Ü. KAASIK · H. SALUM · M. SINISOO

ELEKTRON-ARVUTUS-MASINAD

Raamatus kirjeldatakse diskreetse toimega elektron-arvutusmasinate töötamise põhimõtteid ning nende konstruktsiooni matemaatilisi ja füüsikalisi aluseid. Niisuguste masinate ekspluateerimise küsimusi illustreeritakse näidetega nõukogude arvutustehnika vallast. On toodud lühike ülevaade elektron-arvutusmasinate kasutamise võimalustest ning tähtsamatest kodu- ja välismaa arvutusmasinatest.

Raamat on määratud keskharidusega lugejatele, kes tunnevad huvi tänapäeva arvutustehnika küsimuste vastu. Toodud elektriliste skeemide funktsioneerimise täielikuks mõistmiseks on aga vajalik raadiotehnika aluste mõningane tundmine.

Ü. KAASIK, H. SALUM, M. SINISOO

ELEKTRON-ARVUTUSMASINAD

EESTI RIIKLIK KIRJASTUS
TALLINN 1960

Retsenseerinud tehn. tead. kand. H. Sillamaa

I peatükk

SISSEJUHATUS

1. ARVUTUSTEHNIKA ARENGUST

Väga paljud tänapäeva teaduslik-tehnilised probleemid taanduvad sellistele matemaatilistele ülesannetele, mille tegelik lahendamine on seotud üpris mahukate arvutustega. Arvutuste maht on sageli isegi niivõrd suur, et vaatamata ülesande põhimõttelisele lahendatavusele peab teda tavaliste arvutusvahendite (aritmomeetrite jms.) kasutamise korral siiski lugema praktiliselt lahendamatuks. Nii näiteks tuleb kaasaja tuumafüüsika, aerodünaamika, astronoomia, energeetika, kvantmehaanika jms. ülesannete lahendamiseks mõnikord teostada isegi sadu miljoneid arvutusoperatsioone. Arvutusaja lühendamiseks ollakse niisuguseid ülesandeid praktikas tihti sunnitud lahendama tunduvalt lihtsustatud kujul, jättes vaadeldavat nähtust mõjutavatest faktoritest mõned arvestamata. Loomulikult muudab see saadud tulemused vähem täpseteks (vahel aga koguni ebaõigeteks).

Tänapäeva teaduse ja tehnika arengut iseloomustab üha suurema täpsuse vajadus ülesannete lahendamisel. See muudabki mahukate arvutustööde kiire läbiviimise probleemi eriti aktuaalseks. Pealegi tuleb arvestada, et mitmete ülesannete puhul võib pikkadele arvutustele kulunud aeg muuta saadava tulemuse hoopis väärtusetuks. Näiteks kaugjuhitava aerodünaamilise objekti ruumilist liikumist kirjeldav võrrandsüsteem on niivõrd keeruline, et selle ligikaudseks lahendamiseks (vajaliku täpsusega) kuluks lihtsamaid arvutusmasinaid kasutaval arvutusbürool hulk aastaid, objekti enese liikumise aeg on aga mõõdetav minutitega. Umbes samasugune olukord leiab aset ka ilmade ennustamisel küllaldase täpsusega ja paljudes teistes probleemides.

Öeldut arvestades on arusaadav, et kiirelt töötavate arvutusmasinate leiutamine muutis radikaalselt matemaatika võimalusi praktikas esilekerkivate ülesannete lahendamisel. Täie õigusega loetakse seda leiutust inimkonna suurimate saavutuste hulka.

Kaasaegse arvutustehnika täiustamine on toimunud kahes põhilises suunas. Esimene suund on seotud nn. pideva toimega ning teine suund nn. diskreetse toimega (ehk numbriliste) arvutusmasinate loomisega.

Pideva toimega matemaatiline masin kujutab endast teatud füüsikalist süsteemi (mehhanismi, elektrilist ahelat jms.), milles pidevalt muutuvate suuruste (pikkuste, nurkade, elektriliste pingete) vahel on loodud samasugune sõltuvus nagu lahendada tuleva ülesande andmete ja tulemuste vahel. Selliseid matemaatilisi masinaid nimetatakse sageli ka analoogia-masinateks. Lihtsaimaks seda tüüpi arvutusvahendiks on logaritmiline arvutuslükati, millel arve kujutatakse pikkustena. Pideva toimega elektronmasinates toimub vajalike tehete sooritamine spetsiaalsete elektriliste skeemide abil, kusjuures iga skeem on määratud ainult mõne tehte sooritamiseks. Tavaliselt sisaldab masin skeeme liitmiseks, lahutamiseks, korrutamiseks, jagamiseks, diferentseerimiseks, integreerimiseks, logaritmimiseks, eksponent- ja trigonomeetriliste funktsioonide väärtuste leidmiseks. Enne ülesande lahendamist ühendatakse need skeemid sobivasse järjekorda.

Pideva toimega arvutusmasinad võimaldavad tavaliselt küllaltki kiiresti lahendada matemaatilisi ülesandeid, kuid nende abil saadud tulemuste täpsus on piiratud (tavaliselt ei võimalda need masinad suuremat täpsust kui 0,01%) ning sõltub oluliselt masinate üksikute osade valmistamise täpsusest. Pideva toimega masinad on ka alati enam või vähem spetsialiseeritud teatavat tüüpi ülesannete lahendamiseks. Näiteks on selliseid masinaid ehitatud lineaarsete algebraliste võrrandsüsteemide lahendamiseks, harilike diferentsiaalvõrrandite süsteemide lahendamiseks, seniitkahurite tule juhtimiseks ning paljude teiste konkreetsete probleemide jaoks.

Pideva toimega arvutusmasinad omavad tänapäeval küllaltki suurt rakenduslikku tähtsust, kuid käesolevas raamatus me nendel pikemalt ei peatu.[1]

---

[1] Üksikasjalisemaks tutvumiseks pideva toimega matemaatiliste masinate ehituse ning töötamise põhimõtetega võib kasutada näiteks raamatut: Н. Е. Кобринский. Математические машины непрерывного действия. Москва, 1954.

D i s k r e e t s e t o i m e g a a r v u t u s m a s i n a t e s kujutatakse matemaatilisi suurusi arvudena, kusjuures iga numbri kujutamiseks kasutatakse seadist (elementi), mis võib omada mitut üksteisest rangelt erinevat seisundit, milledest igaühele vastab mingi kindel number. Sellist tüüpi masina lihtsamaks näiteks on tavaline aritmomeeter, milles numbreid kujutavateks elementideks on kümne hambaga hammasrattad (iga hammas vastab ühele kindlale numbrile).

Diskreetse toimega masinate üheks oluliseks eeliseks pideva toimega masinatega võrreldes on nendes teostatavate arvutuste suurem võimalik täpsus. Sealjuures pole suurema täpsuse saavutamiseks tarvis suurendada üksikosade valmistamise täpsust, vaid tuleb lihtsalt suurendada elementide hulka, mis kujutavad kasutatavate arvude numbrikohti. Seega saab suure täpsusega töötavaid arvutusmasinaid ehitada suhteliselt mitte eriti täpselt töödeldavatest detailidest. Tähtis on vaid, et elementides ei kaoks erinevus seisundite vahel, mis kujutavad erinevaid numbreid.

Teiseks diskreetse toimega masinate eeliseks on nende universaalsus. Kuigi sellised masinad võimaldavad vahetult teostada ainult aritmeetilisi tehteid arvudega, saab nende abil lahendada siiski väga mitmesuguseid matemaatilisi ja loogilisi ülesandeid. Nimelt võimaldavad väljatöötatud numbrilised meetodid peaaegu iga ülesande taandada nelja aritmeetilise tehte järjestikusele sooritamisele.

Diskreetse toimega arvutusmasinate oluliseks puuduseks oli pika aja vältel nende suhteliselt aeglane töötamine. Mõningal määral võimaldas masina töötamiskiirust suurendada küll näiteks aritmomeetris vända asendamine elektrimootoriga, kuid arvutuskiiruse otsustavat suurenemist see siiski kaasa ei toonud. Põhjuseks oli muuhulgas asjaolu, et arvutamisel aritmomeetriga kulub suhteliselt suur osa ajast arvude masinasse viimiseks ning tulemuste (sealhulgas ka ainult järgnevateks arvutusteks vajalike vahepealsete tulemuste) väljakirjutamiseks.

Otsustavamaks edusammuks diskreetse toimega arvutusmasinate täiustamisel kujunes nn. a n a l ü ü t i l i s t e a r v u t u s m a s i n a t e loomine käesoleva sajandi alguses. Selliseid masinaid ehitatakse peamiselt statistika, raamatupidamise ja pangaasjanduse vajaduste rahuldamiseks ning neis on arvude sisseviimine masinasse juba mehhaniseeritud ja arvutusprotsessist eraldatud. Nimelt kantakse arvud eelnevalt aukude süsteemidena standardsetele kartongkaartidele — nn. perfokaartidele[1] ja paigutatakse niisuguste perfokaartide pakk masinasse. Masin «loeb» perfokaartidel olevaid arve eriliste kontaktidega kompamise teel ja teostab nendega vajalikke tehteid. Sealjuures saab ühes masinas teostada ainult üht aritmeetilist tehet.

Suurimat tähtsust analüütiliste arvutusmasinate hulgas omavad liitmismasinad — tabulaatorid. Nendes on teataval määral ette nähtud ka arvutuste automaatne juhtimine. See võib olla realiseeritud näiteks nii, et koos arvudega kantakse (eraldi aukude süsteemi näol) perfokaartidele ka nende arvude mingid tunnused. Tabulaator liidab siis vaid need järjestikku paiknevad arvud, milledel on ühesugused tunnused. Kui tunnus muutub, siis annab tabulaator eelmise summa trükituna välja ja asub liitma uusi arve.

Peale tabulaatorite on ehitatud ka korrutamist ning mõningaid loogilisi tehteid (näiteks kaartide sorteerimist) teostavaid masinaid. Nende masinate peamisteks töötavateks elementideks on enamasti elektromagnetilised releed.

Selleks et arvutustulemusi oleks võimalik kasutada edasistes arvutustes, on analüütiliste arvutusmasinate puhul tulemuste väljaandmine võimalik mitte ainult trükitud kujul, vaid ka aukude süsteemidena uutel perfokaartidel. Need perfokaardid peab viima teist tehet teostavasse (või ka samasse) masinasse juba inimene.

Kuigi analüütiliste arvutusmasinate kompleks saab seega põhimõtteliselt sooritada kõiki neid aritmeetilisi ja loogilisi tehteid,[2] mida on vaja enamiku matemaatiliste ülesannete lahendamisel, ei ole seda tüüpi masinate rakendamine matemaatiliste ülesannete lahendamiseks siiski efektiivne. Selle peamiseks põhjuseks on asjaolu, et matemaatilise ülesande lahendamisel tuleb tavaliselt sooritada järjestikku erinevaid tehteid, mida aga teostavad erinevad masinad. Arvutusprotsessi peab seetõttu jaotama paljudeks väga lühikesteks osadeks.

---

[1] Sõnast «perforeerima» = auke lööma, mulgustama.

[2] Loogiliseks tehteks on näiteks arvude võrdlemine suurima (või vähima) valimiseks neist. Täpsemalt iseloomustame loogilisi tehteid hiljem (vt. § 5 ja § 10).

Teaduse ja tehnika poolt püstitatavatele matemaatilistele ülesannetele on enamasti iseloomulik:

1) vajadus teostada väga suur arv mitmesuguseid aritmeetilisi ja loogilisi tehteid;

2) suure arvu vahepealsete tulemuste olemasolu, mis on vajalikud ainult edaspidistes arvutustes;

3) vajadus sageli muuta arvutuskäiku sõltuvalt eelmistes arvutustes saadud tulemustest.

Selleks et matemaatilise ülesande lahendamine toimuks täielikult masina abil, peab masin järelikult suutma teostada kõiki vajalikke aritmeetilisi ja loogilisi tehteid, omama küllaldase mahtuvusega seadme lähteandmete ning vahepealsete tulemuste säilitamiseks (sellist seadet nimetatakse tavaliselt m ä l u k s) ning suutma automaatselt valida edaspidiste arvutuste suuna sõltuvalt vahepealsetest tulemustest. Neid nõudeid rahuldavaid arvutusmasinaid hakati ehitama käesoleva sajandi neljakümnendatel aastatel.

Esimesed automaatsed arvutusmasinad[1] koostati analüütiliste arvutusmasinate standardsetest blokkidest ning varustati keskse juhtimisseadmega, mis automaatselt lülitas arvutustesse ühe või teise bloki. Konstruktsioonilt olid need masinad äärmiselt komplitseeritud (ühe masina koosseisu kuulus enamasti üle 10 000 relee) ja nende töötamiskindlus väike, kuid arvutustehnika arengus kujutasid nad siiski olulist sammu edasi.

Arvutusi teostas niisugune masin varem koostatud p r o g r a m m i järgi. Programmi saamiseks kodeeriti kõik sooritada tulevad tehted arvudena ja kanti aukude süsteemi näol näiteks erilisele lindile (n n. p e r f o l i n d i l e). Lint liikus masinas kindlas suunas ning masin teostas järjekorras seal ettenähtud tehteid. Nendes masinates polnud veel võimalik muuta arvutuste käiku sõltuvalt vahepealsetest tulemustest. Ka oli programmi koostamine seotud väga suure töökuluga. Seetõttu osutus selliste masinate kasutamine otstarbekaks vaid siis, kui tuli lahendada suurel arvul ühetüübilisi ülesandeid.

Esimeste automaatsete arvutusmasinate töötamiskiirus oli küllaltki väike: kahe arvu liitmisele või lahutamisele

---

[1] Esimeseks seda tüüpi masinaks oli nähtavasti «Automatic Sequence Controlled Calculator» (lühidalt ASCC ehk Mark I), mille konstrueerimist alustati 1937. a. Harvardi ülikoolis Ameerika Ühendriikides. Masin valmis lõplikult 1944. aastal.

kulus 0,2—0,5 sekundit, korrutamisele 0,7—6 sekundit ja jagamisele 1,5—15 sekundit. Ka oli nende masinate mälu suhteliselt väikese mahuga: seal sai säilitada umbes 100 kümnekohalist arvu.

Automaatsete arvutusmasinate edasine arendamine toimus töötamiskiiruse ja mälu mahu suurendamise ning juhtimissüsteemi täiustamise suunas.

Olulisi edusamme sellel teel saavutati, kui elektromagnetilised releed ja muud mehaanilised elemendid asendati elektronlampide süsteemidega. Vastavaid masinaid hakatigi nimetama elektron-arvutusmasinateks.[1] Liikuvate osade (hammasrataste, releede jt.) asendamine praktiliselt inertsivabade elektronlampide süsteemidega võimaldas tuhandeid kordi suurendada arvutamiskiirust.

Koos elektronlampide kasutuselevõtmisega tehti otsustavaid edusamme ka mäluseadmete ja juhtimissüsteemi täiustamisel. Perfolindi asendamine elektronseadistega võimaldas sooritada tehteid mitte ainult kindlas järjekorras, vaid ka muuta ülesande lahendamise programmi (seda ei olnud võimalik teha perfolindile kantud aukudega). Programmi niisuguste muutmiste võimalus arvutuste käigus lihtsustab tunduvalt programmi koostamist ja laiendab seega masina rakendatavuse ulatust, tema universaalsust. Kõik need täiustused on võimaldanud elektron-arvutusmasinate töökiiruse tõsta tänapäevaks kümnetele tuhandetele tehetele sekundis ja tohutult laiendada nende rakenduspiirkonda.

Tänu oma suurele töötamiskiirusele ning universaalsusele leiavadki tänapäeva elektron-arvutusmasinad ja üldse programmjuhtimise printsiip üha ulatuslikumat rakendamist, sealhulgas ka väljaspool puhtmatemaatiliste ülesannete lahendamise piire. Eeskätt tuleb siin nimetada masinaid, mis juhivad üksikute tööpinkide või isegi terve tehase tööd. Näiteks on juba ehitatud väiksemaid tehaseid, kus kogu tootmisprotsess toimub täiesti automaatselt elektron-arvutusmasinate juhtimisel. Sellise tehase teenindamiseks on peale toormaterjali juurdevedajate tarvis ainult paar inimest (vahetusinsener ja koristaja).

---

[1] Esimest seda tüüpi masinat «Elektronic Numerical Integrator and Calculator» (lühidalt ENIAC) hakati Ameerika Ühendriikides konstrueerima 1943. aastal.

Olulist inimtöö kokkuhoidu on võimaldanud suhteliselt lihtsa ehitusega elektron-arvutusmasinate kasutamine raamatupidamises ja laoasjanduses; majanduslikku efekti tõotavad esimesed katsed kasutada elektron-arvutusmasinaid tänavaliikluse reguleerijatena suurlinnades, kõige ökonoomsemate raudtee sõiduplaanide koostajatena jne. Veidi lähemalt kirjeldame elektron-arvutusmasinate n. ö. mittearitmeetilisi rakendusi käesoleva raamatu IV peatükis (§ 12).

Praktika vajadusi silmas pidades ongi kõrvuti universaalsete elektron-arvutusmasinatega, mis võimaldavad arvutusi teostada mistahes programmi järgi, viimasel ajal üha enam tähelepanu pööratud ka spetsialiseeritud elektron-arvutusmasinate väljatöötamisele, milles arvutusprogramm on kas täielikult või osaliselt masinasse viidud juba ehitamisel. Sellised masinad on ette nähtud ainult üht kindlat tüüpi ülesannete lahendamiseks (näiteks etteantud kujuga võrrandite lahendamiseks, ühe tööpingi automaatseks juhtimiseks jne.). Nad on lihtsama ehitusega ja odavamad, kusjuures töötamine nendega on tunduvalt lihtsam kui universaalsete elektron-arvutusmasinatega. Võimaldades suure töötamiskiiruse juures saavutada suurt täpsust, võistlevad nad paljudel aladel edukalt pideva toimega arvutusmasinatega.

Elektron-arvutusmasinate loomine Nõukogude Liidus algas põhiliselt alles pärast Suure Isamaasõja lõppemist, kuid tänapäeval ehitatakse meil juba suurel arvul nii spetsialiseeritud kui ka mitmesuguseid universaalseid elektron-arvutusmasinaid. Käesolevas raamatus leiavad käsitlemist peamiselt vaid viimast tüüpi masinad, kusjuures enamik konkreetseid näiteid tuuakse just vastavate nõukogude elektron-arvutusmasinate «Ural», M-3, «Strela», M-2 ja БЭСМ baasil.

Lähema ülevaate nii nõukogude kui ka välismaa mõnedest elektron-arvutusmasinatest võib leida raamatu lõppu paigutatud lisast.

2. ELEKTRON-ARVUTUSMASINA TÖÖTAMISE PÕHIMÕTTEST

Tänapäeva universaalsed elektron-arvutusmasinad koosnevad enamasti järgmisest viiest põhilisest seadmest:

1) aritmeetiline seade (lühidalt AS),

2) mälu, mis jaguneb nn. sisemiseks ja väliseks mäluks (SM ja VM),

3) juhtimisseade (JS),

4) sisendseade lähteandmete masinasse sisseviimiseks (SS) ja

5) väljundseade arvutustulemuste masinast väljatoomiseks (VS).

Selgitame üldjoontes nende seadmete otstarvet ja nende omavahelist koostööd.

A r i t m e e t i l i n e s e a d e teostab arvudega aritmeetilisi ja loogilisi tehteid. Vajalikud arvud võtab ta tavaliselt sisemisest mälust ning saadab arvutustulemused sinna tagasi. Üksikasjalisemalt selgitame selle seadme töötamist hiljem (vt. § 9—10).

M ä l u ülesandeks on algandmete ja arvutustulemuste salvestamine. Mälu peab rahuldama kaht põhilist nõuet, mida alles kõige viimasel ajal on õnnestunud ühes seadmes korraga realiseerida: arvude vastuvõtmise ning väljaandmise suur kiirus ja suur mahutavus. Seepärast on senistes elektron-arvutusmasinates ehitatud tavaliselt kaks mälu. Neist s i s e m i n e m ä l u ehk operatiivmälu mahutab enamasti 1024 või 2048 arvu ja töötab kiiresti. Sisemise mälu seadist, mis on ette nähtud ühe arvu säilitamiseks, nimetatakse p e s a k s. Kõik pesad on nummerdatud (0-st 1023-ni või 2047-ni), neid numbreid nimetatakse a a d r e s s i d e k s.

V ä l i n e m ä l u omab praktiliselt piiramatu mahu, kuid töötab suhteliselt aeglasemalt. Ta kujutab endast tegelikult sisemise mälu reservi, kuhu viiakse hoiule arvud, mida niipea tarvis ei lähe, kuid mida ei tohi ka «unustada». Arvude ülekandmine sisemisest mälust välimisse ja vastupidi toimub enamasti suuremate rühmade kaupa. Mäluseadme füüsikalist ehitust kirjeldame üksikasjalisemalt käesoleva raamatu III peatükis (§ 7).

J u h t i m i s s e a d e on vajalik masina töö automaatseks juhtimiseks ja ta tagab tehete sooritamise ettenähtud järjekorras. Tarbe korral (näiteks masina töö kontrollimiseks) võib kas osaliselt või täielikult üle minna ka käsitsi juhtimisele (lähemalt vt. § 9).

Lähteandmete universaalsesse arvutusmasinasse viimiseks kantakse nad enamasti (samuti nagu analüütiliste arvutusmasinate puhulgi) perfokaartidele või perfolindile. S i s e n d s e a d m e ülesandeks ongi näiteks perfolindil aukude süsteemidena kujutatud arvude teisendamine elektrilisteks impulssideks ja nende hoiulesaatmine masina sisemisse mälusse. Iga arv saadetakse sealjuures teatud kindlasse pesasse, mille aadress tuleb muidugi sisendseadmesse või juhtimisseadmesse ette anda.

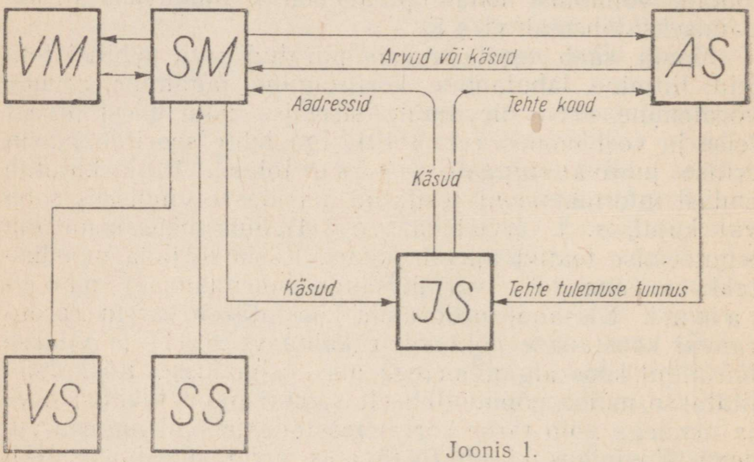
V ä l j u n d s e a d m e ülesandeks on arvutustulemuste teisendamine loetavale kujule (nende kandmine perfokaartidele või trükkimine). Sellised sisend- ja väljundseadmed on muide masina kõige aeglasemalt töötavad osad (alles viimastel aastatel on õnnestunud välja töötada võimalusi nende operatsioonide tunduvaks kiirendamiseks; lähemalt vt. § 8).

Masin saab teostada vaid piiratud arvu põhilisi tehteid: liitmine, lahutamine, korrutamine, jagamine, arvude võrdlemine, arvu üleviimine sisemise mälu ühest pesast teise ja veel mõned (vt. § 10). Iga tehte sooritab masin erilise juhtiva signaali — k ä s u toimel. Käsk kujutab endast informatsiooni (esitatud masinasse viimiseks sobival kujul, s. t. arvudena, vt. § 4), mis määrab masina tegutsemise teatud ajavahemikus. Käskude jada moodustabki masina töö (või ülesande lahendamise) p r o g r a m m i. Ülesande programm koostatakse varem (programmi koostamise põhimõtete kohta vt. § 11) ja viiakse töö algul koos algandmetega masina mälusse. Käske säilitatakse mälus põhimõtteliselt samuti nagu tavalisi arve ja nendega võib tarbe korral teostada tehteid nagu arvudega. Enamikus kiirelt töötavates arvutusmasinates täidetakse käske selles järjekorras, milles nad on mälusse paigutatud (kui vahepeal ei esine käsku, mis käsib seda järjekorda muuta).

Masina automaatse töö korral toimub käsu täitmine kahes järgus: algul antakse käsk mälust juhtimisseadmesse ja siis alles täidetakse. Täitmisele asudes mõjutab käsk kõigepealt aritmeetilist seadet, kohandades ta vajaliku tehte sooritamiseks. Selleks peab käsk ühe osana sisaldama arvu, mis iseloomustab selle käsu toimel teostatavat tehet (seda arvu nimetatakse t e h t e k o o d i k s). Samal ajal teatatakse mälusse, millistest pesadest tuleb võtta andmed ja milline pesa tuleb ette valmistada vastuse salvestamiseks (käsu vastavaid osi nimetatakse k ä s u a a d r e s s i d e k s). Mälust valitud arvud saadetakse seejärel aritmeetilisse seadmesse, kus nendega teostatakse nõutud tehe ja viiakse tulemus hoiule mälu vastavasse pesasse.

Kui tulemus on salvestatud, siis annab sisemine mälu välja järgmise käsu selle saatmiseks juhtimisseadmesse. Masina seadmete omavahelise koostöö küsimusi käsitleme lähemalt IV peatükis (§ 9). Arvutusmasina põhiliste osade vaheline seos on skemaatiliselt kujutatud joonisel 1.

Et töö käigus tuleb edaspidiste arvutuste suund sageli valida sõltuvalt eelmise tehte tulemusest (näiteks ruut-



Joonis 1.

võrrandi lahendite arvutamise käik sõltub diskriminandi märgist), siis on paljudes masinates ette nähtud, et aritmeetiline seade annab pärast tehte sooritamist juhtimisseadmesse erilise signaali, mis iseloomustab tehte tulemust (enamasti selle märki või suurusjärku). Loomulikult peab siis programmis olema ära määratud, millise signaali puhul kumma variandi järgi arvutusi tuleb jätkata. Sellise harunemise realiseerimiseks kasutatakse masinais nn. suunamiskäske (vt. § 10), mis sõltuvalt saadud signaalist suunavad edaspidiste arvutuste käiku. Suunamiskäsu aadressideks on tavaliselt nende pesade aadressid, kus asuvad järgmised käsud, millede vahel signaalist sõltuvalt tuleb valida.

Paljudes universaalsetes elektron-arvutusmasinates on peale ülalloetletud põhiseadmete veel mõned täiendavad seadmed, mis kergendavad ülesannete programmide koostamist ja masina tööd arvutamisel. Nende hulka kuulub näiteks eriline väikese mahuga mälu (mahutab tavaliselt 200—400 arvu) sageli esinevate konstantide alaliseks salvestamiseks. Selles nn. k o n s t a n t i d e m ä l u s säilitatakse arve nagu 0, 1, 100, π, $\frac{1}{\pi}$, e jne.

Mõnedes masinates on ehitatud veel erilised seadmed rea sageli vajaminevate programmide salvestamiseks. Need on enamasti programmid elementaarfunktsioonide ($\sin x$, $\cos x$, $\ln x$, $\sqrt{x}$ jne.) arvutamiseks ning kannavad s t a n d a r d s e t e a l a m p r o g r a m m i d e nime (vt. § 10—11).

* * *

Piirdudes esialgu selle lühikese ülevaatega elektronarvutusmasinatest, asume järgnevalt üksikasjalisemalt selgitama nende masinate ehitust ja töötamist. Käsitluse jaotame sealjuures kolme enam-vähem iseseisvasse ossa.

Teises peatükis anname vajalikud matemaatilist laadi mõisted, kusjuures erilisi eelteadmisi selle peatüki lugemiseks pole eeldatud. Täiesti küllaldaseks osutub keskkooli programmi mõnede osade tundmine.

Kolmas peatükk on pühendatud arvutusmasinate üksikute seadmete füüsikalise ehituse põhimõtete kirjeldamisele. Selle peatüki puhul on eeldatud, et lugeja on teatud määral tuttav raadiotehnika mõningate küsimustega. Vastavate eelteadmiste puudumise korral võib aga üksikute seadmete töökirjeldused ka vahele jätta, mis muide oluliselt ei takista järgmise peatüki mõistmist.

Neljandas peatükis selgitatakse elektron-arvutusmasina kui terviku töötamist ja nende masinate kasutamist matemaatiliste ülesannete lahendamisel. Eraldi paragrahv on sealjuures pühendatud lühikese ülevaate andmisele arvutusmasinate rakendamisvõimalustest üldse. Erilisi eelteadmisi (peale eelmises kahes peatükis toodute) selle peatüki lugemiseks tarvis ei ole.

Käesolevas sissejuhatuses toodud mõisted defineeritakse nende lähema käsitlemise puhul enamasti uuesti.

Elektron-arvutustehnika spetsiifikast tingituna võib raamatus esitatav materjal oma uudsuse ning mittekonventsionaalse käsitluse tõttu tunduda väsitavana, kuid ta annab järjekindlale lugejale võimaluse heita pilku kiiresti areneva tehnikaala küsimuste valdkonda.

MATEMAATILISED ALUSED

3. MASINARVUTUSES KASUTATAVAD ARVUSÜSTEEMID

Positsioonilised arvusüsteemid. Üldiselt tarvitusel olevas arvusüsteemis (s. t. arvude nimetamise ja üleskirjutamise süsteemis) kirjutatakse arvud numbrite jadadena, kusjuures kasutatakse kümmet erinevat numbrimärki 0, 1, 2, 3, 4, 5, 6, 7, 8 ja 9. Need numbrid tähistavad ühtlasi esimesi täisarve nullist kuni üheksani. Arvu kümme tähistamiseks tuuakse sisse liitsümbol 10. Kõik teised arvud avaldatakse (astendamise, korrutamise ja liitmise tehete abil) nüüd juba nende esimeste täisarvude kaudu, sest näiteks kirjutis

$$43508,741$$

tähendab tegelikult summat

$$4 \cdot 10^4 + 3 \cdot 10^3 + 5 \cdot 10^2 + 0 \cdot 10^1 + 8 \cdot 10^0 + 7 \cdot 10^{-1} + \\ + 4 \cdot 10^{-2} + 1 \cdot 10^{-3}.$$

Selline arvusüsteem on *positsiooniline* selles mõttes, et iga numbri väärtus sõltub tema asukohast (positsioonist) arvus. Toodud juhul näiteks vasakult esimesena kirjutatud number 4 tähendab nelja kümnetuhandelist, eelviimasena kirjutatud 4 aga ainult nelja sajandikku.

Erinevate numbrimärkide arvu järgi nimetatakse seda arvusüsteemi *kümnendsüsteemiks* ja arvu kümme tema *aluseks*.

Analoogilisel põhimõttel saab konstrueerida arvusüsteeme ka kümnest erineval alusel [1], mis paljude ülesannete lahendamisel (sealhulgas ka arvutusmasinates)

---

[1] Ei tule arvata, nagu oleks kümnendsüsteem siiski põhimõtteliselt kõige sobivam ja loomulikum arvusüsteem. Igapäevases tarvitamises on ta vaid sellepärast, et me oleme temaga harjunud. Ajalooliselt hakati kümnendsüsteemi eelistama ilmselt sel põhjusel, et inimesel on kümme sõrme.

osutuvad kümnendsüsteemist otstarbekohasemateks. Kui aluseks on valitud arv k, siis vastavat arvusüsteemi nimetatakse k-ndsüsteemiks. Kümmet mitte ületava aluse korral võib k-ndsüsteemi numbrimärkideks valida tavalised numbrid 0, 1 . . ., $k-1$, kuna aga vastasel juhul peab neile lisama veel uusi sümboleid. Alus k tuleb igal juhul kirjutada kujul 10. Üldiselt tähendab numbrite jada

$$a_n a_{n-1} \ldots a_1 a_0, a_{-1} a_{-2} \ldots a_{-m}$$

k-ndsüsteemis summat

$$a_n \cdot 10^n + a_{n-1} \cdot 10^{n-1} + \ldots + a_1 \cdot 10^1 + a_0 \cdot 10^0 + \\ + a_{-1} \cdot 10^{-1} + a_{-2} \cdot 10^{-2} + \ldots + a_{-m} \cdot 10^{-m},$$

kus 10 kujutab arvu k ning astendajad täisarve (kirjutatud samuti k-ndsüsteemis).

Kaheksand-, kuueteistkümnend- ja kahendsüsteem. Valides $k=8$, s. t. piirdudes ainult kaheksa erineva numbrimärgiga 0, 1, 2, 3, 4, 5, 6 ja 7, saame kaheksandsüsteemi, milles arv kaheksa kirjutatakse kujul 10, arv üheksa kujul 11, arv kümme kujul 12, arv kakskümmend kuus kujul 32, arv[1] **149** kujul 225 jne. Siin viimaste kirjutiste[2] õigsuse kontrollimiseks asendame summades

$$225 = 2 \cdot 10^2 + 2 \cdot 10^1 + 5 \text{ ja } 32 = 3 \cdot 10 + 2$$

(kus 10 tähendab kaheksat) kõik arvud nende vastetega kümnendsüsteemis:

$$\mathbf{2} \cdot \mathbf{8}^2 + \mathbf{2} \cdot \mathbf{8} + \mathbf{5} = \mathbf{149} \text{ ja } \mathbf{3} \cdot \mathbf{8} + \mathbf{2} = \mathbf{26}.$$

Analoogiliselt kontrollides leiame, et näiteks

$$643{,}74 = 6 \cdot 10^2 + 4 \cdot 10 + 3 + 7 \cdot 10^{-1} + 4 \cdot 10^{-2} = \\ = \mathbf{6} \cdot \mathbf{8}^2 + \mathbf{4} \cdot \mathbf{8} + \mathbf{3} + \mathbf{7} \cdot \mathbf{8}^{-1} + \mathbf{4} \cdot \mathbf{8}^{-2} = \mathbf{419{,}9375}$$

---

[1] Siitpeale kasutame selles paragrahvis kümnendsüsteemi arvude kirjutamisel (ka siis, kui nad on asendatud tähtedega) rasvast kirja.

[2] Kuigi arvusüsteemi muutmine peaks endaga tegelikult kaasa tooma ka arvude lugemise muutumise, pole selleks praktikas siiski vajadust. Näiteks kaheksandsüsteemis kirjutatud arvu 32 (=**26**) loetakse lihtsalt «kolm kaks» (aga mitte «kolmkaheksat kaks» või midagi sarnast); arvu 643,74 loeme «kuus neli kolm koma seitse neli» jne. Täpselt samasugust lugemist kasutatakse ka teistes kümnendsüsteemist erinevates arvusüsteemides.

ning

$$12\,132\,047\,573=1\cdot 10^{12}+2\cdot 10^{11}+1\cdot 10^{10}+3\cdot 10^{7}+$$
$$+2\cdot 10^{6}+0\cdot 10^{5}+4\cdot 10^{4}+7\cdot 10^{3}+5\cdot 10^{2}+7\cdot 10+3=$$
$$=\mathbf{1}\cdot \mathbf{8}^{\mathbf{10}}+\mathbf{2}\cdot \mathbf{8}^{\mathbf{9}}+\mathbf{1}\cdot \mathbf{8}^{\mathbf{8}}+\mathbf{3}\cdot \mathbf{8}^{\mathbf{7}}+\mathbf{2}\cdot \mathbf{8}^{\mathbf{6}}+\mathbf{0}\cdot \mathbf{8}^{\mathbf{5}}+\mathbf{4}\cdot \mathbf{8}^{\mathbf{4}}+$$
$$+\mathbf{7}\cdot \mathbf{8}^{\mathbf{3}}+\mathbf{5}\cdot \mathbf{8}^{\mathbf{2}}+\mathbf{7}\cdot \mathbf{8}+\mathbf{3}=\mathbf{1\,365\,790\,587}.$$

K u u e t e i s t k ü m n e n d s ü s t e e m i s tuleb lisaks tavalistele araabia numbritele 0,1, . . . , 9 võtta tarvitusele veel numbrimärgid arvude kümme, üksteist, kaksteist, kolmteist, neliteist ja viisteist kirjutamiseks. Valime[1] nendeks vastavalt sümbolid $\overline{0}$, $\overline{1}$, $\overline{2}$, $\overline{3}$, $\overline{4}$ ja $\overline{5}$. Siis arv **16** tuleb kirjutada kujul 10, arv **17** kujul 11, arv **26** kujul $1\overline{0}$, arv **149** kujul 95 jne. (tõepoolest: $1\overline{0}=1\cdot 10+\overline{0}=\mathbf{1}\cdot\mathbf{16}+\mathbf{10}=\mathbf{26}$, aga $95=9\cdot 10+5=\mathbf{9}\cdot\mathbf{16}+\mathbf{5}=\mathbf{149}$). Arvud **419, 9375** ja **1 365 790 587**, mis me ülal kirjutasime kaheksandsüsteemis, omandavad kuueteistkümnendsüsteemis vastavalt kuju $1\overline{0}3{,}\overline{5}$ ja $11684\overline{5}7\overline{1}$, sest

$$1\overline{0}3{,}\overline{5}=1\cdot 10^{2}+\overline{0}\cdot 10+3+\overline{5}\cdot 10^{-1}=$$
$$=\mathbf{1}\cdot\mathbf{16}^{\mathbf{2}}+\mathbf{10}\cdot\mathbf{16}+\mathbf{3}+\mathbf{15}\cdot\mathbf{16}^{-\mathbf{1}}=\mathbf{419{,}9375}$$

ja

$$11684\overline{5}7\overline{1}=1\cdot 10^{7}+1\cdot 10^{6}+6\cdot 10^{5}+8\cdot 10^{4}+4\cdot 10^{3}+$$
$$+\overline{5}\cdot 10^{2}+7\cdot 10+\overline{1}=\mathbf{1}\cdot\mathbf{16}^{\mathbf{7}}+\mathbf{1}\cdot\mathbf{16}^{\mathbf{6}}+\mathbf{6}\cdot\mathbf{16}^{\mathbf{5}}+\mathbf{8}\cdot\mathbf{16}^{\mathbf{4}}+$$
$$+\mathbf{4}\cdot\mathbf{16}^{\mathbf{3}}+\mathbf{15}\cdot\mathbf{16}^{\mathbf{2}}+\mathbf{7}\cdot\mathbf{16}+\mathbf{11}=\mathbf{1\,365\,790\,587}.$$

Kõige vähem, nimelt kaks erinevat numbrimärki on tarvis k a h e n d s ü s t e e m i s. Kui võtta nendeks numbriteks 0 ja 1, tuleb arv kaks kirjutada kujul 10, arv kolm kujul 11, arv neli kujul 100, arv viis kujul 101, arv **26** kujul 11010, arv **149** kujul 10010101 jne. Siin näiteks

$$10010101=1\cdot 10^{111}+1\cdot 10^{100}+1\cdot 10^{10}+1$$

(10 on arv kaks) annab kümnendsüsteemi üle minnes tõepoolest

$$\mathbf{1}\cdot\mathbf{2}^{\mathbf{7}}+\mathbf{1}\cdot\mathbf{2}^{\mathbf{4}}+\mathbf{1}\cdot\mathbf{2}^{\mathbf{2}}+\mathbf{1}=\mathbf{149}.$$

---

[1] Samaks otstarbeks kasutatakse (näiteks elektron-arvutusmasina M-2 kohta puutuvas kirjanduses) ka vastavalt tähistusi *a*, *b*, *c*, *d*, *e* ja *f*.

