

. Einarsto

A-38970

ENSV TA Füüsika ja
Astronoomia Instituut

JUHTIMISARVUTI

"ВНИИЭМ - 3"

ABIKS PROGRAMMEERIJALE

TARTU ÜLKOOL
RAAMATUKOGU

Tartu 1968

0523

RAAMATUKOGU
TARTU ÜLICOOLI

Koostanud M. Sarv, H. Niilisk, M. Maasik

RAAMATUKOGU
TARTU ÜLICOOLI

RAAMATUKOGU
TARTU ÜLICOOLI

TARTU ÜLICOOLI
RAAMATUKOGU

S i s s e j u h a t u s .

Käesolev juhend on koostatud ENSV TA Füüsika ja Astronoomia Instituudi matemaatika laboratooriumis instituudi 1,5 m teleskoobi AST - 12 juhtimisarvuti "ВНИИЭМ - 3" kasutajaile. Arvuti loogilist struktuuri ja programmeerimise süsteemi on juhendis püütud kirjeldada ulatuses, mis annaks lugejaile piisavalt informatsiooni programmide kirjutamiseks juhtimisarvuti käskude "keeles". Programmide kirjutamiseks algoritmilise keele ALGOL - 60 realisatsioonis arvutile "ВНИИЭМ - 3" ilmub täiendav juhend.

Käesoleva juhendi koostamisel kasutati mitmesuguseid Üleliidulise Elektromehhaanika Teadusliku Uurimise Instituudi materjale, mille koostajaiks olid instituudi töötajad V. D. ROZENKNOP, G. J. ZUJEV, A. P. ROZEN-ZWEIG, V. F. TKATŠ, V. N. STEPANOV, O. N. IVANOV jt. Ülalnimetatud on ka juhendis kirjeldatud programmide süsteemi põhiosa autoriteks. Interpreteerivad süsteemid MC - 3 ja MC - 3M on valmistatud ENSV TA Füüsika ja Astronoomia Instituudi matemaatikaboratooriumis.

I JUHTIMISARVUTI

"ВНИИЭМ - 3" KIRJELDUS.

§ 1. Üldisi andmeid arvutist "ВНИИЭМ - 3".

Arvuti "ВНИИЭМ - 3" on üldiselt üheaadressiline ja teostab tehteid fikseeritud komaga kahendarvudega, milles koma asub märgikoha järel. Sõna pikkus on 24 kahendkohta, kuid on võimalik tehteid teostada ka 12-kohaliste poolsõnadega. Seega saab arvutis kujutada 23-või 11-kohalisi kahendarve vahemikus $(-1, +1)$.

Käske täidab arvuti loomulikus järjekorras. Käsku kujutatakse üldreeglina ühes mälupesas, kuid on ka käske, mis paiknevad kahes järjestikuses pesas.

Arvuti keskmine kiirus on fikseeritud koma režiimis 30000 - 40000 tehet sekundis ja 1200 - 2000 tehet sekundis aritmeetilistel tehetel liikuva koma režiimis (interpreteerivates süsteemides).

Juhtimisarvuti "ВНИИЭМ - 3" põhiseadmed.

1. Operatiivmälu ОЗУ (оперативное запоминающее

устройство) on ferriitsüdamikel ja koosneb ühest kuni seitsmest mäluväljast, milledest igaüks sisaldab 4096 pesa. Arvuti põhikomplekti kuuluvale mäluväljale on omistatud number 0. Operatiivmälu välju saab põhikomplektile lisada kahe kaupa vastavalt numbritega 2 ja 3, 4 ja 5, 6 ja 7. 1,5 m teleskoobi juhtimisarvuti komplekti on planeeritud kolm mäluvälja (numbritega 0,2 ja 3) kogumahuga 12288 pesa.

2. Magnetlint-välismälu "ВНИИЭМ - 37" *, mille üks kassett mahutab 700 m linti, kuhu saab salvestada ca kaks miljonit sõna. Lindi töö kiirus on 2 m/sec.

3. Paljukanaliline muundaja (УПУ)"B - 3I" , mille abil arvuti vahetab informatsiooni juhitavate aparatuuridega .

4. Esimesse väljund-sisendseadmete komplekti kuulub teletaip, perforaator "ПЛ - 20" ja fotosisend "B - 34" Fotosisendil on kaks töörežiimi: a) kaheksandsüsteemis olevat informatsiooni loetakse kiirusega 1000 silpi sekundis, b) kümnendsüsteemi arve ja tähti loetakse start-stopp režiimis kiirusega 400 silpi sekundis.

5. Teise väljund-sisendseadmete komplekti kuulub elektriline trükimasin "ПУВВИ - 92" , perforaator "ПЛ-20" ja fotosisend "B - 34" .

6. Laitrükuri abil saab informatsiooni väljastada täheliste või numbriliste tabelitena ja graafikutena.

Laitrükkuri trükireas on 128 täheruumi, erinevaid trükimärke on 64 ja trükimiskiirus on 5 rida sekundis.

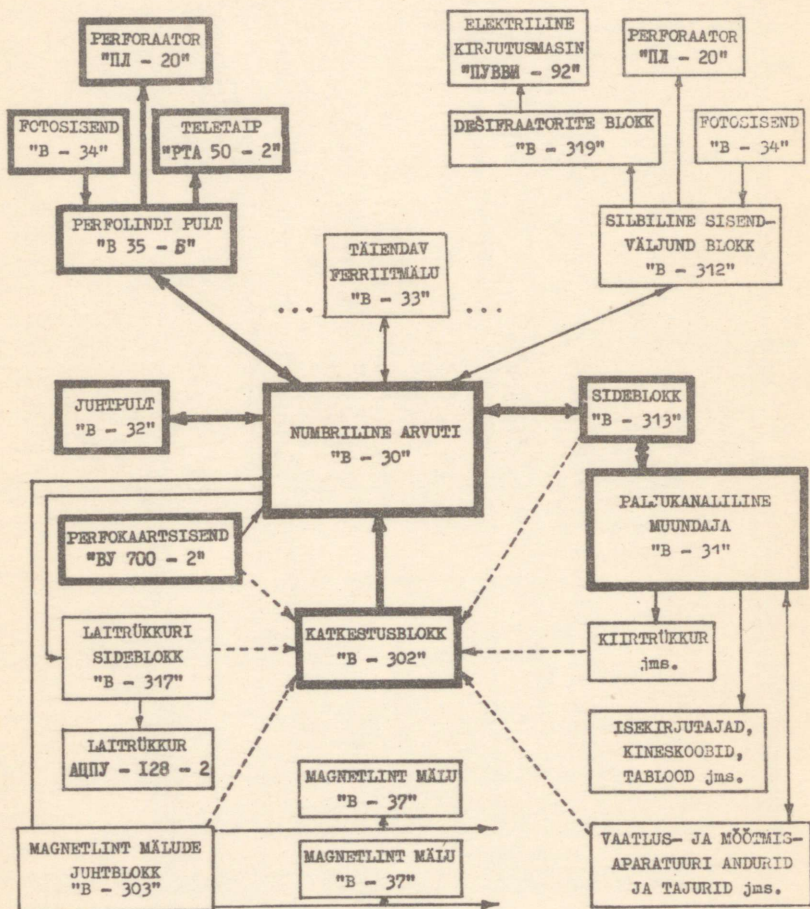
7. Arvutil on multiprogrammeerimise võimalused. Arvutile võib komplekteerida kuni kaheksa katkestusblokki, millest igaüks võimaldab valikut kahekümne ühe programmi vahel. Mitme katkestuspalve üheaegsel esinemisel eelistatakse väiksema bloki numbriga ja blokis väiksema numbriga kanalit. Meie arvutile on ette nähtud üks katkestusblokk. Informatsiooni massiivide vahetamiseks arvuti ja välisseadmete vahel on katkestussüsteemil kümme nn. informatsioonilise katkestuse kanalit.

8. Juhtimispuul on ette nähtud arvuti töö juhtimiseks ja operaatori informeerimiseks arvuti tööst.

Esitame joonisel nr. 1 ** arvuti "ВНИИЭМ - 3" struktuuri blokskeemi, mis üldiselt langeb kokku traditsioonilise elektronarvutite struktuuriga - AS (aritmeetiline seade), JS (juhtimisseade), OM (operatiivmälu) ja välisseadmed. On iseärasusi, mis muudavad arvuti töö paindlikumaks, nagu näiteks informatsiooni vahetamise register P3Y, unifitseeritud informatsiooni vahetamise kanalid jt.

* Arvuti komplekti seadmete korral kasutame edaspidi tähistust B, näiteks "B - 37".

** Joonisel nr.2 on Tõravere komplekti lihtsustatud blokskeem.



JOONIS 2. Juhtimisarvuti "BHM-3" struktuur.

Põhikomplekt - tähistatud intensiivse joonega, planeeritud ja teeksil olev laiendamine - nõrga joonega.

§ 2. Unifitseeritud informatsiooni vahetamise kanalid.

Välisseadmete ühendamiseks arvutiga "ВНИИЭМ - 3" on kaks nn. informatsiooni vahetamise kanalit. Kummagi kanaliga saab ühendada kuni kaheksa välisseadet.

Unifitseeritud kanaliteks nimetatakse neid sellepärast, et informatsiooni vahetus kanaliga ühendatud välisseadmega toimub vastavale kanalile omasel välisseadme olemusest sõltumatul kujul. Sellise kanali kaudu välisseadmete poole pöördumise käskudes näidatakse tingimata ära välisseadme number (0 - 7), mis avab vastava välisseadme ühenduse informatsiooni vahetamise kanaliga. Sellega lõpeb antud kanali külge ühendatud välisseadmete eristamine arvuti poolt.

Välisseadmed kohandatakse tööks vastava kanaliga vastavate vaheblokkide abil.

1. Ühekordse informatsiooni vahetamise kanal.

Ühekordse informatsiooni kanaliga ühendatavateks seadmeteks on fotosisendi ja väljundseadmete komplektid.

Seadmeks nr. 0 on perfolindi pult "B - 35B," mille kaudu on ühendatud teletaip "PTA - 50 - 2M," perforaator "ПЛ - 20" ja fotosisend "B - 34."

Tõravere komplekti seadmeks nr. 1 on planeeritud

sisend-väljundseade "B - 312", mille kaudu ühendatakse fotosisend "B - 34" ja kuni neli väljundabonenti, milledeks võivad olla suvalistes kombinatsioonides trükkurid "ПУБВМ - 92" ja perforaatorid "ПМ - 20" .

Ühekordse informatsiooni kanaliga võib veel ühendada teisi seadmeid, mida Tõravere komplekti planeeritud ei ole (seadmed informatsiooni vahetamiseks telegraafi ja telefoni liine pidi jt.).

Ühekordse informatsiooni kanaliga ühendatud seadmetele on informatsiooni arvutisse viimise ja väljatoomise käsud erinevad. Mõlemas käsus on näidatud välisseadme number ja väljatoomise käsus ka selle seadmega ühendatud abonendi number, mille alusel valitakse välisseade ja temaga ühendatud abonent. Ühe käsu põhjal vahetatakse arvuti ja välisseadme vahel ainult üks sõna või selle osa. Erandiks on käsk A 2M 00 (vt. pkt. II §8).

2. Grupilise informatsiooni kanal.

Antud kanali kaudu ühendatud seade vahetab ühe kahte mälupesa hõlmava käsuga sõnade grupi (1 kuni 4096 sõna). Grupilise informatsiooni kanali poole pöördumise käsk näitab selle välisseadme numbrit, millega tuleb informatsiooni vahetada, operatiivmälu pesade algus- ja lõpuaadressi ja nn. välisseadme tehtekoodi. Tehtekood võib sõltuvalt välisseadmest omada erineva sisu.

Käsu täitmine toimub järgmiselt. Välisseade, mille poole arvuti pöördus, salvestab käsu oma registrites, ühendub iga sõna üleandmise momendiks informatsiooni kanaliga ja vahetab informatsiooni. Jooksva aadressi registrisse kirjutatakse käsu saabumise momendil massiivi algusaadress. Peale iga sõna üleandmist suurendatakse registri sisu 1 võrra. Jooksva aadressi ühtimisel lõpuaadressiga operatsioon lõpetatakse.

Grupilise informatsiooni kanaliga võib ühendada paljukanalilise muundaja "B - 31" (VII), laitrükkuri jne., kuid kokku mitte üle kaheksa seadme.

§ 3. Prioriteetse katkestuse süsteem

Tänapäeva arvutid on järjestikulise toimega — korraga kuulub täitmisele üks ja ainult üks käsk. Sellest tulenevalt täidetakse programme arvutis ükshaaval.

Arvutite mälumaht, suur töökiirus ja sisend-väljundkanalite hulk võimaldavad paralleelselt hoida töövalmis mitmeid programme. Töö efektiivsuse huvides on vajalik, et arvuti automaatselt koostaks endale programmide täitmise järjekorra vastavalt etteantud prioriteedile.

Seepärast peavad täitmisele tulevad programmid moodustama ühtse nn. multiprogrammeerimise süsteemi, mida arvuti tõlgendab kui ühte terviklikku programmi.

Eriti on vaja sellist töö organiseerimise meetodit juhtimisülesannete puhul. Juhtimisarvutil tuleb automaatselt jälgida ja reguleerida juhitava objekti mitmeid parameetreid, millede muutumist üldiselt ennustada ei saa. Parameetrite reguleerimiseks tuleb paralleelselt töövalmis hoida vastavad programmid, millede täitmiseks võib tulla vajadusi korduvalt ja suvalises järjekorras. On täiesti loomulik, et juhtimisprotsessis esineb olukordi, kus juhtimisalgoritmi moodustavate programmide prioriteedi järjekorda tuleb operatiivselt muuta.

Nende vajaduste rahuldamiseks kuulub arvuti "BHVIMEM - 3" seadmete hulka katkestusblokk "B - 302" kahekümne ühe nn. programmilise katkestuse kanaliga (numbritega 0 - 20). Arvuti komplekti võib maksimaalselt kuuluda 8 katkestusblokki (numbritega 0 - 7) — see-ga kuni 168 programmilise katkestuse kanalit.

Programmilise katkestuse kanalid võimaldavad:

- 1) seada mitmesugused arvuti poolt täidetavad programmid hierarhilisse tähtsuse järjekorda; 2) olenevalt juhitalval objektil tekkinud situatsioonist, seda järjekorda programmiliselt muuta; 3) korraldada automaatselt programmide täitmine antud momendil kehtivas tähtsuse

järjekorras.

Katkestuse süsteemil on veel teine ülesanne. Rida välisseadmeid töötab arvuti endaga võrreldes suhteliselt aeglaselt. Näiteks magnetlint-välismälust lugemine toimub kiirusega ca 5700 sõna sekundis, samal ajal kui arvuti võiks informatsiooni vastu võtta sisemällu ca 10 korda kiiremini. Seega, kui arvutit "B - 30" koormata ainult informatsiooni magnetlindilt lugemisega, siis ta ca 9/10 oma ajast ei tee kasulikku tööd. Veel halvem on olukord informatsiooni väljatrükkimisel, samuti muude informatsiooni vahetamise operatsioonide puhul.

Selles suhtes arvuti töö ökonoomsemaks muutmiseks on katkestussüsteemis veel 10 nn. informatsioonilise katkestuse kanalit (numbritega 0 - 9), milledest kahte esimest kasutatakse magnetlintvälismälu jaoks, ülejäänud kaheksat grupilise informatsiooni kanalisse ühendatud välisseadmete jaoks. Kui välisseade ühendada arvutiga läbi sellise kanali, siis võib arvuti antud seadmega informatsiooni vahetamisest ülejäävat aega kasutada kas jooksva programmi täitmiseks või samaaegseks informatsiooni vahetamiseks veel mõne teise välisseadmega (või seadmetega).

Vaatleme lähemalt nii programmilise kui informatsioonilise katkestuse kanalite tööd.

1. Programmiline katkestus.

Programmiline katkestuse läbiviimiseks on katkestusseadmel kolm registrit, kusjuures kõikides registrites on iga katkestuskanali jaoks oma triger. Need registrid on järgmised :

- 1) katkestuse nõudmiste salvestamise register PЗП (регистр запоминания прерывания) ;
- 2) katkestuste blokeerimise register РБП (регистр блокировки прерывания) ;
- 3) programmilise katkestuse register РПП (регистр программного прерывания) .

Programmiline katkestuse protsess ise toimub järgmiselt. Kõigepealt peab saabuma juhitavalt objektilt või arvuti komplekti kuuluvalt seadmelt nõudmine programmi katkestamiseks. See nõudmine võib olla antud 1) impulsi kujul, 2) pinge hüppelise muutumise kujul, 3) lihtsalt releekontaktide ümberlüümisel kujul. Impulsi teravuse ja pikkuse, samuti hüppelise pingemuutuse teravuse kohta praktiliselt mingeid kitsendavaid nõudeid ei ole. Need võivad muutuda küllalt laiades piirides. Impulsi amplituud või hüppelise pinge muutumise amplituud peab olema suurusjärgus $-10V \pm 40\%$. Katkestuse nõudmise signaali peale viiakse registris PЗП sellele kanalile vastav triger seis "1". Kui katkestuse nõud-

mine saabub korraga mitmest kanalist, siis kõik vastavad trigerid registris P3II viiakse seisus "1". Mitme katkestuse nõudmise korral hakatakse neid rahuldama järjekorras. Mida väiksem number, seda varem tuleb rahuldamisele vastav katkestuse nõudmine. Rahuldamisele tuleva katkestusnõude valikule mõjub veel katkestuse blokeerimise registri P6II sisu. Selles registris on samuti iga kanali jaoks oma triger, kuid register täidetakse arvuti poolt programmilisel teel, s.o. täidetakse programmis oleva käsu järgi. Kui antud kanalile vastav triger registris P6II on seisus "1", siis on katkestamine selles kanalis lubatud, kui aga "0" siis mitte. Seisus "1" võivad olla korraga mitu P6II trigerit. Registri P6II kaudu toimub programmide täitmise järjekorra programmi-line muutmine. Kolmandas registris P1111 on samuti igale kanalile vastav triger, kuid siin võib korraga olla seisus "1" ainult üks triger. See näitab, milline katkestuse nõudmistest rahuldatakse. Seega üks registri P1111 trigeritest, mis vastab antud kanalile, võib minna seisus "1" järgmiste tingimuste kokkulangemisel:

1) antud kanalile vastavad trigerid registrites P3II ja P6II on mõlemad seisus "1", s.t. on olemas katkestuse nõudmine ja katkestus on lubatud,

2) ei ole väiksema numbriga kanaleid, milledele vastavad trigerid registrites P3II ja P6II oleksid mõle-

mad seisus "1", s.t. ei ole rahuldamata suurema prioriteediga katkestuse nõudmisi,

3) kahes eelmises punktis toodud tingimuste täitmise momendil on registri PIII kõik trigerid seisus "0", s.t. kõik eelnevad katkestuse nõudmised on arvuti poolt vastu võetud. Näiteks olgu registrid järgmistes seisudes:

0	1	2	3	4	5	6	7	8	9	10	15	16	17	18	19	20	kanali number
0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	1	0	P3II
0	1	0	1	0	0	1	0	0	0	0	0	0	1	0	1	0	P6II
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	P11II

Praegusel juhul tuleb täitmisele kolmanda kanali poolt nõutud programm. Kanalites 2 ja 5 on küll katkestuse nõudmine, kuid neid programme ei luba täita register P6II. Kanalites 6 ja 19 on olemas katkestuse nõudmine, samuti on siin katkestus lubatud, kuid kuna kanali numbrid on suuremad kui 3, siis esimeses järjekorras täidetakse kanali 3 poolt nõutud programm ja kolmas triger registris P11II viiakse seisus "1".

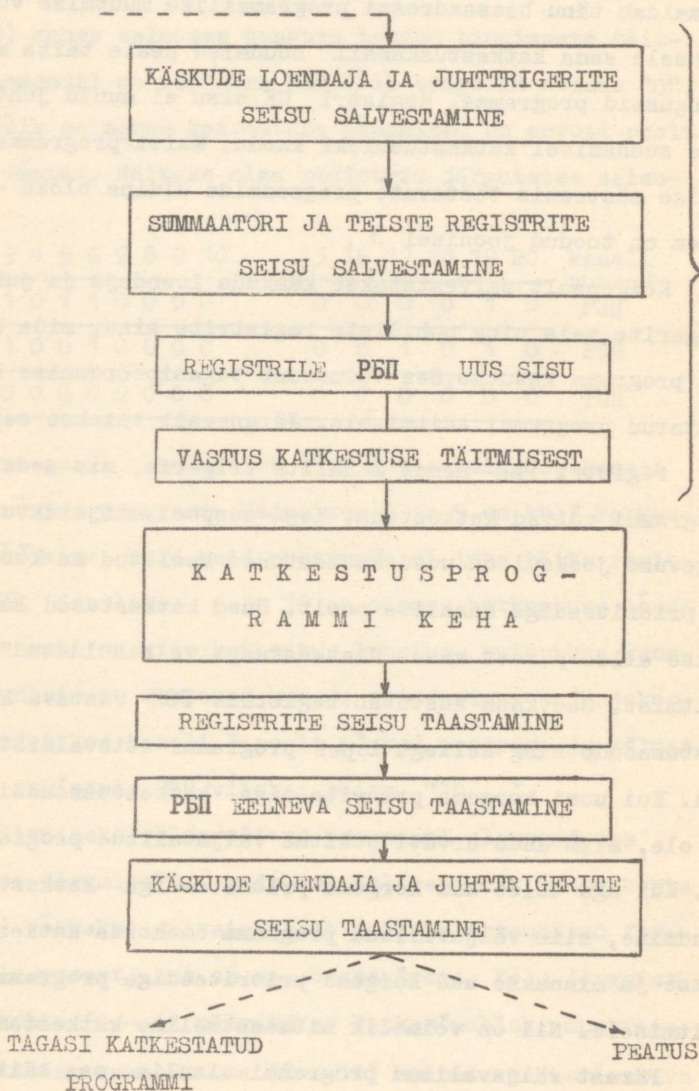
Täitmiseks väljavalitud programmi algusaadress saadakse järgmiselt: aadressi neli esimest kahendkohta (0 - 3) võetakse spetsiaalselt selleks ettenähtud ferriitmälu pesast, kus on nn. baasaadress. Kolm järgmist kahendkohta (4 - 6) määratakse katkestusbloki numbriga. Viis viimast kahendkohta (7 - 11) määratakse katkestava

kanali numbriga. Selline aadressi moodustamise süsteem võimaldab tänu baasaadressi programmilise muutmise võimalusele sama katkestuskanali nõudmise peale täita mitmesuguseid programme. Registri CK sisu ei muutu juhtimise suunamisel katkestusbloki kaudu. Multi-programmeerimise süsteemis töötavate programmide üldine blokk - skeem on toodud joonisel 3 .

Kõigepealt salvestatakse käskude loendaja ja juhttrigerite seis ning põhiliste registrite sisu, mida antud programm kasutab. See võimaldab tagasipöördumise katkestatud programmi täitmisele. Järgnevalt viiakse seisu "1" registri PБII nende kanalite trigerid, mis seda programmi võivad katkestada. Kogu eespool kirjeldatud tegevuse jooksul on uued katkestused keelatud ka kõrgema prioriteediga kanalite poolt. Uued katkestused lubatakse alles pärast käsu "katkestuspalve rahuldatud" täitmist. See käsk kustutab registris P3II vastava katkestusnõude ning sellega lõpeb programmi ettevalmistav osa. Kui uusi kõrgema prioriteediga katkestusnõudeid ei ole, siis asub arvuti täitma väljavalitud programmi. Kui aga tuleb uus kõrgema prioriteediga katkestusnõudmine, siis väljavalitud programm omakorda katkestatakse ja minnakse uue kõrgema prioriteediga programmi täitmisele. Nii on võimalik mitmeastmeline katkestamine.

Pärast väljavalitud programmi sisulise osa täit-

KATKESTUSPROGRAMMI ESIMISE KÄSU
AADRESSI FORMEERIMINE



JOONIS 3. MULTIPROGRAMMI BLOKKSKEEM

mist taastatakse põhiliste registrite seis, registri PБП seis ning käskude loendaja ja juhttrigerite seis. Käskude loendaja seisu taastamine tähendab tagasipöördumist katkestatud programmi täitmisele. Arvuti võib vajaliku programmi täitmist alustada ka seisust "stopp", s.t. mittetöötavast olekust (muidugi peab arvuti olema võrku lülitatud ja valmis arvutamiseks). Arvuti seisu näitab üks juhttrigeritest (T-CM), mis salvestati koos teistega katkestava programmi alguses. Kui see triger on seisus "0", siis arvuti töötab, kui "1", siis seisab (on asendis "stopp"). Vastavalt trigeri T-CM seisule arvuti kas jääb pärast katkestava programmi täitmist seisma (seis "1") või jätkab katkestatud arvutusi (seis "0"). Arvuti komplekti võib kuuluda maksimaalselt 8 katkestusblokki.

