



PROGRAMME KÕIGILE



TARTU
1972

TARTU RIIKLIK ÜLIKOOL

Arvutuskeskus

Programme kõigile

III

ELEKTRONARVUTI „NAIRI - 2“

Koostanud: R. Noorma

Ü. Kaasik

I . A R V U T I E H I T U S

§ 1. ÜLEVAADE ARVUTIST

1. Põhilisi andmeid. "Nairi-2" on kaheaadressiline pooljuhtskeemidel töötav universaalne elektronarvuti, mille paindlik käskude süsteem ning püsimalus asuv ulatuslik programmide kogu võimaldavad teda kasutada väga mitmesuguste teaduslik-tehniliste ülesannete lahendamisel. Automaatse programmeerimise süsteemi (selle kirjeldus ilmub "Programme kõigile" järgmises numbris) kaudu saavad arvutit edukalt kasutada ka programmeerimisalase ettevalmistuseta inimesed. Vastav interpreteeriv programm võimaldab arvutit kasutada veel üli-võimsa lauarvutina. Arvuti on töökindel ja tema teenindamine lihtne.

Arvuti sooritab tehteid kahendsüsteemis nii täisarvudega, fikseeritud koma arvudega, liikuva koma arvudega, topeltarvudega kui ka kompleksarvudega. Arvude teisendamine kümnendsüsteemist kahendsüsteemi ja vastupidi toimub vastavates sisestus-väljastusprogrammides automaatselt.

Mälu koosneb operatiivmälust (2048 pesa) ja püsimalust (16384 pesa), aktiivset välimälu arvutil ei ole.

Käskude süsteem sisaldab 102 käsku, kusjuures pseudo-käskude kasutamine võimaldab ühe käsuga leida elementaar-

funktsiooni väärtuse või teisendada ja väljastada suvalise pesa sisu vajalikul viisil. Indeksipesasid on arvutil ainult üks, kuid käsu tingimuslike modifikatsioonide abil saab indeksipesa ülesannetes kasutada ka mälu teisi pesi.

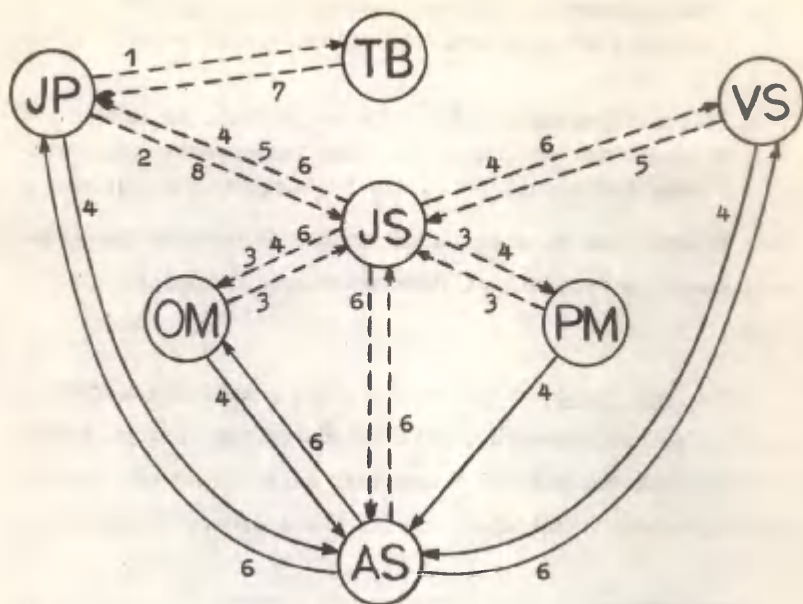
Arvuti põhisagedus on 100 kHz, tarbitav võimsus kuni 1,5 kW. Püsimälust lugemine kestab 10 μ sek, operatiivmälu poole pöördumine 24 μ sek. Keskmiselt teostab arvuti:

fikseeritud koma arvude liitmisi	- 4000 tehet sekundis,
fikseeritud koma arvude korrutamisi	- 300 tehet sekundis,
liikuva koma arvude liitmisi	- 1000 tehet sekundis,
liikuva koma arvude korrutamisi	- 200 tehet sekundis,
elementaarfunktsioonide väärtuste leidmisi	- 20 väärtust sekundis.

Välisseadmeteks on fotosisend, perforaator ja teletaip "Rioni" või kirjutusmasin "Consul"; informatsioonivahetuse kiirused vastavalt 150, 80, 7 ja 10 sümbolit sekundis. Välisseadmeid saab kasutada ka autonoomselt.

2. Funktsionaalskeem. Arvuti koosneb kahest kirjutuslaua taolisest kapist. Põhikappi on koondatud aritmeetikaseade (AS), juhtseade (JS), operatiivmälu (OM), püsimälu (PM) ja juhtpult (JP). Juhtkapis on toiteblokid (TB) ja välisseadmete juhtseade. Juhtkapi peal asuvad välisseadmed (VS). Samas leiduvad ka lülitid välisseadmete käivitamiseks ning dubleerivad juhtimislülitid.

Arvuti lihtsustatud funktsionaalskeem on esitatud joo-



Joonis 1.

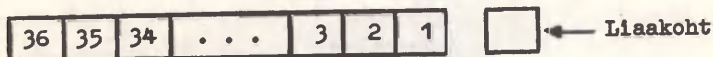
nisel 1. Sellel skeemil tähistavad arvud järgmisi signaale:

1. Juhtpuldilt antakse korraldus arvuti sisselülitamiseks.
2. Juhtpuldilt määratakse esimese käsu aadress ja käivitatakse arvuti.
3. Järjekordne käsk loetakse operatiiv- või püsimalust.
4. Käsu täitmiseks vajalike algandmete lugemine välisseadmetelt, juhtpuldilt, operatiiv- või püsimalust.
5. Välisseadme tõrkest või viivitusest algandmete saamisest teatatakse juhtseadmele ja selle kaudu juhtpuldile.
6. Tulemuse salvestamine või väljastamine (et tulemus võib mõjustada edasist tööd, siis võtab salvestamisest aktiivselt osa ka juhtseade).

7. Toitepingete väärtused teatatakse juhtpuldile ning suurema häire korral toitepingetes lülitatakse arvuti välja.
8. Arvuti peatamine võib toimuda juhtpuldilt saadud korralduse või täidetava käsu tulemusena; kui peatuks korraldust pole, algab järgmise käsu täitmine.

Punktiiriga on skeemil tähistatud juhtsignaalide, pideva joonega - töödeldava informatsiooni liikumine.

3. Registrid. Aritmeetikaseadme põhiliseks registriks on 37-kohaline summaator, mille kaudu toimub ka kogu informatsioonivahetus mälu. Summaatori kahendkohad on nummerdatud joonisel 2 näidatud viisil. Nummerdamata lisakohta ka-



Joonis 2.

utatakse vaid korrutamise ja jagamise tehete korral tulemuse ümardamiseks. Informatsioonivahetusest mälu ja teiste registritega lisakoht osa ei võta.

Mikroprogrammi täitmisel salvestatakse vahetulemused operatiivmälu fikseeritud aadressiga pesadesse $a_1 - a_6$, mis oma funktsioonilt on registrid, kuid konstruktsioonilt operatiivmälu pesad.

Juhtseadme põhiliseks registriks on 36-kohaline käsuregister, mille kohad on nummerdatud paremalt vasakule ühest kolmekümne kuueni. Käsuregistris asub täidetav käsk ja

seal toimub (summaatori abil) ka selle käsu analüüs.

Järgmisena täitmisele tulevat käsku sisaldava pesa aadress asub 14-kohalises käsuloendajas. Vastavalt sellele aadressile loetakse käsk kas operatiiv- või püsimalust (summaatori vahendusel) käsuregistrisse, kusjuures ühtlasi suurendatakse käsuloendaja sisu ühe võrra. Nihutamis- ja korrutamistehete korral kasutatakse käsuloendajat veel tavalise loendajana, salvestades seal hoiul oleva aadressi selleks ajaks käsuregistrisse.

Informatsioonivahetust operatiivmäluga juhib 11-kohaline operatiivmälu register, milles asub selle pesa aadress, kust loetakse või kuhu salvestatakse.

Püsimalust lugemist juhib 14-kohaline püsimalu register, mis mikroprogrammi täitmisel täidab ühtlasi mikrokäskude loendaja funktsioone. Kui püsimalu registris asuv aadress on ülimalt 2047, siis tõlgendatakse vastava pesa sisu mikrokäsuna, mis kuulub juhtseadme kaudu täitmisele. Kui aga registris asuv aadress on vähemalt 2048, siis saab olla tegemist vaid konstandiga (arv, käsk), mis loetakse summaatorisse.

Informatsioonivahetus välisseadmetega toimub 7-kohalise väljundregistri kaudu.

Signalisatsioonipaneelile on toodud summaatori, käsuregistri, operatiiv- ja püsimalu registreite näidud neonlampidel. Arvuti seismisel saab sealt lugeda vastava registri sisu kahendsüsteemis (põlev neonlamp tähistab ühte, kustunud lamp nulli vastaval kohal).

4. Mälu. Arvuti 2048-pesalisse operatiivmällu saavad programmid salvestada ja sealt lugeda töödeldavat informatsiooni. Informatsioon säilib pesas kuni järgmise salvestuseni sellesse pesa. Salvestamine ja lugemine on võimalik ainult terve pesa kaupa. Operatiivmälu iga pesa sisaldab 36 kahendkohta, mis nummerdatakse paremalt vasakule numbritega 1 - 36. Pesade aadressideks on kümnendarvud 0 - 2047.

Operatiivmälu juurde kuulub konstruktiivselt veel kuus fikseeritud aadressiga (36-kohalist) pesa, milliseid tähistatakse a_1 , a_2 , a_3 , a_4 , a_5 ja a_6 . Funktsionaalselt kuuluvad need aritmeetilise seadme juurde ja täidavad mikroprogrammide realiseerimisel mitmesuguste registrite funktsioone. Pesas a_2 asub alati eelmise tehte tulemus ja seda on programmides võimalik kasutada teatud käskude korral. Pesas a_5 asub viimasena loetud või väljastatud sümboli registri kood. Teistel fikseeritud aadressiga pesadel sellist kindlat tähendust pole. Informatsioonivaetus fikseeritud aadressiga pesade ja summaatori vahel toimub operatiivmälu registri osavõtuta.

Püsimälu esimesed 2048 pesa (aadressideks kümnendarvud 0 - 2047) on 72-kohalised ning neid kasutatakse käske realiseerivate mikroprogrammide hoidmiseks. Nende pesade poole pole programmides vahetult võimalik pöörduda.

Küsimälu pesad aadressidega 2048 - 16383 on 36-kohalised, kusjuures kohti nummerdame tarbekorral samuti nagu operatiivmälu pesades. Nendest pesadest saab iga prog-

ramm küll lugeda, kuid seal sisalduvat informatsiooni saab muuta üksnes vastava mähise ümberõblemisega.

Konstruktiivselt on püsimälu jagatud 2048-pesalisteks osadeks - kassettideks. Vastavate lisakassettide olemasolu korral võib kassette vahetada ja sellega muuta programmi-teeki.

Aktiivne välismälu magnettrumlite, lintide või ketaste näol arvutil puudub. Välismälu ülesannet võib mõningal määral täita perfolint.

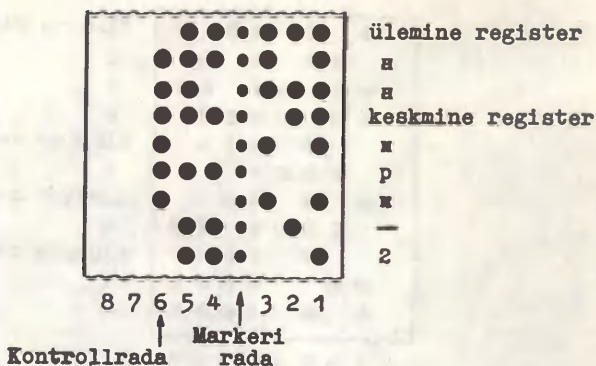
5. Välisseadmed on vajalikud töödeldava informatsiooni sisestamiseks ja tulemuste väljastamiseks. Samuti antakse välisseadmete kaudu arvutile korraldusi ja arvuti teatab mitmesugustest tõrgetest programmi täitmisel.

Arvuti "Nairi-2" välisseadmeteks on fotosisend FS-1500, lindiperforaator ПМ-80 ja teletaip PTA-60 "Rioni" või elektriline kirjutusmasin "Consul". Kui arvuti komplektis on teletaip, siis kuulub sinna juurde veel autonoomne telegraafiaparaat CTA-67B, kuid samaaegselt ei saa kasutada kirjutusmasinat "Consul".

Informatsioonivahetuseks arvutiga kasutatakse kas 5- või 6-kohalist koodi, mis on saadud vastavalt kas standardsest rahvusvahelisest telegraafikoodist ("Rioni" kasutamisel) või "Consuli" koodist mõningate sümbolite muutmise teel. Edaspidi nimetame neid koode lühidalt "Rioni" ja "Consuli" koodideks.

"Rioni" koodis esitatakse iga sümbol 5-kohalise kahendarvuna. Et erinevaid viiendkoode leidub vaid 32, siis kujutatavate sümbolite arvu suurendamiseks on kolm koodi eraldatud nn. registrimärkideks (00 - alumine register, 33 - keskmine register, 37 - ülemine register). Registrimärk ja järgnev viiekohaline kood määravad sümboli üheselt. Informatsioonivahetuse kiirendamiseks loetakse registrimärgi järel olevad koodid kõik sellesse registrisse kuuluvaiks (kuni järgmise registrimärgini). Kolm koodi - kelgu viimine algseisu (02), reavahetus (10) ja tühik (04) - on sama tähendusega kõikides registrites. Kokku on "Rioni" koodis erinevaid sümboleid seega 81, nende koodide täielik tabel on esitatud käesoleva väljaande lisas 1.

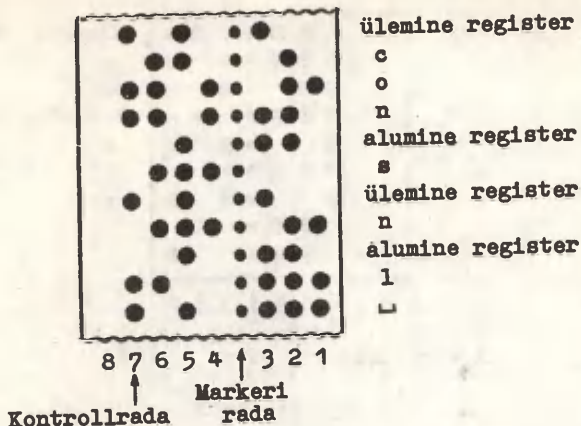
Informatsioonikandjaks on "Rioni" koodi korral 5-rajaline standardne telegraafilint, kuid fotosisendi ja perforaatori konstruktsioon lubab kasutada ka kuni 8-rajalist perfolinti. Laiemal lindi kasutamisel kannavad vajalikku informatsiooni vaid viis parempoolset rada, kusjuures kuues rada võib olla kontrollrajaks. Joonisel 3 on esitatud tüki ke 8-rajalist perfolinti tekstiga "Нарм-2". Kolmanda ja neljanda informatsiooniraja vahel paikneb sellisel lindil veel markeri rada, mida fotosisend kasutab sümboli asukoha määramiseks. Kuues on kontrollrada, kuhu lüüakse auk siis, kui viies informatsioonirajas on kokku paarisarv aukusid. Kontrolliraja olemasolu korral saab sisestamise (ja perforerimise) õigsust automaatselt kontrollida.



Joonis 3.

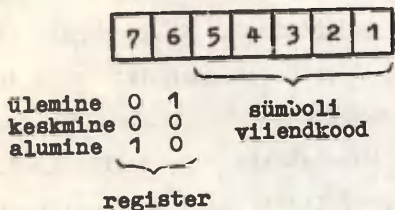
"Consuli" koodis esitatakse iga sümbol 6-kohalise kahendarvuna. Kujutatavate sümbolite arvu suurendamiseks on ka siin kaks koodi eraldatud registrimärkideks (26 - alumine register, 24 - ülemine register), mille tähendus on sama, nagu "Rioni" koodi korral. Mõlemas registris ühesuguse tähendusega on nüüd viis koodi: tühik (27), reavahetus koos kelgu viimisega algseisu (20), reavahetus (21), punane lint (22) ja must lint (23), kus kaks viimast määravad kopeerlindi värvi trükkimisel. Peale nende sisaldab "Consuli" kood veel 92 erinevat sümbolit, koodide täielik tabel on esitatud lisas 2.

Informatsioonikandjana kasutatakse "Consuli" koodi korral vähemalt 7-rajalist perfolinti, kusjuures seitsmes rada on kontrollrajaks (tähendus sama nagu "Rioni" koodi korral). Joonisel 4 on esitatud tükike 8-rajalist perfolinti tekstiga "Consul".

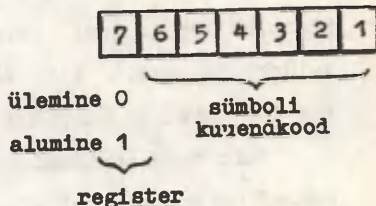


Joonis 4.

Informatsioonivahetus välisseadmetega toimub 7-kohalise väljundregistri kaudu, kus sümboli viiend- või kuuendkood ühendatakse viimati esinenud registrimärgiga. Väljundregistri kohtade jaotus "Rioni" koodi korral on näidatud joonisel 5, "Consuli" koodi korral aga joonisel 6.



Joonis 5.



Joonis 6.

Informatsiooni väljastamisel peab programmeerija veel arvestama, et "Rioni" kasutamisel mahub ühte ritta 69 sümbolit ning üleminek uuele reale toimub automaatselt. "Consuli" kasutamisel võib rea pikkus olla kuni 107 sümbolit ja seda saab sõltuvalt paberi lausest vastavate tõkestajatega muuta. Automaatset üleminekut "Consuli" korral järgmise rea algusesse ei toimu, mistõttu programmeerija peab ise arvestama rea pikkust ning vajalikes kohtades väljastama reavahetuse koodi (20). Väljastamise kiiruseks on "Rioni" korral 7 märki sekundis ja "Consuli" korral 10 märki sekundis.

Tulemuste väljastamine võib peale trükkimise (või sellega samaaegselt) toimuda ka perforeerimise teel kas "Rioni" või "Consuli" koodis. Perforeerimise kiiruseks on 80 märki sekundis.