Analoogiliselt leiame, et

$$110100011{,}1111 = 1 \cdot 10^{1000} + 1 \cdot 10^{111} + 1 \cdot 10^{101} + 1 \cdot 10 + \\ + 1 + 1 \cdot 10^{-1} + 1 \cdot 10^{-10} + 1 \cdot 10^{-11} + 1 \cdot 10^{-100} = \mathbf{1 \cdot 2^{8} +} \\ \mathbf{+ 1 \cdot 2^{7} + 1 \cdot 2^{5} + 1 \cdot 2 + 1 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} +} \\ \mathbf{+ 1 \cdot 2^{-3} + 1 \cdot 2^{-4} = 419{,}9375}$$

ning

$$1010001011010000100111101111011 = 1 \cdot 10^{11110} + 1 \cdot \\ \cdot 10^{11100} + 1 \cdot 10^{11000} + 1 \cdot 10^{10110} + 1 \cdot 10^{10101} + 1 \cdot \\ \cdot 10^{10011} + 1 \cdot 10^{1110} + 1 \cdot 10^{1011} + 1 \cdot 10^{1010} + 1 \cdot \\ \cdot 10^{1001} + 1 \cdot 10^{1000} + 1 \cdot 10^{110} + 1 \cdot 10^{101} + 1 \cdot 10^{100} + \\ + 1 \cdot 10^{11} + 1 \cdot 10 + 1 = \mathbf{2^{30} + 2^{28} + 2^{24} + 2^{22} + 2^{21} + 2^{19} +} \\ \mathbf{+ 2^{14} + 2^{11} + 2^{10} + 2^{9} + 2^{8} + 2^{6} + 2^{5} + 2^{4} + 2^{3} + 2 + 1 =} \\ \mathbf{= 1\,365\,790\,587.}$$

Kuna arvutusmasinate puhul leiavad peale kümnendsüsteemi kasutamist just toodud kolm arvusüsteemi,[1] siis piirdumegi nende näidetega. Väiksemate naturaalarvude kirjutised kõigis nendes arvusüsteemides on antud tabelis 1.

Sellest tabelist ja toodud arvulistest näidetest on näha, et kümnest suurema aluse (kuusteist) puhul arvude üleskirjutus lüheneb, kuid kümnest väiksema aluse (kaheksa, kaks) korral pikeneb. Sealjuures eriti pikaks muutuvad arvud kahendsüsteemis. Seega arvude «lühendamiseks» tuleb võtta kasutusele uusi numbrimärke, numbrimärkide hulga vähendamine aga «pikendab» arvusid.

Selline olukord kehtib muidugi ka murdude puhul, kuigi toodud näidetest võis jääda vastupidine mulje. Nimelt tuleb murd teisendada teise arvusüsteemi sama täpsusega, milline on murru täpsus lähtesüsteemis. Muude andmete puudumisel hindame murru täpsust sel teel, et loeme kõik väljakirjutatud kohad õigeteks. Näiteks ülalvaadeldud murd **0,9375** on seega teada veaga alla $\mathbf{10^{-4}}$ ning ta tuleb teisendada teistesse arvusüsteemidesse (vähemalt ligikaudselt) sama täpsusega. Kuna $\mathbf{10^{-4} \approx 2^{-13} \approx 8^{-4} \approx 16^{-3}}$, siis selle murru peab vastavalt kahend-, kaheksand- ja kuueteistkümnendsüsteemis õieti kirjutama kujul

$$0{,}1111000000000,\ 0{,}7400 \text{ ja } 0{,}\bar{5}00.$$

---

[1] Arvutusmasina M-2 korral kasutatakse küll veel ka neljandsüsteemi.

Tabel 1

Kümnendsüsteem	Kahendsüsteem	Kaheksandsüsteem	Kuueteistsüsteem	Kümnendsüsteem	Kahendsüsteem	Kaheksandsüsteem	Kuueteistsüsteem
1	1	1	1	26	11010	32	$1\overline{0}$
2	10	2	2	27	11011	33	$1\overline{1}$
3	11	3	3	28	11100	34	$1\overline{2}$
4	100	4	4	29	11101	35	$1\overline{3}$
5	101	5	5	30	11110	36	$1\overline{4}$
6	110	6	6	31	11111	37	$1\overline{5}$
7	111	7	7	32	100000	40	20
8	1000	10	8	33	100001	41	21
9	1001	11	9	34	100010	42	22
10	1010	12	$\overline{0}$	35	100011	43	23
11	1011	13	$\overline{1}$	36	100100	44	24
12	1100	14	$\overline{2}$	37	100101	45	25
13	1101	15	$\overline{3}$	38	100110	46	26
14	1110	16	$\overline{4}$	39	100111	47	27
15	1111	17	$\overline{5}$	40	101000	50	28
16	10000	20	10	41	101001	51	29
17	10001	21	11	42	101010	52	$2\overline{0}$
18	10010	22	12	43	101011	53	$2\overline{1}$
19	10011	23	13	44	101100	54	$2\overline{2}$
20	10100	24	14	45	101101	55	$2\overline{3}$
21	10101	25	15	46	101110	56	$2\overline{4}$
22	10110	26	16	47	101111	57	$2\overline{5}$
23	10111	27	17	48	110000	60	30
24	11000	30	18	49	110001	61	31
25	11001	31	19	50	110010	62	32

Teisendamine kümnendsüsteemist teistesse arvusüsteemidesse. Ülaltoodud näidetes üksikute (kümnendsüsteemis etteantud) arvude kirjutamisest kuueteistkümnend-, kaheksand- ja kahendsüsteemis andsime küll juhise toodud kirjutiste õigsuse kontrollimiseks (mis seisnes tegelikult tagasiteisendamises kümnendsüsteemi), kuid me ei näidanud, millisel viisil need kirjutised on saadud. Anname nüüd eeskirja (algoritmi), mis võimaldab leida kümnendsüsteemis antud arvu kirjutist mingis teises arvusüsteemis [1].

[1] Arvude teisendamine mistahes arvusüsteemist teise toimub tegelikult sama algoritmi järgi, kuid kirjutise lihtsustamiseks piirdume esialgu vaadeldava erijuhuga (kui ühe tähtsamaga).

Vaatleme kõigepealt kümnendsüsteemis antud täisarvu **N** teisendamist mingisse *k*-ndsüsteemi.

Oletame ajutiselt, et meil on juba teada selle arvu kuju *k*-ndsüsteemis:

$$N = a_n 10^n + a_{n-1} 10^{n-1} + \ldots + a_1 10 + a_0,$$

kus $a_n, \ldots, a_o$ on *k*-ndsüsteemi numbrid ja 10 selle süsteemi alus (s. t. arv *k*). Kui viimase võrduse paremas pooles kõik arvud asendada nende kirjutistega kümnendsüsteemis, siis saame tagasi arvu **N**:

$$\mathbf{N} = \mathbf{a_n k^n + a_{n-1} k^{n-1} + \ldots + a_1 k + a_0}$$

(siin näiteks $\mathbf{a_n}$ tähendab *k*-ndnumbri a_n kirjutist kümnendsüsteemis).

Järelikult arvu **N** jagamisel arvuga **k** tekib jagatis

$$\mathbf{N_1} = \mathbf{a_n k^{n-1} + a_{n-1} k^{n-2} + \ldots + a_1}$$

ja jääk $\mathbf{a_0}$. Saadud jääk kujutab endast seega arvu *N* viimast numbrikohta *k*-ndsüsteemis (kirjutatult kümnendsüsteemi arvuna [1]). Jagatist $\mathbf{N_1}$ uuesti arvuga **k** jagades leiame uue jagatise

$$\mathbf{N_2} = \mathbf{a_n k^{n-2} + a_{n-1} k^{n-3} + \ldots + a_2}$$

ja jäägi $\mathbf{a_1}$. See jääk on võrdne arvu *N* eelviimase numbrikohaga *k*-ndsüsteemis (kirjutatud jälle kümnendsüsteemi arvuna). Tekkivaid jagatisi niimoodi järjest edasi jagades saame jääkideks järjekorras arvu *N* kõik numbrikohad $\mathbf{a_0}, \mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_n}$ (jagamise lõpetame siis, kui jagatis osutub nulliks). Jääb vaid kirjutada need arvud *k*-ndsüsteemi numbritena ja ülesanne ongi lahendatud.

Näiteid. 1. Teisendada arv **765** kaheksand- ja kuueteistkümnendsüsteemi. Korduvalt jagades leiame:

---

[1] Kui **k**<**10**, siis see märkus pole oluline, sest *k*-ndsüsteemi numbrid kirjutatakse sel juhul samuti nagu kümnendsüsteemi vastavad numbrid.

765	8		
72	95	8	
45	8	11	8
40	15	8	1
5	8	3	
	7		

765	16	
64	47	16
125	32	2
112	15	
13		

Saadud jääke *vastupidises järjekorras* üles kirjutades näeme, et arv **765** omandab kaheksandsüsteemis kuju 1375, kuueteistkümnendsüsteemis aga kuju $2\overline{5}\overline{3}$.

Sellise korduva jagamise üleskirjutamist võib väikese k puhul lihtsustada. Nimelt saab vajalikud jagamised siis erilise vaevata teostada peast ning välja tuleb iga kord kirjutada vaid jagatised ja jäägid. Kirjutised on sealjuures sobiv paigutada nii, et tekkivad jagatised kirjutame ühte ritta *paremalt vasakule* ning jäägid nendest allapoole järgmisse ritta. Kui read eraldada veel joonega, siis joone all saamegi vaadeldava arvu kirjutise k-ndsüsteemis. Näiteks arvu **765** teisendamisel kaheksandsüsteemi omandavad ülaltoodud kirjutised kuju

1	11	95	765
1	3	7	5.

2. Teisendada arv **2431** kahendsüsteemi. Kasutades jagamise lühendatud üleskirjutamist (kahendsüsteemi teisendamisel on see eriti otstarbekohane) leiame:

1	2	4	9	18	37	75	151	303	607	1215	2431
1	0	0	1	0	1	1	1	1	1	1	1

Seega **2431** = 100101111111.

Vaatleme nüüd *lihtmurru* **D** teisendamist kümnendsüsteemist k-ndsüsteemi. Oletame jälle, et selle murru kuju k-ndsüsteemis

$$D = a_{-1}10^{-1} + a_{-2}10^{-2} + a_{-3}10^{-3} + \dots$$

on juba teada ja läheme selles kirjutises tagasi kümnendsüsteemi:

$$D = a_{-1}k^{-1} + a_{-2}k^{-2} + a_{-3}k^{-3} + \dots$$

Korrutades viimast võrdust arvuga **k** saame:

$$Dk = a_{-1} + a_{-2}k^{-1} + a_{-3}k^{-2} + \dots = a_{-1} + D_1,$$

kus a_{-1} on tulemuse täis- ja D_1 murdosa. Korrutades seda murdosa jälle arvuga **k** leiame:

$$D_1 k = a_{-2} + D_2,$$

kus a_2 on täis- ja D_2 murdosa. Niimoodi jätkates leiame murru D numbrikohad k-ndsüsteemis (kümnendsüsteemi arvudena kirjutatult) kui täisosad, mis tekivad arvu **D** ja vahepealsete korrutiste murdosade korrutamisel arvuga **k**. Korrutamist jätkame seni, kuni oleme saanud nii palju k-ndsüsteemi numbrikohti, et täpsus vastab lähtemurru täpsusele. Nimelt kui lähtemurd on antud **n** kümnendkohaga, siis k-ndsüsteemis tuleb leida **m** numbrikohta, kus **m** määratakse võrrandit $10^n = k^m$ parimini rahuldava naturaalarvuna.

Väikese k puhul ($k < 10$) saab kõik vajalikud arvutused teostada jälle peast, üles tuleb kirjutada vaid vahepealsed korrutised. Kui aga k on kümnendsüsteemis mitmekohaline arv ($k > 10$), siis peastarvutamine pole enam läbiviidav ja üles tuleb iga kord kirjutada ka osakorrutised. Arvutused on viimasel juhul otstarbekohane paigutada tulbana, eraldades täisosad murdosadest püstjoonega.

Näide. Teisendada kümnendmurd **0,840315** kuueteistkümnend- ja kaheksandmurdudeks.

Antud juhul lähtemurd on antud kuue numbrikohaga.

Et

$$10^6 \approx 16^5 = 1048476$$

ja

$$10^6 \approx 8^7 = 2096952,$$

siis tuleb meil kuueteistkümnendmurd leida viie ja kaheksandmurd seitsme numbrikohaga.

Teisendamisel kuueteistkümnendsüsteemi võib arvutused paigutada näiteks järgmiselt:

×	**0,**	**840315**
		16
+	**5**	**041890**
	8	**40315**
×	**13**	**445040**
		16
+	**2**	**670240**
	4	**45040**
×	**7**	**120640**
		16
+		**723840**
	1	**20640**
×	**1**	**930240**
		16
+	**5**	**581440**
	9	**30240**
×	**14**	**883840**
		16
+	**5**	**303040**
	8	**83840**
	14	**141440**

Järelikult omandab murd **0,840315** kuueteistkümnendsüsteemis kuju $0,\overline{3}71\overline{44}$.

Väikese *k* puhul kirjutame vahepealsete korrutiste m u r d o s a d ühte ritta v a s a k u l t p a r e m a l e, täisosad aga üks rida allapoole. Vaadeldava arvu teisendamisel kaheksandsüsteemi omandavad kirjutised siis kuju

$$\begin{array}{cccccccc} \mathbf{0{,}840315} & \mathbf{722520} & \mathbf{780160} & \mathbf{241280} & \mathbf{930240} & \mathbf{441920} & \mathbf{535360} & \mathbf{282880} \\ \hline & \mathbf{6} & \mathbf{5} & \mathbf{6} & \mathbf{1} & \mathbf{7} & \mathbf{3} & \mathbf{4} \end{array}$$

Seega kaheksandsüsteemis **0,840315** $=0{,}6561734$.

Kui kümnendsüsteemist tuleb k-ndsüsteemi teisendada segaarv, siis teisendame ülalkirjeldatud viisil eraldi tema täisosa ja murdosa. Näiteks arv **765,840315** omandab kuueteistkümnendsüsteemis kuju $2\overline{53},\overline{3}71\overline{44}$.

Väikese k puhul lihtsustatud kirjutusviisi kasutamisel võib kõik arvutused paigutada ühte ritta, eraldades täis- ja murdosa teisendamise komaga. Näiteks arvu **375,28** teisendamisel kahendsüsteemi kirjutame

1	2	5	11	23	46	93	187	375,28	56	12	24	48	96	92	84
1	0	1	1	1	0	1	1	1,	0	1	0	0	0	1	1

Seega **375,28** = 101110111,0100011.

Seos kahendsüsteemi ja kaheksandsüsteemi vahel. Eriti lihtsaks osutub arvude teisendamine juhul, kui ühe arvusüsteemi alus on teise arvusüsteemi aluse täisaste. Kuna meil edaspidi läheb sedalaadi teisendustest tarvis just kahendsüsteemi teisendamist kaheksandsüsteemi (ja vastupidi), siis peatumegi sellel erijuhul.

Käsitluse lihtsustamiseks kirjutame arvusüsteemide alused (kaheksa ja kaks) ning nende astendajad kümnendsüsteemis. Tähtedega b_i tähistame kaheksandsüsteemi ja tähtedega a_i kahendsüsteemi numbrimärke.

Vaatleme kaheksandsüsteemis antud täisarvu

$$N = b_n \ldots b_1 b_0 = b_n \cdot 8^n + \ldots + b_1 \cdot 8 + b_0$$

ja oletame, et meil on õnnestunud leida tema kuju kahendsüsteemis

$$N = a_k \ldots a_2 a_1 a_0 = a_k \cdot 2^k + \ldots + a_2 \cdot 2^2 + a_1 \cdot 2 + a_0.$$

Neid võrdseid arvusid kaheksaga jagades (ning arvestades, et $8 = 2^3$) saame võrdsed jäägid

$$b_0 = a_2 \cdot 2^2 + a_1 \cdot 2 + a_0 = a_2 a_1 a_0$$

ja võrdsed jagatised

$$b_n \cdot 8^{n-1} + \ldots + b_2 \cdot 8 + b_1 = a_k \cdot 2^{k-3} + \ldots + a_4 \cdot 2 + a_3.$$

Viimast võrdust uuesti kaheksaga jagades saame jälle võrdsed jäägid

$$b_1 = a_5 \cdot 2^2 + a_4 2^2 + a_3 = a_5 a_4 a_3$$

jne. Seega: kaheksandarvu teisendamisel kahendarvuks tuleb iga kaheksandsüsteemi number asendada temaga võrdse kolmekohalise kahendarvuga. Näiteks kaheksandarvu 7436 teisendamisel arvestame, et $7=111$, $4=100$, $3=011$, $6=110$ ja seega selle arvu kuju kahendsüsteemis on

$$111100011110.$$

Sama reegel jääb kehtima ka murdude teisendamisel kaheksandsüsteemist kahendsüsteemi. Tõepoolest, kui

$$D=0,\, b_{-1}b_{-2}\ldots b_{-m},$$

siis $D \cdot 8^m$ on (m-kohaline) täisarv. Pärast selle täisarvu teisendamist kahendsüsteemi tuleb saadud $3m$-kohaline kahendarv vaid jagada suurusega $8^m=2^{3m}$, s. t. paigutada koma nii, et temast paremal oleks $3m$ numbrit. Näiteks kaheksandmurd 0,1536 omandab kahendsüsteemis kuju 0,001101011110, segaarv 273,265 aga kuju

$$10111011{,}010110101.$$

Nüüd on lihtne näha, millise reegli järgi toimub kahendarvude teisendamine kaheksandarvudeks. Selleks tuleb kahendarv komast alates vasakule ja paremale jaotada kolmenumbrilisteks rühmadeks (kui äärmised rühmad ei sisalda kolme numbrit, siis täiendada nad kolmenumbrilisteks nullide lisamise teel), saadud rühmad kui kolmekohalised kahendarvud teisendada kaheksandsüsteemi (peast!) ning kirjutada saadud kaheksandsüsteemi numbrid samas järjekorras, nagu paiknesid vastavad kahendnumbrite rühmad. Koma tuleb paigutada nende kaheksandnumbrite vahele, milledele vastavate rühmade vahel paiknes koma kahendarvus.

Näited:

$$1011010{,}10111=001\ 011\ 010{,}101\ 110=132{,}56;$$
$$1101111011{,}1101001=011\ 011\ 111\ 011{,}110\ 100\ 100=$$
$$=3373{,}644.$$

Täiesti samasuguste reeglite järgi toimub teisendamine kuueteistkümnendsüsteemist kahendsüsteemi ja vastupidi. Kogu erinevus seisneb vaid selles, et esimesel juhul tuleb kuueteistkümnendarvu iga number asendada n e l j a k o h a l i s e kahendarvuga ja teisel juhul peab kahendarvu jaotama neljakohalisteks rühmadeks.

Näited:

$$1\bar{4}3\bar{0},\bar{2}=0001\ 1110\ 0011\ 1010,1100=1111000111010,1100;$$
$$10101100001111,10011=0010\ 1011\ 0000\ 1111,1001\ 1000=$$
$$=2\bar{1}0\bar{5},98.$$

Viimaste tulemuste ühe rakendusena märgime, et praktikas teisendatakse kümnendarvud kahendarvudeks mitte vahetult, vaid kaheksandsüsteemi kaudu. See näiliselt keerulisem tee lihtsustab tunduvalt nii arvutusi kui ka kirjutusi.

Näide. Teisendada arv **725,3333**... kahendsüsteemi. Kõigepealt teisendame selle arvu kaheksandsüsteemi:

1	11	90	725,3333	6664	3312	6496	1968
1	3	2	5,	2	5	2	5

Seega saime kaheksandarvu 1325,2525..., mis kahendsüsteemis omandab kuju 1011010101,010101010101...

Aritmeetika kaheksand- ja kahendsüsteemis. Arvusüsteemi muutmisega ei muutu aga mitte ainult arvude kirjutamine, vaid ka tehted nendega. Näiteks omandab liitmistabel kaheksandsüsteemis tabelis 2 näidatud kuju.

Kahe ühekohalise arvu summa leiame tabelist 2 ühe liidetava numbrit kandva rea ja teise liidetava numbrit kandva veeru lõikekohal. Näiteks

$$3+4=7, \quad 5+5=12, \quad 6+7=15.$$

Mitmekohaliste kaheksandarvude liitmine toimub tabeli 2 abil samade reeglite järgi nagu kümnendarvude liitmine:

$$\begin{array}{r} 165,343 \\ +\ 2074,14 \\ \hline 2261,503. \end{array}$$

Sama tabelit saab kasutada ka kaheksandarvude lahutamisel. Näiteks selleks, et lahutada arvust 14 arv 7, leiame reas, mille järjekorranumbriks on 7, arvu 14. Selle

Tabel 2

+	0	1	2	3	4	5	6	7	10
0	0	1	2	3	4	5	6	7	10
1	1	2	3	4	5	6	7	10	11
2	2	3	4	5	6	7	10	11	12
3	3	4	5	6	7	10	11	12	13
4	4	5	6	7	10	11	12	13	14
5	5	6	7	10	11	12	13	14	15
6	6	7	10	11	12	13	14	15	16
7	7	10	11	12	13	14	15	16	17
10	10	11	12	13	14	15	16	17	20

Tabel 3

×	0	1	2	3	4	5	6	7	10
0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	10
2	0	2	4	6	10	12	14	16	20
3	0	3	6	11	14	17	22	25	30
4	0	4	10	14	20	24	30	34	40
5	0	5	12	17	24	31	36	43	50
6	0	6	14	22	30	36	44	52	60
7	0	7	16	25	34	43	52	61	70
10	0	10	20	30	40	50	60	70	100

veeru number (antud juhul 5), milles viimane arv asub, ongi tulemuseks. Seega $14-7=5$. Analoogiliselt

$$12-4=6, \quad 7-3=4, \quad 13-6=5 \text{ jne.}$$

Mitmekohaliste arvude lahutamine toimub jälle samade reeglite järgi nagu kümnendsüsteemis:

$$\begin{array}{r} -\ 325{,}14 \\ 175{,}513 \\ \hline 127{,}425. \end{array}$$

Kaheksandarvude korrutamisel ja jagamisel tuleb (lisaks liitmistabelile) kasutada tabelit 3 (korrutamistabel).

Näiteid:

$$\begin{array}{r} 25{,}332 \\ \times\ 147{,}6 \\ \hline 200434 \\ 225766 \\ +\ 125550 \\ 25332 \\ \hline 4356{,}7314 \end{array}$$

$$\begin{array}{l|l} 257427{,}26 & 325 \\ \cline{2-2} 2376 & 645{,}76 \\ \cline{1-1} 1762 & \\ 1524 & \\ \cline{1-1} 2367 & \\ 2051 & \\ \cline{1-1} 3162 & \\ 2723 & \\ \cline{1-1} 2376 & \\ 2376 & \\ \cline{1-1} 0 & \end{array}$$

Eriti lihtsaks osutub aritmeetika kahendsüsteemis. Liitmisel ja lahutamisel tuleb siin kasutada vastavalt tabeleid

$$\boxed{\begin{array}{c} 0+0=\ 0 \\ 0+1=\ 1 \\ 1+0=\ 1 \\ 1+1=10 \end{array}} \quad \text{ja} \quad \boxed{\begin{array}{c} 0-0=0 \\ 1-0=1 \\ 1-1=0 \\ 10-1=1 \end{array}}$$

Näited:

$$\begin{array}{r} 1101110{,}01 \\ +\ 11111{,}1 \\ \hline 10001101{,}11 \end{array} \qquad \begin{array}{r} 10111011{,}101 \\ -\ 1110101{,}11 \\ \hline 1000101{,}111. \end{array}$$

Korrutamine ja jagamine toimuvad aga tabeli

$$\boxed{\begin{array}{c} 0 \cdot 0 = 0 \\ 0 \cdot 1 = 0 \\ 1 \cdot 0 = 0 \\ 1 \cdot 1 = 1 \end{array}}$$

abil.

Näiteid:

$$\begin{array}{rr}
\times & 11011{,}101 \\
 & 1001{,}11 \\
\hline
 & 11011101 \\
+ & 11011101 \\
 & 11011101 \\
 & 11011101 \\
\hline
 & 100001101{,}01011
\end{array}
\qquad
\begin{array}{l|l}
11010011{,}11 & 10110 \\
\cline{2-2}
\underline{10110} & 1001{,}101 \\
100011 & \\
\underline{10110} & \\
11011 & \\
\underline{10110} & \\
10110 & \\
\underline{10110} & \\
0 &
\end{array}$$

Peale senivaadeldud arvusüsteemide leiab mõnedes elektron-arvutusmasinates (näit. M-2) kasutamist veel n e l j a n d s ü s t e e m. Selle süsteemi puhul on tegemist nelja erineva numbriga: 0, 1, 2 ja 3, kusjuures liitmistabel omab kuju:

+	0	1	2	3
0	0	1	2	3
1	1	2	3	10
2	2	3	10	11
3	3	10	11	12

Neljandarvu teisendamisel kahendarvuks tuleb iga neljandsüsteemi number asendada temaga võrdse k a h e - k o h a l i s e kahendarvuga. Näiteks neljandarvu 3102 kuju kahendsüsteemis on 11010010.

Üleminek kahendsüsteemile. Diskreetse toimega arvutusmasinates kujutatakse matemaatilisi suurusi arvudena. Sealjuures arvu iga numbrikoha kujutamiseks kasutatakse elementi, mis võib omandada nii palju erinevaid seisundeid, kui palju numbrimärke on kasutatavas arvusüsteemis.

Kiirelt töötavaid ning kümmet rangelt erinevat seisundit omandavaid elemente on praktiliselt väga raske konstrueerida. Seepärast kasutatakse kiirelt töötavates arvutusmasinates peaaegu eranditult[1] kahendsüsteemi, milles on teatavasti ainult kaks erinevat numbrimärki ning mille puhul numbreid kujutavad elemendid peavad seega suutma omandada vaid kaht erinevat seisundit. Mis puutub vahetult kümnendsüsteemis töötavatesse elektronarvutusmasinatesse, siis neis kasutatakse numbrite salvestamiseks mitte kümnepositsioonilisi elemente, vaid neljast kahendelemendist koosnevaid blokke. Iga kümnendnumber kujutatakse sellises masinas teatava neljakohalise kahendarvuna. Neid masinaid me käesolevas raamatus ei käsitle[2].

Kahendsüsteemi kasutamine arvutusmasinas toob küll kaasa lisatööd, mis on seotud algandmete teisendamisega kahendsüsteemi ja tulemuste tagasiteisendamisega kümnendsüsteemi, kuid kuna need tehted teostab masin ise (pealegi on enamiku ülesannete puhul arvutuste tohutu hulga juures tegemist suhteliselt väheste lähteandmete ja tulemustega), siis aja kaotus arvusüsteemide muutmisele osutub tühiseks. Kahendsüsteemi kasutamine pikendab ka märgatavalt arve, nõudes seega suurema hulga elementide kasutamist arvu kujutamiseks, kuid kaheseisundiliste elementide lihtsam (ja seega odavam) ehitus kompenseerib selle kulutuse täielikult.

Kaheseisundiliste elementide füüsikalise ehituse põhimõtteid selgitame järgmises peatükis. Esialgu eeldame lihtsalt, et sellised elemendid on olemas ning loeme, et üks seisunditest tähendab numbrit 0 ja teine numbrit 1.

---

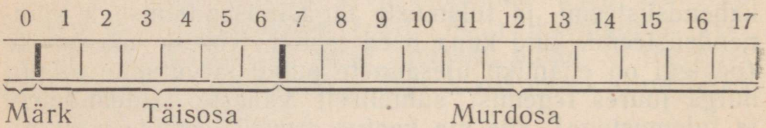
[1] On ehitatud vaid üksikuid elektron-arvutusmasinaid, mis töötavad kolmendsüsteemis.

[2] Lähemalt vt. näit. Р. К. Ричардс. Арифметические операции на цифровых вычислительных машинах. Москва, 1957.

Ühe kahendarvu salvestamiseks määratud seadis — **p e s a** — peab seega koosnema nii mitmest elemendist, kui mitme tüvenumbriga kahendarve soovitakse temas salvestada (s. t. millise täpsusega töötamiseks masin on ette nähtud). Neid elemente nimetame lihtsuse mõttes vastava pesa **k o h t a d e k s.** Enamasti kasutatakse universaalsetes elektron-arvutusmasinates kolmekümne kuni kolmekümne viie tüvenumbriga kahendarve, mis vastab üheksale-kümnele tüvenumbrile kümnendsüsteemis (on muidugi konstrueeritud ka masinaid, mille pesas on rohkem või vähem kohti). Nagu me hiljem näeme, on tegelik täpsus siiski väiksem.

Arvude kujutamiseks kasutatava mooduse järgi jaotatakse masinad kahte klassi: fikseeritud komaga masinad ja liikuva komaga masinad.

Fikseeritud komaga masinates kujutatakse arvud nende loomulikul kujul, s. t. numbrite jadana, mis on komaga jaotatud täis- ja murdosaks. Niisuguse masina mälus on iga pesa esimene koht (järgnevas skeemis koht nr. 0) määratud arvu märgi kujutamiseks, edasi teatud kindel arv kohti (kohad nr. 1—6) arvu täisosa ja ülejäänud kohad (kohad nr. 7—17) murdosa kujutamiseks. Piltlikkuse saavutamiseks esitame sellise pesa järgmise skeemina:



Siin on lihtsuseks esialgu piirdutud kaheksateist-kohalise pesaga, kusjuures kuus kohta on määratud täisosa ja üksteist kohta murdosa kujutamiseks (s. t. koma on fikseeritud kuuenda koha järel). Koma ennast masinas kuidagi ei kujutata, sest masina kõik seadmed on selliselt konstrueeritud, et koma paikneb alati ettenähtud kindlale kohale.

Arvu märgi kujutamiseks lepitakse näiteks kokku, et

$$«+»=0 \text{ ja } «-»=1.$$

Näiteks kümnendarv[1] +45,2584 omab pärast kahend-

[1] Lootes, et lugeja on erinevate arvusüsteemidega juba harjunud, ei kasuta me kümnendsüsteemi tähistamiseks enam rasvast kirja.

süsteemi üleviimist kuju 101101,010000100010011 ning tema salvestamise ajal on pesa kohad täidetud järgmiselt:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1	0	1	1	0	1	0	1	0	0	0	0	1	0	0	0	1

(murdosa viimased kohad jäävad kujutamata, sest pesas pole rohkem ruumi).

Kümnendarv $-19{,}6$ omandab kahendsüsteemis kuju

$$-10011{,}1001100110011001 1\ldots$$

Et vaadeldav pesa mahutab vaid üheteistkohalise murdosa, siis tuleb teostada ümardamine. Seega selle arvu kujutis pesas on

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	0	0	1	1	1	0	0	1	1	0	0	1	1	0	1

Fikseeritud komaga masina pesas saab järelikult kujutada arve ainult suhteliselt kitsast vahemikust. Vaadeldaval juhul näiteks ei tohi kujutatava arvu absoluutväärtus olla suurem kui

$$63\frac{2047}{2048} = 111111{,}11111111111 \,.$$

ega väiksem kui $\frac{1}{4096} = 2^{-12}$. Arvud, mis on absoluutväärtuselt väiksemad kui 2^{-12}, kujutuvad masinas nullidena — need on nn. m a s i n a n u l l i d. Arv $\pm 2^{-12}$ ise kujutub (ümardamise tagajärjel) pesas arvuna $\pm 2^{-11} = \pm$ $\pm 0{,}00000000001$. Arv, mille absoluutväärtus on $63\frac{4095}{4096}$ või veelgi suurem, ümardub kahendarvuks, kus täisosa sisaldab vähemalt seitse numbrikohta. Pesas kujutub sel juhul seitsmes number märgi jaoks ettenähtud kohale, kuna aga täisosa ülejäänud (s. t. kaheksas jne.) numbrid ja märk kaovad üldse. Näiteks kümnendarv $-181{,}3$ (ehk kahendsüsteemis $-10110101{,}0101\ldots$) kujutub pesas selliselt:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1

,

s. t. täpselt samuti nagu arv $+53{,}3$ (kahendsüsteemis $+110101{,}0101\ldots$). Niisugust nähtust nimetatakse *pesa ületäitumiseks.*

Kuna ületäitumine moonutab arvud tundmatuseni, siis peab seda fikseeritud komaga masinais tingimata vältima. Seda tüüpi masin on muide konstrueeritud nii, et pesa ületäitumise korral ta jääb kas seisma või teostab kõikide arvude vähendamise varem ettenähtud programmi kohaselt (sellega seoses väheneb muidugi arvutuste täpsus). Ületäitumise vältimiseks teisendatakse lahendatav ülesanne alati niisugusele kujule, et nii lähteandmed kui ka kõik vahepealsete arvutuste tulemused kuuluksid lubatavasse piirkonda. See saavutatakse eriliste mastaabikordajate sissetoomisega, s. t. kõigi esinevate suuruste mõõtühikute muutmisega.

Mastaabikordajate valik on küllaltki tülikas ja aeganõudev töö. Selle lihtsustamiseks on vaadeldavat tüüpi masinates koma enamasti fikseeritud esimese numbrikoha ees (s. t. otsekohe pärast märgikohta). Masina mälus saab sel korral kujutada ainult lihtmurde — arve, mis asuvad -1 ja $+1$ vahel. Koma niisugune fikseerimine garanteerib, et arvude korrutamisel ei saa tekkida pesa ületäitumist.

Fikseeritud komaga masinaks on näiteks nõukogude elektron-arvutusmasin «Ural», mille mälu igas pesas on kolmkümmend kuus kohta, kusjuures koma on fikseeritud märgikoha järel. Seega saab selles masinas kujutada arvusid, mille absoluutväärtus asub

$$2^{-35}=\frac{1}{34359738368}$$

ja

$$1-2^{-35}=\frac{34359738367}{34359738368}$$

vahel. Näiteks arv

$$-0{,}5234763214=-0{,}100001100000001010001011010100001010$$

kujutub masina «Ural» 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	24	25	26	27	28	29	30	31	32	33	34	35
1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1	1	0	1	0	1	0	0	0	0	1	0	1

Tarbe korral (näiteks siis, kui ülesande võib lahendada väiksema täpsusega) saab siin üht pesa kasutada ka kahe «lühikese» pesana seitsmeteist-kohaliste kahendmurdude salvestamiseks. Sel juhul kujutatakse teise arvu märk kohal nr. 18:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35

$\uparrow$ Esimese arvu märk — Esimene arv — $\uparrow$ Teise arvu märk — Teine arv

Koma on mõlemas «lühikeses» pesas ikka fikseeritud märgikoha järel ja absoluutväärtuselt suuremate kui $\frac{131071}{131072}$ arvude korral toimub pesa ületäitumine. Kui sealjuures ületäitumine toimub teises «lühikeses» pesas, siis moonutuvad sellega ka esimeses pesas oleva arvu viimased kohad.

Teiseks levinumaks fikseeritud komaga nõukogude elektron-arvutusmasinaks on M-3. Tema mälu pesa sisaldab 31 kohta, kusjuures koma on samuti fikseeritud märgikoha järel.

Arvude kujutamist äsjakirjeldatud viisil võib tõlgendada ka teisiti. Nimelt kui märgikohta vaadelda arvu täisosana, siis positiivsed lihtmurrud on kujutatud oma loomulikul kujul (märki «+» tähendav number 0 on tõlgendatav kui täisosa, mis lihtmurru puhul on samuti null). Negatiivse lihtmurru x asemel kujutub pesas aga arv, mille täisosa on 1 ja murdosa numbrid samad, mis arvul x, s. t. pesas kujutub sel korral tegelikult arv $1-x$ $(=1+|x|$, sest x on ju negatiivne).

Lihtkahendmurru x otsekoodiks nimetatakse arvu $[x]_{ok}$, mis on defineeritud valemiga

$$[x]_{ok}=\begin{cases} x, & \text{kui } x \geqslant 0 \\ 1-x, & \text{kui } x \leqslant 0. \end{cases}$$

Näiteks: $[+0{,}10111001]_{ok}=0{,}10111001$
$[-0{,}10110110]_{ok}=1{,}10110110.$

Arvul null on kaks otsekoodi: $[0]_{ok}-0,00\ldots0$ ja $[0]_{ok}=$ $=1,00\ldots0$.

Seega märgikoha järel fikseeritud komaga arvutusmasina pesas kujutub iga arv tema otsekoodina.

Fikseeritud komaga masinate oluliseks eeliseks on odavus. See on tingitud eeskätt sellest, et tehted arvudega toimuvad n. ö. tavalisel viisil ning seetõttu on aritmeetilise seadme ehitamine lihtsam.