2. Informatsiooniline katkestus.

Tavalise välisseadmete poole pöördumise käsu puhul arvuti annab välisseadmele andmed informatsiooni edasiandmiseks (algus- ja lõppaadress, operatsiooni tüüp j.t.) ning ootab kuni välisseade on kogu informatsiooni edasiandnud. Kui välisseade (magnetlintmälu, paljukanaliline muundaja, laitrukkur "АЦПВ-128-2" on ühendatud ühe informatsioonilise katkestuse kanaliga ja tema poole pöördumise käsus on näidatud, et informatsioon tuleb edasi anda katkestusega, siis pärast välisseadmele käsu andmist arvuti jätkab jooksva programmi täitmist. Kui

välisseade on järjekordse sõna operatiivmällu viimiseks (või sealt vastuvõtmiseks) valmis, siis ta nõuab jooksva programmi katkestamist. Pärast parajasti poolelioleva käsu täitmise lõpetamist viiakse järjekordne sõna operatiivmällu või väljastatakse ja arvuti jätkab jooksva programmi täitmist jne. Seega vahetatakse välisseadmega informatsiooni käskude täitmise vahepeal. Korraga võib arvuti vahetada informatsiooni mitme välisseadmega. Järelikult võib üheaegselt saabuda mitu informatsioonilise katkestuse nõudmist. Sel juhul on eesõigus sellel seadmel, mis on ühendatud väiksema numbriga kanaliga. Kanalitega nr. 0 ja 1 on ühendatud magnetlintvälismälud, järelikult on nad teiste seadmetega võrreldes eelisjärjekorras. Informatsioonilisel katkestamisel järjekord on jäik ja seda saab muuta ainult klemmide ümberühendamisega. Programmiliselt järjekorda siin muuta vaja ei ole, sest informatsiooniline katkestus ise on juba programmiliselt esilekutsutav protsess. Informatsioonilise katkestuse nõudmised fikseeritakse vastavas registris РИП (регистр информации прерывания), kus iga kanali jaoks on oma triger. Selle registri seisu järgi määratakse kindlaks prioriteet ja näidatakse vastavale välisseadmele, et tema nõudmine on arvuti poolt arvesse võetud.

Kui mõne välisseadme poole on pöördutud informat -

siooni edasiandmiseks katkestusega, siis enne eelneva informatsiooni vahetuse lõppu jooksvas programmis teist korda sama seadme poole pöörduda ei või. Vastasel korral võib tekkida olukord, kus arvuti ootab välisseadme vabanemist, et täita tema poole pöördumise käsku, välisseade aga ootab käimasoleva käsu lõppemist, et arvutiga vahetada informatsiooni. Nii võivad nad oodata lõpmata kaua. Et teatada arvutile, millal informatsiooni vahetamine välisseadmega on lõppenud, selleks kasutatakse mõnda programmilise katkestuse kanalit.

Informatsioonilisel katkestamisel programmiliselt kättesaadavate registrite sisu arvutis ei rikota, seega informatsiooniline katkestus programmi käiku ei riku. Kui samaaegselt saabub arvutile nii programmilise kui ka informatsioonilise katkestuse nõudmine, siis eelistatud on viimane.

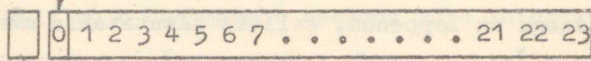
§ 4. Arvuti "ВНИИСМ - 3" tähtsamad registrid.

PЗУ - informatsiooni vahetamise register (регистр запоминающих устройств), millel on 24 kahendkohta. Läbi registri PЗУ käib arvutis igasugune informatsiooni vahetamine. Tänu registrile PЗУ on võimalik paralleelselt aritmeetilises seadmes teostatavate arvutuste-

ga vahetada informatsiooni arvuti ülejäänud seadmete vahel. See on eeliseks niisuguste arvutite ees, millel informatsiooni vahetamine toimub läbi summaatori (näit. "Ural - 4", "Minsk - 22").

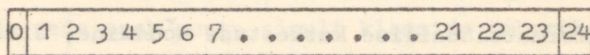
Aritmeetiline seade sisaldab järgmisi registreid:

PC - summaator (регистр суммы) on 25 kahendkohaline. märgi koht



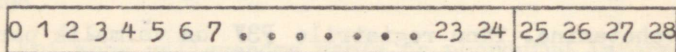
ületäitumise koht numbrikohad

PMЧ - teguri-jagatise register (регистр множителя-частного) on 25 kahendkohaline.



märgi koht numbrikohad paarsuse kontrolli koht

KР - kontrollregister (контрольный регистр) dubleerib oma 25 esimese kohaga summaatorit PC koos arvutusskeemidega ja nelja ülejäänud kohaga registri PMЧ kõrgemaid kohti. Kokku on kontrollregistril 29 kahendkohta.



PC

PMЧ

Loetleme järgnevas juhtimisseadme registrid:

СК - käskude loendaja (счетчик команд) on 12-kohaline, registris СК salvestatakse järgmisena täitmisele tuleva käsu aadress;

ПКК - käskude välja aadressi register (регистр поля команд) näitab, missugusest mäluväljast võetakse järgmine käsk;

ППА - operandide välja aadressi register (регистр поля адреса) näitab, missugusest mäluväljast võetakse operand, kui välja aadressi ei ole näidatud käsus eneses. Registrate ПКК ja ППА sisu saab muuta programmiliselt. Kui registri ПКК või ППА sisu muutva käsuga tuleks registrisse ПКК või ППА kirjutada mäluvälja number 1, siis vastava registri sisu jääb endiseks;

АЗУ - aadressiregister (регистр адреса запоминающих устройств) on 15-kohaline ja jaguneb kaheks alamregistriks: АЗУ/л 12 kahendkohta näitavad pesa aadressi mäluväljas ja АЗУ/п kolm kahendkohta mäluvälja numbrit;

РФА - fikseeritud aadressi register (регистр фиксированного адреса) on kuuekohaline register, mida kasutatakse nullinda mäluvälja esimese 64 pesa poole pöördumisel;

РХП - tunnuse register (регистр хранения признака) on kolmekohaline;

РХМ - modifikaatori register (регистр хранения

модификатора) on kolmekohaline;

PKO - tehtekoodi register (регистр кода операции) on kuuekohaline;

Juhtimisseadme töö sõltub veel rea ühekohaliste juhttrigerite seisust. Tähtsamad neist on:

T - CB (триггер сбоя), mis viiakse seisu "1", kui mõni kontrollsüsteem avastab vea ja juhtimine antakse pesale 0026 (registri СК sisu seejuures säilib).

T - ЗС (триггера запоминания сбоя) Arvutis on 9 sisseehitatud kontrollsüsteemi, millest igähele vastab üks triger.

ТЗП ületäitumise salvestamise triger (триггер запоминания переполнения) viiakse seisu "1", kui arvutuste käigus summaatoris tekkinud arv on suurem kui 1. Ületäitumise signaal registris ТЗП kustub, kui arvutuste tulemusena summaatoris ületäitumise signaal kustub või kui mälust tuuakse summaatorisse uus arv.

Eraldi on kuus triggerit, mille sisu on programmiliselt kättesaadav: ТБП2, ТБП3, ТБП4, ТБП5 (триггера блокировки полей 2,3,4,5). Kui mõni neist trigereist on seisus "1", siis katset pöörduda salvestamiseks vastava mäluvälja poole arvuti tõlgendab kui viga.

Т вл. пер. - ületäitumise mõju triger (триггер влияния переполнения). Kui antud triger on seisus "1", siis triggeri ТЗП seisu "1" tõlgendab arvuti kui viga

(vt. trigger T - СБ),

T - СМ - seisutriger (триггер состояния машины) lubab peale katkestuse täitmist otsustada, kas enne katkestamist arvuti seisis või töötas.

Iga mäluvälja juurde kuulub kaks registrit ПП (регистр поля) ja АП (регистр адреса). Register ПП on kolmekümnekojaline. Siin toimub Hemmingikoodi formeerimine, kontroll ja vigade parandamine Hemmingikoodi järgi. Register АП on 12-kojaline ja on ette nähtud mäluvälja sisese aadressi määramiseks. Tähistustes järgneb neile mäluvälja number, näiteks ППО ja АПО.

§ 5. Arvuti juhtimine puldiit.

Arvuti töö käsitsi juhtimiseks ja visuaalseks kontrollimiseks on juhtimispaneel, mis sisaldab klaviatuuri (vt. joon. 4) ja signalisatsioonipaneeli (vt. joon. 5), millel paiknev indikatsioon näitab arvuti peamiste sõlmede seisundit.

1. Signalisatsioonipaneel.

Indikatsioon signalisatsioonipaneelil vastab järgmistele arvuti sõlmedele:

- 1) "PC" - summaatori järkudele,

- 2) "РЗУ" - informatsiooni vahetamise registri (mä-
luregistri) järkudele,
- 3) "РМЧ" - teguri - jagatise registri järkudele,
- 4) "АЗУ" - aadressiregistri järkudele,
- 5) "РПА" - operandide välja aadressi registri
järkudele,
- 6) "РПК" - käskude välja aadressi registri jär-
kudele,
- 7) "РХП" - tunnusergistri järkudele,
- 8) "РХМ" - modifikaatori registri järkudele,
- 9) "РКО" - tehete koodi registri järkudele,
- 10) "СТ" - taktide loendaja järkudele,
- 11) "СК" - käskude loendaja järkudele,
- 12) "Блок. зап." - mäluväljade 2 - 5 blokeerimise
trigeritele,
- 13) "Влияние переп." - ületäitumise mõjutrigerile,
- 14) "Запомин. переп." - ületäitumise salvestamise
trigerile,
- 15) "Блокир. прерыв." - katkestuse blokeerimise
trigerile,
- 16) "Прерыв. стоп" - katkestuse trigerile,
- 17) "Прерыв. инф." - informatsioonilise katkes-
tuse trigerile,
- 18) "Прерыв. прогр." - programmilise katkestuse
trigerile,

19) "PC ≠ KP"

"АПО ≠ АЗУ"

"РПО ≠ РЗУ"

"Перегрев. терм."

"PC ≠ РМЧ"

- veatrigeritele T - СБ_I →

"Двойной сбой ФЗУ"

→ T - СБ_{I3}

"Перегрев. маш."

kontrollsüsteemide poolt

"Четность РМЧ"

avastatud vigade indit-

"ВНИ ≠ маш"

seerimiseks,

"РЗУ ≠ PC"

"Запрещ. зап."

"РЗУ ≠ РМЧ"

"Сбой внешн."

20) "Блокировка сбоя"- veatrigerite blokeerimise trigeritele.

2. Juhtimispaneel (klaviatuur).

Kirjeldame järgnevas juhtimispaneeli.

- 1) "Oст. по адр." - klahvipaarid 0 - 14 — kui selle klaviatuuri alumistel klahvidel on valitud mingi kombinatsioon, mis vastab mingile aadressile programmis ning töörežiimiks on "Peatus aadressi järgi" (vt. punkt 4c) siis programm peatub sellel aadressil.

- | | | | |
|-------------|----------------|--------|--------------|
| 2) "РЗУ" | - klahvipaarid | 0 - 23 | } -- klahvi- |
| 3) "АЗУ" | - klahvipaarid | 0 - 11 | |
| "СК" | - klahvipaarid | 0 - 11 | |
| "Поле адр." | - klahvipaarid | 1 - 3 | |
| "Поле ком." | - klahvipaarid | 1 - 3 | |

paari alumisele klahvile vajutamine viib registri vasta-
va järgu trigeri seisus "1", ülemine klahv aga seisus "0"
(kustutab järgu). Klahvipaaridest vasakul oleva üksiku
klahviga kustutatakse vastava registri sisu.

- 4) "Гаш. сбоя" - viib veatrigerid seisus "0".
- 5) "Исх. пол." - viib algseisu juhttrigerid ja kustutab
registrite PC ja ПМЧ sisud.
- 6) "Нач. ввод" - lülitile vajutamine käivitab fotosisen-
di (NЗ! fotosisendil peab olema sisse
lülitatud klahv "АВТ").
- 7) "Пуск" - klahvile vajutamine käivitab programmi.
- 8) "Стоп" - klahvile vajutamine seiskab programmi.
- 9) Töörežiimide (vt. punkt 4) lülitid:
 - a) "АВТ" - viib neutraalseisu klahvid "Шар", "Ост.
по адр." ja "Ост. по ГИ" ning lülitab
sisse automaatse režiimi;
 - b) "Шар" - kui lüliti on alumises fikseeritud sei-
sus, siis on sisse lülitatud sammrežiim.
 - c) "Ост. по ГИ" - kui klahv on alumises fikseeritud
seisus, siis on sisse lülitatud

ühetahtiline režiim,

- d) "Ост. по адр." - kui lüliti on alumises fikseeritud seisus, siis on tegu režiimiga, kus arvuti peatub hetkel, kui registrisse АЗУ ilmub see aadress, mis on valitud klahvipaaridel "Ост. по адр."
- 10) "ВЫЗОВ" - klahvile vajutamine toob informatsiooni vahetamise registrisse РЗУ selle mälupeesa sisu, mille aadress on aadressi registris АЗУ .
- 11) "Запом." - klahvile vajutamine viib peesa, mille aadress asub registris АЗУ, informatsiooni vahetamise registris РЗУ asuva informatsiooni.
- 12) "Руч. упр." - kui klahv on välja lülitatud, siis on välja lülitatud ka kõik juhtimispaneeli klahvid, välja arvatud "Нач. ввод", "Пуск", "Стоп" ja "Исх. пол." .
- 13) "Включ. стаб" - lülitab arvuti toitevõrku.
- 14) "Откл. стаб" - lülitab arvuti võrgust välja.
- 15) "Откл. 400 Гц" - lülitab välja voolu ruumisisesestest kaablitest.
- 16) "Вкл. 400 Гц" - lülitab sisse voolu ruumisisesestesse kaablitesse.

- 17) "Пропуск ком." - kuus võtit, mis käsu "suunamine võtme" abil (vt. käsk A 0100 pt.II lk.53) võimaldavad muuta programmi töö käiku programmi en-
nast muutmata.

3. Käivitamine ja peatamine.

Programmi käivitamiseks on vaja teha järgmist:

- 1) lülitada sisse vajalik töörežiim (vt. punkt 4),
- 2) vajutada klahvile "Исх. пол.",
- 3) viia registrisse СК selle pesa aadress, millest algab programm ja registrisse ППК vastava mäluvälja number,
- 4) vajutada klahvile "Пуск".

Arvuti peatamiseks vajutada klahvile "Стоп". Arvuti peatub peale jooksva käsu täitmist ja klahvi "Стоп" valgustuslamp hakkab põlema. Järgmisena täitmisele tuleva käsu aadress on registrites СК ja ППК. Töö jätkamiseks vajutada klahvile "Пуск".

4. Töörežiimid.

a) Automaatrežiim.

Selles režiimis arvuti täidab käskude jada etteantud aadressiga käsust kuni esimese ettejuhtuva peatumise käsuni.

Arvuti lülitamiseks tööle automaatrežiimis

- 1) vajutada klahvile "Исх. пол.",

2) registrisse CK viia täitmisele tuleva programmi algusaadress ja registrisse PPK vastava mäluvälja number,

3) vajutada klahvile "ABT",

4) vajutada klahvile "Пуск".

b) Sammrežiim.

Sammrežiim lülitatakse sisse klahvi "Mar" abil. Vajutamisel klahvile "Пуск" arvuti täidab ühe käsu.

Järgmiste käskude täitmiseks vajutada iga kord klahvile "Пуск".

c) Peatus aadressi järgi.

Režiim "peatus aadressi järgi" erineb automaatrežiimist sellega, et kui registrisse ASV ilmub etteantud täielik aadress, siis arvuti peatub. Režiimi sisselülitamiseks on vaja 1) valida klahvipaaride "Oct. no adr." abil vajalik aadress, 2) lülitada klahvi "Oct. no adr." abil sisse režiim ja vajutada klahvile "Пуск".

Kui arvuti peatub etteantud aadressil, süttib juhtimispuuldil klahvi "Oct. no adr" valgustuslamp ja põleb nupu "Пуск" valgustuslamp.

Aadressi järgi peatuse ajal mälu käsitsi informatsiooni vahetamiseks tuleb arvuti klahvi "Mar" abil lülitada sammrežiimi ja vajutada klahvile "Пуск", millega täidetakse poolelijäänud käsk. Seejärel võib vahetada operatiivmälu informatsiooni klahvide "Вызов"

ja "Запом." abil.

Kui arvuti jäi režiimi "peatus aadressi järgi" ja vajutada klahvile "Пуск", siis arvuti peatub uuesti, kui ilmub jälle etteantud aadress.

d) Ühetaktiline režiim.

Režiim võimaldab jälgida mistahes käskude täitmist taktide kaupa. Režiimi sisselülitamiseks vajutada klahvile "Ост. по ГИ". Klahvile "Пуск" vajutamisel täidetakse üks käsu takt. Järgmise takti täitmiseks vajutada klahvile "Пуск". Peale iga takti täitmist süttib klahvi "Стоп" valgustuslamp.

5. Perfolindil oleva informatsiooni sisestamine.

Kaheksandsüsteemi arvude sisestamiseks perfolindilt tuleb:

1) fotosisendil

- a) vajutada klahvile "Зарядка",
- b) asetada perfolint sisendile ja reguleerida suunajad vastavalt lindi laiusele,
- c) vajutada klahvile "Авт.",

2) juhtimispuldil

- a) vajutada klahvile "Исх. пол.",
- b) lülitada sisse vajalik režiim (vt. punkt 4),
- c) vajutada klahvile "Нач. ввод".

Peale massiivi lõpu märki "" antakse juhtimine

a) automaatrežiimis sisestamise korral pesale 0000 (mälu number registris ППК),

b) sammrežiimis sisestamise korral arvuti peatub. Et anda juhtimine pesas 0000 (mäluvälja number registris ППК) olevale käsule, piisab kui vajutada klahvile "Пуск".

Režiimis "peatas aadressi järgi", kui kõik klahvid "Ост. по адр." on neutraalasendis, viiakse ühekordse klahvile "Нач. ввод" vajutamisega arvutisse üks aadress ja üks sõna. Iga järgmine aadress ja sõna sisestatakse klahvile "Пуск" vajutamisega. Niisuguses režiimis sisestatud sõna on signaalpaneelil näha registris ПМЧ, aadress registris АЗУ. Sisestamise ajal võib käsitsi klahvide "Запом." ja "ВЫЗОВ" abil muuta mälu pesade sisu.

§ 6. Adresseerimise süsteem

operatiivmälu poole pöördumisel.

Arvuti "ВНИИЭМ - 3" operatiivmälu väljad kannavad numbreid 0, 2, 3, jne. Mäluvälja pesad on nummerdatud kaheksandsüsteemis aadressidega 0000 kuni 7777. Operatiivmälu poole pöördutakse kahel eesmärgil: käskude valimiseks ja informatsiooni vahetamiseks.

a) Käskude valimisel määrab register PPK mäluvälja ja register CK aadressi mäluväljas. Peale käsu valimist suurendatakse registri CK sisu ühe viimase koha ühiku võrra (kui ei olnud suunamise käsk). Registri CK ületäitumist ignoreeritakse. Kui programmis ei ole registri PPK sisu muutmise või käskude loomuliku järjekorra muutmise (suunamise) käsku, siis järgneb käsurele aadressiga 7777 sama mäluvälja käsk aadressiga 0000.

b) Informatsiooni vahetamisel saab sõna asukohta mälus näidata lühikese, täieliku või fikseeritud aadressiga.

Poolsonade poole pöördumisel on käsus eraldatud täiendav kahendkoht Π_I , mis määrab kumma poolsõnaga vahetus toimub. Kui $\Pi_I = 0$, siis on vasak poolsõna, kui $\Pi_I = 1$, siis on parem poolsõna. Seega on tunnus Π_I poolsõna aadressi loomulikuks jätkuks. Edaspidi tähistame lühidalt vasakut poolsõna "л. п.", paremat poolsõna "п. п.", poolsõna "п. с."

1. Lühike aadress on sõnade puhul 12- kohaline, poolsõnade puhul 13- kohaline. Mälu välja number võetakse registrist PPA .

2. Täisaadress näitab käsu sees mälu välja numbri ja pesa aadressi mäluväljas. Sõna täisaadress on 15- kohaline, poolsõnadel 16- kohaline. Täisaadressiga pöör-

dutakse operatiivmälu poole ainult üle registri АЗУ ning registrite ППА ja ППК sisu ei muutu. Kui käsus näidatakse mäluvälja numbriga 1, siis võetakse välja number registrist ППА.

3. Fikseeritud aadressiga saab pöörduda nullinda mäluvälja esimese 64 pesa poole. Pöördumine toimub sel korral üle registri РФА ja registrite АЗУ, ППК ning ППК sisu ei muutu. Muidugi on võimalik nende pesade poole pöördumine ka lühikese ja täisaadressiga. Fikseeritud aadressi kasutavad ainult teatud käsud. Edaspidi tähistame fikseeritud aadressiga pesi ФА (Фиксированный адрес)

Osa fikseeritud aadressiga pesadest on eriotstarbelised:

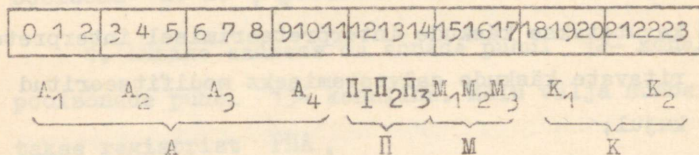
- 0000 - peale programmi sissetoomist (peale massiivi lõpu märki) antakse juhtimine pesasse 0000;
- 01 - 17 -- käsusiseselt kasutatavad modifikaatorid;
- 07 - sisaldab alati sama informatsiooni, mis on registris PC peale käsu täitmist;
- 17 - kasutatakse käskude interpreteerimisel interpreteeritavate käskude salvestamiseks modifitseeritud kujul;
- 20 - interpretatsiooniregister РИТ (регистр интерпретации);
- 21 - teatud tunnuse korral käskudes aritmeetiliste tehete tulemused salvestatakse ka pesas 21 ФА;

- 22 - eelkäsuga näidatud modifikaatori sisu salvestamise pesa;
- 24 - programmilise katkestuse register;
- 25 - vasakusse poolsõnasse salvestatakse vea avastamisel vigade tüüpi näitavate trigerite seis;
- 26 - pesa, millele antakse juhtimine vea tekkimisel;
- 32, 33, 34 - interpreteerivate süsteemide summaator.

Kõigi fikseeritud aadresside, eriti aga ülal loetletute kasutamisel peab olema ettevaatlik ja vajaduse korral kindlustama peale programmi lõppu nende sisu taastamise.

§ 7. Käskude ja arvude kujutamine.

Arvuti "ВНИИЭМ - 3" käsk asub üldiselt ühes pesas. Pesa kohad jaotuvad käsu kujutamisel järgmise skeemi kohaselt:



A - aadress (mõnedel käskudel muu informatsioon).

Π - tunnus, mis varieerib tehte iseloomu. Poolsõnade korral on Π_1 aadressiosa jätkuks olles vasaku

või parema poolsona tunnuseks. Sageli on Π teh-
tekoodi jätkuks.

M - lühikese aadressi korral tavaliselt aadressiosa
modifikaatori fikseeritud aadress, täisaadressi
korral operandi mäluvälja number.

K - tehte kood.

Toodud jaotus on sisuliselt kehtiv üldreeglina.

Leidub käske, mille aadressiosa haarab nii Π kui ka
M ja käske, millel nii Π , M kui ka A võivad täita
tehte koodi funktsioone.

Grupilise informatsiooni vahetamise kanali ja mag-
netlint - välismälu poole pöördumise käsud hõlmavad
kaks järjestikust pesa. Käsu esimese pesa struktuur
vastab üldisele käsu struktuurile. Teises pesas näida-
takse massiivi algus ja lõpp operatiivmälus. Kahepesa-
lise käsu teise pesa jaotamisel kasutatakse järgmisi
variante:

$A_{\text{нач.}}$	№ поля	$A_{\text{конеч.}}$
-------------------	--------	---------------------

$A_{\text{нач.}}$	$A_{\text{конеч.}}$
-------------------	---------------------

1. Fikseeritud komaga arvude kujutamine.

1) Täispesa arvud asuvad pesa kohtadel 1 - 23:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----

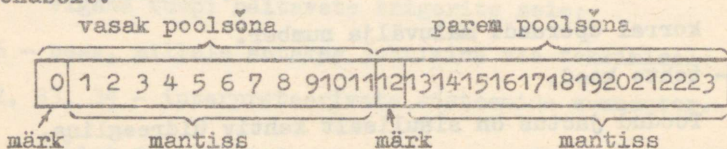
märk

mantiss

Arvude kujutamise vahemik on $2^{-23} \leq |x| < 1$ ja abso-

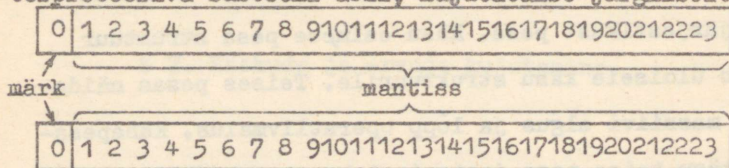
luutne viga ei ületa 2^{-23} , millele kümnendsüsteemis vastab $\approx 1,2 \cdot 10^{-7}$.

2) Poolpesa arvud paiknevad pesas järgmise skeemi kohaselt:



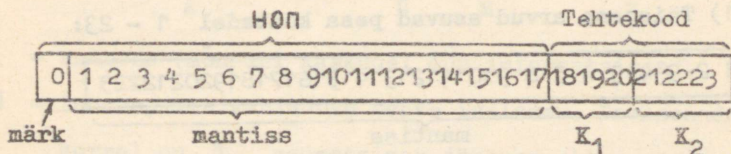
Arvude kujutamise vahemik on $2^{-11} \leq |x| < 1$ ja absoluutne viga ei ületa 2^{-11} , millele kümnendsüsteemis vastab $\approx 4,9 \cdot 10^{-4}$.

3) Topeltpesa arve, mida saab kasutada vastava interpreteeriva süsteemi abil, kujutatakse järgmiselt:



Arvude kujutamise vahemik on $2^{-46} \leq |x| < 1$ ja absoluutne viga ei ületa 2^{-46} , millele kümnendsüsteemis vastab $\approx 1,4 \cdot 10^{-14}$.

4) Vahetu adresseerimisega käskudes, kus käsk koosneb tehte koodist ja operandist HOH, saab kasutada operandina 17 - kohalisi fikseeritud komaga arve.



Vahetu adresseerimise korral operandide kujutamise

absoluutne viga ei ületa $\approx 7,6 \cdot 10^{-6}$.

Liikuva koma režiimis arvutamiseks on seni koostatud viis interpreteerivat süsteemi. Arvude kujutamist nendes süsteemides käsitletakse III peatükis vastavate süsteemide kirjeldustes.