Perfolintide lugemiseks määratud fotosisend FS-1500 töötab kiirusega kuni 150 märki sekundis.

Informatsioonivahetus kõikide välisseadmetega toimub nn. "start-stop" režiimis, mis võimaldab sisestada või väljastada ühe sümboli kaupa. Arvuti konstruktsioon näeb veel ette võimaluse kasutada välisseadmeid autonoomsel ajal kui arvuti töötab ja ei vaja välisseadmeid (näit. pika ülesande lahendamisel, kui seal esineb vähe trükkimisi).

6. Juhtpult on määratud arvuti töö juhtimiseks ja kontrollimiseks nii automaatselt kui ka kõikide häälestus-

režiimide korral. Programmeeriya vajab juhtpulti võrdlemisi vähe, sest põhilised korraldused antakse arvutile talletaibi klaviatuurilt juhtprogrammi vahendusel.

Konstruktiivselt koosneb juhtpult järgmisest kolmest osast.

Juhtkapi vertikaalsel paneelil paiknevad lülitid arvuti käivitamiseks ja seiskamiseks ("Пуск-1", "Останов", "Вариант") ning välisseadmete juhtimiseks. Samuti asub siin väljundregistri indikatsioon. "Consuli" kasutamise korral on osa nendest lülititest paigutatud kirjutusmasina esiservale.

Põhikapi vertikaalsel paneelil paiknevad summaatori, käsuregistri, käsuloendaja, püsi- ja operatiivmälu registre indikatsioonid. Siin on samuti lülitid "Пуск-2", "Ключ", "Вариант", "Шаговый", "Полуавтомат", "Останов по адресу" ja "Останов" ning signaallambid "Команда ввод", "Команда вывод", "Останов" ja "Конец".

Põhikapi eemalelukatava lauaplaadi all asuvad arvutit teenindavale insenerile vajalikud lülitid ning klaviatuur arvuti peatamiseks aadressi järgi.

Programmeerijale vajalike lülitite ja signaalide tähendused on järgmised.

Arvuti juhtimisega seotud lülitid ja signaalid.

Lüliti "Пуск-1": lülitile vajutamine annab juhtimise juhtprogrammi esimesele käsule pesas 2368.

- Lüliti "Пуск-2": lülitile vajutamine annab üldiselt juhtimise käsule, mida sisaldava pesa aadress on käsuloendajas; teisiti toimib see lüliti vaid siis, kui enne on vajutatud lülitile "Маровня" või "Полуавтомат".
- Lüliti "Останов": lülitile vajutamine peatab arvuti pärast järjekordse käsu (ka pseudokäsu alamprogrammi kuuluva käsu) täitmist; süttib signaal "Останов" ja käsuloendajasse moodustub järgmisena täitmisele tulevat käsku sisaldava pesa aadress.
- Lüliti "Маровня": lülitile vajutamine arvuti töö ajal annab sama efekti nagu lülitile "Останов" vajutamine; kui nüüd vajutada lülitile "Пуск-2", siis täidetakse ainult üks järjekordne käsk.
- Lüliti "Полуавтомат": lülitile vajutamine arvuti töö ajal peatab arvuti pärast järjekordse pseudokäsu või pseudokäsu alamprogrammis mitte asuva käsu täitmist; kui nüüd vajutada lülitile "Пуск-2", siis täidetakse järjekordne pseudokäsk tervikuna või siis üks pseudokäsu alamprogrammi mitte kuuluv käsk; lülitit kasutatakse arvuti peatamiseks siis, kui pärast on tarvis jätkata katkestatud kohast.
- Lüliti "Варпант" asendist sõltub tingimuslike käskude täitmine.

Lüliti "Ключ" asendist sõltub käsu u_{3A1} täitmine.

Lüliti "Останов по адресу": kui lüliti on sees (vastav lamp põleb), siis arvuti peatub enne vastaval klaviatuuril valitud aadressiga pesas paikneva käsu täitmist; kui lüliti on väljas, siis klaviatuuril valitud aadress ei avalda mingit mõju.

Signaal "Останов": põleb siis, kui arvuti seisab ükskõik millisel põhjusel.

Signaal "Конец": põleb siis, kui arvuti seisab peatuskäsu tõttu.

Signaal "Команда ввод": süttib arvuti pöördumisel sisen-
di poole; kui fotosisend pole sisse lülitatud, siis
arvuti peatub ja ootab informatsiooni teletaibilt.

Signaal "Команда вывод": süttib arvuti pöördumisel väl-
jundseadmete poole; kui teletaip ja perforaator on
mõlemad välja lülitatud, siis arvuti peatub.

Välisseadmete juhtimisega seotud lülitid.

Lüliti "Автономный": lülitile vajutatakse sel juhul, kui
välisseadmeid soovitakse kasutada autonoomselt; kui
nüüd programm nõuab välisseadmete poole pöördumist,
siis arvuti peatub ja süttib signaal "Команда ввод"
või "Команда вывод".

- Lüliti "Вывод на печать": lülitile vajutatakse sel juhul, kui soovitakse tulemuste trükkimist.
- Lüliti "Вывод на ПЛ": lülitile vajutatakse sel juhul, kui soovitakse tulemuste perforeerimist; kui samaaegselt on vajutatud lülitile "Вывод на печать", siis toimub nii trükkimine kui perforeerimine.
- Lüliti "Ввод ФСМ": lülitile vajutatakse sel juhul, kui soovitakse informatsiooni lugemist perfolindilt.
- Lüliti "Блокировка контроля": lülitile tuleb vajutada juhul, kui kasutataval perfolindil puudub kontrollrada; vastasel korral peatub arvuti alati, kui fotosisendilt loetakse kood, kus mulkude arv (koos kontrollrajaga) on paaritu.
- Lüliti "Блокировка готовности": kui lüliti on sees (vastav lamp põleb), siis välisseadmed (eeldades, et need on sisse lülitatud) töötavad kogu aeg; vastasel korral käivitub välisseade automaatselt igal pöördumisel tema poole ja peatub, kui kümne sekundi jooksul pärast pöördumise lõppemist ei ole uut pöördumist.
- Lüliti "Заправка ленты ПЛ": lülitile vajutamiselega perforeritakse lindile ainult markeri rada (kui perforaator on sisse lülitatud).

Välisseadmete autonoomne kasutamine toimub järgmiselt.

Perforeerimiseks teletaibilt tuleb vajutada lüliteile "Автономный", "Печать-перфорация" ja "Вывод на ПЛ" ning valida teletaibi klaviatuurilt vajalikud sümbolid.

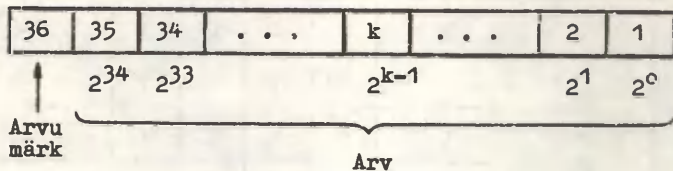
Perfolindi dubleerimiseks tuleb vajutada lüliteile "Автономный", "Ввод ФСМ", "Блокировка контроля" (vastavalt vajadusele), "Вывод на ПЛ" ja "Вывод на печать" (kui soovitakse ka trükkimist) ning panna dubleeritav perfolint sisendile. Süsteem käivitatakse lülitile "Запуск ВУ" vajutamisega. Peatada saab vajutamisega lülitile "Конец".

Perfolindi sisu trükkimiseks tuleb vajutada lüliteile "Автономный", "Ввод ФСМ", "Блокировка контроля" (vastavalt vajadusele) ja "Вывод на печать" ning panna perfolint sisendile. Süsteem käivitatakse lülitile "Запуск ВУ" vajutamisega. Peatada saab vajutamisega lülitile "Конец".

§ 2. ARVUDE KUJUTAMINE

Arvutis on võimalik teostada tehteid nii täisarvudega, fikseeritud koma arvudega, liikuva koma arvudega, topeltarvudega kui ka kompleksarvudega. Kahel viimasel juhul vajatakse arvu kujutamiseks kaht järjestikust pesa, esimesel kolmel juhul aga mahub arv ühte pesa.

1. Täisarvu kujutamisel on pesa kahendkohtadel joo-
nisel 7 näidatud tähendus. Arvu märgi jaoks reservee-
ritud kohal on positiivsete arvude korral 0, negatiivsete



Joonis 7.

korral aga 1. Kahendkohtade arv pesas määrab ka kujutata-
vate arvude diapasooni: arvuti saab opereerida vaid sel-
liste täisarvudega x , mis rahuldavad tingimust

$$-2^{35} \leq x \leq 2^{35} - 1 \quad (= 34\ 359\ 738\ 367) .$$

Negatiivsed arvud kujutatakse nn. täiendkoodis, s.t.
arvu x salvestamisel on pesas tegelikult arv $2^{35} + x$.
Tabelis 1 on toodud näiteid mõningate täisarvude kujutamise
kohta pesas.

2. Fikseeritud koma arvu kujutamisel on pesa kahend-
kohtadel joonisel 8 näidatud tähendus, kusjuures koma fik-
seeritakse kohtade 35 ja 34 vahele. Vastavalt sellele saab
fikseeritud koma arvu x pesas kujutada vaid siis, kui ta
kas on null, või $x > 0$ korral rahuldab tingimust

$$2^{-34} \leq x \leq 2 - 2^{-34} ,$$

Tabel 1.

Pesa kahendkohad								Kujutatav arv
36	35	34	33 ... 5	4	3	2	1	
1	0	0	0	0	0	0	0	-2^{35}
1	0	0	0	0	0	0	1	$1-2^{35}$
1	1	1	1	0	1	1	1	-9
1	1	1	1	1	1	1	1	-1
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	+1
0	0	0	0	1	0	0	1	+9
0	1	1	1	1	1	1	1	$2^{35}-1$

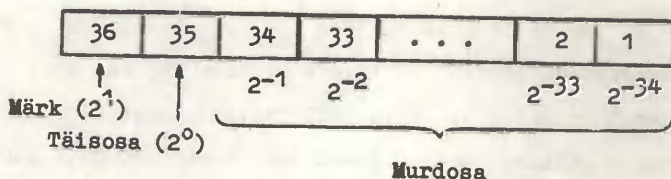
Tabel 2.

Pesa kahendkohad						Kujutatav arv	Märkusi
36	35	34	33 ... 3	2	1		
0	0	0	0	0	0	0	Arv 0
0	0	0	0	0	1	2^{-34}	Vähim positiivne arv
0	0	1	1	1	1	$1-2^{-34}$	Suurim positiivne ületäitumata arv
0	1	0	0	0	0	1	Vähim positiivne ületäitunud arv
0	1	1	1	1	1	$2-2^{-34}$	Suurim positiivne ületäitunud arv
1	1	1	1	1	1	-2^{-34}	Suurim negatiivne arv
1	1	0	0	0	0	-1	Vähim negatiivne ületäitumata arv
1	0	1	1	1	1	$-1-2^{-34}$	Suurim negatiivne ületäitunud arv
1	0	0	0	0	0	-2	Vähim negatiivne ületäitunud arv

või $x < 0$ korral rahuldab tingimust

$$-2 \leq x \leq -2^{-34}.$$

Nendesse piiridesse kuuluvat arvu nimetatakse ületäitunuks, kui ta on kas suurem kui $1-2^{-34}$ või väiksem kui -1 . Iga fikseeritud koma arv, mis on absoluutväärtuselt väiksem kui 2^{-34} ($\approx 0,58 \cdot 10^{-10}$), kujutub arvutis nullina.



Joonis 8.

Negatiivsed arvud kujutatakse täiendkoodis, s.t. arvu x salvestamisel kirjutatakse pesasse arv $4 + x$, kusjuures märgikohta tõlgendatakse kaheliste kohana.

Tabelis 2 on toodud näiteid mõningate fikseeritud koma arvude kujutamise kohta pesas. Sellest tabelist on näha, et vaid ületäitunud arvude korral pole pesa kahendkohad 36 ja 35 võrdsed.

Tehete sooritamine toimub põhiliselt ületäitumata fikseeritud koma arvudega. Ületäitumine on tunnuseks, et arvu suurendamisel pole ta enam kujutatav fikseeritud koma arvuna ning tuleb kas lõpetada arvutused või muuta lähteandmeid.

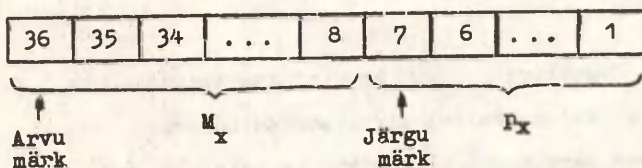
Fikseeritud koma arvude kasutamisel on eeliseks nende töötlemise suurem kiirus, puuduseks aga kujutatavate arvu c

suhteliselt väike diapasoos. Fikseeritud koma arvude korral pole mõtet kirjutada rohkem kui 11 kümnendkohta koma järele.

3. Liikuva koma arvuna kujutamiseks tuleb arv x esitada kujul

$$x = M_x \cdot 2^{p_x},$$

kus M_x on selle arvu mantiss ja p_x tema järk. Pesa kahendkohtade jaotus liikuva koma arvu kujutamiseks on näidatud joonisel 9. Arvu järk salvestatakse pesas täisarvuna, mantiss aga fikseeritud koma arvuna, kusjuu-



Joonis 9.

res ta peab olema normaliseeritud, s.t. rahuldama tingimust

$$0,5 \leq |M_x| < 1.$$

Liikuva koma arvuna saab kujutada nulli ja iga arvu x , mis rahuldab tingimust

$$0,27 \cdot 10^{-19} < |x| \leq 0,91 \cdot 10^{19}.$$

Tabelis 3 on esitatud näiteid liikuva koma arvude kujutamise kohta.

Tabel 3.

Pesa kahendkohad										Kujutatav arv	Märkusi
36	35	34	33	32...9	8	7	6	5...2	1		
0	0	0	0	0	0	0	0	0	0	0	Arv 0
0	0	1	0	0	0	1	0	0	0	$0,5 \cdot 2^{-64}$	Vähim posit. arv
0	0	1	1	1	1	0	1	1	1	$(1-2^{-27}) \cdot 2^{63}$	Suurim posit. arv
1	1	0	1	1	1	1	0	0	0	$-(\frac{1}{2}+2^{-27}) \cdot 2^{-64}$	Suurim negat. arv
1	1	0	0	0	0	0	1	1	1	-2^{63}	Vähim negat. arv

Liikuva koma arvud on laialt kasutatavad just nende kujutamise suure diapasooni tõttu. Liikuva koma arvude sisestamisel-väljastamisel kasutatakse harilikku esitusviisi (komaga). Liikuva koma arvu kujutamise täpsus on määratud mantissile eraldatud ruumiga pesas ning ei saa olla suurem kui $2^{-27} \approx 0,75 \cdot 10^{-8}$. Seetõttu pole liikuva koma arvus kunagi mõtet kirjutada rohkem kui 9 tüvenumbrit (nii näiteks on liikuva koma arvu 123456,78901234 kujutis pesas sama, mis arvul 123456,789).

Liikuva koma arvu järgu muutumine (arvutuste käigus) suuremaks kui 63 toob kaasa ületäitumise; järgu muutumine väiksemaks kui -64 muudab arvu lihtsalt nulliks.

5. Kompleksarvude salvestamiseks kasutatakse samuti kahte järjestikust pesa. Arvu reaali- ja imaginaarosa kujutatakse liikuva koma arvudena vastavalt esimeses ja teises pesas. Reaali- ja imaginaarosa kohta kehtivad samad nõuded nagu liikuva koma arvude korral. Kui reaali- või imaginaarosa puudub, siis kirjutatakse vastavasse pesa null. Teheteks kompleksarvudega on olemas liitmise, lahutamise, korrutamise ja jagamise käsud.

6. Kaheksandarvud ei kujuta sõltumatut arvude liiki, küll aga võib suvalise informatsiooni, mis on pesas salvestatud 36-kohalise kahendarvuna, tarbekorral kirjutada 12-kohalise kaheksandarvuna. Selleks tuleb kahendarv jaotada kolmekohalisteks osadeks (vt. joonis 11) ja iga osa kirjutada kaheksandnumbrina. Nii näiteks võib kaheksandarvu

36 - 34	33 - 31	30 - 28	...	9 - 7	6 - 4	3 - 1
1	2	3	...	10	11	12
K a h e k s a n d k o h a d						

Joonis 11.

100400034005 tõlgendada liikuva koma arvuna 16,1250229, fikseeritud koma arvuna 0,50390696, täisarvuna 8657057797 või ka käsuna $c7n5$. Samuti võib see kaheksandarv olla ka topeltarvu või kompleksarvu üheks pooleks.

Kaheksandarve kasutatakse eelkõige väljaperforeerimisel, sest saadud lindil võivad olla läbisegi nii mistahes liiki arvud kui ka käsud. Kaheksandarvude sisestamine on kõige kiirem, sest langeb ära nende teisendamise vajadus.

II . K Ä S K U D E S Ü S T E E M

§ 1. KÄSU STRUKTUUR

Arvuti "Nairi-2" käsud on kaheaadressilised, kusjuures esimese aadressi jaoks eraldatud kohtadele võib kirjutada kas ühe operandi aadressi või siis operandi enda väärtuse, teise aadressi jaoks eraldatud kohtadele tuleb aga kirjutada teise operandi aadress. Kui tehte tulemus salvestatakse mällu, siis alati teise aadressi järgi.

1. Mikroprogrammid. Arvuti "Nairi-2" iseärasuseks on käskude täitmine vastavate mikroprogrammide abil (mitte iga käsu realiseerimiseks konstrueeritud iseseisvate elektroniskeemidega, nagu see toimub paljudes teistes arvutites). Selleks on iga käsu täitmise käik jaotatud lihtsate elektroniskeemidega realiseeritavateks elementaarteheteks. Selliseid elementaartehteid on arvutis kokku 94, nende sisuks on näiteks juhtsignaalide seisu kontrollimine või muutmine, informatsiooni ülekande arvuti registrite või nende osade vahel, ühe sümboli sisestamine või väljastamine, lihtsate aritmeetiliste või loogiliste tehete teostamine jms. Konkreetse käsu täitmiseks tuleb sooritada terve rida üksikuid elementaartehteid teatud programmi (eeskirja) alusel, mis

moodustabki sellele käsule vastava mikroprogrammi. Käskude mikroprogrammid paiknevad püsimalu pesades 0 - 2047.