Liikuva komaga masinad. Nagu juba ülal nimetatud, on fikseeritud komaga arvutusmasinatel oluline puudus: kasutatavate arvude väike ulatus tingib vajaduse mastaabikordajate leidmiseks, mis on enamasti seotud küllaltki suure tööga.

Sellest puudusest on vabad nn. liikuva komaga masinad, mille mälu pesas on peale märgi- ja numbrikohtade ette nähtud veel kohad, mis näitavad koma asukohta arvus. Niisuguse masina pesas kujutamiseks tuleb kahendarv enne teisendada nn. poollogaritmilisele kujule.

$$N=m\cdot10^{p},$$

kus 10 tähendab kaks, $|m|<1$ ja p on positiivne või negatiivne täisarv. Tegurit m nimetatakse arvu N mantissiks ja astendajat p selle arvu järguks. Järk p näitab seega, mitme koha võrra ja millises suunas tuleb arvu N saamiseks mantissis m (kui lihtkahendmurrus) koma nihutada.

Ilmselt ei ole arvu kujutamine poollogaritmilisel kujul ühene.

Näiteks:

$$N_1=+0,01011011=+0,01011011\cdot10^{0}=+0,1011011\cdot10^{-1};$$
$$N_2=-101,01101=-0,010101101\cdot10^{100}=$$
$$=-0,10101101\cdot10^{11}.$$

Ühesuse saamiseks seatakse nõue (0,1 on üks kahendik)

$$0,1\leqslant|m|<1,$$

s. t. nõutakse, et mantiss omaks kahendsüsteemis kuju 0,1 ... Kui see nõue on täidetud, siis öeldakse, et arv on esitatud normaalkujul. Toodud näidetes viimased kirjutised ongi antud normaalkujul.

Liikuva komaga arvutusmasina pesas salvestatakse arv enamasti järgmiselt: esimesel kohal kujutatakse mantissi (ja seega ka arvu) märk, edasi on teatud arv kohti määratud mantissi murdosa, siis üks koht järgu märgi ja viimased kohad järgu absoluutväärtuse kujutamiseks. Seega mantiss on kujutatud oma otsekoodina.

Nii sisaldab nõukogude elektron-arvutusmasina «Strela» pesa nelikümmend kolm kohta, mis on jaotatud järgmise skeemi kohaselt:

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

23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
0	1	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0	1	1	1

ntissi rk — Mantiss — Järgu märk — Järk

Siin on (normaalkujul) kujutatud arv

$$-1011110{,}1001011010111100100000101111 =$$
$$= -0{,}10111101001011010111100100000101111 \cdot 10^{111}.$$

Veidi teisiti kujutatakse liikuva komaga arve[1] nõukogude elektron-arvutusmasinas M-2. Tema mälu pesa sisaldab 34 kahendkohta, mis on jaotatud järgmiselt:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1	1	0	1	1	0	1	0	1	1	1	0	0	1	1	0

18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
1	0	1	1	0	1	0	1	1	0	1	1	0	1	1	1	1

Järk+32 — Mantissi täisosa — Mantissi murdosa — Arvu märk

Esimesel kuuel kohal kujutatakse järgu p asemel $p+32$. Näiteks kui $p=+101$, siis pesas salvestatakse see kujul 100101; kui aga $p=-101$, siis kujul 011011. Koht nr. 7 on reserveeritud mantissi täisosa jaoks, mis alati võrdub nulliga. Kohtadel nr. 8—33 kujutatakse mantissi murdosa ja kohal nr. 34 arvu märk, kusjuures kasutatakse kokkulepet

[1] Arvutusmasinat M-2 võib kasutada ka fikseeritud komaga masinana; siis on esimesed 33 kohta määratud arvu murdosa ja viimane koht märgi kujutamiseks.

$$«+»=1 \text{ ja. } «-»=0.$$

Ülaltoodud skeemis on kujutatud arv

$$=+0{,}00000101110011010110101101101111=$$
$$=+0{,}101110011010110101101101111 \cdot 10^{-101}.$$

Liikuva komaga masinais saab seega kujutada arve tunduvalt ulatuslikumast vahemikust kui fikseeritud komaga masinais. See kaotab vajaduse mastaabikordajate leidmiseks. Näiteks saab arvutusmasinas «Strela» kujutada positiivseid arve alates arvust

$$\underbrace{0{,}000\ldots001}_{34} \cdot 10^{-111111}=2^{-98}=\underbrace{0{,}000\ldots00318}_{29}\ldots$$

ning lõpetades arvuga

$$\underbrace{0{,}11\ldots11}_{35} \cdot 10^{111111}=2^{63}-2^{28}=9223\,37203\,65863\,40352.$$

Sealjuures on arvutuste maksimaalne täpsus vähemalt sama suur kui kolmekümne kuue kohaliste pesadega fikseeritud komaga masinas. Tegelikult on siin täpsus isegi üldiselt suurem, sest kui arvud on kujutatud normaalkujul, siis sisaldavad nad alati täpselt 35 tüvenumbrit. Seega on suurema täpsuse huvides oluline, et kõik arvud oleksid kujutatud normaalkujul. Seda tüüpi masinates on küll alati võimalik mõnd tehet kasutada arvude teisendamiseks normaalkujule (selline teisendamine seisneb lihtsalt arvu mantissi nihutamises vasakule seni, kuni kohal nr. 1 on number 1; samal ajal vähendatakse järku nihutatud kohtade arvu võrra), kuid ilmselt arvude täpsus niisuguse n o r m a l i s e e r i m i s e g a ei suurene, sest viimased kohad täituvad nullidega. Näiteks kui enne normaliseerimist oli arvutusmasina «Strela» pesas salvestamisel arv

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	0	0	1	1	1	1	0	1	1	1	1	0	0	0	0	1

siis pärast normaliseerimist omandab ta kuju

3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
0	0	1	1	1	0	0	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0

Liikuva komaga masinates on kogu aritmeetiline seade ehitatud enamasti nii, et aritmeetiliste tehete tulemused saadakse alati normaalkujul. Seega võib normaliseerimise tehteks kasutada näiteks liitmist nulliga.

Kümnendarvude kujutamine masinas. Siiani me vaatlesime ainult kahendarvude kujutamist masina mälu pesas. Praktikas tuleb aga lähteandmed masinasse viia (ja arvutustulemused sealt hiljem välja tuua) kümnendsüsteemis, sest arvusüsteemide käsitsi teisendamine oleks liiga aeganõudev töö. Kümnendarvude kujutamiseks masina mälu pesas kasutatakse nn. kahend-kümnendsüsteemi, milles kümnendarvu iga number kujutatakse neljakohalise kahendarvuna vastavalt järgmisele tabelile[1]:

0 = 0000	5 = 0101
1 = 0001	6 = 0110
2 = 0010	7 = 0111
3 = 0011	8 = 1000
4 = 0100	9 = 1001.

Seega erineb kahend-kümnendsüsteem kümnendsüsteemist vaid selle poolest, et iga number on kujutatud mitte ühe-, vaid neljakohalisena. Näiteks kümnendarv 739,56 omab kahend-kümnendsüsteemis kuju 0111 0011 1001,0101 0110. Ka tagasiteisendamine kahend-kümnendsüsteemist kümnendsüsteemi on lihtne. Selleks tuleb kahend-kümnendsüsteemis kirjutatud arv jaotada (komast alates mõlemale poole) neljanumbrilisteks rühmadeks ning asendada iga saadud neljakohaline kahendarv vastava kümnendsüsteemi numbrimärgiga. Näiteks

0110 1001 0110 1000 0100 0000 0001,0111 0101 = 6968401,75.

Vahetult pärast märgikohta fikseeritud komaga masi-

[1] Võib kasutada ka teistsuguseid tabeleid kümnendnumbrite teisendamiseks «kahendarvudeks», kuid teiste võimaluste selgitamisel me siin ei peatu. Lähemalt vt. näit. raamatust: Р. К. Ричардс. Арифметические операции на цифровых вычислительных машинах. Москва, 1957.

nate puhul peavad sisseviidud kümnendarvud samuti omama lihtmurdude kuju. Nii toimub arvutusmasina «Ural» pesas kümnendarvude salvestamine selliselt, et esimene koht (koht nr. 0) on määratud arvu märgi kujutamiseks, kuna aga järgmised jaotatakse neljakohalisteks rühmadeks kümnendmurru numbrite tarvis. Viimasesse (üheksandasse) rühma jääb ainult kolm kohta, mistõttu viimane kümnendnumber tuleb ümardada. Näiteks kümnendmurru −0,579436517 kujutamisel on pesa täidetud 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	24	25	26	27	28	29	30	31	32	33	34	35
1	0	1	0	1	0	1	1	1	1	0	0	1	0	1	0	0	0	0	1	1	0	1	1	0	0	1	0	1	0	0	0	1	1	0	0
—	5				7				9				4				3				6				5				1				8		

Siit on muide näha, mispärast kolmekümne kuue kohaliste pesade kasutamine annab kümnendsüsteemis mitte kümme, vaid ülimalt üheksa õiget tüvenumbrit. Põhjuseks on kohtade näiliselt ebaratsionaalne kasutamine kahendkümnendsüsteemi kujutamisel. Kümme õiget tüvenumbrit võiks saada sel juhul, kui kõik arvud viia masinasse kahendsüsteemis ja kahendsüsteemis saadud arvutustulemused teisendada hiljem käsitsi kümnendsüsteemi. Seda moodust (eriti tulemuste teisendamisel) pole aga mõtet kasutada, sest arvutuste teostamisel paratamatult toimuvate ümardamistega tekkivad vead muudavad vähemalt paar viimast kümnendkohta ikkagi ebaõigeteks.

Liikuva komaga masinas kujutamiseks tuleb ka kümnendarvud kõigepealt teisendada poollogaritmilisele kujule. Näiteks

$$14372985600 = 0{,}143729856 \cdot 10^{11}.$$

Kümnendarvude salvestamisel jaotuvad arvutusmasina «Strela» pesas kohad järgmiselt: koht nr. 0 on määratud arvu märgi kujutamiseks, kohtadel nr. 1 kuni 36 kujutatakse mantissi üheksa numbrikohta, kohal nr. 37 järgu märk ja viimasel viiel kohal järk (nendel viiel kohal saab kahend-kümnendsüsteemis kujutada järke, mille absoluutväärtus pole suurem kui 19, sest 19 kujutub kujul 11001). Viimatitoodud arv salvestub arvutusmasina «Strela» pesas seega järgmiselt:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
0	0	0	1	0	1	0	0	0	0	1	1	0	1	1	1	0	0	1	0	1	0	0	1	1	0	0	0	0	1	0	1	0	1	1	0	0	1	0	0	0	1
1				4				3				7				2				9				8				5				6				+1		1			

Käskude kujutamine mälu pesas. Kõiki tehteid sooritab arvutusmasin spetsiaalsete signaalide — käskude toimel, mis koos algandmetega tuleb salvestada masina mälusse. Iga käsk kujutab endast arvude kompleksi, milledest üks on alati tehte kood, s. t. sooritada tuleva tehte leppeline number. Teised arvud kannavad käsu aadresside nime ning näitavad, kust tuleb võtta tehte sooritamiseks vajalikud arvud ja kuhu salvestada tehte tulemus. Käsu aadressideks on seega masina mälu pesade aadressid, s. t. pesade järjekorranumbrid.

Olgu masina (sisemises) mälus 2048 pesa, mis on nummerdatud kaheteistkohaliste kahendarvudega

$$000\,000\,000\,000 = 0$$
$$000\,000\,000\,001 = 1$$
$$\dots\dots\dots\dots$$
$$011\,111\,111\,111 = 2047$$

(nii on see näiteks nõukogude arvutusmasinates «Strela», M-3 ja «Ural»). Kuna ülesande programmi (s. t. selle ülesande lahendamiseks vajalike käskude jada) koostamisel tuleb käsud käsitsi üles kirjutada, siis on kahendsüsteemi kasutamine pesade aadresside kirjutamisel arvude pikkuse tõttu ilmselt ebamugav. Seepärast kasutatakse käskude üleskirjutamisel enamasti kaheksandsüsteemi [1]. Pesade aadressid tuleb siis kirjutada neljakohaliste kaheksandnumbritena: 0000, 0001, . . . , 3777; nende teisendamine kahendsüsteemi toimub käskude masinasse sisseviimisel automaatselt. Kaheksandsüsteemi kasutatakse ka tehte koodi üleskirjutamisel, mis enamikus masinates on kahekohaline (kahendsüsteemis seega 6-kohaline).

Kuna igas aritmeetilises tehtes on enamasti tegemist kolme arvuga (näiteks kaks liidetavat ja summa), siis

[1] Mõnede masinate (näiteks nõukogude elektron-arvutusmasinate M-2 ja БЭСМ) puhul kasutatakse sel otstarbel ka kuueteistkümnendsüsteemi; samuti leiab mõnedes masinates (M-2) kasutamist veel neljandsüsteem.

peab käsk oma kõige loomulikumal kujul sisaldama kolm aadressi. Käsitsi kirjutatult koosneb käsk seega neljast kaheksandarvust: kolmest aadressist ja tehte koodist. Mõnikord (näiteks arvutusmasinas «Strela») lisandub neile veel ühekohaline nn. k o n t r o l l m ä r k, mis võib võrduda ühega või nulliga. Käsu sisule kontrollmärk mõju ei avalda, kuid ta on vajalik just programmide kontrollimisel. Selleks lülitatakse masin erilisele kontrollmärkide järgi töötamise režiimile, mis seisneb selles, et masin peatub iga kord, kui ta on lõpetanud niisuguse käsu täitmise, mille kontrollmärk võrdus ühega.

Käske salvestatakse mälu pesades põhimõtteliselt samuti nagu tavalisi arve. Nii on arvutusmasina «Strela» pesa kohad käsu salvestamisel jaotatud selliselt, et kohtadel nr. 0—11, 12—23 ja 24—35 paiknevad käsu aadressid (nimetame neid edaspidi vastavalt esimeseks, teiseks ja kolmandaks aadressiks), kohal nr. 36 kontrollmärk ning kohtadel nr. 37—42 tehte kood. Näiteks kui pesa kohad on täidetud järgmise skeemi kohaselt

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
0	0	1	0	0	0	1	1	1	0	0	1	0	0	1	0	0	1	0	1	1	0	1	0	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	1

1. aadress 2. aadress 3. aadress Kontroll-märk Tehte kood

siis salvestatakse seal käsk

1071 1132 1250 0 01.

See käsk tähendab: «võtta arv pesast nr. 1071 ja teine arv pesast nr. 1132, liita need arvud (tehteks nr. 01 on arvutusmasina «Strela» puhul liitmine) ja salvestada summa pesas nr. 1250. Olgu muide märgitud, et tulemuse viimisel pesasse nr. 1250 pole oluline, mis arv selles pesas varem salvestamisel oli. Elektron-arvutusmasinad on nimelt enamasti ehitatud nii, et arvu viimisel mingisse pesasse kustub automaatselt selles pesas varem olnud arv.

Masinate liigitamine vastavalt käskude struktuurile. Elektron-arvutusmasinad jagunevad käskude täitmise järjekorra poolest kahte liiki.

1) Masinad, milledes käskude täitmine toimub n. ö. l o o m u l i k u s j ä r j e k o r r a s. Niisuguses masinas kan-

takse käsud järjestikku nummerdatud pesadesse. Pärast ühe käsu täitmist asub masin täitma käsku, mis asetseb ühe võrra suurema aadressiga pesas. Nii toimub seni, kuni esineb käsk, mis annab juhtimise üle mitte järgmises, vaid mingis teises pesas asetsevale käsule (selliseid nn. suunamiskäske vaatleme lähemalt neljandas peatükis — § 10—11).

2) Masinad, milledes käskude täitmine toimub sundjärjekorras. Niisuguse masina puhul on igas käsus peale tavaliste andmete antud veel selle pesa aadress, kuhu on salvestatud järgmisena täitmisele tulev käsk. Sellist võimalust kasutatakse peamiselt masinais, mille sisemiseks mäluks on pöörlev magnettrummel (vt. III pt.). Trumli ühele pöördele kuluv aeg on suhteliselt suur ning, täites käske loomulikus järjekorras, peaks masin pärast tehte sooritamist ootama, kuni eelmist käsku sisaldanud pesale järgnev pesa jõuab uuesti lugemisseadise alla. Tehete sundjärjekorras teostamine võimaldab käske mälus paigutada nii, et järgmist käsku sisaldav pesa jõuab lugemisseadise alla just selleks ajaks, millal lõpeb eelmise tehte sooritamine.

Aadresside arvu ning nende otstarbe järgi liigitatakse käske ja ühtlasi ka vastava ehitusega arvutusmasinaid järgmiselt.

1. Käske, mis omavad eelmises punktis kirjeldatud kuju, nimetatakse kolmeaadressilisteks (samuti nimetame ka vastavat arvutusmasinat). Kolmeaadressiline masin täidab käske enamasti loomulikus järjekorras. Nende masinate eeliseks on programmeerimise lihtsus, sest kolmeaadressilised käsud vastavad kõige enam aritmeetiliste tehete olemusele. Kuid see pole mitte ainus võimalik käskude kuju. Järgmisena vaadeldaval juhul näiteks on eeliseks konstruktsiooni lihtsus ja seega masina odavus.

2. Üheaadressiline masin täidab käske loomulikus järjekorras. Lisaks tavalisele sisemisele mälule on sellise masina aritmeetilises seadmes veel eriline ühepesaline mälu, mida nimetatakse summaatoriks (mõnedes masinates sisaldab aritmeetiline seade ka rohkem kui ühepesalise mälu; teisi pesi nimetatakse sel juhul registriteks). Üheaadressiline käsk omab näiteks kuju

02 1071,

mis tähendab[1]: «võtta arv pesast nr. 1071 ja kanda ta summaatorisse (kui summaatoris oli enne arv, siis kustutada see)».

Selleks et teostada tehe, mille kolmeaadressiline masin sooritab ülal näiteks toodud käsu järgi, läheb üheaadressilises masinas tarvis veel kaht käsku:

01 1132

ja

16 1250,

mis tähendavad vastavalt «võtta arv pesast nr. 1132 ja liita ta summaatoris olevale arvule; tulemus salvestada summaatoris seal varem olnud arvu asemel» ja «kanda summaatoris olev arv pesasse nr. 1250».

Toodud näiteist ei tule aga järeldada, nagu oleks üheaadressilises masinas tehte sooritamiseks alati tarvis kolm korda rohkem käske kui kolmeaadressilises masinas. Näiteks pesades nr. 1071, 1132 ja 1173 olevate arvude summa leidmise pesasse nr. 1250 teostab kolmeaadressiline masin kahe käsu

1071 1132 1250 01
1173 1250 1250 01

toimel, üheaadressilises masinas aga kulub selleks vaid neli käsku:

02 1071
01 1132
01 1173
16 1250.

3. K a h e a a d r e s s i l i s i masinaid võib olla kahte süsteemi. Sundjärjekorras käske täitev masin töötab põhimõtteliselt samuti nagu üheaadressiline masin. Näiteks käsk

02 1071 0034

tähendab: «viia pesas nr. 1071 olev arv summaatorisse ja

---

[1] Üheaadressiline arvutusmasin on näiteks «Ural». Toodavates näidetes ongi kasutatud selle masina tehete koode ja käsu osade kirjutamise järjekorda. Käskude salvestamiseks kasutatakse elektronarvutusmasinas «Ural» lühikesi pesasid, mis parajasti mahutavad 18 kahendnumbrit ehk kuus kaheksandnumbrit.

asuda seejärel pesas nr. 0034 oleva käsu täitmisele». Pesas nr. 0034 on säilitamisel näiteks käsk

01 1132 0103,

s. t. «liita summaatoris olevale arvule pesas nr. 1132 olev arv ning täita seejärel pesas nr. 0103 olev käsk» jne.

Loomulikus järjekorras käske täitva kaheaadressilise masina käsk omab näiteks kuju [1]

00 1071 1132,

mis tähendab: «pesas nr. 1071 olev arv liita pesas nr. 1132 oleva arvuga ja salvestada tulemus pesas nr. 1132». Selle käsu täitmisel läheb kaotsi pesas nr. 1132 varem olnud arv, s. t. teine liidetav. Tulemuse ülekandmiseks pesasse nr. 1250 on vajalik eraldi käsk

05 1132 1250

(«pesas nr. 1132 olev arv viia üle pesasse nr. 1250»).

4. Viimati vaadeldud kaheaadressiliste masinate erikujuks on nn. p o o l t e i s e a a d r e s s i l i s e d masinad. Niisuguse masina käsud on tegelikult kaheaadressilised, kuid teine aadress kirjutatakse mitte nelja-, vaid kahekohalise kaheksandarvuga. Selline lihtsustus on võimalik seetõttu, et masina konstruktsioonis on ette nähtud vaid väike arv (mitte rohkem kui kuuskümmend neli) pesasid, millede aadresse võib kasutada käsu teise aadressina. Olgu need näiteks pesad nr. 3700—3777. Selleks et liita pesades nr. 1071 ja 1132 olevad arvud ning salvestada tulemus pesas nr. 1250, on tarvis järgmised kolm käsku:

24 1071 32
01 1132 32
26 1250 32,

mis tähendavad vastavalt: «pesas nr. 1071 olev arv viia pesasse nr. 3732», «pesas nr. 1132 olev arv liita pesas nr. 3732 oleva arvuga ning salvestada tulemus pesas nr. 3732» ja «pesas nr. 3732 olev arv viia pesasse nr. 1250».

---

[1] Seda tüüpi masin on näiteks M-3. Siin ongi toodud just selle masina tehte koodid.

(Esimese ja kolmanda tehte koodid on erinevad sellepärast, et ühel juhul viiakse arv esimesena kirjutatud pesast teisena kirjutatusse, teisel juhul aga vastupidi.)

5. Nelja-aadressilised masinad on tegelikult samalaadsed kolmeaadressilistega, ainult käskude täitmine toimub sundjärjekorras. Neljandaks aadressiks ongi neil järgmisena täitmisele tulevat käsku säilitava pesa aadress. Näiteks käsk

01 1071 1132 1250 0050

tähendab «liita pesades nr. 1071 ja 1132 olevad arvud, saata tulemus pesasse nr. 1250 ning asuda siis pesas nr. 0050 oleva käsu täitmisele».

6. On ehitatud ka viieaadressiline arvutusmasin (Tšehhoslovakkias), mille käsk

01 1071 1132 1250 0521 0413

tähendab: «liita pesades nr. 1071 ja 1132 olevad arvud ning saata tulemus pesasse nr. 1250; kui tulemus on positiivne, siis asuda pesas nr. 0521, vastasel juhul aga pesas nr. 0413 oleva käsu täitmisele».

Järgnevas käsitluses me piirdume peamiselt ainult ühe- ja kaheaadressiliste masinate vaatlemisega, kuigi praktikas leiavad laialdast kasutamist ka kolmeaadressilised masinad (näit. «Strela», M-2, БЭСМ). Mis aga puutub arvude kujutamiseks kasutatavasse moodusesse, siis selles osas tulevad vaatlemisele eeskätt fikseeritud komaga masinad. Niisuguse valiku põhjuseks on asjaolu, et Eesti NSV-s loodud arvutuskeskused on varustatud just elektron-arvutusmasinatega «Ural» ja M-3. Nende masinate baasil on seetõttu toodud ka enamik järgnevatest konkreetsetest näidetest (§ 10—11).

5. MATEMAATILISE LOOGIKA PÕHIMÕISTED

Loogika on teadus mõtlemise vormidest ja seadustest; matemaatiline loogika on loogika niisugune haru, mis on arendatud matemaatika vajaduste rahuldamiseks. Matemaatilise loogika esimeseks vajalikuks osaks on lausearvutus.

Lauseks nimetatakse iga väidet, mille sisu kohta saab ütelda, et ta on tõene või väär (näiteks laused «5 on paaritu arv» ja «matemaatika on teadus» on tõesed, aga laused «lumi on must» ja «8 on algarv» väärad). Sealjuures eeldatakse, et iga lause saab olla ainult kas tõene või väär (mitte mõlemad korraga ega ei kumbagi). Lauseid tähistame suurte tähtedega A, B, C, ...

Kaht lauset loetakse erinevateks (ning tähistatakse erinevate tähtedega), kui neil on erinev sisu. Lausete sisu me arvestame aga üksnes nende tähistamisel. Edaspidi meid huvitab ainult see, kas lause on tõene või väär. Kui lause on tõene, siis loeme tema t õ e s u s e võrdseks ühega, kui ta on väär, siis võrdseks nulliga.

Lause tõesust tähistame enamasti sama tähega, millega lauset ennast. Lause tõesus saab seega omada vaid ühe kahest väärtusest: 0 või 1.

Lauseid A ja B nimetatakse e k v i v a l e n t s e t e k s ning kirjutatakse $A=B$, kui nende tõesused on võrdsed, s. t. kui A ja B on ühekorraga kas tõesed või väärad. Kirjutis $A=1$ tähendab, et lause A on tõene, ning kirjutis $B=0$, et lause B on väär.

Siiani me vaatlesime tegelikult ainult lauseid, mille tõesus omab ühe kindla (lause sisuga äramääratud) väärtuse. Loogika rakendustes on aga tegemist peamiselt lausetega, mille tõesus muutub mitmesuguste välistingimuste mõjul. Niisuguste muutuvate lausete hulka kuulub muuhulgas ka m i s t a h e s l a u s e, s. t. lause, millel pole konkreetset sisu, kuid on tõesus ja mis võib seega endast kujutada kas üldse iga võimalikku (lausearvutuse) lauset või lauset mingist teatud lausete hulgast. Näiteks võib niisugune hulk koosneda lausetest, mis väidavad midagi ilmastiku kohta.

Mistahes lause mõistega on tegemist muuhulgas iga omaduse puhul, mis lausearvutuses (lausete kohta üldse) tõestatakse. Seepärast järgnevas me enamasti mõistamegi sümbolite A, B, ... all muutuva tõesusega lauseid, kusjuures tõesuse muutumine võib olla tingitud väga mitmesugustest põhjustest. Kirjutis $A=1$ tähendab tavaliselt seda, et lause A on a n t u d o l u k o r r a s või a n t u d v a l i k u (konkretiseerimise) p u h u l tõene. Näiteks lausete A, B, ... vaatlemise korral on meil järgnevas tegelikult oluline ainult see, et nende lausete tõesused võivad omandada mistahes väärtuste kombinatsiooni. Mistahes lausete konkretiseerimine tähendab seega vaid nende tõesuste konkretiseerimist.

Loogilised tehted. Ühest või mitmest lausest saab nende ühendamise teel moodustada l i i t l a u s e i d. Lausete ühendamisel arvestatakse sealjuures tavaliselt ainult komponentide tõesust, mitte aga nende konkreetset sisu. Seetõttu võime lausete mitmesuguste ühendamiste defineerimisel piirduda sellega, et anname eeskirja liitlause tõesuse leidmiseks komponentide tõesuse väärtuste kõigi võimalike kombinatsioonide jaoks. Sellist eeskirja nimetatakse liitlause moodustamisele vastavaks l o o g i l i s e k s t e h t e k s. Vaatleme järgnevalt tähtsamaid liitlause moodustamise viise ja neile vastavaid loogilisi tehteid.

1. Lause A e i t u s e k s nimetatakse lauset, mis on tõene siis, kui A on väär, ja väär siis, kui A on tõene. Lause A eitamist tähistatakse $\overline{A}$ ning loetakse «mitte A». Näiteks kui A tähendab lauset: «lumi on roheline», siis $\overline{A}$ on lause: «lumi pole mitte roheline». Eitamise (inverteerimise) loogilise tehte võib defineerida tabeliga

$$\boxed{\begin{array}{c} \overline{0}=1 \\ \overline{1}=0 \end{array}}\,.$$

2. Lausete A ja B k o r r u t i s e k s ehk k o n j u n k t s i o o n i k s nimetatakse liitlauset, mis on tõene siis ja ainult siis, kui laused A ja B on tõesed (s. t. ülejäänud juhtudel loetakse liitlause vääraks). Lausete korrutamist tähistatakse[1] sümboliga $A \wedge B$, mida loeme «A ja B». Näiteks kui laused A ja B konkretiseerida järgmiselt:

A = «lumi on valge»
B = «8 on algarv»,

siis nende konjunktsioon tähendab lauset

$A \wedge B$ = «lumi on valge ja 8 on algarv»

(mis ilmselt on väär, sest üks komponentlausetest on väär).

Vastav loogilise korrutamise tehe on määratud tabeliga

[1] Samaks otstarbeks kasutavad paljud autorid ka tähistust A & B.

$$\boxed{\begin{aligned} 0 \wedge 0 &= 0 \\ 0 \wedge 1 &= 0 \\ 1 \wedge 0 &= 0 \\ 1 \wedge 1 &= 1 \end{aligned}}$$

3. Lausete A ja B summaks ehk disjunktsiooniks nimetatakse liitlauset, mis on väär siis ja ainult siis, kui laused A ja B on väärad (ülejäänud juhtudel loetakse summa tõeseks). Lausete summat tähistatakse $A \vee B$ ning loetakse «(kas) A või B või mõlemad». Kui A ja B konkretiseerida samuti nagu eelmises näites, siis

$A \vee B =$ «kas lumi on valge, või 8 on algarv, või kehtivad mõlemad väited»

(antud juhul liitlause on tõene, sest üks komponentlausetest on tõene).

Loogiline liitmine on seega defineeritud tabeliga

$$\boxed{\begin{aligned} 0 \vee 0 &= 0 \\ 0 \vee 1 &= 1 \\ 1 \vee 0 &= 1 \\ 1 \vee 1 &= 1 \end{aligned}}$$

Kõiki võimalikke tõesuse väärtuse kombinatsioone läbi proovides võib lugeja kontrollida, et äsjadefineeritud loogiliste tehete puhul on kehtivad järgmised seosed:

$$A \vee B = B \vee A; \quad A \wedge B = B \wedge A;$$
$$A \vee B = \overline{\bar{A} \wedge \bar{B}}; \quad A \wedge B = \overline{\bar{A} \vee \bar{B}};$$
$$A \vee \bar{A} = 1; \quad 1 \wedge A = A; \quad 0 \vee A = A;$$
$$A \vee A = A \wedge A = A;$$
$$(A \wedge B) \vee (A \wedge C) = A \wedge (B \vee C);$$
$$(A \vee B) \wedge (A \vee C) = A \vee (B \wedge C).$$

Need seosed leiavad laialdast kasutamist peamiselt mitmesuguste loogilisi tehteid sisaldavate valemite lihtsustamisel. Näiteks neid seoseid kasutades saame:

$$(A\wedge B\wedge C)\vee(A\wedge B\wedge \bar{C})\vee(\bar{A}\wedge B\wedge C)=$$
$$=[(A\wedge B\wedge C)\vee(A\wedge B\wedge \bar{C})]\vee[(A\wedge B\wedge C)\vee(\bar{A}\wedge B\wedge C)]=$$
$$=[(A\wedge B)\wedge(C\wedge\bar{C})]\vee[(B\wedge C)\wedge(A\vee\bar{A})]=$$
$$=(B\wedge A)\vee(B\wedge C)=B\wedge(A\vee C).$$

Peale toodud kolme loogilise tehte on arvutusmasinate puhul tarvis veel kaht loogilist tehet (loogikas on vastavate tehete arv suurem).

4. Lausete A ja B samaväärsuseks nimetatakse liitlauset, mis on tõene siis ja ainult siis, kui laused A ja B on mõlemad kas tõesed või väärad. Lausete samaväärsust tähistatakse $A\sim B$ ja loetakse «A on samaväärne B-ga». Samaväärsuse loogilise tehte määrab tabel

$$\boxed{\begin{array}{c} 0\sim 0=1\\ 0\sim 1=0\\ 1\sim 0=0\\ 1\sim 1=1 \end{array}}\;.$$

5. Sageli on otstarbekas lugeda eraldi loogiliseks tehteks ka samaväärsuse eitamist $\overline{A\sim B}$, mida tähistatakse lühidalt $A\overline{\sim}B$ ja nimetatakse mittesamaväärsuse tehteks. See tehe on määratud tabeliga

$$\boxed{\begin{array}{c} 0\overline{\sim}0=0\\ 0\overline{\sim}1=1\\ 1\overline{\sim}0=1\\ 1\overline{\sim}1=0 \end{array}}\;.$$

Loogiliste tehete seos kahendaritmeetikaga. Vaadeldud tehete rakendamist vahetult loogikas me siin ei käsitle. Edaspidises pole isegi oluline, et need tehted on defineeritud lausetega (või nende tõesustega). Tähtis on vaid, et tehteid teostatakse suurustega, mis saavad omandada üksnes väärtusi null ja üks (niisuguseid suurusi nimetatakse kahendmuutujaiks). Sellise omadusega on ilmselt aga ka kahendsüsteemi numbrimärgid (s. t. ühekohalised kahendarvud) ning nendega me järgnevas peamiselt loogilisi tehteid sooritamegi.

Loogilisi tehteid on arvutusmasinate puhul tarvis nii kahendarvude aritmeetika realiseerimiseks kui ka vahetult mõningate arvutuste läbiviimisel. Osutub nimelt, et kui elektrilisi signaale vaadelda samuti kahendmuutujatena (s. t. kui need signaalid saavad omandada vaid kaks väärtust), siis on loogiliste tehete sooritamine nendega tehniliselt kõige lihtsam (vt. § 6). Teiselt poolt saab aga tõestada, et üldse kõik võimalikud tehted kahendmuutujatega on alati avaldatavad loogiliste tehete kaudu. Sealjuures piisab koguni ainult kolmest loogilisest tehtest: loogilisest eitamisest, loogilisest liitmisest ja loogilisest korrutamisest. Näiteks neljas ülalvaadeldud tehe — loogiline samaväärsus on avaldatav nende kaudu valemitega

$$A \sim B = (A \vee \bar{B}) \wedge (\bar{A} \vee B) = (\bar{A} \wedge \bar{B}) \vee (B \wedge A)$$

ja loogiline mittesamaväärsus valemitega

$$A \nsim B = (A \wedge \bar{B}) \vee (\bar{A} \wedge B) = (\bar{A} \vee \bar{B}) \wedge (A \vee B)$$

(nende valemite kehtivuses veendumiseks tuleb lihtsalt läbi proovida kahendmuutujate A ja B väärtuste kõik neli võimalikku kombinatsiooni).

Järgnevas on meil tarvis loogiliste tehete kaudu avaldada eeskätt aritmeetilisi tehteid. Et aga kõik aritmeetilised tehted on taandatavad liitmisele, siis piisab, kui me vaatleme kahendarvude liitmist loogiliste tehete abil (esitatav käsitlusviis on tegelikult rakendatav iga operatsiooni taandamisel loogilistele tehetele).

Kahe kahendarvu liitmise puhul tuleb igal numbrikohal teatavasti liita antud arvude vastavad numbrikohad ja eelmiselt numbrikohalt tulnud ülekanne. Näiteks kahendarvude 101 ja 111 liitmine toimub järgmiselt:

1. liidetav	0101
2. liidetav	0111
ülekanded	1110
summa	1100.

Seega igal numbrikohal tuleb liita kolm ühekohalist kahendarvu, kusjuures summa saab olla ülimalt kahekohaline. Tähistame liidetavate vastavad numbrikohad tähtedega A ja B, eelmiselt kohalt tulnud ülekande $\ddot{U}_2$, summa vastava numbrikoha S ning järgmisele kohale mineva ülekande $\ddot{U}_1$. Siis vaadeldaval numbrikohal toimuv liitmine on määratud tabeliga

A	B	$\ddot{U}_2$	S	$\ddot{U}_1$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Leiame kõigepealt ülekande $\ddot{U}_1$ avaldise. Tabelist näeme, et $\ddot{U}_1=1$ ainult neljal juhul: kui $A=0$, $B=1$, $\ddot{U}_2=1$ või $A=1$, $B=0$, $\ddot{U}_2=1$, või $A=1$, $B=1$, $\ddot{U}_2=0$, või $A=1$, $B=1$, $\ddot{U}_2=1$ (s. t. kui tegemist on tabeli neljanda, kuuenda, seitsmenda või kaheksanda reaga). Seda asjaolu võib üles kirjutada kujul

$$\ddot{U}_1=(\bar{A}\wedge B\wedge \ddot{U}_2)\vee(A\wedge\bar{B}\wedge\ddot{U}_2)\vee(A\wedge B\wedge\overline{\ddot{U}_2})\vee(A\wedge B\wedge\ddot{U}_2).$$

Tõepoolest, selle valemi parema poole tõesus on üks ainult siis, kui (vähemalt) ühe sulgavaldise tõesus on üks. Kuid näiteks esimese sulgavaldise tõesus on üks vaid juhul, kui tegemist on tabeli neljanda reaga (vt. loogilise korrutamise definitsioon). Analoogiliselt teine sulgavaldis vastab tabeli kuuendale reale, kolmas sulgavaldis seitsmendale reale ja neljas sulgavaldis kaheksandale reale.

Saadud valemit eelmises punktis nimetatud viisil lihtsustades omandab see kuju

$$\ddot{U}_1=(A\wedge B)\vee(B\wedge\ddot{U}_2)\vee(A\wedge\ddot{U}_2)=[(A\wedge B)\vee(A\vee B)\wedge\ddot{U}_2].$$

Summa S jaoks saame analoogilisel viisil valemi

$$S=(\bar{A}\wedge\bar{B}\wedge \ddot{U}_2)\vee(\bar{A}\wedge B\wedge\overline{\ddot{U}}_2)\vee(A\wedge\bar{B}\wedge\overline{\ddot{U}}_2)\vee(A\wedge B\wedge \ddot{U}_2).$$

Ainult kolme loogilist tehet kasutades sellele valemile oluliselt lihtsamat kuju anda ei õnnestu. Tuues aga juurde ka loogilise samaväärsuse või mittesamaväärsuse tehte, saame vaadeldava valemi kõige lihtsamateks kujudeks

$$S=\ddot{U}_2\sim(A\sim B)$$

või

$$S=\ddot{U}_2\not\approx(A\not\approx B).$$

Praktilistes rakendustes (vt. § 9) osutub otstarbekaks teisendada see valem kujule

$$S=[(A\vee B)\vee\ddot{U}_2]\wedge[\overline{\ddot{U}}_1\vee(A\wedge B\wedge\ddot{U}_2)],$$

kus kasutatakse ära juba leitud ülekanne $\ddot{U}_1$ ja milles ei esine lähteandmete (liidetavate ning ülekande $\ddot{U}_2$) eitusi.

