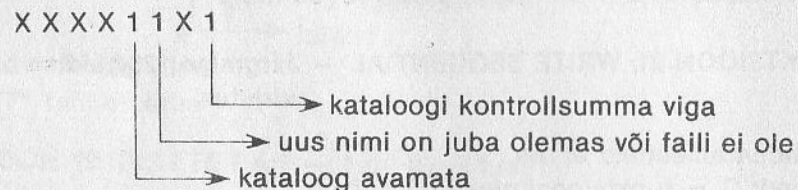


Valmistatakse ette töö 128-baidiste blokkidega, millele järgneb faili loomine funktsiooniga 44 (Make file).

FUNKTSIOON 23: RENAME FILE — Faili ümbernimetamine

Sisendparameetrid: DE = FJP address

Väljund: C = funktsiooni olek:



Otsitava faili nimi ja uus nimi antakse FJP-s:

0 1 9 12 16 17 25 28

X	NNNNNNNN	LLL	XXXX	X	NNNNNNNN	LLL	X...
---	----------	-----	------	---	----------	-----	------



Kui uue nimega fail on kataloogis juba olemas, jääb faili nimi muutmata.

FUNKTSIOON 26: SET DMA ADDRESS — Otsemällupöörduse 128-baidise puhvri viida väärtustamine

Sisendparameetrid: DE = mälu aadress

Väljund puudub

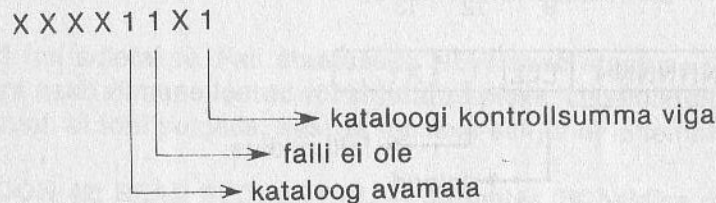
Paigaldatakse 128-baidine otsemällupöörduspuhver selle viida väärtustamisega.

FUNKTSIOON 30: SET FILE ATTRIBUTES — Faili staatusatribuudi tegemine

Sisendparameetrid: DE = FJP address

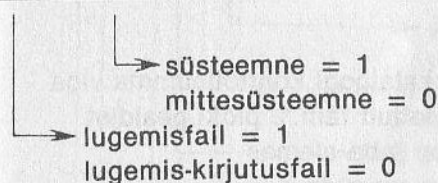
A = faili staatus

Väljund: C = funktsiooni olek:



Väärtustatakse faili staatuse atribuudid: lugemisfail või lugemis-kirjutusfail ning süsteemne või mittesüsteemne. Vajalik atribuutide komplekt esitatakse A-registris:

0	1	2
R	S	X...

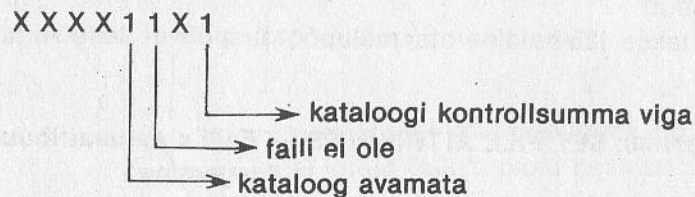


Märkide «*» ja «?» kasutamisel faili nimes kuuluvad väärtustamisele kõigi nende failide staatuse atribuudid, mille nimed rahuldavad FJP-s esitatud tingimusi.

FUNKTSIOON 35: COMPUTE FILE SIZE — Faili pikkuse arvutus

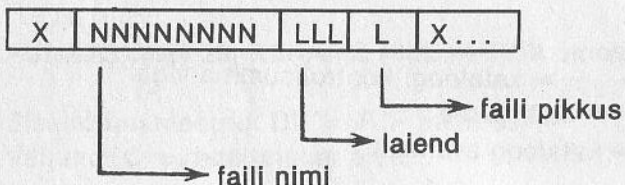
Sisendparameetrid: DE = FJP aadress

Väljund: C = funktsiooni olek:



Faili pikkus arvutatakse 2K baidistes ühikutes. Saadud tulemus pannakse FJP-sse:

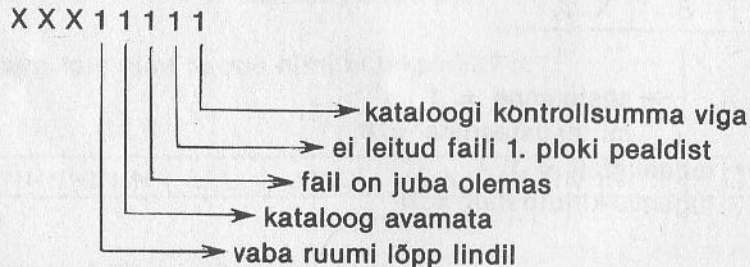
0 1 9 12 13



FUNKTSIOON 40: OPEN FILE — 2K baidiste plokkidega faili avamine

Sisendparameetrid: DE = FJP aadress

Väljund: C = funktsiooni olek:



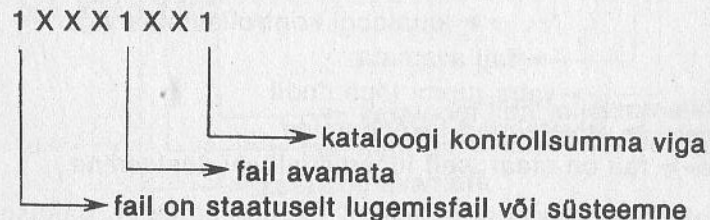
Fail avatakse. Faili jaoks staatusega NEW otsitakse kataloogist maksimaalne vaba ruum lindil ja paigutatakse ta sinna. Faili jaoks staatusega OLD leitakse vaba ruum failile järgnevas lindi osas tema võimalikuks laiendamiseks.

damiseks. Järgneb faili asukoha otsimine lindil. Et korruga saab olla avatud ainult üks fail, siis uue faili avamine teeb eelmise faili kättesaamatuks.

FUNKTSIOON 41: CLOSE FILE — 2K baidiste plokkidega faili sulgemine

Sisendparameetreid ei ole

Väljund: C = funktsiooni olek:

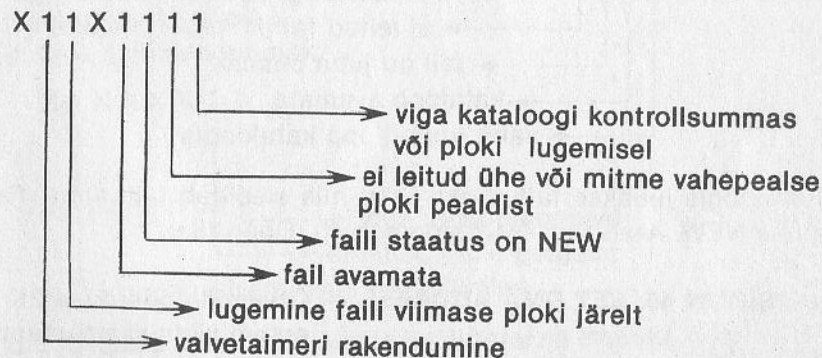


Avatud fail suletakse. Fail staatusega NEW saab staatuse OLD. Faili lõpuplokkiks saab viimane loetud või kirjutatud plokk. Lugemisfaili, mida ei loetud lõpuni, ei tohi sulgeda, sest ta tehakse sel juhul lõhemaks.

FUNKTSIOON 42: READ SEQUENTIAL — Järgmise 2K baidise ploki lugemine

Sisendparameetreid ei ole

Väljund: C = funktsiooni olek:

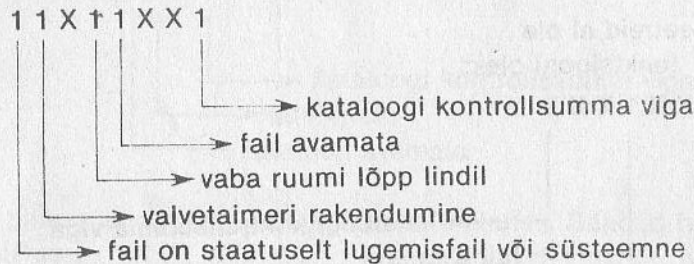


Avatud failist loetakse järgmine 2K baidine plokk 2K baidisesse otsemällupöörduse puhvrise.

FUNKTSIOON 43: WRITE SEQUENTIAL — Järgmise 2K baidise ploki kirjutamine

Sisendparameetreid ei ole

Väljund: C = funktsiooni olek:

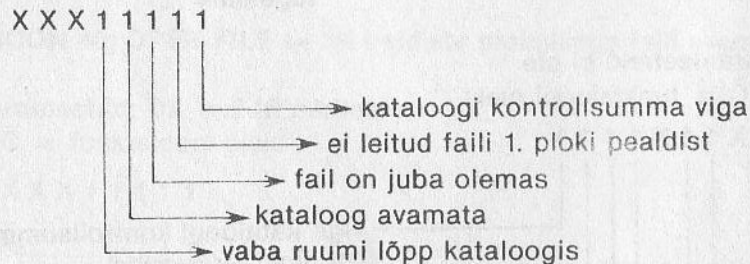


Avatud faili kirjutatakse järgmine 2K baidine plokk 2K baidisest otsemällupöörduse puhvrist.

FUNKTSIOON 44: MAKE FILE — 2K baidiste plokkidega faili loomine

Sisendparameetrid: DE = FJP aadress

Väljund: C = funktsiooni olek:



Kataloogis luuakse faili jaoks kirje, mis sisaldab faili nime. Fail saab staatuse NEW. Avab faili funktsiooniga 40 (Open file).

FUNKTSIOON 45: SET DMA ADDRESS — Otsemällupöörduse 2K baidise puhvri viida väärtustamine

Sisendparameetrid: DE = mälu aadress

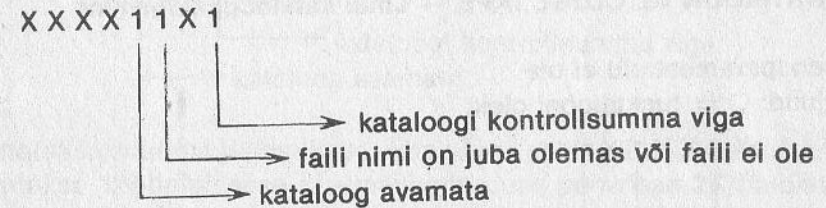
Väljundandmeid ei ole.

Paigaldatakse 2K baidine otsemällupöörduse puhver tema viida väärtustamisega.

FUNKTSIOON 46: RESTORE FILE — Kustutatud faili taastamine

Sisendparameetrid: DE = FJP aadress

Väljund: C = funktsiooni olek:

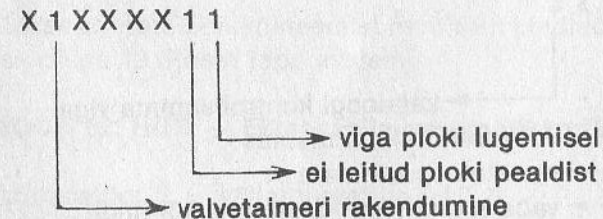


Taastatakse kustutatud fail, mille nimi on antud FJP-s. Kui sellise nimega fail on juba olemas, siis kustutatud faili otsimist ei toimu. Kui kustutatud faili kuulunud plokkid on viidud mõnda teise faili, siis kustutatud faili ei taastata. Kui kustutatud faili kuulunud plokkid on endiselt vabad, siis kuulutatakse nad uuesti hõivatuks ja fail saab staatuse OLD. Märkide «*» ja «?» kasutamisel faili nimes kuuluvad taastamisele kõik failid, millede nimed rahuldavad FJP-s esitatud tingimusi.

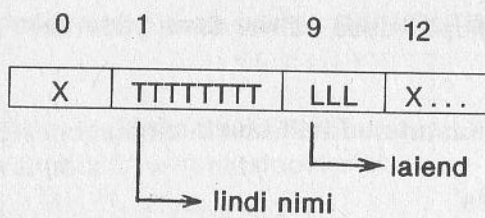
FUNKTSIOON 47: OPEN TAPE — Lindi kataloogi avamine

Sisendparameetreid ei ole

Väljund: C = funktsiooni olek:



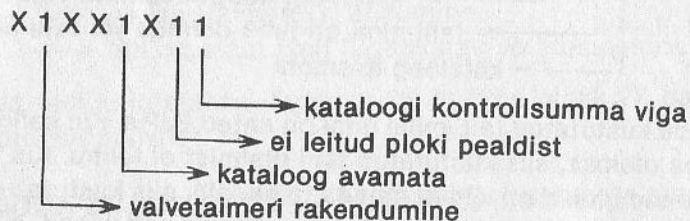
Lindilt loetakse kataloog mällu ja kuulutatakse avatuks. Lindi nimi kantakse 128-baidisesse otsemällupöörduse puhvrissi:



FUNKTSIOON 48: CLOSE TAPE — Lindi kataloogi sulgemine

Sisendparameetreid ei ole

Väljund: C = funktsiooni olek:



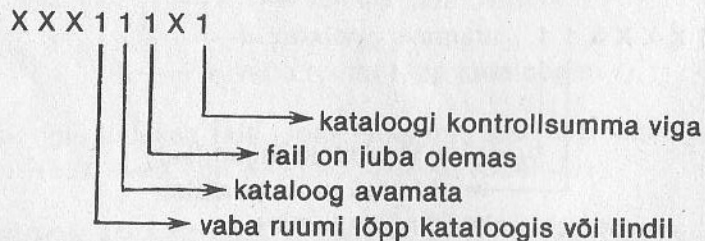
Kataloog kirjutatakse mälust lindile ja kuulutatakse suletuks.

FUNKTSIOON 49: CREATE FILE — Etteantud pikkusega faili loomine

Sisendparameetrid: DE = FJP adress.

A = vajalik faili pikkus 2K baidistes ühikutes

Väljund: C = funktsiooni olek:

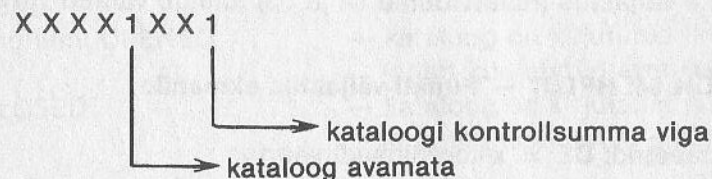


Lindile luuakse tühi fail staatusega OLD. Selleks leitakse lindilt vaba ruum faili plokkide jaoks, luuakse kataloogis faili kirje ning faili kuuluvad plokid kuulutatakse hõivatuiks. Fail jääb suletuks.

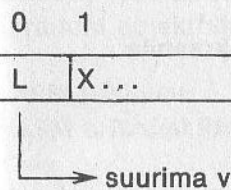
FUNKTSIOON 50: MAX FREE SPACE — Lindi maksimaalse vaba lõigu pikkuse leidmine

Sisendparameetreid ei ole

Väljund: C = funktsiooni olek:



Arvutatakse suurima järjestikuse vaba ruumi pikkus lindil. Saadud tulemus kantakse 128-baidisesse otsemällupöörduse puhvrisesse 2K baidistes ühikutes:



FUNKTSIOON 51: EXIT — Väljumine monitori

Sisendparameetreid ei ole

Väljundandmeid ei ole

Väljutakse operatsioonisüsteemist monitori. Lindisüsteem lähtestatakse funktsiooniga 13 (Reset tape system).

FUNKTSIOON 52: HGR — Ekraani viimine graafikarežiimi

Sisendparameetrid: E = väljastusrežiim (vt.7.2)

Väljundandmeid ei ole

Graafikarežiimi kursor viiakse punkti koordinaatidega (0,0). Värvuseks võetakse valge. Kustutatakse tärgirežiimi kursor (samuti funktsioonide 53,54 ja 55 puhul).

FUNKTSIOON 53: HCOLOR — Värvuse valimine graafikarežiimis

Sisendparameetrid: E = värvuse indeks (vt. 7.3.2)

Väljundandmeid ei ole

Graafiline väljastus (funktsioonid 54 ja 55) toimub valitud värvusega.

FUNKTSIOON 54: HPLLOT — Punkti väljastus ekraanile

Sisendparameetrid: DE = x-koordinaadi väärtus

B = y-koordinaadi väärtus

Väljundandmeid ei ole

Graafikarežiimi kursor nihutatakse punkti koordinaatidega (x,y). See punkt esitatakse ekraanil funktsiooniga 53 valitud värvusega.

FUNKTSIOON 55: HPLLOT TO — Sirglõigu väljastus ekraanile

Sisendparameetrid: DE = x-koordinaadi väärtus

B = y-koordinaadi väärtus

Väljundandmeid ei ole

Lõigu alguspunkti määrab graafikakursori positsioon (s.o. viimavalitud punkti koordinaadid). Lõik esitatakse funktsiooniga 53 valitud värvusega. Alguspunkti esitust ei uuendata.

8.5 OPERATSIOONISÜSTEEMI TEATED

CHECKSUM ERROR	— kontrollsumma viga
BLOCK NOT FOUND	— ei leitud osutatud plokki
NOT FOUND DIRECTORY BLOCK	— ei leitud kataloogi plokki
FILE NOT FOUND	— kataloogist ei leitud osutatud faili
FILE NOT FOUND OR FILE EXISTS	— faili ei leitud või nimi on juba olemas
FILE NOT OPEN	— fail ei ole avatud
DIRECTORY NOT OPEN	— kataloog ei ole avatud

TAPE FULL OR DIRECTORY FULL	— kataloog või lint on täis
END OF FILE	— faili lõpp
TIME OUT	— valvetaimeri rakendumine
FILE IS READ ONLY	— lugemisfail
BAD ADDRESS	— väär aadress
NO FILE(S)	— faile ei ole
TAPE lindinimi OPENED	— kataloog on kirjutatud lindilt mällu ja lint kuulutatud avatuks
TAPE CLOSED	— kataloog on kirjutatud mälust lindile ja lint kuulutatud suletuks

9 ASSEMBLER**9.1. SISSEJUHATUS**

Assembler loeb lähteteksti magnetlindilt ja annab lindile välja transleeritud programmi objektifaili kujul. Assembler **käivitatakse** korraldusega:

ASM failinimi
või **ASM failinimi.PARM**

Mõlemal juhul otsitakse lindilt faili

failinimi.ASM,

mis peab sisaldama assemblerkeeles lähteteksti. Teine väljakutsevorm lubab kasutada parameetrit, mille abil saab näidata, kas töö tulemusena saadakse listingu- või objektifail. Kui parameetrit ei anta, kirjutatakse lindile objektifail

failinimi.OBJ

Kui kasutada translaatori väljakutsumiseks vormi

ASM failinimi.P,

kirjutatakse lindile listingufail, mis sisaldab lähteteksti, genereeritud koodi ja veamärgise.

Väljakutse:

ASM.failinimi.X

Lindile kirjutatakse transleeritud programmi fail ja listing saadetakse ekraanile.

9.2 PROGRAMMI VORMING

Translaatorile vastuvõetav assemblerkeelne käsk võib sisaldada järgmisi väljasid:

reanr märgend operatsioonikood operand; kommentaar,

kus iga väli võib esineda või puududa, sõltuvalt olukorrast. Iga assembler-käsu rida lõpetatakse märkidega RETURN ja LF. Ühel real paiknevad käsud eraldatakse üksteisest hüüumärgiga «!».

Rea number võib esineda või puududa. Rea numbriks võib olla suvaline kümnendarv.

Märgend võib olla tähega algav suvaline kuni 16-märgiline tähtede ja numbrite jada. Kõik märgendielemendid on tähenduslikud, välja arvatud märk «\$», mida võib kasutada märgendi loetavuse parandamiseks.

Operatsioonikoodi väli võib sisaldada assembleri direktiivi, pseudo-operatsiooni koodi või masinakäsu mnemokoodi.

Operandi väli võib sisaldada avaldist, mis on koostatud konstantidest, märgenditest ja loogilistest tehetest nende vahel. Kommentaari väljas võivad märgi «;» järel olla suvalised tärgid. Translaator lubab kasutada kommentaari tunnuseks ka märki «*» esimeses veerus.

Assemblerprogramm koosneb eespool kirjeldatud kujul vormistatud lausetest. Viimaseks lauseks on tavaliselt END-lause. Translaator ignoreerib kõiki END-ile järgnevaid lauseid.

9.3 OPERANDIDE KOOSTAMINE

Operandiks võib olla aritmeetika- ja loogikaoperaatoritega ühendatud märgenditest, konstantidest ja reserveeritud sõnadest koosnev avaldis. Avaldise väärtus arvutatakse transleerimise käigus ja see tohib olla kuni 16-bitine.

9.3.1 ARVKONSTANDID

Arvkonstant on 16-bitine arv; arvusüsteemi tähistab täht arvu lõpus. Lubatavad arvkonstantide arvusüsteemid on

- B — kahendkonstant (binaararv)
- O,Q — kaheksandkonstant (oktaalarv)
- D — kümnendkonstant (detsimaalarv)
- H — kuueteiskümnendkonstant (heksadetsimaalarv)

Kui arvul puudub arvusüsteemi näitav täht, siis loetakse ta kümnend-arvuks. Kuueteiskümnendkonstandi esimene number peab olema kümnendnumber, et mitte segada ära konstanti identifikaatoritega. Näiteks OFFH, mitte FFH.