2. Kümnenndüsteemi arvude ja tähelis-numbrilise informatsiooni kujutamine.

1) Programmid interpreteerivates süsteemides

MC - I ja MC - 2 eeldavad, et arvud $\alpha \cdot 10^{\beta}$ on kujutatud mälus kahes järjestikus pesas:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----

mantissi α_1 α_2 α_3 α_4 α_5 α_6
märk

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----

järgu α_6 α_7 α_8 α_9 ↑ β_1 β_2
märk vabad

kus α_i on mantissi ja β_i järgu kümnenndkohad kahendkoodis.

2) Tähelis-numbrilise informatsiooni kujutamiseks kasutatakse arvutis kahendkoodi, milles alfabeedi iga märk kodeeritakse kuue kahendkohalise silbiga

$\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6$. Kohad $\alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6$ vastavad rahvusvahelisele telegraafikoodile N 2. Esimene kahendkoht α_1 näitab registrit. Kui $\alpha_1 = 1$, siis kujutavad $\alpha_2 \dots \alpha_6$ tähte, kui $\alpha_1 = 0$, siis numbrit. Seega

saab ühes mälupesas kujutada nelja tähelis-numbrilist silpi

0 1 2 3 4 5	6 7 8 9 10 11	12 13 14 15 16 17	18 19 20 21 22 23
-------------	---------------	-------------------	-------------------

I silp II silp III silp IV silp

Nende märkide loetelu, mida saab arvutis kujutada, on toodud Lisas 5.

Programmeerimise seisukohalt on lihtsam, kui märgid "буквы" ja "цифры" kirjutatakse teksti sisse. Näiteks tekst "Машина ВНИИЭМ - 3" kujutuks mälus järgmiselt:

1 0 0 1 1 1	1 0 1 0 0 1	0 1 1 0 1 1	0 0 0 0 0 1
-------------	-------------	-------------	-------------

М А циф Ш

0 1 1 1 1 1	1 0 0 0 0 1	1 0 0 1 1 0	1 1 1 0 0 0
-------------	-------------	-------------	-------------

буквы И Н А

0 0 0 1 0 0	0 1 1 0 0 1	0 0 0 1 1 0	1 0 1 1 0 0
-------------	-------------	-------------	-------------

Л В Н И

1 0 1 1 0 0	0 1 1 0 1 1	0 1 0 1 1 0	0 1 1 1 1 1
-------------	-------------	-------------	-------------

И циф Э буквы

1 0 0 1 1 1	0 1 1 0 1 1	0 1 1 0 0 0	0 1 0 0 0 0
-------------	-------------	-------------	-------------

М циф — 3

"ПУВВИ" kood on seitsmekohaline ja viiakse arvutisse kaheksarealiselt lindilt paarsuse kontrolliga. Pesa jagatakse sümbolite kujutamiseks kolmeks grupiks.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----

I

II

III

§ 8. Käskude modifitseerimine.

Arvutusalgoritmide sageli esinev tsükliline iseloom nõuab arvutilt paindlikku tsüklite organiseerimise ja aadresside muutmise süsteemi. Arvutis "ВНИИЭМ - 3" on see süsteem organiseeritud nn. modifikaatorite abil. Modifikaatorina kasutatava pesa sisu tõlgendatakse järgmiselt: kohtadel 0 - 11 asub aadressi muut ja seda osa nimetatakse modifikatsiooni registriks РМ (регистр модификации). Kohti 12 - 23 nimetatakse modifikatsiooni loendajaks СМ (счетчик модификации), mille sisuks on ühe võrra vähendatud tsükli kordamiste arv. №! Modifitseerimine rikub registri РМЧ sisu.

Käsu aadressiosa muutmiseks kasutatava modifikaatori näitamiseks on kaks võimalust.

1. Modifitseerimine spetsiaalse käsuga, millega võib modifikaatorina näidata nullinda mäluvälja

suvalist pesa. Käsule võib eelneada mitu modifitseerimise käsku.

2. Modifikaatori näitamine muudetava käsu sees.

Käsu sees saab näidata modifikaatoritena teatud fikseeritud aadressiga pesi järgmisel kolmel juhul:

a) modifikaatori näitamiseks on käsus ainult üks kahendkoht Π_I . Modifikaatorina saab näidata pesa $OI \Phi A$.

b) käsus on modifikaatori näitamiseks kolm kohta M. Modifikaatorina saab näidata pesi $OI \div O7 \Phi A$.

c) modifikaatori näitamiseks on käsus neli kahendkohta Π_{IM} . Modifikaatoritena saab näidata pesi $OI \div I7 \Phi A$.

Täitmisele tuleva käsu aadress saadakse kõigi näidatud modifikaatorite modifikatsiooni registrite ja käsu aadressiosa summana.

Kui vastavatel modifikaatori näitamise kohtadel on nullid, siis käsu sees modifikaatoreid näidatud ei ole.

Modifitseerimisel aadressi kasvamist üle 7777 ignoreeritakse ja jätkatakse loendamist sama mäluvälja algusest. Seda arvuti omadust saab kasutada negatiivsete juurdekasvude andmiseks. Poolsõnu kasutavate käskude puhul aadressiosa muut modifikatsiooni registrist PM pesa 11. koha ühikutes liidetakse käsu aadressiosale 12. koha ühikutes. Seega poolsõnade puhul aadressi muut 0001 ei anna üleminekut järgmisele pesale vaid järgmisele poolsõnale.

II p e a t ü k k

K Ä S K U D E S Ü S T E E M I K I R J E L D U S

§ 1. Aritmeetilised ja loogilised operatsioonid.

Arvutis "БНММЭМ - 3" teostatakse aritmeetilisi tehteid operandidega, millest üks asub registris PC ja teine pesas, mille aadress A on näidatud käsus. Registrik PC asub esimene operand (esimene liidetav, vähendatav, esimene tegur, jagatav) ja pesas A teine operand (teine liidetav, lahutatav, teine tegur, jagaja). Tehte tulemus saadakse registrisse PC ja mõnede tehete korral ka OBY pessa.

Käesolevas paragrahvis vaatleme käske, millel

$K_1 = 4 - 7$, $K_2 = 0 - 7$ ja $\Pi_2 \Pi_3 \neq 3$. Need kaheksa eri käsku, mille sisu määrab tehte koodi teine number

K_2 , on järgmised:

$K_2 = 0$ - informatsiooni salvestamine operatiivmällu;

$K_2 = 1$ - informatsiooni toomine operatiivmälust min-

gisse registrisse;

$K_2 = 2$ - liitmine;

$K_2 = 3$ - lahutamine;

$K_2 = 4$ - moodulite lahutamine;

$K_2 = 5$ - loogiline korrutamine;

$K_2 = 6$ - korrutamine;

$K_2 = 7$ - jagamine.

Käsu kohad $\Pi_2\Pi_3$ määravad tehte sooritamise iseloomu.

1) Informatsiooni vahetamise käskude ($K_2 = 0 ; 1$) puhul määravad nad, millise registri ja operatiivmälu pesa vahel informatsiooni vahetus toimub. Kui

$\Pi_2\Pi_3 = 00$, siis vahetatakse informatsiooni registriga PC,

$\Pi_2\Pi_3 = 01$, siis registriga PMЧ ja kui

$\Pi_2\Pi_3 = 10$, siis registritega CK, PИA, PИK ja YT.

2) Ülejäänud käskude puhul ($K_2=2 - 7$) määravad $\Pi_2\Pi_3$ tehte tulemuse salvestamise koha. Kui

$\Pi_2\Pi_3 = 00$, siis jääb tulemus summaatorisse PC,

$\Pi_2\Pi_3 = 01$, siis tulemus jääb summaatorisse ja salvestatakse ka pesa 21 ΦA ,

$\Pi_2\Pi_3 = 10$, siis tulemus jääb summaatorisse ja salvestatakse operatiivmälu sellesse pesa, millest võeti teine operand.

Tehte koodi esimene number K_1 näitab adressee-

rimise iseloomu. Kui $K_1 = 4$, siis võtab tehtest osa täieliku aadressiga sõna. Käsu sisese modifikaatorina saab kohal Π_I näidata ainult modifikaatorit $I\Phi A$. Kui $K_1 = 5$, siis on tegemist lühikese aadressiga sõnaga ja modifikaator näidatakse kohtadel Π_{IM} . Kui $K_1 = 6$, siis pöördutakse täieliku aadressiga poolsõna poole. Kohal M on mäluvälja number ja käsu sees modifikaatori näitamiseks ruumi ei ole. Käskudes, millel $K_1 = 7$, kasutatakse lühikese aadressiga poolsõna. Käsu sees jääb modifikaatori näitamiseks M . Peab silmas pidama, et $K_1 = 6, 7$ korral liidetakse käsu aadressiosale modifikatsiooni registri PM sisu ühe koha võrra paremale nihutatuna, kuna on tegemist poolsõnadega (vt. I pt. § 8.).

Kõiki vaadeldud käske on võimalik modifitseerida ka eelkäesuga.

§ 2. Interpreteeritavad käsud.

Interpreteeritavate käskude rühma kuuluvad 32 käsku koodidega 40 — 77, st. sama koodiga käsud, mis eelmises paragrahvis, kuid interpreteerimise tunnuseks on $\Pi_2\Pi_3 = 3$. Interpreteeritavad käsud on omamoodi juhtimise üleandmise käsud, nende täitmine toimub järgmiselt:

- 1) Kui on näidatud modifitseerimine, siis käsu

aadressiosale liidetakse aadressi juurdekasv ja modifitseeritud käsk salvestatakse pesa 17 ΦA. Käsusisesed modifikaatori näitamise võimalused on samasugused kui eelmises paragrahvis.

2) Juhtimine antakse pesale aadressiga (л.п.20ΦA) + K, kus K on interpreteeritava käsu kood. Pesa 20 ΦA nimetatakse interpretatsiooniregistriks (PWT). Mäluvälja number võetakse pesa 20 ΦA kohalt M (15 - 17 kohalt). Kui M = 001, st. niisugust mäluvälja pole olemas, siis mäluvälja number võetakse registrist PИK. Pesa 20 ΦA 18 - 23 kohal peavad olema nullid.

Interpreteeritava käsu täitmisel käskude loendaja CK sisu ei muutu, järelilikult on võimalik tagasi pöörduda interpreteeritavale käsule järgnevale käsule.

Näide: Olgu (PWT) = 7400 0000 ja interpreteeritavad käsud koodiga 50 - 57. Käsud A 3 0 50
A 3 0 51
A 3 0 52 jne.
annavad juhtimise üle pesadesse 7450, 7451, 7452, jne. Neis järjestikustes pesades on tavaliselt alamprogrammide ülemineku käsk A И M И3, mis suunab juhtimise mõne alamprogrammi täitmisele. Seega võime igale interpreteeritavale käsule automaatselt vastavusse seada mõne alamprogrammi. Programmi osa, kuhu interpreteeritavad käsud annavad üle juhtimise (an-

tud näites pesad 7450, 7451, 7452, jne.), ning etatakse interpretatsiooni kommutaatoriks. Muutes pesa PWT sisu, saame juhtimist anda mitmesugustesse kommutaatoritesse. Taoline süsteem võimaldab sama koodiga käsku, näiteks liitmise käsku A 3 M 52 üks kord kasutada fikseeritud komaga arvude liitmiseks, teine kord aga liikuva komaga arvude liitmiseks, vektorite liitmiseks, maatriksite liitmiseks jne., kui muidugi on olemas vastavate alamprogrammide süsteem. Ühelt liitmise liigilt teisele üleminekuks on vaja pessa 20 Φ A viia ainult sellele liitmise liigile vastav suunamise konstant.

§ 3. Vahetu adresseerimine.

1. Aritmeetilised operatsioonid.

Vahetu adresseerimisega käskude puhul on $K_1 = 3$ ja $K_2 \neq 0$. Need käsud täidetakse arvutis suure kiirusega. Vahetu adresseerimise korral asub üks operandidest summaatoris PC (kohtadel 0 - 17), teine käsu sees. (vt. Ipt. §7 lk.40). Tehte ajal loetakse käsu kohad 18 - 23 nulliks. Tehte sisu näitab K_2 samuti nagu aritmeetiliste operatsioonide korral (vt. §1 lk.45).

Käsu aadressi kohal olevaid operandi kahendkohti

võib eelkäsuga modifitseerida.

2. Vahetu adresseerimisega loendamise ja suunamise käsud.

Käskudes koodidega 20 - 27 asuvad loendajad ja nende muutmiseks valitud andmed käsu sees. Käskude täitmisel summaatori PC sisu säilitatakse registris PM4. Seega pärast käsu täitmist on endine registri PC sisu nii summaatoris PC kui ka registris PM4 ja endine PM4 sisu läheb kaduma.

20 - isemuutuv üleminek. Käsu enda aadressiosale A liidetakse arv käsu kohtadelt ПМ ja kirjutatakse A asemele. Juhtimine antakse käsule $A + ПМ$.

Loendamise käskudes, millel kood on 21, 22, 23, asub kaheksateistkohaline loendaja kohtadel АПМ. Käsud kontrollivad, kas loendaja $АПМ = 0$ või $АПМ \neq 0$. Kui $АПМ \neq 0$, siis loendajat vähendatakse ühe võrra. Suunamine toimub järgmise tabeli kohaselt:

kood	21	22	23	juhtimine
tingimus	$АПМ = 0$	$АПМ \neq 0$		$K + 2$
	$АПМ \neq 0$	$АПМ = 0$	$АПМ \neq 0$	$K + 1$

kus K on käsu enda aadress. Koodi 23 puhul, kui $АПМ = 0$, arvuti peatub.

Käsud koodiga 24, 25, 26, 27 lahutavad summaatori

PC sisust algebraliselt käsus asuva vahetu operandi

ATM. Suunamine toimub vahe märgi sign (PC) järgi:

kood		juhtimine	kood		juhtimine
24	25		26	27	
sign (PC):					
" + "	" - "	K + 2	" + "	" - "	stopp
" - "	" + "	K + 1	" - "	" + "	K + 1

§ 4. Juhtimiskäsud.

10 - juhtimiskäsk tsüklite organiseerimiseks modifikaatori muutmise ja kontrollimise teel. Kontrollitava modifikaatori aadress on $\Pi_1 M$. Tehte iseloom sõltub tunnusest $\Pi_2 \Pi_3$.

1) Kui $\Pi_2 \Pi_3 = 00$, siis muudetakse ainult modifikatsiooni loendajat. Pesas $\Pi_1 M$ olevat loendajat (CM) vähendatakse ühe viimase koha ühiku võrra. Kui enne käsu täitmist (CM) $\neq 0$, siis antakse juhtimine pesale A, PIK. Kui (CM) = 0, siis antakse juhtimine järgmisele käsule.

2) Kui $\Pi_2 \Pi_3 = 01$, siis muudetakse modifikatsioo-

ni loendajat (CM) ja registri PM sisu. Kui enne käsu täitmist (CM) $\neq 0$, siis toimub loendaja CM muutmine ja suunamine samuti kui tunnuse $\Pi_2\Pi_3 = 00$ korral, kuid samaaegselt liidetakse modifikatsiooni registri PM sisule üks üheteistkümnenda koha ühik. Liitmisel märgikohta vaadeldakse arvuna ja ületäitumist ignoreeritakse. Kui enne käsu täitmist (CM) = 0, siis registri PM sisu ei muutu, loendajaks CM kirjutatakse 7777 ja juhtimine antakse järgmisele käsule.

3) Kui $\Pi_2\Pi_3 = 10$, siis muudetakse ainult modifikatsiooni registri PM sisu, millele liidetakse käsu aadressiosas olev arv A. Märgikohta vaadeldakse arvuna ja ületäitumist ignoreeritakse. Loendaja CM ei muutu. Juhtimine antakse järgmisele käsule.

Kui käsule koodiga 10 eelneb järgmise käsu modifitseerimise käsk A 0 4 0 0, siis muudetakse ja kontrollitakse modifikaatorit aadressiga A ja käsu sisestelt näidatud modifikaator aadressiga Π_1M jääb tähele panemata. Kui käsu ees on mitu eelmodifitseerimise käsku, siis muudetakse ainult viimases käsus näidatud modifikaatorit.

11 - suunamine summaatori sisu järgi. Suunamise tingimust näitab M (vt. IVpt. lk.98). Tingimuse täidetuse korral antakse juhtimine pesale A mäluväljast Π , muidu järgmisele käsule. Kui $\Pi=1$, siis registri PPK sisu ei muutu.

12 - tingimusteta suunamine. Juhtimine antakse pesale A mäluväljast II. Arv M viiakse uueks operandide välja numbriks registrisse PIIA.

13 - üleminek alamprogrammile. Juhtimine antakse pesale A + 1 mäluväljast II. Pesasse A, II kirjutatakse registrite CK, PIK, PIIA ja YT sisud ja registrisse PIIA uus operandide välja number M. Põhiprogrammle tagasi pöördumine toimub käsuga 41.

0100 - suunamine võtmele. Arvutil on kuus võtit numbritega 0, 1, 2, 3, 4, 5. Neile vastavad käsus kuus kahendkohta A_3A_4 . (Kohtadel A_1A_2 on mistahes informatsioon). Kui üks või mitu võtit on sisse lülitatud ja kasvõi ühele võtmele vastaval kahendkohal on 1, siis täidetakse sellele käsule järgnev käsk. Kui kõigile sisselülitatud võtmetele vastavatel kahendkohtadel on nullid, siis jääb järgmine käsk vahele ja täidetakse ülejäämine. Kui ühtegi võtit pole sisse lülitatud, siis jääb alati järgmine käsk vahele. Olgu näiteks $A_3A_4 = 001100$. Järgnev käsk täidetakse, kui võti 2 või 3 on sisse lülitatud.

0000 - stopp. Käsu 0000 korral jääb arvuti seisma. Seisutriger T - CM kustub ja A ilmub signaalsioonitabloole registrisse A3V / II.

§ 5. Informatsiooni vahetamine fikseeritud aadressiga pesade ja operatiivmälu pesade vahel.

Vaadeldavasse gruppi kuuluvad käsud koodidega 01 - 07, mis vahetavad informatsiooni nullinda mäluvälja esimese 64 pesa (täisarvud) või 32 pesa (poolsõnad) operatiivmälu kõigi pesade vahel. Käsus näidatakse ühe pesa aadress lühikese aadressina kohal A ja teise pesa aadress fikseeritud aadressina kohtadel II ja M. Käskude sisu vastavalt tehte koodile vaata IV pt. §5 lk.99).

§ 6. Magnetlintmälu poole pöördumise käsud.

1. Magnetlindilt lugemine ja lindile kirjutamine.

14 - lugemise ja kirjutamise käsud on kahepesalised.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
K'	P	A''										Π	0	0	N	1 4							
A'						Π A						A''											

Esimese pesa nullindal kohal on kanali number K' , mis on null või üks, kuna arvutil on kaks magnetlindiga informatsiooni vahetamise kanalit. Kohtadel 6 - 11 võib olla mistahes informatsioon (need kohad käsu täitmist ei mõjuta). Kaheteistkümnendal kohal on tunnus Π_I . Kui $\Pi_I = I$, siis teostatakse operatsiooni katkestusega, st. arvuti võib sõna lugemise vahepeal teha ka muud. Kui $\Pi_I = 0$, siis kirjutatakse katkestuseta. Kohtadel 13 - 14 peab olema 0, kohtadel 15 - 17 on kapi number N , kohtadel 18 - 23 on tehte kood ja kohtadel 3 - 5 on informatsioon kontrolli kohta. Kui $A = 0$, siis operatsioon teostatakse dubleeritud ja aktiivse kontrolliga; kui $A = I$, siis kirjutatakse dubleerimata, kuid aktiivse kontrolliga; kui $A = 2$, siis kirjutatakse dubleerimata ja passiivse kontrolliga.

Teise pesa kohtadel 9 - 11 on mälu aadressvälja number. Kohtadel 0 - 8 on massiivi algusaadress A' ja kohtadel 12 - 23 lõppaadress mälus A'' . Olenevalt käsu esimese pesa kohtade 1 - 2 sisust on magnetlinde poole pöördumise käsul kolm juhtu.

a) $P = 0$.

Sõnade rühma kirjutamine ferriitmälust magnetlinde ühte tsooni. Kirjutamist võib alustada niisugusest pesast, mille aadressi A' viimane number on 0, näiteks 6450, 2340. Arvude massiiv $A'' - A'$ ei või olla väiksem kui 8 (vt. punkt 1.c). Üle kaheksa võib massiivi pikkus

olla meelevaldne, sest toimub automaatne üleminek järg-
misesse tsooni.

b) $P = 1$.

Sõnade kirjutamine magnetlindile varem täiskirju-
tatud tsooni. Arvude massiivi pikkus $A'' - A'$ ei või olla
pikem kui tsoon, sest ei ole automaatselt üleminekut järg-
misesse tsooni.

c) $P = 2$.

Magnetlindilt lugemine. Lugeda saab massiive, mil-
les on vähemalt kaheksa arvu.

2. Lindi kerimise käsud.

Lindi kerimise käsk paikneb ühes mälu pesas.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
K'	m											Π_1	N		1		4						

Nullindal kohal on kanali number K' , kaheteistkümnendal
kohal on informatsioon katkestuse kohta, kohtadel
15 - 17 on kapi number N , kohtadel 18 - 23 on tehte
kood. Käsu abil saab teostada kolme liiki kerimist.

a) Kohtadel 13 - 14 on 1, siis keritakse magnet-
linti m tsooni võrra edasi. Tsoonide arv peab rahulda-
ma tingimust $m \leq 2047$.

b) Kui kohtadel 13 - 14 on 2, siis keritakse mag-
netlinti m tsooni võrra tagasi ($m \leq 2047$).

c) Kui kohtadel 13 - 14 on 3, siis keritakse luge-

mispea ette magnetlindi algus.

§ 7. Grupilise informatsiooni kanali

poole pöördumise käsud.

15 -- informatsiooni vahetamise käsk paikneb kahes järjestikuses pesas.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
																M						1		5
A'												A''												

Seade, millega informatsiooni vahetatakse, kannab numbrit M, mis paikneb kohtadel 15 - 17. Kohtadel 18 - 23 on tehte kood. Teises mälu pesas on ferriitmälu algusaadress A' kohtadel 0 - 11 ja lõpuadress A'' kohtadel 12 - 23. Üksikasjaliselt on käsitletud käsku 15 paljukanalilise muundaja (VII) poole pöördumiseks (pt.IV §7).

0600 - määrab välisele seadmele mäluvälja numbri, millega hakkab teimuma informatsiooni vahetus (käsus $K = 15$ selle kohta andmeid ei sisaldu). Käsk näitab välisseadmele veel mõningaid operatsiooni täitmise iseärasusi:

A_4 - välisseadme (VIIY, AIIIV) number grupilise informatsiooni kanalis,

A_3 - mäluvälja number,

A_1 ja A_2 annavad täiendavat informatsiooni.

VIIY puhul omab A_2 järgmist tähendust:

Kui A_2 on paarisarv, siis käsu $K = 15$ puhul võetakse kõik järjestikused lugemid samast kanalist, st. kui käsus $K = 15$ on näidatud algus- ja lõppaadressid A' ja A'' , siis võetakse samast kanalist N järjestikused lugemid pesadesse $A', A' + 1, \dots, A''$.

Kui A_2 on paaritu arv, siis käsu $K = 15$ puhul koos ferriitmälu aadressi muutmisega muutub ka kanali number, millest võetakse lugemid. Näiteks kui on näidatud kanal N ja aadressid A' ja A'' , siis lugemid võetakse järjestikustest kanalitest $N, N + 1$ jne. ning lugemid lähevad pesadesse $A', A' + 1, \dots, A''$. Kui kanali numbrid ei lähe järjest, nagu teisenduste Д/Н, Д/Н_{ОИП} puhul, siis liidetakse igal järgmisel teisendusel kanali numbrile mitte 1, vaid vastav arv (5 või 6).

§ 8. Ühekordse informatsiooni kanali

poole pöördumise käsud.

Ühekordse informatsiooni kanali poole pöördumise käskudes näidatakse kohal M seadme number ühekordse informatsiooni kanalis.

2M00 - kaheksandinformatsiooni viimine arvutisse perfolindilt. Aadress A on salvestatava informatsiooni algusaadress varem määratud mäluväljas PHA. Informatsiooni loetakse sõnade kaupa. Sõna aadress saadakse eelmisele aadressile ühe liitmise teel või on aadress ja mäluvälja number lindil. Viimasel juhul käsus näidatud aadressi ignoreeritakse. Üheaegselt lindilt lugemisega teisendatakse informatsioon telegraafikoodist kaheksandsüsteemi. Lugemise kiirus on 1000 silpi sekundis. Kasutatakse järgmisi juhtmärke:

- "," (koma) - aadressi lõpu märk,
- "," (punkt) - sõna lõpu märk,
- ":" (koolon) - mäluvälja numbrilise lõpu märk,
- "" (apostroof) - massiivi lõpu märk.

Peale märgi "" lugemist antakse juhtimine pesasse O ΦA, kui arvuti on automaatrežimis, või peatub, kui arvuti on sammrežimis.

3MOO - tähestikulis-arvulise silbi viimine arvutisse. Pesa aadress, kuhu silp viiakse, on A. Mäluvälja number võetakse registrist PIIA. Silbi pikkus võib olla kuni 8 kahendkohta. Silp viiakse pesa A ja registri PC parempoolsesse otsa ning pesade A ja PC ülejäänud kohad kustuvad. Ühe käsuga viiakse sisse üks silp. Lint liigub start - stopp režiimis maksimaalse kiirusega 400 silpi sekundis.

IIIM30 - tähestikulis-numbrilise silbi väljastamine. Silp asub pesa A, PIIA viimasel kaheteistkümnel kahendkohal. Π_I näitab katkestusrežiimi, M seadme numbrit ühekordse informatsiooni kanalis. Kui

$\Pi_I = I$, siis töö katkestuseta,

$\Pi_I = 0$, siis töö katkestusega ja üleminekuga alamprogrammile.