ELEKTRON-ARVUTUSMASINATES KASUTATAVAD LÜLITUSELEMENDID JA SEADMED

6. LOOGILISI TEHTEID REALISEERIVAD LÜLITUSELEMENDID

Lülitusskeemide põhitüübid. Enne elektron-arvutusmasina kui terviku töötamise põhimõtete selgitamise juurde asumist tuleb lühidalt tutvuda tema üksikute koostisosade ehitusega. Loomulikult sõltub nende koostisosade ehitus ja osalt ka valik oluliselt konkreetsest masinatüübist ning on seega väga mitmekesine. Laialivalgumise vältimiseks on järgnevas orienteeritud enamkasutatavate skeemide järgi, jättes kõrvale vananenud või harvarakendatavad lahendused.

Nagu me eelmises paragrahvis nägime, on põhilised aritmeetilised operatsioonid taandatavad teatud loogiliste tehete sooritamisele. Seda arvestades jaotame ka vastavaid tehteid realiseerivad elektrilised skeemid n. ö. elementaarskeemideks, milledest sobivate lülituste teel saab juba konstrueerida vajalikke keerulisemaid skeeme.

Osutub, et elektron-arvutusmasina aritmeetilise seadme ja juhtimisseadme kõik põhilised lülitused on taandatavad järgmisele neljale põhitüübile:

1) loogilist liitmist teostav lülitus (loogiline liitja ehk «või»-lüli);

2) loogilist korrutamist teostav lülitus (loogiline korrutaja ehk «ja»-lüli);

3) loogilist eitamist teostav lülitus (nn. invertor ehk «ei»-lüli);

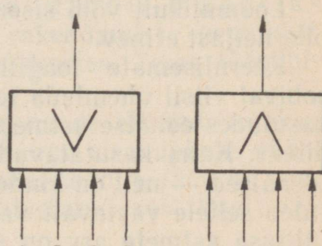
4) ühte kahendnumbrit (0 või 1) salvestav mäluelement (kuna selliseks lülituseks oli alguses nn. triger-lülitus, siis nimetatakse kõiki seda funktsiooni täitvaid lülitusi elektron-arvutusmasinates tavaliselt t r i g e r i t e k s).

Muud lülituselemendid ja -skeemid, nagu näiteks võimendajad, katoodjärgijad, impulsse formeerivad ahelad, impulsstrafod, viiteahelad jt., on elektron-arvutusmasinates enamasti vajalikud vaid masina normaalse töö tagamiseks. Nad ei teosta ühtki loogilist tehet ning nende vajaduse tingivad konstruktsioonilised iseärasused ja mõningad tehnilised põhjused. Näiteks juhtmetevahelise mahtuvuse mõjul impulsid «nürinevad», muutuvad lamedaks ning nende esialgne kuju kaob. Vajaliku kuju taastamiseks tulebki kasutada erilisi vahelülitusi. Mõnikord on vajalik, et samast allikast lähtuv impulss jõuaks masina ühte sõlme märksa hiljem kui mingisse teise. Sel juhul tuleb impulsside allika ja vaadeldava sõlme vahele lülitada nn. viiteahel. Kui mõne skeemi väljundisse on lülitatud palju energiat tarbiv koormus (mistõttu impulsi kõrgus väheneb ja front muutub lamedaks), siis lülitatakse vahele veel katoodjärgija, mis sobitab allika sisetakistuse koormustakistusega. Niisuguseid eriolukordi, mis nõuavad täiendavate lülituste rakendamist, on palju ning nende puhul kasutatavad võtted ei muuda arvutusmasina funktsionaalset ehitust. Seetõttu me selliste abilülituste lähemal selgitamisel enamasti ei peatugi.

Kõikide järgnevate skeemide kirjeldamisel tähistame nende sisendeid tähtedega A, B, C, D, ... ja väljundit tähega F. Samad tähed tähistavad ka vastavaid sisend- ja väljundpingeid. Sealjuures lepime kokku, et kõik pinged saavad omandada vaid kaks võimalikku väärtust: kõrge ja madala. Asjaolu, et pinge näiteks sisendil A (pinge A) on kõrge, kirjutame lühidalt kujul $A=1$. Analoogiliselt, kui pinge B on madal, siis kirjutame $B=0$.

Dioodlülitused leiavad kasutamist peamiselt loogilise liitmise ja loogilise korrutamise teostamiseks. Vastavaid lülitusskeeme tähistatakse lihtsustatult joonisel 2 näidatud kujul (konkreetsuse andmiseks on nendel joonistel ja ka järgnevas käsitluses sisendite arvuks valitud neli).

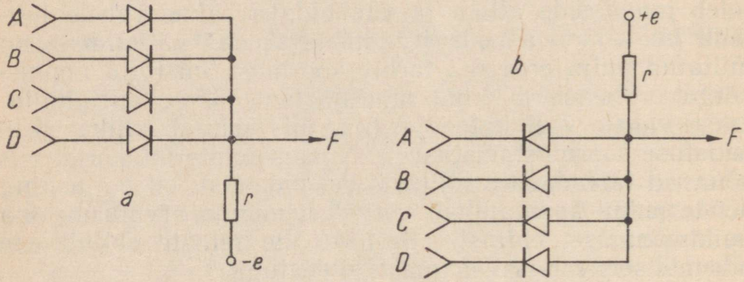
Teatavasti $F=A \vee B \vee C \vee D$ korral on $F=0$ ainult sel juhul, kui $A=B=C=D=0$ (kõik sisendpinged on madalad). Seda tehet realiseeriv dioodlülitus on esitatud joonisel 3, *a*.



Joonis 2

Tõepoolest, kui kõigis sisendites A, B, C ja D on pinge «null» (s. t. madal), siis väljundpinge F on $-e$, sest läbi dioodide tõkkesuunas voolu ei kulge ning pingelangu takistil r ei ole. Kui aga vähemalt ühel sisendjuhtmel on positiivne potentsiaal, siis on see ka väljundjuhtmel F, sest vastav diood on sel juhul avatud (juhib).

Pingete A, B, C ja D loogiline korrutamine tähendab neile vastavate «ühtede» ja «nullide» loogilist korrutamist. Väljundpinge F on «üks» vaid siis, kui kõik sisendpinged $A, \ldots, D$ on samaaegselt «ühed». Kui üks või mitu



Joonis 3

sisendpinget on «nullid», siis on ka pinge F «null». Vastava loogilise skeemi praktiline lülitus dioodide abil on esitatud joonisel 3, *b*.

Sellelt jooniselt näeme, et F saab omada potentsiaali $+e$ vaid juhul, kui kõik neli dioodi on suletud. Kui mõni dioodidest oleks avatud, tekiks seal vool ja seega ka pingelang takistil r, mistõttu pinge väljundil F oleks madalam kui $+e$. Dioodid saavad olla suletud vaid siis, kui neile on rakendatud pinge tõkkesuunas, antud juhul peab sisenditel $A, \ldots, D$ selleks olema potentsiaal $+e$. Seega toodud skeem realiseerib tõepoolest tehte $F = A \wedge B \wedge C \wedge D$.

Loomulikult võib sisendite arv viimases kahes skeemis olla neljast erinev.

Keerulisemate loogiliste tehete sooritamiseks tuleb sobival viisil ühendada terve rida «ja»- ning «või»-lülisid, kasutades eelmise astme väljundeid järgmise astme sisenditeks. Kuna kasutatavad lülituselemendid (dioodid) pole ideaalsed — neil on kindel ava- ja tõkkesuunatakistus, siis tuleb sellele vastavalt valida ka takisti r väärtus. Kui lülituse astmete arv on suur, siis on ka koormus suur ning r tuleb valida väiksem. Üldreeglina: mida keerulisem lülitus on dioodidest koostatud, seda tugevamad on lülituse töötamiseks vajalikud voolud. Enam kui neljaastmelise lülituse konstrueerimisel esinevad praktikas juba väga suured raskused.

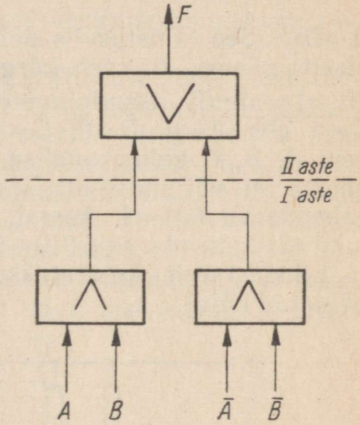
Keerulisema skeemi konstrueerimise näitena koostame lülituse, mis realiseerib signaalide A ja B loogilise samaväärsuse $F=A\sim B$. Teatavasti (vt. § 5)

$$F=A\sim B=(A\wedge B)\vee(\bar{A}\wedge\bar{B}).$$

Vajaliku lülituse puhul ei piisa järelikult üksnes sisendpingetest A ja B, vaid tuleb rakendada ka nende eitusi $\bar{A}$ ja $\bar{B}$. Tavaliselt aga annavad paljud elektron-arvutusmasinates kasutatavad lülitused (näit. trigerid) välja üheaegselt kaks pinget — väljumispinge ning tema eituse. Seetõttu eeldame, et eitused $\bar{A}$ ja $\bar{B}$ on varasemast etapist valmis kujul saadavad.

Koostatava lülituse saamiseks tuleb üksikuid loogilisi tehteid teostavad skeemid ühendada joonisel 4 näidatud viisil.

Vaadeldav skeem on kaheastmeline. Dioodide abil teostatud kujul on ta esitatud joonisel 5.



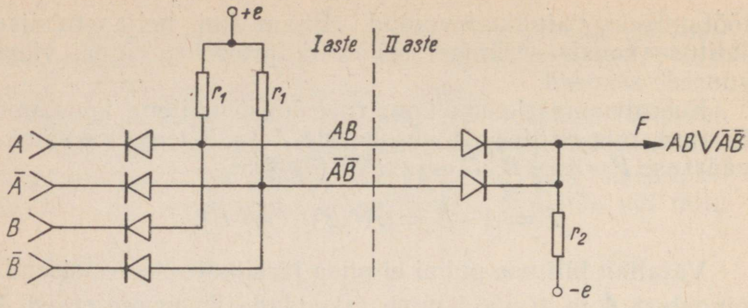
Joonis 4

Sageli kasutatakse elektron-arvutusmasinates veel lihtsat üheastmelist «ja»-lülidest koosnevat mitmeväljundilist skeemi, mida nimetatakse **diood-maatriksiks**[1]. Diood-maatriks leiab rakendamist peamiselt elektron-arvutusmasina juhtimisseadmetes. Tema tööpõhimõte selgub jooniselt 6.

Sisendpinged A, B ja C ning nende eitused antakse maatriksi horisontaaljuhtmetele. Kolme sisendpinge väärtuste kaheksale võimalikule kombinatsioonile ($2^3=8$) vastab kaheksa väljundit, mis on tähistatud «1» kuni «7»

---

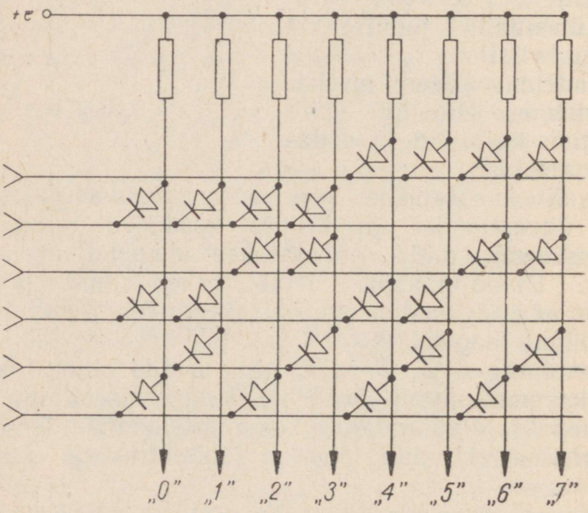
[1] Nimetus tuleneb asjaolust, et selle skeemi lülituselemente saab korraldada ridadesse ja veergudesse.



Joonis 5

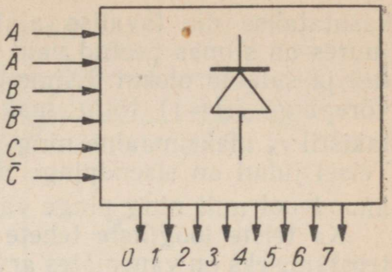
ja «0». See tähistusviis pole juhuslik, sest nagu me jooniselt näeme, kujuneb kõrge potentsiaal näiteks väljundil «1» ainult sisendpingete kombinatsiooni $\bar{A}=1$, $\bar{B}=1$, $C=1$ ehk $A=0$, $B=0$, $C=1$ korral. Lugedes sisendpingeid A, B, C kolmekohalise kahendarvu numbrikohtadeks näeme, et väljundil «1» saame kõrge potentsiaali just kahendarvu $001=1$ korral. Analoogiliselt vastab väljund «2» kahendarvule $010=2$ jne.

Lihtsustatult kujutatakse diood-maatriksit joonisel 7 näidatud kujul.



Joonis 6

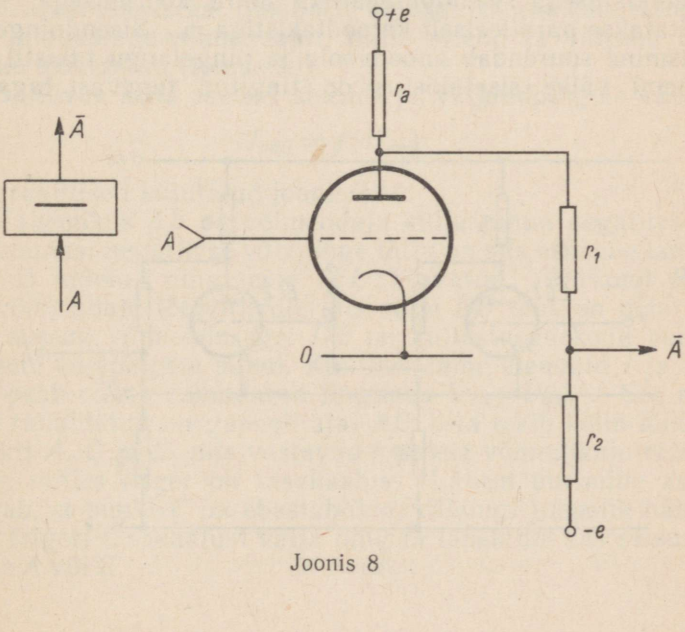
Tänapäeval on arvutusmasinates loogilisi tehteid teostavad lülitused koostatud peamiselt pooljuhtdioodidest, mis on odavad, konstruktsioonilt lihtsad ning töökindlad. Dioodide arv kaasaegsetes arvutusmasinates on väga suur: ühe kasutatava elektronlambi või transistori kohta tuleb tavaliselt 4—8 dioodi, dioodide arv ühes arvutusmasinas on 4—30 tuhat ning enamgi.



Joonis 7

Lülitused elektronlampidega. Dioodide abil teostatavad tehted (loogiline korrutamine ning liitmine) ei võimalda üldiselt iga operatsiooni realiseerida ning nendele lisaks vajame veel lülitust ka loogilise eituse teostamiseks («ei»-lüli ehk invertorit). Lihtsaimaks invertoriks osutub võimendajana lülitatud triood, mille sümboolne tähistus ja põhimõtteskeem on esitatud joonisel 8.

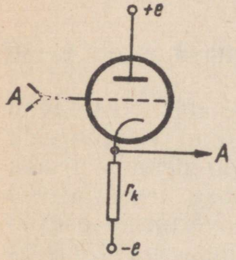
Pingejagaja $r_1 — r_2$ on vajalik väljundpinge viimiseks samasse muutumisvahemikku nagu sisendpinge. Trioodi



Joonis 8

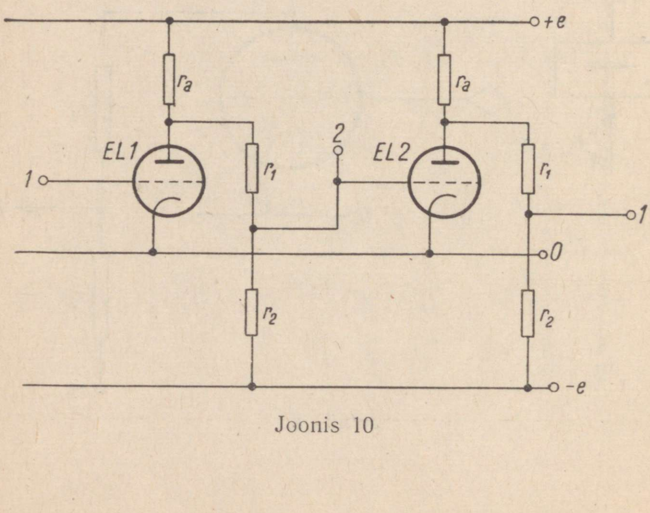
kasutatakse siin tavalise alalispinge-võimendajana, kusjuures on silmas peetud vaid kaht töörežiimi: trioodi avatud ja suletud olekut. Esimesel juhul on anoodvool kõrge võrepinge ($A=1$) tõttu maksimaalne, seega pingelang takistil r_a maksimaalne ning väljundpinge «null» ($\bar{A}=0$). Teisel juhul on sisendpinge madal ($A=0$), lamp suletud, anoodvool null ning pinge väljundis kõrge ($\bar{A}=1$).

Ka teiste loogiliste tehete (liitmine ja korrutamine) teostamiseks on vanemates arvutusmasinate tüüpides kasutatud elektronlampe, kuid seoses kõrgekvaliteediliste pooljuhtdioodide laialdase tootmisega on neist nüüd loobutud. Samuti kasutati varem skeeme, kus mitu trioodi oli lülitatud tööle alalispingevõimendajana ühise anoodkoormustakistiga, või siis katoodjärgijaid ühise katoodtakistiga.

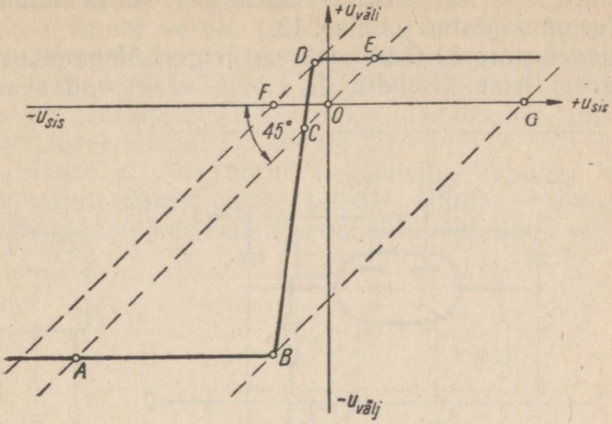


Joonis 9

Elektronlamp leiab veel kasutamist katoodjärgija-lülituses (joonis 9), mis küll ei soorita mingit loogilist tehet, vaid sobitab suure sisetakistusega sisendpingeallika suure koormusega, mis lülitatakse paralleelselt katoodtakistiga r_k. Sisendpinge A tõusmine suurendab anoodvoolu ja pingelangu takistil r_k. Skeemi väike sisetakistus on tingitud tugevast tagasi-



Joonis 10



Joonis 11

sidest r_k kaudu, sest pinge takistil r_k mõjub vastu sisendpingele A.

Eriti olulise tähtsusega lülituseks elektron-arvutusmasinas on t r i g e r. Skeemi põhiliseks omaduseks on võime salvestada ühte kahendnumbrit, s. t. olla mälu elementaarosakeseks. Sisuliselt kujutab skeem kaheastmelist alalispinge-võimendajat, mille sisend ja väljund on kokku ühendatud (joonis 10).

Sõltuvus selle skeemi sisend- ja väljundpingete vahel

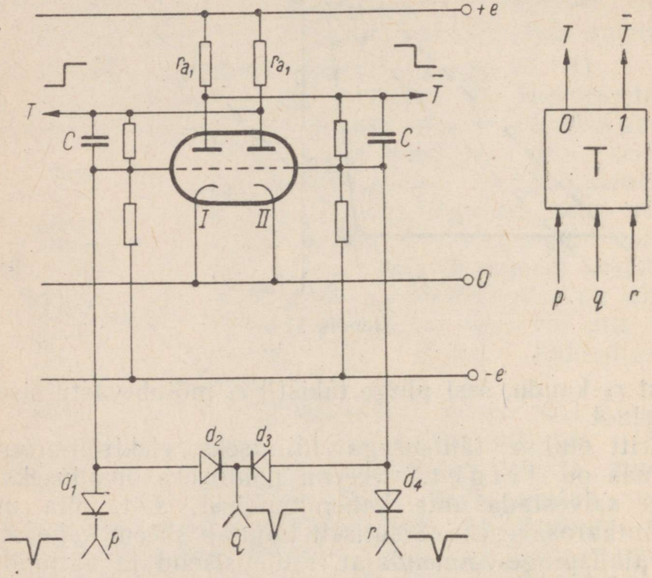
$$U_{välj} = f(U_{sis})$$

on graafiliselt kujutatud joonisel 11.

Vahemikus AB on võimendaja küllastunud negatiivses piirkonnas: negatiivse võrepinge tõttu on siis esimene lamp (EL 1) suletud ning teine (EL 2) avatud. Piirkond BD on võimendaja tööpiirkond, kus sirge BD tõus on määratud skeemi võimendusega. DE on küllastuspiirkond positiivsete võrepingete puhul. Kui ühendame klemmid *1* ja *1'*, siis peab olema rahuldatud tingimus $U_{sis} = U_{välj}$. See on aga rahuldatud nurgapoolitajal AE. Siin tekib kolm lõikepunkti A, C ja E, mis vastavad kolmele võimalikule režiimile, milles triger on tasakaalus. Lähem uurimine aga näitab, et punkt C on ebastabiilne, vähimgi juhuslik häire viib trigeri tasakaalust välja ning ta läheb üle kas seisundisse A või E.

Praktiliselt kasutatava trigeri skeem ja sümboolne tähistus on näidatud joonisel 12.

Kondensaatorid C kiirendavad trigeri üleminekut ühest olukorrast teise. Dioodid $d_1, \ldots, d_4$ eraldavad sisendeid.

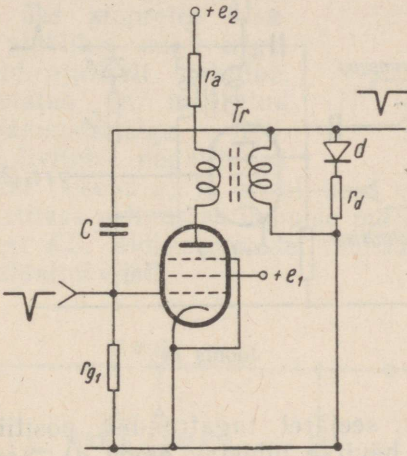


Joonis 12

Trigeril on kaks väljundit: T ja tema eitus ehk inversioon $\overline{T}$. Lepime kokku, et juhul kui on suletud esimene triood ning avatud teine, siis on trigeris salvestamisel arv null; kui aga juhib esimene triood ning on suletud teine, siis arv üks. Seega kui trigeris on salvestatud arv üks, siis $T=1$ ja $\overline{T}=0$, kui aga salvestamisel on arv null, siis $T=0$ ja $\overline{T}=1$.

Trigeri ümberlülitamist seisult «üks» seisule «null» ja vastupidi selgitavad joonised 10 ja 11. Kui rakendame punktide *1* ja *1'* vahele või punkti *2* lisapinge, siis sirge AE nihkub paraleellükkega kas vasakule või paremale. Nagu joniselt 11 näeme, on trigeri tasakaalupunkti üleviimiseks punktist E punkti A vajalik negatiivne pinge suurusega OF. Vastupidiseks üleviimiseks vajalik positiivne pinge OG peab olema tunduvalt suurem. Seega tuleb triger lülitada ümber negatiivse pingeimpulsiga, mis antakse a v a t u d lambi võrele (ümberlülitamiseks vajalik positiivne pingeimpulss s u l e t u d lambi võrele peaks olema märksa suurem). Trigeri ümberlülitamiseks vajalike impulsside jaotamiseks koostatakse dioodskeem, mis eraldab kolm sisendahelat (vt. joonis 12):

1. Sisend *p*. Siia antud negatiivne impulss lülitab trigeri seisult «üks» ümber seisule «null». Vastupidine ümberlülitamine pole selle sisendi abil võimalik.



Joonis 13

2. Sisend *r*. Siia antud negatiivne impulss lülitab trigeri seisult «null» ümber seisule «üks». Vastupidine ümberlülitamine pole võimalik.

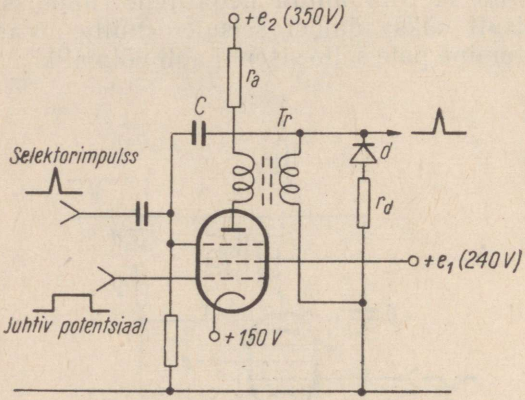
3. Sisend *q*. Siia antud negatiivne impulss lülitab trigeri ümber teise tasakaaluolukorda sõltumata eelmisest, s. t. lülitamine toimub nii seisult «null» seisule «üks» kui ka vastupidi.

Toodud skeem on üks lihtsamaid. Arvutusmasinates kasutatakse sageli aga väga keeruka skeemiga trigereid, mis on tingitud vajadusest tõsta ümberlülitumise kiirust või muuta skeem vähetundlikuks toitepingete muutumisele ja lampide vananemisele.

Tutvume lühidalt ka teiste enamkasutatavate impulsstehniliste skeemidega.

Äärmiselt vajalik lülitus on i m p u l s s v õ i m e n-d a j a, mis on ehitatud tavaliselt trafosidestuses. Sealjuures kasutatakse spetsiaalseid, kõrgsagedusrauast või ferriidist toroidikujulisele südamikule keritud impulsstrafosid (joonis 13).

Impulsi formeerib tegelikult impulsstrafo, mille sekundaarmähises tekib algul sisendimpulsi esifrondist nega-



Joonis 14

tiivne impulss, seejärel tagafrondist positiivne impulss (viimase aga hävitab lühistav diood *d*). Samuti summutab diood kõik võnkumised, mis võiksid tekkida puisteinduktiivsusel ja mahtuvusel.

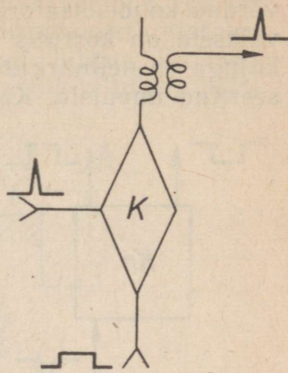
Niisugune võimendusaste valmistatakse tavaliselt kahe sisendiga. Joonisel 14 kujutatud võimendaja on ette nähtud selektorimpulsside[1] jaotamiseks. Kasutatud pentood on lülitatud nii, et anoodvool on null nii negatiivse võrepinge kui ka negatiivse sulgvõrepinge puhul.

Positiivne impulss sulgvõrel võib avada lambi vaid siis, kui võrele on antud positiivne potentsiaal (s. t. potentsiaal «üks»). Ainult sel juhul töötab lamp võimendajana. Lamp funktsioneerib seega lülitina ning sageli nimetatak-

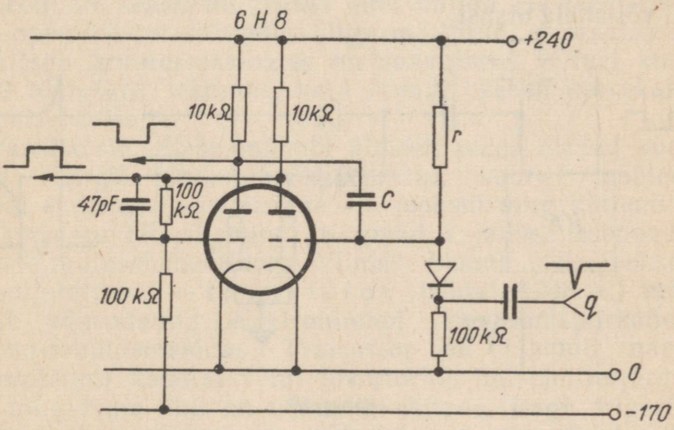
[1] Selektorimpulss on selline impulss arvutusmasinas, mida ei kasutata töödeldava informatsiooni edasiandmiseks, vaid ainult üksikute seadmete töö kooskõlastamiseks ja sünkroniseerimiseks. Näiteks informatsiooni kustutamiseks aritmeetilise seadme registrist kasutatakse selektorimpulssi, mis juhitakse üheaegselt registri kõigi trigerite sisenditele «*p*» (lähemalt vt. § 9).

segi teda «väravaks», «ventiiliks», «klapiks». Lihtsustatult tähistame teda joonisel 15 näidatud kujul.

Teistest impulsstehnika põhilülitusskeemidest leiab elektronarvutusmasinates kasutamist nn. k i p p r e l e e. See on triger, kus üks anoodi-võre sidestus pole teostatud mitte pingejagajaga, vaid *RC*-ahela kaudu (joonis 16). Seetõttu on üks kipprelee tasakaalupunkt stabiilne nagu trigeril, teine vaid ajutiselt stabiilne, kusjuures kestus on määratud *RC*-ahela ajakonstandiga $\tau = RC$. Kipprelee käivitub negatiivsest impulsist võrel (sisend *q*) ja annab kaks väljundit: positiivse ja negatiivse nelinurkse impulsi, millede kestus sõltub korrutisest *RC*. Blokkskeemidel tähistame kippreleed joonisel 17 näidatud viisil.

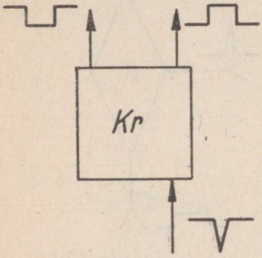


Joonis 15



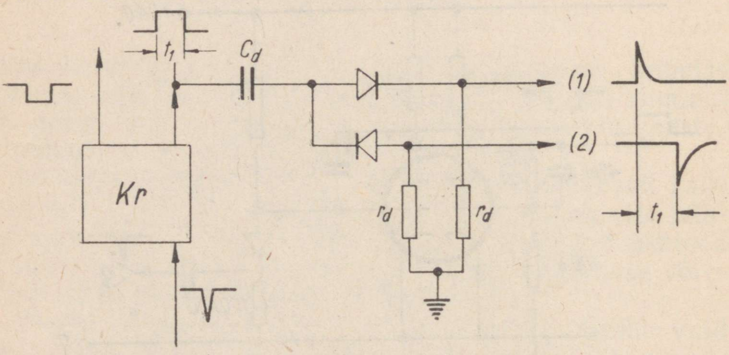
Joonis 16

Kipprelee võimaldab konstrueerida praktiliselt suvalise viitega viiteahelaid. Nagu me joonisel 18 näeme, moodustavad kondensaator C_d ja takistid r_d nn. diferentseeriva ahela: kondensaatori taga tekkiv pinge on ligikaudselt võrdne kondensaatori ees oleva pinge tuletisega[1]. Mida väiksem on korrutis $C_d r_d$, seda väiksemad ja teravama kujuga on nelinurkse impulsi frontidele vastavad diferentseeritud impulsid. Kaks dioodi eraldavad mõlema polaarsusega impulsid: väljundis *2* saame negatiivse impulsi, mis kippreleed käivitava impulsi suhtes omab viite t_1, väljundis *1* aga käivitava impulsiga peaaegu samaaegse positiivse impulsi.



Joonis 17

Teine võimalus lühiajalise viite saamiseks (alates kümnendikest mikrosekunditest kuni kümne mikrosekundini) seisneb kunstlike liinide kasutamises. Selleks võib olla spetsiaalse koaksiaalkaabli lõik (joonis 19, *a*) või induktiivsustest-mahtuvustest koostatud ahel (joonis 19, *b*). Selline liin või ahel osutub mõningal juhul odavamaks ja lihtsamaks kui kipprelee, kuid ta peab olema hoolikalt sobitatud koormusega, et ei tekiks impulsside peegeldusi liini või ahela otsast.



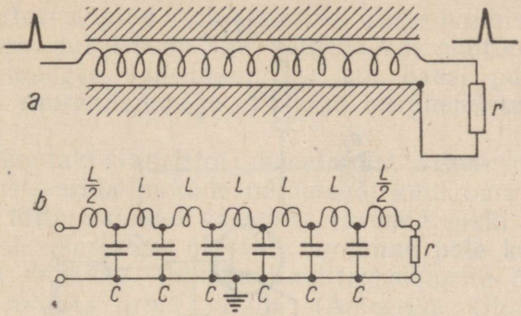
Joonis 18

Ülalvaadeldud lülitusi informatsiooni salvestamiseks võib nimetada s t a a t i l i s t e k s, sest nende seisundi-

---

[1] Tuletis mõõdab suuruse muutumise kiirust. Kui kondensaatori C_d ees olev pinge kasvab, siis kondensaatori taga tekib (muutumise kiirusega võrdeline) positiivne pinge; kui aga C_d ees pinge langeb, siis tema taga kujuneb negatiivne pinge.

teks on tasakaaluasendid. Viimasel ajal on levinud aga ka dünaamilised lülitused, kus informatsioon alatasa regenereerudes pidevalt ringleb. Kogu lülitus koosneb viiteahelast, võimendajast, «ja»- ning «või»-lülidest. Või-

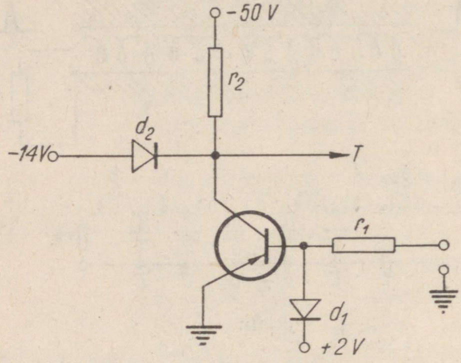


Joonis 19

mendaja väljundist juhitakse signaal sisendisse tagasi ning ta jätkab ringlemist piiramata aja jooksul. Sellised dünaamilised trigerid võimaldavad saavutada aritmeetiliste seadmete kiirust kuni miljon (ja isegi rohkem) operatsiooni sekundis. Olgu märgitud, et staatiliste trigeritega arvutusmasinates on saavutatud kiirusi kuni 16 000 töötsüklit sekundis, kuid sedagi skeemi keerukaks muutmise hinnaga.

Transistorid. Elektronlambi üldiselt teada olevad puudused (kõrge hind, ebaökonoomsus, suured mõõted, katoodi emisiooni vähenemine aja jooksul ning kütteniidi ettenägematu läbipõlemine) tingivad vajaduse asendada ta teiste lülituselementidega. Viimase kümne aasta jooksul on pooljuhttriood — transistor (leiutati 1948. a.) tunduvalt vähendanud elektronlambi kasutamispiirkonda elektron-arvutusmasinas. Transistor on (samuti nagu elektronlamp) kasutatav nii invertorina kui emitterjärgijana ning tema abil on võimalik ehitada isegi trigerit. Ainuke raskus, millega transistori kasutamisel tuleb võidelda, on tema madal sisendtakistus. Seega on transistor küllaltki suur koormus, mida tuleb arvestada tema ühendamisel kõrge sisetakistusega lülitusskeemide väljunditesse. Kuigi transistor on suuteline võimendama voolu kuni 100 korda, ei tähenda see muidugi, et ühe transistoriga võiks tüürida sadat sisendahelat.

Teatavasti kasutatakse tehnikas nii punkt- kui ka pindkontaktiga transistore, kusjuures viimased on suutnud end maksma panna peaaegu kõigis lülitustes ning hakkavad välja tõrjuma kuuma katoodiga elektronseadmeid.

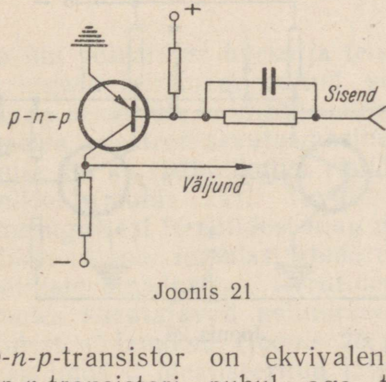


Joonis 20

Meenutame, et punktkontaktiga transistoril on kolm klemmi, mis on ühendatud vastavalt emitteriga, baasiga ja kollektoriga. Emitter pingestatakse baasi suhtes positiivselt, kollektor negatiivselt. Sisendvooluringis kulgeb vool ülemineku emitter-baas suhtes avasuunas, väljundvooluringis ülemineku baas-kollektor suhtes tõkkesuunas. Kollektorivoolu muutuste suhe emitterivoolu muutustesse tähistatakse tavaliselt tähega α (see on transistori vooluvõimendustegur). Asjaolud, et punktkontaktiga transistori puhul α sõltub emitterivoolust ning et mitte väga väikeste emitterivoolude korral on α kahe-kolme piirides, võimaldavad koostada ühe transistori abil lülitusi, millel on kaks stabiilset tasakaaluseisundit nagu trigeril. Üks vastav lülitus on näidatud joonisel 20. Kui transistor on suletud, siis baas on emitteri suhtes positiivne ning diood d_1 ei lase baasi potentsiaalil tõusta üle $+2$ V. Ka kollektori potentsiaal ei tõuse dioodi d_2 tõttu kõrgemale kui -14 V. Kirjeldatud olukord on stabiilne. Kui muuta emitter hetkeks baasist positiivsemaks, siis tekib vool läbi baasi- ja kollektoritakistite r_1 ja r_2. Pingelang takistil r_1 muudab baasi potentsiaali (küllalt suure r_1 puhul) negatiivsemaks emitteri potentsiaalist. Et aga sel juhul esinev emitterivool võimendatakse α-kordselt ning seetõttu üha tugevnev vool läbi kollektori ja baasi suurendab veelgi pingelangu takistil r_1, siis suureneb kollektorivool väärtuseni, mis on määratud takisti r_2 ja transistori karakteristikute poolt. Seega on ka transistori avatud olek stabiilne. Tagasiviimiseks suletud olekusse tuleb transistor sulgeda kas emitteri potentsiaali lühiaegse vähendamisega või baasi potentsiaali lühiaegse suurendamisega. Väljund on ühendatud kollektorile.