Arvuti võib ühe takti ($10 \mu\text{sek}$) vältel täita kuni kuus elementaartehtet, mis kokku moodustavad nn. mikrokäsu. Sõltuvalt käsu keerukusest võib tema mikroprogrammi moodustavate mikrokäskude arv olla oluliselt erinev.

Mikroprogrammide kasutamise eeliseks on arvuti aritmeetika- ning juhtseadme lihtsus, samuti võimaldab see võrdlemisi kergesti muuta mõne käsu sisu või koguni lisada uusi käske. Programmeerija seisukohalt pole aga tegelikult üldse oluline, kas käsk täidetakse mikroprogrammi või keeruka elektronskeemi abil, sest käsu mikroprogramm täidetakse alati tervikuna ja automaatselt. Käsu mikrostruktuuri on programmeerijal võimalik tungida ainult juhtpuldi vahendusel.

Arvuti käskude süsteemis on peale harilike käskude veel rida nn. pseudokäske, mis võimaldavad korraga täita terve harilikes käskudes koostatud alamprogrammi. See annab programmide koostamisel suurt efekti, sest üheainsa pseudokäsuga saab teostada üsna keerulisi arvutusi. Et pseudokäsk tähendab küllaltki pika alamprogrammi täitmist, siis kulub ühe pseudokäsu täitmisele muidugi hoopis rohkem aega, kui hariliku käsu korral. Olgu veel märgitud, et pseudokäskude korral kasutatakse operatiivmälu mõningaid pesi tööpesadena.

Omaette rühma moodustavad käsud aritmeetiliste tehete teostamiseks liikuva koma arvudega: välise kuju poolest on nad pseudokäsud, kuid täidetakse mikroprogrammidega ja on selles mõttes harilikud käsud.

2. Käsu tunnused. Pesasse salvestatult kujutab käsk endast 36-kohalist kahendarvu, kuid käsku täitma hakates tõlgendab juhtseade selle arvu üksikuid kahendkohti või nende rühmi omaette arvudena. Käsu viimaste kahendkohtade tõlgendamine sõltub nn. modifikatsioonist, esimest ühteistkümmend kohta (kahendkohad 36 - 26) tõlgendatakse aga alati ühtmoodi. Seal paiknevad nn. tunnused, mida tähistatakse joonisel 12 näidatud viisil. Tunnuste tähendused on järgmised.

36	35	34	33	32	31	30	29	28	27	26	...
y	l_1	l_2	n	s	x					ϕ	

Joonis 12.

Käsu liigi määrab tunnus y , nimelt $y = 1$ korral on käsk tingimuslik, $y = 0$ korral aga lihtne. Tingimuslik käsk täidetakse vaid sel juhul, kui käsus määratud tingimus on täidetud. Kui see tingimus pole täidetud, siis jäetakse vastav käsk vahele.

Käsu modifikatsiooni määravad tunnused l_1 ja l_2 . Lihtsatel käskudel on neli modifikatsiooni, mida tähistatakse tähtedega n ($l_1 l_2 = 00$), k ($l_1 l_2 = 01$), n ($l_1 l_2 = 10$) ja n ($l_1 l_2 = 11$). Tingimuslikel käskudel teine nendest modifikatsioonidest (k) puudub.

Tunnus n näitab, kas käsk on harilik ($n = 0$), s.t. täidetakse mikroprogrammi vahendusel, või pseudokäsk ($n=1$), s.t. täidetakse harilikes käskudes koostatud alamprogrammi-

na. Erandina on ka liikuva koma arvudega aritmeetilisi tehteid sooritavates käskudes $n = 1$, kuigi nad täidetakse mikroprogrammi vahendusel nagu harilikud käsudki.

Tunnus s näitab enamuse harilikkude käskude (aritmeetilised ja loogilised tehted) korral, kas üks operandid määratakse käsu teise aadressiga ($s = 0$) või võetakse selleks eelmise tehte tulemus ($s = 1$). Ülejäänud harilikkude käskude ja kõikide pseudokäskude korral tunnus s kuulub tunnuse x juurde.

Tunnus x (teatud juhtudel ka $s + x$) määrab tehtekoodi, s.t. näitab, millise tehte vastav käsk peab sooritama.

Tunnus ϕ näitab, kas käsku tuleb indekseerida ($\phi = 1$) või mitte ($\phi = 0$). Indekseerimine tähendab, et käsule kui 36-kohalisele kahendarvule liidetakse enne selle käsu täitmist operatiivmälu esimese pesa sisu (liitmisel loetakse ka märki numbrikohaks, kusjuures seal tekkivat ülekannet ei arvestata). Käske täidetakse muudetud kujul, kusjuures tunnuse ϕ uut väärtust ei arvestata, s.t. kahekordset indekseerimist ei toimu.

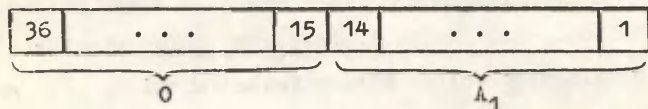
3. Modifikatsioonid. Igal käsul on kokku seitse modifikatsiooni, neist neli lihtsat ja kolm tingimuslikku. Modifikatsioonist sõltub käsu viimase kahekümne viie kohendkoha jaotus esimeseks aadressiks A_1 ja teiseks aadressiks A_2 . Samuti aga sõltub sellest veel ka esimese aadressiga A_1 esitatava operandi määramise viis. Käsu teise aadressi-

ga A_2 esitatav operand võetakse seevastu kõikide modifikatsioonide korral ikka operatiivmälu pesast aadressiga A_2 (topelt- ja kompleksarvude korral kahest järjestikusest pesast aadressidega A_2 ning $A_2 + 1$).

Võimalike modifikatsioonide tähendused on järgmised.

Lihtsa modifikatsiooni n korral ($yl_1l_2 = 000$) moodustavad aadressi A_1 käsu kahendkohad 25 - 12 ja aadressi A_2 kahendkohad 11 - 1 (aadresside maksimaalsed võimalikud väärtused on vastavalt $2^{14} - 1 = 16\ 383$ ja $2^{11} - 1 = 2047$). Sooritatava tehte esimeseks operandiks on nüüd arv, mis võetakse operatiivmälu (kui $A_1 \leq 2047$) või püsimalu (kui $A_1 > 2047$) pesast aadressiga A_1 - ehk lühidalt - pesast A_1 . Tehete korral kompleks- või topeltarvudega võetakse esimene operand kahest järjestikusest pesast, millest esimese aadress saadakse pesa A_1 kahendkohtadelt 14 - 1 (selle pesa ülejäänud kohtadel peavad olema nullid).

Lihtsa modifikatsiooni k korral ($yl_1l_2 = 001$) moodustatakse käsu aadressid samuti nagu eelmisel juhul. Esimeseks operandiks on nüüd täisarv A_1 ise, täpsemalt, operand moodustatakse kahendtäisarvust A_1 joonisel 13 näida-

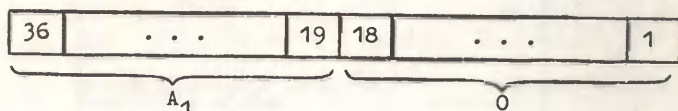


Joonis 13.

tud viisil. Näiteks $A_1 = 512 = 2^9$ annab esimeseks operandiks kahendarvu, milles kohal 10 on üks ja mujal nullid. Liikuva koma arvudega aritmeetilisi tehteid sooritavate käskude puhul on esimeseks operandiks täisarvu A_1 esitus liikuva koma arvuna. Topelt- või kompleksarvude puhul võetakse esimene operand aga pesadest A_1 ja $A_1 + 1$.

Lihtsa modifikatsiooni n korral ($yl_1l_2 = 010$) moodustavad esimese aadressi kahendkohad 25 - 8 ja teise aadressi kahendkohad 7 - 1 (aadresside maksimaalsed väärtused seega vastavalt 262 143 ja 127). Operandid saadakse täpselt samuti nagu eelmise modifikatsiooni puhul, ainult täisarv A_1 võib nüüd suurem olla ning teise operandina saab kasutada üksnes pesades 1 - 127 paiknevaid arve.

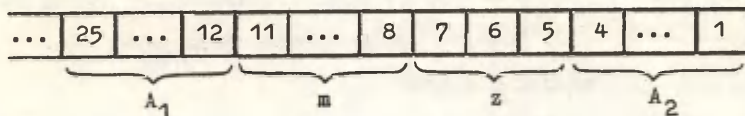
Lihtsa modifikatsiooni n korral ($yl_1l_2 = 011$) moodustatakse käsu aadressid samuti nagu eelmisel juhul. Esimeseks operandiks on nüüd arv, mis saadakse täisarvu A_1 nihutamisel 18 koha võrra vasakule, nagu on näidatud joonisel 14. Liikuva koma arvudega aritmeetilisi tehteid soorita-



Joonis 14.

vate käskude puhul on esimeseks operandiks joonisel 14 näidatud fikseeritud koma arvu esitus liikuva koma arvuna. Topelt- või kompleksarvude puhul seda modifikatsiooni ei kasutata.

Tingimuslikes modifikatsioonides jaotuvad käsu viimased kakskümmend viis kahendkohta joonisel 15 näidatud osadeks, kus m ja z määravad kontrollitava tingimuse, mil-



Joonis 15.

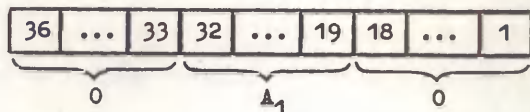
le täidetuse korral vastav käsk üldse täitmisele võetakse. Nimelt kujutab m siin operatiivmälu selle pesa aadressi, mille sisu - tähistame seda (m) - peab rahuldama tunnusega z määratud tingimust. Kontrollitavate tingimuste loetelu on esitatud tabelis 4. Sellest tabelist näeme, et ühel juhul ($z = 110$) sõltub tingimuse täidetuse mitte ar-

Tabel 4.

z	Kontrollitav tingimus
000	Kas $(m) = 0$?
001	Kas $(m) \neq 0$?
010	Kas $(m) > 0$?
011	Kas $(m) \geq 0$?
100	Kas $(m) < 0$?
101	Kas $(m) \leq 0$?
110	Kas lüliti "Вариант" on sees?
111	Kas (m) on ületäitunud?

vust (m) vaid lülitati "Бармант" asendist juhtpuldil. Kui tingimuslikus kásus on $m = 0000$, siis kõikides toodud tingimustes asendub arv (m) fikseeritud aadressiga pesa a_2 sisuga (seal paikneb harilikult eelmise tehte tulemus).

Operandide moodustamine aadresside A_1 ja A_2 abil toimub kõikide tingimuslike modifikatsioonide korral samuti nagu vastavates lihtsates modifikatsioonides. Ainult modifikatsiooni x puhul peab arvestama, et A_1 on nüüd vaid 14-kohaline (maksimaalne väärtus 16 383) ja esimene operand saadakse joonisel 16 näidatud viisil.



Joonis 16.

Kásu struktuurid kõikide modifikatsioonide korral on kokkuvõtliku tabelina esitatud lisas 3.

Elementaarfunktsioonide väärtusi arvutavates pseudo-káskudes tõlgendatakse esimest operandi alati liikuva koma arvuna nii, et kahendkohtade 36 - 8 sisu loetakse mantisiks, kahendkohtade 7 - 1 sisu aga järguks.

4. Káskude sisendkeel. Põhimõtteliselt võiks kásud programmeerimisel kirjutada kaheksandarvudena, kuid et kásu struktuur ei lange kokku tema jaotusega kaheksandnumbriteks, siis tekitaks niisugune kirjutusviis tarbetuid lisaraskusi.

Selle puuduse vältimiseks on arvutis kasutusel spetsiaalne sisendkeel käskude sisestamiseks-väljastamiseks ja üldse programmeerimiseks. Vastavate püsimalus paiknevate programmide abil teisendatakse käsk sisestamisel sisendkeelest kahendsüsteemi ja väljastamisel vastupidi.

Lihtsad käsud kirjutatakse sisendkeeles kujul

$$X A_1 \Theta A_2 \Phi$$

ja tingimuslikud käsud kujul

$$X A_1 \Theta A_2 z m \Phi ,$$

kusjuures ühesuguse tähistusega osade tähendus on mõlemal juhul sama.

Käsu tehtekoodi tähistab üldkujus sümbol X, mis joonisel 12 antud jaotusega võrreldes sisaldab tunnused n, s ja x. Harilike käskude tehtekoodiks on kas üks täht või üks täht koos numbrilise indeksiga, pseudokäskude ja liikuva koma käskude korral aga kaks tähte.

Esimese operandi või tema aadressi määrab A_1 , mis kirjutatakse kümnendarvuna: lihtsate modifikatsioonide n ja л korral võib A_1 olla vahemikust 0 - 262 143, kõikide tingimuslike modifikatsioonide ning lihtsate modifikatsioonide н ja k korral aga vahemikust 0 - 16 383. Modifikatsioonide k, n ja л kasutamisel tuleb arvestada, et kümnendarv A_1 teisendatakse sisestamisel automaatselt kahendsüsteemi. Kui osutub vajalikuks moodustada esimest operandi, mille kindlatel kahendkohtadel on ühed (näiteks loogilistes tehetes), siis võib A_1 kirjutamisel

Tabel 5.

Kahendkoha number modifikatsioonis		Kahendkoha arvuline väärtus
k	n	
1	19	1
2	20	2
3	21	4
4	22	8
5	23	16
6	24	32
7	25	64
8	26	128
9	27	256
10	28	512
11	29	1 024
12	30	2 048
13	31	4 096
14	32	8 192
15	33	16 384
16	34	32 768
17	35	65 536
18	36	131 072

võtta aluseks tabeli 5. Näiteks kui esimeses operandis peavad kohtadel 17 ja 13 olema ühed ning mujal nullid, siis tuleb kasutada modifikatsiooni n (modifikatsioon k võimaldab anda nullist erinevaid väärtusi vaid esimese operandi kahendkohtadele 14 - 1) ning kirjutada käsu esimese aadressi kohale arv $A_1 = 69\,632 (= 65\,536 + 4\,096)$. Analooogiliselt $A_1 = 16\,384$ annab modifikatsiooni n korral esimese operandi, kus kohal 33 on üks ning kõigil teistel kohtadel nullid.

Käsu modifikatsiooni määrab tunnus Θ , mille kohale tuleb kirjutada üks tähtedest n , k , n või x .

Teise operandi aadressi määrab kümnendarv A_2 , mis tingimuslike modifikatsioonide korral peab kuuluma vahemikku 1 - 15, lihtsate modifikatsioonide n ja x korral vahemikku 1 - 127 ning lihtsate modifikatsioonide n ja k korral vahemikku 1 - 2047. Käskudes, kus teist aadressi ei vajata, jäetakse A_2 kirjutamata.

Tingimuslikes käskudes esineva tunnuse z kohale tuleb kirjutada üks sümbolitest $=, \neq, >, \geq, <, \leq, |$ või ∞ , mis määravad kontrollitava tingimuse tabelis 4 esitatud loetelu järjekorras.

Tunnus m määrab pesa aadressi, mille sisu suhtes kontrollitakse tingimuse z täidetust. Selleks aadressiks võib olla kümnendarv vahemikust 1 - 15. Kui tingimust tuleb kontrollida eelmise tehte tulemuse suhtes, siis jäetakse m kirjutamata.

Kui käsku tuleb indekseerida, siis kirjutatakse tema lõppu tunnuse ϕ kohale märk $+$; vastasel juhul ei kirjutata sinna midagi.

Toome nüüd mõningaid näiteid käskude kirjutamisest sisendkeeles.

1^o Käsk $n15n1780$ on harilik, modifikatsiooniga n , ei kuulu indekseerimisele. Sisuliselt käseb see kanda pesas 15 oleva arvu pesa 1780.

2^o Käsk $c_115k98+$ on harilik ning kuulub indekseerimisele. Oletades, et 1. pesas oleva arvu kahendkohtadel 2

ja 8 on ühed, mujal aga nullid, täidetakse see käsk kujul $c_116k100$, mille toimetel eelmise tehte tulemuse kui fikseeritud koma arvu kahendkohale 5 liidetakse juurde üks ning salvestatakse tulemus pessa 100.

3° Käsk $v0n7=5$ on harilik tingimuslik, s.t. täidetakse vaid sel juhul, kui pesas 5 on null. Käsu toimetel loetakse perfolindilt üks sümbol ja kirjutatakse see pessa 7.

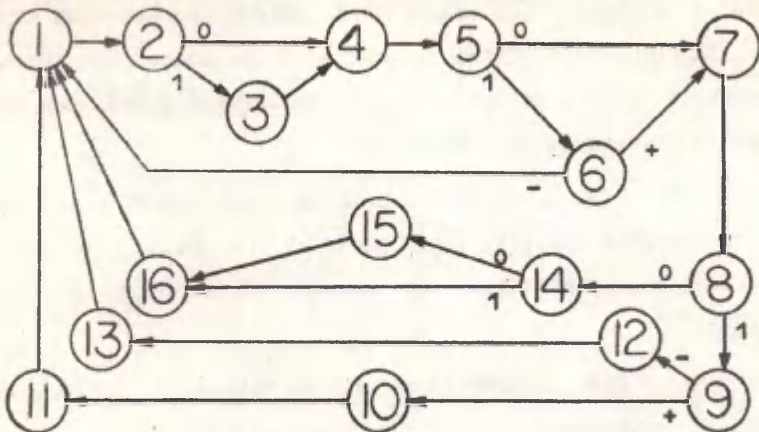
4° Käsk $cn200n110$ on liikuva koma käsk, mille täitmisel liidetakse pesas 110 olevale liikuva koma arvule juurde arv 200 ja salvestatakse tulemus tagasi pessa 110.

5° Käsk $kn74n719$ on pseudokäsk, mille täitmisel leitakse ruutjuur pesas 74 asuvast liikuva koma arvust ja salvestatakse tulemus pessa 719.

6° Käsk $cg1800k1900$ on pseudokäsk, mille täitmisel liidetakse topeltarvud pesadest 1800 ja 1801 ning 1900 ja 1901. Tulemus salvestatakse pesadesse 1900 ja 1901.