$\Pi_2\Pi_3$ on abonendi number.

Arvuti juhtimislauas oleva sisend-väljundseadme

"B - 35B" jaoks on $\Pi_I\Pi_2\Pi_3$ teise tähendusega:

$\Pi = 5$ trükkimine,

$\Pi = 6$ perforeerimine,

$\Pi = 7$ trükkimine ja perforeerimine.

§ 9. Erikärsud.

1. Katkestussüsteemiga seotud kärsud.

1500 - katkestuse blokeerimise registri PБII täitmine. Mäluvälja (PPIA) pesa A kahekümne ühel esimesel kahendkohal olev arv viiakse katkestuse blokeerimise registrisse. Registri numbrit näitavad pesa A kolm viimast kahendkohta.

0200 - "katkestuspalve rahuldatud". Üleminekul katkestusprogrammile blokeeritakse kõik selle programmi katkestused. Blokeerimine lõpetatakse käsuga A 02 00, mis sõltuvalt A_1 -st teeb järgmist:

1) Kui $A_1 = 1$, siis lõpetatakse katkestuse blokeerimine ning kustutatakse katkestuskanalile vastavad katkestusprogrammi tööle kutsunud trigerid registrites PБII ja PIII. Käesoleval juhul A_2 on katkestusbloki number (0 - 7) ja A_3A_4 on nullid. Käsk pannakse alati katkestava programmi ettevalmistava osa lõppu.

2) Kui $A_1 = 2$, lõpetatakse katkestuse blokeerimine ja kustutatakse rikete otsimise blokeeringu trigger ТБС. Käsk pannakse vigade otsimise programmide lõppu.

2. Nihutamiskärsud.

0500 - nihe summaatori sisu järgi. Pesa A, PPIA ja registri PMЧ sisu nihutatakse koos ühe arvuna $(A, PPIA) + (PMЧ) \times 2^{-23}$.

Nihutatakse summaatori viimasel kuuel kohal oleva arvu võrra, nihke suuna määrab (PC) märk. Kui (PC) > 0, siis nihutatakse vasakule, kui (PC) < 0, siis paremale. Nihke tulemus jääb registrisse PC ja PM4. Registrite PC ja PM4 märgikohad tehest osa ei võta. Peale tehet on $\text{sign}(A) = \text{sign}(PC) = \text{sign}(PM4)$.

Enne nihet summaatori ületäitumise signaal ja trigger T3II kustuvad. Kui enne käsu 0500 täitmist (PC) = 0, siis tehte tulemusena viiakse arv (A, ППА) summaatorisse ja $\text{sign}(A) = \text{sign}(PC) = \text{sign}(PM4)$.

1100 - (PC) ja (PM4) nihe. Nihutamise tüübi määravad A_1 kahendkohad.

Kui A_1 I kahendkoht = $\begin{cases} 1 & \text{- tsükliline nihe,} \\ 0 & \text{- lahtine nihe.} \end{cases}$

Tsüklilisel nihutamisel ühest registri otsast välja nihutatavad kahendkohad loetakse teisest otsast sisse. Lahtisel nihkel läheb välja nihutatud informatsioon kaduma.

Kui A_1 II kahendkoht = $\begin{cases} 1 & \text{- nihutamine koos märgikohtadega,} \\ 0 & \text{- ilma märgikohtadeta.} \end{cases}$

Kui A_1 III kahendkoht = $\begin{cases} 1 & \text{- nihutatakse (PC) ja (PM4) koos ühe arvuna,} \\ 0 & \text{- nihutatakse ainult(PC)} \end{cases}$

Nihke suuna määrab A_2 . Kui $A_2 = \begin{cases} 1 & \text{- paremale} \\ 0 & \text{- vasakule.} \end{cases}$

Nihete arvu määrab A_3A_4 .

Koos märgikohtadega nihutamisel summaatori ületäitumise koht ei võta tehtest osa. Ilma märgikohtadeta nihutamisel võtab ületäitumiskoht nihutamisest osa.

Ilma märgikohata (PC) ja (PM4) paremale nihutamisel läheb registri (PC) märgikoht registri (PM4) märgikohale. Märgikohata (PC) ja (PM4) vasakule nihutamisel läheb (PM4) märgikoht (PC) märgikohale.

1200 - normaliseerimine. Registrite PC ja PM4 sisu normaliseeritakse koos ühe arvuna, mille kõrgemad kohad asuvad registris PC.

1) Kui $(T3\Pi) = 0$, siis nihutatakse registrite PC ja PM4 arvukohti vasakule, kuni on täidetud tingimus $I/2 \leq (PC) < I$. Peale nihutamist salvestatakse summaatori sisu pessa A mäluväljast P1A ja nihutatud kohtade arv jääb summaatorisse 23. koha ühikutes negatiivse märgiga.

2) Kui enne normaliseerimist on registrites PC ja PM4 nullid, siis peale normaliseerimist jääb summaatorisse arv 40075000.

3) Kui $T3\Pi = I$, siis nihutatakse registrite PC ja PM4 arvukohti ühe koha võrra paremale. Ületäitumiskohalt nihkub summaatori esimesele arvu kohale 1 ja triger $T3\Pi$ kustub. Normaliseeritud arv viiakse summaatorist pessa A mäluväljast P1A. Summaatorisse jääb 00000001.

3. Erilisi tehteid käskude ja arvudega.

0400 - järgmise käsu modifitseerimine. Aadress A näitab nullinda mäluvälja pesa, kus asub modifikaator. Kui käsule A 04 00 järgneb käsk koodiga 10, siis kuulub kontrollimisele ja muutmisele eelkäsuga näidatud pesas A olev modifikaator.

1400 - jaotamine. Summaatori kuuel viimasel kohal olev informatsioon jaotatakse summaatoris nii, et kohal 17 + i olev number viiakse kohale 3 + 4 (i - 1), kus i = 1, 2, . . . , 6.

(PC) enne käsu täitmist:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	

(PC) peale
käsu täitmist:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	0	0	a_{18}	0	0	0	a_{19}	0	0	0	a_{20}	0	0	0	a_{21}	0	0	0	a_{22}	0	0	0	a_{23}

Käsku 1400 saab kasutada perfokaartidelt arvutisse viidud informatsiooni mälusse paigutamiseks.

1300 - arvu koostamine. Kui registrite PC ja PMY märgid on ühesugused, siis käsk ei tee midagi. Kui nende registrite märgikohad erinevad ja

a) kui (PC) \neq 0, siis lahutatakse summaatori

sisust üks viimase koha ühik ja registris $PM4$ olev arv viiakse pöördkoodi. Registris $PM4$ olevale arvule omistatakse summaatori märk.

b) kui $(PC) = 0$, siis omistatakse summaatoris olevale arvule registris $PM4$ oleva arvu märk.

4MOO - järgmise käsu salvestamine. Sellele käsule järgnev käsk salvestatakse pesa A mäluväljast (PIIK). Edasise töö määrab M . Kui $M = 0$, siis antakse juhtimine salvestatud käsule järgnevale käsule, kui $M = 1$, siis arvuti peatub ja registrisse CK jääb salvestatud käsule järgneva käsu aadress.

5MOO - käsu kustutamine. Kustutatakse mäluväljas (PIIK) pesa A sisu. Kui $M = 0$, siis antakse juhtimine järgmisele käsule. Kui $M = 1$, siis arvuti peatub.

6MOO - märgi omistamine. Märk omistatakse summaatoris või lühikese aadressiga A näidatud pesas olevale arvule vastavalt M väärtusele.

Kui $M = 0$, siis omistatakse pesas A olevale arvule summaatoris oleva arvu märk.

Kui $M = 1$, siis omistatakse summaatoris olevale arvule pesas A oleva arvu märk.

Kui $M = 2$, siis omistatakse pesas A olevale sõnale märk "+".

Kui $M = 3$, siis omistatakse pesas A olevale sõnale märk "-".

Kui $M = 4$, siis muudetakse pesas A oleva sõna märk vas-

tupidiseks.

7000 - tsükliline liitmine. Sõna aadressiga A määratletud numbriga M liidetakse summaatori sisule. Märki kohad võtavad tehest osa kui arvukohad. Ületäitumise korral liidetakse summaatori viimasele kohale üks. Ületäitumise trigeri T3H seis jääb endiseks.

1000 - valjuhääldaja sisse- ja väljalülitamine. Kui $A_1 \geq 4$, siis lülitatakse valjuhääldaja sisse, kui $A_1 < 4$, siis välja.

0700 - isesünkroniseeruva informatsiooni vahetamine.

Isesünkroniseeruva informatsiooni all mõeldakse niisugust diskreetset informatsiooni, mis saabub arvutisse sõltumatult arvutist endast, s.o. meelevaldsetel ajahetkedel. Operatsioon seisneb informatsiooni kirjutamises magnetlindile või sealt väljaandmises momentidel, millal seda nõuab välisseade. Informatsiooni vahetus on grupiviisiline.

0300 - seisu määramine. Arvutisse võetakse informatsioon 24 välise kanali seisu kohta (pingele 0 volti vastab "0", pingele 12 volti vastab "1").

a) kui A on paarisarv, siis seis läheb registrisse PC,

b) kui A on paaritu arv, siis seis läheb pesasse

A, kusjuures summaatori PC sisu jääb endiseks.

1600 - seisu väljaandmine; toitepingete muutmine.

1) Kui $A_1 \neq 0$, antakse välisseadmetele nõudmine sõltuvalt $A_2 A_3 A_4$ - st.

2) Kui $A_1 = 0$, siis toitepingete muutmine. Arvutis on kuus automaatselt muudetavat toitepinge ahelat nominaalidega +20 V, -12 V, -30V, +12 V Ø3Y (ferriitmälu), -12 V Ø3Y ja -30 V Ø3Y. Igaüht neist võib käsu järgi muuta +15 % või -15 % nominaalist.

Ühe pinge muutmisel käsuga kõik teised võtavad automaatselt nominaalväärtuse. Pinge muutuse näitab A (vt. IV pt. §6 lk.104).

III p e a t ü k k

PROGRAMMEERIMISE

S Ü S T E E M I D

§ 1. Interpreteerivad süsteemid.

Käskude interpreteerimise võimalus lubab omistada 32 - le vabaks jäetud tehte koodile sisu vastavalt programmeerija vajadustele. Seega saab kergesti valmistada nn. interpreteerivaid süsteeme liikuva komaga arvudega või mingite teiste matemaatiliste objektidega arvutamiseks. Liikuva komaga arvudega töötamiseks on loodud süsteemid WC-I, WC-2, WC-3, WC-2M ja WC-3M.

Süsteemis WC-I kujutatakse arve kahes järjestikus mälupesas järgmiselt:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----

mantissi
märk

mantiss

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----

järgu
märk

järk

Esimese pesa nullindal kohal asub mantissi märk, kohtadel 1 - 23 asub mantiss. Teise pesa nullindal kohal on järgu märk, kohtadel 18 - 23 on järgu moodul.

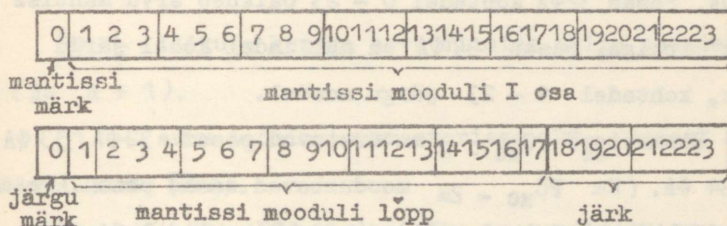
Süsteemis WC - I on arvude kujutamiskiirkond järgmine:

$$2^{-64} \leq |x| \leq (1 - 2^{-23}) \cdot 2^{63} \quad \text{ehk}$$

$$10^{-19} \leq |x| \leq 10^{19}.$$

Mantissi kujutamise täpsus on 6 - 7 kümnendkohta.

Süsteemides WC - 2 ja WC - 2M kujutatakse arve kahe järjestikuses pesas:

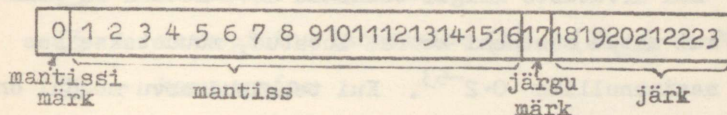


Mantissi kujutamiseks on 41 kahendkohta (koos märgiga), järgule 7 kahendkohta (koos märgiga). Arvude kujutamise kiirkond on

$$2^{-64} \leq |x| \leq (1 - 2^{-41}) \cdot 2^{63} \quad \text{ehk}$$

$$10^{-19} \leq |x| \leq 10^{19}.$$

Süsteemides WC - 3 ja WC - 3M kujutatakse arve ühes mälu pesas:



Nullindal kohal on mantissi märk, kohtadel 1 - 16

on mantiss, 17. kohal on järgu märk ning kohtadel 18 - 23 on järgu moodul. Arvude muutumispiirkond on

$$2^{-32} \leq |x| \leq (1 - 2^{-17}) \cdot 2^{31} \quad \text{ehk}$$
$$10^{-9} \leq |x| \leq 10^9.$$

Tehteid teostatakse operandidega, milledest üks asub interpreteeriva süsteemi summaatoris PC_{MC} , teine pesas. Süsteemides näidatakse käsus ainult liikuva koma-ga arvu esimese pesa aadress ($MC - 1$, $MC - 2$, $MC - 2M$).

Summaatori $PC_{MC} - 1$ moodustavad pesad 32 ΦA ja 33 ΦA . Pesas 32 ΦA kohtadel 0 - 23 paikneb arvu mantiss koos märgiga, pesas 33 ΦA on nullindal kohal järgu märk, kohtadel 1 - 23 järgu moodul.

Summaatori $PC_{MC} - 2$ moodustavad pesad 32 ΦA , 33 ΦA ja 34 ΦA . (Ka $PC_{MC} - 2M$ moodustavad samad pesad.) Pesades 32 ΦA (kohtadel 0 - 23) ja 33 ΦA (kohtadel 1 - 17) paikneb mantiss koos märgiga. Pesas 34 ΦA on nullindal kohal järgu märk ja kohtadel 1 - 23 on järgu moodul.

Summaatori $PC_{MC} - 3$ (ja $PC_{MC} - 3M$) moodustavad pesad 32 ΦA ja 33 ΦA . Pesas 32 ΦA (kohtadel 0 - 23) paikneb arvu mantiss koos märgiga, pesas 33 ΦA on nullindal kohal järgu märk ja kohtadel 1 - 23 järgu moodul.

Kui arvutuste käigus tekkinud arvu moodul on väiksem kui antud süsteemi korral lubatud, muudetakse see nn. masinanulliks $0 \cdot 2^{-63}$. Kui tekkinud arvu moodul on lubatust suurem, siis see arv summaatoris PC_{MC} küll

säilitatakse, aga operatiivmällu salvestamisel (käsk 50) arvuti peatub vastavalt käskudel 7646 (MC - I), 7745 (MC - 2), 7747 (MC - 3) või 7624 (MC - 2M).

Käsu kuju interpreteerivates süsteemides on

A $\Pi_1 \Pi_2 \Pi_3$ M K,

kus A on liikuva komaga arvu aadress, $\Pi_1 M$ on modifikaator, K on tehte kood ja $\Pi_2 \Pi_3 = 3$ on interpreteerimise tunnus.

Süsteemid MC - I, MC - 2, MC - 3 kasutavad käske koodiga 50 - 63.

50 - liikuva komaga arvu viimine mällu pesadesse A (ja A + 1).

51 - arvu toomine summaatorisse PC_{MC} pesast A mäluväljast (ППА).

Käskude 52 - 57 puhul on üks operandidest summaatoris PC_{MC} ja teine pesas, mille aadress on käsus näidatud aadressi ja modifikaatoritega antud aadresside juurdekasvude summa. Tehte tulemus salvestatakse summaatoris PC_{MC} .

52 - liitmine

53 - lahutamine (PC_{MC}) - (A) $\rightarrow PC_{MC}$

54 - moodulite lahutamine

55 - vastupidine lahutamine (A) - (PC_{MC}) $\rightarrow PC_{MC}$

56 - korrutamine

57 - jagamine (PC_{MC}) : (A) \rightarrow PC_{MC}

60 - modifikaatori loendaja muutmine ja kontroll.

Kontrollitav modifikaator on $\Pi_I M$. Kui (CM) \neq 0, siis antakse juhtimine pesasse A + (Л.П. 5ΦA) ja (CM) - 1 \rightarrow CM. Kui (CM) = 0, siis (CM) - 1 = 7777 \rightarrow CM ja (PM) - 1 \rightarrow PM. Pesa 5ΦA vasakus poolsonas hoitakse sõltumatus adresseeringus kirjutatud alamprogrammi algusaadressi.

61 - tingimusega suunamine. Suunamise tingimused vt. käsk 11. Silmas peab pidama, et interpreteerivate süsteemide korral tingimus käib (PC) mitte (PC_{MC}) järgi. Juhtimine antakse pesale A + (Л.П.5ΦA).

62 - tingimusteta suunamine. Juhtimine antakse käsule A + (Л.П. 5ΦA), kus M on aadressväli.

63 - üleminek alamprogrammile, kus A on alamprogrammi algusaadress. Käsk valmistab ette alamprogrammi jaoks rea modifikaatoreid:

(I ΦA) \rightarrow A
(PC) \rightarrow I ΦA
(5 ΦA) \rightarrow A + I
(I7 ΦA) \rightarrow 5ΦA
(CK, PPK, PPA, YT) \rightarrow A + 2
A + 3 \rightarrow CK

Pessa 5ΦA viiakse alamprogrammi algusaadress ja pessa 1 ΦA alamprogrammile vajaliku lähteandmete massiivi algusaadress.

Käskude 52 - 57 korral on summaatoris PC tulemuse järk. Käskude 60 - 62 puhul ei muutu registrite PC ja PM4 sisu. Käsu 63 korral tavaliselt registreid PC ja PM4 kasutab alamprogramm.

Interpreteerimine ja alamprogrammile ülemineku käsk ei muuda registri CK sisu.

§ 2. Interpreteerivate süsteemide standardprogrammid.

Süsteemides WC - I, WC - 2 ja WC - 3 pöördutakse standardprogrammide poole järgmise kahe käsuga:

$$(k + 0) = B \ 00 \ 41$$

$$(k + 1) = A \ 30 \ 63,$$

kus $(B) = C \ n - 1$ ja A on vastava alamprogrammi algusaadress, C on argumentide massiivi algusaadress ja n on argumentide arv.

Standardprogrammid paigutavad tulemused argumentide asemele.

Süsteemides WC - I, WC - 2 ja WC - 3 on koostatud standardprogrammid järgmiste elementaarfunktsioonide arvutamiseks: $y = \ln x$, $y = e^x$, $y = \sqrt{x}$, $y = \sin x$, $y = \arcsin x$, $y = \arctg x$ ning programmid kümnendarvude teisendamiseks kahendarvudeks ja kahendarvude teisendamiseks kümnendarvudeks.

Süsteemide MC - I, MC - 2 ja MC - 3 standardprogramm on kirjutatud sõltumatus adresseeringus alates aadressist 0000. Nende viimine perfolindilt mällu toimub käsuga A 2000, kus A on vastava standardprogrammi algusaadress.

1. Standardprogramm kümnendarvude sisestamiseks "ВВОД".

Standardprogramm "ВВОД" viib arvutisse kümnendarve kujul $x = d \cdot 10^p$, kus

$$d = \varepsilon_d (\alpha_1 \cdot 10^{-1} + \alpha_2 \cdot 10^{-2} + \dots + \alpha_9 \cdot 10^{-9}),$$

$$p = \varepsilon_p (10\beta_1 + \beta_2)$$

ning kus ε_d ja ε_p on vastavalt mantissi ja järgu märk,

$\alpha_1, \alpha_2, \dots, \alpha_9$ - mantissi kümnendkohad,

β_1, β_2 - järgu kümnendkohad.

Lindile tuleb need arvud perforeerida järgmiselt:

$$\varepsilon_d \varepsilon_p \alpha_1 \alpha_2 \dots \alpha_9 \beta_1 \beta_2 \quad \text{või}$$

$$\varepsilon_d \alpha_1 \alpha_2 \dots \alpha_9 \varepsilon_p \beta_1 \beta_2$$

Peale iga kümnendarvu tuleb perforeerida salvestusmärk "." (punkt).

Kui kümnendarvude massiivi ees ei ole aadressi lindile perforeeritud, viiakse massiiv mällu sellest aadressist, mis on näidatud sisseviimise standardprogrammi poole pöördumise käsus.

Aadress perforeeritakse lindile neljakohalise kaheksandarvuna, mille järel peab olema aadressi lõpu märk "," (koma). Massiivi lõpu märk on "'" (apostroof).

Standardprogramm salvestab kümnendarvu kahte järjestikusesse mälu pessa (vt. pt. I § 7, punkt 2).

Programm kasutab tööpesi 0017 ja 32 - 37.

Programm "ВВОД" on ühine interpreteerivatele süsteemidele ИС - I ja ИС - 2.

Programmi "ВВОД" poole võib pöörduda kahel viisil:

a) käskudega $(k + 0000) = L \ 17 \ 01$
 $(k + 0001) = A_m \ 00 \ 13$, kus

$(L) = A \ n - 1$ või

b) käskudega $(k + 0000) = 0017 \ 40 \ 00$
 $(k + 0001) = A \ n - 1$
 $(k + 0002) = A_m \ 00 \ 13$, kus

A on selle pesa aadress, kuhu salvestatakse esimene arv,

A_m on programmi "ВВОД" algusaadress ja

n on salvestatava massiivi pikkus.

Kui perfolindil on esimese arvu ees aadress, siis käsus näidatud aadressi A ignoreeritakse.

Kui massiivi lõpp määratakse märgiga "'", siis peab n olema suurem kui sisseviidavate sõnade arv.

2. Liikuva komaga kümnendarvude massiivi trükkimine

"ВЫВОД".

Trükitav kümnendarv peab paiknema kahes järjestikusse mälu pesas vastavalt pt. I § 7 punkt 2 esitatud skeemile.

Programm trükitab 5 arvu ritta. Iga arvu järel trükitakse üks märk "sõnavahe". Iga arv koosneb kolmeteistkümnest trükimärgist: kümme kohta mantissile koos määrgiga ja kolm kohta järgule koos määrgiga. Esimene arv trükitakse alati uue rea algusest.

Programm kasutab tööpesi 0017, 0032 - 0035.

Programm "Вывод" on ühine süsteemidele ИС - I ja ИС - 2.

Trükkimisprogrammi poole pöördumine on analoogiline programmi "Ввод" poole pöördumisega.

§ 3. ПОП - süsteem.

Standardprogrammide poole pöördutakse vastava pöördumisprogrammiga ПОП - I või ПОП - 2 (ПОП - программа обращения к подпрограммам). Programmi ПОП - I kasutatakse siis, kui vastav standardprogramm on operatiivmälus, ПОП - 2 siis, kui standardprogramm on magnetlindil. Sisuliselt ПОП - 2 hõlmab ПОП - I.

Programmide ПОП - I ja ПОП - 2 peavad paiknema mälus fikseeritud kohtadel:

ПОП - I vahemikus 7674 - 7777 ja

ПОП - 2 vahemikus 7640 - 7777.

Pöördumisprogramm teostab järgmisi operatsioone.

1) Loeb SP (standardprogramm) magnetlindilt mällu ja pöördub arvutamiseks; pöördub arvutamiseks, kui SP on mälus.

2) Häalestab SP esimesel SP poole pöördumisel.*

3) Teostab abioperatsioone, mis lihtsustavad SP koostamist:

a) täidab SP esimesed kolm pesa A, A + 1 ja A + 2, mida kasutatakse ПП - I ja ПП - 2 tööpesadena;

b) säilitab lähteinformatsiooni (nende pesade algus-aadressi, kus paiknevad argumendid või muud faktilised parameetrid) pesas 1 ΦA;

c) võimaldab peale SP töö lõppemist juhtimise andmise põhiprogrammile ühe käsuga.

4) Kontrollib SP sisseviimise õigsust. Perfolindil olevate SP -de sisseviimise õigsust peab kontrollima põhiprogrammi abil spetsiaalse pöördumisega. Magnetlindilt operatiivmällu loetavate SP -de õigsust kontrollitakse automaatselt.

5) Kontrollib enda (ПП - I või ПП - 2) sisseviimise õigsust.

* Kõik standardprogrammid on kirjutatud aadressist 0000 alates. Algusaadressist sõltuvate käskude aadressiosa muutmist nimetatakse SP häalestamiseks.

§ 4. Standardprogrammide ülesehitus

ИОП - süsteemis.

1. Kõik SP -d peavad olema kirjutatud ühes massiivis aadressist 0000 alates.

2. Kolme esimest SP pesa 0000 - 0002 kasutatakse ИОП - I (ИОП - 2) tööpesadena. SP koostamisel tuleb need pesad täita nii, nagu on näidatud punktis 7.

3. Punktidest 1. ja 2. tulenevalt on SP tegelik algusaadress 0003.

4. Informatsiooni SP argumentide kohta saadab ИОП - I (ИОП - 2) pesa 10A vahetult enne SP järgi arvutamist.

5. Põhiprogrammi poole tagasi pöördumine toimub käsu

7772 00 12.

6. Ühes või mitmes pesas SP lõpus, olenevalt SP pikkusest, asub skaala. Skaala pesa iga kahendkoht, välja arvatud märgikoht, vastab ühele SP käsule. Skaala esimese pesa esimene kahendkoht vastab pesas 0000 asuvale käsule, teine kahendkoht pesas 0001 asuvale käsule, . . . , kahekümne kolmas kahendkoht pesas 0026 asuvale käsule, teise pesa esimene kahendkoht käsule 0027 jne. Skaala nendele kohtadele, mis vastavad SP käskudele,

mille adressiossa sõltub SP algusaadressist, kirjutatakse 1, teistele 0. Skaala esimese pesa esimesed kolm kohta on alati nullid.