Punktkontakt-transistori kasutamist arvutusmasinates võiks illustreerida paljude näidetega, kuid oma komplitseerituse tõttu vajavad vastavad skeemid pikki selgitusi, mistõttu üksikasjalise ülevaate andmine pole käesolevas raamatus mõeldav. Punktkontakt-transistorite baasil on ehitatud näiteks firma «Bell» (Ameerika Ühendriigid) arvutusmasin TRADIC, mille impulsside sagedus on 1 MHz.

P i n d k o n t a k t i g a   t r a n s i s t o r koosneb kolmest tsoonist, mis puutuvad kokku kahel kontaktpinnal. Valmistatakse kahte tüüpi pindkontakt-transistore: *p-n-p* ja *n-p-n*. Tähed *n* ja *p* tähistavad üksteisel asuvaid tsoone: tsoonis *n* on negatiivsed vabad laengukandjad — elektronid, tsoonis *p* aga positiivsed vabad laengukandjad — «augud». Suund *p-n* on kontaktpinna avasuund, *n-p* tõkkesuund. Pindtransistor pingestatakse samuti nagu punkttransistor, keskmine tsoon vastab baasile.

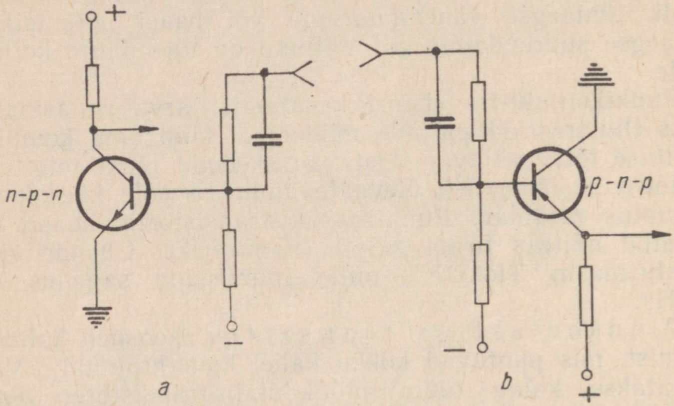


Joonis 21

Sealjuures *p-n-p*-transistor on ekvivalentne punkttransistoriga, *n-p-n*-transistori puhul aga tuleb vahetada kõikide pingete polaarsused. Muide on valmistatud ka

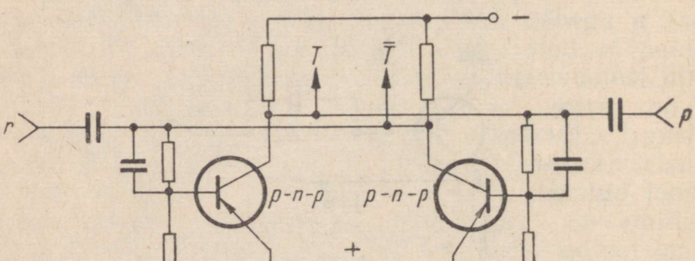
n-p-n-punkttransistore, kuid nende kasutamine on piiratud ebarahuldavate parameetrite ja tehnoloogiliste raskuste tõttu valmistamisel.

Pindtransistori vooluvõimendustegur α on alati väiksem kui 1, kuid kui lülitada transistor tööle maandatud emitteriga võimendajana (joonis 21), võime saavutada



Joonis 22

suuri vooluvõimendusi. Tõepoolest, joonisel 21 toodud lülituste vooluvõimendustegur $\beta=\frac{\alpha}{1-\alpha}$ on $\alpha=1$ puhul ∞ ja ühest veidi väiksema α puhul küllaltki suur.



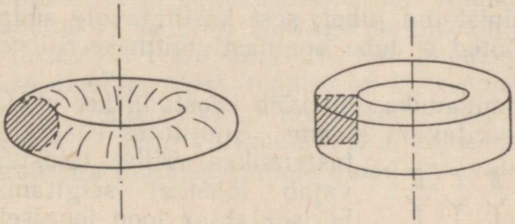
Joonis 23

Pindkontakt-transistori baasil võime koostada kaks põhilist lülitust: invertori (joonis 22, *a*) ja emitterjärgija (joonis 22, *b*). Invertor on tegelikult ülalkirjeldatud võimendaja, emitterjärgija aga täiesti sarnane elektronlampkatoodjärgijaga.

Analoogiliselt vastavale elektronlamp-skeemile on võimalik valmistada ka staatilist trigerit, mille skeem on toodud joonisel 23.

Transistoride kasutamise puhul vähenevad arvutusmasina gabariidid ning energiatarvitus tunduvalt. Nii näiteks firma IBM arvutusmasina tüüp 605 ümberehitamine transistoridele vähendas masina mahtu 3 korda ning energiatarvitust 6 kW-lt 300 W-le. Lisakokkuhoiuna avanes võimalus loobuda ventilatsioonisüsteemist.

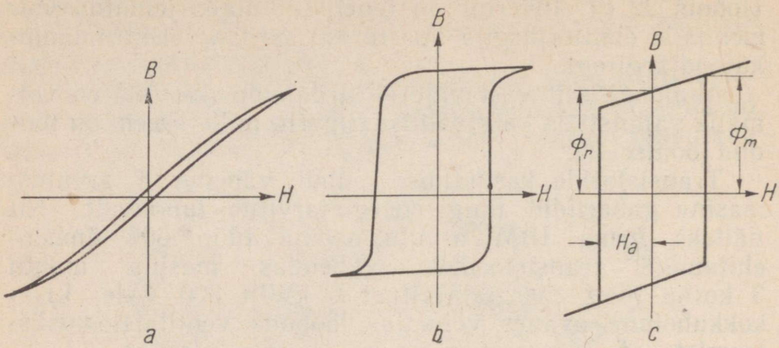
Ferromagnetilised südamikud arvutusmasinates. Ferromagnetiline südamik on kasutatav tänu oma omadustele salvestada jääkmagnetvälja ühes või teises suunas. Kahendsüsteemis töötava masina puhul on sobiv lugeda



Joonis 24

magnetvälja üht polaarsust üheks ja teist nulliks. Võimalik on veel lugeda lahtimagneeditud südamikku numbriks *2* ja minna seega üle kolmendsüsteemile.

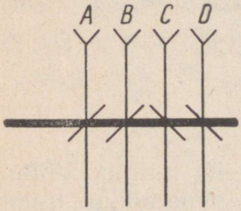
Kaasaegsetes elektron-arvutusmasinates kasutatakse nii ümmarguse kui ka ristkülikulise ristlõikega rõngaid — toroidsüdamikke (joonis 24). Tavaliselt valmistatakse toroide mitmesugustest ferriitidest ning neid võib liigitada kahte põhilisse rühma: impulsstrafode südamikud pehme raua magnetiliste omadustega ferriitidest ja mälu- ning lülitusseadmetes kasutatavad nelinurkse hüstereesisilmusega ferriitidest südamikud (joonis 25, *a* ja *b*). Joonisel on hüstereesisilmus *b* idealiseeritud kujule *c*, kus Φ_r/Φ_m on nelinurksuse teguriks, Φ_r — jääkvoog ja H_a — südamiku koertsitiivjõud.



Joonis 25

Südamiku magneetimiseks vajalik magnetomotoorne jõud tekitatakse südamikule keritud mähises voolu abil. Sealjuures tuleb mähise all mõista ka üks kord südamiku seest läbipistetud juhet, sest kasutatavate südamike väikesed mõõted ei luba enamasti mitmekeeruliste mähiste kerimist [1].

Toroidsüdamike abil võib koostada keerukaid loogilisi tehteid sooritavaid skeeme. Siinjuures kasutatakse arvutustehnikas erilist tähistusviisi, mis vajab lähemat selgitamist. Jäme horisontaalne joon joonisel 26 tähistab südamikku, vertikaalsed jooned aga mähiseid, kus voolu suunaks on loetud suund ülalt alla. Mähise poolt tekitatava magnetvoo suund on määratud kaldjoonega ristumiskohal, kusjuures kaldjoont tuleb vaadelda nagu peeglit, millelt voolu suunas langev valguskiir peegeldub südamikus tekkiva magnetvoo suunas.



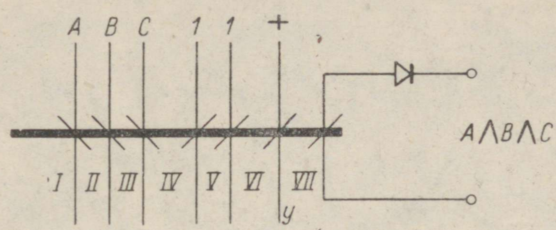
Joonis 26

Selleks et selgitada loogiliste tehete teostamist ferriitsüdamike abil, analüüsime lähemalt lülitust joonisel 27.

Mähiseid *IV* ja *V* läbiv vool tugevusega «üks» tekitab vasakule suunatud magnetvoo, kusjuures voolu tugevus «üks» on küllaldane südamiku ümbermagneetimiseks.

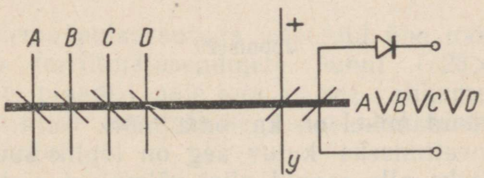
[1] NSV Liidus on valmistatud masin toroidtrafode kerimiseks südamikele välise läbimõõduga 3 mm, ava läbimõõduga 2 mm ja paksusega 1 mm. Kasutamist leiavad aga ka väiksemad toroidsüdamikud.

Seega on südamik sel juhul magneeditud suunaga vasakule. Lepime kokku, et niisugune magneetumise suund vastab numbrile null, vastupidine aga numbrile üks. Kui laseme läbi mähise *VI* tugeva vooluimpulsi *y*, siis ümbermagneetumist ei toimu ning mähises *VII* seega pingeimpulsse ei indutseerita. Kui juhime läbi mähiste *I*, *II*



Joonis 27

ja *III* voolud *A*, *B* ja *C*, siis kaks neist kompenseerivad voolud mähistes *IV* ja *V*, kolmas aga magneedib südamiku ümber. Mähises *VII* tekib siinjuures küll impulss, kuid see tõkestatakse dioodiga, sest impulss on teise polaarsusega. Impulsi *y* mõjul suunas «üks» magneeditud südamik magneeditakse ümber ning väljundis *VII* indutseeritakse sellise polaarsusega impulss, millele diood on avatud. Seega

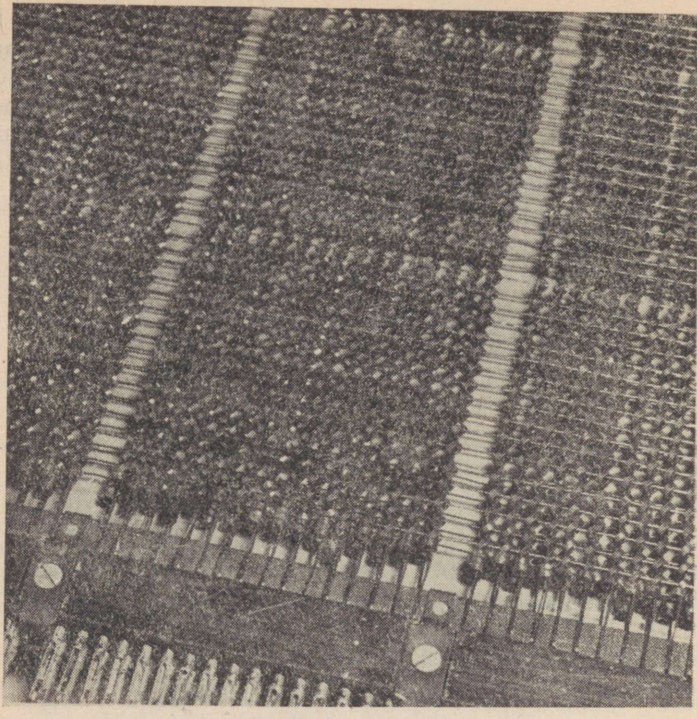


Joonis 28

saime väljundimpulsi vaid siis, kui $A=1$, $B=1$ ja $C=1$, millest nähtub, et skeem teostab signaalide *A*, *B*, *C* loogilise korrutamise.

Loogilist liitmist teostav skeem on näidatud joonisel 28.

Ferriitsüdamike abil võib koostada ka trigeriga analoogilisi lülitusi. Eriti sobivad on ferriitsüdamikud aga lülituste moodustamiseks koos transistoridega. Sellisel viisil on võimalik konstrueerida tervet arvutusmasinat.

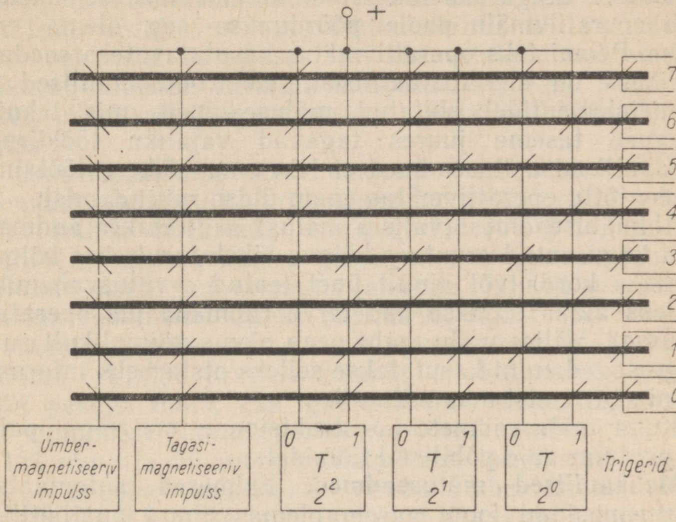


Joonis 29

Kuid ferriitsüdamikel on ka oma nõrk koht: südamiku ümbermagneetimiseks kuluv aeg on lõplik suurus, mille vähendamiseks alla teatud piiri võimalusi ei tunta. Seetõttu on maksimaalne ümbermagneetimiste arv sekundis veidi üle ühe miljoni.

Üks olulisemaid ferriittoroididega loogiliste skeemide rakendusi on ferriitmäluseadme (joon. 29) aadressivalija. Kuna järgmises paragrahvis tuleb vaatlemisele ferriitmäluseade, siis on ka aadressivalijaga tutvumine vajalik. Joonisel 30 on toodud kaheksa aadressi valikuga lülitus, milles kasutatakse kaheksat ferriitsüdamikku. Parempoolne mähisterida on ühendatud ferriitmäluseadme koordinaatjuhtmetele.

Algul on seadme magnetsüdamikud magneeditud suunaga vasakule. Trigerite seisundite iga kombinatsiooni puhul leidub üks kindel südamik, mille ühtki mähist vool ei läbi ja seega ainult selle südamiku magneedib vastav vooluimpulss ümber. Parempoolses mähises indutseeri-



Joonis 30

takse mõlema polaarsusega impulsid ühe töötakti vältel.

Suurte ferriitmäluseadmete puhul (128×128 südamikku ühel plaadil) pole aga ka see valikumeetod rakendatav ning siis kasutatakse lisaks veel süsteeme dioodmaatriksitega.

7. ELEKTRON-ARVUTUSMASINATE MÄLUSEADMED

Mäluseadmete põhitüübid. Matemaatilise ülesande lahendamise käigus tuleb teatud aja vältel säilitada mitmesuguseid arvutusteks vajalikke suurusi. Sellisteks suurusteks on näiteks algandmed, arvutuste käigus saadud vahepealsed tulemused, arvutamiseks vajalikud konstandid, kogu arvutuse käiku juhtiv programm, abiprogrammid vahepealsete arvutuste jaoks jne. Nende salvestamiseks peabki elektron-arvutusmasin omama vastavaid mäluseadmeid. Oma otstarbe ja kasutamise viisi poolest jagunevad mäluseadmed operatiiv- (sisemine mälu) ja abimäluseadmeteks (väline mälu).

Operatiivmälu on määratud piiratud arvu kõige sagedamini (antud arvutustes) vajaminevate suuruste säilitamiseks. Seega masina töökiiruse tõstmise seisukohalt peab operatiivmälu poole pöördumise aeg olema eriti lühike. Peamisteks operatiivmälus kasutatavateks seadmetüüpideks on elektrostaatilised, elektrodünaamilised ja magnetelementidel ehitatud mäluseadmed, mis tehnika praeguse taseme juures tagavad vajaliku töökiiruse. Mehaanilised mäluseadmed ei leia oma väikese töötamiskiiruse tõttu operatiivmälus enam üldse rakendamist.

Abimäluseadmes (välises mälus) säilitatakse andmeid, mida kogu antud arvutuse käigus läheb tarvis kas kõigest üks-kaks korda või ainult ühel teatud arvutusvahemikul (selleks ajaks tuuakse nad terve rühmana üle operatiivmälusse). Välise mälu maht peab olema võimalikult suur. Kõige sagedamini kasutatakse selleks otstarbeks informatsiooni säilitamist magnetlindil.

Kuna mäluseadmete konstruktsioone on väga palju, siis peatume vaid põhilistel tüüpidel.

Mehaanilised mäluseadmed. Esimesed automaatsed arvutusmasinad loodi varem olemas olnud analüütiliste arvutusmasinate üksteisega seostamise teel. Informatsioon anti ühest masinast teise edasi ning säilitati perfokaartide või perfolintidena. Perfokaartidel või -lintidel kujutatakse kahendarve vastavate aukude süsteemi abil, kusjuures augu olemasolu antud kohal tähendab numbrit 1 ning augu puudumine numbrit 0.

Praegusel ajal kasutatakse perfokaarte ja -linte vaid elektron-arvutusmasinate sisend- ja väljundseadmetes. Seetõttu kirjeldame seda juhtu lähemalt järgmises paragrahvis.

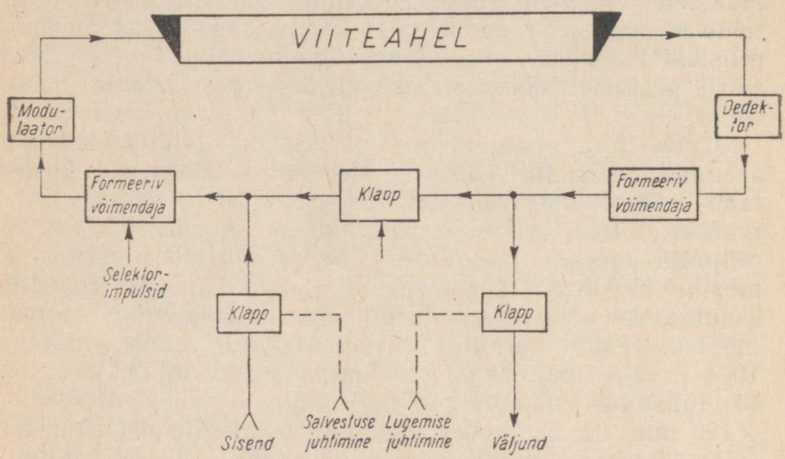
Elektromehaanilised releed olid ajalooliselt esimeseks elektriliseks konstruktsiooniks, mida kasutati elektron-arvutusmasinais vajaliku informatsiooni säilitamiseks (mäluseadmeks). Informatsioon anti elektriliste impulsside näol mäluseadmeisse ning salvestati seal releede ergutatud või vaba oleku näol. Et salvestada informatsiooni sel viisil näiteks elektron-arvutusmasinas «Strela», tuleks iga pesa koostada 43 releest (vastavalt kohtade arvule pesas — vt. § 4). Selline konstruktsioon oleks ilmselt keeruline, kuid peamise negatiivse külje moodustaks siiski elektromehaaniliste releede aeglane töötamine. Praktiliselt osutub nimelt raskeks konstrueerida releed, mis lülituks ümber tihedamini kui 50—100 korda sekundis. Sel põhjusel ei ulatunudki esimeste, elektromehaaniliste releede baasil ehitatud automaatsete arvutusmasinate töötamiskiirus üle paarikümne tehte sekundis. Praegusel ajal kasutatakse elektromehaanilistest releedest ehitatud mäluseadmeid vaid väga lihtsates, peamiselt demonstratsiooni otstarbel ehitatud masinates.

Ümberlülitamise kiirendamiseks on tehtud katseid asendada mehaanilised releed elektronreleedega, s. t. kasutada selleks otstarbeks lamptrigereid (vt. § 6), mis võimaldavad tõsta mälusse pöördumise sagedust kümnete miljonite kordadeni sekundis ja seega oluliselt suurendada masina töökiirust. Üheks peamiseks takistuseks trigerite kasutamisel mäluseadmena on aga saadava mälu suured mõõted. Lihtne arvutus näitab, et kui tuleb ehitada 1024 pesaga mälu, kus igas pesas on 43 kohta, siis üle 44 tuhandest lamptrigerist (mõõted umbes $40 \times 90 \times 120$ mm) ja kümnetest tuhandetest loogilistest elementidest koosnev mäluseade nõuab umbes 20 m^3 ruumi. Kasutades 2,5 m kõrgusi ja 0,6 m sügavaid kappe võtab mälukappide rida enda alla üle 13 jooksva meetri ning tema energiatarvitus ulatub 100 kilovatini.

V i i t e a h e l a t e l (viiteliinidel) ehitatud mäluseadme aluseks on järgmine seaduspärasus: täielikult kooskõlastatud pikas ahelas kandub informatsioon ahela ühest otsast teise antud olukorras kindla kiirusega ja teatud ajavahemiku möödumisel (sissekandmisest) lahkub ahela teisest otsast ilma peegelduseta.

Signaali ühekordne kulgemine ahela ühest otsast teise kestab standardses viiteliinis mõned millisekundid. Informatsiooni kestvamaks salvestamiseks tuleb see anda uuesti liini algusesse, luues seega ringahela. Liinid pole aga ideaalsed, mistõttu on vältimatud mõningad kaod ja moonutused. Antud signaali säilitamiseks muutumatuna lastakse see enne järjekordset ahelassesaatmist läbi formeeriva võimendaja, mis taastab impulsi algkuju. Perioodiliselt taastatav informatsioon võib antud ahelas ringelda praktiliselt piiramatu aja vältel.

Viiteliini skeem on antud joonisel 31. Liini geomeetriliste mõõdete vähendamiseks kasutatakse konstrueerimisel aineid, milledes võnkumised levivad võimalikult aeglaselt, kuid sealjuures ilma eriti suurte kadude ja moonutusteta. Informatsioonikandjana kasutatavate materjalide järgi jaotatakse viiteahelaid elektroakustilisteks, piesoelektrilisteks ja elektromagnetilisteks, milledest enam kasutatava-



Joonis 31

mad on elektroakustilised mäluliinid elavhõbedaga. Viimasel ajal on esitatud rida soovitusi kasutada viiteahelaks ka koaksiaalkaablit.

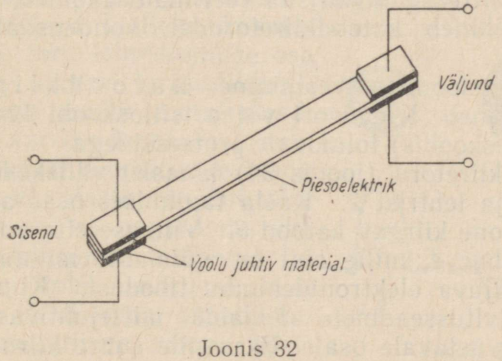
Elavhõbeliin on 50—100 cm pikkune ja 1—2 cm jämedune mõlemast otsast kvartsplaatidega suletud metalltoru. Kvarts omab piesoelektrilisi omadusi ning väljalõigatud kvartsplaadi painutamisel tekivad tema külgpindadel erineva märgiga laengud. Samal põhjusel hakkab pindadelt voolu juhitava ainega kaetud ja vooluahelasse lülitatud kvartsplaat painduma tema külgpindade vahel oleva pinge muutumise taktis. Seda omadust kasutataksegi elektroakustilistes mäluliinides, kus kvartsi elektrilisel teel sunnitakse sooritama võnkumisi ultraheli sagedusel (piirkonnas 25—50 kHz). Enne liini sisseviimist moduleeritakse seda võnkumist informatsiooni-signaalidega ning alles saadud kombineeritud võnkumised antakse sisendplaadi poolt üle temaga kokkupuutuvale elavhõbedale. Liini vastuvõtu-otsas muudetakse mehaanilised võnkumised kvartsi abil uuesti elektrilisteks, demoduleeritakse (eraldatakse informatsioon teda edasi kandnud ultrahelisageduslikest võnkumistest), võimendatakse ja formeeritakse uuesti, s. o. antakse neile nende algkuju. Kui informatsiooni ei ole arvutusteks veel tarvis kasutada, siis antakse ta uuesti liini algusesse ning kirjeldatud protsess kordub.

Ühes torus võib salvestada praktiliselt umbes 1000 informatsiooni-impulssi, mis antakse sisendile ühe mikrosekundiliste intervallidega. Et alati täpselt teada, millisele kohale kuuluva informatsiooniga meil on tegemist ning tagada arvutusmasina kõikide sõlmede üheaegne töö, tuleb väga täpselt säilitada elavhõbeda temperatuuri (helilainete levimiskiirus elavhõbedas sõltub temperatuurist). Et säilitada sõlmede üheaegset tööd vähemate tehniliste vahenditega, kasutatakse mõnikord meetodit, kus levimiskiiruse muutumisel muudetakse vastavalt ka liinile antavate impulsside kordumissagedust (sellega säilitatakse impulsskohtade arv liinil). Temperatuuri mõju vähendamiseks on soovitav kasutada võimalikult lühemaid liine.

Seoses kõigi nende raskustega on kodumaistes konstruktsioonides loobutud elavhõbeliinidel ehitatud mäluseadmeist ja üle mindud teistele mälutüüpidele. Välismaistest elektron-arvutusmasinatest on UNIVAC, BINAC ja mõned teised siiski ehitatud elavhõbeliin-mäluga.

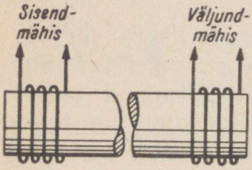
Peale elavhõbeliinide kasutatakse veel tahkest ainest ehitatud piesoelektrilisi ja magnitostriktsioonilisi ehk elektromagnetilisi liine.

P i e s o e l e k t r i l i n e akustiline mäluliin (joonis 32)



Joonis 32

erineb elavhõbeliinist põhimõtteliselt vaid mehaanilisi võnkeid edasikandva keskkonna poolest. Tema puhul pole tarvis spetsiaalseid muundajaid liini sisendil ja väljundil, sest kogu liin koosneb ühest terviklikust piesokristallist. Liini puuduseks on suur energiakadu muundumisel.



Joonis 33

M a g n i t o s t r i k t s i o o n i l i s e liini (joonis 33) põhimaterjaliks on nikkeltraat või ferriit. Sisendmähise poolt tekitatud väga lühiajalised magnetimpulsid levivad piki impulsikandjat. Impulsi levimisele nendes liinides kaasnev suur energiakadu ei luba teha pikki liine ning seetõttu tuleb suurendada impulsside kordumissagedust (vastavalt lühendades iga impulsi kestust).

Kõikide mäluliinide üldiseks puuduseks on vajadus oodata seni, kuni vajalik informatsioon jõuab väljundile ning uuesti formeeritakse. See protsess kestab mõne millisekundi ning seega on põhimõtteliselt võimatu saavutada üle 1000 pöördumise mälu poole sekundis (küllaldase mahuga mäluseadme puhul).

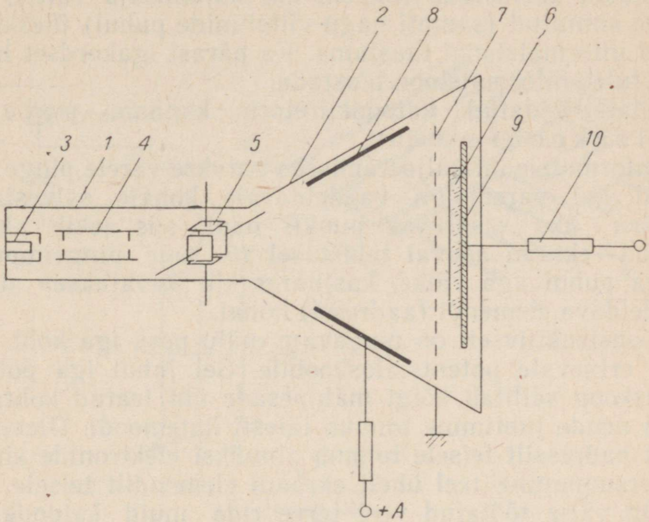
Praegusel ajal kasutatakse viiteahelaid kodumaises aparatuuris peamiselt vaid impulsside ühelt sõlmelt teisele üleandmise aja pikendamiseks. Neil juhtudel kasutatakse harilikult nn. «pikke liine koondatud parameetriga», mis koosnevad omavahel sobivalt ühendatud mahtuvustest ja induktiivsustest (vt. joonis 19).

Elektrostaatilised mäluseadmed võimaldavad arvude küllaltki kiiret salvestamist ja taastekitamist. Peale staatiliste trigerite kuuluvad siia veel mäluseadmed, mille töötamine põhineb katoodkiiretorudel, kondensaatoritel ja dioodidel.

Informatsiooni salvestamine k a t o o d k i i r e t o r u d e s sarnaneb televiisori või ostsilloskoobi katoodkiiretorus (kineskoobis) toimuvate protsessidega.

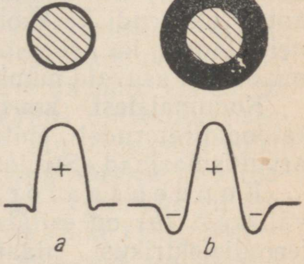
Katoodkiiretoru (joonis 34) klaasist väliskest koosneb kaelast *1* ja lehtrist *2*. Kaela tagumises osas asub pidevalt elektrone kiirgav katood *3*. Viimase ette on asetatud elektronkahur *4*, mille abil on võimalik muuta ekraani suunas väljuva elektronidekimbu tihedust. Kimp suunatakse hälvitusseadmete *5* abil mittejuhtivast ainest ekraani *6* vastavale osale. Ekraanile suure kiirusega langevad elektronid löövad ekraani kattepinnast *6* välja rohkem elektrone kui temale langes. See nähtus kannab sekundaaremisiooni ja uued elektronid sekundaarelektronide nime.

Ekraani ette on elektronide joa teele asetatud peenest



Joonis 34

traadist punutud võre *7*. Kui võrele anda ekraani suhtes positiivne pinge, siis kõik elektronide kimbu abil ekraanist väljalöödud sekundaarelektronid langevad võrele ja selle taga asuvaile kõrget positiivset pinget omavaile kollektori plaatidele *8* (selleks kasutatakse tihti toru lehtrikujulise osa sisepinnale kantud grafiidikihti). Ekraani taga asuva metallplaadi *9* (elementaar-kondensaator) suhtes tekib ekraanil positiivne laeng (joonis 35, *a*). Vastupidi, kui võre on ekraani suhtes negatiivne, siis sekundaarelektronid langevad tagasi sama koha ümbrusse, kust nad välja löödi ning moodustavad seal negatiivse rõnga posi-



Joonis 35

tiivse punkti ümber (joonis 35, *b*). Ühelt ekraani kohalt teisele ülemineku ajaks elektronide kimp «kustutatakse». Kuna ekraani katoodipoolne külg on dielektrik, siis temal tekkinud laeng ei kao mitte kohe, vaid püsib teatud aja vältel (kuni 0,2 sekundit). Kui meil on tarvis seda informatsiooni salvestada rohkem kui hajumisaja vältel, siis oleme sunnitud (samuti nagu viiteliinide puhul) üleskirjutatud informatsiooni taastama. Ka pärast igakordset lugemist tuleb informatsioon taastada.

Ülalkirjeldatud katoodkiiretoru kannab p o t e n t s i a l o s k o o b i nimetust.

Informatsiooni väljastamiseks antakse võrele pinge «0». Juhul kui varem oli vaadeldavale kohale salvestatud number üks positiivse punkti näol, siis tekib ahelas katood—ekraan asuval takistusel *10* suur pingeimpulss, rõnga puhul aga väike, kusjuures kiir «avatakse» ainult vaadeldava elemendi (aadressi) kohal.

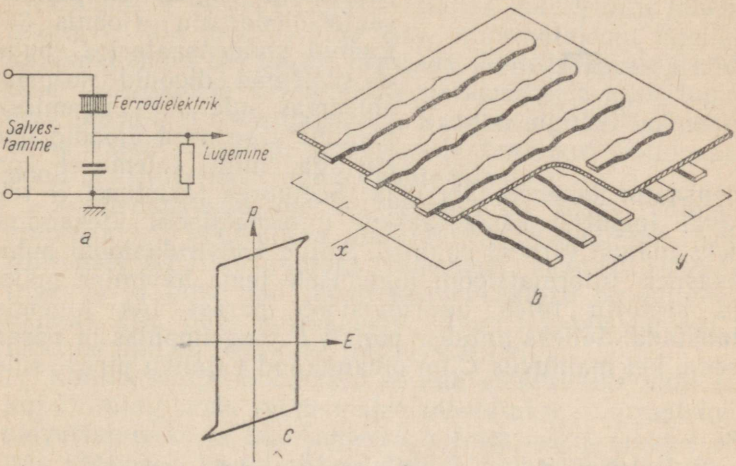
Konstruktiivselt on mugavam mälu pesa iga koht asetada erinevale potentsialoskoobile. Sel juhul iga potentsialoskoop säilitab kõigi mälupesade üht teatud kohta ja kõigi nende juhtimine toimub täiesti ühtemoodi. Üleminek ühelt aadressilt teisele toimub ainuüksi elektronide kimbu ümbersuunamise teel ühelt ekraani elemendilt teisele.

On välja töötatud veel terve rida muid katoodkiiretorude konstruktsioone ja kasutamisviise (selektron, potentsialoskoop laengu ümberjaotamisega ekraanil, kahe kiirega potentsialoskoop jne.). Kuid ka need ei suuda ületada katoodkiiretorudel ehitatud mäluseadme põhimõttelisi puudusi, milledest peamised on mäluseadme suured geomeetrilised mõõted ning piiratud arv pöördumisi (umbes 400 korda) ühe ja sama koha kõrval asuvate kohtade poole. Viimane puudus on tingitud sellest, et numbri null (rõnga) salvestamisel ei lange elektronid mitte ainult antud elemendi keskkoha otsesesse lähedusse, vaid osa neist langeb ka kaugemale, «külvates täis» ja maskeerides naabruses asuvaid numbrikohti.

Kodumaistest konstruktsioonidest omavad praegu katoodkiiretorudel ehitatud mäluseadet vaid elektronarvutusmasinad «Strela» ja M-2.

K o n d e n s a a t o r m ä l u elemendi skeem (*a*) ja väliskuju (*b*) on esitatud joonisel 36. Põhiosaks on siin ferrodielektrikust plaatisolaatoriga kondensaator, mille elektroodid on asetatud plaadile juhtivast ainest ribadena.

Plaadi kahel vastasküljel asuvate ribade ristumiskohal tekibki elementaarne kondensaator. Ferrodielektrikut iseloomustab jäik polarisatsioon, mille aste ja suund muutuvad järsult, kui kondensaatori plaatide vahele antud pinge ületab teatud piirväärtuse (joonis 36, *c*). Olles seega andnud

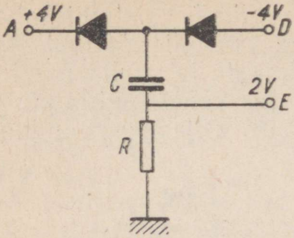


Joonis 36

ferrodielektrikust isolaatoriga kondensaatorile ühesuunalise laengu, võime me seda ilma lisaenergiakuludeta piiramatu aja jooksul säilitada. Igasugused mõjutused, mille suurus on väiksem teatavast kriitilisest pingest E_c, ei suuda muuta dielektriku polarisatsiooniseisundit ja seega ka kondensaatori laengut. Kui aga mõjutuse pinge ületab E_c, siis ferrodielektriku polarisatsioon muutub hüppeliselt ning kondensaator omandab uue laengu.

Informatsiooni salvestamiseks või lugemiseks antakse pinge vastavatele elektroodidele x ja y (elektrood y sealjuures maandatakse). Iga elektroodi x ja maa vahele on lülitatud koormustakistus. Kondensaatori ümberpolariseerimisel tekib takistusel vooluimpulss, mis antaksegi edasi skeemi (lugemisel ja nulli salvestamisel antakse elektroodile x positiivne, ühe salvestamisel aga negatiivne pinge).

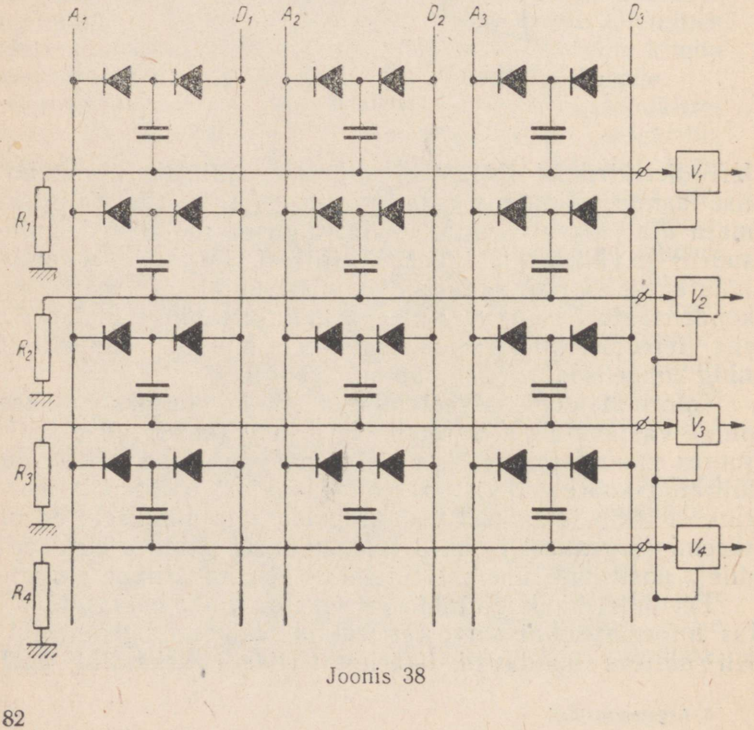
Ferrodielektrikuga mälu eeliseks on väike energiatarvitus informatsiooni salvestamisel ja väikesed gabariidid. Nii näiteks suudavad baariumtitanaadi kristallid ühel



Joonis 37

ruutsentimeetril säilitada 80 kahendnumbrit. Samuti on ferrodielektrilist mälu lihtne sidestada transistoridega.