9.3.2 RESERVEERITUD SÕNAD

On olemas hulk reserveeritud sõnu, millel on operandiväljal kindel tähendus ja väärtus. Need on järgmised protsessori registre nimesid.

A	7	Paremal pool on antud nimele vastav ühekordne väärtus
B	0	
C	1	
D	2	
E	3	
H	4	
L	5	
M	6	
SP	6	
PSW	6	

Operandiosas võib kasutada ka masina käsukoode. Nende väärtuseks on nende vaste masinakoodis.

Märgi «\$» väärtuseks operandiosas on järgmise käsu aadress.

9.3.3 SÜMBOLKONSTANDID

Sümbolkonstantide moodustamiseks võib kasutada suvalisi KO! - 8 märki-jadaid. Maksimaalne sümbolkonstandi pikkus on 64 märki ja sümbolkonstant peab olema ülakomade vahel. Ülakoma sümbolkonstandi keskel

tuleb lüüa kahekordselt. Enamikel juhtudel võib sümbolkonstant olla ühe või kahe märgi pikkune, välja arvatud käsk DB. Sümbolkonstandi väärtuseks on tema KOI-8 kood.

9.3.4 ARITMEETIKA-LOOGIKAOPERAATORID

Ülalkirjeldatud operande võivad ühendada järgmised aritmeetika- ja loogikaoperaatorid:

a + b	märgita aritmeetiline summa
a - b	märgita aritmeetiline vahe
+b	unaarne pluss
-b	unaarne miinus (0-b)
a * b	märgita aritmeetiline korrutis
a/b	märgita aritmeetiline täisarvuline jagatis
a MOD b	a/b jääk
NOT b	loogiline eitus
a AND b	loogiline korrutis (bitthaaval)
a OR b	loogiline liitmine, VÕI
a XOR b	välistav VÕI
a SHL b	a kahendnihe vasakule b biti võrra
a SHR b	a kahendnihe paremale b biti võrra

Aritmeetika- ja loogikaavaldiste moodustamiseks võib kasutada sulge. Kõik arvutused teostatakse 16-bitiste väärtustega.

9.3.5 OPERAATORITE PRIORITEET

Programmeerija töö lihtsustamiseks on operaatoritele kinnistatud prioriteedid. Võrdse prioriteediga operaatoreid töödeldakse vasakult paremale.

Operaatorite prioriteedid on kõrgeimast alates järgmised (võrdse prioriteediga operaatorid on ühes reas):

*,/,MOD,SHL,SHR

-,+

NOT

AND

OR,XOR

Näiteks avaldist

a-b MOD c*d SHL e

tuleb tõlgendada nii:

a-(((a MOD b)*c)SHL d)

9.4 ASSEMBLERI DIREKTIIVID

Assembleri direktiivid on mõeldud transleerimise ajal märgenditele väärtuste omistamiseks, mälutsoonide defineerimiseks, programmi algusaadresside määramiseks ja tingimuslikuks transleerimiseks. Iga direktiivi tähistatakse pseudooperatsioonikoodiga. Lubatavad pseudooperatsioonikoodid on järgmised:

ORG	— seada programmi algusaadressiks operandi väärtus
END	— lõpetada programmi transleerimine
EQU	— anda identifikaatorile algväärtus
SET	— anda identifikaatorile uus väärtus
IF	— alustada tingimuslikku transleerimist
ENDIF	— lõpetada tingimuslik transleerimine
DB	— anda andmebaidile operandi väärtus
DW	— anda andmesõnale operandi väärtus
DS	— defineerida mälu

Allpool defineeritakse pseudooperatsioonid täpsemalt.

9.4.1 ORG

Direktiivi ORG süntaks on järgmine:

märgend ORG avaldis

Siin **märgend** võib olla suvaline identifikaator. **Avaldis** on 16-bitise väärtusega avaldis, mille operandid peavad olema eelnevalt defineeritud. Assembler paigutab järgmise transleeritud käsu avaldisega määratud aadressile. Programmis võib ORG-lauseid olla suvaline arv. Mälupiirkondade ülekattuvust ei kontrollita. Märgendile antakse avaldisega määratud väärtus.

9.4.2 END

Direktiiv END võib programmis puududa. Kui ta aga esineb, siis määrab ta programmi lõpu. END-ile järgnevaid programmi lauseid ei transleerita. Direktiivil END on kaks vormingut:

märgend END
märgend END avaldis

Märgend on vabalt valitav. Esimese vormi puhul transleerimisprotsess lõpetatakse ja programmi käivitusaadressiks antakse 0000. Teise vormi puhul antakse käivitusaadressiks **avaldise** väärtus; see lisatakse objektprogrammi viimasesse kirjesse.

9.4.3 EQU

EQU-lauset kasutatakse programmi identifikaatoritele numbriliste väärtuste omistamiseks.

märgend EQU avaldis

Märgend on kohustuslik. Sama märgend ei tohi esineda programmis ühegi teise lause ees. Märgendile antakse avaldisega määratud väärtus.

9.4.4 SET

Direktiiv SET on sarnane direktiiviga EQU:

märgend SET avaldis

Erinevuseks on see, et SET-direktiivi ees olevat märgendit võib kasutada ka teistes SET-lausetes. Märgendi väärtus kehtib programmis seni, kuni ta uue SET-lausega ümber defineeritakse. SET-lauset kasutatakse tihti tingimuslikul transleerimisel.

9.4.5 IF JA ENDIF

IF ja ENDIF määravad lauserühma, mis võidakse transleerimise käigus programmile lisada või mitte.

IF avaldis
lause 1
lause 2
...
lause n
ENDIF

Kui transleerimine jõuab IF-lauserühmeni, arvutatakse avaldiselise väärtus. Kui avaldiselise väärtus erineb nullist, siis järgmised laused transleeritakse ja lisatakse programmi koosseisu. Kui väärtus on null, siis järgmised laused kuni ENDIF-ini jäetakse vahele.

9.4.6 DB

Direktiiv DB lubab programmeerijal määrata mäluühebaadi baitformaadis. Direktiivi kuju on järgmine:

märgend DB e1,e2,...,en

kus e1..en peavad olema avaldiselised, mis defineerivad 8-bitise väärtuse (kõrgemad järgud peavad olema nullid), või sümbolkonstandid pikkusega kuni 64 märki. Avaldiseliste väärtused arvutatakse ja paigutatakse masinakoodi faili järjestikuste baitidena. Märgikoodid sümbolkonstantidest paigutatakse objektprogrammi alates esimesest märgist ja lõpetades viimasega.

9.4.7 DW

DW-lause on sarnane DB-lausega, kuid selle korral salvestatakse väljundprogrammi 2-baidised väärtused. Käsundi kuju on

märgend DW e1,e2,...,en

e1..en väärtused on 16-bitised. Sümbolstringe pikkusega üle kahe märgi ei lubata kasutada. Objektprogrammi paigutatakse enne madalam ja siis kõrgem bait.

9.4.8 DS

DS-lauset kasutatakse mälu reserveerimiseks algväärtustamiseta. Direktiivi kuju on

märgend DS avaldis

Transleerimisel jätab assembler vahele **avaldisega** määratud arvu baite.

9.5 VEATEATED

Kui translaator avastab programmis vea, märgistab ta selle veatähisega, mis listingufailis paikneb käsu ees. Vigane rida väljastatakse ka ekraanile. Veatähised on järgmised:

- D** Andmete viga. Avaldise väärtus on suurem lubatust.
- E** Avaldise viga. Avaldise väärtust pole võimalik arvutada.
- L** Märghendi viga. Märghend on kontekstis lubamatu (näiteks korduv märghend).
- O** Ületäitumine. Avaldis on liiga komplitseeritud.
- P** Faasi viga. Märghendil ei ole sama väärtus kahes järgnevas faasis.
- R** Registri viga. Väärtus, mis on määratud registriks, ei vasta operatsiooni koodile.
- V** Väärtuse viga. Avaldises olev operand on lubamatu.

Kuvarile väljastatakse ka järgmised teated transleerimist katkestavate vigade kohta:

CANNOT OPEN SOURCE FILE	Nimetatud lähtetekstiga fail lindil puudub
NO DIRECTORY SPACE	Lindi kataloogis puudub ruum faili jaoks
SOURCE FILE NAME ERROR	Lähtefaili nimes on süntaksiviga
SOURCE FILE READ ERROR	Sisestusviga lähtefaili lugemisel
OUTPUT FILE WRITE ERROR	Viga väljundfaili kirjutusel
CANNOT CLOSE FILE	Väljundfaili ei saa sulgeda

9.6 OBJEKTFAILI STRUKTUUR

Programmid paigutatakse lindile kahel kujul:

- käsufailidena (COM),
- objektfailidena (OBJ).

Erinevalt käsufailidest võivad objektfailid olla transleeritud suvalisele mäluadressile ja neid on võimalik laadida samuti suvalisest aadressist alates. Objektfailid koosnevad plokkidest, mille pikkuse määrab spetsiaalne bait plokis. Maksimaalne võimalik pikkus on 255 baiti.

Objektfaili plokk koosneb järgmistest elementidest:

- alguse tunnus «:» (1 bait);
- andmebaitide arv plokis (1 bait);
- ploki laadimisadress (2 baiti);
- andmebaidid (max 255 baiti).

Faili lõpu tunnuseks on plokk, mille pikkuseks on 0. Translaatorid vormistavad tavaliselt ploki pikkusega 20 baiti.

MIKROPROTSESSORI KP580MK80 KÄSUSTIK

Käesolev lisa on toodud illustatsiooniks, sest programmeerimisel tuleb kasutada arvuti juhendeid. Käsukoodi parameetiline osa on tabelis tähistatud tähtedega X ja Y. Muutuva koodiga käskudel on tabelis toodud üksnes kahendsüsteemne kuju, sest parameetribitte pole võimalik esitada kuueteistkümmendsüsteemis.

Mnemokoodide sisu avamiseks on ingliskeelsetes käsunimedes esitatud võtmetähed suurtähtedena.

Mnemokood	Masinakood		Käsu pikkus baitides	Selgitused
	16nd- süsteem	2nd- süsteem		
1	2	3	4	5
Käsud ülekande lipu muutmiseks				
CMC	3F	0011 1111	1	<i>CoMplement Carry</i> Ülekande lipp pöö- ratakse $CY = CY + 1$
STC	37	0011 0111	1	<i>SeT Carry</i> ülekande lipp seatakse $CY = 1$
Käsud registri ja mälupeesa sisu muutmiseks				
INR r		00XX X100	1	<i>INcRement register</i> Käsu koodis osuta- tud numbriga re- gistri sisu kasvata- takse ühe võrra
DCR r		00XX X101	1	<i>DeCRement regis- ter</i> Nimetatud re- gistri sisu kahan- datakse ühe võrra
INR M		00XX X100	1	<i>INcRement Memo- ry</i> Registripaar- is H&L oleva aadres- siga mälupeesa sisu kasvatatakse ühe võrra
DCR M		00XX X101	1	<i>DeCRement Memo- ry</i> Nagu INR M, kuid kahandatakse ühe võrra

Muudavad lippe: CY,Z,S,P,AC

1	2	3	4	5
Ühebaidise operandiga aadressita käsud				
CMA	2F	0010 1111	1	<i>CoMplement A</i> Re- gistrisse viiakse A sisu pöördkood
ENNE (A) = 0101 1011 = 5BH PÄRAST (A) = 1010 0100 = A4H				
DAA	27	0010 0111	1	<i>Decimal Adjust A</i> Kümnenndkorrektsi- oon, kasutatakse pärast detsimaalar- vudega tehtud arit- meetikatehet
KIRJELDUS: 8-bitine arv akumulaatoris viiakse 16-süsteemist kahend- kümnenndkoodi järgmiselt:				
1) Kui madalamas 4 bitis on arv >9 või AC=1 (ülekanne kõrgemasse poolbaiti), liidetakse aku sisule 6;				
2) Kui nüüd neljas kõrgemas bitis tekib arv >9 või CY=1 (ülekanne kõrgeimast bitist), liidetakse kõrgemale poolbaidile 6.				
NOP	00	0000 0000	1	<i>No OPeration</i> Kä- suloendurit kasva- tatakse ühe võrra
Käsud ühe baidi teisalduseks				
MOV r1,r2		01XX XYYY	1	<i>MOVe register r1 to register r2</i> Regist- rist 2 saadetakse bait registrisse 1
MOV M,r		01XX XYYY	1	<i>MOVe register to memory</i> Registrist r saadetakse bait peessa, mille aad- ress on registripaa- ris H&L
MOV r,M		01XX XYYY	1	<i>MOVe memory to register</i> Vastupid- ine käsule MOV M,r

1	2	3	4	5
STAX B	02	0000 0010	1	<i>Store A indirect via B Registris A olev bait salvestatakse aadressile, mis on registripaaris B&C</i>
STAX D	12	0001 0010	1	<i>Store A indirect via D Nagu eelmine, B&C asemel on D&E</i>
LDAX B	0A	0000 1010	1	<i>Load A indirect via B Registrisse A laaditakse bait, mis on B&C-s oleva aadressiga mälupesas</i>
LDAX D	IA	0001 1010	1	<i>Load A indirect via D Nagu eelmine, B&C asemel on D&E</i>

Need käsud lippe ei muuda.

Aritmeetika- ja loogikakäsud

ADD r		1000 0XXX	1	<i>ADD r to A Registris r olev bait liidetakse A sisule</i>
-------	--	-----------	---	---

Olgu (D)=2EH ja (A)=6CH, siis ADD D paneb 2EH+6CH=9AH akusse, Z ja CY viiakse nulli, P=S=AC=1.

ADC r		1000 1XXX	1	<i>ADD r to A with Carry Nagu eelmine, juurde liidetakse veel ülekande lipp CY</i>
SUB r		1001 0XXX	1	<i>SUBtract r from A A sisust lahutatakse registris r olev bait</i>

1	2	3	4	5
---	---	---	---	---

Lahutamiseks viiakse r sisu enne täiendkoodi ja see liidetakse; kui see juures ei olnud ülekannet kõrgeimast järgust, s.t. laenu ei olnud, siis CY=1, vastasel korral CY=0.

Näit. SUB A

3EH=00111110

$$\begin{array}{r} + \\ (-3EH) = 11000001 \text{ (pöördkood)} \\ + \quad \quad \quad 1 \text{ (täiendkood)} \\ \hline \end{array}$$

(1)00000000

Et ülekanne oli, siis CY=0, AC=P=Z=1, S=0.

SBB r		1001 1XXX	1	<i>Subtract r from A with Borrow A sisust lahutatakse r sisu, millele on liidetud ülekande lippu CY väärtus</i>
-------	--	-----------	---	---

Näit. SBB L

(L)=2, (A)=4, CY=1, siis

1) 02H+CY=03H,

2) 03H täiendkood on 11111101 ja

3) 04H=00000100

$$\begin{array}{r} + \quad 11111101 \\ \hline 00000001 = 01H, \end{array}$$

Ülekanne muudab lippu CY, seega CY=0, P=Z=S=0, AC=1, (A)=01H.

ANA r		1010 0XXX	1	<i>AND register r with A A ja r loogiline korrutis, jääb A-registrisse</i>
XRA r		1010 1XXX	1	<i>exclusive OR, register r with A Välis-tav VÕI-tehe A ja r sisudega</i>

1	2	3	4	5
ORA r		1011 0XXX	1	OR register r with A Registrite r ja A sisude loogiline summa
Kuna VÕI 1-ga annab 1, nulliga aga ei muuda väärtust, siis kasutatakse seda tihti bitigrupile ühtede omistamiseks.				
CMP r		1011 1XXX	1	CoMPare register r with A Võrreldakse r ja A sisu, A ei muutu

Võrdlus toimub sisemise lahutamise, seega lipp Z=1, kui tulemused võrdsed; CY=1, kui ei toimunud ülekannet kõrgeimast järgust, s.t. kui r sisu on suurem akumulaatori omast, vastasel korral CY=0.

Märkus: Kõigis aritmeetika- ja loogikakäskudes võib r tähistada ka paari H&L, seega mälupea aadressi

Nihkekäsud

RLC	07	0000 0111	1	Rotate A Left A sisu ringnihe vasakule, vasakult väljanihkuv bitt ilmub ka ülekande lippu CY
RRC	0F	0000 1111		Rotate A Right Nagu RLC, kuid paremale
RAL	17	0001 0111		Rotate A Left trough carry Nagu RLC, kuid ülekande lipp CY on A laiendiks, s.t. endine CY väärtus siseneb A-sse paremalt
RAR	IF	0001 1111	1	Rotate A Right trough carry Nagu RAL, kuid nihe toimub paremale

1	2	3	4	5
Käsud kahe baidiga opereerimiseks				
PUSH rp		11XX 0101	1	PUSH register pair rp on stack Registripaar rp salvestatakse pinusse; rp võib osutada paare B&C, D&E, H&L
PUSH PSW		11XX 0101	1	PUSH A and flags on stack Registri A sisu ja olekulipud (CY,AC,C1,Z,S,P) salvestatakse pinusse
POP rp		11XX 0001	1	POP register pair rp off stack Registripaar rp võetakse pinust ja salvestatakse rp-sse; rp võib osutada paare B&C, D&E, H&L
POP PSW		11XX 0001	1	POP A and flags (Program Status Word) off stack A ja olekulipud ennistatakse pinu järgi
DAD rp		00XX 1001	1	Add register pair to H&L (Double ADd) Topeltpikkusega arvude liitmine; rp võib osutada B&C, D&E, SP, H&L

Näide: Olgu (B)=33H, (C)=9KH, (H)=0AIH, (L)=7BH, siis käsk DAD B teeb (BC)=339F + (HL)=A17B (HL)=05A1, kuna ülekannet ei olnud, siis CY=0.

1	2	3	4	5
INX rp		00XX 0011	1	<i>INcrement register pair rp</i> rp sisu kasvatatakse ühe võrra
DCX rp		00XX 1011	1	<i>DeCrement register pair rp</i> Sisu kahandatakse ühe võrra
XCHG	EB	1110 1011	1	<i>eXCHanGe D&E and H&L register pairs</i> Vahetatakse omavahel H&L ja D&E
XTHL	E3	1110 0011	1	<i>eXchange Top of stack to H&L</i> Vahetatakse omavahel viimane sissekanne
SPHL	F9	1111 1001	1	pinus ning H&L <i>Load SP from H&L</i> Registripaari H&L sisu võetakse uueks pinuviida väärtuseks

Viimased viis käsku ei muuda olekulippe

Vahetu adresseerimisega käsud

LXI rp,v		00XX 0001 value	3	<i>Load register pair Immediate</i> Käsukoodile vahetult järgnevad kaks baiti (v=väärtus) laaditakse registripaari rp;
MVI r,v		00XX X110 value	2	<i>MoVe Immediate operand to register</i> Käsukoodile vahetult järgnev bait laaditakse regist-

1	2	3	4	5
				risse r; kui r=H&L, siis mällu
Lipud jäävad muutmata				
ADI v	C6 v	1100 0110 value	2	<i>ADd Immediate to A</i> Käsukoodile vahetult järgnev bait liidetakse registri A sisule
ACI v	CE v	1100 1110 value	2	<i>Add Immediate to A with Carry</i> Käsukoodile vahetult järgnev bait liidetakse registri A sisule, tulemusele liidetakse CY
SUI v	D6 v	1101 0110 value	2	<i>SUBtract Immedia- te from A</i> Registri A sisust lahutatakse käsukoodile vahetult järgnev bait
SBI v	DE v	1101 1110 value	2	<i>Subtract Immedia- te from A with Bor- row</i> Nagu SUI, kuid lahutatakse veel ülekande lipp CY
ANI v	E6 v	1110 0110	2	<i>ANd Immediate with A</i> Registri A sisu ja käsukoodile vahetult järgneva baidi loogiline korrutis
XRI v	EE v	1110 1110 value	2	<i>eXclusive oR Imme- diate with A</i> Välis- tav VÕI-tehe registri A ja käsukoodile

1	2	3	4	5
ORI v	F6 v	1111 0110 value	2	vahetult järgneva baidiga <i>OR Immediate with A VÕI-tehe</i> registri A sisu ja käsukoodile vahetult järgneva baidiga
CPI v	FE v	1111 1110 value	2	<i>ComPare Immediate with A A</i> sisu võrdlus käsukoodile vahetult järgneva baidiga

Lipud muudetakse nagu vastavatel registrikäskudel

Otsese adresseerimisega käsud

STA ad	32 ad	0011 0010 address	3	<i>STore A direct</i> A sisu salvestatakse käsukoodi järel antud aadressile
LDA ad	3A ad	0011 1010 address	3	<i>LoaD A direct</i> Registrisse A laaditakse bait käsukoodi järel antud aadressilt
SHLD ad	22 ad	0010 0010 address	3	<i>Store H&L Direct</i> Registrate H&L sisu salvestatakse käsukoodi järel antud aadressile
LHLD ad	2A ad	0010 1010 address	3	<i>Load H&L Direct</i> Registritesse H&L laaditakse käsukoodi järel oleva aadressiga pesa sisu