7. Esimesed kolm SP pesa täidetakse järgmiselt:

$$(0000) = A_{sk} \quad n$$

$$(0001) = 0000 \quad n_1 - 1$$

(0002) = kontrollsumma täiend, kus

A_{sk} on skaala esimese pesa suhteline aadress,

n on skaala pesade arv,

n_1 on SP pesade arv (kaasa arvatud skaala).

8. SP pikkus n_1 ei tohi olla suurem kui 2000_8 .

9. Kontrollsumma täiend pesa 0002 saadakse $\Pi\Pi\Pi - I$ ($\Pi\Pi\Pi - 2$) abil. Selleks viia SP mällu mingist pesast A alates, kustutada pesade A + 2 ja 7702 sisud ning pöörduda $\Pi\Pi\Pi - I$ poole kahe käsuga:

$$(k + 0000) = \quad A \quad 30 \quad 63$$

$$(k + 0001) = 0000 \quad 00 \quad 00.$$

Peatus käsul 0020. SP kontrollsumma ilma pesata A + 2 jääb PC -i. Otsitav konstant saadakse selle täiendina 7777 7777 -ni.

$\Pi\Pi\Pi - 2$ abil on töökäik sama, ainult pesa 7702 asemel tuleb kustutada pesa 7647 sisu.

10. Ei tohi kasutada pesa 27 ΦA , kuna seda kasutavad $\Pi\Pi\Pi - I$ ja $\Pi\Pi\Pi - 2$ tööpesadena.

11. Tööpesad peavad paiknema SP lõpus skaala pesadest alates.

12. Tuleb meeles pidada, et kui SP alustab tööd (käsu A + 3 täitmise ajal), on juhttrigerite seis 0 (need kustutab ПП).

§ 5. Standardprogrammide kasutamine

ПП - süsteemis.

Kui ПП - I (ПП - 2) ja SP on mälus, siis pöördumine SP järgi arvutamiseks toimub kahe käsuga

$$(k + 0000) = A \quad 3M \quad 63$$

$$(k + 0001) = B \quad 00 \quad 00, \text{ kus}$$

A on SP algusaadress,

M on suvaline arv 0 - 7 (mõnede SP -de jaoks omab M konkreetne väärtus tähtsust).

Pesas B ≠ 0 on lähteinformatsioon SP jaoks.

Eelpool toodud pöördumise korral ПП kontrollib automaatselt, kas SP on häälestatud. Kui ei ole, siis kontrollitakse SP sissetoomise õigsust. Kui SP on valesti sisse toodud, peatub arvuti käsul 0020. Pesade 7700 (ПП - I korral) ja 7645 (ПП - 2 korral) vasakus poolsonas on põhiprogrammi aadress k + 2, mille järgi saab otsustada, missugune SP oli valesti sisse toodud.

Kui SP on õigesti sisse toodud, siis

a) häälestatakse ta aadressist A alates,

- b) täidetakse esimesed kolm SP tööpesa A, A + 1, A + 2,
 c) viiakse lähteinformatsioon pesa 1 Φ A

(B) \rightarrow I Φ A,

- d) teostatakse arvutused SP järgi pesast A + 3 alates. Peale arvutuste lõpetamist antakse juhtimine standardsele väljundprogrammile, mis taastab pesade I Φ A ja 27 Φ A sisud ja annab juhtimise põhiprogrammi käsule k + 2.

Kui SP, mille poole pöörduetakse, on juba häälestatud, alustatakse tööd punktist b) alates.

Pöördumine SP sisseviimise õigsuse kontrollimiseks toimub kahe käsuga

$$(k + 0000) = A \quad 3 \text{ M } 63$$

$$(k + 0001) = 0000 \quad 0 \ 0 \ 00,$$

kus A on SP algusaadress,

M on suvaline arv 0 - 7.

Toodud pöördumine omab mõtet ainult häälestatud SP korral.

Kui SP on valesti sisse viidud, peatub arvuti käsul 0020 ((CK) = 0021), kui õigesti, siis antakse juhtimine käsule k + 2.

§ 6. Standardprogrammid ПОН - süsteemis.

Fikseeritud koma režiimi standardprogrammid kasutavad tööpesi 1 Φ A, 17 Φ A ja 21 Φ A. Pesas 1 Φ A sisu

taastatakse peale SP töö lõppu, pesade 1 Φ A, 17 Φ A ja 21 Φ A sisud aga mitte.

1. Funktsioonide $y = e^x$, $y = \ln x$ ja $y = \sqrt{x}$ arvutamine.

Argumendid x peavad olema mastabeeritud kujul $x \cdot 2^{-P_1}$. Funktsiooni väärtus saadakse mastaabis 2^{-P_0} . Järgud p_0 ja p_1 peavad paiknema vahetult argumentide massiivi ees. Näiteks kui informatsioon argumentide massiivi kohta omab kjuu $A \quad n - 1$, siis on pesas

$$\begin{aligned}(A + 0000) &= p_0 \cdot 2^{-23}, \\(A + 0001) &= p_1 \cdot 2^{-23}, \\&\dots \dots \dots \\(A + 0001 + n) &= x_n \cdot 2^{-P_1}.\end{aligned}$$

Pesa $A + 2$ on argumentide $x_k \cdot 2^{-P}$ massiivi algusaadress.

Funktsiooni väärtused paigutatakse argumentide $x_k \cdot 2^{-P_1}$ asemele.

2. Funktsioonide $y = 1/2 \sin x$, $y = 1/2 \arcsin x$ ja $y = 1/2 \arctg x$ arvutamine.

Argumendid x peavad olema kujul $x \cdot 2^{-P}$ (mastabeeritud). Argumendi järk $p \cdot 2^{-23}$ peab paiknema vahetult argumentide massiivi ees. Näiteks, kui informatsioon algandmete kohta on kujul $A \quad n - 1$, siis

$$\begin{aligned}(A + 0000) &= p \cdot 2^{-23}, \\(A + 0001) &= x_1 \cdot 2^{-P}, \\&\dots \dots \dots\end{aligned}$$

$$(A + n) = x_n \cdot 2^{-p}$$

Funktsiooni väärtused salvestatakse argumentide väärtuste asemele.

3. Sõnade trükkimine kaheksandsüsteemis ilma aadressi-
ta, 7 sõna ritta.

Teletaibilindile trükitakse nullindas mäluväljas järjestikustes pesades paiknevat informatsiooni. Igasse ritta trükitakse 7 sõna. Sõna koosneb kaheksast märgist. Sõnad eraldatakse üksteisest tühikuga.

Programmi pikkus on 36 pesa (0000 - 0043). Töö-
pesadena kasutatakse pesi 1 ΦA, 2 ΦA ja 17 ΦA. Peale
SP töö lõppu 1 ΦA ja 2 ΦA sisud taastatakse, 17 ΦA
sisu mitte.

4. Programmi trükkimine lehekülgede kaupa.

Teletaibilindile trükitakse kaheksandsüsteemis mälu
järgestikustes pesades paiknevaid sõnu koos aadresside-
ga. Igasse ritta trükitakse üks käsk. Iga 40_8 rea jä-
rel trükitakse reavahe. Saab trükkida nii tõelistes aad-
ressides kui aadressist 0000. Programmi poole pöördutak-
se järgmiselt:

$$(k + 0000) = A \quad 3 \quad M \quad 63$$

$$(k + 0001) = B \quad N \quad , \text{ kus}$$

$$(B) = A_M \quad n - 1, \quad (\text{sisendinformatsiooni pesa})$$

A on SP algusaadress,

N on SP number,

A_M on trükitava massiivi algusaadress,

n on trükitava massiivi pikkus.

Kui $M = 0$, trükitakse programm aadressidega 0000 -st alates. Kui $M = 1$, trükitakse programm tõelistes aadressides.

Kui soovitakse muuta käskude arvu trükitaval leheküljel, tuleb viia vajalik konstant kaheksandsüsteemis pesa II. II. ($A + 0064$).

5. Täisarvude teisendamine kahendsüsteemist kümnendsüsteemi ja tulemuse trükkimine.

Programm teisendab ja trükkib järjestikustes mälu pesades olevaid arve. Arvud trükitakse teletaibilindile 10 - arvuliste gruppina. Mälus kahendarvud säilivad.

Kirjeldatud programmiga saab trükkida ainult nullindas mäluväljas olevaid arve.

6. Fikseeritud komaga arvude teisendamine kahendsüsteemist kümnendsüsteemi ja tulemuse trükkimine.

Arvud trükitakse teletaibil 10 -arvuliste gruppina. Mälus kahendarvud säilivad. Teisendada (ja trükkida) saab ainult nullindas mäluväljas olevaid arve.

7. Kaheksandarvude massiivi väljastamine.

Standardprogrammi pikkus on 130_8 pesa. Programm kasutab tööpesi 02 ΦA , 03 ΦA , 30 ΦA , 31 ΦA , 32 ΦA , 33 ΦA , millede sisu taastatakse väljumisel standardprogrammist.

Programmi poole pöördumine on standardne:

$$(k + 0000) = A \ 3 \ M \ 63$$

$$(k + 0001) = B \ 0 \ 0 \ 00, \text{ kus } (B) = C \ n - 1,$$

A on standardprogrammi algusaadress,

C on väljastatava massiivi algusaadress,

n on väljastatavate pesade arv.

Pöördumise käsus number M määrab väljastamise ise-
loomu:

$M_3 = 1$ koos aadressidega

$M_2 = 1$ teletaibile

$M_3 = 0$ ilma aadressideta

$M_1 = 1$ perforaatorile.

Programm trükitab massiivi teletaibil kaheksa sõna
ritta. Arvud eraldatakse üksteisest punktiga.

Aadressidega väljastamisel trükitakse iga rea ees
eraldi ritta rea esimese arvu aadress ja eraldatakse
see komaga. Massiivi lõppu trükitakse " ' " (apostroof).
Perforaatoril perforeeritakse veel iga massiivi algusse
ja lõppu 20 sümbolit "vene".

Saadud massiivi perfolint on valmis massiivi sisen-
damiseks arvuti operatiivmällu käsuga A 2000 või ope-
ratsiooniga "Начальный ввод".

Juhul kui on vaja muuta sõnade arvu reas või süm-
bolite arvu sõnas olgu märgitud, et SP pesas

л.п. (A + 107) on viimase koha ühikutes n - 1, kus n
on sõnade arv reas ja pesas л.п. (A + 115) on viimase
koha ühikutes m - 1, kus m on sümbolite arv sõnas. Kui
SP töö jääb mingil põhjusel pooleli, siis programm ei

ole korras järgmiseks pöördumiseks.

8. Kümneendüsteemi täis- ja ühest väiksemate murdarvude arvutisse viimine ja kahendsüsteemi teisendamine.

Käesolev programm viib perfolindilt arvutisse kümneendüsteemi täis - ja murdarve ning samaaegselt teisendab need kahendsüsteemi. Mälusse viidavate arvude hulk peab alati olema antud SP poole pöörduva käsuga. Siseseviidavate arvude algusaadressi võib näidata käsuga või perforeerida lindile. Kui aadress on lindil, siis jätab arvuti käsus näidatud algusaadressi tähele panemata.

Täisarvud ei või olla rohkem kui 7 - kohalised ja peavad rahuldama tingimust

$$N \leq 8388607 = 2^{23} - 1.$$

Täisarv N teisendatakse kahendsüsteemi ning kirjutatakse mällu kujul $N \cdot 2^{-23}$.

Murdarve $|x| < 1$ võib perforeerida suvalise kohtade arvuga, aga kahendsüsteemi teisendamisele lähevad ainult kuus esimest kohta.

Arvude kirjutamine (blanketile) ja perforeerimine.

Kümneendarvud kirjutatakse blanketile tavalisel kujul: 0; 0,5488764; - 548,05; - 0; 33; jne. Täisarvul on esimene number alati nullist erinev, murdarvul

null. Märki " + " võib jätta kirjutamata. Vajaduse korral kirjutatakse ette tema aadress neljakohalise kaheksandarvuna.

Perforeerimisel pidada silmas järgmist. Iga arvu lõpu perforeerida arvu lõpu märk "." (punkt). Vahetult peale viimase arvu salvestus-märki tuleb perforeerida massiivi lõpu märk "' (apostroof). Murdarvude perforeerimisel võib koma ära jätta, s.t. $0,5 = 05$. Kui juhuslikult positiivsele arvule perforeeriti märk "-", siis perforeerida märk "-" veelkord, siis vastav arv viiakse arvutisse õige märgiga. Näiteks $--0,5 = 0,5$.

Aadress perforeeritakse neljakohalise kaheksandarvuna. Enne aadressi peab olema tühik, mis on aadressi alguse märgiks. Peale aadressi perforeeritakse aadressi lõpu märk "," (koma).

Kui on valesti perforeeritud aadress ja aadressi lõpu märk on veel lõõmata, tuleb uuesti lüüa õige aadress ja siis koma. Kui vale aadressi järgi on koma juba perforeeritud, tuleb uuesti lüüa tühik, aadress ja seejärel koma.

Kui valesti on perforeeritud arv, siis tuleb lüüa arvu lõpu märk "." ning alles siis tühik, arvu aadress, koma, arv ja punkt.

Lindile perforeeritud märk, mida sisendprogramm ei kasuta, ei sega arvude sisseviimist.

§ 7. Süsteemid ИС - 2М ja ИС - 3М.

Interpreteerivate süsteemide standardprogrammide lihtsustamiseks on loodud süsteemid ИС - 2М ja ИС - 3М, mis sisuliselt ühendavad endas vastavalt süsteemid ИС - 2 ja ПОП - 2 ning ИС - 3 ja ПОП - 2. Mõlemas süsteemis on jäetud kaheksa pesa vaba kommutaatorit koodide 40 - 47 jaoks, mis lihtsustab uute interpreteerimisprogrammide süsteemi lülitamist. Võrreldes süsteemidega ИС - 2 ja ИС - 3 on ära jäetud käsud koodiga 60, 61, 62, kuna kõik standardprogrammid on ümber tehtud vastavalt ПОП- süsteemi nõuetele: SP algusaadressi ei määrata pesa 5 ФА abil, vaid SP häälestatakse enne tööle hakkamist. Standardprogrammides on veel täienduseks, et on ühendatud programmid "ВВОД" ja "10 → 2" ning "2 → 10" ja "ВЫВОД". Viimane programm kahendarve mälus ei riku. Trükkimisprogrammi pöördumiskäsus näitab M, mitu arvu soovitakse trükkida ritta $M = 0, 2, \dots, 7$. Mantissi kohtade arv paikneb pesas Л.П. (A + 0137). Esialgsel kujul on seal 0010, s.t. programm trükkib arvud üheksakohaliste mantissidega.

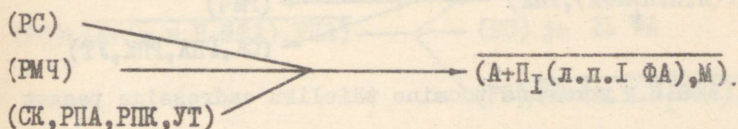
IV p e a t ü k k

K Ä S K U D E S Ü S T E E M

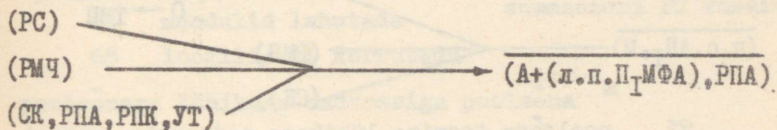
§ 1. Põhiliste aritmeetiliste ja

loogiliste operatsioonide käsud.

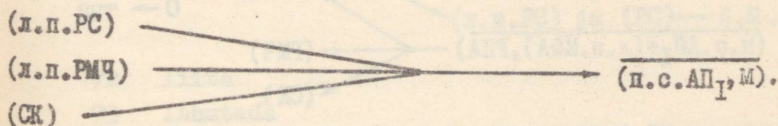
40 sõna salvestamine täieliku aadressiga pesasse



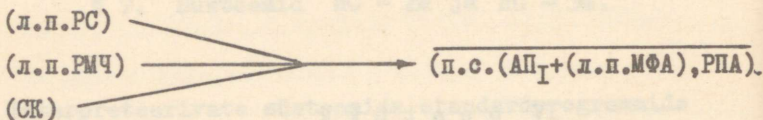
50 sõna salvestamine lühikese aadressiga pesasse



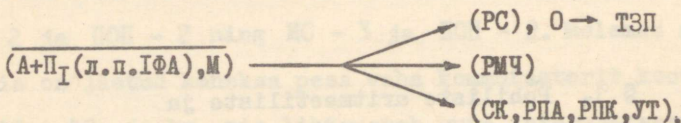
60 poolsõna salvestamine täieliku aadressiga pesasse



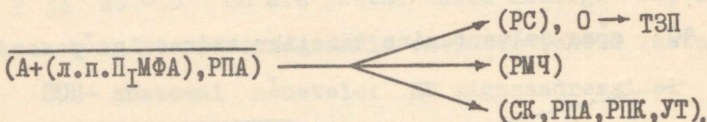
70 poolsõna salvestamine lühikese aadressiga pesasse



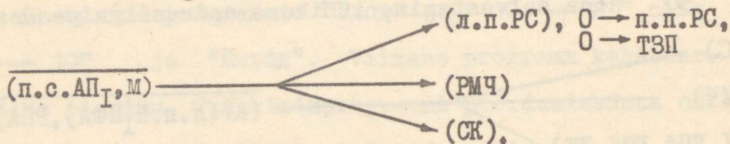
41 sõna toomine täieliku aadressiga pesast registrisse



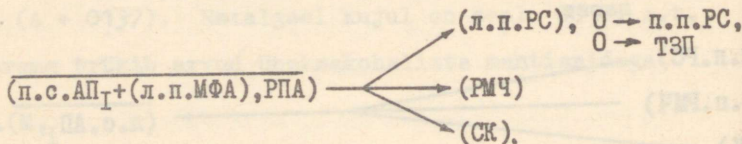
51 sõna toomine lühikese aadressiga pesast registrisse



61 poolsõna toomine täieliku aadressiga pesast registrisse

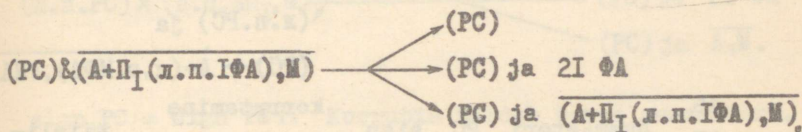


71 poolsõna toomine lühikese aadressiga pesast registrisse



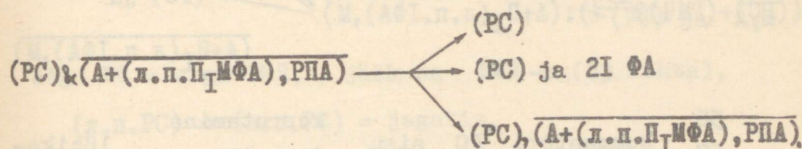
- 42 liita
- 43 lahutada
- 44 moodulid lahutada (PC) ja täieliku
- 45 loogiliselt korrutada

aadressiga sõna



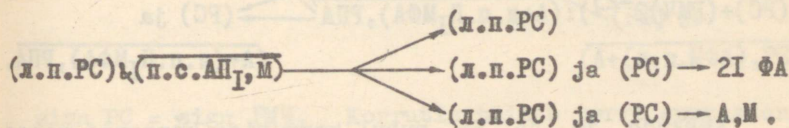
- 52 liita
- 53 lahutada
- 54 moodulid lahutada (PC) ja lühikese
- 55 loogiliselt korrutada

aadressiga sõna



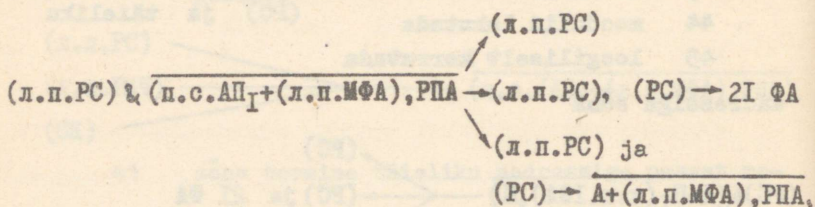
- 62 liita
- 63 lahutada
- 64 moodulid lahutada summaatori PC vasaku
- 65 loogiliselt korrutada

poolsõnaga lühikese aadressiga poolsõna \equiv



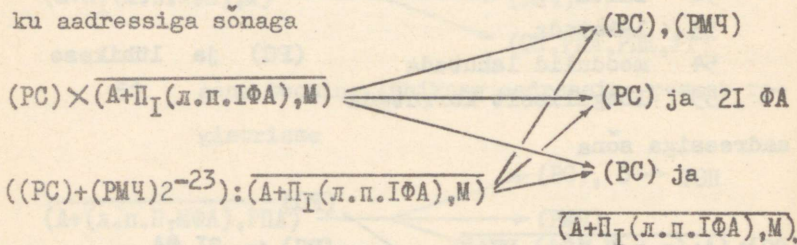
- 72 liita
- 73 lahutada
- 74 moodulid lahutada summaatori PC vasaku

75 loogiliselt korrutada
 poolsõnaga lühikese aadressiga poolsõna [⊗]



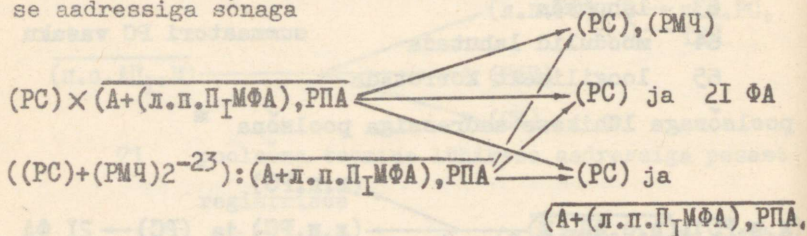
46 summaatori PC sisu korrutamine täieli-
 47 jagamine

ku aadressiga sõnaga



56 summaatori PC sisu korrutamine lühike-
 57 jagamine

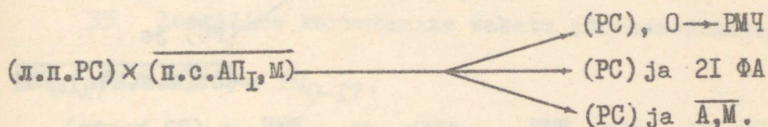
se aadressiga sõnaga



Käskude 46 ja 56 puhul korrutise kõrgemad kohad on PC -s ja madalamad РМЧ -s. Käskude 47 ja 57 puhul jagatis on PC -s ja jääk РМЧ -s. Mõlemal

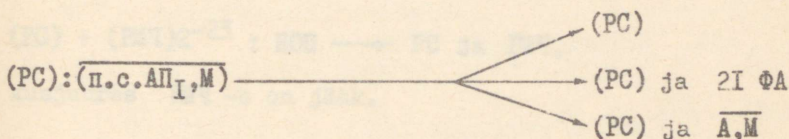
$\text{sign}(PC) = \text{sign}(PM\check{C})$

66 summaatori PC sisu korrutamine täieliku aadressiga poolsõnaga ☒



$\text{sign } PC = \text{sign } PM\check{C}$. Korrutis täidab terve summaatori.

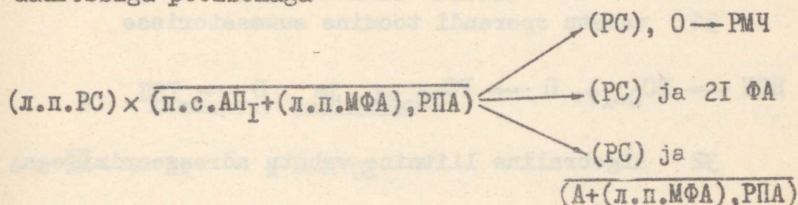
67 summaatori PC sisu jagamine täieliku aadressiga poolsõnaga ☒



$\text{sign } PC = \text{sign } PM\check{C}$, jääk on $PM\check{C}$ -s (23 kohta),

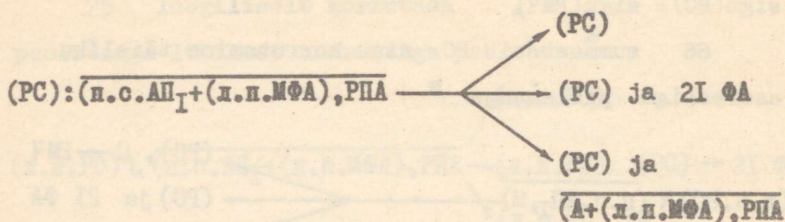
$(\text{л.п.}PC) = (\text{п.п.}PC) = \text{jagatis}$.

76 summaatori PC sisu korrutamine lühikese aadressiga poolsõnaga ☒



$\text{sign } PC = \text{sign } PM\check{C}$. Korrutis täidab terve summaatori.

77 summaatori PC sisu jagamine lühikese aadressiga poolsõnaga ☒



sign PC = sign PMЧ, jääk on PMЧ -s (23 kohta),
 $(X.H.PC) = (H.H.PC) = \text{jagatis.}$

* Käsusisesel salvestamisel ($H_2H_3 = 01; 10$) summaatori sisu salvestatakse tervikuna vastavasse pessa.

§ 2. Vahetu adresseerimisega aritmeetilised ja loogilised operatsioonid.

31 vahetu operandi toomine summaatorisse

$H0\Pi \rightarrow PC_{0-17}, 0 \rightarrow PC_{18-23} \text{ ja } 0 \rightarrow T3\Pi.$

32 algebraline liitmine vahetu adresseerimisega

$(PC_{0-17}) + H0\Pi \rightarrow PC_{0-17}.$

33 lahutamine vahetu adresseerimisega

$(PC_{0-17}) - H0\Pi \rightarrow PC_{0-17}.$

34 moodulite lahutamine vahetu adresseerimisega

$$(PC_{0-I7}) - |HOH| \rightarrow PC_{0-I7}.$$

35 loogiline korrutamine vahetu adresseerimisega

$$(PC_{0-I7}) \wedge HOH \rightarrow PC_{0-I7}.$$

36 korrutamine vahetu adresseerimisega

$$(PC) \times HOH \rightarrow PC \text{ ja } PMY.$$

37 jagamine vahetu adresseerimisega

$$(PC) + (PMY)2^{-23} : HOH \rightarrow PC \text{ ja } PMY,$$

kusjuures PMY -s on jääk.