Teiseks perspektiivseks operatiivmälu liigiks on kondensaator-dioodmälu (joonis 37). Laetud kondensaatori *C* puhul on mõlemad dioodid normaalolukorras suletud. Lugemiseks maandatakse need dioodid ühekorraga, mille tulemusel kondensaator laheneb läbi maa. Seejuures takistusel *R* tekkivat impulssi kasutataksegi informatsiooni ülekandmiseks mälust tarbiva sõlmeni. Antud konstruktsiooni puhul kaasneb informatsiooni lugemisele tema hävimine mälus ja seetõttu tuleb informatsioon pärast iga lugemist taastada. Selleks antakse punkti *E* pingeimpulss ja pärast seda, kui mahtuvus *C* on omandanud vajaliku pinge, sule-



Joonis 38

takse mälurakk dioodidele vastupinge andmise teel. Vastupinge on dioodide kindla sulgumise tagamiseks valitud kahekordse tagavaraga, mis on vajalik selleks, et pingemuutused juhtmes *E* ei mõjutaks salvestatavat informatsiooni.

Dioodides esineva tagasisuunalise voolu tõttu laheneb mahtuvus aja jooksul ning enne informatsiooni täielikku hävimist tuleb see uuesti taastada. Selleks antakse kõikidele aadressidele kordamööda perioodiliselt käsk lugeda neis sisalduvat informatsiooni. Saadud signaal võimendatakse vastava (pinge polaarsusele tundliku) võimendaja poolt ja kirjutatakse tagasi vanale kohale.

Kondensaator-dioodmälu eeliseks on pidev võimalus automaatselt kontrollida mälurakkude töökindlust. Kontrollimisel tehakse kindlaks lugemisel saadava signaali tugevus ja juhul kui laeng kondensaatoril on langenud näiteks alla 30% normaalsest, ilmub hoiatav signaal, et lähemal ajal on selle raku tõttu oodata viga.

Kondensaator-dioodmälu osutub efektiivseks vaid siis, kui üht ning sama võimendajat kasutatakse informatsiooni salvestamiseks ja taastamiseks mitmes rakus (joonis 38), mis vähendab ka vajalike detailide hulka. Joonisel 38 on liinid *A* ja *D* ühe pesa kõikide kohtade jaoks ühised, võimendaja aga on ühine kõikide pesade ühe ja sama numbrikoha tarvis. Ühe arvu lugemine (ja salvestamine) ei mõjuta teistes pesades asuvat informatsiooni, sest arvu lugemisel on kõik teised pesad suletud.

Harilike germaaniumdioodide puhul võib üks võimendaja juhtida 32—64 pesa. Eriliste paljukihiliste dioodide kasutamine võimaldab tõsta ühe võimendaja poolt teenindatavate mälukohtade arvu isegi kümnele tuhandele. Kondensaator-dioodmälus on otstarbekas kasutada täisnurkse hüstereesikõveraga mahtuvusi, sest sel juhul otsevoolu ja tagasivoolu suhe ei mõjuta regenereerimise aega.

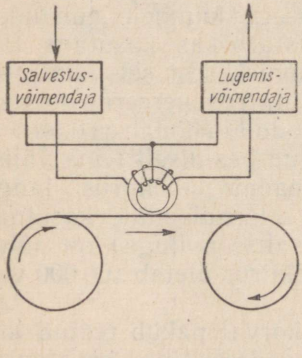
1000 pesaga kondensaator-dioodmälu nõuab ca 0,6—1 m^3 ruumi. Mälu töökiirus on määratud peamiselt tema väliste elementide omadustega. Mäluseadme töökiirus langeb aga ka suure sisemahtuvusega seleendioodide kasutamisel. Senitehtud katsetused (8-pesalise mäluga) on näidanud, et võimalik maksimaalne töökiirus ületab 100 000 valikut sekundis.

Äsjakirjeldatud mälutüüpide kõrval pakub teatud huvi veel dioodidel ehitatud mõne-aadressiline konstantide mälu, mida kasutatakse peamiselt elektron-arvutusmasinate häälestamisel. Mälus salvestatud informatsiooni muutused teostatakse mehaanilise kommutatsiooni teel.

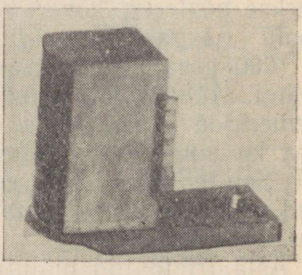
Omapärane konstantide mälu on välja töötatud NSV Liidu Teaduste Akadeemia Elektrilise Modelleerimise Laboratooriumis Teaduste Akadeemia informatsioonimasina jaoks. Üksikud mäluosad kujutavad endast õhukesi paberilehti, mille mõlemale küljele on hõbedaga või alumiiniumiga kantud informatsioon. Informatsiooni lugemine toimub dioodmaatriksite abil, milleks läbi kogu mälulehtede paki on asetatud metallvardad. Sellest väga kompaktsest mäluseadmest saab informatsiooni (ilma seda hävitamata) valida kiirusega 10 000 korda sekundis. Kogu informatsioon võib niisuguses mäluseadmes säilida aastate kaupa. Informatsiooni muutmine on võimalik vaid ühe informatsioonilehe mehaanilise kõrvaldamise ja uuega asendamise teel.

Magnetilised mäluseadmed on elektron-arvutusmasinate antud arenguperioodil kõige levinumaks informatsiooni salvestamise vahendiks. Konstruktsioonilt ning tööpõhimõttelt jagunevad magnetsalvestusel põhinevad mäluseadmed kahte põhiliiki: liikuva ja liikumata magnetkandjaga. Liikuva magnetkandjaga salvestusseadmete kõige iseloomulikumaks esindajaks on magnetlindid (koos vastavate kirjutavate ja lugevate helipeadega), liikumatu magnetkandjaga seadmete esindajaks aga ferriitidel ehitatud mälu.

M a g n e t l i n t m ä l u (joonisel 39 on antud põhimõtteskeem, joonisel 40 aga elektron-arvutusmasina

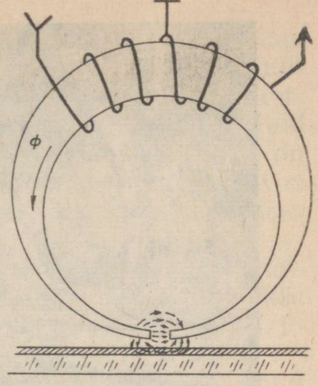


Joonis 39



Joonis 40

«Ural» magnetlintmälu peadeblokk) konstruktsioon on sarnane hariliku magnetofoniga. Samuti nagu magnetofonis, antakse salvestamisele kuuluv informatsioon ka siin pärast võimendamist helipeasse (joonis 40). Pinge helipea klemmidel kutsub esile voolu mähises, mis omakorda tekitab magnetvoo helipea magnetjuhtmes (joonis 41). Magnetjuhe omab harilikult toroidi kuju, millesse on lõigatud radiaalne pilu. Selle pilu kohal laieneb magnetvoog ruumiliselt ja läbib osalt ka pilu lähedusse asetatud magnetkandjat.

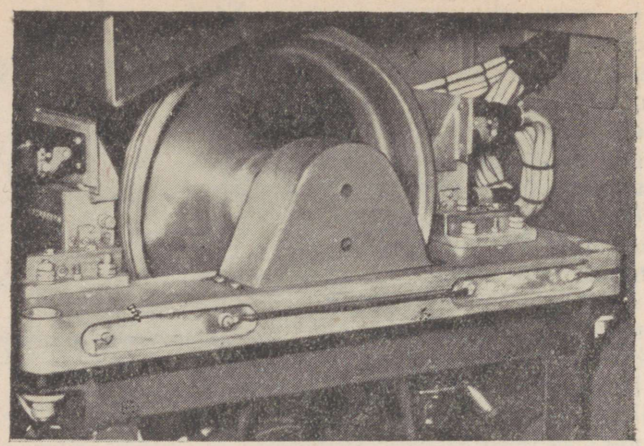


Joonis 41

Magnetkandjana kasutatakse tavaliselt tugevate jääkmagnetiliste omadustega materjalist kattega painduvat linti. Niisugused materjalid säilitavad neis tekitatud magnetilist olukorda pika aja vältel. Kui lint möödub helipeast, milles on tekitatud ajas muutuv magnetvoog, siis lindile kantud kihi erinevad osad magneetuvad erinevalt. Kui lasta see lint uuesti mööduda helipeast, tekib helipea klemmidel pinge, mis on võrdeline magnetvälja tugevuse muutumisega pilu eest mööduval lindil ning võimaldab lugeda salvestatud informatsiooni.

Seega ei erine kogu salvestamise ja lugemise protsess põhimõtteliselt harilikus magnetofonis toimuvast protsessist, küll on aga nendes terve rida konstruktiivseid erinevusi. Erinevused ilmnevad kõigepealt helipea mõõtmetes, sest elektron-arvutusmasinates kasutatavad helipead on gabariitidelt tunduvalt väiksemad kui helisalvestuses. Selline vahe on tingitud erinevatest töösagedustest: helisalvestamisel haarab sagedusriba piirkonna 20 Hz — 20 kHz, elektron-arvutusmasinates aga piirkonna 100 kHz — 10 MHz. Niivõrd kõrgete sageduste salvestamiseks tuleb helipea läbimõõtu vähendada 40 millimeetrilt 5 millimeetrini ning mähiskeerdude arvu kolmesajalt (ja enamalt) 5—6 keeruni. Elektron-arvutusmasinas kasutatavas helipeas on ühele ja samale magnetjuhtmele keritud nii salvestus- kui ka lugemismähised (joonis 39).

Olenevalt sellest, kas vaadeldavas arvutusmasinas



Joonis 42

kasutatakse informatsiooni üheaegset või järjestikust ülekandmist aritmeetilisest seadmest mälusse, töötab ühekorraga 1 kuni 46 helipead. Sellest oleneb ka magnetlindi laius ja kõik teised konstruktiivsed elemendid.

Pika lindi pealt vajaliku aadressi leidmine on aeganõudev töö. Seetõttu kasutatakse magnetlinte vaid välises mälus, kuna aga sisemiseks mäluks kasutamise korral asendab linti lai, magnetilise kihiga kaetud trummel (joonisel 42 on kujutatud arvutusmasina «Ural» magnettrummel). Trummel pöörleb kiirusega 3000 (M-3) või 6000 («Ural») pööret minutis, millega iga pesa koos temasse salvestatud informatsiooniga möödub helipeade alt 50 või 100 korda sekundis. Niisugune väike sagedus põhjustab vastavat tüüpi masinate väikese töökiiruse (nn. optimaalse programmeerimise abil on mälu poole pöördumise tihedust sundjärjekorras käske täitva masina korral võimalik tõsta enam kui kahekordseks — vt. § 4).

Elektron-arvutusmasina M-3 puhul magneeditakse kogu trumli ferromagnetiline kiht vastava kustutuspea abil algul suunas, mis vastab numbrile null. Vajaliku pesa kohalejõudmise hetkel antakse kõikidesse helipeadesse salvestamisele kuuluv informatsioon. Juhul kui antud kohale tuleb salvestada number üks, siis vastava helipea magnetpilu ümber tekib magnetvoog, mille suund on vastupidine eelmagneetimisel loodud magnetvoo suunale ning trumli pinnaelement magneetub ümber. Nulli salvestamisel mingit ümbermagneetimist ei toimu. Informatsiooni lugemisel tekivad helipeas vooluimpulsid vaid nendel hetkedel, kui pilu alt möödub pesakoht, millesse on salvestatud üks, sest ülejäänud ajal mingeid magnetvoo muudatusi ei toimu ning seetõttu ei indutseerita mähises ka elektromotoorset jõudu.

Vajaliku aadressi kindlaksmääramiseks on informatsiooni kandva ferromagnetilise kihi äärele kantud kas mehaaniliselt või elektrilisel teel nn. m a r k e r i m p u l - s i d.

Üheks käesoleval ajal kõige perspektiivsemaks mäluseadmete liigiks on liikumatu magnetkandjaga süsteemid, milledes kasutatakse täisnurkse hüstereesisilmusega materjale. Neist valmistatakse ferriittoroide, kahe või kolme auguga kettaid (transfluksoreid) ja teisi seadiseid, mis võimaldavad salvestada informatsiooni kahe erineva magnetilise seisundina.

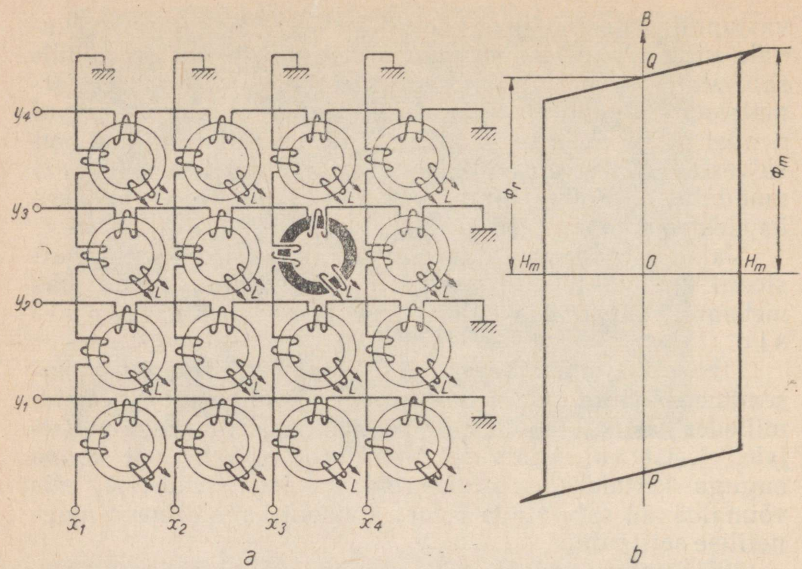
Mäluseadme tööpõhimõte sarnaneb ülalkirjeldatud ferrodielektrilise kondensaatormälu (joonis 36) omaga, sest ühel juhul kasutatakse ferromagneetiku, teisel ferrodielektriku füüsikalise olukorra muutmise analoogilisi seaduspärasusi.

Ferriitsüdamiku täielikuks ümbermagneetimiseks on tarvis luua magnetväli teatud tugevusega H_m (joonis 43, *b*), kusjuures välja väiksema tugevuse korral südamik ei magneetu ümber isegi paljukordsel mõjutamisel. Näiteks kui südamiku magnetilist olukorda iseloomustab punkt P, siis ümbermagneetimiseks tuleb luua magnetväli, mille tugevus on vähemalt H_m, ja vastupidi. Kui südamiku olukord iseloomustub punktiga Q, siis tema ümbermagneetimiseks on vajalik väljatugevus $-H_m$.

Tehnilise teostamise lihtsustamiseks paigutatakse kõik südamikud mälus korrapäraste nelinurkadena. Selline 4×4 südamikku sisaldav süsteem on kujutatud joonisel 43, *a* ja 5×5 südamikku sisaldav süsteem[1] joonisel 44. Arvu salvestamiseks antakse mööda vastavaid liine X ja Y vooluimpulsid, mis kutsuvad toroidis esile magnetväljad H_x ja H_y.

---

[1] Selliselt korraldatud ferriitidesüsteeme nimetatakse tavaliselt m a a t r i k s i t e k s.



Joonis 43

Juhul kui H_x ja H_y on ühekaupa liialt väikesed piirtugevuse H_m tekitamiseks, koos aga selleks küllaldased, siis nende voolude ristumiskohal asuv ferriit (joonisel 43, *a* ferriit x_3y_3) magneetub ümber, kuna aga kõik teised ferriidid säilitavad oma endise seisundi. Ferriidi ümberlülitamisel tekib mähises L pinge. Samuti nagu elektrostaatilise mälu puhul potentsialoskoopidel, kujutab iga ferriitmaatriks siin üht kindlat kohta mälu kõikides pesades. See võimaldab anda liinidesse X ja Y lugemisvoolu impulsi ühekorraga, s. t. lülitada need liinid paralleelselt.

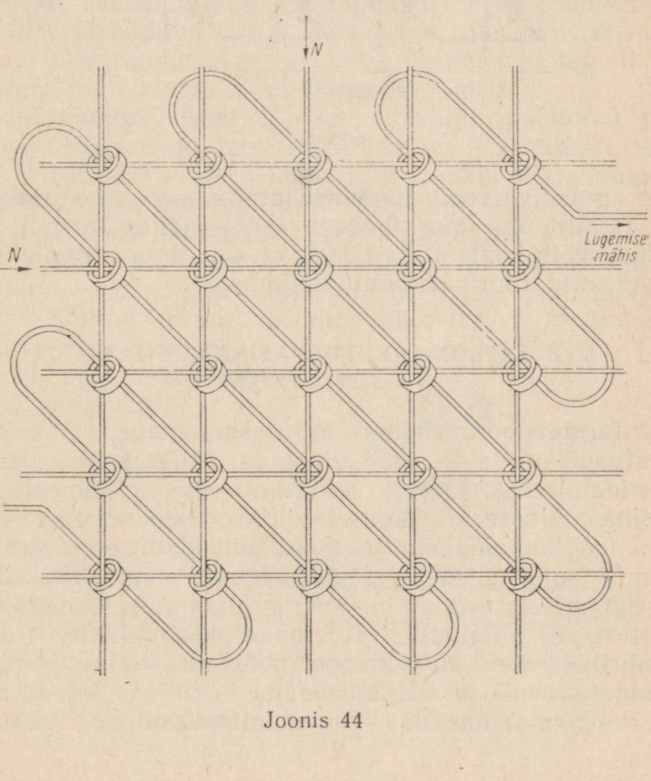
Lugemisel informatsioon hävineb ning vastav automaatne seade peab selle uuesti salvestama. Salvestamise ajal antakse liinidesse X ja Y vooluimpulsid, mis kutsuvad kumbki esile magnetvälja tugevusega $^1/_2\, H_m$. Ümbermagneetimise vältimiseks nulli salvestamisel antakse mähise L kaudu negatiivne impulss.

Kuna südamike hüstereesisilmused pole ideaalselt nelinurksed (suhe $\Phi_r/\Phi_m < 1$), siis tekitab ka kriitilisest nõrgem vool teatud muutusi südamikes, mida ta läbib. Kuigi mähis L on ühendatud ühe ja sama rea või veeru ferriitide suhtes vastassuundades ning parasiitimpulsid peaksid üksteist teoreetiliselt kompenseerima, ei toimu seda tegelikult ferriittoroidide mittetäieliku identsuse ja muude nähtuste tõttu siiski. Informatsiooni edasikandev impulss maskeeritakse seega parasiitimpulsside poolt kas osaliselt või täielikult.

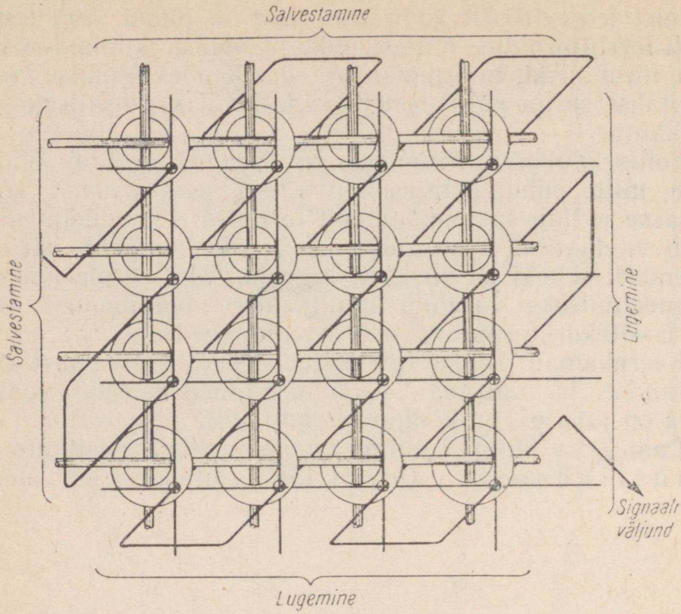
Sellise olukorra vältimiseks on välja töötatud nn. *Z*-süsteem, mille puhul maatriksi ühe rea peal asuvad kõik samasse mälupesasse kuuluvad ferriidid. Lugemisimpulss läbib vaid antud rea südamikke ning mingeid häireid lugemisel ei teki ka siis, kui lugemist forsseerida ümbermagneetimiseks vajaliku minimaalse vooluimpulsi asemel 5—10 korda suurema impulsi andmise teel.

Keerukamale kommutatsioonisüsteemile (eriti just salvestamise ja informatsiooni taastamise käigus) vaatamata on juba peagu kõikjal üle mindud *Z*-süsteemile.

Laia ketta kujulisi mitme avaga ferromagneetikute — t r a n s f l u k s o r i t e (joonis 45) kasutamisel on võima-



Joonis 44



Joonis 45

lik magneetiva voolu tugevuse muutmisega saavutada osalist (ainult ühe ava ümber) ümbermagneetumist, mille suunda võib kontrollida teise ava abil ilma transfluksorile salvestatud informatsiooni hävitamata.

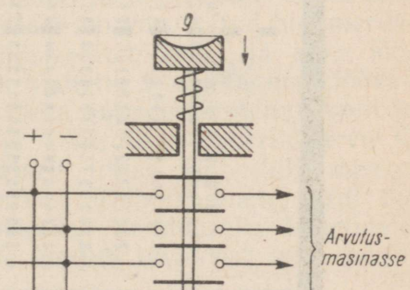
8. ELEKTRON-ARVUTUSMASINATE SISEND- JA VÄLJUNDSEADMED

Informatsioonivahetus arvutusmasinaga. Selleks et arvutusmasin saaks tööd alustada, tuleb kogu ülesande lahendamiseks vajalik informatsioon kõigepealt viia masina mälusse (niisuguseks informatsiooniks on eeskätt ülesande lähteandmed ja tema lahendamise programm). Samuti tuleb arvutuse lõpptulemused nende praktiliseks rakendamiseks esitada loetaval kujul, s. t. teisendada elektrilistest või magnetilistest signaalidest näiteks trükitud numbriteks. Neid funktsioone täidavadki elektron-arvutusmasina sisend- ja väljundseadmed, millede kaudu seega teostub masina ühendus «välismaailmaga».

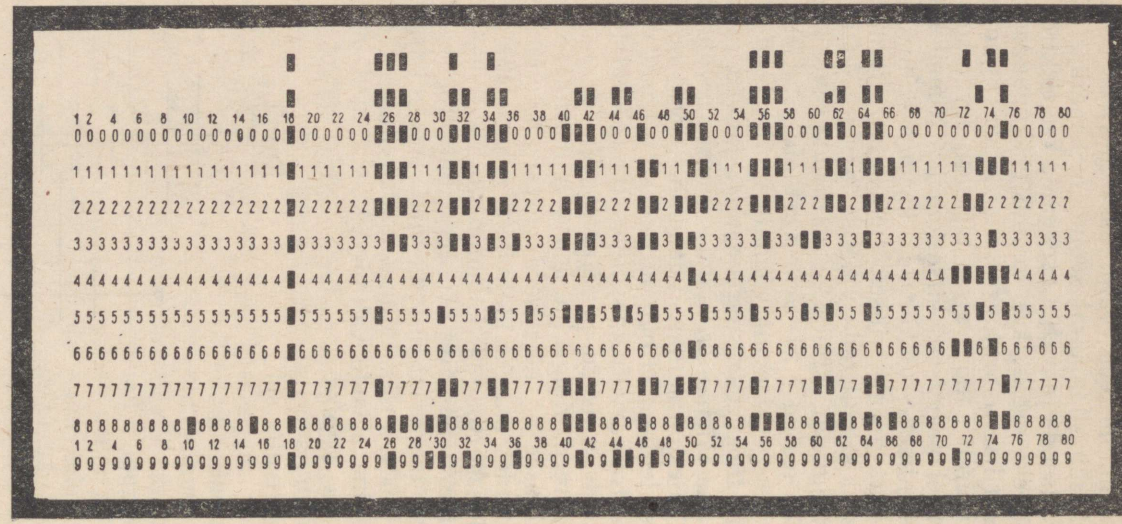
Trükitud arvude teisendamine elektrilisteks signaalideks (sisendseadmes) ja elektriliste signaalide teisendamine trükitud numbriteks (väljundseadmes) on paratamatult vähemalt mingil määral seotud mehaanilise liikumisega. Seega sisend- ja väljundseadmed on elektron-arvutusmasina selleks osaks, kus toimub kokkupuude mehaanilise ja elektrilise aparatuuri vahel. See asjaolu ongi tänapäeva kiirelt töötavate elektron-arvutusmasinate konstrueerimisel peamiseks kitsaskohaks. Nimelt piiravad elektroonikaseadmete töökiiruse kasvu põhimõtteliselt ainult masina geomeetrilised mõõted: informatsiooni leviku kiirus ei saa ületada valguse kiirust. Mehaaniliste seadmete töötamiskiirust ei saa aga nende suure inertsuse tõttu tõsta üle teatud piiri. Selle vastuolu osaliseks ületamiseks otsitaksegi pidevalt võimalusi nii sisend- ja väljundseadmete töökiiruse suurendamise kui ka nende töö mehaanilise osa täieliku eraldamise suunas arvutusmasina enda tööst. Täiesti rahuldavat lahendust need probleemid aga seni veel leidnud ei ole. Järgnevas tulevad vaatlemisele vaid elektron-arvutusmasinates tänapäeval tegelikult kasutamist leidvad sisend- ja väljundseadmed.

Sisendseadmed. Kõige lihtsam viis informatsiooni viimiseks masinasse on teha seda vastava klaviatuuri abil käsitsi. Vajutades näiteks klahvile *9* (joonis 46) saadame arvutusmasinasse pingete kombinatsiooni +——+. Kui masin dešifreerib positiivset pinget numbrina 1 ja negatiivset pinget numbrina 0, siis viiakse masinasse seega kahendarv 1001, mis kümnendsüsteemis tähendabki arvu 9. Selline arvude valimine klaviatuuril on ilmselt aegavõttev ja kulukas (kuna ju arvutusmasin sel ajal tegelikult seisab).

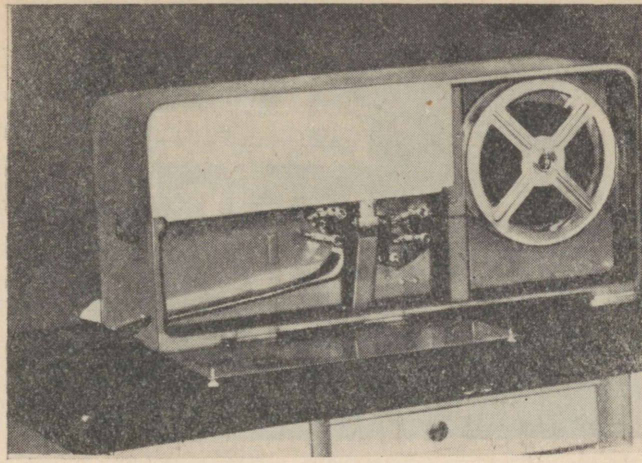
Töö kiirendamiseks jaotatakse arvude masinasse viimine tavaliselt kahte etappi. Esimesel etapil viiakse arvud mitte kohe masinasse, vaid kantakse nad näiteks perfokaartidele (joonis 47) aukude süsteemide



Joonis 46



Joonis 47

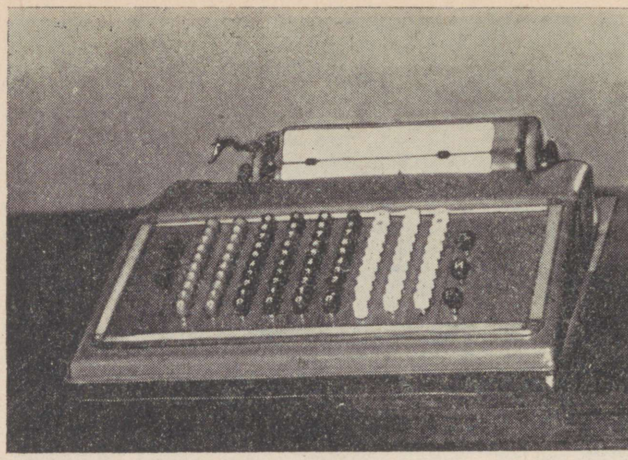


Joonis 48 a.

näol põhimõtteliselt samal kujul, nagu nad hiljem pesas säilitamisele tulevad. Numbreid kujutavaks «elemendiks» on nüüd koht perfokaardil, kus augu olemasolu tähendab numbrit 1 ja augu puudumine numbrit 0. Iga arvu jaoks on perfokaardil ette nähtud nii mitme kohaga rida, kui palju kohti on vastava masina mälu pesas (kusjuures ka kohtade jaotus on sama). Joonisel 47 on kujutatud nõukogude elektron-arvutusmasinas «Strela» kasutatav perfokaart, millele saab kanda 12 arvu.

Arvude kandmine perfokaartidele toimub eriseadme, nn. perforaatori abil. See seade on varustatud nummerdatud klaviatuuriga, trükkimismehhanismiga ja auke löövate stantsidega. Näiteks klahvile *9* vajutamisel löövad stantsid perfokaardisse vastava aukude kombinatsiooni ja trükkimismehhanism trükib kontrolliks paberile arvu 9. Sealjuures saab perforaatorit lihtsa ümberlülitamisega kasutada nii kümnendarvude (lähteandmete) kui ka kahendarvude (eeskätt käskude) kandmiseks kaartidele. Ühe arvutusmasina juurde kuulub tavaliselt mitu perforaatorit.

Kümnendarvude puhul on perfokaardi reas iga numbri jaoks ette nähtud neli kohta, sest kümnendnumber kujutatakse ju neljakohalise kahendarvuna. Kümnendnumbrite



Joonis 48 b.

teisendamise neljakohalisteks kahendarvudeks teostab perforaator automaatselt, sobiva kombinatsiooni valimise teel. Klaviatuurile on ikka märgitud kümnendnumbrid. Kui tähistada valge ristkülikuga augu puudumist ja musta ristkülikuga augu olemasolu, siis kümnendnumbrid näevad perfokaardil välja järgmiselt:

0 = ▯▯▯▯	5 = ▯▮▯▮
1 = ▯▯▯▮	6 = ▯▮▮▯
2 = ▯▯▮▯	7 = ▯▮▮▮
3 = ▯▯▮▮	8 = ▮▯▯▯
4 = ▯▮▯▯	9 = ▮▯▯▮

Perforeeritavad arvud peavad enne muidugi olema kirjutatud plangile sellisel kujul, nagu nad vastavas arvutusmasinas säilitamisele tulevad, s. t. fikseeritud komaga masina korral lihtmurdudena ja liikuva komaga masina korral poollogaritmilisel kujul. Näiteks arvutusmasina «Strela» jaoks tuleb arv

$$-5436{,}13872 = -0{,}543613872 \cdot 10^4$$

kirjutada plangile kujul

$$-543\,613\,872+04.$$

Masinasse viidavad kahendarvud kirjutatakse plangile enamasti kaheksandsüsteemis. Kui selline «kahendarv» on tegelikult käsk, siis kirjutatakse ta plangile tavalisel kujul (arvutusmasina «Strela» korral tingimata koos kontrollmärgiga). Kui aga on tarvis masinasse sisse viia ka n. ö. päris kahendarve, siis tuleb nende kirjutamisel enamasti ka arvu märk lugeda numbrikohaks. Näiteks arvutusmasina «Strela» puhul kirjutatakse kahendarv

$$-0{,}10111011000011011010101101110101101 \cdot 10^{10110}$$

plangile kujul

$$6730\ 3325\ 5655+26.$$

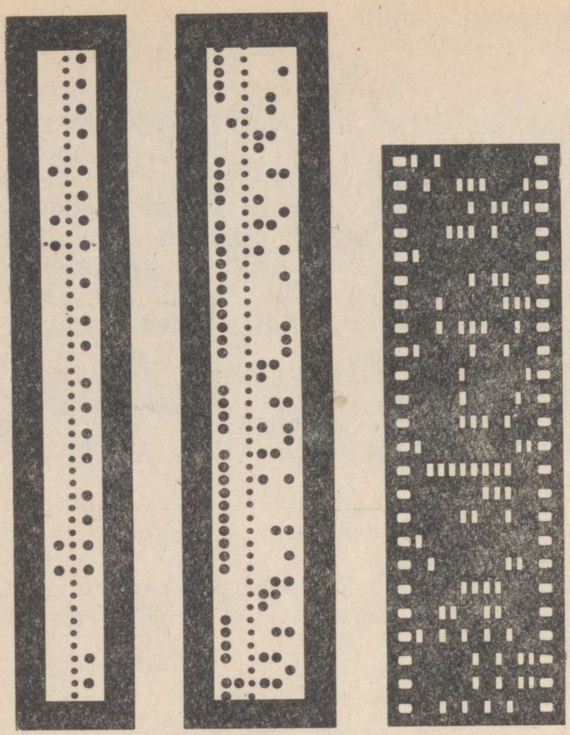
Niisuguse moonutatud kirjutusviisi kasutamine ei tekita erilisi raskusi, sest kahendarve on suhteliselt väga harva tarvis masinasse viia.

Iga kaheksandnumbri teisendab perforaator automaatselt kolmekohaliseks kahendarvuks ja lööb selle perfokaardile järgmiselt:

0=□□□	4=■□□
1=□□■	5=■□■
2=□■□	6=■■□
3=□■■	7=■■■

Arvude masinasse viimise teisel etapil asetatakse perforeeritud kaardid pakina vastavasse perfokaart-sisendseadmesse. Sisendseade eraldab vaakuumi abil paki pealmise kaardi, mis läbib kompamisseadme ja langeb kombatud kaartide paki peale. Kompamisseade saadab augu esinemise puhul mehaanilise kontaktsüsteemi kaudu arvutusmasinasse impulsi, mis salvestatakse mälus numbrina 1. Üheaegselt kuulub kompamisele kas üks rida või terve perfokaart.

Perfokaartide asemel kasutatakse paljudes elektronarvutusmasinates ka perfolinti, millel arvud kujuta-

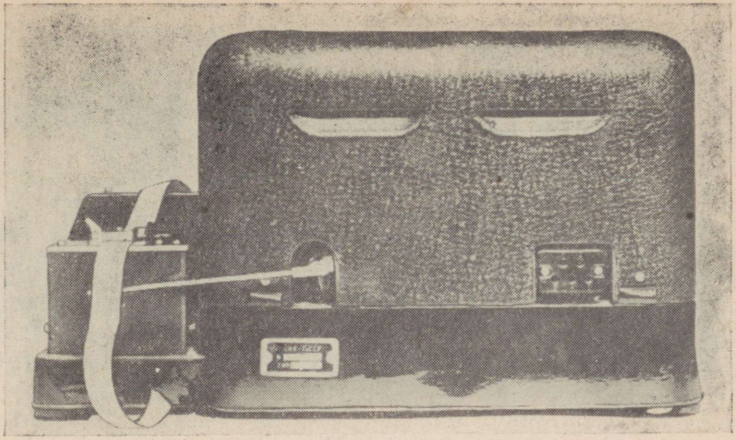


a b c

Joonis 49

takse samuti mulkude süsteemidena. Arvude kandmine lindile toimub põhimõtteliselt samasuguse perforaatori abil nagu perfokaartide korral. Joonisel 48, *a* on kujutatud elektron-arvutusmasina «Ural» perforaator ja joonisel 48, *b* klaviatuur koos trükkimismehhanismiga.

Pärast perforeerimist on lindis mitu rida mulke. Nendest üks või kaks rida on ette nähtud lindi edasinihutamiseks (nagu 2 rida mulke filmilindis; arvutusmasinas «Ural» lüüaksegi mulgud mustale filmilindile), ülejäänud mulguread (nn. kanalid) sisaldavad aga masinasseviidava informatsiooni. Joonisel 49, *a* on näidatud kahekanaliline perfolint (kasutatakse elektron-arvutusmasinas БЭСМ), joonisel 49, *b* viiekanaliline lint (kasutatakse elektron-arvutusmasinates M-2 ja M-3) ning joonisel 49, *c* ühe-

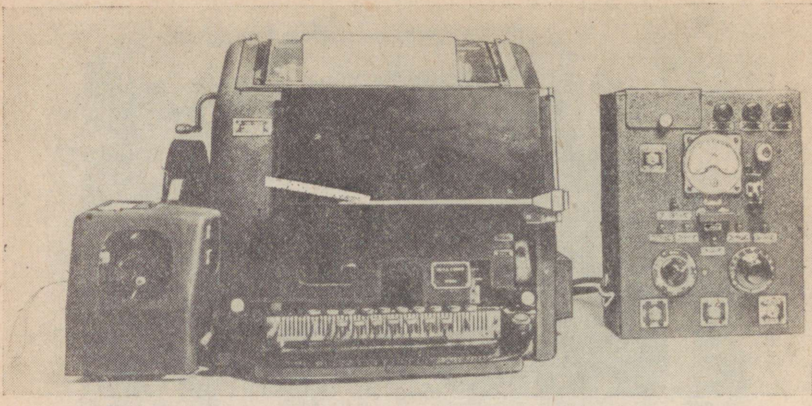


Joonis 50

teistkanaliline lint (kasutatakse elektron-arvutusmasinas «Ural»). Igale numbrile vastab lindil kindel mulkude kombinatsioon. Viiekanaliline lint võimaldab märkida $2^5=32$ erinevat kombinatsiooni, kahekanaliline aga $2^2=4$ erinevat kombinatsiooni. Esimesel juhul on kombinatsioonide arv (32) küllaldane kümne numbri, kahe märgi (+ ja —) ning sisendoperatsioone juhtivate käskude (aadressi lõppu tähistav märk, arvu lõppu tähistav märk, sisendoperatsiooni lõpetamise käsk jt.) tähistamiseks; teisel juhul tuleb üks sümbol moodustada mitmest üksteisele järgnevast kombinatsioonist.