Lippe ei muudeta

1	2	3	4	5
Siirdekäsud				
HLT	76	0111 0110	1	<i>HaLT</i> Käsuloendur viiakse järgmisele käsule; arvuti jääb seisu, millest saab edasi vaid katkestus- või restardisignaali abil
PCHL	E9	1110 1001	1	<i>Load PC from H&L</i> Toimub tingimatu siire aadressile, mis oli registris H&L
JMP ad	C3 ad	110 0011 address	3	<i>JuMP unconditional</i> Tingimatu siire käsukoodi järele kirjutatud aadressile
JC ad	DA ad	1101 1010 address	3	<i>Jump if Carry</i> Kui CY=1, siirduakse käsus toodud aadressile
JNC ad	D2 ad	1101 0010 address	3	<i>Jump if No Carry</i> Kui CY=0, siirduakse käsus näidatud aadressile
JZ ad	CA ad	1100 1010 address	3	<i>Jump if Zero</i> Kui Z=1, siirduakse käsus näidatud aadressile
JNZ ad	C2 ad	1100 0010 address	3	<i>Jump if Not Zero</i> Kui Z=0, siirduakse käsus näidatud aadressile

1	2	3	4	5
JM ad	FA ad	1111 1010 address	3	<i>Jump if Minus</i> Kui S=1, siirdutakse käsus näidatud aadressile
JP ad	F2 ad	1111 0010	3	<i>Jump if Positive</i> Kui S=0, siirdutakse käsus näidatud aadressile
JPE ad	EA ad	1110 1010 address	3	<i>Jump if Parity Even</i> Kui P=1, siirdutakse käsus näidatud aadressile
JPO ad	E2 ad	1110 0010 address	3	<i>Jump if Parity Odd</i> Kui P=0, siirdutakse käsus näidatud aadressile
CALL ad	CD ad	110 1101 address	3	<i>CALL unconditional</i> Tingimatu siire käsus näidatud aadressile, PC päästetakse pinuse (alamprogrammi väljakutse)
CC ad	DC ad	1101 1100 address	3	<i>Call if Carry</i> Kui CY=1, siis nagu CALL
CNC ad	D4 ad	1101 0100 address	3	<i>Call if No Carry</i> Kui CY=0, siis nagu CALL
CZ ad	CC ad	1100 1100 address	3	<i>Call if Zero</i> Kui Z=1, siis nagu CALL
CNZ ad	C4 ad	1100 0100 address	3	<i>Call if Not Zero</i> Kui Z=0, siis nagu CALL

1	2	3	4	5
CM ad	FC ad	1110 1100 address	3	<i>Call if Minus</i> Kui S=1, siis nagu CALL
CP ad	F4 ad	1111 0100 address	3	<i>Call if Positive</i> Kui S=0, siis nagu CALL
CPE ad	EC ad	1110 1100 address	3	<i>Call if Parity Even</i> Kui P=1, siis nagu CALL
CPO ad	E4 ad	1110 0100 address	3	<i>Call if Parity Odd</i> Kui P=0, siis nagu CALL
RET	C9	1100 1001	1	<i>RET</i> urn Tingimatu siire aadressile, mis on pinus kõige peal
RC	D8	1101 1000	1	<i>Return if Carry</i> Kui CY=1, siis nagu RET
RNC	D0	1101 0000	1	<i>Return if No Carry</i> Kui CY=0, siis nagu RET
RZ	C8	1100 1000	1	<i>Return if Zero</i> Kui Z=1, siis nagu RET
RNZ	CO	1100 0000	1	<i>Return if Not Zero</i> Kui Z=0, siis nagu RET
RM	F8	1111 1000	1	<i>Return if Minus</i> Kui S=1, siis nagu RET
RP	F0	1111 0000	1	<i>Return if Positive</i> Kui S=0, siis nagu RT
RPE	E8	1110 1000	1	<i>Return if Parity Even</i> Kui P=1, siis nagu RET

1	2	3	4	5
RPO	E0	1110 0000	1	<i>Return if Parity Odd</i> Kui P=0, siis nagu RET
RST n		11XX X111	1	<i>ReStart level n PC</i> kirjutatakse pinusse, juhtimine siirdub aadressile 8n (nagu n-nivoo katkestuse korral), 0 < n < 7

Juhtimine antakse aadressile 0000000000NNN000, s.t. RST number X 8

Sisend-väljundkäsud

EI	FB	1111 1011	1	<i>Enable Interrupts</i> Lubatakse katkestused
DI	F3	1111 0011	1	<i>Disable Interrupts</i> Keelatakse katkestused
IN ad	DB ad	1101 1011 address	2	<i>INput</i> Registrisse A sisestatakse üks bait välisseadmelt, mille füüsiline (vä-rati) aadress on antud käsukoodi järel oleva baidiga
OUT ad	D3 ad	1101 0011 address	2	<i>OUTput</i> Registris A olev bait väljastatakse seadmele, mille aadress on antud käsukoodi järel oleva baidiga

PROGRAMMEERIMISNÄITEID

1) Mitmebaidine liitmine ja lahutamine

Ülekandelippu ja käsku ADC saab kasutada suvalise pikkusega märgita arvu liitmiseks.

Näit.

219E79

+ 84BA90

A65909 (tulemus, mis paigutatakse ESIM kohale)

Liitmiseks kasutame ADC käsku.

MADD: LXI B,ESIM ; laadime ESIM aadressi
LXI H,TEIN ; laadime TEIN aadressi
XRA A ; CY nullimine
MVI E,3 ; loendurile algväärtus

VEEL:

LDAX B
ADC M ; salvestame ESIM
STAX B
DCR E
JZ LOPP
INX B
INX H ; järgmisele baidile
JMP VEEL

LOPP: ; tehtud

ESIM: DB 90H
DB 0BAH
DB 84H
TEIN: DB 79H,9EH,21H

2) Korrutamine

Vaatleme kahe 1-baidise arvu korrutamist. D-registris on korrutatav, C-registris korrutaja, B-registrisse pannakse tulemuse kõrgem bait, C-sse madalam.

MULT:	MVI	B,0
	MVI	E,9
MULT0:	MOV	A,C
	RAR	
	MOV	C,A
	DCR	E
	JZ	LOPP
	MOV	A,B
	JNC	MULT1
	ADD	D
MULT1:	RAR	
	MOV	B,A
	JMP	MULT0
LOPP:		

3) 16-kohalise kahend-kümneandaru lahutamise

Vähendatav (madalam bait ees) aadressil VAH, lahutatav aadressil LAH, tulemus salvestatakse esimese operandi kohale.

DSUB:	LXI	D,VAH	;D,E = vähendav
	LXI	H,LAH	
	MVI	C,8	;iga tsükkel lahutab kaks kohta
	STC		;ülekanne: lipp CY:=1
VEEL:	MVI	A,99H	
	ACI	0	;liida 0 CARRY-ga
	SUB	M	
	XCHG		
	ADD	M	
	DAA		;kümneandkorrektsioon
	MOV	M,A	
	XCHG		
	DCR	C	
	JZ	LOPP	
	INX	D	
	INX	H	;järgm. operandide adresseerimine
	JMP	VEEL	;võtta järgmised 2 numbrit
LOPP:	NOP		;TEHTUD

10 MINIASSEMBLER

10.1 OTSTARVE

Miniassembler on mõeldud mnemokoodis programmeerimiseks arvutitel, millel pole välismäluseadmeid ei programmi, lähteteksti ega translaatori salvestamiseks. Selle tõttu on tal traditsiooniliste assembleritega võrreldes hulk kitsendusi (neid käsitletakse järgmises jaotises), aga ka lisavõimalusi. Translaatorit võib säilitada ja temaga töötada püsimalus (on muidugi ka võimalus translaatori laadimiseks välismälust muutmällu). Translaator võib olla hea vahend assemblerikeele õppimiseks ja lühemate programmide koostamiseks ning silumiseks. Translaatorile on lisatud võimalused sisestatavate assemblertekstide salvestamiseks muutmällu ja nende redigeerimiseks, mällu olevate masinakoodis programmide pöördtransleerimiseks ning programmide töö juhtimiseks ja jälgimiseks.

10.2 ERIOMADUSED

10.2.1 ALGTEKSTI SISESTUS

Klaviatuurilt sisestatavat teksti töödeldakse märkhaaval. Kontrollitakse iga märgi sobivust konteksti. Märki, mis ei ole kontekstiga lubatud, ekraanile ei saadeta ja vastu ei võeta. Sisestatud märki kustutada ei saa. Kustutada võib kogu rea korraga. Rea kustutamiseks tuleb sisestada kood CNTRL Y. Tühikuid ei ole vaja sisestada, tabuleerimine on automaatne. Sisestatud tühikud väljastatakse ekraanile, kuid neid ei töödelda.

10.2.2 ERINEVUSED KÄSUKOODIDES

Osa üheselt määramata käsukoodide tuleb lõpetada komaga, s.t. koma on käsukoodi osa. Näiteks kui on sisestatud CM, pole teada, kas mõeldakse sisestada käsku CM, CMA, CMC või CMP. CM-i käsukood tuleb sisestada kujul CM,. Käsud, mis nõuavad koma käsukoodi lõpetamiseks, on:

CM; CP; IN; JM; JP; LDA; STA.

Ülejäänud käsukoodide lõpust saab translaator automaatselt aru.

10.2.3 MÄRGENDITE TÖÖTLEMINE

Identifikaatorite kasutamine on piiratud. Realiseeritud ei ole tavalisi translaatori juhtkorraldusi SET, EQU ja DS. Lubatud on kasutada fikseeritud formaadiga märgendeid. Märgendi esimene märk peab olema «.», millele järgneb suvaline number või täht. Märgendite maksimaalne arv on 32. Juhul, kui märgend on määramata ja teda kasutatakse käsu operandiosas, täidetakse transleeritud käsus ekraanil operandiosa nullidega. Mälusse salvestatakse operandiossa viit vastavale märgendite tabeli reale. Kui märgend määratakse (s.t. ta esineb märgendiväljas), siis asendatakse mälus viidad märgendi väärtusega. Ekraanil olevat teksti ei parandata.

10.2.4 KONSTANDID

Lubatud on kasutada kolme tüüpi konstante. Vaikimisi loetakse kõik arvud kuueteiskümnendkonstantideks. Kui tahetakse sisestada kümnendkonstanti tuleb see lõpetada tähega K. Sümbolkonstant võib olla pikkusega 1 bait (märk) ja temale peab eelnema märk «'» (ülakoma). Arvkonstanti võib sisestada suvalise pikkusega. Sõltuvalt kontekstist kasutatakse konstandi kahte või nelja viimast kohta.

10.3 KASUTAMINE

Pöördumine miniassembleri poole väljastab ekraanile teksti:

MINIASSEMBLER

=

Võrdusmärk esimeses positsioonis näitab, et translaator on käsurrežiimis, mitte programmi transleerimise režiimis. Käsurrežiimis võib sisestada käske, mis võimaldavad juhtida programmi transleerimist, silumist või töötamist mällu salvestatud tekstidega. Transleerimise juhtimise käsud on järgmised:

On Alustada programmi transleerimist aadressile n. Kui n puudub, transleeritakse programm vaikimisi määratud aadressile või jätkatakse aadressilt, kus transleerimine varem katkestati. Transleerimisrežiimi võib katkestada CTRL X vajutamisega või märgiga «:» esimeses positsioonis.

- Rn** Määrata suhtelise transleerimise aadressiks n. Käsk mõjutab märgendite töötlust. Arvestatakse, et märgendid transleeritakse aadressist n alates.
- I** Järgnevalt sisestatav programm transleerida ja ka täita kohe, esitades koos registrite sisuga. Suunamiskäsud täidetakse formaalselt, s.t. registrit PC muudetakse, kuid transleerimist jätkatakse järgmiselt käsult. Režiimist väljumiseks on vaja sisestada I - .
- Ln,m** Pöördtransleerida m käsku, alates aadressist n. Kui n puudub (L,m), algab pöördtransleerimine vaikimisi määratud aadressist või aadressist, kus pöördtransleerimine varem katkestati. Kui m puudub (Ln), siis transleeritakse 10 käsku aadressilt n.
- Järgmine juhtkäskude grupp võimaldab siluda mälusse salvestatud programme, anda neile juhtimist, vaadata ja muuta mälu ja registrite sisu.
- Gn,m** Anda juhtimine aadressile n ja peatada programmi töö aadressil m. Nii algus- kui ka stoppaadressid võivad puududa. Kui algusaadress puudub, siis alustatakse programmi täitmist programmiloenduri jooksvast väärtusest (register P).
- Tn** Transleerida programmi tööd n käsu ulatuses. Kui n puudub, võetakse vaikimisi n=1. Programmi täitmist alustatakse programmiloenduri (register P) jooksvast väärtusest. Ekraanil näidatakse täidetav käsk ja registrite seis pärast käsu täitmist.
- Un** Nagu käsk T, kuid ekraanile ei saadeta jooksvalt informatsiooni, esitatakse ainult registrite seis peale n käsu täitmist. Töötab kiiremini kui käsk T.
- XX** Näidata ekraanil protsessori registrite sisu.
- Xn** Muuta registrit n. Lubatud registrid on F(lipud), A (akumulaator), topelregistrid S (pinuviit) ja P(käsuloendur) ning registripaarid B,D,H (vastavalt BC, DE ja HL).
- Pn** Määrata programmi peatuspunkt aadressile n. Kui programmi on antud juhtimine käsuga G, siis peale peatuspunkti läbimist antakse juhtimine translaatorile. Peatuspunktide läbimise arvu loendatakse. Pn — kustutab peatuspunkti. P — kustutab kõik peatuspunktid.
- Dn,m** Saata ekraanile mälu sisu alates aadressist n kuni aadressini m kuueteiskümnendkujul.

- Sn** Muuta baithaaval mälu sisu alates aadressist n kuuteistkümnendkujul. Muutmise lõpetab mitteküueteistkümnendarvu sisestamine. Tühiku sisestusel jäetakse vastav bait muutmata.
- Mm,n,d** Saata mälu sisu piirkonnast algusaadressiga m ja lõppaadressiga n piirkonda, mille algusaadress on d.
- Fm,n,d** Täita aadressilt m algav ja aadressil n lõppev mälupiirkond baidiga d.
- Cm,n,d** Võrrelda mälupiirkonda, mille algusaadress on m ja lõppaadress n, piirkonnaga, mille algusaadress on d.

Järgmine juhtkaskude grupp on määratud töötamiseks mällu salvestatud assemblertekstiga ja teksti mällu salvestamise juhtimiseks. Assemblerit tekst salvestatakse mällu tihendatult, puuduvad tühikud ja tabuleerimiskoodid. Teksti parandamine on võimalik ainult teksti transleerimisel ühest mälupiirkonnast teise. Tekstiga töötamise käsud on järgmised:

- In** Määrata teksti salvestamise algusaadressiks n. Kõigi järgnevate transleeritavate käskude tekst salvestatakse sümbolkujul alates aadressist n. Transleeritud käsu järel näidatakse aadress, kuhu käsk on salvestatud.
- E** Teksti salvestamise lõpp. Peale seda korraldust sümbolkujul teksti enam ei salvestata.
- An** Määrata mälust transleeritava teksti algusaadressiks m. Käsk ei käivita transleerimist mälust.
- Nn** Transleerida mälust n käsku teksti. Ekraanile ilmub transleeritava käsu järele aadress, millelt käsk on transleeritud. Ekraanile saadetakse transleeritud käsk ja mällu salvestatakse käsukood.
- Vn** Jätta mällu salvestatud tekstist vahele n käsku. Transleeritud käsk ilmub ekraanile, kuid mällu koodi ei anta. Teksti salvestamise režiimis jäetakse antud tekstilõik salvestamata.

10.4 SALVESTATUD TEKSTIGA TÖÖTAMISE NÄIDE

= J 5000 salvestada tekst aadressile 5000
 = O 3000 transleerida programm aadressile 3000
 3000 <

Siit alates sisestame assembleriprogrammi

```
3000 < 3E 10          MVI  A,10    5000
3002 < B3 20          OUT  20      5007
3004 < 00             NOP   5011
3005 <                CTRL X      siirdume käsurežiimi
=
```

Nüüd tahame antud programmi lõiku parandada, lisades ühe käsu OUT-käsu ette ja asendades käsu OUT käsuga IN.

= J 6000 uus tekst salvestada aadressile 6000
 = A 5000 mälust transleeritava teksti algus
 = O 3000 programm transleerida endisesse kohta

```
3000 <                CTRL X      käsurežiimi tagasi
=N1                    transleerida 1 käsk

3000 < 3E 10          MVI  A,10    6000 5000
3002 < 21 34          LXI  H,1234  6007
3005 DB 20            IN   20      6016
3007 <                CTRL X      käsurežiimi
=V1 jätta üks käsk vahele
3007 < D3 20          OUT  20      5007
3007 <                CTRL X      käsurežiimi
= N1 transleerida 1 käsk
3007 < 00             NOP           6020 5011
3008 <
```

Nüüd on meil aadressil 6000 programmitekst

```
MVI  A,10
LXI  H,1234
IN   20
NOP
```

ja aadressil 3000 vastav transleeritud programm.

Režiimi vahetamiseks ja töö lõpetamiseks on assembleris kolm CTRL-klahvi, mida võib kasutada suvalisest seisust. Sisestamise keskel neid klahve ei arvestata.

CTRL X Üleminek käsurežiimi
CTRL Y Üleminek transleerimise režiimi
CTRL Q Väljumine assemblerist monitori või CP/M-i.

11 PÜSIMÄLU-BASIC

Peatükk sisaldab lühiülevaate ENSV TA Küberneetika Instituudi AT EKB-s realiseeritud BASIC-keelest.

11.1 ERISÜMBOLID

CTRL C	Katkestab programmi täitmise, viib translaatori käsurežiimi ja väljastab «READY»
CTRL H	Kustutab viimase sisestatud märgi
CTRL X	Kustutab sisestamisel oleva rea
RETURN	Lõpetab iga sisestatava rea
:	Eraldab samal real paiknevaid lauseid
?	Ekvivalentne käsuga PRINT
\$	Stringi muutuja tähis

11.2 ANDMETÜÜBID

String	— 0 kuni 255 märki
Täisarv	— -32768 kuni 32767
Ujukomaarv	— -1.7E+38 kuni 1.7E+38
Identifikaator	vaba pikkusega, koosneb tähtedest ja numbritest (esimene peab olema täht), arvestatakse kahte esimest märki.

11.3 KÄSUD

Käsk	Süntaks/Funktsioon	Näide
CHANGE	CHANGE Loob võimaluse lindi vahetamiseks	CHANGE
CLEAR	CLEAR <avaldis> Eraldab <avaldisega> määratud hulga baite stringipuhvrile	CLEAR 500
CLOAD	CLOAD «<failinimi>» Loeb lindilt mälusse programmi, mis asub tekstifailis <failinimi>.BAS	CLOAD «FAIL»
CLS	CLS Kustutab ekraani ja viib kursori vasakusse ülanurka	CLS
COLOR	COLOR= <number> Määrab värvuse (indeksi 0-7 vt. 7.3.2), millega teostatakse järgnevad graafikaoperatsioonid	COLOR=0
CONT	CONT Jätkab programmi täitmist, mille katkestas STOP-lause või CTRL C	CONT
CSAVE	CSAVE «<failinimi>» Kirjutab mälust lindile BASIC-programmi ja nimetab selle <failinimi>.BAS	CSAVE «ABC»
CUR	CUR <rida>, <veerg> Viib kursori positsioonile, mis on määratud kuvari rea ja veeru numbritega	CUR 10,20
DATA	DATA <konstandid> Säilitab mälus konstante, mida on võimalik lugeda READ-lausega	DATA 10,20, «ABC»
DEF FN	DEF FN <nimi>(<argument>)= <avaldis> Defineerib aritmeetilise ühe argumentiga funktsiooni	DEF FNSQ(x)= =X*X

DIM	DIM <massiivikirjeldus>[, <massiivikirjeldus>...] Eraldab mälu massiividele ja deklareerib nende indeksite maksimaalsed väärtused	DIM A(3), X\$(4,10)
END	END Lõpetab programmi täitmise	END
FOR	FOR <muutuja> = <avaldis> TO <avaldis> [STEP <avaldis>] Kasutatakse koos NEXT-lausega programmiridade grupi korduvaks täitmiseks. Muutuja väärtust suurendatakse STEP-iga määratud suuruse võrra (vaikimisi 1)	FOR I=1 TO 10
GOSUB	GOSUB <reanumber> Täidab BASIC-u alamprogrammi, mis algab näidatud realt.	GOSUB 1000
GOTO	GOTO <reanumber> Tingimatu siire näidatud reale	GOTO 190
HGR	HGR Viib interpretaatori graafikarežiimi	HGR
IF/THEN IF/GOTO	IF <avaldis> THEN { <lause>[:<lause>...] :<reanumber> } Kui <avaldis> ei ole null, täidetakse THEN-ile järgnevad laused või siirduakse näidatud numbriga reale. Kui <avaldis> on null, täidetakse järgmine rida	IF X>Y THEN M=X A<0 GOTO 300
INPUT	INPUT [«<tekst>»];<muutuja>[, <muutuja>...] Loeb andmeid klaviatuurilt ja omistab need vastavatele muutujatele	INPUT «NIMI»; A\$
LET	[LET]<muutuja> = <avaldis> Omistab muutujale väärtuse	LET A\$ = «012»