§ 3. Vahetu adresseerimisega loendamise

ja suunamise käsud.

20 isemuutuv suunamine

$$A + \overline{PM} \rightarrow A; \quad A \rightarrow CK.$$

21 loendamine I

$$\text{kui } \overline{ATM} - I = 0, \text{ siis } K + 2 \rightarrow CK$$

$$\text{kui } \overline{ATM} - I \neq 0, \text{ siis } K + I \rightarrow CK \text{ ja } \overline{ATM} - I \rightarrow \overline{ATM}$$

22 loendamine II

kui $\overline{ATIM} - 1 = 0$, siis $K + 1 \longrightarrow CK$

kui $\overline{ATIM} - 1 \neq 0$, siis $K + 2 \longrightarrow CK$ ja

$$\overline{ATIM} - I \longrightarrow \overline{ATIM}.$$

23 loendamine III

kui $\overline{ATIM} - 1 = 0$, siis stopp

kui $\overline{ATIM} - 1 \neq 0$, siis $K + 1 \longrightarrow CK$ ja

$$\overline{ATIM} - I \longrightarrow \overline{ATIM}.$$

24 suunamine vahe järgi I

(PC) - $\overline{ATIM} \longrightarrow PC$

kui sign (PC) = "+", siis $K + 2 \longrightarrow CK$

kui sign (PC) = "-", siis $K + 1 \longrightarrow CK$.

25 suunamine vahe järgi II

(PC) - $\overline{ATIM} \longrightarrow PC$

kui sign (PC) = "+", siis $K + 1 \longrightarrow CK$

kui sign (PC) = "-", siis $K + 2 \longrightarrow CK$,

26 suunamine vahe järgi III

(PC) - $\overline{ATIM} \longrightarrow PC$

kui sign (PC) = "+", siis stopp

kui sign (PC) = "-", siis $K + 1 \longrightarrow CK$.

(PC) - $\overline{A\overline{P\overline{M}}}$ \rightarrow PCkui sign (PC) = " + ", siis $K + 1 \rightarrow CK$

kui sign (PC) = " - ", siis stopp .

§ 4. Juhtimiskäsud.

10^{3E} modifikaatori muutmine.Muudetava modifikaatori number on Π_1^M .a) $\Pi_2\Pi_3 = 00$. Kui enne käsu täitmist: $\overline{(\text{п.п.}\Pi_1^M\Phi A)} \neq 0$, siis $\overline{(\text{п.п.}\Pi_1^M\Phi A)} - I \rightarrow \overline{(\text{п.п.}\Pi_1^M\Phi A)}$ ja $\overline{A, P\overline{PK}} \rightarrow (CK, P\overline{PK})$; $\overline{(\text{п.п.}\Pi_1^M\Phi A)} = 0$, siis $\overline{(\text{п.п.}\Pi_1^M\Phi A)} - I \rightarrow \overline{(\text{п.п.}\Pi_1^M\Phi A)} = 7777$ ja $(CK) + I \rightarrow (CK)$.b) $\Pi_2\Pi_3 = 01$. Kui enne käsu täitmist: $\overline{(\text{п.п.}\Pi_1^M\Phi A)} \neq 0$, siis $\overline{(\text{п.п.}\Pi_1^M\Phi A)} - I \rightarrow \overline{(\text{п.п.}\Pi_1^M\Phi A)}$, $\overline{(\text{п.п.}\Pi_1^M\Phi A)} + I \rightarrow \overline{(\text{п.п.}\Pi_1^M\Phi A)}$ ja $\overline{A, P\overline{PK}} \rightarrow (CK, P\overline{PK})$; $\overline{(\text{п.п.}\Pi_1^M\Phi A)} = 0$, siis $\overline{(\text{п.п.}\Pi_1^M\Phi A)} - I \rightarrow \overline{(\text{п.п.}\Pi_1^M\Phi A)}$ ja $(CK) + I \rightarrow (CK)$.c) $\Pi_2\Pi_3 = 10$.

$$(\overline{\text{л.п.П}_I\text{M}\Phi\text{A}})+\text{A} \longrightarrow (\overline{\text{л.п.П}_I\text{M}\Phi\text{A}}) \text{ ja } (\text{CK}) + \text{I} \longrightarrow (\text{CK}).$$

11³⁸ tingimusega suunamine,

Kui tingimus on täidetud, siis $\overline{\text{A,П}} \longrightarrow (\text{CK,ППК})$

ja kui tingimus ei ole täidetud, siis $(\text{CK}) + 1 \longrightarrow (\text{CK}),$

M = 0	(PC) = 0
M = 1	(PC) < 0
M = 2	(PC) ≥ 0
M = 3	(PC) ≤ 0
M = 4	(PC) > 0
M = 5	(PC) = 0
M = 6	(PC) = +0
M = 7	TЗП = 1.

12³⁸ tingimusteta suunamine

$\overline{\text{A,П}} \longrightarrow (\text{CK,ППК}), \text{ M} \longrightarrow (\text{ППА}).$

13³⁸ üleminek alamprogrammile

$(\text{CK,ППК,ППА,УТ}) \longrightarrow (\text{A,П}), \overline{\text{A+I, П}} \longrightarrow (\text{CK,ППК}),$

$\text{M} \longrightarrow (\text{ППА}).$

³⁸ Kui suunamise käskudes tehakse katsed registritesse ППК või ППА saata seis 001, siis jäetakse nendesse registritesse endine seis.

§ 5. Informatsiooni vahetamine fikseeritud

aadressiga pesade ja operatiivmälu

pesade vahel.

01 sõna toomine fikseeritud aadressiga pesasse

$(A, PPA) \rightarrow (IM\Phi A)$.

02 poolsõna toomine fikseeritud aadressiga pesa

vasakuks poolsõnaks

$(n.c.A\pi_1, PPA) \rightarrow (n.n.\pi_2\pi_3M\Phi A)$.

03 poolsõna toomine fikseeritud aadressiga pesa

paremaks poolsõnaks

$(n.c.A\pi_1, PPA) \rightarrow (n.n.\pi_2\pi_3M\Phi A)$.

04 fikseeritud aadressiga pesa vasaku poolsõna

viimine pesasse A

$(n.n.\pi_2\pi_3M\Phi A) \rightarrow (n.c.A\pi_1, PPA)$.

05 fikseeritud aadressiga pesa parema poolsõna

viimine pesasse A

$(n.n.\pi_2\pi_3M\Phi A) \rightarrow (n.c.A\pi_1, PPA)$.

06 sõna viimine fikseeritud aadressiga pesast pesasse A

$$\overline{(\Pi\Phi A)} \rightarrow \overline{(A, P\Pi A)}.$$

07 sõna vahetamine fikseeritud aadressiga pesa ja operatiivmälu pesa vahel

$$\overline{(\Pi\Phi A)} \rightleftharpoons \overline{(A, P\Pi A)}.$$

§ 6. Mitteamarilised käsud.

0000 stopp.

Tabloom on informatsioon kohalt A.

0100 suunamine võtmele.

6 - le võtmele, numbritega 0, 1, 2, 3, 4, 5 vastab kuus kahendkohta A_3A_4 . Kui mingile sisselülitatud võtmele vastaval kahendkohal on 1, siis täidetakse järgmine käsk, muidu ülejäämine.

0200 vastus katkestuse täitmisest.

Kui $A_1 = 1$, siis kustutatakse vastava katkestuskanali trigerid registrites P3Π ja PIII.

A_2 - katkestusbloki number (0 - 7),

A_3A_4 - katkestuskanali number.

Kui $A_1 = 2$, siis lõpetatakse katkestuse blokeering ja kustutatakse triger TBC.

0300 seis määramine.

Aritmeetilisel seadmel on 24 klemmi.

1) kui A on paarisarv, siis viiakse nende klemmide seis summaatorisse PC,

2) kui A on paaritu arv, siis viiakse pesa A ja (PC) ei muutu.

0400 järgmise käsu aadressi modifitseerimine.

A - modifikaatori aadress. Käsu A 0400 järel ei toimu programmilist ega informatsioonilist katkestamist, samuti üleminekut vigade otsimise programmile. Kui käsure A 0400 järgneb käsk koodiga 10, siis muudetakse modifikaator pesas A.

0500 nihe summaatori PC sisu järgi.

$(A, PPA) + PM\checkmark \cdot 2^{-23}$ nihutatakse (n.n.PC) koha

võrra: kui sign PC = " + ", siis vasakule

kui sign PC = " - ", siis paremale.

Nihutatud arv jääb registrisse PC ja PM \checkmark

sign (A) \longrightarrow sign (PC) \longrightarrow sign (PM \checkmark).

Enne nihet summaatori ületäitumise koht ja triger T3П kustuvad.

0600 mäluvälja määramine välisseadmete kanalis.

A_4 - välisseadme (УПЧ, АЦПЧ, jt.) number grupilise informatsiooni kanalis,

A_3 - mäluvälja number,

A_1 ja A_2 annavad täiendavat informatsiooni (näiteks YHV kohta vt. pt. II §7 lk.58).

0700 isesünkroniseeriva informatsiooni vahetamine (vt. pt.II §9 punkt 3 lk.66).

1000 valjuhääldaja sisse- ja väljalülitamine,

$A_1 \geq 4$ valjuhääldaja sisse lülitada

$A_1 < 4$ valjuhääldaja välja lülitada

1100 PC ja PMY nihe.

A_1 I kk. = $\begin{cases} 0 - \text{lahtine} \\ 1 - \text{tsükliline} \end{cases}$

A_1 II kk. = $\begin{cases} 0 - \text{ilma märgita} \\ 1 - \text{märgiga} \end{cases}$

A_1 III kk. = $\begin{cases} 0 - \text{aihult PC nihe} \\ 1 - (\text{PC}) + (\text{PMY}) \cdot 2^{-23} \text{ nihe} \end{cases}$

$A_3 A_4$ - nihete arv

A_2 = $\begin{cases} 0 - \text{vasakule} \\ 1 - \text{paremale.} \end{cases}$

Koos märgiga nihutamisel ületäitumiskoht nihkest osa ei võta, ilma märgita nihutamisel võtab.

Ilma märgita PC ja PMY nihkel vasakule $\text{sign}(\text{PMY})$

$\rightarrow \text{sign}(\text{PC})$ ja paremale $\text{sign}(\text{PC}) \rightarrow \text{sign}(\text{PMY})$.

1200 normaliseerimine.

PC ja PMY vaadeldakse ühe registrina.

1. Kui $(T3\Pi) = 0$, siis nihutatakse (PC) ja $(PM4)$ vasakule nii, et $1/2 \leq (PC) < 1$. Pärast nihet $(PC) \rightarrow \overline{A, P1A}$, Summaatorisse jääb nihete arv, $\text{sign}(PC) = "-"$.

Kui arv oli enne normaliseeritud kujul, siis pärast nihet $(PC) = 0000\ 0000$.

Kui (PC) ja $(PM4)$ enne normaliseerimist on nullid, siis pärast nihet $(PC) = 4000\ 7500$.

2. Kui $(T3\Pi) = 1$, siis nihutatakse (PC) ja $(PM4)$ ühe koha võrra paremale, $(PC) \rightarrow \overline{A, P1A}$ ja $0000\ 0001 \rightarrow (PC)$.

1300 arvu koostamine.

1) Kui $\text{sign}(PC) \neq \text{sign}(PM4)$ ja

- a) kui $(PC) \neq 0$, siis lahutatakse $(PC) - 1 \cdot 2^{-23}$ ja $(PM4)$ - st võetakse pöördkood ning $\text{sign}(PC) \rightarrow \text{sign}(PM4)$.
- b) kui $(PC) = 0$, siis $\text{sign}(PM4) \rightarrow \text{sign}(PC)$.

2) Kui $\text{sign}(PC) = \text{sign}(PM4)$, siis ei toimu midagi.

Käsk ei mõjuta trigeri $T3\Pi$ seisust.

1400 kohtade jaotamine.

Jaotatakse summaatori viimasel kuuel kohal asuvad numbrid. Käsu skeemi vt. pt. II § 9 punkt 3 lk. 64.

1500 katkestuse blokeerimise registri PБП täitmine, (A, ПИА) 21 esimest kahendkohta → (PБП), kolm viimast kohta näitavad katkestusbloki numbrit.

1600 $A_1 \neq 0$ seisumääramine

$A_1 = 0$ toitepingete muutmine.

Arvutis on kuus automaatselt muudetavat toitepinge ahelat nominaalidega + 20 V, - 12 V, - 30 V, + 12 V $\Phi 3V$ (ferriitmälu), - 12 V $\Phi 3V$ ja - 30 V $\Phi 3V$. Igaüht neist võib käsu järgi muuta + 15 % või - 15 % nominaalist. Ühe pinge muutmisel käsuga kõik teised võtavad automaatselt nominaalväärtuse. Pinge muutuse näitab A:

A	Toitepinge	
0000	kõik nominaalpinged	
0020	+ 20 V	+ 15 %
0040	+ 20 V	- 15 %
0060	- 12 V	+ 15 %
0100	- 12 V	- 15 %
0120	- 30 V	+ 15 %
0140	- 30 V	- 15 %
0160	+ 12 V	+ 15 % $\Phi 3V$
0200	+ 12 V	- 15 % $\Phi 3V$
0220	- 12 V	+ 15 % $\Phi 3V$
0240	- 12 V	- 15 % $\Phi 3V$
0260	- 30 V	+ 15 % $\Phi 3V$
0300	- 30 V	- 15 % $\Phi 3V$

2M00 kahendinformatsiooni sisseviimine perfolindilt.

A - sisseviidava informatsiooni algusaadress mälus,

PIIA - aadressvälja number,

M - sisend-väljundseadme number ühekordse informatsiooni kanalis.

Kui aadress ja mäluvälja number on perfolindil, siis aadressi A ja väljanumbrit PIIA ignoreeritakse.

3M00 tähestikulis - arvulise silbi viimine arvutisse.

A - pesa aadress, kuhu viiakse silp,

PIIA - mäluvälja number,

M - seadme number ühekordse informatsiooni kanalis,

A_4 III kk. = $\begin{cases} 0 & \text{töö katkestusega} \\ 1 & \text{töö katkestuseta.} \end{cases}$

Silp viiakse pesa A ja summaatori paremasse otsa, ülejäänud kohad kustuvad. Ühe käsuga viiakse mällu üks silp. Perfolint liigub start-stoppp režiimis kiirusega kuni 400 silpi sekundis.

4M00 järgmise käsu salvestamine $\overline{A,PIIK}$ - sse.

M = 0 - jätkub töö salvestatud käsule järgnevast käsust,
M = 1 - stopp.

5M00 käsu kustutamine.

Kustutatakse pesa $\overline{A, PIR}$ sisu.

M = 0 täidetakse järgmine käsk,

M = 1 stopp.

6M00 märgi omistamine.

M = 0 sign(PC) \longrightarrow sign (A)

M = 1 sign (A) \longrightarrow sign (PC)

M = 2 " + " \longrightarrow sign (A)

M = 3 " - " \longrightarrow sign (A)

M = 4 sign (A) muudetakse vastupidiseks.

7M00 tsükliline liitmine.

$\overline{(A, M)}$ + (PC)

Märgikohti liidetakse kui arvukohti. Ületäitumise korral liidetakse 1 (PC) madalamale kohale.

0M30 tähestikulis - numbrilise silbi väljastamine

Väljastatakse $\overline{\Pi, \Pi, A, PIA}$ kaheteistkümmel viimasel kohal olev silp.

M - seadme number ühekordse informatsiooni kanal

$$\Pi_I = \begin{cases} 0 & \text{töö katkestusega ja üleminekuga alamprogrammile,} \\ 1 & \text{töö katkestuseta.} \end{cases}$$

Arvuti juhtimislauas oleva sisend - väljundseadme jaoks

M = 0, väljastamise viisi näitab Π .

$\Pi = 5$ trükkimine

$\Pi = 6$ perforeerimine

$\Pi = 7$ trükkimine ja perforeerimine.

§ 7. Informatsiooni vahetamise käsud

magnetlindmäluga ja paljukanalilise

muundajaga.

14 magnetlinde poole pöördumine (vt. pt. II § 6).

15 informatsiooni vahetamine grupilise informatsiooni kanaliga. Käsu üldkuju vt. pt. II § 7. lk. 57.

Paljukanalilise muundaja (VII) jaoks kasutatakse käsu kohta järgmiselt:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
B				N							$\Pi_1 \Pi_2 \Pi_3$			M			15						
A ₁											A ₂												

N - näitab VII kanali numbrit, mille kaudu ta on ühendatud juhitava objektiga,

B - näitab teisenduse liiki,

Π_3 - teisenduse iseärasusi.

N, B ja Π_3 väärtused mitmesuguste teisenduste jaoks on toodud Lisas 3, kusjuures teisenduste märkimiseks

on kasutatud Lisas 1 toodud lühendatud tähistusi.

Π_I näitab arvuti "B - 30" ja $\mathbb{N}\mathbb{N}$ vahelise side liiki.

$$\Pi_I = \begin{cases} 1 & \text{töö katkestuseta} \\ 0 & \text{töö informatsioonilise katkestusega.} \end{cases}$$

Π_2 näitab, kas toimub tavaline teisendus või kontrollteisendus $\mathbb{N}\mathbb{N}$ - sisese kontrolliga.

$$\Pi_2 = \begin{cases} 0 & \text{toimub tavaline teisendus} \\ 1 & \text{toimub kontrollteisendus.} \end{cases}$$

Järgnevas vaatleme konkreetseid teisendusi ja teisendatud või teisendamisele tuleva informatsiooni kujutamist ferriitmälu pesas (teisenduste numbrid vastavad Lisa 3 toodud numbritele).

1. H/\mathbb{A} - pinge teisendamine kahendarvuks,
2. T/\mathbb{A} - voolu teisendamine kahendarvuks,
5. H_M/\mathbb{A} - nõrkade pingete ja voolude teisendamine kahendarvuks,
4. $H_{\mathbb{N}}/\mathbb{A}$ - vahelduvpinge amplituudi teisendamine kahendarvuks.

Teisenduste 1, 2, 5, 4 tulemus paikneb mälu pesas järgmiselt:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
0											0	0												

alati teisendatud täitub nullidega
null suurus 0 informatsiooni usaldusväärsus

Iga teisenduse liigi jaoks on antud teisendamispiirkond.

Teisendatud suurus saadakse kohtadele 1 - 10 järgmiselt: teisendamispiirkonna alumisele piirile vastab kahendarv 0000000000 ja ülemisele piirile vastab kahendarv 1111101000 (see on 1000_{10} kümnenda koha ühikutes). Ülejäänud väärtustele vastavad teisendatud suurused saadakse lineaarse interpoleerimise teel. Kümnenda koha ühikule vastab seega tuhandik teisendamispiirkonnast. Maksimaalne arv, mida nendel kümnel kohal kujutada saab, on $1111111111_2 = 1023_{10}$ kümnenda koha ühikutes. Kui teisendatud suurus tuleb suurem sellest arvust, siis kirjutatakse pees ikkagi arv 1111111111. Seega saame teisendada ka natuke suuremaid arve kui teisendamispiirkond näitab.

Näide: teisendamispiirkonna $-5\text{ V} \div +5\text{ V}$ korral pingele -5 V vastab arv 0000000000 ja maksimaalne pingeväärtus, mis teisendatakse õigesti, on $+5,23\text{ V}$; või piirkonna $0 \div +5\text{ mA}$ korral vastab voolutugevusele $5,11\text{ mA}$ ja temast suurematele voolutugevustele arv 1111111111_2 .

3. H_p/Δ - releeline pingeline \rightarrow kahendarvuks,

6. H_{PM}/Δ - vahelduvpingeline \rightarrow kahendarvuks,

Teisenduste 3 ja 6 tulemus paikneb pesas järgmiselt:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	α_1	α_2	α_{12}	0	0	0	0	0

teisendatud suurus täitub nullidega

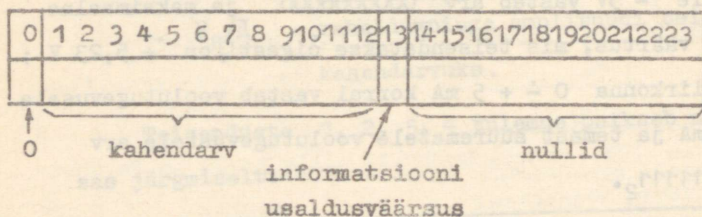
Teisenduse alumisel nivool $\alpha_i = 0$, ülemisel $\alpha_i = 1$ ($i = 1, 2, \dots, 12$). Märkikohta ei kasutata. Pinge polaarsus tehakse kindlaks aadressi järgi.

Teisenduse B_{03} puhul vastab pingele -10 V $\alpha_i = 0$ ja pingele $+0\text{ V}$ $\alpha_i = 1$.

Teisenduse B_{04} puhul (spetsiaalsed kanalid) vastab pingele 0 V $\alpha_i = 0$ ja pingele -10 V $\alpha_i = 1$.

7. Π_{nap}/Π — paralleelne numbriline informatsioon kahendarvuks.

Paralleelne arvuline kahendinformatsioon kirjutatakse pesasse järgmiselt:



8. B/Π — aeg —> kahendkümnendkoodi

Aeg kirjutatakse kahendkümnendkoodis pesa kõigile 24 kahendkohale järgmiselt:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1) kui $\Pi_2\Pi_3 = 00$																							
Kümned tunnid				Tunnid				Kümned minutid				Minutid				Kümned sek.				Sekundid			
2) kui $\Pi_2\Pi_3 = 01$																							
Minutid				Kümned sekun- did				sekun- did				Sajad milli- sek.				Kümned milli- sek.				Millisek.			

9. Δ/H - viiekohaline kahendarv pingeks.

Pesas 1 - 10 kohal olev kahendarv teisendatakse alalispingeks, mis on proportsionaalne viie madalama kahendkoha ulatuses. Igal kanalil on kaks väljundit. Ühelt saadakse positiivne pinge ja teiselt sama absoluutväärtusega negatiivne pinge. Seega toimub pinge teisendamine järgmise valemi järgi:

$$U \text{ (volti)} = \begin{cases} \frac{\pm k \cdot 10,23}{31}, & \text{kui } k \leq 31 \\ \pm 10,23, & \text{kui } k > 31, \end{cases}$$

kus k on kümnendarvuks teisendatud lähte arv.

10. Δ/H_{TOV} - kahendarv \rightarrow täpseks alalispingeks.

Kümnekohaline kahendarv teisendatakse proportsionaalseks alalispingeks kogu arvu ulatuses järgmiselt:

$$U \text{ (volti)} = - \frac{k}{100},$$

kus k on kümnendarvuks teisendatud lähtearv. See teisendus toimub kahe järjestikuse käsuga $K = 15$. Esimeses käsus näidatakse kanali number, mis asub tabelis murrujoone peal ja teises käsus murrujoone alune kanali number. Esimese käsu aadressi osas (2. pesa) näidatud pesas peab olema teisendatava arvu 5 esimest kahendkohta 6. - 10. kohal ja teises käsus näidatud pesas 5 viimast kahendkohta 6. - 10. kohal. Kohtadel 1 - 5 peavad mõlemal juhul olema nullid. Need kaks informatsiooni pesa võivad olla ka ümberpööratud järjekorras, st. enne võib välja anda viimased kahendkohad ja siis esimesed. Kahe käsu vahel võib olla teisi käske, st. nad ei pea olema järjest.

Esimese käsu informatsioon:

k

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Nullid											Nullid												

5 esimest kahendkohta

Teise käsu informatsioon:

n

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Nullid											Nullid												

5 viimast kahendkohta

Esimese kanali kaudu niisuguse informatsiooni väl-
jaandmise käsk oleks seega järgmine:

1. pesa: 4345 4 0 15

2. pesa: k k

3. pesa: 4352 4 0 15

4. pesa: n n

11. $\mathbb{D}/\mathbb{H}_{\text{BIP}}$ - algebraline kahendarv bipolaarseks
pingeks.

Pesas 0. ÷ 10. kahendkohal olev arv teisendatak-
se kahe polaarsusega muutuvaks alalispingeks. Nullis ka-
hendkoht näitab pinge märki. Pinge on proportsionaalne
nelja viimase kahendkoha ulatuses.

$$U \text{ (volti)} = \begin{cases} \frac{k \cdot 10,23}{15} & , \text{ kui } k \leq 15 \\ 10,23 & , \text{ kui } k > 15, \end{cases}$$

kus k on kümnendarvuks teisendatud kahendarv.

12. \mathbb{D}/\mathbb{H}_p - kahendarv kahenivooliseks releeli-
seks pingeks.

Teisendamisele tulev informatsioon asub pesas
kohtadel 1 ÷ 10. Kui kohtadel 1 ÷ 5 on kas või üks "1",
siis antakse välja maksimaalpinge. Kui kohtadel 1 ÷ 10

on "0", siis väljundklemmidel pinge puudub.

13. \bar{A}/\bar{H}_B -- kahendarv ajaimpulsiks

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
0	α_1	α_2	α_{11}	α_{12}	0	0	0	0

$\underbrace{\hspace{10em}}_{\tau}$
 $\underbrace{\hspace{10em}}_{\text{Nullid}}$

α_{11} näitab kõige madalama kahendkoha vastavust aja-intervallile.

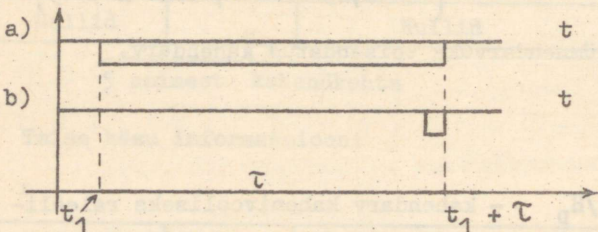
$\alpha_{11} = 0$ - kõige madalama kahendühiku väärtus on üks millisekund.