Lõpuks veel mõningaid näiteid valesti kirjutatud käskude kohta. Käsk $c7k4\infty$ on vigane, sest arvutis puudub tingimuslik modifikatsioon k. Käsus $n7n20|$ on teine aadress liiga suur, samuti käsus $b210n200$. Käsus $1_1200000k$ seevastu on esimene aadress liiga suur.

5. Käsu täitmine toimub joonisel 17 esitatud üldise skeemi kohaselt. Blokkide tähendused sellel skeemil on järgmised.



Joonis 17.

1. Valitakse järjekordne käsk käsuloendajas oleva aadressiga pesast.
2. Analüüsitakse tunnust ϕ (käsu kahendkoht 26).
3. Juhul $\phi = 1$ liidetakse käsule juurde 1. pesa sisu.
4. Käsuloendaja sisu suurendamine ühe võrra.
5. Analüüsitakse tunnust γ (käsu kahendkoht 36).
6. Juhul $\gamma = 1$ kontrollitakse tingimuse täidetust; kui tingimus pole täidetud, siis alustatakse järgmise käsu täitmist.
7. Esimene operand kantakse fikseeritud aadressiga pesa a_1 .
8. Analüüsitakse tunnust n (käsu kahendkoht 33).

9. Juhul $n = 1$ kontrollitakse, kas on tegemist liikuva koma käsuga (+) või pseudokäsuga (-).
10. Operandide mantissid ja järgud salvestatakse fikseeritud aadressidega pesadesse.
11. Täidetakse liikuva koma käsu mikroprogramm; kui sealjuures tekib ületäitumine, siis trükitakse ∞ ja peatatakse arvuti.
12. Operatiivmälu pesadesse 3 - 7 salvestatakse operandide mantissid ja järgud, samuti tulemuse salvestamise aadress ning käsuloendaja seis.
13. Käsuloendajasse kantakse vastava pseudokäsu programmi algusaadress ja täidetakse see programm, mille lõpus toimub käsuloendaja seisu taastamine (pseudokäsu programmi iga käsk täidetakse omakorda sama skeemi järgi).
14. Analüüsitakse tunnust s (käsu kahendkoht 32).
15. Juhul $s = 0$ võetakse teiseks operandiks käsu teise aadressiga määratud pesa sisu ja salvestatakse see fikseeritud aadressiga pesa a_2 .
16. Täidetakse hariliku käsu mikroprogramm; kui käsk peab kontrollima ületäitumist, siis ületäitumise korral salvestatakse järgmise käsu aadress pesa 2046 ning antakse juhtimine pesas 2047 olevale käsule.

§ 2. SELGITUSI KÄSKUDE KOHTA

1. Tabel lisas 4 esitab käskude süsteemi tervikuna, kusjuures käskude kasutamise otstarvet silmas pidades on ta jaotatud kaheteistkümneks osaks.

Tabeli esimeses ja teises veerus on esitatud käsu tehtekood, s.t. tunnuste n , s ning x väärtused vastavalt kaheksand arvuna ja käskude sisendkeele sümboolikas. Nii näiteks tehtekood 56 (ehk sisendkeeles x_1) tähendab, et käsu kahendkohtadel 33 - 27 paikneb kahendarv 0101110; tehtekoodi 101 (= cn) korral paikneb seal aga kahendarv 1000001.

Tabeli kolmandas veerus toodud tehte nimetusele tuleb aritmeetiliste tehete korral lisada veel tabeli pealkirjast arvu liigi nimetus. Nii saame tehete nimeks mitte lihtsalt "liitmine", vaid näiteks "fikseeritud koma arvude liitmine".

Tabeli neljandas veerus on antud tehete sisu kirjeldus, kusjuures on kasutatud järgmisi tähistusi:

$[A_1]$ - esimese aadressiga määratud operand; meenutame, et aritmeetilisi tehteid liikuva koma arvudega sooritavates käskudes on esimeseks operandiks $[A_1]$ teisendus liikuva koma arvuks, pseudokäskude korral aga $[A_1]$ tõlgendus liikuva koma arvuna;

$[A_1]^P$ - esimese aadressiga määratud operand tehete kor-

ral kompleks- või topeltarvudega, mis võetakse teatavasti kahest järjestikusest pesast (paaris-pesast), kusjuures esimese pesa aadress on määratud $[A_1]$ neljateistkümneme viimase kohaga;

() - mingi aadress või registri nimi ümarsulgudes tähendab vastava pesa või registri sisu; nii tähendavad (A_2) , (a_2) , (KL) ja (VR) vastavalt käsu teise aadressiga määratud pesa, fikseeritud aadressiga pesa a_2 , käsuloendaja ja väljundregistri sisu;

$(A_2)^D$ - teise aadressiga määratud operand tehete korral kompleks- või topeltarvudega (võetakse pesadest A_2 ja $A_2 + 1$);

$[]^1$ või $()^1$ tähendab arvilise astendaja 1 korral vastava koodi viimasest 1 kohast moodustatud kahendarvu;

$\overrightarrow{}$ või $\overleftarrow{}$ tähendab sulgudes antud pesa sisu nihet $[A_1]^7$ koha võrra kas aritmeetiliselt paremale või loogiliselt vasakule;

\rightarrow - tehte tulemuse salvestamine selle märgi järele kirjutatud aadressiga pesadesse nagu $A_2, 3$ või registritesse nagu KL, VR, a_2 ; kompleks- ja topeltarvude korral tähistatakse kirjutisega $\rightarrow A_2^D$ salvestust pesadesse A_2 ja $A_2 + 1$.

Kui tehte sisu kirjelduses ei esine sümbolit A_1 või

A_2 , siis vastavas käsus võib mittevajaliku aadressi kohale kirjutada mistahes (sisendkeele reeglitega kooskõlas oleva) arvu. Tavaliselt kirjutatakse puuduva esimese aadressi kohale null, kuid puuduv teine aadress jäetakse hoopis kirjutamata. Nii näiteks teise übersalvestamiskäsu võib kirjutada kujul n_106A_2 , liitsuunamise käsu aga kujul $uA_1\theta$. Teise aadressi võib jätta kirjutamata ka juhul $A_2 = 0$. Näiteks kui suhtelise suunamise tagasipöördumist ei soovita kasutada, siis kirjutatakse see käsk kujul $e_1A_1\theta$.

Tabeli kõrvale paremale on kirjutatud märkuste numbrid, kusjuures märkused ise paiknevad vastava tabeliosa lõpus.

2. Aritmeetilised tehted fikseeritud koma arvudega on kõik kahes variandis sõltuvalt sellest, kas soovitakse arvestada või ignoreerida ületäitumist. Ületäitumist arvestavad käsud (tehtekoodiks täht koos indeksiga 2 või 3) kirjutavad ületäitumise korral käsuloendaja sisu pesa 2046 (täisarvuna) ja annavad juhtimise pesas 2047 olevale käsure. Seda arvestades peab pesa 2047 varem olema kirjutatud peatuskäsk või suunamiskäsk ületäitumist likvideerivasse programmi.

Aritmeetiliste tehete sooritamisel liikuva koma arvudega tuleb modifikatsioonide k , n ja π korral arvestada, et esimeseks operandiks on $[A_1]$ kui täisarvu ($\theta = k, n$) või kui fikseeritud koma arvu ($\theta = \pi$) teisen-
dus liikuva koma arvuks. Seetõttu tuleb täisarvuliste kons-

tantide korral kasutada käsu modifikatsioonid k ja n ning absoluutväärtuselt ühest väiksemate konstantide korral modifikatsiooni x . Ületäitumist analüüsivad kõik liikuva koma käsud; ületäitumise korral trükitakse ∞ ja peatatakse arvuti, kusjuures käsuloendajas asub ületäitumist põhjustanud käsku sisaldava pesa aadress pluss üks.

Tehtekoodide parema meeldejätmise huvides on tabelis 6 näitena esitatud lahutamiskäskude tehtekoodide venekeelne tõlgendus. Teiste liikuva koma käskude korral on tõlgendus analoogiline.

Tabel 6.

Kood	Tõlgendus	Tehe
bn	<u>В</u> ыч <u>и</u> тание <u>П</u> лаваш <u>ч</u> ее	$[A_1] - (A_2) \rightarrow A_2, a_2$
ob	<u>О</u> братное <u>В</u> ыч <u>и</u> тание	$(A_2) - [A_1] \rightarrow A_2, a_2$
bб	<u>В</u> ыч <u>и</u> тание <u>Б</u> ез за <u>п</u> иси	$[A_1] - (A_2) \rightarrow a_2$
bc	<u>В</u> ыч <u>и</u> тание из <u>С</u> ум <u>м</u> атора	$(a_2) - [A_1] \rightarrow A_2, a_2$

Kõikides pseudokäskudes trükitakse ületäitumise korral ∞ ja peatatakse arvuti, ainult ületäitumist põhjustanud pseudokäskudele järgnevat käsku sisaldava pesa aadress asub nüüd summaatoris (ja pesas 7) kohtadel 14 - 1.

Tehetest täisarvudega on eraldi käsud üksnes korrutamise ja jagamise jaoks, sest liitmise ja lahutamise puhul saab kasutada fikseeritud koma arvude vastavaid käske. Esi-

mest operandi $[A_1]$ tõlgendatakse kõikides modifikatsioonides täisarvuna.

Teheteks kompleks- ja topeltarvudega kasutatakse põhiliselt modifikatsioon k ja n. Esimene operand asub sel juhul pesades $[A_1]^{14}$ ja $[A_1]^{14} + 1$, teine aga pesades A_2 ja $A_2 + 1$.

3. Loogiliste tehete korral käsitletakse operande 36-kohaliste kahendarvudena, kusjuures igal kohal teostatava tehte kirjeldus on antud tabelis 7.

Tabel 7.

1.operand 2.operand	Formeerimine		Osa eraldamine		Võrdlemine	
	0	1	0	1	0	1
0	0	1	0	0	0	1
1	1	1	0	1	1	0

Aritmeetilise nihke korral jääb koht 36 (märgikoht) paigale, kusjuures vasakul vabanevatele kohtadele tekivad positiivse arvu korral nullid, negatiivse arvu korral - ühed. Topeltarvu aritmeetilisel nihkel läheb ülekanne esimese pesa lõpust teise pesa kohale 34. Loogilistest nihetest võtavad osa kõik 36 kohta, kusjuures paremal vabanevatele kohtadele tekivad nullid. Topeltarvu loogilise nihke korral läheb ülekanne teise pesa kohalt 34 esimese pesa

lõppu. Nihke suuruse määrab igal juhul esimese operandi kohtadel 7 - 1 olev kahendarv, nihke suund on määratud aga käsuga (loogiline nihe - vasakule, aritmeetiline nihe - paremale).

4. Suunamiskäsud u ja u_1 annavad mõlemad juhtimise pesas $[A_1]^{14}$ olevale käsule, ainult käsk u_1 kirjutab ühtlasi pesa A_2 käsu $u(KL)n$, millega tagatakse tagasipöördumine järgmisele käsule pärast vastava alamprogrammi täitmist (mis kas lõpeb pesas $A_2 - 1$ või annab muul teel juhtimise käsule pesas A_2). Kui näiteks pesas 800 asub käsk $u_1 1500k1640$, siis selle käsu täitmise tulemusena salvestatakse pesa 1640 käsk $u801n$ ning antakse juhtimine pesas 1500 asuvale käsule.

Käsku u_2 kasutatakse ainult pseudokäsu alamprogrammi lõpukäsuna, sest lisaks suunamisele muudab ta ühe juhtimissignaali seisu, tagades pseudokäsu alamprogrammi täitmise tervikuna ka siis, kui lüliti "Полуавтомат" on sees (pseudokäsu alamprogrammi mitte kuuluvaid käske täidetakse samas olukorras ühekaupa).

Suhtelise suunamise käsu $e_1 A_1 \Theta A_2$ toimel antakse juhtimine pesas $[A_1] + (KL)$ olevale käsule, kus liitmine toimub mooduli $2^{14} = 16\ 384$ järgi. Tagasipöördumise käsk salvestatakse pesa A_2 samuti nagu käsu u_1 korralgi. Näiteks pesas 200 asuv käsk $e_1 Nk400$ kirjutab pesa 400 käsu $u201n$ ning annab juhtimise $N = 100$ korral pesas

301 olevale käsule ($301 = 201 + 100$) ning $N = 16\ 375$ korral pesas 192 olevale käsule ($192 = 201 + 16\ 375 - 16\ 384$).

Tingimusliku suhtelise suunamise käsk $e_2 A_1 \ominus A_2$ annab juhul $(a_2) = [A_1]$ juhtimise pesas (KL) + A_2 olevale käsule ja juhul $(a_2) \neq [A_1]$ suunamiskäsule järgnevale käsule. Analoogiliselt töötab ka teine tingimusliku suhtelise suunamise käsk $e_3 A_1 \ominus A_2$, ainult kontrollitav tingimus on keerulisem. Sisuliselt kontrollitakse, kas arvu (a_2) kõik need kahendkohad võrduvad ühega, kus arvus $[A_1]$ ühed on. Kontrollimisel sooritatavate tehete käik on näidatud tabelis 8.

Tabel 8.

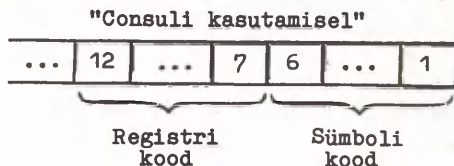
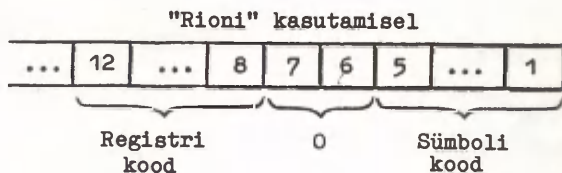
(a_2)	$[A_1]$	$(a_2) \wedge [A_1]$	$(a_2) \wedge [A_1] \sim [A_1]$
0	0	0	0
0	1	0	1
1	0	0	0
1	1	1	0

Tingimusliku suhtelise suunamise käsuga saab juhtimist anda ainult "tahapoolle", sest liitmine (KL) + A_2 toimub mooduli 2^{14} järgi ja aadressi A_2 maksimaalne väärtus on 2047. Kõikide suunamiskäskude korral säilib pesa a_2 sisu.

5. Sümboli sisestamise käsk $\vee Ok A_2$ kodeerib selle sümboli ühtlasi (kas "Rioni" või "Consuli" koodist) spetsiaalsesse 36-kohalisse koodi. See n.ö. sisemine kood

saadakse pesast, mille aadress moodustatakse arvu 2048 liitmisega sisestamisel väljundregistrisse tekkinud seitsmekohalisele koodile. Sümbolite sisemised koodid ja vastavate pesade aadressid on antud tabelina lisas 5. Siin tuleb silmas pida, et kuigi "Rioni" ja "Consuli" koodide korral vastab samale sümbolile ühesugune sisemine kood, on sümbolite koodide järjekord tabelis siiski erinev. Kui sisestatav kood osutub registrimärgiks, siis salvestatakse see fikseeritud aadressiga pessa a_5 ning pöördutakse sellesama käsu toimele uuesti sisendi poole.

6. Sümboli väljastamine toimub käsuga o või o_1 . Mõlema variandi korral peab väljundregistrisse saadetaval arvul (käsu o puhul $[A_1]$, käsu o_1 puhul $(2176 + (a_2))$) olema joonisel 18 näidatud kuju.



Joonis 18.

Sobiva arvu saamiseks kasutatakse kõige sagedamini erilist väljastuskoodide massiivi, mis paikneb püsimalus alates pesast 2176. Täielikult on see massiiv toodud lisas 6. Nii näiteks sümboli a saab väljastada käsuga 02221H nii "Rioni" kui ka "Consuli" korral, sest pesas 2221 paikneb esimesel juhul kaheksandarv 000 000 007 627 ja teisel juhul 000 000 005 012. Käsku o_1 kasutades saab sama sümboli väljastada siis, kui $(a_2) = 45$, sest sellega väljastatakse kood pesast $2176 + 45 = 2221$. Ühe sümboli väljastamise käskude täitmisel säilib pesa (a_2) sisu.

Liikuva koma arvu trükkimise käsul on kaks modifikatsiooni sõltuvalt A_2 väärtusest. Kui $A_2 \leq 31$, siis trükitakse liikuva koma arv A_2 kohaga koma järel. Kui parameeter $A_2 = 512 + q$, siis trükitakse arv q kohaga koma järel ja nii mitu tühikut, et trükitud arvu pikkus koos tühikutega oleks 20 sümbolit.

Kompleksarvud trükitakse kujul $A \pm Bi$, kus A ja B on liikuva koma arvud.

Käsud trükitakse sisendkeeles, kusjuures tundmatu tehete koodiks trükitakse täht m .

Indeksite trükkimisel peab olema $[A_1] < 1000$, vastasel juhul trükitakse *nu ja arvuti peatub.

7. Funktsioonide arvutamise pseudokäskude kirjeldamisel on tabelit lisas 4 täiendatud veeruga "Avariipeatus", kus näidatakse tingimus, mille täidetuse korral arvuti peatub. Peatumist põhjustanud pseudokäsule järgnevat

käsku sisaldava pesa aadressi saab summaatori (ja pesa 7) kohtadelt 14 - 1. Funktsiooni arvutamise käskudes kasutatakse põhiliselt modifikatsiooni H.

Normaliseerimiskäsu H või H_1 täitmisel tõlgendatakse normaliseeritavat arvu fikseeritud koma arvuna. Nihe-
te arv kirjutatakse täiendkoodis pesa a_2 .

Käsud H_G ja G_H võimaldavad üleminekut topeltarvult liikuva koma arvule ja vastupidi.

Käsud M_N ja M_X võimaldavad leida minimaalset või maksimaalset elementi liikuva koma arvude massiivist, mis asub pesades $[A_1]$, $[A_1] + 1, \dots, A_2$. Minimaalne (või maksimaalne) element saadetakse pesa a_2 , mooduli poolest minimaalne (või maksimaalne) element pesa 9.

Pseudokäskude korral tuleb arvestada, et nende alamprogrammid kasutavad mõningaid operatiivmälu pesi (pesi 3 - 7 alati) tööpesadena; nende pesade loetelu on toodud lisas 7.

III . A R V U T I K A S U T A M I N E

§ 1. TEENINDUSPROGRAMMID

Arvuti "Nairi-2" programm-varustus sisaldab terve rea universaalse iseloomuga programme, mida kasutatakse kas mitmesuguste sageliesinevate n.ö. tüüpülesannete lahendamiseks või siis programmeerimise ja üldse arvuti käsitlemise lihtsustamiseks. Viimaste (nn. teenindusprogrammide) hulgas on olulisemateks järgmised.