LIST	LIST[<reanumber>] Kuvab programmiread alates esimesest reast või reanumbriga näidatud reast. Kuvamise katkestab CTRL C	LIST 1000
NEW	NEW Kustutab programmi mälust	NEW
NEXT	NEXT[<muutuja>[, <muutuja>...]] Fikseerib FOR-tsükli lõpu	NEXT I
ON/GOSUB	ON <avaldis> GOSUB <rida> [, <rida>...] Sõltuvalt avaldise väärtusest (selle täisosast) täidetakse üks näidatud alamprogrammidest (Kui avaldis=1, siis esimene jne.)	ON I+1 GOSUB 200, 300
ON/GOTO	ON <avaldis> GOTO <rida> [, <rida>...] Sõltuvalt avaldise väärtusest (selle täisosast) siirdub täitmine ühele näidatud ridadest	ON LEN(A\$) GOTO 10,20,25
OUT	OUT <värat>,<bait> Väljastab andmebaidi väljundvõrgu	OUT I,D(K)
PLOT	PLOT[TO]<x>,<y>[TO <x>,<y>...] PLOT <x>,<y> joonistab kuvarile punkti koordinaatidega <x> ja <y>. PLOT TO <x>,<y> tõmbab sirglõigu graafikakursori asukohast punkti (<x>,<y>). PLOT <x1>,<y1> TO <x2>,<y2> tõmbab sirglõigu punktist (<x1>,<y1>) punkti (<x2>,<y2>).	PLOT 12,10 TO 120,100
POKE	POKE <aadress>,<bait> Kirjutab kümnendkujul antud baidi mälu näidatud aadressile	POKE 32000,255
PRINT	PRINT[<avaldis>[!,<avaldis>...]] Väljastab andmed kuvarile	PRINT A\$, I+2

READ	READ <muutuja> [, <muutuja>] Loeb andmed DATA-lausest ja omistab need vastavatele muutujatele	READ I,J,L\$
REM	REM <kommentaar> Lubab kasutajal kirjutada programmile kommentaare	REM-ALAM-PROGRAMM-
RESTORE	RESTORE Lähtestab DATA-viida nii, et lugemine algab uuesti esimesest DATA-lausest	RESTORE
RETURN	RETURN Lõpetab alamprogrammi täitmise ja pöördub tagasi väljakutse kohta	RETURN
RUN	RUN Käivitab programmi täitmise	RUN
STOP	STOP Katkestab programmi täitmise, väljastab BREAK-teate ja siirdub käsurežiimi	STOP
SYSTEM	SYSTEM Naasmine BASIC-ust opsüsteemi või monitori	SYSTEM

11.4 OPERAATORID

Sümbol	Funktsioon
=	Omistamine või võrdsuse kontroll
-	Lahutamine või negatiivne väärtus
+	Liitmine või stringide ühendamine
*	Korrutamine
/	Jagamine
õ	Astendamine
NOT	Loogiline eituse
AND	Loogiline korrutis
OR	Loogiline liitmine
=, <, >	Võrdlus (resultaadiks on
< =, > =	TRUE = -1 või FALSE = 0)
< >	

11.5 FUNKTSIOONID

X,Y,I ja J tähistavad numbrilisi avaldisi
X\$ ja Y\$ tähistavad stringiavaldisi

Funktsioon	Väärtus	Näide
ABS(X)	X absoluutväärtus	Y=ABS(A-3)
ASC(X\$)	X\$ esimese märgi KOI-kood	? ASC("K")
ATN(X)	X arkustangens	PRINT ATN(B)
CHR\$(X)	väljastab märgi, mille KOI-kood on X	PRINT CHR\$(7)
COS(X)	X koosinus	A=COS(3.14)
EXP(X)	e astmes X	B=EXP(U)
FRE(0)	Vaba mälu pikkus	PRINT FRE(0)
FRE("α")	Vaba stringipuhvri pikkus	PRINT FRE("α")
INKEY\$(1)	Ühemärgiline string, mis loetakse klaviatuurilt (tühi string, kui ükski klahv pole alla vajutatud)	G\$=INKEY\$(1)
INP(X)	Andmebait sisendvõrjatist	X O=INP(21)
INT(X)	Suurim täisarv, mis on väiksem kui X	C=INT(RND(1)*100)
LEFT\$(X\$,Y)	X\$ vasakpoolsed Y märki	PRINT LEFT\$(X\$,6)
LEN(X\$)	X\$ pikkus	PRINT LEN(B\$)
LOG(X)	X naturaalloogaritmi	G=LOG(Y-3)
MID\$(X\$,X[,Y])	Stringi X\$ alamstring X-ndast tärgist, Y märki; kui Y pole antud, siis lõpuni	A\$=MID\$(X\$,5,10)
PEEK(X)	Andmebait mälust aadressilt X	PRINT PEEK(32700)
POS(1)	Kursori positsioon kuvaril	IF POS(1)>20...
RIGHT\$(X\$,Y)	Stringi X\$ parempoolsed Y märki	C\$=RIGHT\$(A\$,10)
RND(1)	Juhuslik arv vahemikus 0 kuni 1	?RND(1)*100

SGN(X)	0, kui X=0 ABS(X)/X, kui X < > 0	A=SGN(I)
SIN(X)	X siinus	A=SIN(B)
SPC(X)	PRINT-lauses väljastab X tühikut	?SPC(5),A\$
SQR(X)	X ruutjuur	D=SQR(C)
STR\$(X)	Teisendab X stringiks	PRINT STR\$(28*I)
TAB(X)	PRINT-lauses viib kursori positsioonile X	PRINT TAB(10);
TAN(X)	X tangens	A=TAN(3.14*I)
USR(X)	Kutsub välja masinakoodfunktsiooni aadressil X; funktsioon tagastab baidise väärtuse A-registris	A=USR(5000)
VAL(X\$)	String X arvkujul	X=VAL(«3.14»)

11.6 VEATEATED

Vea nr.	Vea tüüp
1	NEXT ilma FOR-ita
2	Vigane süntaks
3	RETURN ilma GOSUB-ita
4	Andmete lõpp
5	Vigased andmed
6	Ületäitumine
7	Mälu on täis
8	Defineerimata rida
9	Massiivi indeks on väljaspool lubatud piire
10	Korduvalt deklareeritud(DIM) massiv
11	Jagamine nulliga
12	Keelatud käsk käsurrežiimis
13	Tüüpide vastuolu
14	Stringipuhver on täis
15	String on liiga pikk
16	Stringiavaldis on liiga keeruline
17	Ei saa jätkata(käsus CONT)

18	Defineerimata funktsioon
19	Faili ei ole
20	Otserežiimi käsk failis
21	Vigane failinimi
22	Sisend-väljundoperatsiooni viga

12 PL/M

PL/M on masinorienteeritud kõrgkeel. PL/M-keeles saab arvutit süsteem-programmeerida, pääseb ligi protsessori ressurssidele: registritele, mälule, pinule. Lisaks on võimalik siduda PL/M ja assemblerprogramme.

PL/M-keeles saab kasutada kaasaegset struktuurprogrammeerimise meetodit.

PL/M-keele programmi struktuur, keele operaatorid ja protseduurid on kirjeldatud lisaköites «Programmeerimiskeel PL/M».

12.1 ÜLDANDMED

PL/M-keeles kirjutatud programm koosneb kahte tüüpi lausete jadadest:

- töödeldavate andmete kirjeldused
- nõutavate tegevuste kirjeldused

Andmed kirjeldatakse PL/M-keeles nn. **deklareerimislausetega**, tegevused **operaatoritega**. Andmed esitatakse kas konstantidena või muutujatena. Muutujat võib kasutada deklareerivalt või kasutavalt (viimasel juhul on ta operaatori koosseisus). Esimesele muutuja kasutavale esinemisele programmis peab eelnema tema deklareerimine. Iga muutujat võib programmis deklareerida ainult üks kord. PL/M on plokkstruktuuriga keel. Plokis saab deklareerida uusi muutujaid, andes nende nime ja tüübi, vajaduse korral ka algväärtuse, ja juhtida mälujaotust.

PL/M-keeles on kahte tüüpi andmeid: baiditüüpi ja aadressitüüpi, mis on vastavalt 8- ja 16-bitised. Muutujaid saab kokku ühendada massiivideks või struktuurideks.

Kui operaatorile antakse nimi, mille kaudu tema poole pöördudes ta täidetakse, siis nimetatakse operaatorit **protseduuriks**. Protseduur määratakse nn. protseduurikirjelduses. Protseduuri kirjeldus võib sisaldada andmete deklareerimisi ja täiendavaid protseduurikirjeldusi, mis on siis lokaalsed protseduuris, kus nad kirjeldatakse.

PL/M-keeles on operaatorid hargnemistingimuste kontrollimiseks, tsükli juhtimiseks, protseduuride väljakutsumiseks ja parameetrite üleandmiseks. Sisend-väljundoperaatorid lubavad baiditüüpi andmeid kirjutada lugeda arvuti väratite (portide) kaudu, keerulisemate sisend-väljundoperatsioonide jaoks saab kirjutada protseduurid, mis kasutavad kas monitori või opsüsteemi funktsioone.

12.2 KOMPILAATORI KASUTAMINE

Selles jaotises esitatakse kompilaatori kasutamise juhend, ta seos operatsioonisüsteemiga ja ta funktsioonide kasutamise viisid. Kompilaator on ette nähtud PL/M-keelse programmi ühekäiguliseks transleerimiseks. Lähtetekst peab olema lindil ja mitte suurem kui 6K baiti. Kompilaatori väljundiks võib olla transleeritud programm objektfailina, või listingufail lindil, samuti võib listingufaili väljastada kuvarile. Transleerimise parameetrid võib anda kas kompilaatori poole pöördumisel (väljakutsel) või kompileeritava programmi lähtetekstis.

Kompilaator on lindil failina PLM.COM ja kompilaator nõuab tööks operatiivmälu 0H..C600H. Kompileeritav tekst peab olema lindil ja selle lähtefaili nime laiendiks peab olema .PLM. Genereeritav väljundkood esitatakse failis ???OBJ ja selle vorming on esitatud jaotises 4.6. Listingufail viiakse lindile nimega ???PRN ja selle väljastus võib olla keelatud kas osaliselt või täielikult vastavate kompilaatori juhtimise direktiividega lähtetekstis.

Assemblerkeeles loodud programme ühendatakse PL/M-keelsetega kas silumisprogrammi SID abil või kasutades OBJ-faili laadurit ja salvestusprogrammi.

12.2.1 PÖÖRDUMINE

Kompilaatori väljakutse vorming on järgmine:

A>PLM

misjärel kompilaator laetakse lindilt mälli. Et translaator on küllalt pikk, siis kestab see toiming mitu minutit. Kui PL/M käivitub, ilmub kuvarile viip:

input command?

millele tuleb vastata:

<nimi>[.XY] \$<parameetrid> ,

kus

<nimi> — lähtefaili nimi (laiendiga .PLM)

X — määrab listingu väljastuse,

A — lindile, siis OBJ-faili ei väljastata,

T — kuvarile,

N — ei väljastata,

Y — määrab OBJ-faili väljastuse

A — lindile,

N — ei väljastata;

<parameetrid> — listingu väljastust juhtivad parameetrid, mis selgitatakse allpool

Näiteid:

- 1) A>PLM TEST, transleerida lähtefail TEST.PLM, viia TEST.OBJ lindile, listingufaili moodustamiseta
- 2) A>PLM TEST.T — listing kuvarile, TEST.OBJ lindile
- 3) A>PLM TEST.A — listing lindile TEST.PRN failina (objektfaili ei looda)

Listingu väljastamise lubatud parameetrid:

Parameeter	Funktsioon
H	Aadressiloenduri väärtuse väljastus iga lähteteksti rea ees
C	Genereeritud assemblerkoodi väljastus koos listingufailiga
S	Sümbolitabeli väljastus
NW	Kompilaatori hoiatuslausete väljastuse keelamine

PL/M-keelse programmi lähteteksti lisatakse kompilaatori juhtkäsud, vorminguga

\$<käsu kood>[<parameetrite loetelu.];

Need käsud võivad olla transleeritava programmi tekstis suvalises kohas, iga käsk peab olema eraldi real ja algama esimesest positsioonist, kommentaarid pole lubatud.

Käsu kood	Funktsioon
O	objektfaali juhtimine
P	listingu juhtimine.

Käsuga \$O saab kasutaja anda koostatava objektkoodi paigutamise aadresse:

\$O (<aadressid>), kus

<aadressid> = [<adr1>][<adr2>][<adr3>],

<adr1> — koodiosa paigutamise aadress;

<adr2> — andmete paigutamise aadress (muutujad ja konstandid);

<adr3> — nn. mäluvektori aadress (vt. PL/M keele kirjeldus).

Aadressid antakse PL/M-keeles ettenähtud viisil, s.t. kuueteistsüsteemi arvudel peab lõpus olema H. Seega paigutatakse käsu \$O järel tulev programm käsus näidatud mälupeas alates. Kui seda käsku pole kasutatud, on koodiosa algusaadress 100H, andmevälja algusaadress on koodi algusaadress + 4000H, mäluvektori algusaadress on OH.

Käsuga \$P[<p>] juhitakse listingu väljastamist. Kui p=L, siis listingu väljastus lülitatakse sisse, p=N-listingut ei väljastata.

Transleerimise lõpul väljastatakse listingufaili lõpus järgmine info:

koodiväli:

...CODE AREA:<algusaadress>.<lõpuaadress+1>

tööväli:

...WORK AREA:<algusaadress>.<lõpuaadress+1>

mälukontroll:

...MEMORY START:<mäluvektori algusaadress>

ja kuvarile väljastatakse:

*****END OF COMPILATION.nn. DIAGNOSTICS**

(«transleerimise lõpp, nn veateadet»).

Transleerimise käigus väljastatakse kahte tüüpi veateateid:

- vigadest töös failidega,
- vigadest transleeritavas lähtetekstis.

Esimesel juhul väljastatakse failile veatekst, teisel juhul veanumber.

12.3 VEATEATED

12.3.1 VEAD TÖÖS FAILIGA

TAPE OVERFLOW.FILE OBJ CANNOT BE WRITTEN

— lint täis, ei saa sulgeda OBJ faili

DIRECTORY IS FULL — kataloogi ületäitumine

TAPE FULL:PRN — lint täis:PRN

NO PLM FILE — pole faili tüübiga PLM

12.3.2 PROGRAMMI SÜNTAKSIVEAD

20H	Lubamatu märk DO järel
21H	Puudub END
22H	Väär nimi END järel
23H	DO WHILE järel puudub «;»
24H	Plokis DO WHILE pole deklareerimine lubatud
25H	Operaatoris DO CASE puudub «:»
26H	Operaatoris DO CASE pole deklareerimine lubatud
27H	Sammu määramisel väär alumine piir
28H	Sammu määramisel puudub märk «=»
29H	Sammu määramisel puudub TO
2AH	Sammu määramisel puudub «;»
2BH	Iteratiivses tsükklis pole deklareerimised lubatud
2CH	Tingimusoperaatoris puudub THEN
2DH	Puudub operaator IF
2EH	DO CASE operaatoris liiga palju harusid või määramata märgendid
2FH	Liigne END
30H	Puudub identifikaator
31H	Kaks korda defineeritud identifikaator
33H	BASED järel puudub identifikaator

34H	Vigaselt loodud identifikaatorite nimistu
35H	Vigane välise parameetri atribuut
36H	Märgend on baseeritud tüüpi
37H	Märgend on massiivi tüüpi
38H	Puudub tüüp
39H	INITIAL-i deklareerimine keelatud
3AH	INITIAL-i nimistu liiga pikk
3BH	Identifikaatori nimistu pole selles kontekstis lubatud
3CH	LITERALLY järel puudub tekst
3DH	Baasi identifikaator pole aadressitüüpi
3EH	Massiivi defineeritud pikkuse ületamine
3FH	Puudub «(»
40H	Baseeritud formaalne parameeter
41H	Mitteskalaarne formaalne parameeter
42H	Vääralt antud või olematu register
43H	Kaks korda kirjeldatud register
44H	Varem kirjeldatud element
45H	Baasi identifikaator ei ole struktuuri liige
46H	Vigane parameetrite nimistu
47H	Puudub «,» või «)»
48H	Vääralt lõpetatud elementide nimistu
49H	BY järel puudub identifikaator
4AH	BY järel pole struktuur
4BH	Konstantide nimistu vormistuse viga
4CH	Konstantide loetelus puudub «,»
4DH	Viga literaali(märgistringkonstandi) defineerimiseks
4EH	Lubamatu väljumine protseduurist
4FH	Siire protseduurist kõige sisemisse plokki
50H	Lubamatu märk
51H	Liiga pikk string, näiteks puudub «'»
52H	Lause süntaksi viga
53H	Lause süntaksi viga
54H	Puudub «;»
55H	Operaatori väär algus
56H	Number lubamatus kohas
57H	Numbriline märgend
58H	Korduvalt defineeritud märgend

59H	Siirdeoperaatoris viidatav märgend defineeritud väär kohas
5AH	Määramata märgend viimases plokkis
5BH	GO järel puudub TO
5CH	Siire absoluutaadressile
5DH	GOTO järel puudub aadress
5EH	Suunamine mitteaadressmuutuja järgi
5FH	GOTO järel puudub viide märgendile
62H	Literaali kihilisus liiga suur
63H	Liiga palju identifikaatoreid või literaale
65H	Arv ületab 16 bitti
66H	Lubamatu märk arvus
67H	Koodiväli ja andmeväli kattuvad
68H	Koodiväli ja mäluvektor kattuvad
69H	Viga kirjelduses
6BH	Väär «*» kasutamine
6CH	Puudub «;» käsu järel
6DH	Käsk \$REENTRANT antud liiga hilja
6FH	Puudub «,»
70H	Määramata identifikaator
71H	Puudub parameeter protseduuris
72H	Parameetri väär tüüp protseduuri kutses
73H	Parameetri järel puudub «)»
74H	Viga SHIFT-i transleerimisel
75H	Viga SHIFT-i transleerimisel
76H	Punkti kasutamise viga
77H	Puudub operand
78H	Vale STACKPTR/OUTPUT kasutamine
79H	Operatsioonile NOT peab eelnema operand
7AH	Operaator väär kohal
7BH	«(» väär kohal
7CH	Puudub «)»
7DH	Lubamatu operand vasakul pool võrdusmärki
7EH	Lubamatu indekseerimine või puudub operaator
7FH	Omistamisel puudub märk «=»
80H	Kompilaatori viga koodi genereerimisel
81H	Lubamatu struktuurielement
82H	Struktuur ei tohi olla operand
83H	Protseduur ilma tüübita ei tohi olla operand

84H	Katkestustöötluse protseduur ei tohi olla operand
85H	Funktsioon TIME ei tohi olla operand
86H	Tühi string on defineerimata
88H	Stringimuutuja on pikem kui 2 märki
89H	RETURN väljaspool protseduuri
8AH	Viga tagasipöördeoperaatori transleerimisel
8CH	Vigane rekursiivne väljakutse
8CH	Parameetriteta protseduurile antakse üle parameetreid
8DH	Liiga palju parameetreid (pöördumisel)
8EH	Viga parameetrite nimistu vormingus
8FH	Määramata parameeter
90H	Protseduuri nime korduv määramine
91H	Katkestustööde protseduur plokis või protseduuris
96H	Protseduuri pealdise järel puudub «:»
98H	Puudub protseduuri keha
99H	Väär nimi END järel
9BH	Vastuolu parameetrite arvus
9CH	Vastuolu parameetrite tüübis
9DH	Vastuolu protseduuri tüübis
9EH	Lubamatu identifikaator (defineeritud vääras kohas)
9FH	Protseduuri kutse süntaksi viga
AOH	Identifikaatori CALL järel ei ole protseduuri nimi
A1H	Funktsiooni kutses on CALL

13 SILUR SID

Programm SID on vahend assembler- ja PL/M-keelsete programmide silumiseks masinakeele tasemel. Saab jälgida programmi üksikkäskude täitmist, vaadata ja muuta mälu ning protsessori registreeride sisu.

13.1 ÜLDANDMED

13.1.1 KÄIVITUS

- (1) **SID** — silur käivitub, silutavat programmi ei laadita
 (2) **SID x.y** — programm x,y (üldjuhul y=COM) laaditakse aadressile 100H

- (3) **SID x.OBJ** — nagu (2), kuid fail on objektcodeis
 (4) **SID x.y u.v** — laaditakse x,y (üldjuhul x.SYM) ja märgenditabel u.v.

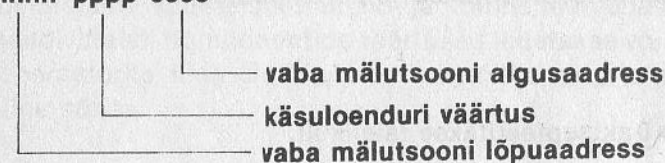
Näide:

SID SORT.COM SORT.SYM

SID väljastab pärast käivitamist järgmisi teateid:

- (1) **#** — silur ootab direktiivi
 (2) **SYMBOLS** — märgenditabel laaditud; tõrke korral «?» järgmisel real
 (3) **NEXT PC END** — programm (ja märgenditabel) laaditud

nnnn pppp eeee



13.1.2 DIREKTIIVID

A Assembler	P Pass Point
C Call	R Read
D Display	S Set Memory
F Fill Memory	T Trace
G Go	U Untrace
H Hex	V Verify
I Input Line	X Examine
L List	
M Move	

Direktiiv tuleb sisestada klaviatuurilt süsteemiviiba «#» järel. Iga direktiiv koosneb tähtnemoonikust ja suvandparameetrist sümbolavaldisest näol. Direktiivi pikkus ei tohi ületada 64 märki. Direktiivi lõputunnuseks on RETURN. Eraldajana võib kasutada koma või tühikut. Silurist väljumiseks võib kasutada CTRL C või direktiivi G0.