$\alpha_{11} = 1$ - kõige madalama kahendühiku väärtus on üks.

Antakse välja:

1) kui $\alpha_{12} = 0$, siis on ühekordne teisendus;

- a) pingepulss pikkusega τ ,
- b) standardne impulss ajavahemiku τ järel

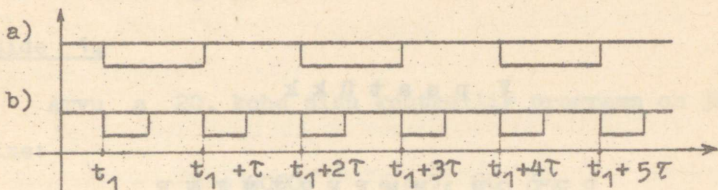


t_1 - teisenduse algusmoment.

2) kui $\alpha_{12} = 0$, siis toimub teisendus pidevalt

- a) impulsid pikkusega τ vahelduvad vaheaegadega pikkusega τ ,

b) standardsed impulsid τ , 2τ , 3τ jne. järele.



t_1 - teisenduse algusmoment.

Impulsid a) ja b) saadakse paralleelselt eri väljunditest.³⁸

14. $\mathbb{A}/\text{By}\ddot{\text{U}}$ - kahendarvu ja kanali numbri andmine puhverseadmesse.

Puhverseadmesse antakse käsus oleva kanali numbri N 8 kahendkohta ja pesas 1. - 6. kahendkohal oleva informatsiooni. Puhverseadmest võib arve välja anda impulssidena (impulsi pikkus 0,1 - 0,2 sek), kusjuures "1" - le vastab pinge olemasolu, "0" - le selle puudumine.

³⁸ Peale teisendusega \mathbb{A}/M_B saadavate ajaimpulsside võib "B - 31" insenerpuldist saada mitmesuguseid pidevalt väljaantavaid ajasignaale vahemikus 62,5 μs (16 kHz) kuni 4096 tundi (~ 170 ööpäeva).

V p e a t ü k k

P R O G R A M M E E R I M I N E

§ 1. Elementaarseid

programmeerimisvõtteid.

Järgnevas paigutame programmide näited tinglikesse pesadesse $k + 0000$, $k + 0001$, . . . , algandmed ja konstandid pesadesse $c + 0000$, $c + 0001$, . . . ning arvutuste tulemused pesadesse $b + 0000$, $b + 0001$, . . .

Summaatori sisu saab kustutada käsuga $\langle 0 \rangle 00 51$. Selle võtte puhul on vaja, et mälu mingis pesas oleks salvestatud konstant 0. Kiirem kustutamise käsk on $0000 00 31$. Summaatori sisu kustutamiseks nihutamisega on käsk $2030 11 00$. Pesa A sisu saab kustutada käsuga $A 50 00$. Mingi konstandi C pesa saatmiseks on näiteks järgmine võimalus

$k + 0000$ A 4000

$k + 0001$ C .

Loogilise (märgita) nihutamisega saab kontrollida ferriitmälu mingi pesa mistahes kahendkoha sisu.

Näide 1.

Arvu a 20. koha sisu kontrolliv programm on järgmine:

k + 0000	< a >	00	51
k + 0001	2024	11	00
k + 0002	k + i	01	11 .

Selle programmiga nihutatakse summaatorisse toodud arv a 20 koha võrra vasakule. Arvus 20. kohal olnud number läheb summaatori nullindale (märgi) kohale. Programmiga nihutamiskonstanti muutes võib järjest kontrollida ferriitmälu pesa kõiki kohti (vt. §2. Näide 3.).

Fikseeritud komaga arvu aritmeetiline nihutamine + k koha võrra on ekvivalentne tema korrutamisega konstandiga 2^{+k} . Seejuures hoitakse küll kokku üks konstandipesa ja aega, kuid tuleb arvestada ületäitumise võimalust.

Aritmeetilise nihutamise käsuga 0130 1100 eraldatakse summaatoris oleva arvu märk. Märgi võib eraldada ka loogilise korrutamise abil, näiteks vahetu adresseerimise käsuga 4000 0035.

§ 2. Tsükliid.

Modifikaatori muutmise ja kontrollimise käsu (koodiga 10) abil saab korrata programmi teatud osi ja muuta vajaduse korral käskude aadresse.

Näide 1.

Arvutada suurused $x_i = A_i \cdot B_i$ ($i = 1, 2, 3$). (1)

Programm valemi (1) realiseerimiseks on järgmine:

c + 0000	A ₁
c + 0001	A ₂
c + 0002	A ₃
c + 0003	B ₁
c + 0004	B ₂
c + 0005	B ₃
c + 0006	0000 0002

k + 0000	c + 0006	01	01
k + 0001	c + 0000	01	51
k + 0002	c + 0003	01	56
k + 0003	b + 0000	01	50
k + 0004	k + 0001	11	10

Käsk k + 0000 täidab modifikaatori 10A. Käsud k + 0001 ÷ k + 0003 tulevad esimesel tsükli läbimisel täitmisele nii nagu nad on programmis kirjutatud (sest PM = 0). Käsk k + 0004 liidab modifikaatori

1 0A vasakule poolsõnale PM ühe ja paremast poolsõnast CM lahutab ühe. Teistkordsel tsükli läbimisel tuleb programm täitmisele kujul

k + 0001	c + 0001	00	51
k + 0002	c + 0004	00	56
k + 0003	b + 0001	00	50

ja kolmandal korral kujul

k + 0001	c + 0002	00	51
k + 0002	c + 0005	00	56
k + 0003	b + 0002	00	50 .

Peale kolmandat tsükli läbimist on $CM = 0$ ja käsk k + 0004 ei anna juhtimist pesale k + 0001, vaid käsule k + 0005.

Näide 2.

Arvutada $y = \sum_{i=1}^5 x_i$, kui x_i paiknevad pesades üle ühe. Programm on järgmine:

c + 0000	x_1
c + 0002	x_2
c + 0004	x_3
c + 0006	x_4
c + 0010	x_5
c + 0011	0000 0004
b + 0000	0000 0000

k + 0000	c + 0011	01	01
k + 0001	c + 0000	01	51

k + 0002	b + 0000	20	52
k + 0003	0001	21	10
k + 0004	k + 0001	11	10
k + 0005	0000	00	00 .

Käsud k + 0003 ja k + 0004 kindlustavad modifikatsiooniregistri PM muutmise kahe kaupa.

Teine variant:

k + 0000	b + 0000	50	00
k + 0001	0001	40	00
k + 0002	0000	00	04
k + 0003	c + 0000	01	51
k + 0004	b + 0000	20	52
k + 0005	0002	21	10
k + 0006	k + 0003	01	10
k + 0007	0000	00	00 .

Selle variandi puhul jääb vabaks konstandi pesa c + 0011. Konstant formeeritakse programmi sees, mis vähendab eksemplide võimalusi. Kuna pesa b + 0000 sisu kustutamine on programmi sees, siis võib olla kindel, et pesas b + 0000 on tõesti 0.

Näide 3.

Kontrollida pesas A paikneva arvu kõiki kahendkohti ja teha kindlaks nulli sisaldavate kohtade arv.

b + 0000 0000 0000

c + 0000	0000 0026		
k + 0000	c + 0000	01	01
k + 0001	A	00	51
k + 0002	0001	04	00
k + 0003	2001	11	00
k + 0004	k + 0007	01	11
k + 0005	0001	00	31
k + 0006	b + 0000	20	52
k + 0007	k + 0001	11	10
k + 0010	0000	00	00 .

Tsükli esimesel läbimisel liidetakse käsu

k + 0003 aadressiosale 0000, teine kord 0001 jne. See-
ga nihutatakse arvu (A) vastavalt 1, 2, 3 jne. koha
võrra vasakule ja kontrollitakse järjekorras kõikide
kohtade sisud. Nullide arv fikseeritakse pesas b + 0000
üheteistkümnenda koha ühikutes.

Näide 4.

Olgu vaja paigutada ümber pesades 0100, 0110,
0120, 0130, 0140 paiknevad arvud vastavalt pesadesse
0200, 0204, 0210, 0214, 0220. Programm on järgmine:

c + 0000	0000 0004		
k + 0000	0002	50	00
k + 0001	c + 0000	01	01
k + 0002	0100	01	51

k + 0003	0200	02	50
k + 0004	0004	22	10
k + 0005	0010	21	10
k + 0006	k + 0002	01	10
k + 0007	0000	00	00.

Käsk k + 0005 muudab registri PM sisu kümne kaupa, mis kindlustab arvude valimist pesadest sammuga kümme. Tsükli loendajat CM kontrollib ja vähendab käsk k + 0006. Salvestamine toimub modifikaatori 2ΦA abil, mis kindlustab salvestamise pesadesse sammuga neli.

Näide 5.

Paigutada ümber arvud pesadest c + 0000, c + 0001, c + 0002, c + 0003, c + 0004, c + 0005 vastavalt pesadesse c + 0025, c + 0024, c + 0023, c + 0022, c + 0021, c + 0020. Programm on järgmine:

c + 0017	0000	0005
k + 0000	c + 0017	01 01
k + 0001	c + 0017	02 01
k + 0002	c + 0000	01 51
k + 0003	c + 0025	02 50
k + 0004	k + 0005	11 10
k + 0005	7777	22 10
k + 0006	k + 0002	02 10 .

Modifikaatori 1ΦA abil valitakse arve ühe kaupa

kasvavate aadressidega pesadest, 2 ΦA abil salvestatakse need ühe kaupa kahanevate aadressidega pesadesse. Negatiivse juurdekasvu annab käsk k + 0005.

Näide 6.

Arvutada funktsiooni $z = \frac{p x_i^2}{y_j}$ väärtuste

tabel, kui $i = 1, 2, \dots, n$ ja $j = 1, 2, \dots, m$.

Arvud x_i paiknegu pesades $d + i$ ja y_j pesades $a + j$. Tulemused paigutada pesadesse $b + i + j$. Argumentide x_i valimine toimub välise, y_j valimine sisemise tsükliga. Programm on järgmine:

c + 0000	0001	n - 1
c + 0001	0001	m - 1
c + 0002	p	
k + 0000	0003	50 00
k + 0001	c + 0000	01 01
k + 0002	c + 0001	02 01
k + 0003	d + 0000	01 51
k + 0004	d + 0000	01 56
k + 0005	c + 0002	00 56
k + 0006	a + 0000	02 57
k + 0007	0003	04 00
k + 0010	b + 0002	02 50
k + 0011	k + 0003	12 10
k + 0012	m	23 10

k + 0013 k + 0002 11 10

k + 0014 0000 00 00 .

Arve x_i valitakse modifikaatori 1 ΦA ja arve y_j modifikaatori 2 ΦA abil. Salvestatakse modifikaatorite 2 ΦA ja 3 ΦA abil.

Näide 7.

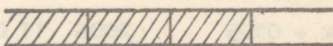
Salvestada telegraafi koodis perforeeritud silbid pesadesse 5000 - 6000. Programm on järgmine:

k + 0100	k + 0127	04 01	
k + 0101	k + 0130	50 00	
k + 0102	k + 0107	40 00	} isemuutuva ülemineku taastamine
k + 0103	k + 0104	04 20	
k + 0104	k + 0106	14 10	massiivi lõpu kontroll ja sona ümberadresseerimine
k + 0105		00 00	Peatus, mälutsoon on täis.
k + 0106	0000	30 00	silbi sisestamine
k + 0107	k + 0104	04 20	isemuutuv üleminek
k + 0110	2022	11 00	← 18
k + 0111	k + 0130	70 00	S + (tööpesa) → tööpesa
k + 0112	k + 0130	00 50	
k + 0113	k + 0106	00 12	
k + 0114	2014	11 00	← 12
k + 0115	k + 0130	70 00	S + (tööpesa) → tööpesa
k + 0116	k + 0130	00 50	

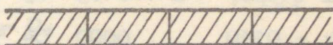
k + 0117	k + 0106	00 12
k + 0120	2006	11 00
k + 0121	k + 0130	70 00
k + 0122	k + 0130	00 50
k + 0123	k + 0106	00 12
k + 0124	k + 0130	70 00
k + 0125	0000	04 50
k + 0126	k + 0101	00 12
k + 0127	4777	10 01
k + 0130		

← 6

S + (tööpesa) → töö-
pessa



S + (tööpesa) → mällu



Programmis toimub massiivi lõpu kontroll alguses, seepärast võetakse loendaja (П.П.4ΦА) vastavalt ühe võrra suurem ja aadressiosa (Л.П.4ΦА) ühe võrra väiksem. Käsk k+0107 suunab esimesel korral käsule k+0110, teisel korral käsule k+0114, kolmandal korral käsule k+0120 ja neljandal korral käsule k+0124. Vastavates blokkides nihutatakse silpi 18, 12 ja 6 koha võrra vasakule ning käsuga 7000 "pakitakse" neli silpi ühte tööpesa k+0130. Käsk k+0125 salvestab need mällu.

Programm on käsu АТМ 20 kasutamise illustreerimiseks. Programm pole antud probleemi jaoks kõige ökonoomsem.

Näiteks oleks sama ülesande jaoks teine programm:

k + 0100	k + 0115	04 01	
k + 0101	k + 0114	50 00	tööpesa kustutamine

k + 0102	k + 0113	05 01	5. modifikaatori loendaja taastamine
k + 0103	0000	30 00	silbi sisestamine
k + 0104	k + 0114	70 00	silpide "pakkimine" ühte pesa
k + 0105	2006	11 00	
k + 0106	k + 0114	00 50	
k + 0107	k + 0103	05 10	kas pesa on täis
k + 0110	0000	04 50	täidetud pesa salvestamine
k + 0111	k + 0101	14 10	kas massiiv on täis
k + 0112	0000	00 00	peatus (massiiv on sees)
k + 0113	0000	00 03	konstant silpide "pakkimise" tsüklile
k + 0114			tööpesa
k + 0115	5000	10 00	konstant massiivi tsüklile

§ 3. Iteratsioonitsüklid.

Iteratsioonitsükli te korral ei tea me ette, mitu korda on vaja vastavat menetlust rakendada. Sel korral on ette antud hinnangud, mille alusel toimub arvutuskäigu suunamine.

Näide 1.

Olgu vaja arvutada piirväärtus $x = \lim_{n \rightarrow \infty} x_n$, kusjuures suurused x_n arvutame suurustest x_{n-1} ja

kindlatest konstantidest teatud kindlal viisil, s.t.

$$x_n = f(x_{n-1}), \quad (n = 1, 2, \dots). \quad (1)$$

Piirväärtuse x asemel võtame tema lähisväärtuse x_n , kui on täidetud tingimus

$$\begin{aligned} |x_n - x_{n-1}| &\leq \varepsilon, \quad \text{ehk} \\ |x_n - x_{n-1}| - \varepsilon &\leq -0. \end{aligned} \quad (2)$$

Kui tingimus (2) pole täidetud, arvutame valemist (1) suuruse x_{n+1} jne.

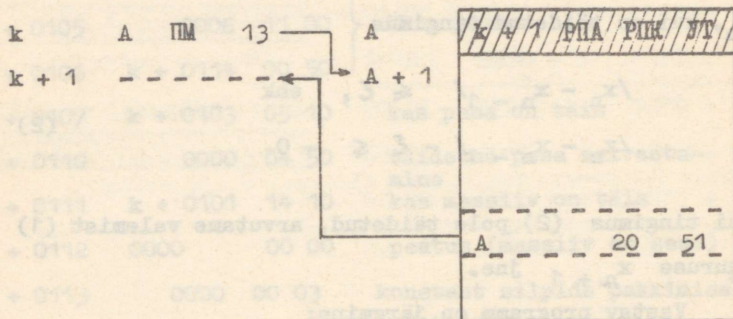
Vastav programm on järgmine:

c + 0000	ε		
b + 0000	x_{n-1}		
b + 0001	x_n		
k + 0000	-----		valemi f(x) jär-
-----	-----		gi arvutamine
			(itereerimine)
k + 1 + 1	b + 0000	00	51
k + 1 + 2	b + 0001	00	54
k + 1 + 3	c + 0000	00	53
k + 1 + 4	k + 0000	04	11
k + 1 + 5	arvutus jätkub		.

§ 4. Standardprogrammide struktuur.

1) Kõige lihtsam standardprogramm kujutab endast

terviklikku programmi, mille poole pöördatakse SP poole pöördumise käsuga A ПМ 13 ja millest väljutakse kas käsuga A 20 51 või A 2M 41:



Pöördumise käsuga salvestatakse informatsioon tagasipöördumiseks pesa A ja alustatakse SP täitmist pesast $A+1$. Käsk $A \text{ 20 } 51$ kindlustab tagasipöördumise põhiprogrammi käsule pesast $k+1$.

2) Soovitav on vormistada standardprogrammeid ühtse massiivina koos konstantide ja tööpesadega. Siis on neid kergem lülitada põhiprogrammi ja SP töö teineteist ei sega.

Arvuti "ВНИИЭМ - 3" programmides on otstarbekas kasutada teatavaid spetsialiseeritud mälupeesi (üldkujul kõik fikseeritud aadressidega pesad) ja oleks ülearune kitsendus piirata nende kasutamist standardprogrammides. Kuid standardprogrammidel nõutakse, et nad säilitaksid ja pärast oma töö lõppu taastaksid üldkasutatavate mälupeaside sisu, mille nad võivad rikkuda.

Järelikult peab SP olema järgmise struktuuriga:

A	tagasipöördumise andmete säilitamise pesa
A + 1	üldkasutatavate mälupesade sisu salvestamine standardprogrammi sisestesse tööpesadesse
	S T A N D A R D P R O G - R A M M I K E H A
	üldkasutatavate mälupesade sisu taastamine
A	20 51
	konstantide pesad ja programmisisesed tööpesad

3) Kuna terve rida üldkasutatavaid mälupesid võivad olla kasutusel eranditult kõigis mingi programmeerimissüsteemi standardprogrammides (näiteks 1 ΦA ja 5 ΦA süsteemide WC - I, WC - 2 ja WC - 3 korral), siis on otstarbekas organiseerida nende salvestamine ja taastamine pöördumisel. Standardprogramm ise säilitab ja taastab ülejäänud üldkasutatavate mälupesade sisu (kui neid on kasutusel). Programmeerimissüsteemides toimub SP poole pöördumine reeglina interpreteeritava käsu A 30 63 abil. Pöördumise ja SP täitmise skeem

oleks järgmine:

PÕHIPROGRAMM

k + 0000 A 30 63

k + 0001 ← *

STANDARDPROGRAMM

A + 0000			
A + 0001	üldkasutatavate		
-----	tööpesade säili-		
-----	tamise pesad		
A + j	täiendavate üld-		
-----	kasutatavate pe-		
-----	sade salvestamine		
A + k	standardprogrammi		
-----	keha		
A + l	täiendavate pesa-		
-----	sade sisu taastami-		
-----	ne		
A + m		00	12
A + m + 1	programmisisised		
-----	konstandid ja töö-		
-----	pesad		

PROGRAMMEERIMISE
SÜSTEEMI PROGRAMM

B+0000			

B+r	pöördumisope-		
-----	raator		
B+s	taastamisoperaa-		
-----	tor		
B+p	A	20	51

4) Standardprogramme on mugav kasutada, kui nad on töö-
kõlblikud suvalisel kohal mälus. Interpreteerivates

süsteemides MC - I, MC - 2 ja MC - 3 saavutatakse SP sõltumatus asukohast nii, et algusaadressist sõltuvate käskude aadressiosa modifitseeritakse 5 ΦA abil (vt. pt.III §1 lk.72).

ΠΠ - süsteemis (MC - 2M jne.) muudetakse vajalikud aadressid enne esimest pöördumist automaatselt skaala abil (vt. pt.III §3 lk.77).

5) Reeglina on kõik standardprogrammid valmistatud tööks massiividega. Pöördumisel tuleb seega ära näidata töödeldava massiivi pikkus ja algus (vt. Lisa 2).

Massiivi töötlemise tsükkel organiseeritakse standardprogrammides tavaliselt modifikaatori 1 ΦA abil.

§ 5. Kontrollsüsteemidest.

Arvutis "BHИИЭМ - 3" kuuluvad aparatuurse kontrolli alla registritevahelised informatsiooni vahetusel, perfolindilt sisseviimine, aritmeetilise seadme töö ning informatsiooni vahetamine ferriitmäluga ja magnetlindiga.

Informatsiooni üleandmiste kontrolliks võrreldakse järkude kaupa üleantud ja vastuvõetud sõnade ühtlasevust ja üleantud sõnade ühtede summa paarsust.

Esimest moodust kasutatakse enamikul juhtudel ja see võimaldab kõigi ühe sõna üleandmisel tekkinud vigade avastamist. Samal ajal on niisugune meetod kiire, kuna ta ei nõua eelnevalt mingite kontrolljärkude moodustamist nagu paarsuse kontroll. Üleantava ja vastuvõetava sõna järkude kontrolli skeem on ehitatud diodelementide "ja" ja "või" abil. Iga kahendjärgu jaoks kasutatakse kahte elementi "ja", mis annavad välja signaali, kui üleantava ja vastuvõetava sõna vastavatel kahendkohadel olevad seisud ei ühti.

Paarsuse kontrolli kasutatakse ainult neil juhtudel, kui see on tingitud seadme eripärast, eeskätt siis kui puudub võimalus alginformatsiooni võrdlemiseks. Nii on see näiteks perfolindilt ja magnetlindilt sissetoomisel, nihutamisel, ferriitmälu jaoks korrigeeriva koodi formeerimisel.

Aritmeetilise seadme töö kontrollimiseks on summaator PC ja osaliselt register PMU dubleeritud kontrollregistris KP. Kontrollregistri kasutusele võtmine suurendab küll märkimisväärselt arvuti aparatuuri, kuid võimaldab teostatavate operatsioonide täielikku kontrollimist ning kõigi üheaegselt tekkinud vigade avastamist igasuguste võimalike operandide kombinatsioonide korral.

Dubleerimiseks on sisse toodud:

a) kontrollregister KP ja ülekannete kontrolli kogumik, mis töötavad summaatoriga paralleelselt,

b) summaatori PC ja kontrollregistri KP ühtivuse kontrollimise skeem, mis teatab veast, kui PC \neq KP.

Registrist PMЧ on dubleeritud 4 esimest kohta. Nihutamisi kontrollitakse paarsuse kontrolli abil peale nihutamistehte sooritamist.

Ferriitmälu töökindlus nõuab erilist tähelepanu, kuna informatsiooni riknemine magnetsüdamikel vigase lugemise või salvestamise tõttu võib viia taastamatule informatsiooni kaotusele. Seepärast on arvutis "ВНИИЭМ - 3" täiendatud ferriitmälu töökindluse suurendamiseks aparatuuri, mis võimaldab mitte ainult vigu avastada, vaid ka parandada.

Arvutis "ВНИИЭМ - 3" salvestatakse informatsiooni operatiivmälus Hemmingi koodis, mis võimaldab avastada kahekordseid ja parandada ühekordseid vigu ning nõuab minimaalselt lisaseadmeid. Kahekümmeneljakohaliste sõnade salvestamiseks kasutatakse kolmekümnekohalisi pesi. Viis lisakohta on Hemmingi koodi jaoks ja üks paarsuse kontrolli jaoks. Tehniliselt moodustatakse Hemmingi kood standardse paarsuse arvutamise skeemiga enne mällu kirjutamist. Peale mälust lugemist näitab Hemmingi kood vigase kahendkoha või teatab kahekordsest veast. Hemmingi koodi kasutamine nõuab umbes 30 % lisaseadmeid,

kuid kindlustab ferriitmälu suure töökindluse. Prakti-
liselt töötab arvuti "ВНИИЭМ - 3" normaalselt ka siis,
kui tal on puudu mistahes üks lugemise võimendaja või
mälu ferriitsüdamik.

Avastatud vead fikseeritakse kontrollimpulsside ja
vastava trigerite süsteemi abil. Igale arvutis olevale
kontrollskeemile on vastavusse seatud oma vea triger ja
kontrollimpulss. Kontrollimpulsid väljastatakse ühe tak-
ti võrra hiljem nendest funktsionaalsetest impulssidest,
mida kontrollitakse. Viivitus on vajalik selleks, et
funktsionaalse impulsi mõjutatavad elemendid jõuaksid
püsivasse asendisse.

Vea esinemisel viib kontrollimpulss vastava vea
trigeri seisu "1". Iga taktiga moodustatakse samaaegselt
eelmise takti kontrollimisega uued funktsionaalsed im-
pulsid, mille toimet kontrollitakse järgmise takti ajal.
Tänu niisugusele ajalisele jaotusele ei kulu kontrolli-
misele lisa-aega.

Vea avastamise korral tuleks arvuti peatada ja vi-
ga käsitsi parandada. Juhtimisarvutis aga ei ole alati
ettenägematu peatus lubatud. Seepärast arvuti

"ВНИИЭМ - 3" vea esinemise korral ei peatu, vaid asub
täitma varem ettevalmistatud korrigeerivat programmi.

Tehnilised tingimused ei sea korrigeerivale prog-
rammile mingeid kitsendusi. Sõltuvalt programmeerija
suvast võib korrigeeriv programm lasta korrata vea and-

nud programmiosa, üle minna teise algoritmi järgi arvutamisele, anda objekt käsitsi juhtimise alla või hoopis seisma jätta.

Samal ajal üleminek korrigeerivale programmile toimub automaatselt koos vajaliku informatsiooni andmisega. On olemas tehnilised vahendid (aparatuur), mis tõstavad arvuti korrigeerimisvõimet.