1. Juhtprogramm on määratud arvuti töö organiseerimiseks - tema abil saab sisestada informatsiooni ja anda juhtimist teistele (nii operatiiv- kui ka püsivõimega paiknevatele) programmidele.

Kui juhtprogrammi poole soovitakse pöörduda mingi programmi täitmise ajal, siis tuleb arvuti kõigepealt peatada lülitile "Полуавтомат" vajutamise teel (lüliti "Останов" või "Шаговый" kasutamisel arvuti peatamiseks pole pärast juhtprogrammi töö lõppemist reeglina enam võimalik jätkata katkestatud programmi täitmist pooleliäänud kohast). Lülitile "Пуск-1" vajutamine salvestab nüüd käsuloendaja sisu (KL) automaatselt pessa 2045 ja annab juhtimise juhtprogrammi esimesele käsule, mis paikneb püsi-

mälu pesas 2368. Juhtprogrammi poole võib pöörduda ka suvalisest programmist, andes lihtsalt juhtimise pesas 2368 olevale käsule. Sel juhul jääb aga ära käsuloendaja sisu salvestamine.

Juhtprogrammi töö algab sellega, et ta salvestab fikseeritud aadressiga pesa a_2 sisu pesa 2044 ja pesa 1 sisu pesa 2043 (need arvud võivad katkestatud programmi täitmise jätkamisel vajalikeks osutuda). Seejärel arvuti peatub, süttib signaal "Команда ввод" ja juhtprogramm ootab edasist tegevust täpsustavat korraldust.

Korraldus juhtprogrammile antakse kas vahetult tele-
taibi klaviatuurilt või siis varem ettevalmistatud perfori-
lindi abil fotosisendilt. Kuju ja otstarbe järgi jagunevad
korraldused järgmisse nelja klassi.

Kümnendarvust N ja tähest u koosnev korraldus ku-
jul Nu annab juhtimise pesas N paiknevale käsule. Näi-
teks korralduse 128u toimel asutakse täitma pesas 128
algavat programmi ning juhtprogrammi töö on sellega lõppe-
nud.

Kahest tähest koosneva korraldusega asutakse täitma
programm-varustuse hulka kuuluvat tüüpülesande lahendamise
või teenindusprogrammi. Sellisel viisil rakendatavate prog-
rammide ja vastavate korralduste loetelu on esitatud tabe-
lis 9. Nende programmide üksikasjalikud kirjeldused tuuak-
se "Programme kõigile" järgmises väljaandes.

Lisaks loetletutele on veel ette nähtud kaks ühetähe-
list korraldust pöördumiseks teenindusprogrammide poole.

Tabel 9.

Korraldus	Täidetav programm	Korraldus	Täidetav programm
bb	väljastusprogramm	OM	pöördmaatriksi leidmine
cc	lauaarvuti programm	MO	
on	silumisprogramm	Pk	1. järku harilike diferentsiaalvõrrandite süsteemi lahendamine
o6	testprogramm	SY	
yy	tagasipöördumine põhiprogrammi	cy	lineaarvõrrandite süsteemi lahendamine
an	transleerimise alustamine	hk	lineaarvõrrandite süsteemi lahendamine vähimruutude meetodiga
na	transleerimise jätkamine		
YM	maatriksite korrutamine	un	interpoleerimine
bo	determinandi leidmine	CM	lineaarse planeerimisülesande lahendamine simpleksmeetodiga

Nimelt korraldus ν annab juhtimise kuupäeva salvestamise programmile ja korraldus μ peatab arvuti (kasutatakse peamiselt viimase korraldusena perfolindil).

Informatsiooni sisestamise korraldus koosneb kümnend-arvust ja sellele järgnevast ühetähelisest koodist, mis määrab sisestatava informatsiooni liigi. Koodide lubatavad väärtused ja nende tähendused on loetletud tabelis 10. Korraldusse kuuluv täisarv N annab selle pesa aadressi, millest alates tuleb sisestatav informatsioon salvestada. See informatsioon antakse arvutile samuti nagu korralduski kas

Tabel 10.

Kood	Informatsiooni liik
m	täisarvud
v	fikseeritud koma arvud
n	liikuva koma arvud
g	topeltarvud
m	kompleksarvud
k	käsud
b	kaheksandarvud

teletaibi klaviatuurilt või perfolindiga. Informatsiooniühikute (arvude või käskude) eraldajatena tuleb sealjuures kasutada kas tühikut () või reavahetuse märki ((bk)). Standardse teletaibi abil perforeerides tuleb silmas pidada, et reavahetuse märki (nc) võib perforeerida ainult pärast

eraldajat. Kümndarvud ja sisendkeeles esitatud käsud teisendatakse sisestamisel automaatselt kahendsüsteemi. Kui pärast sisestamise lõppu soovitakse anda juhtprogrammile uut korraldust, tuleb viimase eraldaja järele sisestada sümbol m .

Sisestamisel loetakse kõik järjestikused sümbolid kuuni esimese eraldajani üheks informatsiooniühikuks (arv või käsk) ning salvestatakse pärast teisendamist pessa N, topelt- või kompleksarvude korral pesadesse N ja N + 1. Järgmine informatsiooniühik salvestatakse juba pessa N + 1 (topelt- või kompleksarvude korral pesadesse N + 2 ja N + 3) jne.

Sisestatava informatsiooni esitamisel tuleb silmas pidada järgmisi nõudeid.

Positiivse arvu ette märki " + " ei panda. Kõik arvud peavad olema pesas kujutatavad: suuremad arvud loetakse vigasteks, väiksemad võrdsustatakse nulliga. Kaheksand-arvudes võetakse arvesse 12 viimast kohta. Liikuva koma arvud, samuti kompleksarvude reaali- ja imaginaariosad, esitatakse koma abil. Kompleksarvud sisestatakse kujul $X + Yi$, kus X ja Y on liikuva koma arvud (nulliga võrduva imaginaar- või reaalosa võib kirjutamata jätta). Käsud peavad olema esitatud sisendkeeles.

Kui need nõuded pole täidetud, siis tunnistatakse informatsioon vigaseks ning trükitakse * . Kui lülititi "Бермант" on sees, siis trükitakse ühtlasi selle pesa aadress, kuhu pidi salvestatama vigane informatsiooniühik ja jäetakse see pesa (topelt- või kompleksarvude korral kaks pesa) vahele. Niiugust moodust kasutatakse põhiliselt andmete sisestamisel perfolindilt, sest ta võimaldab juhusliku vea korral vältida nihet järgnevate arvude või käskude salvestamisel. Kui lülititi "Бермант" on väljas, siis vea korral pesa aadressi ei suurendata ning järgnev informatsiooniühik paigutatakse samasse pesa. Niiugust moodust kasutatakse andmete sisestamisel klaviatuurilt, sest vea avastamisel saab seda kohe parandada.

Näide 1. Olgu tarvis salvestada pesadesse 56 ja 57 liikuva koma arvud $3 \cdot 10^{-5}$ ja $-12,5$, pesa 70 täisarv 1970 ning anda seejärel juhtimine pesas 370 paiknevale kä-

sule. Selleks tuleb vajutada lülitile "Пуск-1" ning valida klaviatuurilt (või varem valmis perforeerida) sümbolid:

56n0,00003_12,5_m70m1970_m370u

Näide 2. Olgu tarvis katkestada käsiloleva ülesande lahendamine, saata pesadesse 1700 - 1707 kompleksarvud 1,79 - 0,91, -5,8 + i, 2,3 , 3i ning jätkata seejärel programmi täitmist. Selleks tuleb vajutada lülitile "Полупавтомат" kuni süttib signaal "Останов", vajutada siis lülitile "Пуск-1" ning valida klaviatuurilt sümbolid:

1700m1,79+0,91_-5,8+i_2,3_3i_my

2. Väljastusprogramm võimaldab kas trükkida või perforeerida käske ja arve, viimaseid nii kahend-, kaheksand- kui kümnendsüsteemis. Väljastusprogrammi poole pöörduetakse tavaliselt juhtprogrammist korraldusega bb. Selle toimel jääb arvuti ootama väljastusviisi täpsustavat nn. väljastuskorraldust.

Väljastuskorralduse üldkujus

$N \times M \times y$ (bk)

on N ja M kümnendtäisarvud, neist N määrab väljastatava massiivi esimese pesa aadressi, M aga väljastamisele kuuluvate informatsioonihikute (arvude või käskude) arvu. Sümboli x kohale tuleb väljastuskorralduses panna

informatsiooni liiki määrav täht tabelist 10 (lisaks on trükkimise korral lubatud kasutada veel tähte c, mis tähendab trükkimist kahendarvuna). Väljastuskorralduse neljas osa y täpsustab väljastatava informatsiooni vormistamisviisi.

Osa y puudumisel trükitakse väljastatav informatsioon kahes veerus - esimeses pesade aadressid kümnendsüsteemis, teises nende pesade sisud tähega x määratud kujul. Näiteks korraldusega 36n3(bk) trükitakse pesade 36, 37 ja 38 sisud liikuva koma arvudena, korraldusega 36g3(bk) aga kolm topeltarvu pesadest 36 - 41.

Kui korralduse osaks y on täht b, siis toimub trükkimine neljas veerus - pesa aadress kümnend- ja kaheksandsüsteemis ning sisu kaheksandsüsteemis ja tähega x määratud kujul (lubatud on nüüd vaid x väärtused c, m, v, n ja k). Näiteks korraldusega 270k112b(bk) trükitakse pesades 270 - 381 paiknevad käsud nii kaheksandsüsteemis kui ka sisendkeeles.

Trükitav tekst vormistatakse mõlemal vaadeldud juhul lehekülgedena - igal leheküljel 50 rida. Kui lülititi "Вариант" on sees, siis arvuti peatub pärast iga lehekülje trükkimist ja jätkamiseks tuleb vajutada lülitile "Пуск-2".

Kui väljastuskorralduse y kohale kirjutada täht n, siis väljastatav informatsioon perforeeritakse juhtprogramiga sisestamiseks sobiva skeemi järgi. Nimelt perforeeritakse kõigepealt väljastuskorralduse algus Nx ja vahetult

selle järele M informatsiooniühikut, eraldades need üks- teisest tühikutega. Viimase tühiku järele perforeeritakse veel täht m, et pärast saadava lindi sisestamist toimuks taas pöördumine juhtprogrammi algusesse.

Kirjutades väljastuskorralduses y kohale tähed nk toimub perforeerimine samuti nagu eelmisel juhul, ainult tähe m järele perforeeritakse veel täht n, s.t. pärast sisestamist arvuti peatub. Kaht viimativaadeldud väljastusviisi võib kasutada ka trükkimise asemel, sest pesade aadresside väljastamatajätmine lühendab väljastusaega.

Väljastusprogrammi kasutatakse üksikute pesade või mas- siivide väljastamiseks eelkõige programmide silumisel ja avaripeatuste põhjuste väljaselgitamisel. Väljastusprog- rammi võib samuti kasutada programmide vormistamisel.

Väljastusprogrammist võib korraldusega yy pöörduda tagasi põhiprogrammi koos pesade a_2 ja 1 sisu taastami- sega. Korraldusega cc saab pöörduda lauaarvuti programmi poole.

3. Lauaarvuti programm võimaldab teostada küllalt- ki keerulisi arvutusi ilma vastavat programmi eelnevalt koostamata. Põhiliselt kasutatakse seda moodust suuremate programmide silumisel kontrollarvutusteks, kuid ta võib osutada otstarbekohaseks ka väiksemate ülesannete vahetul lahendamisel.

Lauaarvuti programmi poole saab pöörduda kas juht- või väljastusprogrammist korraldusega cc. Selle toimet arvuti

trükitab kõigepealt pesa 2045 sisu täisarvuna (see on kä-suloendaja seis enne viimast vajutamist lülitile "Пуск-1") ning tähed cc, millega teatab, et ta on valmis vahetuteks arvutusteks.

Arvutust.ks lauaarvuti režiimis tuleb kasutada erilist keelt, mis on lähedane automaatse programmeerimise keelega (viimase kirjeldus ilmub "Programme kõigile" järgmises väl-jaandes). See keel võimaldab kasutada tähelisi muutujaid, omistada neile väärtusi, arvutada aritmeetiliste avaldiste väärtusi, samuti teisendada nurga kraadimõõtu radiaanmõõ-duks ja vastupidi.

Reaalarvuliste muutujate tähistamiseks võib kasutada tähti π , a, б, b, c, r, g, e, u, л, l, m, ш, н, o, p, ч, s, t, ш, x, y, z ja μ ning "Consuli" variandi korral veel tähti ж, э, й, ъ, ф, ц ja ш. Muutujate algväärtus-teks (kuni esimese omistamiseni) on juhuslikud arvud, ai-nult muutuja π algväärtuseks omistatakse automaatselt 3,14 ... Muutujate väärtuste salvestamiseks kasutab laua-arvuti programm muide samu pesi nagu automaatse program-meerimise süsteem.

Arvulise väärtuse A omistamiseks muutujale x tu-leb teletaibi klaviatuuril trükkida

$$x = A(bk) \quad .$$

Arvud trükitakse sealjuures tavalisel kujul, eraldades täisosas murdosast komaga (näiteks $a = 2,5(bk)$).

Muutuja x väärtuse teadasaamiseks tuleb trükkida

$$x = \underline{\quad} ,$$

mille järele arvuti automaatselt trükkib selle muutuja väärtuse.

Radioaannõodus antud nurga A teisendamiseks kraadimõõtu tuleb trükkida

$$A = \quad ,$$

mille järele arvuti trükkib vastava kraadimõõdu. Vastupidiseks teisendamiseks tuleb trükkida

$$\underline{\quad} A = \quad ,$$

kasutades kraadide, minutite ja sekundite märkimiseks astendajaid 0,1 ja 11 (näiteks $\underline{\quad} 7^{\circ} 10^1 30^{11} =$). Puuduvad sekundid või minutid ja sekundid võib trükkimata jätta (näiteks $\underline{\quad} 3^{\circ} 30^1 =$ või $\underline{\quad} 17^{\circ} =$).

Elmise tehte tulemuse omistamiseks muutuja x väärtuseks tuleb trükkida

$$= x$$

Näiteks muutuja x väärtuse omistamiseks muutujale z tuleb trükkida $x = \underline{\quad}$ ja pärast väärtuse automaatset trükkimist veel

$$= z$$

Aritmeetilise avaldise väärtuse arvutamiseks tuleb trükkida see avaldis ning tema järele võrdusmärk. Leitud väärtuse trükkib arvuti võrdusmärgi järele.

Aritmeetilise avaldise kirjutamisel võib kasutada tehtemärke + (liitmine), - (lahutamine), × (korrutamine), / (jagamine), | või ! (faktoriaal vastavalt "Rioni" või "Consuli" variandi korral), $\sqrt{\quad}$ (ruutjuur) ja $\sqrt[n]{\quad}$ (n-astme juur, kus n tuleb kirjutada arvulise astendajana,

$2 < n < 32$). Astendajana võib kasutada vaid naturaalarvusi (näiteks a^{-2} asemel tuleb kirjutada $(1/a)^2$). Korrutamismärk tuleb kirjutada alati siis, kui teine tegur on arv (muudel juhtudel võib selle märgi ära jätta).

Peale juurte võib avaldistes kasutada veel järgmisi elementaarfunktsioone: \sin , \cos , tg , arcsin , arccos , arctg , \exp , \ln , \lg (kümnendlogaritm), sh (hüperboolne siinus), ch (hüperboolne koosinus), th (hüperboolne tangens) ja ra (gammafunktsioon). Funktsiooni (ka juurte) tunnuseks on tühik tema nimetuse ees, nii näiteks $_ \cos$ tähendab koosinusfunktsiooni, \cos aga muutujate c , o ja s korrutist.

Funktsioonide nimetustest tuleb täielikult trükkida ainult \ln , \lg , sh , ch ja th ; ülejäänutes trükkib arvuti osa tähti automaatselt. Nii jätkub funktsioonide \sin , \cos , tg , \exp ja ra korral vaid esimese tähe trükkimisest, arkusfunktsioonide korral aga tuleb trükkida kõigepealt täht a ja pärast tähtede uc automaatselt trükkimist veel kas s , c või t .

Trigonomeetriliste funktsioonide argument võib olla kas radiaan- või kraadimõõdus, kuid viimasel juhul ainult arvilise väärtusena, millele peab eelnema tühik (näiteks $_ \sin _ 16^{\circ} 0' 30''$).

Näide. Muutja x väärtuse arvutamiseks valemi

$$x = 2 \cdot 10^3 z (\sin^2 e^z - \frac{z^{1,5}}{\sqrt[3]{1,5 \log z}})$$

järgi tuleb trükkida:

$$2 \times 10^3 z ((\sin \exp z)^2 - (\exp 1,5 \ln z) / \sqrt[3]{1,5 \lg z}) = \\ = x$$

4. Silumisprogramm võimaldab kontrollida kordamiste arvu tsüklis (sealjuures kuni kuut tsüklit tsüklis), väljastada andmeid suunamiste kohta ja vahetulemusi. Kasutada saab teda kõigi selliste käskudes koostatud programmide või programmiosade silumisel, mis ei sisalda vahetuid suunamisi püsimälus paiknevatesse programmidesse.

Silumisprogrammi poole pööratakse juhtprogrammist korraldusega on, misjärel arvuti jääb ootama täpsustavaid silumiskorraldusi. Iga silumiskorraldus algab ühega sõnadest *начало, вывести, добавим, ошибка* või *исполним* (trükkida tuleb vaid sõna esitähnt, ülejäänu lisab arvuti automaatselt).