13.1.3 ARVUD

Vaikimisi interpreteerib SID kõiki arve kui kuueteistkümnendsüsteemi arve vahemikus 0H . . OFFFFH. Arvudest, mis sisaldavad rohkem kui 4 numbrit, aktsepteeritakse neli viimast numbrit.

Näide:

303FFF3FF3 aktsepteeritakse 3FF3

13.1.4 KÜMNENDARVUD

Kümnendarvude tunnus on numbriosund (#) arvu ees. Arvudest, mis on suuremad kui 65535, aktsepteeritakse 16 parempoolseimat bitti.

Näiteid:

#48 #9999 #65535 #0 aktsepteeritakse täielikult,
#65539 aktsepteeritakse 4(65539-65535)

13.1.5 MÄRGID

SID aktsepteerib KOI-8 märke apostroofide vahele paigutatuna. Stringides, mis sisaldavad üle kahe märgi, tunnistatakse kehtivaiks kaks parempoolset. Parempoolne märk salvestatakse 16-bitise sõna madalamasse baidi. Ühe märgi puhul on kõrgem bait 00. Nullise pikkusega stringid pole lubatud. Stringi elemendiks olev apostroof kirjutatakse kahekordselt ('...').

Näiteid:

'a' 'A' 'xy' '#' 'siin' " on'

13.1.6 MÄRGENDIVIITED

Kui on olemas märgendite tabel, võivad siluri sümbolavaldised sisaldada viiteid märgendeile:

- (1) **.s** — märgendile s vastav aadress
 (2) **s** — 16-bitine sõna aadressil .s
 (3) **=s** — bait aadressil .s
 s on märgenditabeli element, tüübilt märgistring

13.1.7 SÜMBOLAVALDISED

Sümbolavaldis kujutab endast omavahel operaatoritega «+» ja «-» seotud kombinatsiooni kuueteistkümnend- ja kümnendarvudest, stringidest ning märgendiviitidest. Komponentide väärtused liidetakse või lahutatakse vastavalt operaatorile, ilma ületäitumise kontrollita ning tulemuseks saadakse 16-bitine sõna.

Süntaks:

- x = 0 - x
 + x = x' + x, kus x' on plussile eelneva avaldise väärtus
 ^^ .^ = n-s sõna pinumälus; märk «^» kirjutatakse n korda järjest

NB! Avaldise sees pole lubatud kasutada tühikuid.

Sümbolavaldisete kasutamisenäiteid on toodud direktiive kirjeldavates jaotistes.

13.2 DIREKTIIVIDE KIRJELDUSED

13.2.1 A ASSEMBLE — REAASSEMBLER

- (1) **As** Käivitab reatransleerimisrežiimi aadressist s. Iga järgmine võimalik aadress uue assemblerkäsu sisestamiseks tuuakse automaatselt ekraanile pärast viimase käsu sisestamist. Režiimist väljumiseks tuleb sisestada kas tühi rida (ainult RETURN) või «.«.

- (2) **A** Nagu (1), kuid stardiaadressiks võetakse viimase direktiivi Assemble, List või Trace poolt töödeldud aadress.
- (3) **-A** Eemaldab silurist assembler- ja pöördassembler-mooduli ning märgendite tabeli, ühtlasi desaktiveerib rektiivid Assemble ja List. Sel juhul väljastatakse ka Trace puhul ainult masinakood 16-ndkujul.

Näiteid:

A100
A # 256
A.CRLF+5
AGAMMA+X- =1
A+30

13.2.2 C (CALL) — ALAMPROGRAMMI KUTSE

- (1) **Cs** Kutse silurist aadressile s, kusjuures testitava programmi registre olek säilib. Alamprogrammi sisemisel BC=0000 ja DE=0000.
- (2) **Cs,b** Nagu (1), BC=b, DE=0000, kus b on avaldis.
- (3) **Cs,b,d** Nagu (1), BC=b, DE=d, kus b ja d on avaldised.

Näiteid:

C100
C # 4096
C.DISPLAY
CJMPVEC+ =X
C.CRLF, #34
C.CRLF, X, +=X

13.2.3 D (DISPLAY MEMORY) — MÄLUTÖMMIS

Mälusisu väljastatakse baitidena, kui aga direktiivis on «W», siis 16-bitiste sõnadena. Iga rea lõpus näidatakse vastavad KOI-8 märgid.

- (1) **Ds DWs** Alates aadressist s, poole ekraani täitumiseni
- (2) **Ds,t DWs,f** Aadressidelt s..f
- (3) **D DW** Alates viimasena väljastatud baidi järelt või registripaaris HL sisalduvast aadressist pärast registre väljastust (vt. direktiiv X), poole ekraani täitumiseni
- (4) **D,f DW,f** Nagu (3), kuid aadressini f

Näiteid:

DF3F
D # 256, # 512
D.gamma,.DELTA+ #30
D.GAMMA
DW ALPHA,+ #100

13.2.4 F (FILL MEMORY) — MÄLU TÄITMINE

F s,f,d Täidab mälu aadressidel s..f väärtusega d (1 bait).

Näited:

F100,3FF,ff
Fgamma,+ #100, # 23
F ALPHA,+ =I,=X

13.2.5 G (GO TO PROGRAM) — PROGRAMMI KÄIVITUS

- (1) **G** — G
- (2) **Gp** — Gp
- (3) **G,a** — G,a
- (4) **Gp,a** — Gp,a
- (5) **G,a,b** — G,a,b
- (6) **Gp,a,b** — Gp,a,b

Silutavat programmi täidetakse reaajas; juhtimine antakse programmist silurisse ainult katkestus- või kontrollpunktides, kui need on defineeritud (vt. direktiiv P), või RST 7 puhul.

Parameetrid:

- p** — käivitusaadress, laaditakse enne programmi käivitust käsu loendurisse PC; kui direktiiv ei sisalda seda parameetrit, käivitatakse programm tema olekuvektoris sisalduva PC väärtusega
- a, b** — katkestuspunktide määratlused; kui määratluseks on märk «^», antakse juhtimine pinumälu kõige pealmises 16-bitises sõnas sisalduvale aadressile, niisiis on seda mugav kasutada katkestamiseks alamprogrammi naasmisaadressil
- miinus** — direktiiviga P defineeritud kontrollpunktide esitus blokeeritakse kuni kontrollpunktide loenduri jõudmiseni nullile

Teated:

Katkestuspunktini jõudmisel või välise RST 7 korral väljastatakse katkestuspunkti aadress kujul

* nnnn

ja katkestuspunkt tühistatakse.

Näiteid:

G,
G100
G100,103
GCRLF,PRINT,#1024
GJMPVEC+=I,.ENDC,.ERRC
G,ørrsub
G,ERRSUB,+30
—G100,+10,+10

13.2.6 H (HEXADECIMAL VALUES) — KUUETEISTKÜMNENDARVUTUSED

- (1) **Ha,b** — väljastab 16-ndkujul summa (a+b) ja vahe (a-b)
- (2) **Ha** — väljastab a väärtused 16-nd- ja kümnendkujul ning
- (3) **H** — KOI-märgina «c»; järgneb märgend, kui see on olemas:

hhhh #dddd 'c' .ssss

Näiteid:

H100,200
H41000,4965
H.GAMMA +=I,ALPHA - 410
H453
H X +=Y - 5

13.2.7 I (INPUT LINE) — LOS-i DIREKTIIVI SISESTUS

lc1c2 .. cn,

kus c1..cn on KOI-märgid, mis LOS-i käsuras järgneksid silutava programmi nimele.

Initsialiseerib vastavad mälutsoonid direktiivi R või silutava programmi jaoks, nagu oleksid c1..cn sisse loetud LOS-i kaudu. Initsialiseeritakse vastav FJP, sisend-väljundpuhver defineeritakse standardsele kohale (BOOT+80H).

Näiteid:

ix dat
ix.inp y.out
ia.x.inp b:y.out \$ - p
ITEST.COM
ITEST.OBJ TEST.SYM

13.2.8 L (LIST CODE) — PÖÖRDASSAMBLER

Masinakood pöördtransleeritakse, ekraanile väljastatakse käsu aadress 16-ndkujul ja käsu mnemokood. Kui direktiiv algab miinusmärgiga, siis aadresse ja märgendeid ei väljastata.

- (1) Ls — Ls Alates aadressist s, poole ekraani täitumiseni
 (2) Ls,f — Ls,f Aadressidelt s..f; väljastuse katkestab suvaline klavh
 (3) L — L Alates viimasest direktiivist L,A või T poolt töödeldud aadressist, poole ekraani täitumiseni

KP580MK80 käsustikku mitte kuuluvad koodid väljastatakse kujul:

??=hh,

kus hh on baidi väärtus kuueteistkümnendkujul.

Näiteid:

L100
 L #1024, #1034
 L.CRLF
 LICALL,+30
 — L.PRBUFF+=l,+ 'A'

13.2.9 M (MOVE MEMORY) — MÄLUSISU TEISALDUS

M s,h,d Aadressidel s..h asuvad andmed paigutatakse aadressiga d algavasse mälutsooni. Lähte- ja sihtsoonid mälus võivad kattuda.

Näiteid:

M100,1FF,300
 M.X,.Y,.Z
 M.GAMMA,+FF,.DELTA
 Malpha+=X,+ #50,+100

13.2.10 P (PASS COUNTER) — KONTROLLPUNKTIDE SEADMINE

- (1) Pp Kontrollpunkt seatakse aadressile p; punktiloendur = 1
 (2) Pp,c Nagu (1), punktiloendur = c
 (3) P Ekraanile väljastatakse kõigi kontrollpunktide aadressid ja vastavate punktiloendurite väärtused
 (4) — Pp Kontrollpunkti kustutus aadressilt p
 (5) — P Kõigi kontrollpunktide kustutus

Kontrollpunkt on käsuloenduri PC väärtus, mille läbimisel kontrollitakse CPU registrite sisu ja soovi korral väljastatakse ekraanile. Iga kontrollpunkti juurde kuulub loendur, mille väärtus võib olla 0..FF (0.. #255) ja mida vähendatakse ühe võrra igal kontrollpunkti jõudmisel. Kui loendur saab väärtuseks 1, muutub kontrollpunkt automaatselt püsivaks katkestuspunktiks. Loendur säilitab väärtuse 1. Erinevalt ajutisest katkestuspunktist (vt. direktiiv G), katkestab kontrollpunkt programmi täitmise pärast märgitud aadressil asuva masinakäsu täitmist.

Korruga võib esineda kuni 8 kontrollpunkti. Igal kontrollpunkti jõudmisel väljastatakse järgmine tekst:

nn PASS hhhh .ssss,

kus

nn — punkti loenduri väärtus
 hhhh — aadress
 ssss — märgend, kui on olemas

Direktiivid —G ja —U blokeerivad registrite sisu väljastuse, kuni loendur saab väärtuseks 1. Programmi täitmist võib katkestada vajutamisega suvalisele klavhile.

Näiteid:

P100,ff
 P.BDOS
 PICALL+30, #20
 —P.CRLF

13.2.11 R (Read CODE/SYMBOLS) — LAADIMINE

Enne laadimist peab direktiiviga I sisestama laaditava(te) faili(de) nime(d). Sõltuvalt selle direktiivi parameetritest laaditakse programm ja/või märgenditabel.

Laadimistsoonide baasaadress on programmil: 100H, märgenditabelil: vaba mälutsooni lõpp

- (1) R Laadimine alates baasaadressist
 (2) Rd Laadimine nihkega d, s.t. algusaadress = baas+d

Kasutamisi variante:

Ix.y Masinakoodifail x.y laaditakse (normaaljuhul y=COM) aadressile 100H; kui y OBJ, peab fail olema LOS-i objekt kujul

R
 Ix.y u.v Lisaks koodifailile laaditakse ka märgenditabel (normaaljuhul v=SYM):

R
 I* u.v Mälu algusosa ei muutu, laaditakse ainult märgenditabel.

R

Märgenditabeli laadimisel väljastatakse ekraanile teade

Symbols

Veateated:

«?» enne teadet «Symbols» — viga masinakoodi laadimisel
 «?» peale teadet «Symbols» — viga märgenditabeli laadimisel

Näiteid:

ISORT.OBJ SORT.SYM

R

Imest.com mest.sym

R — *256

13.2.12 S (SET MEMORY) — KIRJUTUS MÄLLU

- (1) Ss Baitide sisestus alates aadressist s
 (2) SWs 16-bitiste sõnade sisestus alates aadressist s

Mõlemal juhul väljastatakse ekraanile aadress ja pesa(de) sisu. Tühja rea sisestamisel (ainult RETURN) jääb pesa(de) sisu muutmata ja aadressi suurendatakse salvestusüksuse (1 või 2 baidi) või selle kordse (vastavalt sisestatud baitide hulga) võrra. Uue teksti sisestamisel muudetakse mälu sisu ja aadressi suurendatakse ülalkirjeldatud viisil. Sisestamise lõpetab RETURN. Stringe sisestatakse vorminguga (1), kusjuures stringi tähiseks on jutumärgid («») stringi alguses ja RETURN stringi lõpus.

Järgnevates näidetes on kasutaja poolt sisestatud märgid tähistatud alljoonega.

Näiteid:

S100

0100 C3 34

0101 24 #254

0102 CF

0103 4B «ASCII»

0108 6E =X+5

0109 D4 .

SW.X+ #30

2300 006D 44F

2302 4F32 GAMMA

2304 33E2

2306 FFII 0+.X+=I- #20

2308 348F

13.2.13 T (TRACE MODE) — JÄLITUSREŽIIM

Enne iga jälitatava käsu täitmist väljastatakse ekraanile registrite sisu ja pöördtransleeritud käsk. Kui direktiivi ees on miinusmärk, siis operande ja märgendeid ei pöördtransleerita ja programmi täitmine kulgeb kiiremini.

Jälitusrežiimist väljumiseks tuleb vajutada suvalisele klahvile.

- (1) Tn –Tn Täidetakse n masinakäsku
 (2) T –T Täidetakse üks masinakäsk
 (3) TWn –TWn n käsku, alamprogrammide jälituseta
 (4) TW –TW 1 käsk, alamprogrammide jälituseta

Näiteid:

T100
 –TW #10

13.2.14 U (UNTRACE MODE) — SAMMUREŽIIM

U ..

Täidab samu funktsioone, mis T, registrite sisu ekraanile ei väljastata
 Sammurežiimist väljumiseks tuleb vajutada suvalisele klahvile.

Näiteid:

U140
 UW #10

13.2.15 V (VERIFY) — MÄLUTSOONIDE VÖRDLEMINE

Vs,h,d Aadressidel s..h asuvate pesade sisu võrreldakse vastavate, aadressilt d algavate pesade sisuga. Ekraanile väljastatakse kõik mitteidentsed baidid ja nende aadressid

Näited:

V100,2+F,200 — võrdleb omavahel baite aadressidega 100,200; 101,201;...;10F,20F
 V100,1FF,101 — võrdleb piirkonnas 100..1FFF omavahel kõiki kõrvuti asetsevaid baite

13.2.16 X (EXAMINE CPU STATE) — PROGRAMMI OLEKUVЕКТОР

Direktiiv võimaldab kontrollida ja muuta registrite sisu ja lippude väärtusi.
 Lippude tähised on selles direktiivis järgmised:

C — ülekanne M — negatiivsus I — dekaadülekanne
 Z — null E — paarsus

(1) X Registrite sisu väljastatakse järgmisel kujul:

f A=a B=b D=d H=h S=s P=p i s, kus

f — lippude olekud:
 lipu tähis: väärtus=1, miinusmärk: väärtus=0
 a — akumulaatori sisu
 b — registripaari BC sisu
 d — registripaari DE sisu
 h — registripaari HL sisu
 s — pinuviida SP väärtus
 p — käsuloenduri PC väärtus
 i — viimati defineeritud masinakäsk
 s — märgend, kui on olemas; kaudaadresseerimisega käskude puhul (näit. INRM,ADDM) on s vastava mälupeesa sisu **enne** käsu täitmist

(2) Xf Lipu f (=C,Z,M,E või I) väärtuse esitus; väärtuse muutmiseks sisetada 0 või 1

(3) Xr Registri r (=A,B,D,H,S või P) sisu muutmine

Näiteid (alljoon tähistab kasutaja sisestatud andmeid):

XM
 M 0
 XB
 3E04 3EFFF
 XP
 446E .CRLF+10

14 TEKSTIREDAKTOR EDIT

Tekstiredaktor EDIT on ette nähtud kasutamiseks magnetofoniga komplekteeritud mikroarvutil operatsioonisüsteemiga LOS. Kutse:

EDIT <failinimi>

Failinime puudumisel väljastatakse veateade 1 (vt.14.5.2).

Kui sisestatud failinime ei leita kataloogist, väljastatakse teade «NEW» ja avatakse uus fail.

Peale faili leidmist või uue avamist väljastab redaktor oma viiba (numbriosundi#) ja jääb ootama direktiive.

Teksti sisestamine ja redigeerimine toimub rida-realt. Töödeldava teksti ja ühe rea maksimaalsed pikkused on vastavalt 32512 ja 80 märki. Et redaktorit saaks kasutada ka arvuteil, mille kuva laius on väiksem kui 40 positsiooni, on EDIT varustatud lisavõimalusega teksti selle osa ilmutamiseks, mis muidu jääks kuvari ekraanilt välja.

See osa tekstist, mis mahub ekraanile korraga, moodustab nn. akna, mida saab klahvidega **CTRL A** ja **CTRL F (f5 ja f6)** nihutada vasakule ja paremale. Akna asendit kirjeldab ekraani ülaosas olev järgmise kujuga rida:

```
:pl ..... pr:
```

siin

pl — vasakpoolseim positsioon aknas

pr — parempoolseim positsioon aknas

pl, pr — kümnendarvud

Kui teksti väljastusel rida ei mahu täielikult aknasse, tähistab seda ekraani viimases positsioonis märk «+» ja kõik märgid sellest paremal jäävad ilmutamata. Kui teksti sisestamisel on kursor jõudnud ekraani äärmise parempoolse positsioonini, ja sisestatav märk ei ole RETURN, väljastatakse märk «+», kursor läheb järgmise rea algusse ja rea sisestamine jätkub tavalises korras.

14.1 TEKSTI SISESTAMINE KLAHVIGA

Tekst sisestatakse ridahaaval. Sisestatava rea korrigeerimiseks on kasutada järgmised talitlusklahvid:

<-	— viimase märgi kustutus
DEL	— rea kustutus
TAB	— tabulaator
RETURN	— rea lõpp (reavahetuscode salvestatakse)

14.1.1 MÄRGI KUSTUTAMINE

Iga sisestatav rida kantakse sisendpuhvrise. Sisestatud informatsiooni analüüsi alustatakse alles pärast rea täielikku sisestamist (rea sisestamise lõpetab RETURN). Rea sisestamise käigus on igal hetkel võimalik talitlusklahvi <- abil kustutada puhvris viimasel kohal asuv märk.

Näiteid:

ABCDEF<-EF	annab	ABCDEEF
abcdfg<-<-	annab	abcd

14.1.2 REA KUSTUTAMINE

Rea sisestamise käigus on võimalik igal hetkel (enne rea lõpetamist RETURN-iga) tühistada kogu sisestatud rida klahviga **DEL**. Selle tulemuseks kustutatakse sisendpuhvri sisu ja ka vastav rida kuvari ekraanil. EDIT jääb ootama uue rea sisestamist.

14.1.3 TABULEERIMINE

Klahvi **TAB** kasutatakse kursori positsioneerimiseks lähimale paremal asetsevatele tabulaatoriveerule. EDIT ei kannu puhvrise tabulaatori KOI-koodi 09H, vaid vastava hulga tühikuid (KOI-koodid 20H). Redaktori käivitamisel võetakse tabulaatoriveergudeks vaikimisi järgmised kursori positsioonid:

9,17,25,33,41,49,57,65

EDIT võimaldab neid väärtusi ümber defineerida (vt. direktiiv Tabpos,14.4.1).

14.1.4 KOMBINEERITUD SISESTAMINE

Peale ülalnimetatud talitlusklahvide on redaktori kasutaja käsutuses veel terve rida juhtkoode, mis võimaldavad mis tahes rea formeerimiseks lülitada sellesse elemente viimati sisestatud reast. Juhtkoodi sisestamiseks vajutada korraga klahvile CTRL ja vastavale märgiklahvile.

CTRL L	— kopeerida rea lõpuni
CTRL E	— asendada märk
CTRL C	— kopeerida märk
CTRL S	— jätta märk vahele
CTRL O x	— kopeerida märgini x
CTRL P x	— jätta vahele märgini x
CTRL V	— kopeerida üks sõna
CTRL T	— jätta üks sõna vahele
CTRL W	— kustutada üks sõna
x	— sisestada märk x

Märgijada üheks otspunktiks on alati kursori tegelik asukoht viimati sisestatud reas. Seejuures märkide sisestamisel uude ritta liigub viimati sisestatud rea kursor paralleelselt uue rea kursoriga.

Näide:

EDIT töötab lisamisrežiimis (direktiiv A,14.3.10) ja viimane sisestatud rida on:

```
CRLF:  PUSH   PSW
```

Olgu järgmisena sisestatav rida

```
SAVE:  PUSH   PSW      ;säilitada registrid
```

Selleks on vaja sooritada järgmised operatsioonid:

- sisestada klaviatuurilt uus märgend SAVE;
- sisestada CTRL O W (kopeerib realõigu märgist «:» kuni märgini «W», viimane välja arvatud);
- sisestada märk «W»;
- sisestada kommentaar.