Arvutis "БВММЗМ - 3" viiakse igasugune operatsioon, olgu ta õige või vale, lõpuni. Kui impulsi "operatsiooni lõpp" ajal on kõik vigade registreerimise trigerid seisus "0", siis arvuti jätkab käskude täitmist normaalses järjekorras, Kui kasvõi üks vigade triggeritest on seisus "1", annab arvuti juhtimise pesale 0026, milles peab olema korrigeerivale programmile suunamise käsk.

Korrigeerivale programmile üleminekul tehakse programmiliselt kättesaadavaks vigade registreerimise triggerite seis pessa 0025 salvestamisega.

Nagu programmilise katkestuse korral, toimub korrigeeriva programmi algusaadressi valimine registri CK sisu muutmata. See võimaldab korrigeerival programmil teha kindlaks vigaselt sooritatud tehte aadressi ja pärast vea analüüsi ning korrigeerimist jätkata tööd.

Üleminekul korrigeerivale programmile kustutatakse automaatselt vigade triggerite sisud ja katkestatakse nende mõju arvutile. Vastasel korral vea tekkimine kor-

rigeerivas programmis võib viia lõpmatusse tsüklisse. Vigade trigerite mõju ignoreeritakse spetsiaalse vigade blokeerimise trigeri abil, mis lülitatakse sisse korrigeeriva programmi tööle hakkamisel ja keelab uued korrigeeriva programmi poole pöördumised. Arvutis on spetsiaalne vigade blokeerimise kustutamise käsk koodiga 0200, mis peab üsuma korrigeeriva programmi lõpus ja taastama kontrollsüsteemi normaalse töö.

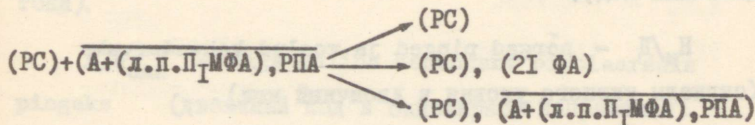
T Ä H I S T U S I

1) (A) - pesa sisu, mille aadress on A.

2) <a> - selle pesa aadress, kus asub arv a.

3) Faktilise aadressi moodustamiseks vajalikud andmed ühendame pealt joonega. Näiteks $\overline{A + (\pi.n.\Pi_I M\Phi A)}, P\Pi A$ tähendab, et faktiline aadress, mille järgi toimub operatiivmälu poole pöördumine, saadakse sel teel, et A-le liidetakse modifikaatori $\Pi_I M\Phi A$ vasaku poolsõna sisu ning mäluvälja number võetakse registrist P\Pi A.

4) Informatsiooni ülekandmist näitame noolega. Näiteks (B) $\rightarrow \overline{(A, P\Pi A)}$ tähendab, et pesa B sisu viiakse pesasse aadressiga A, mille mäluvälja number võetakse registrist P\Pi A. Kui sama käsuga on võimalik operatsiooni tulemust, olenevalt tunnusest, saata mitmesse kohta, siis seda näitame harunevate nooltega. Näiteks:



tähendab, et summaatori PC sisule liidetakse pesa

$\overline{A + (\pi.n.\Pi_I M\Phi A)}, P\Pi A$ sisu ja tulemus salvestatakse

olenevalt tunnusest $\Pi_2\Pi_3$ järgmistesse registritesse ja pesadesse: PC ($\Pi_2\Pi_3 = 00$), PC ja 21ΦA ($\Pi_2\Pi_3 = 01$) või PC ja samasse pesasse, kust võeti üks operand ($\Pi_2\Pi_3 = 10$).

1. Paljukanalilise muundaja teisendused.

H/D - pinge kahendarvuks
(напряжение постоянного тока в двоичный код).

T/D - alalisvool kahendarvuks
(постоянный ток в двоичный код).

H_p/D - kahenivooline releeline pinge kahendarvuks (релейно - изменяющееся двухпозиционное напряжение в двоичный код).

H_{II}/D - vahelduvvoolu amplituud kahendarvuks (амплитуд напряжения и тока переменного тока в двоичный код).

H_M/D - nõrgad pinged ja voolud kahendarvuks (сигналы низкого уровня в двоичный код).

H_{PM}/D - kahenivooline negatiivne releeline pinge kahendarvuks (релейно - изменяющееся двухпозиционное отрицательное напряжение в двоичный код).

$\Pi_{\text{пар}}/\text{Д}$ - paralleelne numbriline informatsioon
kahendarvuks (приём кодов от датчиков параллельной
цифровой информации).

$\text{Ив}/\text{Д}$ - ajaintervall kahendarvuks
(интервал времени в двоичный код).

$\text{N}/\text{Д}$ - impulsside loendamise
(пересчёт импульсов).

$\Pi_{\text{пар.посл.}}/\text{Д}$ - järjestikulis - paralleelne numb-
riline informatsioon kahendarvuks (цифровая параллельно-
последовательная информация в двоичный код).

$\text{В}/\text{Д}$ - jooksev aeg kahend - kümnendkoodiks
(текущее время в двоично - десятичный параллельный код).

$\text{Д}/\text{N}$ - viiekoaline kahendarv pingeks
(пятиразрядный двоичный код в напряжение постоянного
тока).

$\text{Д}/\text{N}_{\text{точ}}$ - kümnekoaline kahendarv pingeks
(десятиразрядный двоичный код в напряжение постоянного
тока).

$\text{Д}/\text{N}_{\text{блп}}$ - algebraline kahendarv bipolaarseks
pingeks (двоичный код в биполярное напряжение).

$\text{Д}/\text{N}_{\text{р}}$ - kahendarv kahenivooliseks releeliseks
pingeks (двоичный код в двухпозиционное релейное
напряжение).

Д/ИВ - kahendarv ajaimpulsiks
(двоичный код в импульс времени).

Д/Буф - kahendarvu ja kanali numbri andmine
puhverseadmesse (передача двоичного кода и адреса
канала в буферное устройство).

2. Lühendid.

- AЗУ - aadressiregister
- АП - mäluvälja aadressi register
- АПО - nullinda mäluvälja aadressi register
- АЦПУ - laitrükkur
- ИС - interpreteeriv süsteem
- КР - kontrollregister
- л.п. - vasak poolsõna
- НОП - vahetu operand
- ОЗУ - operatiivmälu
- п.п. - parem poolsõna
- ППЛ - perfolindi pult
- п.с. - poolsõna
- РБП - katkestuse blokeerimise register
- РИП - informatsioonilise katkestuse register
- РИТ - interpretatsiooniregister
- РКО - tehtekoodi register
- РМ - modifikatsiooni register
- РМЧ - teguri - jagatise register

- PH - mäluvälja register
- PHА - operandide välja aadressi register
- PHK - käskude välja aadressi register
- PHO - nullinda mäluvälja register
- PHП - programmilise katkestuse register
- PHP - katkestusregister
- PC - summaator
- PC_{HC} - interpreteeriva süsteemi summaator
- P3П - katkestuspalvete salvestamise register
- P3V - informatsiooni vahetamise register
- PΦA - fikseeritud aadressi register
- PXM - modifikaatori M register
- PXП - tunnuse register
- CK - käskude loendaja
- CM - modifikatsiooni loendaja
- CT - taktide loendaja
- T - БП2
- T - БП3
- T - БП4
- T - БП5
- T - БПP - katkestuse blokeerimise triger
- TBC - vea mõju blokeerimise triger
- T_{вл.пер.} - ületäitumise mõju triger
- T3П - ületäitumise salvestamise triger
- T - 3C - vea salvestamise triger

Т - СБ1 ÷ Т - СБ13 - veasignaali salvestamise trigerid
 Т - СМ - seisutriger
 УПУ - paljukanaliline muundaja
 УТ - juhttrigerid
 ФА - fikseeritud aadress
 ФЗУ - ferriitmälu

 АЗУ - регистр адреса запоминающих устройств
 АП - регистр адреса
 АПО - регистр адреса памяти
 АЦПУ - альфавитно - цифровое печатающее устройство
 ИС - интерпретирующая система
 КР - контрольный регистр
 л.п. - левое полуслово
 НОП - непосредственный операнд
 ОЗУ - оперативное запоминающее устройство
 п.п. - правое полуслово
 ППЛ - пульт перфоленты
 п.с. - полуслово
 РБП - регистр блокировки прерывания
 РЗП - регистр запоминания прерывания
 РЗУ - регистр запоминающих устройств
 РИТ - регистр интерпретации
 РИП - регистр информационного прерывания
 РКО - регистр кода операции

- РМ - регистр модификации
 - РМЧ - регистр множителя частного
 - РП - регистр поля
 - РПА - регистр поля адреса
 - РПК - регистр поля команд
 - РПО - регистр поля
 - РПП - регистр программного прерывания
 - РПР - регистр прерывания
 - РС - регистр суммы
 - РС_{ис} - регистр суммы интерпретирующей системы
 - РФА - регистр фиксированного адреса
 - РХМ - регистр хранения модификатора
 - РХП - регистр хранения признака
 - СК - счетчик команд
 - СМ - счетчик модификации
 - СТ - счетчик тактов

 - Т - БП2
 - Т - БП3
 - Т - БП4
 - Т - БП5
- } - триггера блокировки полей 2, 3, 4, 5
- Т - БПР - триггер блокировки прерывания
 - ТБС - триггер блокировки сбоя
 - Т_{вл.пер.} - триггер влияния переполнения
 - ТЗП - триггер запоминания переполнения
 - Т - ЗС - триггер запоминания сбоя

FÖÖRDUMINE STANDARDPROGRAMMIDE POOLE

<p>Süsteemid MC - I ja MC - 2</p>	<p>"Б В О Д" "Б Н В О Д" Kõik ülejäänud standard-programmid</p>	<p>a) (k + 0000) = L 17 01 (k + 0001) = A 00 13, kus (L) = B n - 1. Tagasi (k + 0002). kus (L) = B n - 1 (k + 0000) = L 00 41 (k + 0001) = A 30 63, kus (L) = B n - 1</p>	<p>b) (k + 0000) = 0017 40 00 (k + 0001) = B n - 1 (k + 0002) = A 00 13 Tagasi (k + 0003)</p>
<p>Süsteemid ПОН - I, MC - 2M ja MC - 3M</p>	<p>Tagasi (k + 0002) kus (L) = B n - 1 (k + 0000) = A 3M 63 (k + 0001) = L 00 00, kus (L) = B n - 1 M on suvaline, mõnedel SP-del on tal tähendus. Tagasi (k + 0002)</p>		

Tähistused : A - SP algusaadress, B - argumentide massiivi algusaadress,
n - argumentide arv.

PALJUKANALILINE

Jrk. nr.	Teisendus	Sümb. täh.	Teisenduse liigi liigi kahendkood B	Teisenduse liigi ja kanalid lii numbrid kood kahekand süsteemis	Ka- na- li- te arv	Sisendsignaali (otsene teisendus) või väljundsignaal (pöördteisendus)
1	2	3	4	5	6	7
O T S E S E D T E I S E N -						
1.	H/D	B ₀₀	0000	0000-0047	40	0 - +10V (10,23)
				0050-0053	4	-10 - 0V (+0,23)
				0054-0057	4	-5 - 0 - +5V
				0060-0067	8	-5 - 0 - +5mA
2.	T/D	B ₀₁	0001	0070-0077	8	0 - 20mA (20,46)
				0640-0737	64	0 - +5mA (5,11)
				1500-1617	80	0, +10 V
3.	H _p /D	B ₀₃	0011	1520-1627	8	kontakt
				1630-1637	8	-10 V, 0 V

M U U N D A J A " B - 31 "

Sisend- või koor- musta- kistus	Teisen- duse kiirus	Teisen- duse täpsus	Teisenduse iseärasus, mille mää- rab	Inform. paikne- mine pesas	Kanali- te ise- loomus- tus	Mär- ku- sed
8	9	10	11	12	13	14
D U S E D						
7,5 kΩ	500μsek või 300μsek	0,5% või 1,5%	0 - täpne (0,25 - - 0,5 %)	1. - 10.	4 kau- pa omava- hel	1. Pin- ge või
			1 - kiire (1 - 1,5%)	13.	vahe- tata- vad	voo- lu
2 kΩ	- " -	- " -	- " -	- " -	kana- lid. Sama kana-	mi- sele ni- voo-
2 - 3,5 kΩ	50μsek	-	0 - ühe- aegselt kanali valikuga	1. - 12.	lit võib korra- ga ka-	le vas- tab kood

1	2	3	4	5	6	7
4.	H _{II} /Д	B ₀₇	0111	(3400-3477) (3640-3747)	(64) (64)	pinge vool
5.	H _M /Д	B ₀₂	0010	(1000-1017) (1020-1027)	(32)	0 - 50 mV või 0-0,5mA
6.	H _{PM} /Д	B ₀₄	0100	2000 - 2077	64	0, -10 V
7.	II _{пар} /Д	B ₀₅	0101	2400, 2420, 2440, 2460, 2500, 2520, 2540, 2560, 2600, 2620, 2640, 2660, 2760	8 5	12 signaali 0, +10 V 12 signaali 0, -10 V
7a.	II _{пар} /Д	B ₀₅		2700 - 2722	19	12 signaali 0, -10 V

8	9	10	11	12	13	14
			1 - peatusega pärast kanali valikut		sutada ainult üheks teisen- dustest	0. 2. Kanalid 340 - 377 on reserveeritud väliskommutatoritele.
7,5kΩ 2kΩ	200msek	1-2%	0 - pinge 1 - vool	1.-10 13.	B ₀₀ , B ₀₁ , B ₀₃ , B ₀₇ . Kokku 256 kanalit.	
100 kΩ	100msek	1%	- " -	- " -	Väliskommutatorid, 16 kanalit igaühes	
2 - 3,5 kΩ	50μsek	-	Sama, mis B ₀₃ puhul	1.-12.	spetsiaalsed kanalid	
- " -	- " -	-	0 - andurid mitte üle 12 1 - andureid üle 12, grupiline vastuvõtt	1.-13. 1.-13.	spetsiaalsed kanalid	
	150μsek	-		1.-13.	Vastuvõtt 2.	

1	2	3	4	5	6	7
7b.		B ₀₅		2740 - 2745 2746 - 2750 2751	6 3 1	19- kohaline V-kood 15- kohaline V-kood 13- kohaline V-kood
7c.	ИВ/Д	B ₀₅		2724 - 2731 2723 2732 - 2733	6 1 2	täitesagedus 16 kHz 0-0,064 sek täitesagedus 8 kHz 0 - 0,128sek täitesagedus 2 kHz 0 - 0,512sek
7d.	И/Д	B ₀₅		2735 - 2737	3	0-4096 imp.
7e.	Ц пар. посл./Д	B ₀₅		2734	1	10 kahend - kümnend koh- ta 1+4+4+1
8.	В/Д	B ₀₆	0110	3377	1	Aja kahend- kümnendkood
				P Ö Ö R D T E I S E N D U -		
9.	Д/И	B ₀₈	1000	4120, 4125, 4132, 4140, 4145, 4152,	21	0 - +10 V või 0 - -10 V

8	9	10	11	12	13	14
	250 μ sek			3.-13. 5.-13. 6.-13	panee- li puh- ver- sead- mete kaudu	
	0,065sek	0,1%		1.-10. 13.		
	0,13 sek					
	0,55sek					
				1.-13.		
	1sek			1.-10. 13.		
2 k Ω	200 μ sek	1 sek ööpäe- vas	0 - tund, minut, sekund 1 - minut, sekund, millisek.	0.-23.	spet- siaal- ne ka- nal	
S E D						
300 Ω	50 μ sek	\pm 3%		1.-10.	Mälu- seade 240	

1	2	3	4	5	6	7
				4160, 4165, 4172, 4200, 4205, 4212, 4220, 4225, 4232, 4240, 4245, 4252, 4260, 4265, 4272		olenevalt skeem- plaatidest
10.	Д/Н _{ТОЧ}	В ₀₈	1000	4345/4352 4365/4372	2	0 - -10 V
11.	Д/Н _{ОЛШ}	В ₀₉	1001	4700, 4705, 4712, 4720, 4725, 4732, 4740, 4760	8	-10 - 0 - - +10 V
12.	Д/Н _Р	В ₁₀	1010	5000 - 5016 5020 - 5036 5040 - 5056 5060 - 5076 5100 - 5116	15 60	0, +10 V 0, -10 V
13.	Д/ИВ	В ₁₀	1010	5017	1	Ajaintervall
14.	Д/БуФ	В ₁₁	1011	5400 - 5777		Antakse väl- ja aadress (8kk.) ja (6kk.)

8	9	10	11	12	13	14
			0 - kasuta-		kahend-	
			takse nomi-		kohaga	
			naalkomp-	1.-10.	Võib 5	
			lekti	(6.-10.)	kaupa	
			1 - kasuta-	0.-10.	ühen-	Antak-
			takse puh-		dada	se
			verseadet		mälu-	välja
15 k ²	200 μ sek	$\pm 0,5\%$		1.-10.	regist-	kahe
300 μ	50 μ sek	$\pm 3\%$			ri ka-	käsu-
					hend-	ga.
					kohti	
					ühelt	
				1.-12.	teisen-	
- " -	- " -	-			duse	
100 μ	- " -	-		1.-6.	lii-	
				(arv)	gilt	
					teisele.	
2k ²		$\pm 0,1\%$				
2k ²						

Elektrilise kirjutusmasina "ПУБВИ-92" kood.

Nr.	Sümbol	Kood	Nr.	Sümbol	Kood
1.	0	0000000	26.	≠	0011001
2.	1	0000001	27.	ℓ	0011010
3.	2	0000010	28.	'	0011011
4.	3	0000011	29.	≠	0011100
5.	4	0000100	30.	<	0011101
6.	5	0000101	31.	>	0011110
7.	6	0000110	32.	:	0011111
8.	7	0000111	33.	А	0100000
9.	8	0001000	34.	Б	0100001
10.	9	0001001	35.	В	0100010
11.	+	0001010	36.	Г	0100011
12.	-	0001011	37.	Д	0100100
13.	/	0001100	38.	Е	0100101
15.	,	0001101	39.	Ж	0100110
15.	.	0001110	40.	З	0100111
16.	⌊	0001111	41.	И	0101000
17.	10	0010000	42.	Й	0101001
18.	↑	0010001	43.	К	0101010
19.	(0010010	44.	Л	0101011
20.)	0010011	45.	М	0101100
21.	x	0010100	46.	Н	0101101
22.	=	0010101	47.	О	0101110
23.	;	0010110	48.	П	0101111
24.	[0010111	49.	Р	0110000
25.		0011000	50.	С	0110001

Rahvusvaheline telegraafikood Nr. 2.

Register			Kood					Paar- suse kont- roll
Ladina	Vene	Arvud	1.	2.	3.	4.	5.	
A	А	-	1	1	0	0	0	1
B	Б	?	1	0	0	1	1	0
C	С	:	0	1	1	1	0	0
D	Д	кто там	1	0	0	1	0	1
E	Е	3	1	0	0	0	0	0
F	Ф	Э	1	0	1	1	0	0
G	Г	Ш	0	1	0	1	1	0
H	Х	Щ	0	0	1	0	1	1
I	И	8	0	1	1	0	0	1
J	Й	Д(зв)	1	1	0	1	0	0
K	К	(1	1	1	1	0	1
L	Л)	0	1	0	0	1	1
M	М	.	0	0	1	1	1	0
N	Н	,	0	0	1	1	0	1
O	О	9	0	0	0	1	1	1
P	П	0	0	1	1	0	1	0
Q	Я	1	1	1	1	0	1	1
R	Р	4	0	1	0	1	0	1

S	C	'	1	0	1	0	0	1
T	T	5	0	0	0	0	1	0
U	Y	7	1	1	1	0	0	0
V	X	=	0	1	1	1	1	1
W	B	2	1	1	0	0	1	0
X	Ь	/	1	0	1	1	1	1
Y	Ы	6	1	0	1	0	1	0
Z	З	+	1	0	0	0	1	1
<			0	0	0	1	0	0
≡			0	1	0	0	0	0
Ladina register			1	1	1	1	1	0
Arvu register			1	1	0	1	1	1
Tühik			0	0	1	0	0	0
Vene register			0	0	0	0	0	1

S I S U K O R D

Sissejuhatus	3
I peatükk. Juhtimisarvuti "ВНИИЭМ - 3" kirjeldus	
§ 1. Üldisi andmeid arvutist "ВНИИЭМ - 3"	4
§ 2. Unifitseeritud informatsiooni vahetamise kanalid	9
1. Ühekordse informatsiooni vahetamise kanal	9
2. Grupilise informatsiooni vahetamise kanal	10
§ 3. Prioriteetse katkestuse süsteem	11
1. Programmiline katkestus	14
2. Informatsiooniline katkestus	19
§ 4. Arvuti "ВНИИЭМ - 3" tähtsamad registrid	21
§ 5. Arvuti juhtimine puldilt	25
1. Signalisatsioonipaneel	25
2. Juhtimispaneel	29
3. Käivitamine ja peatamine	32
4. Töörežiimid	32
5. Perfolindil oleva informatsiooni salvestamine	34

§ 6. Adresseerimise süsteem operatiivmälu poole pöördumisel	35
§ 7. Käskude ja arvude kujutamine	38
1. Fikseeritud komaga arvude kujutamine	39
2. Kümnendsüsteemi arvude ja tähelis-numbri- lise informatsiooni kujutamine	41
§ 8. Käskude modifitseerimine	43
II peatükk. Käskude süsteemi kirjeldus	
§ 1. Aritmeetilised ja loogilised operatsioonid.	45
§ 2. Interpreteeritavad käsud	47
§ 3. Vahetu adresseerimine	49
1. Aritmeetilised operatsioonid	49
2. Vahetu adresseerimisega loendamise ja suunamise käsud	50
§ 4. Juhtimiskäsud	51
§ 5. Informatsiooni vahetamine fikseeritud aadres- siga pesade ja operatiivmälu pesade vahel	54
§ 6. Magnetlintmälu poole pöördumise käsud	54
1. Magnetlindilt lugemine ja lindile kirjutamine	54
2. Lindi kerimise käsud	56
§ 7. Grupilise informatsiooni kanali poole pöör- dumise käsud	57
§ 8. Ühekordse informatsiooni kanali poole pöör- dumise käsud	59

§ 9. Erikäsud	61
1. Katkestussüsteemiga seotud käsud	61
2. Nihutemiskäsud	61
3. Erilisi tehteid käskude ja arvudega	63
III peatükk. Programmeerimise süsteemid	
§ 1. Interpreteerivad süsteemid	68
§ 2. Interpreteerivate süsteemide standard-	
programmid	73
1. Standardprogramm kümnendarvude sises-	
tamiseks "ВВОД"	74
2. Liikuva komaga kümnendarvude massiivi	
trükkimine "ВЫВОД"	75
§ 3. ПОН - süsteem	76
§ 4. Standardprogrammide ülesehitus ПОН - süsteemis	
	78
§ 5. Standardprogrammide kasutamine ПОН - süs-	
teemis	80
§ 6. Standardprogrammid ПОН - süsteemis	81
1. Funktsioonide $y = e^x$, $y = \ln x$ ja	
$y = \sqrt{x}$ arvutamine	82
2. Funktsioonide $y = 1/2 \sin x$, $y = 1/2 \arcsin x$	
ja $y = \arctg x$ arvutamine	82
3. Sõnade trükkimine kaheksandsüsteemis ilma	
adressita, 7 sõna ritta	83
4. Programmi trükkimine lehekülgede kaupa	83
5. Täisarvude teisendamine kahendsüsteemist	

kümnendsüsteemi ja tulemuse trükkimine .	84
6. Fikseeritud komaga arvude teisendamine kahendsüsteemist kümnendsüsteemi ja tulemuse trükkimine	84
7. Kaheksandarvude massiivi väljastamine. .	84
8. Kümnendsüsteemi täis- ja ühest väksemate murdarvude arvutisse viimine ja kahendsüs- teemi teisendamine	86
§ 7. Süsteemid MC - 2M ja MC - 3M"	88

IV peatükk Käskude süsteem

§ 1. Põhiliste aritmeetiliste ja loogiliste ope- ratsioonide käsud	89
§ 2. Vahetu adresseerimisega aritmeetilised ja loogilised operatsioonid	94
§ 3. Vahetu adresseerimisega loendamise ja suunami- se käsud	95
§ 4. Juhtimiskäsud	97
§ 5. Informatsiooni vahetamine fikseeritud aadressiga pesade ja operatiivmälu pesade vahel . . .	99
§ 6. Mitteamitmeetilised käsud	100
§ 7. Informatsiooni vahetamise käsud magnetlint- mälu ja paljukanalilise muundajaga . . .	107

V peatükk. Programmeerimine

§ 1. Elementaarseid programmeerimisvõtteid . .	116
§ 2. Tsüklid	118
§ 3. Iteratsioonitsüklid	126

§ 4. Standardprogrammide struktuur	127
§ 5. Kontrollsüsteemidest	131
Lisa 1. Tähistusi	137
1. Paljukanalilise muundaja teisendused .	138
2. Lühendid	140
Lisa 2. Pöördumine standardprogrammide poole . .	145
Lisa 3. Paljukanaliline muundaja "B - 31"	146
Lisa 4. Elektrilise kirjutusmasina "ПУВВИ - 92" kood	154
Lisa 5. Rahvusvaheline telegraafikood Nr.2	155
Lisa 6. "ВНИИЭМ - 3" käskude süsteem	157

Управляющая вычислительная машина
"ВНИИЭМ-3"

На эстонском языке

Составили: М.Сарв, Х.Нийлиск, М.Маасик

Редакционно-издательский совет Академии наук
Эстонской ССР. Таллин, ул.Сакала, 3.

Toimetaja U.Jürgenson.

Korrektor M.Lutter

Trükkida antud 9.09.1968. Trükipoognaid 10,25.

Arvestuspoognaid 5,39. Tiraaz 200. MB-06065.

Tellimise nr.209. TA rotaprint, Tallinn,

Sakala 3.

Hind 54 kop.

TÜ RAAMATUKOGU



10300015133400

A-38970

15 21
1915 ¹⁷⁵ 17 116 911 70 100
4213 963 1321
3