Sõnaga *начало* algav korraldus sisestab silumisplaani (lühidalt SP). Korralduse päis sisaldab kolm rida:

начало(bk)

S(bk)

AконL(bk)

(vahetult tuleb trükkida vaid allakriipsutatud osad), kus S on SP jaoks eraldatava mäluosa esimese pesa aadress, A on silutava programmi alguse ja L selle lõpu aadress. Korralduse edasiseks sisuks on SP, mille iga rida teatab

kontrollimisele kuuluva koha programmis ning mida seal tuleb väljastada. Rida algab selle pesa aadressiga N, milles sisalduva käsu täitmise järel soovitakse informatsiooni väljastada. Väljastatava informatsiooni trükkimisviisi määrab rea lõppu pandav täht (üldkujudes x), milleks võib olla n (liikuva koma arv), u (fikseeritud koma arv), m (täisarv), g (topeltarv), u (indeks), c (kahend-arv) või k (käsk). SP iga rida peab olema vormistatud ühel järgmisest neljast üldkujust:

$$N_A_x(bk)$$

(pesa A sisu trükkimine kujul x),

$$N_x(bk)$$

(fikseeritud aadressiga pesa a_2 sisu trükkimine kujul x),

$$N_r(bk)$$

(pesas N asuva tingimusliku suunamiskäsuga lõppeva tsükli kordamiste arvu trükkimine pärast selle tsükli esmakordset täitmist),

$$N_A_r_M_x(bk)$$

(tsükli kordamiste arvu trükkimine koos pesast A algava M-elementilise massiivi väljastamisega kujul x).

SP lõppu tuleb eraldi reale trükkida kas k või n_k, mis annab juhtimise tagasi silumisprogrammi algusesse (uue korralduse sisestamiseks). Täht n lisatakse siia siis, kui soovitakse silumisel lisaks SP-s loetletule veel kõikide suunamiste trükkimist.

Sõnaga добавим algav korraldus võimaldab silumise käigus sisestada SP uusi ridu äsjakirjeldatud skeemi kohaselt.

Sõnaga вывести algav korraldus sisaldab veel vaid ühe rea kujul

$$N(bk) \quad \text{või} \quad N_{LM}(bk) \quad ,$$

mille toimet trükitakse välja pesa N kohta käivast reast alates kas kogu SP või siis M rida sellest.

Sõnaga ошибка algav korraldus võimaldab kustutada või muuta SP üksikuid ridu. Nimelt iga sellele sõnale järgnev rida

$$N(bk)$$

kustutab pesa N kohta käiva rea, ülaltoodud üldkujul trükitav rida aga asendab varem sama N kohta käinud rea. Paranduste lõppu tuleb jälle trükkida k, mis annab juhtimise silumisprogrammi algusesse.

Sõnast исполним koosnev korraldus alustab programmi silumist.

Lisaks silumisplaanis ja silutavas programmis ettenähtud trükkimistele väljastab silumisprogramm veel informatsiooni ületäitumiste kohta. Nimelt kui ületäitumist analüüsiva käsu ($c_2, c_3, b_2, b_3, y_2, y_3, \xi_2$ või ξ_3) toimet tekkis ületäitumine, siis trükitakse

$$(N) = \infty \quad ,$$

kus N on ületäitumist põhjustanud käsku sisaldava pesa

aadress. Programmis esinevate peatuskäskude puhul ja samuti silutava programmilõigu lõpus trükitakse

(N) = кончаем ,

kus N on peatust põhjistanud käsku sisaldava pesa aadress. Silumise jätkamiseks tuleb nüüd vajutada lülile "Пуск-2".

5. Testprogramm võimaldab kontrollida arvuti korrasolekut mitmesuguste elementaarsete näiteülesannete lahendamise ja tulemuste võrdlemise teel püsivalt salvestatud õigete vastustega. Mittevõrdumise korral arvuti peatub ning peatuse aadressi järgi saab määrata vea ligikaudse asukoha. Ülesanded on valitud nii, et kontrollimine toimuks arvuti funktsionaalsete sõlmede kaupa.

Testprogrammi poole pöördutakse juhtprogrammist korraldusega 06. Kui testprogramm viga ei avasta, siis trükitakse kontrollarvud ja alustatakse uut tsüklit. Ühele tsüklile kulub 2,5 minutit. Arvuti profülaktiliseks kontrolliks piisab tavaliselt kahest tsüklist.

Teenindusprogrammide hulka kuulub veel translaator, kuid selle kasutamist selgitatakse automaatsele programmeerimisele pühendatud väljaandes.

§ 2. PROGRAMMEERIMINE

1. Algoritmi valik sõltub suuremate ülesannete korral eelkõige mälu jaotamise võimalustest. Kui mälu piiratus algoritmi valikut ei kitsenda, siis tuleb valida selline algoritm, mille korral lahendusaeg on minimaalne. Lahendusaega võib oluliselt mõjutada trükitavate tulemuste arv, sest trükkimise kiirus on arvutil suhteliselt väike. Trükkimisskeem tuleb seega koostada selliselt, et trükitaks nii vähe kui võimalik, samuti on loomulik vältida liigsete komajärgsete kohtade trükkimist. "Rioni" variandi korral võib trükkimismahukate ülesannete tulemused väljastada perforiindile (ca 10 korda kiirem kui trükkimine) ning saadud lindi järgi trükkida need tulemused eraldi teletaibil (saab kasutada standardset teletaipi).

Et käskudes koostatakse eelkõige tüüpülesannete programme, siis tuleb algoritmi valikul arvestada ka algandmete perforeerimise lihtsust. Programmi liigset universaalsust tuleb reeglina vältida, sest see pikendab tavaliselt lahendusaega ja raiskab mälu.

Algoritmi koostamisel tuleb arvestada veel püsivõime olemasolevaid alamprogramme, sest nende kasutamisega saab säästa ruumi mälus.

2. Mälu jaotus on programmeerimise esimene etapp, mille käigus tuleb otsustada, kuidas 2048-pesalisse opera-

tiivmällu mahutada programm, programmi tööpesad ning algandmed. Ühtlasi tuleb silmas pidada, et operatiivmälus asuvad veel pseudokäskude ja teenindusprogrammide tööpesad. Mistahes ülesande lahendamisel võib programmi, tööpesade ja algandmete mahutamiseks kasutada pesi 32 - 1940. Kui programmi silumisel ei kavatseta rakendada lauarvuti programmi, siis võib täiendavalt võtta tarvitusele veel pesad 1941 - 2009. Kui ka sellest ruumist ei piisa, siis tuleb igal konkreetsel juhul selgitada, milliseid tööpesi vajavad programmis sisalduvad pseudokäsud ja kasutatavad teenindusprogrammid (vt. lisa 7) ning võtta vastavalt kasutusele veel mõningaid pesi piirkonnast 1 - 31, 2010 - 2047.

Harilikult on tööpesade ja algandmete hulk võrdlemisi täpselt määratav juba enne programmi koostamist. Programmi käskude arv selgub aga alles töö lõpul. Seetõttu on soovitatav kasutada järgmist operatiivmälu jaotamist: alustada programmi kirjutamist pesast 32; algandmed (koos väikese reserviga) paigutada mälu lõppu kuni pesani 1940; tööpesad paigutada algandmete ette, kusjuures andmete lühiajaliseks säilitamiseks võib neid võtta ka piirkonnast 1 - 31, 2010 - 2047 (arvestades lisa 7).

Kui programmi koostamisel selgub, et programm siiski ei mahu mällu, siis väikese puudujäägi saab tavaliselt korvata kas mälu alguse või lõpu pesade ajutise kasutamisega tööpesadena või pisemate muudatustega programmis. Suure puudujäägi korral tuleb kas programm jagada osadeks (kui see on võimalik) või muuta algoritmi nii, et algandmeid

saaks sisestada osade kaupa. Kõige halvemal juhul pole ülesanne arvutil "Nairi-2" üldse lahendatav.

3. Programmi ja algandmete sisestamine (koos teisen-
damisega kahendsüsteemi) toimub tavaliselt juhtprogrammi
abil. Näiteks kui on tarvis sisestada programm alates pe-
sast 30 ja selle reaalarvulised algandmed alates pesast
1250, siis tuleb lihtsalt fotosisendile panna perfolint:

30k (programm) _ m1250n (algandmed) _ m30u

ning vajutada lülitile "Тыск-1". Perforeerimisel tuleb
iga kahe käsu (samuti iga kahe arvu) vahele panna tühik.
Perfolindi algusesse ja lõppu jätta ca 10 cm tühja linti
(40 - 45 tühikut või registrimärki).

Kui ülesannet on kavas lahendada mitme komplekti alg-
andmetega, siis on soovitatav lõpetada programm käsuga
u2368k (mis annab juhtimise juhtprogrammi algusesse) ja
perforeerida algandmete järgmised komplektid samale lindi-
le, jättes iga kahe komplekti vahele 2 - 3 cm tühja ruumi.
Viimase komplekti lõppu pärast korraldust 30u perforeerida
veel peatuskorraldus u .

Kui algandmete perforeerimisel pole teada, kuhu anda
juhtimine pärast sisestamist (näiteks kui sama algandmete
linti on kavas kasutada mitme ülesande puhul), siis tuleb
algandmete lõppu perforeerida myy ja põhiprogrammist
pöörduda juhtprogrammi poole kas käskudega .

$$(1 + 0) = e1k2045$$

$$(1 + 1) = u2368k$$

või käskudega

$$(1 + 0) = n(1+2)k2045$$

$$(1 + 1) = u2368k$$

Mõlemal juhul annab korraldus yy pärast sisestamist juhtimise käsule, mida sisaldava pesa aadress (antud juhul 1 + 2) on pesas 2045.

Kui erinevate ülesannete puhul peame samad algandmed sisestama erinevatesse pesadesse, siis tuleb juhtprogrammi algus vahele jätta, asendades selle põhiprogrammi vastava lõiguga. Näiteks kui viimases näites vaadeldud algandmete lint tuleb sisestada alates pesast 890 ning anda seejärel juhtimine pesas 115 paiknevale käsule, siis saab seda teha põhiprogrammi järgmise lõiguga:

$(1 + 0) = u0n6$	(lugeda sümbol juhtprogrammi tööpesa)
$(1 + 1) = b_49n6$	(kas loetud sümbol on number?)
$(1 + 2) = e_116381n\neq$	(jaataval juhul lugeda järgmine)
$(1 + 3) = n890n1$	(massiivi algusaadress juhtprogrammi tööpesa)
$(1 + 4) = n115k2045$	(juhtprogrammist väljumise aadress)
$(1 + 5) = u2448k$	(jätta vahele juhtprogrammi algus)

Juhtprogrammi üht osa saab kasutada ka liikuva koma arvude sisestamiseks ühekaupa. Selleks tuleb põhiprogrammi lülitada lõik:

$(1 + 0) = n0k28$	(positiivse arvu tunnus tööpessa)
$(1 + 1) = 40k22$	(lugeda sümbol tööpessa)
$(1 + 2) = e_22050n5$	(kas sümbol on (bk)?)
$(1 + 3) = e_22052n4$	(kas sümbol on $_$?)
$(1 + 4) = e_22074n4$	(kas sümbol on -?)
$(1 + 5) = b_49n22$	(kas sümbol on number?)
$(1 + 6) = e_13n\neq$	(jaataval juhul $1+10 \rightarrow KL$)
$(1 + 7) = k22n$	(stop! vale sümbol)
$(1 + 8) = e_116376n$	(tagasi sümboli lugemise juurde)
$(1 + 9) = n1k28$	(negatiivse arvu tunnus tööpessa)
$(1 + 10) = u_13153n2$	(täita osa juhtprogrammist)

Juhtprogrammi vastav osa loeb perfolindilt liikuva koma arvu, teisendab selle kahendsüsteemi ja salvestab pessa 26. Ühtlasi loetakse veel arvule järgnev sümbol pessa 25 (vigade vältimiseks on soovitatav kontrollida, kas selleks ikka osutub (bk) või $_$).

4. Lihtsamaid programmeerimisvõtteid. Arvude massiivi übersalvestamine, näiteks pesadest 500 - 517 pesadesse 600 - 617 on teostatav programmilõiguga:

$(1 + 0) = n0k1$	(indeksi algväärtuseks null)
$(1 + 1) = n500n600+$	(übersalvestamine)
$(1 + 2) = c2049n1$	(indeksi suurendamine $2^{11} + 2^0$ võrra)
$(1 + 3) = b_134833n$	(kui indeks $\leq 17 \cdot (2^{11} + 2^0)$, siis
$(1 + 4) = e_116380n\neq$	tagasi übersalvestamisele)

Käskude (1 + 2) ja (1 + 3) esimeste aadresside leidmisel on siin kasutatud tabelit 5. Käsu (1 + 4) asemele võiks kirjutada ka käsu $u(1+1)n \leq$, kuid sel juhul programm sõltuks oma asukohast mälus.

Pesades 1700 - 1900 paiknevate liikuva koma arvude hulgast kõigi positiivsete summa leiab pessa 1200 programmilõik:

(1 + 0) =	$n0k1200$	(summa algväärtuseks null)
(1 + 1) =	$n1700n1$	(indeksi algväärtuseks 1700)
(1 + 2) =	$c0k0+$	(kui järjekordne arv ≤ 0 , siis jätta liitmine vahele)
(1 + 3) =	$e_1 1n \leq$	
(1 + 4) =	$cc1200k1200$	(positiivse arvu liitmine)
(1 + 5) =	$c1k1$	(indeksi suurendamine ühe võrra)
(1 + 6) =	$b_1 1900n$	(kui indeks ≤ 1900 , siis tsükli kordamine)
(1 + 7) =	$e_1 16378n \leq$	

Ka siin võiks käsud (1 + 3) ja (1 + 7) asendada vastavalt käskudega $u(1+5)n \leq$ ja $u(1+2)n \leq$, mis annaks asukohast sõltuva programmi.

Pesades 1000 ja 1001 paikneva topeltarvu kümnenda astme trükkib seitsme kohaga koma järel välja programmilõik:

(1 + 0) =	$om1000k12$	(übersalvestamine tööpesa)
(1 + 1) =	$n9k15$	(loendaja algseisuks 9)
(1 + 2) =	$yg1000k12$	(järjekordne korrutamine)
(1 + 3) =	$b1k15$	(loendaja vähendamine)
(1 + 4) =	$e_1 16381n >$	(sama, mis $u(1+2)n >$)
(1 + 5) =	$ng12k7$	(trükkimine)

Järgnev programmilõik loeb perfolindilt ühe sümboli, kontrollib kas see on täht a ja vastavalt kontrolli tulemusele kas annab juhtimise pesas 89 paiknevale käsule või peatab arvuti (tuues loetud koodi summaatorisse):

- (1 + 0) = 40k2 (sümboli lugemine)
(1 + 1) = m₁2103H (võrdlemine a-ga)
(1 + 2) = u89n= (juhtimise üleandmine)
(1 + 3) = k2H (arvuti peatamine)

"Consuli" kasutamisel peab käsk (1 + 1) olema kujul m₁2058H . Mõlema variandi korral võib kasutada samas kohas käsku m₁522n , sest arv 522 on "Rioni" variandi puhul pesas 2103 ja "Consuli" variandi puhul pesas 2058 (vt. lisa 5).

Järgnevate käskudega trükitakse sümbolid c= (vt. lisa 6) ja nende järele liikuva koma arv pesast 33 (kolme kohaga pärast koma):

- (1 + 0) = o2212H (c trükkimine)
(1 + 1) = o2250H (= trükkimine)
(1 + 2) = nn33H3 (arvu trükkimine)

Pesas 170 paikneva liikuva koma arvu täisosa saab pessa 180 käskudega:

- (1 + 0) = gH170H10
(1 + 1) = n10H180 ,

pesas 175 paikneva täisarvu teisendamine liikuva koma arvuks toimub aga käskudega:

$$(1 + 0) = n175\pi10$$

$$(1 + 1) = n0k11$$

$$(1 + 2) = \text{hg}10k175$$

5. Alamprogramm on suhteliselt iseseisev programmi-osa, mida tarbekorral saab lülitada mistahes põhiprogrammi mistahes kohta. Alamprogrammidenä on otstarbekohane vormistada näiteks programmid mitmesuguste sageliesinevate funktsioonide arvutamiseks, tüüpiliste trükkimis skeemide realiseerimiseks ning üldse n.ö. tüüpülesannete lahendamiseks. Selliselt kujuneb aegamööda alamprogrammide kogu, mis tunduvalt lihtsustab uute programmide koostamist.

Selleks, et alamprogrammi poole saaks pöörduda põhiprogrammi erinevatest kohtadest, peab olema teada alamprogrammi esimest käsku sisaldava pesa aadress A_1 ja selle pesa aadress A_2 , kuhu alamprogramm oma töö lõpul annab juhtimise (alamprogrammi väljundpesa). Pöördumine toimub sel juhul käsuga $u_1 A_1 k A_2$, mis kindlustab pärast alamprogrammi täitmist tagasipöördumise põhiprogrammi järgmise käsu juurde. Universaalse väljundpesana kasutatakse alamprogrammides sageli pesa 31.

Selleks, et alamprogrammi oleks lihtne kasutada erinevates põhiprogrammides, tuleb ta koostada võimalikult sõltumatuna oma asukohast mälus (et uude põhiprogrammi lülitamisel poleks tarvis muuta alamprogrammi käskude aadresse). See saavutatakse eeskätt harilike suunamiskäskude (u) asen-

damise teel suhteliste suunamistega (e_1), samuti vajalike konstantide esitamisega käskude aadressides (kasutades käskude modifikatsioone k , n ja λ). Programmi niisugust muutmist alamprogrammiks illustreerib järgmine näide.