14.2 TEKSTI ADRESSEERIMINE

EDIT võimaldab adresseerida redigeeritava teksti iga rida või lõiku.

14.2.1 REA AADDRESS

Teksti adresseerimise põhiüksus on rida. Kõik read on nummerdatud, esimese rea number on 1.

Rea adresseerimiseks on kuus erinevat võimalust:

- numbr järgi (kümnenärv)
- märgendi järgi
- märgijada järgi
- reavilda järgi (.)
- viimase rea otseadresseerimine (\$)
- eelnimetatud tunnustest koostatud avaldise järgi

Rea number. Iga rida saab adresseerida tema numbr järgi. Selleks on vaja sisestada nõutava rea number kümnenärvil. Tuleb meeles pidada, et redigeerimise käigus võivad ridade numbrid muutuda.

Märgend. Kui programmiteksti rida algab märgendiga, võib seda rida adresseerida märgendi kaudu. Selleks on vaja sisestada märgend kolonite vahel.

Näide 1. Olgu antud järgmine tekst:

```
(1) ;          START OF ASSEMBLY PROGRAM
(2) ;
(3) START:  PUSH   PSW      ;SAVE REGISTERS
(4)         PUSH   H
(5) LOOP:   MVI    M,0      ;RESET BUFFER
(6)         INX    H
(7)         DCR    A
(8)         JNZ   LOOP
(9)         POP    H        ;UNSAVE REGISTERS
(10)        POP    PSW
(11) ;
(12) ;          END OF INITIALIZATION
```


Selle tekstis oleks korrektne esitada aadressid:

:START: — rida nr. 3;
:LOOP: — rida nr. 5.

Märgijada. Rida võib adresseerida selles reas esineva märgijada kaudu. Jada tuleb asetada jutumärkide vahele.

Näide 2: Kasutame teksti näitest 1. Korrektsed oleksid järgmiselt esitatud aadressid:

«START OF» — rida nr. 1;
«PSW» — rida nr. 3;
«DCR» — rida nr. 7.

Reaviit. Märgiga «.» saab adresseerida rida, millega sooritati viimane operatsioon. Reaviida väärtused pärast eridirektiivide täitmist on toodud lisas 14.5.3.

Näide 3. Oletame, et viimase toiminguna väljastati kuvarile tekst näitest 1. Reaviida väärtuseks on nüüd viimase väljastatud rea number.

. — rida nr.12

Viimane rida. Teksti viimast rida võimaldab adresseerida märk «\$».

Näide 4. Näitest 1 võetud teksti puhul:

\$ — rida nr.12.

Kombinatsioon. EDIT lubab esitada rea aadressi aritmeetilise avaldise, kasutades lisaks ülal loetletud aadressmärkidele veel kümnendarve ning tehetena liitmist ja lahutamist.

Näide 5. Kasutame jälle teksti näitest 1. Korrektsed oleksid järgmised aadressid:

:START:+1 — rida nr.4;
«LOOP:»-1 — rida nr.4;
\$-5 — rida nr.7.

14.2.2 TEKSTILÕIGU ADRESSEERIMINE

Tekstilõik koosneb ühest või mitmest üksteisele järgnevast tekstireast. Ta on määratud üldjuhul lõigu esimese ja viimase rea aadressidega (mida eraldab koma).

Erijuhte:

- * Kui lõik sisaldab ainult ühe rea, piisab ainult ühest aadressist.
- * Kui lõigu lõpuaadress on väiksem lõigu alguse aadressist, sisaldab lõik ainult ühe rea.
- * Kui lõigu algusaadress puudub, võetakse vaikimisi intervall lõigu alguseks teksti esimene rida.
- * Kui lõigu lõpuaadress puudub, võetakse lõpureaks vaikimisi teksti viimane rida.

Näide. Olgu puhvis järgmine tekst:

```
(1) ; ROUTINE TO PRINT CARRIAGE RETURN
(2) ; LINE FEED
(3) ; ENTRY CONDITIONS: NONE
(4) ; EXIT CONDITIONS: ALL REGISTERS UNCHANGED
(5) ;
(6) CRLF: PUSH PSW ;SAVE REGISTERS
(7) MVI A,0DH ;PRINT CARRIAGE RETURN
(8) CALL TTY0
(9) MVI A,0AH ;PRINT LINE FEED
(10) CALL TTY0
(11) POP PSW ;UNSAVE REGISTERS
(12) RET
(13) ;
(14) ; END OF CRLF-ROUTINE
```

Korrektsed oleksid järgmiselt esitatud lõigud:

CRLF:+3 — rida nr.9
«A.OPH»,10 — read nr.7..10
«EXIT» — read nr.1..4
6, — read nr.6..14
«RET»,5 — rida nr.12

14.2.3 OTSINGUTSOONI ETTEANDMINE

Adresseerimisel märgijadaga või märgendiga võib veel lisaks määrata tsooni, mille sees antud jada või märgendit otsitakse. Selleks on vaja otsitavale jedale lisada sulgudes rea aadress või tekstilõigu adresseerimisandmed.

Lisa-aadress määrab ära rea, kust alates hakatakse antud stringi või märgendit otsima, tekstilõik aga määrab ära tsooni, mille ulatuses toimub otsimine.

Selline tsooni lokaliseerimine on oluline, kui otsitav jada esineb tekstis mitu korda.

Näide. Kasutame teksti eelmisest näitest. Korrektselt on esitatud järgmised otsinguandmed:

«REGISTERS»	— rida nr. 4;
«REGISTERS»<:CRLF:+1>	— rida nr. 11;
«PRINT»	— rida nr. 1;
«PRINT»<2>	— rida nr. 7;
«PRINT»<«TTY0»,10>	— rida nr. 9.

14.3 REDIGEERIMISOPERATSIOONID

Iga kord, kui redaktor väljastab kuvarile teksti:

*EDIT COMMAND:

teatab ta sellega, et asub režiimis, kus ta ootab mõne allpool kirjeldatud direktiivi sisestamist.

Direktiivi üldkuju on järgmine:

direktiivinimi parameetrid RETURN

Tühikud võib ära jätta. Direktiivide suvandosad on kirjeldustes nurksulgudes, parameetrijada tähistavad ümarsulud.

Küsimärk («?») direktiivina väljastab redaktori kõigi direktiivide loendi. Enne redigeerimise algust tuleb defineerida sisend- ja väljundfail.

14.3.1 R — TEKSTI SISESTUS FAILIST

Süntaks: R[ead] [*] [n]

Parameetrid: n — sisestavate ridade arv, kümnendkujul
* — sisestus puhvrite täitumiseni

Teated: *BUFFER IS 3/4 FULL* («3/4 puhvril on täis»)
BUFFER IS FULL («Puhver on täis»)

Katkestamine: klahviga ESC; muud klahvid ei mõjuta käimasolevat sisestust

14.3.2 W — TEKSTI VÄLJASTUS FAILI

Süntaks: W[rite] [N]

Parameetrid: N — puhvri kustutuse keeld

Parameetri puudumisel puhver kustutatakse, kuid kirjutusvea korral puhvri sisu säilib.

Katkestamine: klahv ESC; muud klahvid ei mõjuta käimasolevat väljastust.

14.3.3 C — KOPEERIMINE VÄLJUNDFAILI

Süntaks: C[opy]

Väljundfaili kopeeritakse redigeeritud tekst ja see osa sisendfailist, mis puhvriss ei mahtunud.

Katkestamine: klahv ESC; muud klahvid ei mõjuta käimasolevat kopeerimist.

14.3.4 L — TEKSTI ESITUS

Tekst väljastatakse puhvril ekraanile või printerile (vt. direktiiv .A L). Ekraanile väljastatakse ainult aknasse mahtuv osa.

Süntaks: L[ist] [*] [i]

- Parameetrid:** i — tekstilõigu adresseerimisandmed;
i puudumisel esitatakse kogu puhvri sisu
* — ridade valimine esituseks märgijada alusel.
- Valikesitus:** Kui direktiivis on «*», nõuab redaktor märgijada etteandmist teatega
ENTER TEXT STRING:
Märgijada sisestatakse jutumärkide vahel ning võib sisaldada kuni 20 märki.
- Juhtimine:** Akent saab esituse käigus nihutada klahvidega CTRL A (f5) ja CTRL F (f6).

14.3.5 N — KUVA EDASIKERIMINE

Ekraanile väljastatakse aknasse mahtuv osa. Seda saab nihutada klahvidega CTRL A (f5) ja CTRL F (f6).

Süntaks: N[ext] [n]

- Parameetrid:** n — rea number kümnendkujul, alates realoenduri väärtusest; parameetri puudumisei esitatakse järgmine rida

Realoenduri väärtused vt.14.5.3.

14.3.6 P — KUVA TAGASIKERIMINE

Ekraanile väljastatakse aknasse mahtuv osa. Seda saab nihutada klahvidega CTRL A (f5) ja CTRL F (f6).

Süntaks: P[revious] [n]

- Parameetrid:** n — rea number kümnendkujul, realoenduri väärtusest tagasi lugedes; parameetri puudumisel esitatakse eelmine rida.

Realoenduri väärtused vt.14.5.3.

14.3.7 I — TEKSTI SISESTUS JA LISAMINE

Süntaks: I[nsert] [r]

- Parameetrid:** r — rea aadress; sisestatav tekst paigutatakse selle rea järele.

Ilma parameetrita saab teksti sisestada ainult tühja puhvrissi, vastasel juhul väljastab EDIT veateate.

Teated: * **LINE ADDRESS ERROR** («Viga rea aadressis»)

Väljumine: Teksti sisestamise lõpetab CTRL Z uue rea alguses.

14.3.8 J — TEKSTI PARANDAMINE

Süntaks: J[ustify] [*] i

- Parameetrid:** i — tekstilõigu adresseerimisandmed
* — teksti valimine märgijada alusel
- Otsingujada:** Kui direktiivis on «*», nõuab redaktor märgijada etteandmist, teatega
STRING:
NÜÜd sisestatakse märgijada.
- Töötlus:** — Rea saab asendada uuega (vt.14.1.4)
— Saab kasutada vana rea osi.
- Väljumine:** CTRL Z rea alguses.

14.3.9 D — TEKSTILÕIGU KUSTUTUS

Süntaks: D[ele] i

- Parameetrid:** i — kustutatava tekstilõigu adresseerimisandmed

14.3.10 A — TEKSTI LISAMINE

Uus tekst lisatakse adresseeritud rea järele.

Süntaks: A[ppend] r

- Parameetrid:** r — rea aadress
- Väljumine:** CTRL Z uue rea alguses.

14.3.11 E — TEKSTILÕIGU ASENDAMINE

Sisuliselt on direktiivide D ja I kombinatsioon. Adresseeritud tekstilõigu asemele sisestatakse uus tekst.

Süntaks: E[xchange] I

Parameetrid: i — asendatava tekstilõigu adresseerimisandmed;
Väljumine: CTRL Z uue rea alguses.

14.3.12 M — TEKSTILÕIGU TEISALDUS

Süntaks: M[ove] [*] I>r

Parameetrid: i — teisaldatava lõigu adresseerimisandmed
r — rea aadress; lõik teisaldatakse selle rea ette
* — tekstilõiku ei kustutata lähteasukohas

14.3.13 S — MÄRGIJADADE ASENDAMINE

Adresseeritud tekstilõigus saab osutatavad märgijadad asendada uutega.

Süntaks: S[ubstitute] I

Parameetrid: i — tekstilõigu adresseerimisandmed
Märgijadade sisestus:
Vastuseks direktiivi sisestusele väljastab EDIT teate

STRINGS:

ja jääb ootama märgijadade paaride sisestust. Jga jada asub jutumärkide vahel, paarid tuleb üksteisest eraldada komadega. Paaris esimesel kohal olev jada asendatakse teisel kohal olevaga.

Näide:

STRINGS: «strng 11» «strng 12», «strng 21» «strng 22»

Tekstilõigu i piires asendatakse jada «string 11» jadaga «string 12» ning jada «string 21» jadaga «string 22».

Väljundandmed: enne naasmist ooterežiimi väljastatakse sooritatud asenduste arv

14.3.14 X — VÄLJUMINE REDAKTORIST

Direktiivi sisestamisel vastuseks redaktori viibale kontrollib EDIT, kas redigeeritud fail on tagasi salvestatud. Kui see on nii, tagastatakse juhtimine opsüsteemi ja ekraanile ilmub opsüsteemi viip «>».

Kui teksti ei ole väljastatud, ilmub veateate 10 (vt.14.5.2) koos teatega

EXIT (Y/N)

mis nõuab kinnitust väljumistaotlusele. Jaatava vastuse (Y) korral redigeeritud teksti ei päästeta, eitava vastuse järel saab teksti salvestada direktiiviga W (vt.14.3.2).

14.4 ABIOPERATSIOONID

14.4.1 T — TABULAATORI SEADMINE

Redaktori käivitamisel defineeritakse vaikimisi tabulaatori positsioonideks järgmised veerud:

11,19,27,35,43,51,69,77

Direktiiviga T saab neid väärtusi ümber defineerida, muuta nii positsioonide väärtusi kui ka nende arvu.

Süntaks: T[abpos] [= (n)]

Parameetrid: (n) — tabuleerimispositsioonide väärtused kümnenndkujul
Parameetri puudumisel esitab direktiiv kasutusel olevad väärtused teatega

TAB STOP = N1,N2,N3,...

Näide:

T=11, 18, 37

14.4.2 — VÄLJASTUSE DEFINEERIMINE

(1) .[ASSIGN] L — väljundseadme kinnistamine, väljastusvormi valimine

Vastuseks direktiivi sisestusele väljastab redaktor teate

LIST DEVICE = («Nimetada seade»)

ja jääb ootama soovitud väljundseadme nime.

Võimalikud on järgmised seadmed:

TTY — kuvar

LP — printer

Järgneb küsimus

LINE NUMBERS ON (Y/N)? («Kas väljastada reanumbrid?»)

Jaatava vastuse (Y) korral väljastatakse edaspidi iga rida koos tema numbriga, eitava vastuse (N) korral aga ilma.

Pärast seda pöördub EDIT tagasi direktiivi ootamise režiimi.

(2) .[ASSIGN] U — tähestikurežiimi ümberlülitamine: hakatakse eristama suur- ja väiketähti või loobutakse sellest

14.4.3 K — PUHVRI KUSTUTUS

Süntaks: K[ill]

Direktiiv K kustutab puhvrist kogu redigeeritava teksti, järgneb naasmine direktiivi ootamise režiimi.

14.5 LISAD

14.5.1 TEKSTIREDAKTORI DIREKTIIVID

Append r	Lisamine
Copy	Kopeerimine faili
Delete i	Kustutus
Exchange i	Asendamine
Insert [r]	Sisestus
Justify [*] i	Parandamine
Kill	Puhvri kustutus
List [*] [i]	Esitus
Move [*] i>r	Teisaldus
Next [n]	Edasikerimine
Subst i	Märgijada asendamine
Previous [n]	Tagasikerimine
Read [*] [n]	Sisestus failist
Tabpos [= (n)]	Tabulaator
.Assign < .Assign U	Välisseade, esitusvorm
Write [N]	Väljastus faili
X	Väljumine redaktorist

r = rea aadress

i = tekstilõigu aresseerimisandmed

n = number

14.5.2 TEKSTIREDAKTORI VEATEATED

Tekstiredaktor väljastab järgmisi veateateid:

1 NO FILE SPECIFIED	— redigeeritav fail määramata
2 DEVICE NOT READY	— seade pole valmis
3 INVALID DEVICE	— tundmatu S/V-seade
4 ARGUMENT ERROR	— viga argumendis
5 BUFFER EMPTY. ILLEGAL !	— puhver on tühi, direktiivi ei saa täita
6 LINE ADDRESS ERROR	— viga rea-aadressis
7 COMMAND ERROR	— viga direktiivis

8 TOO LONG LINE(80)	— reas on üle 80 märgi
9 STRING NOT FOUND	— märgijada ei leitud
10 I/O ERROR. TEXT STILL IN BUFFER	— tõrge S/V-operatsioonis, puhvri sisu säilis

14.5.3 REAVIIDA VÄÄRTUSED

Reaviida (vt.14.2.1.4) väärtust peale mingi direktiivi täitmist näitab järgmine tabel

Direktiiv	Reaviida väärtus
Append	Viimase sisestatud rea aadress
Copy	Viimase kopeeritud rea aadress
Delete	Viimasele kustutatud reale järgneva rea aadress
Exchange	Viimase asendatud rea aadress
Insert	Viimase sisestatud rea aadress
Justify	Viimase korrigeeritud rea aadress
List	Viimase väljastatud rea aadress
Move	Esimese ümberpaigutatud rea aadress
Move	Viimase ümberpaigutatud rea aadress
Next	Väljastatud rea aadress
Substitute	Rea aadress, milles viimati asendati märgijada
Previous	Väljastatud rea aadress
Read	Viimase väljastatud rea aadress
Write	Viimase väljastatud rea aadress

15 TESTPROGRAMMID

15.1 PAKETT «DIAGNOSTICS»

«Diagnostics» on pakett mikroprotsessorit 8080,Z80 või KP580IK80 sisal-
dava mikroarvutisüsteemi kompleksseks testimiseks. «Diagnostics» või
maldab eriettevalmistuseta arvutikasutajal teha kindlaks vigade allika (mä-
lu, mikroprotsessor, kettaseade, kuvar), mille seejärel kõrvaldab mikro-
arvutite riistvara spetsialist. Seejuures tuleb siiski silmas pidada, et testide
kasutamiseks peab arvuti suutma programme mällu laadida ja käivitada.

15.1.1 ÜLDISED KASUTAMISJUHISED

«Diagnostics» koosneb järgmistest programmi- ja tekstifailidest:

MTEST	.COM	5K	Mälu test aadresside vahemikus 1500H—FFFFH
MTEST2	.COM	10K	Mälu test aadresside vahemikus 0100H—14FFH
CPU	.COM	19K	Mikroprotsessori test
TERM	.COM	17K	Kuvari test
QRUN	.COM	5K	Mälu, protsessori ja ketta kiirtest
QDISK	.COM	8K	Kettaseadme kiirtest
HELP	.DOC	16K	Kasutamisujuhend

Testida on soovitatav sellises järjekorras:

1. Mälu.
2. Protsessor.
3. Kettaseade.
4. Terminal.
5. Magnetofon.

Kui mälu ei funktsioneerigi õigesti, ei saa teiste testide tulemuste õigsu-
se üle otsustada. Peale mälu testi tuleb kontrollida protsessori korrasole-
kut. Mikroprotsessoril on käske, mida kasutatakse väga harva. Sellest tule-
nevalt võib osa programme töötada ka vigase protsessoriga.

15.1.2 TESTIANDMETE KOGUMINE VÄLISSALVESTISSE

Välissalvestis asuvasse faili A:DIAG.LOG võivad oma teateid saata järg-
mised testprogrammid:

- MTEST.COM
- MTEST2.COM
- CPU.COM

See on väga mugav pikema testide jada korral (näit. 16K mälu bititest
võtab aega umbes 13 tundi), kui operaatori pidev kohalviibimine ei ole või-
malik. Välisseadmele salvestamine ei välista tulemuste esitamist kuvaril.

Välisseadmeks võib olla nii kettaseade kui ka magnetofon. Failis DIAG.LOG paikneb iga testi teadete ees kaks tühja rida ja märgid «**». Seda faili võib hiljem tekstiredaktoriga täiendada ja seejärel printida.

Testiandmete kogumiseks tuleb varuda välisseadmel piisavalt ruumi teadete failile, muidu võib töö katkeda operatsioonisüsteemi veateatega.

15.1.3 KÄSUREA PARAMEETRITE KASUTAMINE

Testprogrammidele MTEST, MTEST2 ja CPU saab parameetreid ette anda käsureaga.

Terminali test ja kompleksne kiirtest ei vaja ühtegi kutseparameetrit.

15.2 MÄLU TESTIMINE

15.2.1 TESTID JA VÕIMALUSED:

- kiirtest (Quick test),
- kulgbititest (Walking bit test),
- mälu taastuvuse test (Burn in test),
- mälu kiiruse test (Speed test),
- mälurühmade ümberlülitamise võimalus,
- testitava mälupiirkonna esitamine,
- vea avastamisel aadressi, kirjutatud ja loetud väärtuse esitus,
- vigade loetelu esitus bittide kaupa.

Kiirtest

See test kirjutab valitud mälupiirkonda 00H, seejärel loeb tagasi iga baidi väärtuse, veendumaks, et kõik bitid on nullitud. Järgmisena kirjutab test antud mälupiirkonda FFH ja loeb tagasi, veendumaks, et kõik bitid on ühed. Edasi kirjutab test antud piirkonda juhuslikke arve ja kontrollib tagasi lugedes saadud väärtusi. Kiirtest avastab üle 90% kõigist mälu riketest.

Kulgbititest

See test avastab vead, mida kiirtest ei leia, näiteks adresseerimisvead. See test võtab oluliselt rohkem aega — võrdeliselt testitavate baitide arvu ruuduga. Näiteks 16K baidi mälu testimiseks kulub umbes 13 tundi.