Arvutusmasinas «Ural» kasutatava perfolindi (selleks on tavaline 35-millimeetrine kinolint) puhul on esimene kanal määratud arve eraldavate markerite, teine kanal arvu märgi ning ülejäänud üheksa kanalit üheksakohalise arvu perforeerimiseks. Joonisel 49, *c* toodud perfolindile on löödud arvud:

$$
\begin{array}{r}
-0{,}815743206 \\
+0{,}301514632 \\
+0{,}007910924 \\
-0{,}445566788 \\
-0{,}112233440 \\
+0{,}909696966
\end{array}
$$



Joonis 51

Perfolindile kantud informatsiooni masinasse viimiseks võib kasutada tavalist telegraafitransmitterit (joonis 50). See koosneb elektrimootoriga käivitatavast veomehhanismist (lindi edasiviimiseks) ja mehaanilisest kompamisseadmest (samuti nagu perfokaart-lugemisseade). Kuna mehaaniline kontaktsüsteem on inertne (töökiirus 6—10 märki sekundis), siis on hoopis sobivam kasutada fototransmitterit, milles kompava mehhanismi kontaktsüsteem on asendatud valgusallika ja fotoelementidega (nii on see näiteks arvutusmasinas «Ural»). Kui valguskiir läbib lindis oleva mulgu, satub ta fotoelemendile, mis muudab selle elektriliseks impulsiks. See süsteem on vaba mehaanilise inertsi mõjust ja sõltuvalt lindi mehaanilisest tugevusest võib tema liikumise kiirust tõsta mõnele meetrile sekundis.

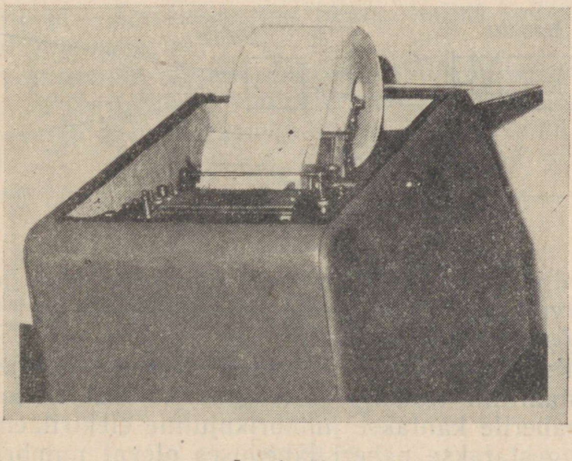
Kõige moodsamaks sisendseadmeks on spetsiaalne magnetofonikomplekt, mis koosneb kahest osast: magnetilisest trükkijast ning tavalisest magnetofonist. Magnetiline trükkija töötab analoogiliselt perforaatoriga, ainult mulgu läbilöömise asemel magneeditakse vastav koht lindil. Hiljem linti läbi magnetofoni lastes indutseerib iga magneeditud punkt elektrilise impulsi, mis võimendatult saadetakse masina mälusse. Selline sisendseade on varemkirjeldatutega võrreldes tunduvalt suurema töötamiskiirusega.

Väljundseadmed. Väljundseadmete konstrueerimisel on inseneridel tulnud lahendada põhiliselt samad probleemid mis sisendseadmetegi puhul, ainult teatud mõttes vastupidises suunas.

Elektriimpulsside muutmist trükitud kirjamärgiks teostab üldiselt tuntud viisil telegraafi-teletaip, kus trükkimine toimub pikale paberiribale. Et vabaneda tülikast pabeririba kleepimisest plangile, kasutatakse elektronarvutusmasinates enamasti teletaipe PTA või PTM (vt. joonis 51), mis trükivad arvud tulpadena paberilehele (keerates paberit automaatselt edasi ja alustades vajaduse korral uut rida). Kuid üle 8 märgi sekundis see aparaat trükkida ei suuda. Arvutusmasinas «Ural» kasutatav trükkimisseade (joonis 52) trükib 100 üheksakohalist arvu minutis.

Märksa kiirem on elektromehaaniline kiirtrükkija (joonis 53). See koosneb vajadusele vastava pikkusega võllist, mis on kaetud tähe- ja numbrikujunditega. Võll pöörleb kiirusega kuni 3000 pööret minutis.

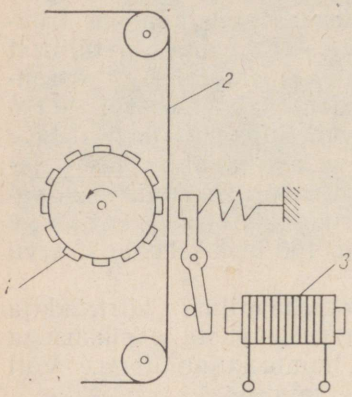
Numbrivõlli *1* läheduses on paber *2*. Kui soovime trükkida näiteks arvu 7, siis ergutame solenoidi *3* hetkel, mil võllil olev märk 7 on pöördunud paberi kohale. Haamer annab hoobi ja paber surutakse (väga lühiajaliselt) vastu trükimustaga kaetud numbrimärki. Kui kogu arv on trükitud, liigub paber ühe rea võrra edasi. Kuigi selline trükkija võib põhimõtteliselt trükkida kuni 50 rida sekun-



Joonis 52

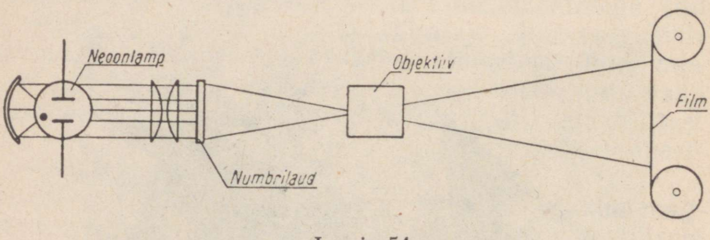
dis, ei ületa praktikas kasutatavate seadmete kiirus siiski 15—25 rida sekundis.

Veelgi kiiremaks väljundseadmeks on fototrükkija (joonis 54). Kogu süsteem koosneb siin neoonlambist ja projektorist. Projektor projitseerib numbrimärgi filmilindile. Omades kümme projektorit iga kümnendkoha ning kaks projektorit märkide + ja — kujutamiseks, saab vastavatele neoonlampidele elektrilisi impulsse andes üheaegseit projitseerida filmilindile terve arvu koos märgiga. Tundliku filmilindi puhul võib kasutada väga lühiajalisi impulsse, mis võimaldab trükkida kuni 200 arvu sekundis. Hiljem tuleb film tavalisel viisil ilmutada ja kinnitada. Kirjeldatud seadme põhiliseks puuduseks on tema kogukus, näiteks 10-kohalise arvu ja märgi trükkimiseks peab ta sisaldama 102 projektorit.



Joonis 53

Viimasel ajal on väljundseadmetes hakatud kasutama



Joonis 54

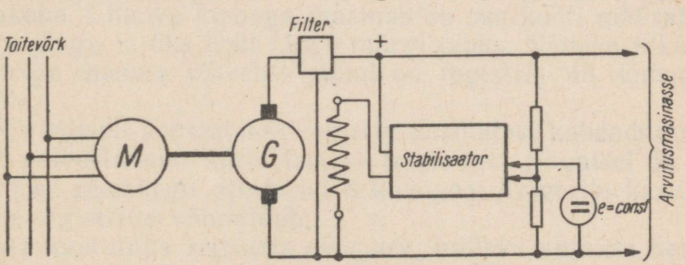
elektriväljale tundlikku paberit, mis võimaldab loobuda trükivärvist või fototehnilistest seadmetest ja sellega veelgi tõsta kiirust.

Üheks kiireimaks senituntud väljundseadmeks on elektrostaatiline trükkija (kserograaf). Elektrit mittejuhtivale paberile kantakse numbrikujuline elektrilaeng. Selleks pingestatakse paberi läheduses olevat numbrimärgikujulist elektroodi lühiaegselt kõrgepingelise impulsiga. Numbri nähtavaks tegemiseks juhitakse paberile peeneteralist värvaine tolmu. Elektrilaengu kohtadel jääb tolm paberile püsima, moodustades selgeltloetava teksti. Et tolm laengu kadumisel maha ei variseks, selleks kuumutatakse paberit. Tolm sulab ja vedel värvaine imbub paberisse. Kuna sellise konstruktsiooni puhul puuduvad igasugused haamrid ja muud mehaanilised seadmed, siis on tema töötamiskiirus aukartustäratav: ühe minutiga võib trükkida umbes 300-leheküljelise raamatu.

Elektron-arvutusmasinate toiteseadmed ei kuulu küll sisend- või väljundseadmete hulka, kuid on siiski teatud määral vahendajaks masina ja «välismaailma» vahel.

Suure hulga üksteisest sõltuvate lülitusskeemide normaalse koostöö kindlustamine arvutusmasinas on küllaltki raske probleem. See seab suured nõuded iga sõlme sisend- ja väljundpinge vastavusele teiste sõlmedega. Kuigi üksikud skeemid on suutelised töötama väga laiades piirides muutuvate toitepingete puhul, võivad paljude skeemide koostöö puhul pingete kõrvalekalletest tekitatud häired summeeruda, mille tagajärjeks on arvutusviga. Samuti võivad toitevõrgust arvutusmasinasse pääseda pinge järskudest muudatustest tekitatud impulsid, mis häirivad tööd ja tekitavad juhuslikke vigu. Nende puuduste vältimiseks eraldatakse elektron-arvutusmasin toitevõrgust näiteks mootor-generaatori abil, mille mehaaniline inerts tasandab järsud tõuked. Pingete stabiilsuse tagavad elektron-stabilisaatorid. Arvutusmasina põhimõtteline toiteskeem on kujutatud joonisel 55.

Arvutusmasin vajab elektronlampide toiteks küttepinget ja mitmesuguseid stabiliseeritud alalispingenivoosid; üldse esineb arvutusmasinas sageli 30 ja enam pingeni-



Joonis 55

vood, mida tuleb hoida täpsusega ±0,5%. Erandiks on vaid küttepinge, mida (lampide eluea pikendamiseks) hoitakse tavaliselt kuni 8% alla nominaalse ning stabiliseeritakse täpsusega ±3%. Üksiku pinge kõrvalekaldumisel juba 5% võrra (välja arvatud küttepinge) lakkab masin tavaliselt töötamast.

Arvutusmasinas muutub kogu tarbitav võimsus (see võib ulatuda 100 kW-ni ja enamgi) soojuseks, mis tuleb vastavate ventilaatorite abil eemaldada. Suured eralduvad soojushulgad teevad arvutuskeskuste kütmise seega tarbetuks.

IV peatükk

ELEKTRON-ARVUTUSMASINATE TÖÖTAMISE PÕHIMÕTE JA KASUTAMINE

9. ARVUTUSMASINA PÕHISEADMED JA NENDE KOOSTÖÖ

Register. Eelmises peatükis kirjeldatud üksikelementidest võib koostada keerulisemaid seadmeid ja nendest omakorda kogu arvutusmasina. Konstruktiivselt kujutab iga triger koos juurdekuuluvate klappide ja loogiliste elementidega omaette (arvutusmasinast väljavõetavat) blokki, mis on ühendatud vastava pistiku ja pesaga. Suuremad blokkide rühmad moodustavad s e a d m e (näiteks juhtimisseade, sisendseade, väljundseade, aritmeetiline seade jne.), mis tavaliselt asub ühel raamil ning on ühendatud teiste raamidega juhtmekimpude abil. Käesolevas paragrahvis kirjeldame peamiselt aritmeetilist seadet ja juhtimisseadet ning nende koostööd.

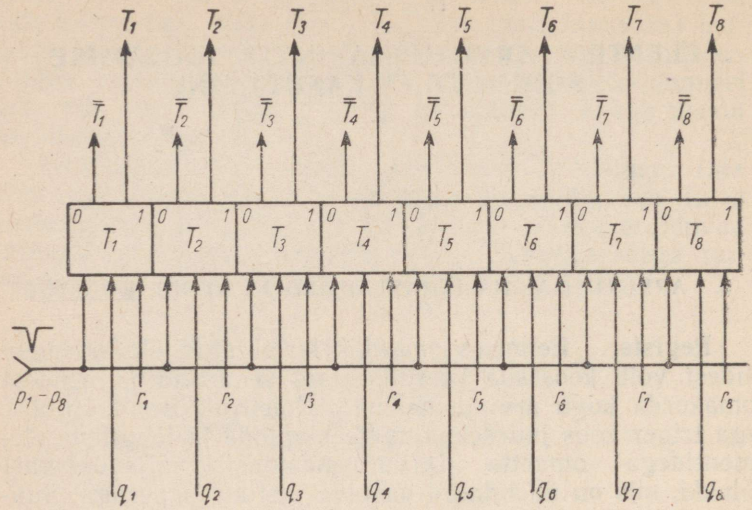
Aritmeetilise seadme üheks põhiliseks koostisosaks on r e g i s t e r — mäluseadis ühe arvu salvestamiseks. Et enamus arvutusmasinaid töötleb informatsiooni kahendsüsteemis, siis kasutatakse nendes ka kahendregistrit. Ilmselt peab n-kohalise kahendarvu salvestamiseks määratud kahendregister koosnema n elementaarlülitusest, millest igaüks salvestab ühe kahendnumbri.

Arvu märgi salvestamiseks sisaldab register veel ühe lisakoha. Liikuva komaga masinas on osa kohti määratud arvu järgu ja üks koht järgu märgi jaoks. Näiteks liikuva komaga masina «Strela» puhul on registris 43 kahendkohta.

Tavaliselt koostatakse register üksikutest kahendnumbreid salvestavaist lülitustest — trigeritest (joonisel 56 on lihtsuse eesmärgil piirdutud 8 kohaga). Trigerite kasutamine registrites võimaldab:

1) kustutada registris olev arv, milleks antakse negatiivne impulss üheaegselt kõigile sisendeile p_1—p_8;

2) viia kustutatud registrisse uus arv; selleks tuleb



Joonis 56

negatiivne impulss anda nende trigerite sisendile q või r, kuhu soovitakse salvestada number 1;

3) üheaegselt saada trigerite väljunditelt «1» salvestatud arvu numbrikohtadele vastavad pingenivood ja väljunditelt «0» nende loogilistele eitustele vastavad pingenivood;

4) muuta registris oleva arvu kõik numbrikohad nende eitusteks, milleks antakse negatiivne impulss kõigile sisendeile q_1—q_8.

Registrisse võib arve salvestada ka käsitsi, kui anda tema sisenditele r vastava lüliti abil sobivaid impulsse. Registrisse salvestatud arvu saab lugeda, kui väljundite T_1, T_2, ... , T_8 külge lülitada neoonlambikesed, mis süttivad, kui neil väljundeil on kõrge pingenivoo (süttinud lamp vastab numbrile 1). Arvutusmasina aritmeetilises seadmes on tavaliselt 2—5 registrit (tähistatakse A, B, C, . . .); neis salvestatavaid arve saab tarbe korral lugeda juhtpuldile paigutatud neoonlambikeste ridadelt. Kuna arvutusmasin teostab kuni tuhandeid operatsioone sekundis, siis vahelduvad arvud registris väga kiirelt ja neoonlampidelt on nad loetavad muidugi alles siis, kui masin on peatatud.

Arvude edasiandmine ühest registrist teise toimub esimese registri väljundite ühendamise teel teise registri sisenditega. Näide sellisest ülekandest on kujutatud joonisel 57, kus trigerid *A*1 ja *A*2 kuuluvad registrisse *A*, trigerid *B*1 ja *B*2 aga registrisse *B*.

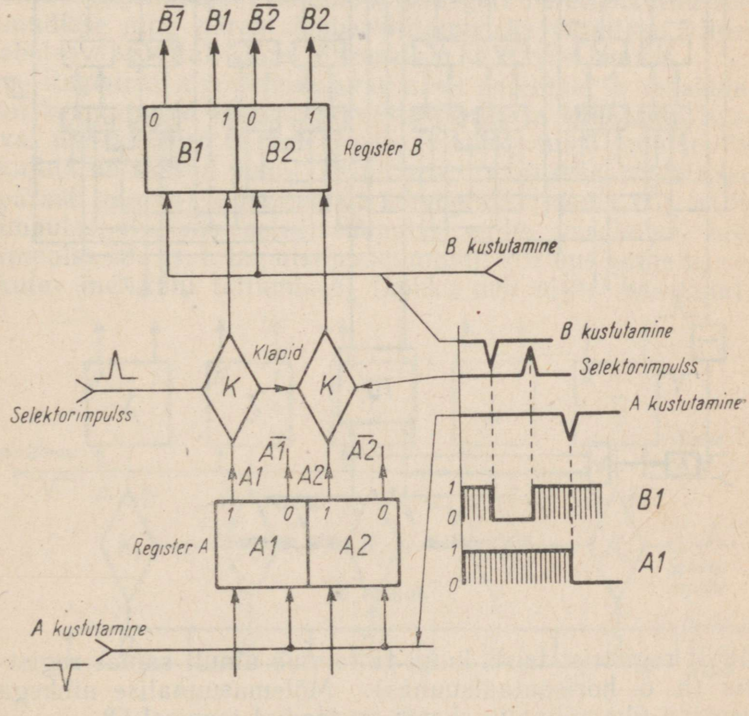
Informatsiooni ülekandmiseks registrist *A* registrisse *B* tuleb vastavatele sisenditele anda kolm impulssi (vt. graafik joonisel 57):

1) impulss registri *B* kustutamiseks.

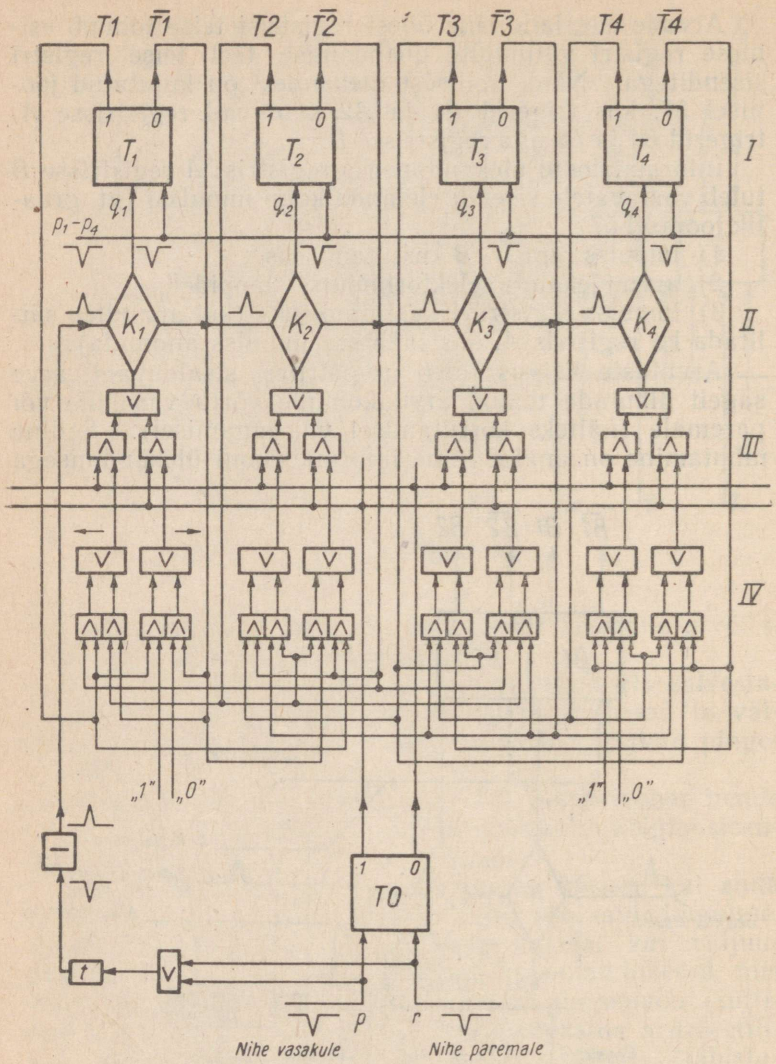
2) arvu ülekande selektorimpulss klappidele,

3) impulss registri *A* kustutamiseks (kui arv tuleb säilitada ka registris *A*, siis jääb see impulss andmata).

Arvutuste käigus tuleb registrites sisalduvaid arve sageli nihutada teatud arvu kohtade võrra vasakule või paremale (näiteks korrutamisel või jagamisel). Selline nihutamine on analoogiline informatsiooni ülekandmisega



Joonis 57



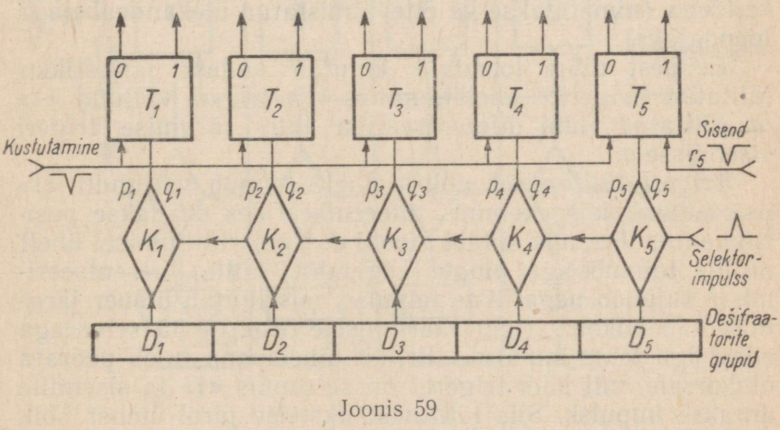
Joonis 58

ühest registrist teise, kuigi ta toimub ainult samas registris (n. ö. horisontaalsuunas). Mõlemasuunalise nihkega registri üks võimalik skeem on toodud joonisel 58.

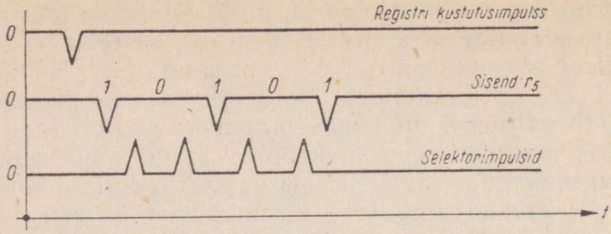
Reas *I* on registri trigerid, mida juhitakse läbi reas *II* asuvate klappide. Klappide sisenditeks on reas *III* asuvad loogilised skeemid (nn. d e š i f r a a t o r i d). Neid tüürib trigeri *TO* väljundpinge. Sõltuvalt sellest, kummal trigeri *TO* väljundil on kõrge pingevoo, osutub igas dešifraatoris avatuks kas vasakpoolne (arvu nihe vasakule) või parempoolne (arvu nihe paremale) loogiline korrutaja. Reas *IV* asuvad loogilised skeemid sooritavad kahe naabertrigeri väärtustega tegelikult loogilise mittesamaväärsuse tehet. Klappe läbivad impulsid antakse sisenditele *q*, millega toimub trigeri ümberlülitamine. See on vajalik vaid siis, kui triger ja tema naabertriger on mittesamaväärsed, s. t. ei salvesta sama numbrit. Registri vabanevale äärmisele trigerile antakse nihke korral peale nullnivoo. See on võrdne kõrvaltrigerile, mille väljundeid imiteerivad alalise pingevooga sisendid «0» ja «1» (sisendil «0» on alaliselt kõrge pinge ja vastupidi).

Selektorimpulsi andmine viib trigeri *T*0 ettenähtud seisundisse ning pärast ümberlülitamisteks vajalikku viidet ahelas *t* käivitub kogu nihkeskeem vastavas suunas.

Registrid, mis võimaldavad nihet paremale ja vasakule, on kasutatavad ka arvude vastuvõtmiseks koht-koha haaval ühe äärmise trigeri kaudu. Selleks tuleb joonisel 59 kujutatud skeemi puhul arvu 10101 viimiseks registrisse pärast registri kustutamist kõigepealt (joonis 60) anda impulss «1», siis selektorimpulss «nihe vasakule», siis impulss «0» (s. t. impulsi puudumine), siis uus «nihe vasakule» jne. kuni täitumiseni (kokku neli nihet vasakule).



Joonis 59



Joonis 60

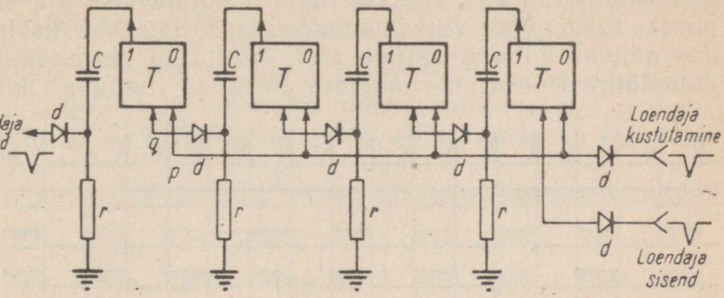
Sellist registrite täitmise süsteemi kasutatakse peamiselt nn. järjestikuse toimega elektron-arvutusmasinates, kus kõik arvud antakse üle kohthaaval kas paremalt vasakule või vastupidi. Paralleelse toimega masinates seevastu toimub kogu arvu üleandmine korraga, ühe impulsi mõjul (seetõttu on vastav masin kiirem).

Loendajad on masinas vajalikud mitmesuguste impulsside loendamiseks. Elektron-arvutusmasinates kasutatakse lihtsuse tõttu peamiselt kahendloendajaid. Kahendloendaja koosneb trigeritest, millede seisundid «0» ja «1» moodustavad loendatud impulsside arvu kahendsüsteemis. Nii on näiteks kümnekohalises kahendloendajas suurim loendatavate impulsside arv 1023 (sest $2^{10}-1=$ $=1023$) ja loendaja täitumisel on kõik 10 trigerit seisundis «1».

Kahendloendajate põhitüüpe on kaks. Esimene (joonis 61) on järjestikuse, teine (joonis 62) paralleelse ülekandega (nimetatakse ka ettevalmistatud ülekandeahelaga loendajaks).

Esimest tüüpi loendaja kujutab endast järjestikku lülitatud trigerite ahelat, milles iga trigeri väljund «1» on lülitatud (läbi diferentseeriva lüli) järgmise trigeri sisendisse q.

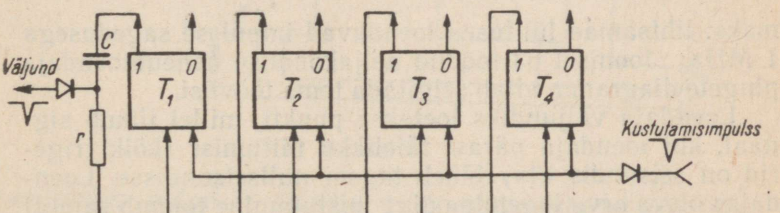
Trigeri lülitamisel nullilt ühele toimub väljundil «1» pinge järsk kõrgenemine, diferentseerides saadakse positiivne impulss, mis ei läbi dioodi d. Ümberlülitumisel ühelt nullile toimub aga pinge langemine, mille diferentseerimisel kujuneb negatiivne impulss, mis lülitab ümber järgmise kahendkoha trigeri. Järgmisele trigerile pääseb seega edasi iga teine impulss. Erilist tähelepanu tuleb pöörata olukorrale, mil kõik trigerid on seisundis «1» ja sisendile antakse impulss. Siis lülituvad üksteise järel ümber kõik trigerid, milleks kuluv aeg võrdub üksikute trigerite ümberlülitamise aegade summaga. Seetõttu võib arvu kujunemiseks vajalik aeg paljukohalises loendajas olla küllaltki



Joonis 61

suur ja sekundis loendatavate impulsside arv suhteliselt väike.

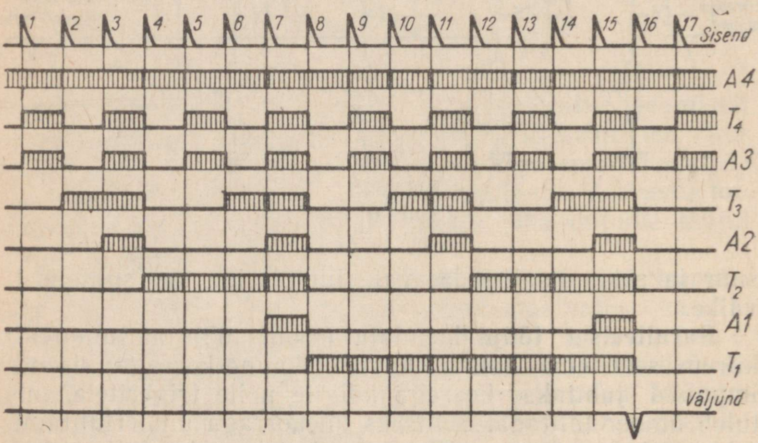
Paralleelset tüüpi loendaja (joonis 62) on tunduvalt kiirem, sest temas toimub kogu ülekanne korraga: sisendimpulsid juhitakse korraga kõigile neile trigeritele, mis tuleb ümber lülitada. Selliseks ühekorraga ümberlülitamiseks peab loendajas olema ettevalmistav «ja»-lülide ahel. Arvu kujunemise ajaks niisuguses loendajas on trigeri



Joonis 62

ümberlülitumise aeg pluss klappide avamiseks vajaliku pinge kujunemise aeg.

Et viimase loogilise korrutaja väljundpinge sõltub kõigist eelmistest, siis viimase numbri kujunemise aja saamiseks tuleb nende viited summeerida. Kuid siiski (kasutades näiteks katoodjärgijate abil sidestatud germaaniumdioodlülitusi) osutub vaadeldav loendaja tunduvalt kiire-



Joonis 63

maks: lihtsamad lülitused loendavad impulsse sagedusega 1 MHz. Joonisel 63 toodud neljakohalise kahendloendaja pingete diagramm aitab selgitada tema tööviisi.

Loendaja väljundiks loetakse punkti, millel ilmub signaal, kui loendaja pärast täielikku täitumist (kõik trigerid on seisundis «1») läheb tagasi nullseisundisse. Loendajas oleva arvu lugemine ning kustutamine toimub samuti nagu registris.

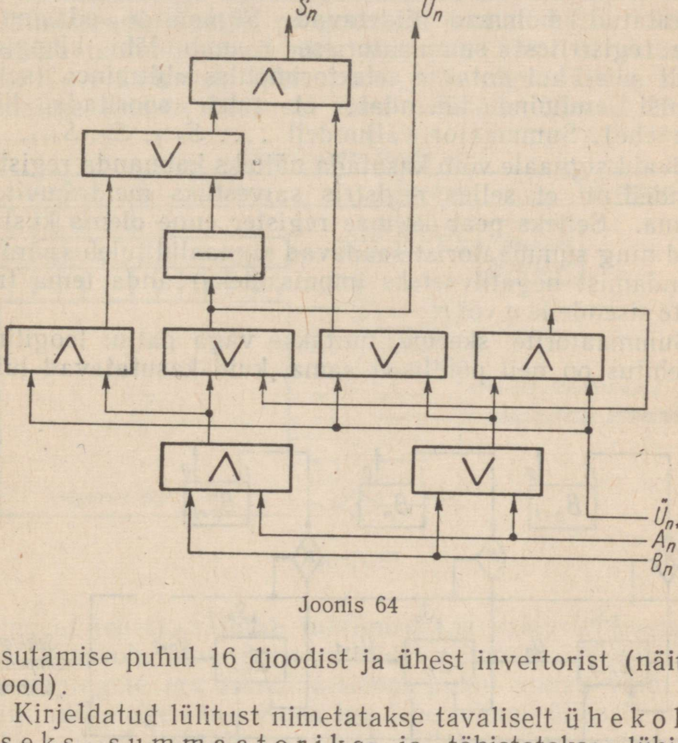
Summaatorid. Nagu me varem nägime (vt. § 5), tuleb kahe kahendarvu liitmiseks igal numbrikohal liita kolm ühekohalist kahendarvu, kusjuures summa saab olla ülimalt kahekohaline.

Tähistame liidetavate vaadeldavad (vasakult paremale lugedes n-ndad) numbrikohad tähtedega A_n ja B_n, summa vastava numbrikoha S_n ning järgmisele numbrikohale mineva ülekande $Ü_n$ (eelmiselt kohalt tulevat ülekannet on siis loomulik tähistada $Ü_{n+1}$). Teises peatükis me leidsime, et vastavad signaalid on omavahel seotud valemitega (siin on muudetud vaid indeksid):

$$\ddot{U}_n = (A_n \wedge B_n) \vee [(A_n \vee B_n) \wedge \ddot{U}_{n+1}];$$

$$S_n = [(A_n \vee B_n) \vee \ddot{U}_{n+1}] \wedge [\overline{\ddot{U}}_n \vee (A_n \wedge B_n \wedge \ddot{U}_{n+1})].$$

Mõlemaid neid valemeid saab realiseerida üheainsa (kolme sisendi ja kahe väljundiga) lülituse abil näiteks joonisel 64 näidatud viisil. See lülitus koosneb dioodide

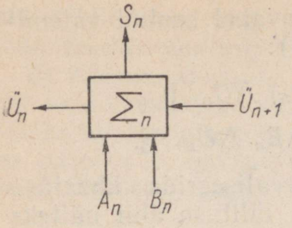


Joonis 64

kasutamise puhul 16 dioodist ja ühest invertorist (näiteks triood).

Kirjeldatud lülitust nimetatakse tavaliselt ü h e k o h a l i s e k s s u m m a a t o r i k s ja tähistatakse lühidalt joonisel 65 näidatud viisil.

Niisuguste ühekohaliste summaatorite ühendamine mitmekohaliste kahendarvude summaatoriks toimub sel teel, et iga ühekohalise summaatori ülekandeväljund viiakse järgmise summaatori ülekandesisendiks. Ühekohalisi summaatoreid nimetatakse siis tavaliselt mitmekohalise summaatori k o h t a d e k s. Saadud summaatori

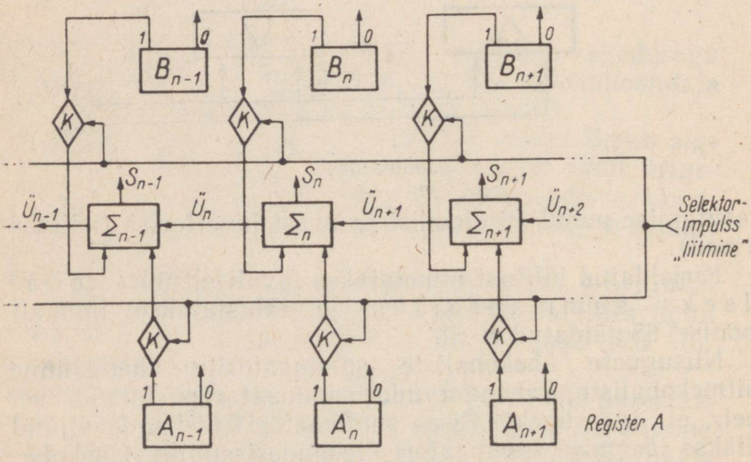


Joonis 65

kasutamine masina aritmeetilises seadmes toimub näiteks joonisel 66 toodud lihtsustatud skeemi kohaselt.

Siin trigerid ..., A_{n-1}, A_n, A_{n+1}, ... moodustavad aritmeetilise seadme ühe registri (register A) ja trigerid ..., B_{n-1}, B_n, B_{n+1}, ... teise (register B). Nendes registrites on enne liitmist salvestatud mõlemad liidetavad. Signaalide edasiandmine registritest summaatorisse toimub läbi klappide ainult siis, kui antakse selektorimpulss «liitmine» (selle impulsi andmine tähendab, et tuleb sooritada liitmise tehe). Summaatori väljundeil ..., S_{n-1}, S_n, S_{n+1}, ... tekkivaid signaale võib kasutada näiteks kolmanda registri sisendeil nii, et selles registris salvestuks meid huvitav summa. Selleks peab kolmas register enne olema kustutatud ning summaatorist saadavad signaalid tuleb (pärast muundamist negatiivseteks impulssideks) anda tema trigerite sisendeile q või r.

Summaatorite skeeme tuntakse väga palju: loogiline ülesehitus on neil põhiliselt sama, kuid kasutatavad lüli-

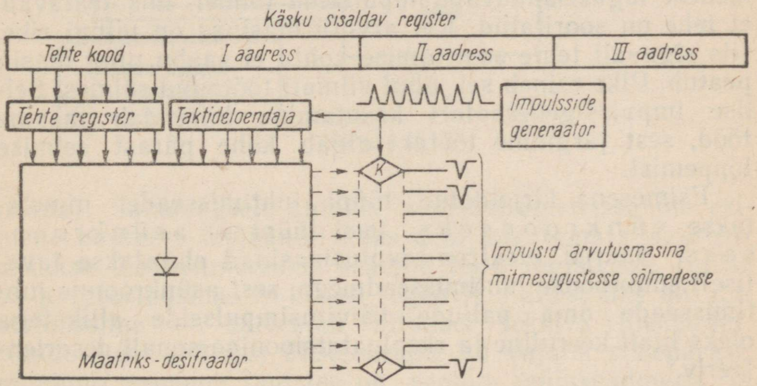


Joonis 66

tuselemendid erinevad. Mõningates elektron-arvutusmasinates (näiteks M-2 ja M-3) kasutatakse summa salvestamiseks ühe liidetava registrit, milleks temas lülitatakse ümber need trigerid, kus summa ja liidetava vastavad kohad kujunevad erinevad. Selline summaator vajab veel abiregistrit ülekande säilitamiseks, kuid on siiski suhteliselt lihtsa ehitusega.

Liitmise ja teiste tehete sooritamise matemaatilist külge vaatleme lähemalt järgmises paragrahvis.