Suuruse

$$c = 0,75 + \begin{cases} -4,78a, & \text{kui } b \geq 0 \\ a/10, & \text{kui } b < 0 \end{cases}$$

arvutamiseks pesa 1830 (eeldades, et a ja b paiknevad vastavalt pesades 1830 ja 1900) ning juhtimise andmiseks pesa 31 võib kasutada programmilõiku:

$$\begin{aligned} (1 + 0) &= n1900H && \text{(jätkata } b \geq 0 \text{ korral} \\ (1 + 1) &= u(1+5)n \geq && \text{pesast } 1 + 5) \\ (1 + 2) &= og10k1830 && (c = a/10) \\ (1 + 3) &= cn(1+8)n1830 && (c = c + 0,75) \\ (1 + 4) &= u31k && \text{(väljumine)} \\ (1 + 5) &= yn(1+7)n1830 && (c = -4,78a) \\ (1 + 6) &= u(1+3)k && \text{(jätkata pesast } 1 + 3) \\ (1 + 7) &= -4,78 && \\ (1 + 8) &= 0,75 && \left. \vphantom{\begin{matrix} (1 + 7) \\ (1 + 8) \end{matrix}} \right\} \text{(konstandid)} \end{aligned}$$

Et siin neli aadressi sõltuvad programmi asukohast mälus, siis alamprogrammina pole teda otstarbekohane kasutada. Vastavaid muutusi tehes saame veidi aeglasemalt töötava, kuid asukohast täiesti sõltumatu programmi:

n1900H	}	(jätta $b \geq 0$ korrall 4 käsku vahele)
e ₁ 4n \geq		
og10k1830		(c = a/10)
n49152A		(0,75 summaatorisse)
cc1830H1830		(c = c + 0,75)
u31k		(väljumine)
on478k	}	(-4,78 summaatorisse)
gc100k		
yc1830H1830		(c = -4,78a)
e ₁ 16377k		(jätkata 6 käsku eestpoolt)

Tingimuslikes modifikatsioonides peituvaid võimalusi illustreerib sama programmi kolmas, eelmistest lühem variant:

on478k	}	(-4,78 \rightarrow 4)
gc100k4		
n1900H3		(b \rightarrow 3)
n49152A2		(0,75 \rightarrow 2)
n1830H		(a \rightarrow a ₂)
gc10n \geq 3		(kui $b \geq 0$, siis a/10 \rightarrow a ₂)
yc4H < 3		(kui $b < 0$, siis -4,78a \rightarrow a ₂)
cc2H1830		((a ₂) + 0,75 \rightarrow c)
u31k		(väljumine)

Kui konstantide moodustamine niisuguste võtetega ei osutu otstarbekohaseks (näiteks kui sama konstanti tuleb tarvis korduvalt) siis võib kasutada järgmist moodust. Alamprogrammi esimeste käskudega

$$(1 + 0) = e0k$$

$$(1 + 1 \rightarrow a_2)$$

$$(1 + 1) = 0,11k1$$

$$(2^{11} \cdot (a_2) \rightarrow 1)$$

moodustatakse pessa 1 konstant, milles käsu esimese aadressi kohal (kahendkohtadel 25 - 12) on arv 1 + 1, mujal aga nullid. Kui nüüd hiljem on tarvis näiteks liita pesas 1830 olev arv pesas 1 + 21 oleva konstandiga, siis võib seda teha käsuga

cn20w1830+

Peale vahetult koostatavate alamprogrammide saab sageli ära kasutada ka arvuti programm-varustuse hulka kuuluvaid. Need püsivalt paiknevad alamprogrammid on määratud kõige sagedamini vaja minevate tüüpülesannete lahendamiseks. Sisuliselt kuuluvad siia ka pseudokäskude alamprogrammid ja varem vaadeldud teenindusprogrammid. Käskudes programmeerimisel pakuvad siiski peamist huvi vast ainult järgmised alamprogrammid.

Käsuga u₁11103k31 võib pöörduda liikuva koma arvu lugemise ja teisendamise alamprogrammi. Loetav arv peab sealjuures olema esitatud nii, nagu seda nõuab juhtprogramm liikuva koma arvude sisestamisel. Arv salvestatakse pessa, mille aadress on antud pesas 1. Vea korral korratakse lugemist. Alamprogrammi tööpesadeks on 1 - 6, 16, 17, 22, 24 - 26, 28, 31, 2010 ja 2011.

Käsuga u₁14521k2 pöördatakse täisarvu lugemise ja teisendamise alamprogrammi. Arv salvestatakse pessa 2012, pessa 2014 salvestatakse aga 0 või 1 vastavalt selle-

le, kas arvu lõpus oli $_$ või (bk). Vea korral trükitakse * ja korratakse lugemist. Tööpesad on 2, 2012 ja 2014.

Käsuga $u_112008k68$ pöördatakse naturaalarvu lugemise ja teisendamise alamprogrammi. Arvu ees ei tohi olla midagi liigset, arvu lõpus peab olema üks tühik. Arv salvestatakse pesa 3. Vea korral trükitakse ∞ ja antakse juhtimine käsule, mida sisaldava pesa aadress on pesas 66. Tööpesad on 3, 4, 7, 66 ja 68.

Käsuga $u_13810k2001$ trükitakse pesa 4 sisu kaheksand-süsteemis. Tööpesad on 4 ja 2001.

Käsuga $u_111800k9$ perforeeritakse kaheksandarvudena massiiv, mille algusaadress ja elementide arv on täisarvudena antud vastavalt pesades 1 ja 68. Massiivi ette perforeeritakse algusaadress kümnendsüsteemis ja täht b. Arvu lõpu tunnuseks on tühik. Kui mõnes pesas on null, siis perforeeritakse lühidalt 0 $_$. Kui algusaadressi ja tähe b väljastamist ei vajata, siis tuleb pöörduda käsuga $u_111802k9$. Tööpesad on 1, 5, 6, 9 ja 68.

Käsuga $u_110583k66$ pöördatakse alamprogrammi, mis trükib $_cm_primev$ ja pesa 26 sisu täisarvuna. Tööpesad on 26 ja 66.

Käsuga $u_110597k26$ pöördatakse alamprogrammi, mis trükib pesas 3 näidatud arvu tühikuid. Tööpesad on 3 ja 26.

Käsuga $u_111276k31$ pöördatakse reavahede trükkimise alamprogrammi. Trükitakse pesas 11 näidatud arv reavahe-sid. Tööpesad on 11 ja 31.

Käsuga $u_111488k31$ pöördatakse alamprogrammi, mis trü-

kib N punkti järgi ühe või kahe funktsiooni graafiku. Pessa 10 tuleb eelnevalt saata täisarvuna vastavalt kas 1 või 2 ja pessa 1997 kas N või 2N. Funktsiooni väärtuste massiivi algusaadress peab olema saadetud pessa 64. Kahe graafiku trükkimisel peavad mõlema funktsiooni väärtused paiknema vaheldumisi. Tööpesad on 1, 3, 10, 11, 26, 31, 64, 341 - 350, 1996 ja 1997.

6. Silumine ja vormistamine. Pärast valminud programmi hoolikat kontrollimist blokk-skeemi järgi (erilist tähelepanu pöörata tingimusi kontrollivatele blokkidele ja suunamistele!) tuleb koostada silumisülesanded programmi üksikosade kontrollimiseks. Enne esimest silumist kontrollida veel tingimata programmi perforeerimise õigsust ja lasta siis programmil töötada üksikosade (blokkide) kaupa, kasutades peatust aadressi järgi ning trükkides väljastusprogrammiga vajalike tööpesade sisud.

Kui on saavutatud üksikosade normaalne funktsioneerimine, siis tuleb koostada silumisülesanded terve programmi jaoks, kasutades selleks näiteks lähteülesande mitmesuguseid piirjuhte, kus õiged vastused on kas teada või vähemalt hinnatavad.

Pärast silumist ja paari-kolme tegeliku ülesande lahendamist tuleb programm vormistada. Kui aja jooksul tekib vajadus programmi mõningate osade muutmiseks, siis tuleb need muudatused eraldi vormistada ja lisada põhiprogrammi kirjeldusele.

Programmi vormistamise eesmärgiks on muuta ta tarvita-
miskõlblikuks ka teistele kasutajatele. Vormistus peab
koosnema järgmisest kuuest osast:

- 1° Probleemiseade.
- 2° Algoritmi kirjeldus koos blokk-skeemiga.
- 3° Programm.
- 4° Mälujaotus.
- 5° Lahendamise juhend.
- 6° Kontrollülesanne koos vastustega.

Probleemiseade peab sisaldama ülesande üksikasjaliku kirjelduse koos kõikide täiendavate nõuete näitamisega. Algoritmi kirjeldus esitatakse tavaliselt blokk-skeemi kujul, kusjuures tuleb ära näidata, milliste käskudega on iga blokk realiseeritud. Keeruliste ülesannete korral näidata ka kirjandus, kust algoritm pärineb. Programm peab olema kirjutatud loetavalt blankettidele või trükitud väljastusprogrammiga (soovitav samaaegselt kaheksandsüsteemis ja sisendkeeles). Mälujaotuses tuleb näidata programmi, algandmete ja põhiliste tööpesade asukohad. Lahendamise juhendis peab olema kirjas programmi ja algandmete sisestamise kord (koos algandmete perforeerimise üksikasjalise skeemiga). Samuti tuleb näidata võimalused programmi katkestamiseks ja jätkamiseks. Kui programmis kasutatakse muudetavaid parameetreid (näiteks täpsuse konstant), siis selgitada nende valiku printsiipe. Programmis esineda võivad avariipeatused on sobiv esitada tabelina, kusjuures peatuse momendil käsuloendajas olev arv esitada

ka kaheksandsüsteemis. Iga avariipeatuse kohta kirjutada, millest see on tingitud ja mida teha edasi. Kui tulemuste põhjal on võimalik vahetult hinnata lahendamise õigsust, siis anda ka kontrollimise eeskiri. Kindlasti peab olema näidatud lülitite "Ключ" ja "Вармант" asend ning lahendamise orienteeruv aeg. Kontrollülesandena esitatakse tavalliselt silumisülesanne koos algandmete ja vastustega.

L I S A D

Lisa 1. "Rioni" koodide tabel.

Kood	Lindil					S ü m b o l			
	5	4	m	3	2	1	Ü	K	A
00			•				Alumine register		
01			•			•	g	5	×
02			•		•		Kelk		
03			•		•	•	ast 7	9	ind 7
04			•	•			Tühik		
05			•	•		•	∞	u	≠
06			•	•	•		i	(l
07			•	•	•	•	n	x	<
10		•	•				Rida		
11		•	•			•	ast 8	c	ind 8
12		•	•		•		ast 6	4	ind 6
13		•	•		•	•	j	e	t
14		•	•	•			u	8	≤
15		•	•	•		•	ast 5	0	ind 5
16		•	•	•	•		л	b	ind j
17		•	•	•	•	•	ast 0	n	ind 0

Lisa 1 (järg).

Kood	Lindil					S ü m b o l			
	5	4	m	3	2	1	Ü	K	A
20	●		●				s	3	z
21	●		●			●	6	k	√
22	●		●		●		ast 9	o	ind 9
23	●		●		●	●	=	,	
24	●		●	●			y	m	≥
25	●		●	●		●	ast 2	6	ind 2
26	●		●	●	●		ast 1	+	ind 1
27	●		●	●	●	●	a	/	*
30	●	●	●				ast 4	p	ind 4
31	●	●	●			●	ast 3	2	ind 3
32	●	●	●		●		r	-	ind i
33	●	●	●		●	●	Keskmine register		
34	●	●	●	●			m	7	ind k
35	●	●	●	●		●	n	1	>
36	●	●	●	●	●		w)	π
37	●	●	●	●	●	●	Ülemine register		

Tähed Ü, K ja A tähistavad vastavalt ülemist, keskmist ja alumist registrit. Tavalised numbrid kuuluvad keskmisse, astendajad (ast) ülemisse ja indeksid (ind) alumisse registrisse.

Lisa 2. "Consuli" koodide tabel.

Kood	Lindil							Sümbol	
	7	6	5	4	3	2	1	Ü	A
00	●				●			0	ast 0
01					●		●	1	ast 1
02					●	●		2	ast 2
03	●				●	●	●	3	ast 3
04					●	●		4	ast 4
05	●				●	●	●	5	ast 5
06	●				●	●	●	6	ast 6
07					●	●	●	7	ast 7
10				●	●			8	ast 8
11	●			●	●		●	9	ast 9
12	●			●	●		●	a	'
13				●	●		●	б	'
14	●			●	●	●		g	ind j
20			●	●				Kelk+rida	
21	●		●	●			●	Rida	
22	●		●	●		●		Punane lint	
23			●	●		●	●	Must lint	
24	●		●	●		●		Ülem.reg.	
26			●	●		●	●	Alum.reg.	
27	●		●	●		●	●	Tühik	
30	●		●	●	●			b	*
31			●	●	●		●	ь	ind 9
40		●			●			=	≠
41	●	●			●		●	й	ind 7
42	●	●			●		●	ш	<
43		●			●		●	+	j
44	●	●			●			н	л
45		●			●		●)	∞
46		●			●		●	e	ind k
47	●	●			●		●	/	l

Lisa 2 (järg).

Kood	Lindil							Sümbol	
	7	6	5	4	3	2	1	Ü	A
50	•	•		•	•			p	ind 4
51		•		•	•		•	x	!
52		•		•	•		•	(!
53	•	•		•	•		•	o	f
54		•		•	•	•		k	ind 3
55	•	•		•	•	•	•	,	!
56	•	•		•	•	•	•	n	ä
57		•		•	•	•	•	φ	в
60	•	•	•		•			y	э
61		•	•		•		•	z	ind 2
62		•	•		•		•	c	ind 5
63	•	•	•		•		•	ц	≠
64		•	•		•	•		ч	≥
65	•	•	•		•	•	•	√	z
66	•	•	•		•	•	•	m	щ
67		•	•		•	•	•	s	ind 1
70		•	•	•	•			-	s
71	•	•	•	•	•		•	x	t
72	•	•	•	•	•		•	m	>
73		•	•	•	•		•	u	ind 6
74	•	•	•	•	•	•		r	ind 1
75		•	•	•	•	•	•	m	ind 8
76		•	•	•	•	•	•	x	ind 0

Tähed Ü ja A tähendavad vastavalt ülemist ja alumist registrit. Tavalised numbrid kuuluvad ülemisse, astendajad (ast) ja indeksid (ind) aga alumisse registrisse.

Lisa 4. Käskude süsteem.

I. Aritmeetilised tehted fikseeritud koma arvudega.

Tehte kood		Tehte nimetus	Tehte sisu	
01	c	Liitmine	$(A_2) + [A_1] \rightarrow A_2, a_2$	
11	c ₂	Liitmine	$(A_2) + [A_1] \rightarrow A_2, a_2$	1
41	c ₁	Liitmine	$(a_2) + [A_1] \rightarrow A_2, a_2$	
51	c ₃	Liitmine	$(a_2) + [A_1] \rightarrow A_2, a_2$	1
02	b	Lahutamine	$(A_2) - [A_1] \rightarrow A_2, a_2$	
12	b ₂	Lahutamine	$(A_2) - [A_1] \rightarrow A_2, a_2$	1
42	b ₁	Lahutamine	$(a_2) - [A_1] \rightarrow A_2, a_2$	
52	b ₃	Lahutamine	$(a_2) - [A_1] \rightarrow A_2, a_2$	1
03	y	Korrutamine	$(A_2) \times [A_1] \rightarrow A_2, a_2$	
13	y ₂	Korrutamine	$(A_2) \times [A_1] \rightarrow A_2, a_2$	1
43	y ₁	Korrutamine	$(a_2) \times [A_1] \rightarrow A_2, a_2$	
53	y ₃	Korrutamine	$(a_2) \times [A_1] \rightarrow A_2, a_2$	1
04	g	Jagamine	$(A_2) : [A_1] \rightarrow A_2, a_2$	
14	g ₂	Jagamine	$(A_2) : [A_1] \rightarrow A_2, a_2$	1
44	g ₁	Jagamine	$(a_2) : [A_1] \rightarrow A_2, a_2$	
54	g ₃	Jagamine	$(a_2) : [A_1] \rightarrow A_2, a_2$	1
26	b ₄	Moodulite lahutamine salvestuseta	$ (A_2) - [A_1] \rightarrow a_2$	

1. Ületäitumise korral (KL) \rightarrow 2046 ja 2047 \rightarrow KL.

II. Aritmeetilised tehted liikuva koma arvudega.

Tehte kood		Tehte nimetus	Tehte sisu
101	cn	Liitmine	$[A_1] + (A_2) \rightarrow A_2, a_2$
145	c6	Liitmine	$[A_1] + (A_2) \rightarrow a_2$
146	cc	Liitmine	$(a_2) + [A_1] \rightarrow A_2, a_2$
102	bn	Lahutamine	$[A_1] - (A_2) \rightarrow A_2, a_2$
143	ob	Lahutamine	$(A_2) - [A_1] \rightarrow A_2, a_2$
147	b6	Lahutamine	$[A_1] - (A_2) \rightarrow a_2$
150	bc	Lahutamine	$(a_2) - [A_1] \rightarrow A_2, a_2$
103	yn	Korrutamine	$[A_1] \times (A_2) \rightarrow A_2, a_2$
151	y6	Korrutamine	$[A_1] \times (A_2) \rightarrow a_2$
152	yc	Korrutamine	$(a_2) \times [A_1] \rightarrow A_2, a_2$
104	gn	Jagamine	$[A_1] : (A_2) \rightarrow A_2, a_2$
121	og	Jagamine	$(A_2) : [A_1] \rightarrow A_2, a_2$
153	g6	Jagamine	$[A_1] : (A_2) \rightarrow a_2$
154	gc	Jagamine	$(a_2) : [A_1] \rightarrow A_2, a_2$
122	bm	Moodulite lahutamine	$ (A_2) - [A_1] \rightarrow A_2, a_2$
155	on	Vastandarvu ülekanne	$- [A_1] \rightarrow A_2, a_2$

Esimene operand saadakse $[A_1]$ teisendamisel liikuva koma arvuks. Ületäitumise korral trükitakse ∞ ja peatatakse arvuti.

Lisa 4 (järg).

III. Aritmeetilised tehted täisarvudega.

Tehte kood	Tehte nimetus	Tehte sisu
100 ym	Korrutamine	$(A_2) \times [A_1] \rightarrow A_2$
144 gm	Jagamine	$(A_2) : [A_1] \rightarrow A_2$ Murdosa $\rightarrow 3$

IV. Aritmeetilised tehted topeltarvudega.

Tehte kood	Tehte nimetus	Tehte sisu
112 cg	Liitmine	$(A_2)^P + [A_1]^P \rightarrow A_2^P$
113 bg	Lahutamine	$(A_2)^P - [A_1]^P \rightarrow A_2^P$
114 yg	Korrutamine	$(A_2)^P \times [A_1]^P \rightarrow A_2^P$
115 gg	Jagamine	$(A_2)^P : [A_1]^P \rightarrow A_2^P$

V. Aritmeetilised tehted kompleksarvudega.

Tehte kood	Tehte nimetus	Tehte sisu
135 ck	Liitmine	$(A_2)^P + [A_1]^P \rightarrow A_2^P$
136 bk	Lahutamine	$(A_2)^P - [A_1]^P \rightarrow A_2^P$
137 yk	Korrutamine	$(A_2)^P \times [A_1]^P \rightarrow A_2^P$
140 gk	Jagamine	$(A_2)^P : [A_1]^P \rightarrow A_2^P$

Ületäitumise korral trükitakse ∞ ja peatatakse arvuti. Viimases kahes rühmas kasutatakse põhiliselt modifikatsioone k ja n.