Taastuvuse test

See test on mõeldud spetsiaalselt dünaamiliste mälude tarvis. Etteantud mälupiirkonnal täidetakse kindlat kirjutamistsükli 1000H korda. Seejärel kirjutatakse sinna salvestise pöördkoodid ja loetakse. Kõik bitid peavad ümber lülituma.

Mälu kiiruse test

See test kontrollib, kas mälu töötab sama kiiresti kui protsessor (CPU). Test saadab kuvarile konstantse perioodiga kellasignaali. See periood peab olema kooskõlas CPU taktsagedusega:

CPU taktsagedus Kellasignaale minutis

2 MHz	1
4 MHz	2
5 MHz	3

Kui signaalide arv ei vasta CPU taktsagedusele, ei funktsioneerigi mälu korrektselt.

Mälu testimiseks on kaks eraldi programmi: MTEST.COM, mis testib piirkonda 1500H—FFFFH, ja MTEST2.COM, mis testib piirkonda, milles asub MTEST (0100H—14FF)

15.2.2 MÄLU TESTI KÄIVITAMINE

1. Kutse: MTEST või MTEST2, opsüsteemi kaudu.
Ilmub küsimus

LOG TO DISK (Y/N) («Kas koguda ka kettale?»)

Eitava vastuse (N) korral väljastatakse teated ainult kuvarile.

2. Algusaadressi küsib tekst

ENTER START ADDRESS (HEX) («Sisestada algusaadress»)

Address sisestatakse 4-kohalise 16-ndarvuna. Lubatud on ainult märgid 0..9 ja A..F. Aadressiga eksimisel saab sisestuse katkestada mis tahes kee-

latud märgiga, see viib sammu 2 kordamisele. Tekstist väljuda saab klahviga CTRL C. Vaikimisi (klahv RETURN arvu asemel) võetakse algusaadressiks 1500H (MTEST2.COM puhul 100H).

3. Lõpuaadressi küsib tekst

ENTER ENDING ADDRESS (HEX) («Sisestada lõpuaadress»)

Sisestada 4-kohaline 16-ndarv või vajutada klahvile RETURN, mis põhjustab vaikimisi väärtuse valiku. Vaikimisi väärtuseks on kasutajamälu lõpp või (MTEST2 puhul) 1500H.

NB! Testitava piirkonna aadresside õigsust ei kontrollita, seetõttu võib liiga väikese (<0100H) või liiga suure (<fbase) aadressi etteandmine viia süsteemimälu ülekirjutusele või tekitada testi tõrkeid. Aadressi fbase väärtus asub pesades 0005H,0006H.

NB! Kui testitakse mälupiirkonda, mis sisaldab olematuid aadresse, teatab mälu test neist kui vigadest.

4. Ilmub testi tüüpide menüü:

PLEASE SELECT («Palun valida»)

Q — QUICK TEST — kiirtest
 W — WALKING BIT TEST — kulgbititest
 B — BURN TEST — taastuvuse test
 S — SPEED TEST — mälu kiiruse test

Siin tuleb valida ainult üks testidest.

5. Järgmisena algab mälurühma valimise dialoog:

ENTER B FOR BANK SELECT («Rühma vahetuseks sisestada B»)

Vajutamisel klahvile RETURN siirdub dialoog sammule 6.

6. Klahvi B vajutamisel ilmub kuvarile küsimus

POKE WHICH PORT? («Osutada värati»)

Värati tähendab siin värati numbrit või mälu aadressi, mis on konkreetses mälusüsteemis kasutusel mälu rühmade ümberlülitamiseks. Kui sisestatud 16-ndarv on väiksem kui 256, siis eeldatakse, et on tegemist väratiga, vastasel juhul võetakse arv mäluaadressiks.

WHAT VALUE? («Milline väärtus?»)

Siin tuleb 16-ndkujul sisestada juhtkoodi väärtus, mida kasutatakse mälurühmade ümberlülitamiseks.

Mõlemad arvud sõltuvad konkreetsest riistvarast.

7. Viimasena küsitakse testitsükli arvu:

ENTER NUMBER OF ITERATIONS (DEFAULT=1)

(«Sisestada iteratsioonide arv; vaikimisi:1»)

Klahviga RETURN valitakse ühekordne test.

Kui kõik eespool toodud küsimused on vastatud, käivitub valitud mälu test ja töötab soovitud korduste arvu täitumiseni või katkestamiseni mingile klahvile vajutamisega.

15.2.3 MÄLU TESTI KÄSUREA PARAMEETRID

Kõik mälu testi küsimuste vastused saab ette anda testi väljakutse käsu reas. Sõltuvalt sellest, kas mälurühmade ümberlülitamist kasutatakse või mitte, on kaks erinevat väljakutse vormi.

1) Mälurühma ümberlülitamisega

> MTEST L S E T B P V R

L Tulemuste väljastus
 (log) L — faili «DIAG.LOG»

N — ainult kuvarile

S Mälupiirkonna algusaadress 16-ndkujul
 (start) 0 — vaikimisiväärtus

- E Lõppaadress 16-nd kujul
(end) 0 — vaikumisiväärtus
- T Testi tüüp Q — kiirtest (Quick)
(type) W — kulgbititest (Walking bit)
B — taastuvuse test (Burn in)
S — mälu kiiruse test (Speed)
- B Mälurühmade valimine B — toimub
(bank select) N — ei toimu
- P Värati number või mäluaadress 16-ndkujul (port)
- V Mälurühma valimise juhtkood 16-ndkujul (value)
- R Testitsüklite arv 16-ndkujul

Näide: >MTEST L 2000 3000 Q B 23 55 1

2) Kui mälu ümberlülitamist ei kasutata, on käsurida lühem:

>MTEST L S E T N R

N märgib, et ei kasutata mälu ümberlülitamist. Teistel parameetritel on sama tähendus, mis esimeselgi juhul.

Näide: >MTEST C 3000 30FF W N 3

NB! Mälu testi käsurea parameetrid tuleb esitada ülaltoodud järjekorras.

15.3 PROTSESSORI TESTIMINE

15.3.1 TESTIOPERATSIOONID:

- CPU tüübi määramine,
- CPU kiiruse kontrollimine,
- CPU käsustiku kontroll ja vigadest teatamine.

Protseessori test interpreteerib programmi, mis on ette nähtud kõikide antud protseessori üksikute käskude ja paljude mitmest käsust koosnevate jadade testimiseks. Programm kontrollib peale iga käsu täitmist kõikide CPU registrite sisu veendumaks, et ettenähtud registrid muutusid õigesti ja et ainult need registrid muutusid. Sellega avastatakse näiteks sellist laadi CPU vigu, kus A-registrisse salvestamine mõjutab B-registri sisu.

15.3.2 MIKROPROTSESSORI TESTI KÄIVITAMINE

1. Kutse: CPU, opsüsteemi kaudu
2. Testi pealdise alla ilmub rida

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Siin tähistab iga täht ühte lühikest CPU testi. Kui sellist rida ei ilmu või pole ta täielik, pole protsessor korras.

3. Järgneb CPU tüübi kindlaksmääramine. Kuvarile peab ilmuma

CPU IS 8080 (või CPU IS Z80)

4. Kontrollitakse kiirust, väljastades kellasignaali ja teate

BEGIN TIMING TEST («Ajastustesti algus»)

Teatud aja möödudes saadetakse teine kellasignaali koos teatega

END TIMING TEST («Ajastustesti lõpp»)

Ajaintervall kahe kellasignaali vahel on määratud CPU tüübiga:

CPU taktsagedus	Aeg kellasignaali vahel
8080 2 MHz	umbes 2 min
Z80 4 MHz	" 1 min
Z80A 5 MHz	" 40 s

Kui ajaintervall on oluliselt erinev toodud väärtustest, võib juhtuda, et CPU ei ole korras.

5. Järgneb kõigi käskude kontroll
6. Kui CPU test lõppes edukalt, ilmub kuvarile teade:

CPU TESTS OK

Vastasel juhul väljastatakse veateated:

CPU FAILED TEST:	(«CPU ei läbinud testi»)
ERROR COUNT XXXXH	(«Valeloendur»)
INSTRUCTION SEQUENCE WAS XXXXXXH	(«Käsujada oli ..»)
REGISTER X1 CONTAINS XXH	(«Registri . . sisu on . . »)
BUT SHOULD CONTAIN XXH	(«Kuid peaks sisaldama . .»)
REGISTER VALUE BEFORE INSTRUCTION	
SEQUENCE WAS XXH TEST NUMBER XXXXH	(«Registri väärtus enne käsujada oli . .»)

NB! Kui ilmnes, et protsessor ei suutnud täita mingit käsujada õigesti, siis üldjuhul kordub see viga ka järgmisel katsel. Seetõttu, kui soovitakse täpsemat infot vea tekkimise kohta, võib kirjutada vastava lühikese assemblerprogrammi ja käivitada selle siluri (SID) abil.

15.3.3 MIKROPROTSESSORI TESTI KÄSUREA PARAMEETRID

CPU testi väljakutsel saab ära näidata, kas soovitakse veateateid väljastada välissalvestis asuvasse faili või mitte. Kui käsuritta on lisatud parameeter LOG, siis saadetakse teated faili DIAG.LOG:

> CPU LOG

Vastasel juhul saadetakse teated ainult kuvarile.

15.4 KUVARI TESTIMINE

15.4.1 TESTID:

- ekraani kustutamine,
- rea kustutamine,
- kursori liigutamine üles, alla, vasakule ja paremale,
- kursori adresseerimine,
- kursori aadressi lugemine,
- normaalse ja pöördkujutise režiimi vahetamine.

15.4.2 KUVARI TESTIDE KÄIVITAMINE

1. Kutse: TERM, opsüsteemi kaudu. Kuvarile ilmub selgitav tekst.
2. Seejärel käivitub testide jada. Iga testi ees väljastatakse seda testi selgitav teade. Teste ei pea sooritama toodud järjekorras, vaid kasutajal on võimalus täita ainult mõni neist või sooritada testid teises järjestuses. Selleks peab ta tegema valiku enne testi väljastatavast võimaluste loetelust:

Type «B» to bypass, «ESC» to exit or «RET» to test
(«Sisestada vahelejätkuks «B», väljumiseks «ESC», käivituseks «RET»)

3. Kuvari test lõpeb kuvarile ekraani mõõtmetega spiraali joonistamisega.

Terminali test nõuab kasutaja aktiivset osavõttu, seetõttu puudub ka võimalus teadete saatmiseks välisseadmehel asuvasse faili.

15.5 KOMPLEKSNE KIIRTEST

15.5.1 OSATESTID *

- kogu kasutajamälu kiirtest,
- kasutatava ketta kiirtest,
- protsessori test.

Kompleksne kiirtest on ette nähtud avastama põhilist vigade allikat väga lühikese aja jooksul. Täielik kiirtest (s.h. kettaseadme test) võtab aega ainult umbes 4 min. (2 MHz CPU korral).

Kompleksne kiirtest koosneb tegelikult kolmest eri programmist, mis käivitatakse üksteise järel; need on

- QRUN.COM mälu kiirtest;
- QDISK.COM kettaseadme kiirtest (*);
- CPU.COM standardne CPU test (vt.15.3).

Käivitamiseks peavad need kõik kolm programmi asuma ühisel välisseadmehel.

* Kettaseadme teste saab kasutada ainult kettaseadme olemasolul, nad ei ole rakendatavad lindiväljundile.

15.5.2 KOMPLEKSSE KIIRTESTI KÄIVITAMINE

1. Kutse QRUN, opsüsteemi kaudu.
2. Testi pealdise järel ilmub kuvarile teade

Memory test («Mälu test»)

ja käivitub kasutaja mälu kiirtest.

3. Kui mälu test on lõppenud, ilmub teade:

Memory test complete («Mälu test lõpetatud»)

Kui mälu test avastas vigu, teatatakse sellest nii:

LOCATION	DATA WRITTEN	DATA READ
XXXXH	YYH	ZZH
«Aadress»	«Kirjutatud andmed»	«Loetud andmed»

Vigade ilmnmisel on soovitat käivitada spetsiaalne mälu test (vt. 15.2), et saada täielikumat infot.

4. Kui mälu test on lõppenud, käivitatakse kettaseadme kiirtest. Kuvarile ilmuvad mõnesekundiliste vahedega järgmised teated:

Disk test	(«Ketta test»)
Read/write test	(«Lugemise ja kirjutuse test»)
Random seek test	(«Juhupöörduse test»)

5. Kui kettatest on lõppenud, väljastatakse kuvarile teated,

X read/write errors detected
X seek errors detected

6. Kompleksse kiirtesti viimane etapp on standardne CPU test. Nimetatud testi funktsioonid ja väljund on kirjeldatud käesoleva juhendi jaotises 15.3.

15.6 MAGNETOFONI TEST

15.6.1 KONTROLLIOPERATSIOONID:

- magnetofoni salvestusnivoo määramine,
- magnetofoni salvestusrežiimi kontroll,
- kasutatava vormindatud lindi kontroll.

15.6.2 TESTI KÄIVITAMINE

1. Kutse: TTEST, opsüsteemi kaudu.

Kasutaja töö testiga seisneb programmi poolt näidatud operatsioonide täitmises ja seetõttu ei vaja siin lähemat selgitust.

16 KOOLIARVUTITE KOHTVÕRK

16.1 ANDMESIDE ÜLESANDED

Kooliarvutit eristab koduarvutist põhiliselt andmeside vahendite olemasolu. (Muudest lauarvutite liikidest eristab mõlemat lihtsama ja odavama välisseadmestiku kasutamine.) Andmeside võimaldab ühendada omavahel üksikarvutid ja luua klassi piires kohtvõrk.

Sellise võrgu **vajaduse** tingivad järgmised asjaolud.

1. Klassis peab saama kasutada ka selliseid välisseadmeid (printer, ümbriketassalvesti jms.), mis ei saa kuuluda iga õpilasarvuti komplekti juurde ainult hankimisvõimaluste ja majanduslike kaalutluste tõttu. Koolis tuleb pidada normaalseks nende seadmete simultaanset kasutamist.

2. Õppetöö efektiivsuse saavutamiseks peab õpetajal olema võimalus oma laua juurest lahkumata jälgida õpilaste tööd arvutitega ning vajaduse korral operatiivselt sekkuda. Õpetaja arvutilt peab olema võimalus saata andmeid õpilasarvutite mällu, ning teha seda vajaduse järgi simultaanselt, rühmiti (näiteks kontrolltööde puhul) või individuaalse adresseerimisega (näiteks diferentseeritud õpetamise korral).

3. Ka õpilasel peab olema võimalus oma arvuti vahendusel saata teateid õpetajale — teatada lahendamise lõpetamisest, paluda abi jne. Traditsiooniliste moodustega võrreldes tõstab selline kommunikatsioonimoodus

klassi töökultuuri, tugevdab distsipliini ja valmistab õpilasi ette tööks tulevastes kohtvõrkudes töökohtadel.

Andmeside seisukohalt iseloomustavad klassivõrku järgmised **talitlusomadused**.

1. Edastatavate sõnumite maht kõigub laiades piirides. Edastatavateks sõnumiteks on

- programmid (kuni 30K..40K baiti),
- tekstimassiivid: õppematerjal, juhendid (1K bait e. üks kuvari ekraanitäis),
- lühisõnumid: küsimused, vastused (64 baiti e. üks rida kuval).

2. Suhteliselt madal edastuskiirus:

- lühisõnumi edastusaeg halvimal juhul kuni 1s,
- intervall kahe samast arvutist lähtuva järjestikuse lühisõnumi kohalejõudmise vahel kuni 2s.

3. Keskmine abonentarvutite arv. Võrku võib ühendada kuni 31 arvutit.

4. Paindlik adresseerimissüsteem. Sihtjaamade adresseerimise moodused on

- individuaalne,
 - grupiline,
 - globaalne.
5. Lihtne ja ökonoomne rajamine ja kasutamine:
- lihtne laiendatavus,
 - ökonoomne sideliin (harilik paarisjuhe magistraalina),
 - võrgu oleku ja võimalike tõrgete jälgitavus õpetajaarvutilt.

16.2 SIDESÜSTEEMI EHITUS

16.2.1 TEHNILISED ANDMED

Füüsiline sideliin:	2-sooneline bifilaarjuhe
Liini maksimaalpikkus:	50 m
Abonentarvutite maksimaalarv:	31
Füüsiline edastuskiirus:	9600 bit/s
Keskmine efektiivkiirus võrgu maksimaalkoormusel:	170 bit/s
Võrgu struktuur:	magistraalne
Pöördusmeetod:	keskjuhitav lubapöördus
Edastatava kaadri pikkus:	kuni 69 baiti
Edastatava ploki pikkus:	kuni 1K baiti

16.2.2 FÜÜSILINE LIIDESTUS

Aparatuurselt on ühendus sideliiniga realiseeritud jadavärati abil, mis on initsialiseeritud tööks asünkronrežiimis ühe stoppbitiga ja paarituse kontrolliga. Töötlusoperatsioone aktiveerivad saatja ja vastuvõtja katkestused.

Igale abonentarvutile on kinnistatud tema füüsiline individuaaladdress (0..31), s.t. arvuti järjenumbrivõrgus. See address on salvestatud püsimalu, andmesidedraiveri püsiandmete segmendi esimesse pessa.

16.2.3 EDASTUSKAADRID

Andmete ülekandmiseks ühest arvutist teise ning edastusprotsessi koordineerimiseks vahetavad arvutid omavahel järjestikujule viidud informatsiooni struktureeritud massiive — edastuskaadreid. Neid formeerib ja töötleb püsimalus asuv andmesidedraiver.

«Juku» sidesüsteemis kasutatakse kaht kaadrivormingut:

(1)		(2)		
	SYN	2 baiti	SYN	2 baiti
	SA	1 bait	SA	1 bait
	LA	1 bait	LA	1 bait
	JK	1 bait	JK	1 bait
	SP	1 bait	KS	1 bait
	INF	1..64 baiti		
	KS	1 bait		

SYN — sünkroniseerimiskood E4E4H

SA — sihtjaama address (vt.16.2.4)

LA — lähtejaama address, 0..1FH

JK — juhtkood (vt.allpool)

SP — infovälja pikkus baitides

INF — infoväli (edastatavad andmed)

KS — kaadri kontrollsumma

Juhtkoodi JK vorming:

7 6 5 4 3 2 1 0

Sidelipud	F
-----------	---

F — funktsioonikood:

- x x x x x x 0 0 — edastusloa andmine (POLL)
- x x x x x x 0 1 — edastusloa tagastus (EOT)
- K x x x A L 1 0 — andmete saatmine (DATA)
- K x x V x x 1 1 — andmete kviteerimine (ACK)

K — kviteerimisbitt:

DATA-kaadrites vaheldumisi 0 ja 1

ACK-kaadrisse kopeeritakse saadud DATA-kaadrist

A = 1: andmeplokk algas

L = 1: andmeplokk lõppes

V = 1: vastuvõtupuhver on täis

x — bitti ei töödelda

16.2.4 ADRESSEERIMINE

Individaaladresseerimine toimub arvutitele kinnistatud ja nende püsimalus paiknevate aadresside (järjenumbrite) alusel. Dispetšerarvuti individuaaladdress on alati 0, teiste arvutite aadressid on vahemikus 1..31 (1..1FH).

Grupiline adresseerimine võimaldab andmekaadrit korruga väljastada mingile õpilasarvutite suvalise suurusega grupile. Õpilasarvutid saab jaotada 2..6 lõikuvasse või mittelõikuvasse gruppi. Selleks tuleb enne andmeedastust saata erioperatsiooniga (vt.16.2.7) kõigile õpilasarvutitele nende grupikoodid, mis määravad arvutite kuuluvuse gruppidesse. Grupikood on järgmise vorminguga bait:

7	6	5	4	3	2	1	0
x	g	g	g	g	g	g	x

g — kuuluvus gruppidesse

g = 1: arvuti kuulub gruppi, mille number võrdub vastava biti numbriga

x: bitti ei töödelda

Globaalne adresseerimine võimaldab andmekaadrit korruga saata kõigile võrgu arvutitele. Globaalaadressi väärtus on EOH.

Kaadri koostisse kuuluva sihtaadressi SA (vt.16.2.3) variandid võivad seega olla järgmised:

7	6	5	4	3	2	1	0		
0	0	0	0	0	0	0	0	—	dispetšer
0	0	0	a	a	a	a	a	—	individaaladdress
a	a	a	0	0	0	0	0	—	grupi address
1	1	1	0	0	0	0	0	—	globaalaaddress

16.2.5 PÖÖRDUSMEETOD

Sideliini kui ressursi jaotamiseks kasutatakse tsentraliseeritud lubapöördust (ingl. *token access*). Sideliini haldab dispetšerarvutis paiknev eriprotseduur — võrgu dispetšer.

Arvuti, mis saab dispetšerilt pöördusloa (kaadri POLL näol), omandab teatud ajaks õiguse kasutada sideliini. Peale sideseansi lõppu tagastatakse luba dispetšerile kaadriga EOT ning siirdub sealt järgmisele abonentarvutile. Edastusvajaduse puudumisel tagastab abonent loa kohe. Seansi pikust kontrollib valvetaimer; kui luba ei tagastata kontrollajaks, tõlgendab dispetšer seda veana.