VI. Loogilised tehted.

Tehte kood		Tehte nimetus	Tehte sisu
15	1	Disjunktsioon (formeerimine)	$(A_2) \vee [A_1] \rightarrow a_2$
55	1_1		$(a_2) \vee [A_1] \rightarrow A_2, a_2$
16	π	Konjunktsioon (osa eraldamine)	$(A_2) \wedge [A_1] \rightarrow a_2$
56	π_1		$(a_2) \wedge [A_1] \rightarrow A_2, a_2$
17	μ	Mitteekvivalents (võrdlemine)	$(A_2) \sim [A_1] \rightarrow a_2$
57	μ_1		$(a_2) \sim [A_1] \rightarrow A_2, a_2$

VII. Nihutamised.

Tehte kood		Tehte nimetus	Tehte sisu
05	a	Aritmeetiline nihe paremale	$\overrightarrow{(A_2)} \rightarrow A_2, a_2$
45	a_1		$\overrightarrow{(a_2)} \rightarrow A_2, a_2$
06	b	Loogiline nihe vasakule	$\overleftarrow{(A_2)} \rightarrow A_2, a_2$
46	b_1		$\overleftarrow{(a_2)} \rightarrow A_2, a_2$
23	r	Topeltarvu aritmeetiline nihe paremale	$\overrightarrow{(A_2)}^P \rightarrow A_2^P$
63	r_1	Topeltarvu loogiline nihe vasakule	$\overleftarrow{(A_2)}^P \rightarrow A_2^P$

Aritmeetilise nihke korral jääb märgikoht paigale. Vabanevatele kohtadele kantakse positiivse arvu korral nullid, negatiivse arvu korral ühed. Nihe toimub kõikjal $[A_1]^2$ koha võrra.

VIII. Übersalvestamised.

Tehte kood	Tehte nimetus	Tehte sisu
10 n	Übersalvestamine I	$[A_1] \rightarrow A_2, a_2$
50 n_1	Übersalvestamine II	$(a_2) \rightarrow A_2, a_2$
142 om	Paarispesa übersalvestamine	$[A_1]^P \rightarrow A_2^P$

1. Esimesel aadressil pole mingit tähendust.

IX. Suunamised.

Tehte kood	Tehte nimetus	Tehte sisu
20 u	Lihtsuunamine	$[A_1]^{14} \rightarrow KL$
60 u_1	Suunamine tagasipöördumisega	Käsk $u(KL)n \rightarrow A_2$ $[A_1]^{14} \rightarrow KL$
40 u_2	Erisuunamine	$[A_1]^{14} \rightarrow KL$
25 u_3	Võtmeaga suunamine	$[A_1]^{14} \rightarrow KL$, kui lülititi "Ключ" on sees, vastasel juhul $(KL) + 1 \rightarrow KL$
62 e_1	Suhteline suunamine tagasipöördumisega	Käsk $u(KL)n \rightarrow A_2$ $[A_1]^{14} + (KL) \rightarrow KL$
64 e_2	Tingimuslik suhteline suunamine I	Kui $(a_2) = [A_1]$, siis $(KL) + A_2 \rightarrow KL$, vastasel juhul $(KL) \rightarrow KL$
65 e_3	Tingimuslik suhteline suunamine II	Kui $(a_2) \wedge [A_1] = [A_1]$, siis $(KL) + A_2 \rightarrow KL$, vastasel juhul $(KL) \rightarrow KL$

Käsuloadajas asub käsu täitmise ajal juba järgmist käsku sisaldava pesa aadress. Suunamiskäsud ei muuda pesa (a_2) sisu. Käsku u_2 kasutatakse ainult pseudokäsu alamprogrammi lõpukäsuna. Käskudes e_2 ja e_3 kasutatakse teist aadressi parameetrina.

X. Sisestamised ja väljastamised.

Tehte kood	Tehte nimetus	Tehte sisu	
24 v	Sümboli sisestamine	$((VR) + 2048) \rightarrow A_2, a_2$	
21 o	Sümboli väljastamine	$[A_1]^{12} \rightarrow VR$	1
61 o ₁	Sümboli väljastamine	$((a_2) + 2176)^{12} \rightarrow VR$	1
127 nv	Fikseeritud koma arvu trükk	$[A_1]$ trükitakse A_2 kohaga koma järel	2
110 nn	Liikuva koma arvu trükk	$[A_1]$ trükitakse A_2 kohaga koma järel	2
130 nm	Täisarvu trükk	$[A_1]$ trükk	
117 ng	Topeltarvu trükk	$[A_1]^P$ trükitakse A_2 kohaga koma järel	2
161 nw	Kompleksarvu trükk	$[A_1]^P$ trükitakse A_2 kohaga koma järel	2
132 nk	Käsu trükk	$[A_1]$ trükk	
141 nu	Indeksi trükk	$[A_1]$ kui täisarvu trükk	3
131 nc	Kahendarvu trükk	$[A_1]$ trükk	

1. Sümboli väljastuskäsud ei muuda pesa a_2 sisu.
2. Teist aadressi kasutatakse parameetrina.
3. Kui $[A_1] \geq 1000$, siis trükitakse nu ja peatatakse arvuti. TRÜ Arvutuskeskuse arvutil "Nairi-2" saab indeksitena trükkida kõiki täisarve.

XI. Funktsioonide arvutused.

Tehte kood	Tehte nimetus	Tehte sisu	Avariipeatus	
105	kn	\sqrt{x}	$x < 0$	
133	ex	e^x	$x > 43$	
107	ln	$\ln x$	$x \leq 0$	
116	lg	$\log x$	$x \leq 0$	
111	sn	$\sin x$	$x > 2^{27}$	
126	cs	$\cos x$	$x > 2^{27}$	
120	tg	$\tan x$	$x = \pm \frac{\pi}{2}$	
123	as	$\arcsin x$	$x > 1$	
124	ac	$\arccos x$	$x > 1$	
125	at	$\arctan x$		
156	sh	$\operatorname{sh} x$		
157	ch	$\operatorname{ch} x$		
160	th	$\operatorname{th} x$		
162	6j	Besseli funktsioon $I_p(x)$	$I[A_1]((A_2)) \rightarrow A_2$	1
163	6y	Besseli funktsioon $Y_p(x)$	$Y[A_1]((A_2)) \rightarrow A_2$	2
166	ra	Gamma-funktsioon $\Gamma(x)$	$\Gamma([A_1]) \rightarrow A_2$	

1. Kui p on täisarv, siis peab olema $|x| \leq 750$; kui aga p pole täisarv, siis $0 < x \leq 750$.

2. Indeks p suvaline; $0 < x \leq 750$.

XII. Muud käsud.

Tehte kood	Tehte nimetus	Tehte sisu
00 x	Tühikäsk	$(KL) \rightarrow KL$
37 k	Peatus	$[A_1] \rightarrow$ summaator
77 k_1	Peatus	$(a_2) \rightarrow$ summaator
07 n	Normaliseerimine	$[A_1]$ norm. $\rightarrow A_2$
47 n_1	Normaliseerimine	(a_2) norm. $\rightarrow A_2$
22 e	Käsu loendaja salvestamine	$(KL) + [A_1] \rightarrow A_2, a_2$
106 n g	Topeltarvu teisendamine liikuva koma arvukuks	$[A_1]^p \rightarrow A_2$
134 g n	Liikuva koma arvu teisendamine topeltarvukuks	$[A_1] \rightarrow A_2^p$
167 mn	Minimaalse elemendi leidmine	Min(max) element arvudest $([A_1]), ([A_1] + 1), \dots$ $\dots, (A_2) \rightarrow a_2$
170 mx	Maksimaalse elemendi leidmine	

Lisa 5. Sümbolite sisemised koodid.

I. Arvud.

	Sisemine kood	P e s a	
		"Rioni"	"Consul"
0	000 000 000 000	2061	2048
1	000 000 000 001	2077	2049
2	000 000 000 002	2073	2050
3	000 000 000 003	2064	2051
4	000 000 000 004	2058	2052
5	000 000 000 005	2049	2053
6	000 000 000 006	2069	2054
7	000 000 000 007	2076	2055
8	000 000 000 010	2060	2056
9	000 000 000 011	2051	2057

II. Astendajad.

	Sisemine kood	P e s a	
		"Rioni"	"Consul"
0	000 000 002 000	2095	2112
1	000 000 002 001	2102	2113
2	000 000 002 002	2101	2114
3	000 000 002 003	2105	2115
4	000 000 002 004	2104	2116
5	000 000 002 005	2093	2117
6	000 000 002 006	2090	2118
7	000 000 002 007	2083	2119
8	000 000 002 010	2089	2120
9	000 000 002 011	2098	2121
‘	000 000 002 012	—	2122
/	000 000 002 013	—	2123

III. Indeksid.

	Sisemine kood	P e s a	
		"Rioni"	"Consul"
0	000 000 004 000	2127	2174
1	000 000 004 001	2134	2167
2	000 000 004 002	2133	2161
3	000 000 004 003	2137	2156
4	000 000 004 004	2136	2152
5	000 000 004 005	2125	2162
6	000 000 004 006	2122	2171
7	000 000 004 007	2115	2145
8	000 000 004 010	2121	2173
9	000 000 004 011	2130	2137
l	000 000 010 034	2138	2172
J	000 000 010 035	2126	2124
k	000 000 010 036	2140	2150

IV. Tähed.

	Sisemine kood	P e s a		
		"Rioni"	"Consul"	
Л С Б У Е Д М З Н Х А Б Г Ч О Е Р Ш Ш С Т М	000 000 001 000	2142	2148	
	000 000 001 001	2057	2098	
	000 000 001 002	2062	2072	
	000 000 001 003	2100	2096	
	000 000 001 004	2081	2060	
	000 000 001 005	2094	2097	
	000 000 001 006	2108	2102	
	000 000 001 007	2128	2165	
	000 000 001 010	2109	2084	
	000 000 001 011	2055	2089	
	000 000 001 012	2103	2058	
	000 000 001 013	2097	2059	
	000 000 001 014	2106	2108	
	000 000 001 015	2092	2100	
	000 000 001 016	2053	2107	
	000 000 001 017	2066	2091	
	000 000 001 020	2059	2086	
	000 000 001 021	2118	2151	
	000 000 001 022	2072	2088	
	000 000 001 023	2068	2106	
	000 000 001 024	2110	2082	
	000 000 001 025	2096	2168	
	000 000 001 026	2123	2169	
	000 000 001 027	2087	2109	
	Ж В Й Ь Ф Ц Ш Г Я В Э	000 000 001 030	—	2110
		000 000 001 031	—	2103
		000 000 001 032	—	2081
000 000 001 033		—	2073	
000 000 001 034		—	2095	
000 000 001 035		—	2099	
000 000 001 036		—	2166	
000 000 001 037		—	2155	
000 000 001 040		—	2158	
I J K N	000 000 040 030	2086	2157	
	000 000 040 031	2091	2147	
	000 000 040 032	2065	2092	
	000 000 040 033	2063	2094	

Lisa 6. Aadresside tabel sümbolite väljastamiseks.

n	Pesa, juhul kui n on		
	arv	indeks	astendaja
0	2176	2201	2186
1	2177	2202	2187
2	2178	2203	2188
3	2179	2204	2189
4	2180	2205	2190
5	2181	2206	2191
6	2182	2207	2192
7	2183	2208	2193
8	2184	2209	2194
9	2185	2210	2195

Täht	Pesa	Täht	Pesa	Täht	Pesa
л	2211	ч	2224	й	2237
с	2212	u	2225	ь	2238
б	2213	o	2226	ф	2239
у	2214	e	2227	ц	2240
г	2215	l	2228	щ	2241
д	2216	p	2229	f	2242
м	2217	ш	2230	я	2243
z	2218	ш	2231	ю	2244
н	2219	s	2232	э	2245
х	2220	t	2233	i	2246
а	2221	ы	2234	j	2247
б	2222	ж	2235	k	2248
г	2223	а	2236	n	2249

Lisa 6 (järg).

Märk	Pesa	Märk	Pesa	Märk	Pesa
=	2250		2256	√	2262
≠	2251	∞	2257	(2263
>	2252	+	2258)	2264
≥	2253	-	2259	,	2265
<	2254	×	2260	*	2266
≡	2255	/	2261	!	2267

Täheline indeks	Pesa
i	2198
j	2199
k	2200

Tühik - 2270

Kelk+rida - 2274

Ülemine register - 2272

Alumine register - 2273

"Rioni korrall		"Consuli" korrall	
Keskmine register	- 2271	Must lint	- 2271
Rida	- 2268	Punane lint	- 2275
Kelk	- 2269		

Lisa 7. Teenindusprogrammide ja pseudokäskude tööpesad.

Jrk. nr.	Kood	Programmi nimetus	Tööpesad
1	ex	e^x	3-8
2	ln	$\ln x$	3-7
3	lg	$\lg x$	3-7
4	sn	$\sin x$	3-9
5	cs	$\cos x$	3-9
6	tg	$\tan x$	3-9
7	as	$\arcsin x$	3-7
8	ac	$\arccos x$	3-7
9	at	$\arctan x$	3-7
10	sh	$\operatorname{sh} x$	3-7
11	ch	$\operatorname{ch} x$	2-11
12	th	$\operatorname{th} x$	2-11
13	nv	fikseeritud koma arvu trükk	3-9, 16, 17
14	nc	kahendarvu trükk	3-7
15	nm	kompleksarvu trükk	1-10
16	nm	täisarvu trükk	3-9
17	nn	liikuva koma arvu trükk	3-9, 16, 17, 20-26
18	ng	topeltarvu trükk	3-9, 16, 17, 20, 21
19	nk	käsu trükk	3-9
20	nu	indeksi trükk	3-9
21	ck	kompleksarvude liitmine	3-7, 16, 17, 20-23
22	bk	kompleksarvude lahutamine	3-7, 16, 17, 20-23
23	yk	kompleksarvude korrutamine	3-7, 16, 17, 20-23, 25, 26
24	gk	kompleksarvude jagamine	3-7, 9, 16, 17, 20-23, 25, 26
25	cg	topeltarvude liitmine	3-8
26	bg	topeltarvude lahutamine	3-8
27	yg	topeltarvude korrutamine	3-9, 16, 17, 20-23
28	gg	topeltarvude jagamine	3-9, 16, 17, 20-23, 25
29	ra	$\Gamma(x)$	2-8

Lisa 7 (järg).

Jrk. nr.	Kood	Programmi nimetus	Tööpesad
30	ym	täisarvude korrutamine	3-9,16,17
31	gm	täisarvude jagamine	3-10
32	hg	topeltarvu teisendamine liikuva koma arvuks	3-8
33	gh	liikuva koma arvu teisendamine topeltarvuks	3-9
34	kn	\sqrt{x}	3-7
35	om	topeltpesa ülekanne	3-8
36	oj	$Y_p(x)$	2-9,16,17,22,24,25
37	oy	$Y_p(x)$	2-9,16,17,20-26
38	mn	minimaalse elemendi leidmine	3-9
39	mx	maksimaalse elemendi leidmine	3-9
40		Juhtprogramm:	
		"Пыск-1" ja üldosa suunamine aadressi järgi	5,2043-2045 1,6
		suunamine šifri järgi	—
		käskude sisestamine	3-7, 16-20
		kaheksandarvude sisestamine	5,6
		täisarvude sisestamine	3-6,9
		fikseeritud koma arvude sisestamine	3-6,8,9,19
		liikuva koma arvude sisestamine	2-6,8-11,16,17,22,24-26,28,2010,2011
		topeltarvude sisestamine	3-6,8-17,22,24-26,28
		kompleksarvude sisestamine	2-6,8-17,22,24-26,28
41	bb	Väljastusprogramm	2-7,2010,2011
42	cc	Lauaarvuti programm	2-9,16,17,22-26,28,1941-1970,1998-2003,2010-2011,32-55

S i s u k o r d

I. A R V U T I E H I T U S

§ 1. Ülevaade arvutist

1. Põhilisi andmeid	3
2. Funktsionaalskeem	4
3. Registrid	6
4. Mälu	8
5. Välisseadmed	9
6. Juhtpult	13

§ 2. Arvude kujutamine

1. Täisarv	19
2. Fikseeritud koma arv	19
3. Liikuva koma arv	22
4. Topeltarv	24
5. Kompleksarv	25
6. Kaheksandarv	25

II. K Ä S K U D E S Ü S T E E M

§ 1. Käsu struktuur

1. Mikroprogrammid	26
2. Käsu tunnused	28
3. Modifikatsioonid	29
4. Käskude sisendkeel	33
5. Käsu täitmine	37

§ 2. Selgitusi käskude kohta.

1. Tabel	40
2. Aritmeetilised tehted	42
3. Loogilised tehted	44
4. Suunamiskäsud	45
5. Sümboli sisestamine	46
6. Sümboli väljastamine	47
7. Funktsioonide arvutamine	48

III. A R V U T I K A S U T A M I N E

§ 1. Teenindusprogrammid

1. Juhtprogramm	50
2. Väljastusprogramm	55
3. Lauaarvuti programm	57
4. Silumisprogramm	61
5. Testprogramm	64

§ 2. Programmeerimine

1. Algoritmi valik	65
2. Mälu jaotus	65
3. Programmi ja algandmete sisestamine	67
4. Lihtsamaid programmeerimisvõtteid	69
5. Alamprogramm	72
6. Silumine ja vormistamine	77

L I S A D

Lisa 1. "Rioni" koodide tabel	80
Lisa 2. "Consuli" koodide tabel	82

Lisa 3. Käsu modifikatsioonid	84
Lisa 4. Käskude süsteem	85
Lisa 5. Sümbolite sisemised koodid	93
Lisa 6. Adresside tabel sümbolite väl- jastamiseks	96
Lisa 7. Teenindusprogrammide ja pseudo- käskude tööpesad	98

ПРОГРАММЫ ДЛЯ ВСЕХ

II

Электронная вычислительная машина
"Найри-2"

На эстонском языке

Тартуский государственный университет
СССР, г. Тарту, ул. Калкооли, 18
Vastutav toimetaja K. Mäeasa

TRÜ rotaprint 1972. Paljundamisele antud 19.IV 1972. Trükiroognaid 6,38.
Tingtrükiroognaid 5,83. Arvestusroognaid 4,64. Trükiarv 700. Paber 30x42.
1/4. MB 03835. Tell. nr.500.

Hind 35 kop.