Loa andmiseks valib dispetšer abonente võrgu konfiguratsiooni tabeli alusel. Tabeli rea indeksiks on abonendi individuaaladdress ja rea moodustab bait järgmise vorminguga:

7	6	5	4	3	2	1	0
Vigade loendur						a	p

a — abonendi olemasolu sellel aadressil

a = 1: abonent on olemas

p — pöördumisloa valdaja viit

p = 1: luba on sellele reale vastaval abonendil

16.2.6 SIDESEANSS

Liinile saatmiseks võtab sidedraiver programmilt vastu andmeploki maali-
maalpikkusega 1K bait ning jaotab selle eri kaadrite ja seansside vahel.
Ühe sideseansi kestel saab üle kanda kuni 10 kaadrit. Kui plokk on pikem,
jätkub edastus järgmisel dispetšeri poolt eraldatud seansil.

Individaaladresseerimise korral kviteeritakse kõiki vigadeta edastatud
kaadreid DATA kaadriga ACK. Järgmine DATA-kaader väljastatakse ainult
siis, kui on täidetud eelmise normaalset vastuvõttu tõendavad tingimused:

- kaadri ACK saabumine valvetaimeri kontrollaja piires,
- kviteerimisbiti õige väärtus (vt.16.2.3) kaadris ACK,
- vastuvõtupuhvri olekubitt V=0 (vt.16.2.3).

Kui kasvõi üks tingimus pole täidetud, korratakse eelmist DATA-kaadrit.
Kui ka nüüd ei saada normaalset kviitungit, lõpetatakse seanss ning infor-
meeritakse tarbivat programmi sidekanali olekubaidi vahendusel (vt.16.2.7).

Grupilise ja globaalse adresseerimise korral toimub edastus kviteerimi-
seta.

16.2.7 DRAIVERI LIIDESTUS

Süsteemprogrammide ja draiveri vaheline side luuakse andmestruktuuride
ja funktsioonide kaudu.

Edastatava informatsiooni jaoks on igas abonentarvutis eraldatud järg-
mised **puhvrid**:

- saatepuhver, 1K bait,
- vastuvõtupuhver, 1K bait.

Draiveri poole pöördumiseks saab kasutada käsundeid GET ja PUT.
Tagasisideks eraldab draiveri kasutaja oma mälupiirkonnas ühe baidi —
sidekanali olekubaidi, mille vorming on järgmine:

7	6	5	4	3	2	1	0
Q	D	E	V	T	x	x	x

x — bitti ei töödelda

Q = 1: operatsioon on lõpetatud (GET,PUT)

D = süntaksiviga (GET,PUT)

E = 1: edastus ebaõnnestus (GET,PUT)

V = 1: sihtarvuti vastuvõtupuhver on täis (PUT)/vastuvõtupuhver on tühi
(GET)

T = 1: timeout (GET,PUT)

Funktsioonide lühike spetsifikatsioon on järgmine.

GET (S,m,b,i,k) — vastuvõtupuhvri sisu lugemine

Sisendparameetrid:

S — andmete lähtearvuti aadress

m — andmete sihtaadress mällu paigutamiseks

k — kanali olekubaidi aadress

b — loetud baitide arv

i — ploki identifikaator.

PUT (S,b,m,k,i) — andmeploki saatmine teis(t)ele arvuti(te)le

Sisendparameetrid:

S — sihtarvuti aadress (vt.16.2.4)

b — saadetava ploki pikkus baitides

m — saadetava ploki algusaadress

k — kanali olekubaidi aadress

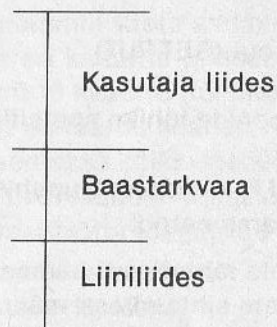
i — ploki identifikaator.

16.3 ANDMESIDETEENUSTE KASUTAMINE

16.3.1 KASUTAJA LIIDESTUS

Lauaarvuti varustamisel andmesidevahenditega tuleb arvestada muu baas-
tarkvara, eriti opsüsteemi, omadusi ja võimalusi ning üldist töötluskesk-
konda arvutis. Seetõttu realiseeritakse andmesideteenuseid mitmekihilise
liidestuse abil, mis ühendab sidedraiveri sujuvalt tarbijaprogrammide ning
operaatoriseadmetega. Sidetarkvara struktuuri ja kohta süsteemis illust-
reerib järgmine skeem.

Inimene	
Kuvar ja klaviatuur	
Tarbija programmid	Võrguhaldur (VH)
Sidemonitor (SM või SMD)	
Lisafunktsioonid	
Opsüsteem	
Püsimonitor	
Sidedraiver	
Värat	



*
*
* Sideliin

Skeemilt on näha, et andmesideteenuste kättesaadavaks tegemine nõuab opsüsteemi laiendamist lisafunktsioonidega ning kaht spetsiaalset süsteemiprogrammi, millest võrguhaldur kuulub ainult dispetšerjaama, s.t. õpetajaarvuti tarkvara koostisse. Liidese ühe osa moodustab edastuspuhvrite süsteem. Kõiki neid komponente on kirjeldatud järgmistes alajaotistes.

16.3.2 OPSÜSTEEMI LISAFUNKTSIOONID

Arvutitevahelise andmevahetuse hõlbustamiseks on nii õpilaskui õpetajaarvuti opsüsteemi viidud kaks täiendavat funktsiooni. Opsüsteemi põhivariandist (LOS) eristamiseks on sellisele võrguvariandile antud nimi VLOS. Lisafunktsioonid on järgmised.

READN (S,b,m,a,k) — andmete lugemine teisest arvutist
Sisendparameetrid:

- S** — andmete lähtearvuti aadress
- b** — lugemisele kuuluvate baitide arv
- m** — andmete sihtaadress mällu paigutamiseks
- a** — andmete mäluaadress lähtearvutis
- k** — olekubaidi aadress (vt.16.2.7)

WRITEN(S,b,m,a,k) — andmete kirjutus teis(tes)se arvuti(te)sse
Sisendparameetrid:

- S** — sihtarvuti aadress (vt.16.2.4)
- b** — saadetava sõnumi pikkus baitides
- m** — saadetava sõnumi algusaadress lähtearvutis
- a** — andmete sihtaadress sihtarvuti mälus
- k** — olekubaidi aadress (vt.16.2.7)

Tagasisideks kasutatakse kanali olekubaiti, mille vorming on kirjeldatud alajaotises 16.2.7.

16.3.3 VÕRGUHALDUR VH

Tüüpilises klassivõrgus on printerite ja välissalvestite (magnetofon, ümbrikketas) arv minimaalne ja vähemalt üht osa neist tuleb kasutada kui võrgu ühisressursse. Nende ressursside jaotamist korraldab õpetajaarvuti. Simul-taanse kasutamise süsteem on üles ehitatud nii, et õpilasarvuti saab oma opsüsteemi vahendusel kasutada õpetajaarvuti välisseadmeid nii, nagu oleksid need õpilasarvuti enda koostises.

Ressursside eraldamiseks on ette nähtud õpetajaarvutis paiknev süsteemiprogramm, võrguhaldur **VH**, mis laaditakse opsüsteemi vastava direktiiviga, dialoogi viibafaasis.

Selle programmi abil saab õpetajaarvutilt saata õpilasarvutitele käsun-deid opsüsteemi viibatasemel ja blokeerida selleks ajaks sisestuse õpilas-arvuti klaviatuurilt. Sel teel saab õpetaja saata õpilasarvutitele tarkvara komponente (translaatorid, redaktorid jne.) ja töötlemisele kuuluvaid tekstifaile.

Suhtlemine võrguhalduriga põhineb teenuste menüül, seetõttu ei vaja see detailsemat kirjeldust.

16.3.4 ANDMESIDEPUHVRID

Edastatava info paigutamiseks on kasutusel kaht tüüpi puhvrid:
— andmemassiivide (pikkade sõnumite) puhvrid: üks saate- ja üks vastuvõ-tupuhver igas arvutis, kumbki mahuga 1K bait;
— teadete (lühisõnumite) puhvrid mahuga kuni 250 baiti, puhvrite arvu saab muuta ja sellele ei ole tehtud programmilisi kitsendusi.

Teadete puhvri struktuur on järgmine:

0	s
1	p
2	n
3	v
4	
5	Andmed

s — puhvri olekubait
 p — andmevälja pikkus baitides
 n — puhvri number
 v — viit järgmisele puhvrile
 v=FFFFH puhul on puhver viimane

Esimese teadete puhvri aadress asub mälu-
 pesades OBH, OCH

Puhvri olekubaidi vorming on järgmine:

7	6	5	4	3	2	1	0
T	V	x	C	C	C	C	C

T — puhvri tüüp
 T=0: väljundpuhver
 T=1: sisendpuhver

V — puhvri hõivatus
 V=0: vaba
 V=1: hõivatud

CC — bitid 0..4 — kirjutuste loendur
 C=0: puhver on tühi
 C=1..31 — tekstijadade arv puhvris

16.3.5 SIDEMONITORID

Need süsteemiprogrammid annavad tarbijaprogrammi käsutusse hulga andmesidefunktsioone, mida saab kasutada nagu opsüsteemi funktsioone (vt. 3.1). Monitori **SM** kasutatakse õpilasarvutis, suurema funktsioonide arvuga monitori **SMD** aga õpetajaarvutis.

Monitor laaditakse nagu tavaline tarbijaprogramm ning ta paigutub vaba mälutsooni lõppu, vahetult baasoperatsioonisüsteemi ette. Parameetrite süsteemitsoonist kasutatakse mälupesi 08H..0AH.

Puhvris võib olla kuni 31 tärgijada (esimene bait jadas näitab talle järgnevate tärkide arvu). Funktsioonide spetsifikatsioon on järgmine.

0: MAKE BUFFER — uue puhvri loomine

Sisend: A = puhvri pikkus baitides

Väljund: A = 0 — puhver on loodud

1: GET BUFFER — puhvri eraldamine

Sisendid: A = puhvri pikkus baitides

B = abonentarvuti aadress

Väljundid: A = eraldatud puhvri pikkus

B = puhvri number

DE = puhvri aadress

2: DELETE BUFFER — puhvri likvideerimine

Sisend: A = puhvri aadress

Väljund: A = 0 — puhver on likvideeritud

3: READ BUFFER — puhvri sisu lugemine

Sisendid: A = tärgijada järjekorranumber puhvris

B = abonentarvuti aadress

DE = loetavate andmete sihtaadress mälus

H = puhvri number

Väljund: A = loetud baitide arv

4: WRITE BUFFER — kirjutus puhvrisesse (lisatakse puhvri lõppu tärgijada)

Sisendid: A = kirjutatavate baitide arv

B = sihtarvuti aadress

DE = andmete lähteaddress mälus

H = puhvri number

Väljund: A = kirjutamata jäänud baitide arv (0:normaalne kirjutus)

5: SHOW BUFFER — puhvri sisu kuvamine

Sisendid: A = väljastatavate baitide arv

B = abonentarvuti aadress

H = puhvri number

6: SEND TO CONSOLE — saatmine sihtarvuti ekraanile

Sisendid: A = saadetavate baitide arv

B = sihtarvuti aadress

HL = saadetava teksti algusaadress mälus

7: BUFFER STATUS — puhvri oleku seadmine

Sisendid: A = puhvri olek

A = 0:väljundpuhver

A = 1:sisendpuhver

A = 2:puhver on vaba

A = 3:puhver on tühi

A = 4:puhvril olevate tärgijadade arvu kōsimine

H = puhvri number

Väljund: A = 0:operatsioon on sooritatud(A=0..2) või tärgijadade arv(A=3)

8: DELETE NETWORK PROGRAM — sidemonitori likvideerimine

Mälust kõrvaldatakse sidemonitor, puhvrid likvideeritakse, viidad parameetrite tsoonis nullitakse.

Sisendparameetreid ei ole.

SISUKORD

I. RIISTVARA

0. PÕHITEATMEID	6
0.1 Struktuur	6
0.2 Tehnilised andmed	8
0.3 Konstruktsioon	9
0.4 Mälujaotus. Adresseerimismoodused	10
0.5 Välisseadmete ühendamine	12
0.5.1 Videoliides	12
0.5.2 Magnetofoni liides	13
0.5.3 Sideliides	15
0.5.4 Kasutaja värat	16
0.6 Arvuti laiendamine	17

II. KASUTAMISJUHISED

1. ARVUTI ÜLESSEADMINE	19
1.1 Teleri ühendamine	19
1.2 Magnetofoni ühendamine	20
1.3 Arvuti sisselülitamine	20
2. ETTEVALMISTUS TÖÖKS	20
2.1 Ettevalmistus tööks püsimonitoriga	21
2.2 Ettevalmistus tööks LOS-iga	21
2.3 Korrasoleku kontroll	23
3. VALMISPROGRAMMIDE KASUTAMINE	24

4. PROGRAMMIDE KIRJUTAMINE	25
4.1 Programmeerimiskeele valimine	25
4.1.1 Assemblerkeel	25
4.1.2 PL/M	25
4.1.3 BASIC	26
4.2 Algteksti sisestus	26
4.2.1 Basic-keelse algteksti sisestus	26
4.2.2 PL/M- ja assemblerkeelse algteksti sisestus	27
4.3 Programmeerimisvahendid	27
4.3.1 BASIC-u interpretaator	27
4.3.2 PL/M-kompilaator	29
4.3.3 Assembler	30
5. PROGRAMMIDE SILUMINE	30
5.1 Silur SID	30
5.2 Veateated	31
5.3 Arvuti mälujaotus	31
III. TARKVARA	
6. TARKVARA ÜLDKIRJELDUS	33
7. PÜSIMONITOR	34
7.1 Monitori direktiivid	34
7.2 Monitori funktsioonid	36
7.3 Kuvari töörežiimid	41
7.3.1 Tärgirežiim	41
7.3.2 Graafikarežiim	43
7.4 Klahvikoodid	43
8. OPERATSIOONISÜSTEEM	46
8.1 Üldandmeid	46
8.2 Käsupsessori residentsed funktsioonid	50
8.3 Teenindusprogrammid	52
8.4 Operatsioonisüsteemi liidestus	53
8.4.1 BLOS-i funktsioonide kasutamine	53
8.4.2 BLOS-i funktsioonide loetelu	54
8.5 Operatsioonisüsteemi teated	68

9. ASSEMBLER	69
9.1 Sissejuhatus	69
9.2 Programmi vorming	70
9.3 Operandide koostamine	70
9.3.1 Arvkonstandid	71
9.3.2 Reserveeritud sõnad	71
9.3.3 Sümbolkonstandid	71
9.3.4 Aritmeetika- ja loogikaoperaatorid	72
9.3.5 Operaatorite prioriteet	72
9.4 Assembleri direktiivid	73
9.4.1 ORG	73
9.4.2 END	74
9.4.3 EQU	74
9.4.4 SET	74
9.4.5 IF ja ENDIF	74
9.4.6 DB	75
9.4.7 DW	75
9.4.8 DS	76
9.5 Veateated	76
9.6 Objektfaili struktuur	77
Lisa A: Mikroprotsessori KP580MK80 käsustik	77
Lisa B: Programmeerimisnäiteid	91
10. MINIASSEMBLER	93
10.1 Otstarve	93
10.2 Eriomadused	93
10.2.1 Algteksti sisestus	93
10.2.2 Erinevused käsukoodides	93
10.2.3 Märkendite töötlemine	94
10.2.4 Konstandid	94
10.3 Kasutamine	94
10.4 Salvestatud tekstiga töötamise näide	96
11. PÜSIMÄLU-BASIC	98
11.1 Erisümbolid	98
11.2 Andmetööbid	98
11.3 Käsud	99

11.4	Operaatorid	102
11.5	Funktsioonid	103
11.6	Veateated	104
12.	PL/M	105
12.1	Üldandmed	105
12.2	Kompilaatori kasutamine	106
12.2.1	Pöördumine	106
12.3	Veateated	109
12.3.1	Vead töös failiga	109
12.3.2	Programmi süntaksi vead	109
13.	SILUR SID	112
13.1	Üldandmed	112
13.1.1	Käivitus	112
13.1.2	Direktiivid	113
13.1.3	Arvud	114
13.1.4	Kümnendarvud	114
13.1.5	Märgid	114
13.1.6	Märgendiviitid	115
13.1.7	Sümbolavaldised	115
13.2	Direktiivide kirjeldused	115
13.2.1	A — Reassembler	115
13.2.2	C — Alamprogrammi kutse	116
13.2.3	D — Mälutõmmis	117
13.2.4	F — Mälu täitmine	117
13.2.5	G — Programmi käivitus	117
13.2.6	H — Kuueteistkümnendarvutused	119
13.2.7	I — CP/M direktiivi sisestus	119
13.2.8	L — Pöördassembler	120
13.2.9	M — Mälusisu teisaldus	120
13.2.10	P — Kontrollpunktide seadmine	121
13.2.11	R — Laadimine	122
13.2.12	S — Kirjutus mällu	123
13.2.13	T — Jälitusrežiim	123
13.2.14	U — Sammurežiim	124
13.2.15	V — Mälupiirkondade võrdlemine	124
13.2.16	X — Programmi olekvektor	125

14.	TEKSTIREDAKTOR EDIT	126
14.1	Teksti sisestamine sõrmistega	127
14.1.1	Märgi kustutamine	127
14.1.2	Rea kustutamine	127
14.1.3	Tabuleerimine	127
14.1.4	Kombineeritud sisestamine	128
14.2	Teksti adresseerimine	129
14.2.1	Rea adresseerimine	129
14.2.2	Tekstilõigu adresseerimine	131
14.2.3	Otsingupiirkonna etteandmine	132
14.3	Redigeerimisoperatsioonid	132
14.3.1	R — Teksti sisestus failist	133
14.3.2	W — Teksti väljastus faili	133
14.3.3	C — Kopeerimine väljundfaili	133
14.3.4	L — Teksti esitus	133
14.3.5	N — Kuva edasikerimine	134
14.3.6	P — Kuva tagasikerimine	134
14.3.7	I — Teksti sisestus ja lisamine	135
14.3.8	J — Teksti parandamine	135
14.3.9	D — Tekstilõigu kustutus	135
14.3.10	A — Teksti lisamine	135
14.3.11	E — Tekstilõigu asendamine	136
14.3.12	M — Tekstilõigu teisaldus	136
14.3.13	S — Märgijadade asendamine	136
14.3.14	X — Väljumine reaktorist	137
14.4	Abioperatsioonid	137
14.4.1	T — Tabulaatori seadmine	137
14.4.2	. — Väljastuse defineerimine	138
14.4.3	K — Puhvri kustutus	138
14.5	Lisad	139
14.5.1	Tekstiredaktori direktiivid	139
14.5.2	Tekstiredaktori veateated	139
14.5.3	Reaviida väärtused	140
15.	TESTPROGRAMMID	140
15.1	Pakett «Diagnostics»	140
15.1.1	Üldised kasutusjuhised	141
15.1.2	Testandmete kogumine välissalvestisse	141
15.1.3	Käsurea parameetrite kasutamine	142

15.2 Mälu testimine	142
15.2.1 Testid ja võimalused	142
15.2.2 Mälu testi käivitamine	143
15.2.3 Mälu testi käsurea parameetrid	145
15.3 Protsessori testimine	146
15.3.1 Testioperatsioonid	146
15.3.2 CPU testi käivitamine	147
15.3.3 CPU testi käsurea parameetrid	148
15.4 Kuvari testimine	148
15.4.1 Testid	148
15.4.2 Kuvari testide käivitamine	149
15.5 Kompleksne kiirtest	149
15.5.1 Osatestid	149
15.5.2 Kompleksse kiirtesti käivitamine	150
15.6 Magnetofoni test	151
15.6.1 Kontrollifunktsioonid	151
15.6.2 Testi käivitamine	151
16. KOOLIARVUTITE KOHTVÕRK	151
16.1 Andmeside ülesanded	151
16.2 Sidesüsteemi ehitus	152
16.2.1 Tehnilised andmed	152
16.2.2 Füüsiline liidestus	153
16.2.3 Edastuskaadrid	153
16.2.4 Adresseerimine	154
16.2.5 Pöördusmeetod	155
16.2.6 Sideseanss	156
16.2.7 Draiveri liidestus	156
16.3 Andmesideteenuste kasutamine	157
16.3.1 Kasutaja liidestus	157
16.3.2 Opsüsteemi lisafunktsioonid	158
16.3.3 Võrguhaldur VH	159
16.3.4 Andmesidepuhvid	159
16.3.5 Sidemonitorid	160

Руководство по эксплуатации микро-ЭВМ «Juku». Составитель К. Мяртин. На эстонском языке. Художник-оформитель Х. Пузанов. Таллин, «Валгус». Toimetaja U. Alas. Kunstiline toimetaja M. Henno. Tehniline toimetaja T. Linkvist. Korrektor S. Hiie.

ИБ № 6086.
Laduda antud 18. 07. 86. Trükkida antud 07. 09. 87. Formaati 70 x 90/16. Trükipaber nr. 1. Kiri Heios. Tekst laotud kirjastuses. Ofsettrükk. Tingtrükipoognaid 12,29. Tingvärivõtmiseid 12,59. Arvestuspooonaid 8,24. Trükiarv 1000. Tellimuse nr. 4842. Hind 30 kop. Kirjastus «Valgus», 2000090 Tallinn, Pärnu mnt. 10. Trükikoda «Pärnutrükki», 203600 Pärnu, Ringi t. 1.



mikroarvuti
kasutamisjuhend **JUKU**

30 kop.

JUKU mikroarvuti
kasutamisejuhend