Juhtimisseade. Arvutusmasina töö korraldamisel võtab juhtimisseade aluseks programmis antud käsu. Sellest eraldab ta tehte koodi ning sõltuvalt käsu struktuurist veel 1—4 aadressi. Tehte koodi, mis olgu näiteks kuuekohaline (masin saab sel juhul teostada ülimalt $2^6=64$



Joonis 67

erinevat tehet), säilitab juhtimisseade vastavas registris. See kuuekohaline register toidab dioodmaatriksit, mis sisendpingete iga kombinatsiooni puhul annab vaid ühel väljundil pinge «1». Kuna iga tehe toimub üldiselt mitme takti jooksul (taktide arv sõltub näiteks sellest, mitu korda peame pöörduma mälu poole jne.), kusjuures iga takti ajal on tarvis mitmesuguseid impulsse eri sõlmede käivitamiseks, siis on maatriksiga ühendatud veel eriline taktideloendaja (joonis 67). Selle väljundid on maatriksi sisendeiks ning taktideloendaja sisend on ühendatud sünkroonimpulsside generaatoriga. Iga takti aja[1] toimub maatriksi sisend- ning väljundpingete muutus, kusjuures väljund-pinged avavad ja sulevad klappe, mille väljunditelt lähtuvad masinat juhtivad impulsid. Igale tehte koodile ja kindlale takti järjekorranumbrile vastab kindel väljaantavate impulsside kombinatsioon.

Tsentraalne impulssgeneraator võib olla isevõnkuv, kuid sünkroniseeritud mäluseadme töötaktidega (magnettrumlite ja magnetofonilintide kasutamise korral markerimpulssidega, dünaamiliste mäluseadmete korral viitega liinis, mis sõltub impulsi leviku kiirusest). Mäluseadme töötaktidega on sel juhul sünkroniseeritud kogu masina sõlmede töö, kusjuures kõikide tehete kestused on võrdsustatud täisarvu töötaktide kestusega.

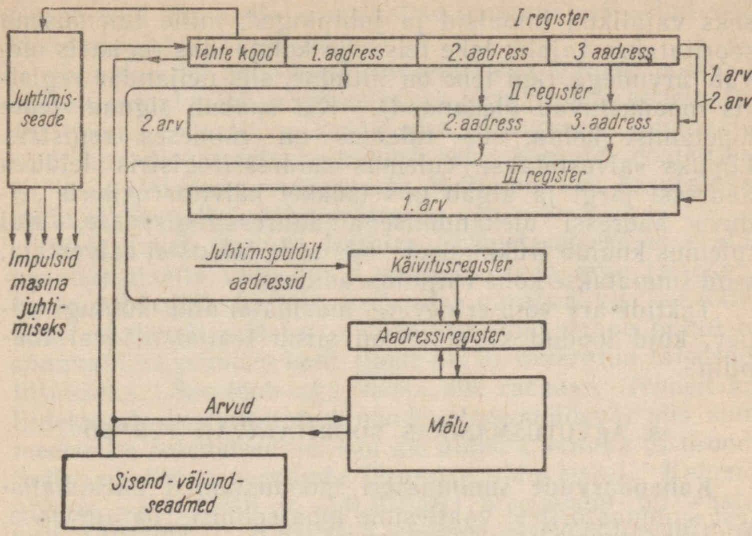
Teine võimalus seisneb välise käivitusega tsentraalse impulssgeneraatori kasutamises. See käivitub masina sõlmedest tagasisaabuvate impulsside toimel, mis teatavad, et tehe on sooritatud. Kui arvutusmasinas on mingi rike, siis signaali tehte sooritamise kohta ei saabu ning masin peatub. Rike esineb sel juhul viimati töötanud sõlmes. Sellise impulss-generaatori kasutamine kiirendab masina tööd, sest järgmine töötakt algab kohe pärast eelmise lõppemist.

Esimesena kirjeldatud tüüpi juhtimisseadet nimetatakse s ü n k r o o n s e k s, teist tüüpi — a s ü n k r o o n s e k s. Suured elektron-arvutusmasinad ehitatakse tavaliselt sünkroonse juhtimisseadmega, sest asünkroonne juhtimisseade oma paljude käivitusimpulsside allikatega oleks liialt keeruline ja ekspluatatsioonipersonali desorienteeriv.

Arvutusmasina sõlmede koostöö. Et anda lugejale ettekujutust arvutusmasina kõikide sõlmede koostööst, vaatleme mingi masina blokk-skeemi (joonisel 68 on toodud elektron-arvutusmasina M-2 blokk-skeem) ning kirjeldame üksikasjaliselt, kuidas täidetakse üks käsk.

Käsu täitmine toimub nelja takti jooksul. Enne masina käivitamist tuleb käivitusregistrisse anda aadress, mille järgi otsitakse esimene käsk. Masina käivitusimpulss kustutab mälu aadressiregistris leiduva aadressi ja annab sinna üle käivitusregistris oleva. Seejärel loetakse vastav käsk ja antakse see üle aritmeetilise seadme esimesse registrisse. Seal paiknevad nüüd (esimese takti järel) kõrvuti tehte kood ja kolm aadressi.

Teise takti ajal antakse tehte kood üle juhtimisseadmesse (mis saab sellega «teada», milline tehe tuleb soo-



Joonis 68

ritada). Samaaegselt antakse aadressiregistrisse üle esimene aadress; ülejäänud kaks seni kasutamata aadressi antakse üle aritmeetilise seadme teise registrisse (üldse on arvutusmasinas M-2 neli registrit, millest üks on abiotstarbeline — salvestab liitmisel tekkiva ülekande). Nende kahe aadressi ülekandmine on vajalik sellepärast, et ainult esimene register on seotud mäluseadmega ja kõik pöördumised mälu poole toimuvad tema kaudu. Esimese aadressi järgi ülesotsitud arv ilmub nüüd esimesse registrisse (selleks tuligi see register vabastada).

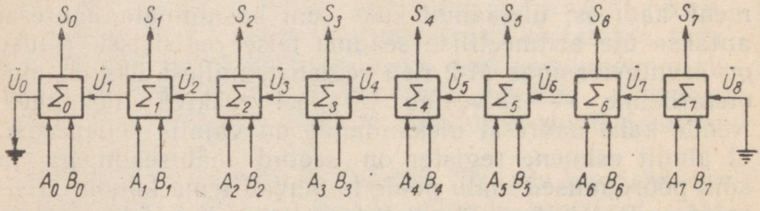
Kolmanda takti ajal antakse aadressiregistrisse üle teine aadress, tuuakse teine arv mälust esimesse registrisse ja samaaegselt antakse esimene arv üle kolmandasse registrisse. Ühtlasi saadetakse impulss käivitusregistrisse, milles olev aadress selle tõttu suureneb ühe võrra (käivitusregister kujutab endast tegelikult loendajat). Sellega on käivitusregister ette valmistatud järgmise käsu ülesotsimiseks.

Neljanda takti ajal antakse aadressiregistrisse üle kolmas aadress ja viiakse esimeses registris olev teine arv teise registrisse (selles varem olnud aadressid on juba ära kasutatud). Seejärel lülituvad sisse tehte sooritamiseks vajalikud impulsid ja juhtpinged, mille tulemusena sooritatakse vajalik tehe teises ja kolmandas registris olevate arvudega (kui tehe on liitmine, siis neljandas registris moodustuvad ülekanded). Kui saabub signaal tehte lõpetamise kohta, siis tulemus on esimeses registris. Lõpuks salvestatakse tulemus aadressiregistris leiduva aadressi järgi ja algab uus tsükkel käivitusregistris leiduva aadressi ülekandmisega aadressiregistrisse. Kui tulemus kuulub trükkimisele, siis seda mälus ei salvestata, vaid suunatakse kohe väljundseadmesse.

Taktide arv võib erinevatel masinatel olla muidugi erinev, kuid toodud kirjeldus on siiski teatud määral tüüpiline.

10. ARVUTUSMASINAS SOORITATAVAD TEHTED

Kahendarvude summaatori moodustamist ühekohalistest summaatoritest vaatlesime juba eelmises paragrahvis. Seal ei pööranud me aga tähelepanu sellele, kuidas on ühendatud summaatori äärmised kohad. Üks võimalus nende kohtade ühendamiseks on näidatud joonisel 69, kus



Joonis 69

lihtsuse mõttes on piirdutud kaheksa kohaga. Kohti nummerdame nagu mälu pesa puhulgi (vt. § 4) numbritega 0, 1, 2,... Samu numbreid on kasutatud nendelt kohtadelt lähtuvate ülekannete ja summa numbrikohtade indeksitena.

Ülekande $Ü_0$ maandatus tähendab tegelikult, et see ülekanne ei leia kasutamist (vastava ühekohalise summaatori võib antud juhul konstrueerida lihtsustatud viisil — ainult väljundiga S_0). Sisendi $Ü_8$ maandatusega on tähistatud seda, et sellel sisendil on pinge alaliselt madal, s. t. viimasele numbrikohale tuleb alati anda ülekanne null (ka selle ühekohalise summaatori puhul võib kasutada lihtsustatud skeemi).

Summaatori ühendamine registritega toimub põhimõtteliselt samal viisil nagu eelmises paragrahvis kirjeldatud (vt. joonis 66). Me ei vaatle siin niisuguseid summaatoreid, kus ülekannete kogumine toimub eraldi registrisse (näiteks elektron-arvutusmasin M-2), või kus kasutatakse muid «ebatavalisi» liitmisvõtteid.

Relatiivsete arvude liitmise lihtsustamiseks ei eralda me märgikohta välja, vaid vaatleme teda arvu täisosana (seda me tegime ka juba neljandas paragrahvis, kui tõime sisse otsekoodi mõiste). Niisuguse kokkuleppe puhul on summaatori esimene koht (koht nr. 0) määratud täisosade liitmiseks. See toob aga kaasa uue raskuse. Nimelt kui liidetavad on kujutatud nende otsekoodidena, siis summeerimise tulemusena ei saa me üldiselt summa otsekoodi. Selles on lihtne veenduda järgmise näite varal. Kahendarvude $-0{,}1011001$ ja $+0{,}1010011$ otsekoodid on vastavalt $1{,}1011001$ ja $0{,}1010011$. Nende otsekoodide liitmisel saame

$$
\begin{array}{r}
1{,}1011001 \\
+\quad \underline{0{,}1010011} \\
10{,}0101100
\end{array}
$$

Kuna aga viimane ülekanne $\ddot{U}_0$ maandub, siis tekib väljunditel S tegelikult arv $0{,}0101100$, s. t. arvu $+0{,}0101100$ otsekood. Vaadeldavate liidetavate algebraline summa on aga hoopis $-0{,}0000110$ (ehk otsekoodina $1{,}0000110$).

Seega kerkib vajadus niisuguse koodi loomiseks, mis säilib liitmisel. Joonisel 69 kujutatud summaatori korral on selliseks koodiks nn. täiendkood.

Lihtkahendmurru x t ä i e n d k o o d i k s nimetatakse kahendarvu $[x]_{tk}$, mis on defineeritud valemiga

$$
[x]_{tk} = \begin{cases} x\ \ , & \text{kui } x \geqslant 0 \\ 10+x, & \text{kui } x<0 \end{cases}
$$

(siin 10 tähendab kaks). Näiteks

$$
\begin{aligned}
&[+0{,}1011101]_{kt} = 0{,}1011101; \\
&[-0{,}1011101]_{tk} = 10 + (-0{,}1011101) = 1{,}0100011; \\
&[-0{,}1101010]_{tk} = 10 + (-0{,}1101010) = 1{,}0010110.
\end{aligned}
$$

Järelikult positiivse arvu täiendkood ühtib tema otsekoodiga (ja arvu endaga). Negatiivse arvu täiendkoodi praktiliseks moodustamiseks võib aga anda järgmise lihtsa reegli: arvu märk jäetakse kirjutamata ning täisosaks pannakse 1 (kui teisendamine toimub otsekoodist täiendkoodi, siis jääb täisosa lihtsalt muutmata); murdosas asendatakse kõik nullid ühtedega ja ühed nullidega ning liidetakse viimasele kohale üks. Näiteks

$$[-0{,}1101010]_{tk}=1{,}0010101+0{,}0000001=1{,}0010110.$$

Selle reegli põhjendamise jätame lugejale, nimetame vaid, et esimene kokkupuutumine täiendkoodi mõistega toimub tegelikult juba keskkoolis. Nii on paljudes tabelites (kus arvude absoluutväärtused muutuvad vaid teatud piirides) negatiivsed arvud asendatud nende täiendustega kümneni. Selle võttega välditakse märkide trükkimine, sest näiteks arv $-0{,}7325$ omab tabelis kuju $9{,}2675$ $(=10-0{,}7325)$. Teiselt poolt aga on niisugused tabelid arvutamisel mugavamad (meenutame, et ühest väiksema arvu logaritmi mantiss kirjutatakse peaaegu erandituit just «täiendkoodis»).

Arvude liitmine fikseeritud komaga masinas võib toimuda näiteks järgmiselt. Aritmeetiline seade on konstrueeritud nii, et mälust võetud liidetavad teisendatakse registrites kõigepealt täiendkoodi ning antakse siis summaatori sisenditele A ja B. Sealjuures koht nr. 0 on määratud täisosa kujutamiseks. Näitame, et kahe arvu täiendkoodide liitmisel joonisel 69 kujutatud summaatoriga saame väljundeil S nende summa täiendkoodi (eeldusel, et summa absoluutväärtus osutub ühest väiksemaks, s. t. et ei toimu pesa ületäitumist).

Olgu x ja y kaks lihtkahendmurdu, millede murdosad on antud ühesuguse arvu kohtadega, kusjuures $|x+y|<1$. Nende murdude liitmisel võib (sõltuvalt liidetavate ja summa märkidest) esineda neli põhimõtteliselt erinevat juhtu.

1. Kui $x>0$ ja $y>0$, siis nende täiendkoodid ühtivad arvude enestega ning liitmisel saame:

$$[x]_{tk}+[y]_{tk}=x+y=[x+y]_{tk},$$

sest $x+y>0$. Kuna sealjuures veel $x+y<1$, siis ülekannet täisosasse ei toimu, s. t. täisosaks jääb 0, nagu see positiivse arvu täiendkoodis peabki olema.

2. Kui $x>0$, $y<0$ ja $x+y<0$, siis täiendkoodide liitmine annab tulemuseks

$$[x]_{tk}+[y]_{tk}=x+(10+y)=10+(x+y)=[x+y]_{tk},$$

sest $x+y$ on ju eelduse kohaselt negatiivne lihtmurd.

3. Kui $x>0$, $y<0$, aga $x+y>0$, siis täiendkoodide liitmisel saame:

$$[x]_{tk}+[y]_{tk}=x+(10+y)=10+(x+y)=10+[x+y]_{tk}.$$

Näiliselt pole siin tõestatav omadus kehtiv, kuid et summaatoris puudub koht täisosa kaheliste jaoks (ülekanne $\bar{U}_0$ on maandatud) siis läheb liidetav 10 kaotsi ja väljunditel S tekib tegelikult ikkagi summa täiendkood.

Näide. $x=0{,}1101101$, $y=-0{,}1000110$ ($x+y=0{,}0100111$).

$$\begin{array}{r} + \begin{array}{r} [x]_{tk}= 0{,}1101101 \\ [y]_{tk}= 1{,}0111010 \end{array} \\ \hline [x]_{tk}+[y]_{tk}=10{,}0100111 \end{array}$$

Väljundeil S saame $0{,}0100111=[x+y]_{tk}$.

4. Kui $x<0$ ja $y<0$, siis ka $x+y<0$. Sel juhul

$$[x]_{tk}+[y]_{tk}=(10+x)+(10+y)=10+[10+(x+y)]=$$
$$=10+[x+y]_{tk}.$$

Tulemus peaks järelikult omama kuju 11,..., kuid väljundi $\bar{U}_0$ maandatuse tõttu läheb kaheliste kohale suunatud ülekanne jälle kaotsi ja summaatori väljunditel S saame tegelikult arvu $[x+y]_{tk}$.

Näide. $x=-0{,}0010111$, $y=-0{,}1010110$ ($x+y=$ $=-0{,}1101101$).

$$\begin{array}{r} + \begin{array}{r} [x]_{tk}=1{,}1101001 \\ [y]_{tk}=1{,}0101010 \end{array} \\ \hline [x]_{tk}+[y]=11{,}0010011. \end{array}$$

Väljunditel S saame $1{,}0010011=[-0{,}1101101]_{tk}$.

Juhul kui $|x+y|_{tk} \geqslant 1$, s. t. kui toimub pesa ületäitumine, ei saa me väljundeil S enam summa täiendkoodi (ühest suurema arvu täiendkood pole üldse defineeritudki). Summaatoris väljendub ületäitumine selles, et toimub lisaülekanne murdosast täisosasse ja märk muutub seega vastupidiseks.

Näide. $x=-0{,}1011100$, $y=-0{,}1101101$ $(x+y= =-1{,}1001001)$.

$$
\begin{array}{r}
+\begin{array}{r} [x]_{tk}=1{,}0100100 \\ [y]_{tk}=1{,}0010011 \end{array} \\
\hline
[x]_{tk}+[y]_{tk}=10{,}0110111.
\end{array}
$$

Väljundeil S saame 0,0110111, mis on positiivse lihtmurru täiendkood, samal ajal kui $x+y$ peab olema negatiivne arv.

Ilmselt saab lihtmurdude liitmisel ületäitumine toimuda ainult siis, kui liidetavad on samamärgilised. Ületäitumise tunnuseks võib seega kasutada asjaolu, et samamärgiliste liidetavate summa on teise märgiga kui liidetavad. Kuna ületäitumine arvutustel pole lubatav, siis peab masinas olema seade, mis võrdleb kõigepealt liidetavate märke ning nende ühtelangemise korral ühe liidetava märki summa märgiga. Kui viimased ei lange ühte, siis annab see seade ületäitumise signaali (nn. signaal φ) või peatab masina.

Ületäitumise kontrollimist võib lihtsustada nn. modifitseeritud täiendkoodi kasutamisega. Selle koodi puhul kujutatakse iga arvu täisosa (märk) kahekohalisena, mille tõttu summaator tuleb ehitada ühe koha võrra suuremana (tuleb lisada koht nr. —1).

Lihtkahendmurru x m o d i f i t s e e r i t u d t ä i e n d k o o d i k s nimetatakse arvu $[x]_{mt}$, mis on defineeritud valemiga

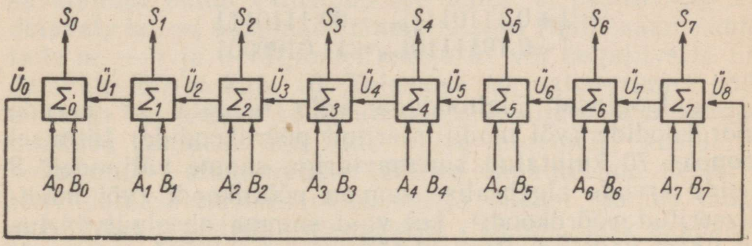
$$
[x]_{mt}=\begin{cases} x & \text{, kui } x \geqslant 0 \\ 100+x, & \text{kui } x<0, \end{cases}
$$

kus 100 tähendab arvu neli. Näiteks

$$
\begin{aligned}
[+0{,}1011011]_{mt}&=00{,}1011011 \\
[-0{,}1000110]_{mt}&=11{,}0111010
\end{aligned}
$$

(negatiivse arvu modifitseeritud täiendkoodi murdosa leitakse sama reegli järgi nagu täiendkoodi murdosa). Selle koodi puhul on ületäitumise tunnuseks asjaolu, et summa täisosa koosneb erinevatest numbritest (s. t. täisosa on kas 01 või 10). Ületäitumise kontrollimiseks tuleb seda koodi kasutavas masinas lihtsalt võrrelda summa täisosa kohti.

Täiend- või modifitseeritud täiendkoodi kasutatakse niisuguste masinate puhul, mille summeerija on ehitatud joonisel 69 toodud skeemi kohaselt. Enamikus masinates (sealhulgas ka näiteks nõukogude arvutusmasinates



Joonis 70

«Ural» ja «Strela») ehitatakse aga summaator nii, et täisosast läheb ülekanne viimasele murrukohale (joonis 70).

Sellise summeerija korral on liitmisel säilivateks koodideks nn. pöördkood ja modifitseeritud pöördkood.

Lihtkahendmurru x p ö ö r d k o o d i k s nimetatakse arvu $[x]_{pk}$, mis on defineeritud valemiga

$$[x]_{pk} = \begin{cases} x & \text{, kui } x \geqslant 0 \\ 10 - 10^{-n} + x, & \text{kui } x < 0, \end{cases}$$

kus 10 tähendab arvu kaks ja n on numbrikohtade arv kasutatavate arvude murdosades. Näiteks ($n = 7$ korral)

$$[+0{,}0101101]_{pk} = 0{,}0101101$$

$$[-0{,}1011011]_{pk} = 10 - 0{,}0000001 - 0{,}1011011 = 1{,}0100100.$$

Negatiivse murru pöördkoodi saame ilmselt sel teel, et täisosaks kirjutame ühe, kuid murdosas asendame kõik ühed nullidega ja nullid ühtedega. See asjaolu, et otsekoodist pöördkoodi on lihtsam teisendada kui täiendkoodi, ongi muide pöördkoodi eelistamise põhjuseks (arvu pöördkoodi saame registris väljundeilt «0», vt. joonis 56).

Modifitseeritud pöördkoodi puhul kujutatakse täisosa (s. t. märk) jälle kahe numbrikohaga. Nimelt lihtkahendmurru x **m o d i f i t s e e r i t u d p ö ö r d k o o d i k s** nimetatakse arvu $[x]_{mp}$, mis on defineeritud valemiga

$$[x]_{mp}=\begin{cases} x & \text{, kui } x\geqslant 0 \\ 100-10^{-n}+x, & \text{kui } x<0 \end{cases}$$

(n omab endise tähenduse, aga 100 on arv neli). Näiteks

$$[+0{,}1110111]_{mp}=00{,}1110111$$
$$[-0{,}1011110]_{mp}=11{,}0100001.$$

Analoogiliselt ülaltooduga võib näidata, et arvude pöördkoodide (või modifitseeritud pöördkoodide) liitmisel joonisel 70 kujutatud summaatoriga saame väljundeil S nende arvude algebralise summa pöördkoodi (või modifitseeritud pöördkoodi), kui vaid summa absoluutväärtus on ühest väiksem. Pesa ületäitumise tunnused on täpselt samad nagu täiendkoodi (või modifitseeritud täiendkoodi) korral.

Arvude liitmine liikuva komaga masinas. Liikuva komaga masina puhul toimub arvude mantisside liitmine põhimõtteliselt samuti nagu arvude liitmine fikseeritud komaga masinas. Peamine erinevus seisneb selles, et mantisside liitmisel tekkida võiv ületäitumine ei tohi nüüd moonutada arvutustulemust, sest järgu suurenemisega ühe võrra saab sellise ületäitumise lihtsalt likvideerida. Seepärast ei saa liikuva komaga masinate korral kasutada täiend- või pöördkoodi, vaid peab kasutama kas modifitseeritud täiendkoodi või modifitseeritud pöördkoodi. Alljärgnevas piirdume modifitseeritud täiendkoodi kasutavate masinate vaatlemisega.

Sellises masinas toimub arvude liitmine järgmiselt. Kõigepealt teisendatakse liidetavate järgud võrdseteks. Selleks suurendatakse väiksema järguga liidetava järku ning nihutatakse samaaegselt mantissi paremale seni, kuni järk saab võrdseks teise liidetava järguga. Seejärel teisendatakse mõlema liidetava mantissid modifitseeritud täiendkoodi ja liidetakse summaatoris. Saadud tulemus teisendatakse tagasi otsekoodi (tarbe korral normaliseerides) ja ta annab summa mantissi. Summa järguks on liidetavate ühine järk (normaliseerimise korral tuleb järku muidugi vastavalt muuta).

Normaliseerimise vajadus saab tekkida kahel põhjusel. Esiteks võib mantisside liitmine anda tulemuseks normaliseerimata mantissi (see saab juhtuda ka normaliseeritud, kuid vastasmärgiliste liidetavate puhul). Sel korral tuleb summa tavalisel viisil normaliseerida, s. t. nihutada mantissi vasakule ning vähendada samaaegselt järku.

Teiseks võib samamärgiliste mantisside liitmisel toimuda kohtade ületäitumine. Modifitseeritud täiendkoodi kasutamise puhul väljendub see selles, et positiivsete liidetavate korral omandab summa täisosa (mille saame kohtade nr. —1 ja 0 väljundeil) kuju 01 või negatiivsete liidetavate korral kuju 10. Tulemuse normaliseerimist võib teostada sel teel, et kõik summaatori väljundeil S saadud numbrid kantakse ühe koha võrra paremale, kusjuures kohal nr. —1 olnud number jääb ka sinna püsima. Seejärel teisendatakse tulemus otsekoodi ja viiakse summa jaoks ettenähtud registrisse mantissiks. Summa järk saadakse liidetavate järgu suurendamisel ühe võrra.

Illustreerime viimaseid juhte arvuliste näidetega, piirdudes seitsmekohaliste mantisside ja neljakohaliste järkudega (koma asemel jätame koodides suuremad vahed).

Näide 1. Liita arvud $-1101{,}101$ ja $+100110{,}1$, mis on mälu pesades kujutatud oma otsekoodidena:

1 1101101 0 0100
0 1001101 0 0110.

Kõigepealt tuleb esimene liidetav järkude võrdsustamiseks teisendada kujule

1 0011011 0 0110.

Mõlema liidetava mantisside üleviimisel modifitseeritud täiendkoodi saame:

11 1100101
00 1001101.

Edasi liidetakse mantissid (sealjuures ülekanne kohalt nr. —1 maandub):

$$
\begin{array}{r}
11\ \ 1100101 \\
+\ 00\ \ 1001101 \\
\hline
00\ \ 0110010.
\end{array}
$$

Kuna ületäitumist ei toimunud, siis teisendatakse tulemus otsekoodi, kusjuures lisatakse ka järk:

0 0110010 0 0110.

Lõpuks tuleb saadud summa normaliseerida ning saata vastuse jaoks ettenähtud pesasse kujul

0 1100100 0 0101.

Seega saime tulemuseks $+0{,}1100100 \cdot 10^{101} = +11001{,}00$.

Näide. 2. Liidetavad on antud oma otsekoodidena:

1 1011010 0 0111
1 1101011 0 0110.

Liitmine teostatakse järgmiste sammude kaupa.

Teise liidetava teisendamine esimesega samajärguliseks (koos ümardamisega):

1 0110110 0 0111.

Mantisside teisendamine modifitseeritud täiendkoodi:

11 0100110
11 1001010.

Mantisside liitmine:

$$
\begin{array}{r}
11\ \ 0100110 \\
+\ 11\ \ 1001010 \\
\hline
10\ \ 1110000.
\end{array}
$$

Ületäitumise likvideerimine:

11 0111000.

Tagasiminek otsekoodi (koos järgu suurendamisega):

1 1001000 0 1000.

Saadud tulemus kantaksegi vastuse jaoks määratud pesasse.

Teiste aritmeetiliste tehete sooritamine. Arvude l a h u t a m i n e taandatakse liitmisele valemi

$$x-y=x+(-y)$$

abil. Seega lahutamiseks tuleb lihtsalt muuta lahutatava märk ja liita tulemused ülalkirjeldatud viisil.

K o r r u t a m i s e l leitakse kõigepealt korrutise märk sel teel, et liidetakse tegurite märgid (s. t. tegurite otsekoodide täisosad) ühekohalises summaatoris, mille ülekandeväljund on maandatud. Lihtne kontroll näitab, et nii saame tõepoolest korrutise märgi (näiteks $0+0=0$ tähendab, et «+» · «+»=«+» jne.).

Arvude eneste korrutamine fikseeritud komaga masinais ja mantisside korrutamine liikuva komaga masinais teostatakse otsekoodis, kusjuures korrutamine toimub tegelikult samuti nagu kahendarvude korrutamine paberil. Nimelt nihutatakse üht tegurit järk-järgult paremale (igakordse nihke pikkuse määrab ühtede jaotus teises teguris) ja liidetakse tulemused. Lõpptulemus viiakse selleks ettenähtud pesasse. Kui mantiss ei mahu pesasse ära, siis tuleb enne teostada ümardamine.

Näide. Arvude $-0{,}1011011$ ja $+0{,}1101001$ (nende otsekoodid on vastavalt 1 1011011 ja 0 1101001) korrutamisel leitakse kõigepealt korrutise märk: $1+0=1$. Murdosade korrutamine toimub järgmise skeemi kohaselt:

$$\begin{array}{r|l}
\times 1 & 1011011 \\
0 & 1101001 \\ \hline
 & 01011011 \\
 & 01011011 \\
 & 01011011 \\
 & 01011011 \\ \hline
1 & 10010101010011.
\end{array}$$

Ümardamine annab tulemuseks 1 1001011 (s. t. $-0{,}1001011$).

Liikuva komaga masina korral tuleb lisaks sellele leida veel korrutise järk, milleks on tegurite järkude algebraline summa, ja tarbe korral normaliseerida tulemus.

Jagatise märk leitakse samuti nagu korrutise puhul, kuna aga jagamine ise teostatakse jälle n. ö. tavalisel viisil. Fikseeritud komaga masina korral peab jagatav sealjuures (ületäitumise vältimiseks) olema väiksem kui jagaja. Jagatise iga number määratakse sel teel, et lahutatakse jagaja jagatavast (või varemtekkinud jäägist) ning kontrollitakse tekkiva jäägi märki. Kui jääk on positiivne, siis kirjutatakse jagatisse number 1, nihutatakse jagaja ühe koha võrra paremale ja korratakse protsessi. Kui aga jääk osutus negatiivseks, siis kirjutatakse jagatisse number 0 ja enne jagaja paremale nihutamist liidetakse ta tekkinud negatiivse jäägiga. Nii toimitakse seni, kuni jagatises on saadud vajalik arv kohti. Arvutustel kasutatakse põhiliselt otsekoodi, sest tehted toimuvad ju arvude absoluutväärtustega (ainult jagaja igakordne lahutamine toimub enamasti täiend- või pöördkoodis).

Liikuva komaga masina puhul langeb nõue, et jagatav peab olema väiksem kui jagaja, muidugi ära. Siin tuleb lisaks mantisside jagamisele leida veel jagatise järk jagatava ja jagaja järkude lahutamise teel ning normaliseerida lõpuks tarbe korral tulemus.

Paljudes liikuva komaga masinates asendatakse aga jagamine kahe eraldi tehtega: kõigepealt leitakse jagaja pöördväärtus ja korrutatakse see siis jagatavaga valemi $\frac{x}{y} = \frac{1}{y}\,x$ kohaselt. Sealjuures pöördväärtuse $\frac{1}{y}$ leidmine toimub masinas alaliselt säilitatava programmi (nn. standardse alamprogrammi) järgi mingi ligikaudse meetodi abil[1]. See võte võimaldab tunduvalt lihtsustada masina aritmeetilise seadme ehitust, sest seal pole nüüd tarvis eraldi seadet arvude jagamiseks.

Standardsete alamprogrammide järgi teostatakse masinates sageli ka tähtsamate elementaarfunktsioonide (näiteks $\sqrt{x}$, $\sin x$, $\log x$, e^x, $\arcsin x$ jne.) väärtuste arvutamine ning arvude teisendamine ühest arvusüsteemist teise. Niisuguste alamprogrammide olemasolu lihtsustab tunduvalt ülesannete programmeerimist, sest pöördumise saab põhiprogrammis teostada üheainsa käsuga (pärast alamprogrammi lõppemist läheb juhtimine automaatselt üle sellele käsule, mis järgneb alamprogrammi sisselülitanule).

---

[1] Nii on see näiteks nõukogude elektron-arvutusmasinas «Strela».

On aga ehitatud ka masinaid, milles iga elementaarfunktsiooni väärtuste arvutamiseks on omaette seade. See muudab masina muidugi kallimaks, kuid kiirendab üldiselt arvutusi.

Loogilised tehted. Peale kirjeldatud aritmeetiliste tehete sooritavad arvutusmasinad veel mitmesuguseid loogilisi tehteid. Need tehted on vajalikud eeskätt masina töö automaatseks juhtimiseks (kuid ka aritmeetiliste tehete sooritamise käigus, näiteks märkide võrdlemiseks). Põhiliselt on masinas tarvis järgmist nelja loogilist tehet.

1. L o o g i l i n e   k o r r u t a m i n e. See tehe seisneb selles, et leitakse tegurite samanimelistel kohtadel asuvate numbrite loogilised korrutised (vt. § 5) tabeli

$$0 \wedge 0 = 0 \qquad 1 \wedge 0 = 0$$
$$0 \wedge 1 = 0 \qquad 1 \wedge 1 = 1$$

kohaselt. Tehet kasutatakse peamiselt arvudest või käskudest ühe osa väljaeraldamiseks (nimetataksegi sageli osa eraldamise tehteks). Sel korral peab teiseks teguriks olema valitud niisugune abiarv, millel on ühed ainult nendel kohtadel, milliseid esimesest tegurist soovitakse välja eraldada. Näiteks kui mingist üheaadressilisest käsust on tarvis välja eraldada tema tehte kood, siis tuleb teostada selle käsu loogiline korrutamine abiarvuga 77 0000 (ehk kahendsüsteemis 111111 000 000000000).

2. L o o g i l i s e   l i i t m i s e puhul sooritatakse liidetavate samanimelistel kohtadel asuvate numbritega loogilise liitmise tehe ($0 \vee 0 = 0$, $0 \vee 1 = 1$, $1 \vee 0 = 1$, $1 \vee 1 = 1$). Seda tehet kasutatakse eeskätt arvude moodustamiseks (formeerimiseks) antud arvude osadest. Näiteks võib osutuda vajalikuks niisuguse käsu moodustamine, mille aadress on võetud ühest olemasolevast käsust, tehte kood aga teisest. Sel juhul tuleb antud käskude vajalikud osad kõigepealt loogilise korrutamise abil välja eraldada ja siis loogilise liitmise teel ühendada uueks käsuks.

3. V õ r d l e m i s e tehe seisneb selles, et võrreldavate arvude samanimelistel kohtadel olevate numbritega teostatakse mittesamaväärsuse loogiline tehe ($0 \overline{\sim} 0 = 0$, $0 \overline{\sim} 1 = 1$, $1 \overline{\sim} 0 = 1$, $1 \overline{\sim} 1 = 0$). Nagu juba nimetus ütleb, kasutatakse seda tehet arvude (või ka käskude) võrdlemiseks, kusjuures arvud on võrdsed siis ja ainult siis, kui tulemus koosneb üksnes nullidest.

4. Loogiliste tehete hulka tuleb lugeda veel ka a r v u n i h u t a m i n e teatud arvu kohtade võrra paremale või vasakule. Nihutamise suund ja ulatus tuleb määrata teise arvuga (mis võib olla kas ette antud või ülesande lahendamise käigus leitud). Kui nihutamist kasutada eraldi tehtena, siis kuuluvad liikuva komaga masinas nihutamisele nii mantiss kui ka järk ühe tervikuna. Aritmeetiliste tehete koostisosana (näiteks normaliseerimisel) toimub nihutamine aga enamasti ainult mantissis.

Arvutusmasinas teostatavatest tehetest parema ülevaate saamiseks toome järgnevalt kahe konkreetse arvutusmasina käskude süsteemid.

Arvutusmasinas «Ural» teostatavad tehted. «Urali» sisemiseks mäluks on 2048-pesaline magnettrummel. Iga niisugune l ü h i k e p e s a sisaldab 18 kahendkohta ja omab aadressi, mis kirjutatakse üles neljakohalise kaheksandarvuga 0000—3777. Masina konstruktsioon aga võimaldab kõrvuti lühikeste pesadega kasutada ka 36-kohalisi p i k k i p e s a s i d. Pikk pesa koosneb kahest järjestikusest lühikesest pesast, kusjuures esimene lühike pesa peab tingimata olema paarisarvulise aadressiga. Pika pesa aadressiks on esimese temasse kuuluva lühikese pesa aadress pluss 4000 (kaheksandsüsteemis). Näiteks lühikestest pesadest 2142 ja 2143 koosneva pika pesá aadressiks on 6142.

Arvutusmasin «Ural» on üheaadressiline, käske täidetakse loomulikus järjekorras. Käsk koosneb seega kahest kaheksandarvust — tehte koodist ja aadressist — ning omab kirjutatult näiteks kuju

16 3427,

kus 16 on tehte kood ja 3427 aadress.

Käskude salvestamine toimub eranditult lühikestesse pesadesse. Sealjuures pesa esimesed kuus kohta on määratud tehte koodi ja ülejäänud kaksteist aadressi kujutamiseks. Tehte koodi jaoks eraldatud kuuest kohast jääb esimene tegelikult kasutamata, sest masinas on ainult 27 erinevat tehte koodi (kuue koha abil saaks kujutada kokku 64 erinevat arvu). Selle koha kasutamisel peatume hiljem.

Masina käskude süsteemiga tutvumisel tuleb arvestada, et aritmeetiline seade sisaldab kaks ühepesalist mälu. Nendest 37-kohaline s u m m a a t o r (järgnevas tähistame teda tarbe korral lühidalt tähega s) säilitab temas olevat arvu seni, kuni sinna suunatakse uus arv; 36-kohalises registris (tähistame r) säilib arv aga ainult ühe tehte vältel. Mälus ja registris kujutuvad arvud otsekoodis, kuid summaatoris modifitseeritud pöördkoodis.

Pärast aritmeetiliste ja loogiliste tehete sooritamist saadab aritmeetiline seade, sõltuvalt tehte tulemusest, juhtimisseadmesse masina töö automaatseks juhtimiseks vajaliku signaali (mida nimetatakse signaaliks ω). Nimelt kui tehte tulemus rahuldab mingit tingimust, näiteks on negatiivne, siis signaal saadetakse (sel juhul ütleme, et $\omega=1$). Kui see tingimus aga pole täidetud, siis signaali ei saadeta ($\omega=0$). Allpool nimetame iga tehte puhul, millal saadetakse signaal $\omega=1$.

Kui tehte tulemusel toimub summaatori ületäitumine, siis saadetakse juhtimisseadmesse signaal φ (ütleme, et $\varphi=1$). Masina reageerimine signaalile $\varphi=1$ oleneb kahe lüliti seisust juhtimispuldil. Lüliti «φ blokeerimine» avatud oleku korral jätab masin signaali φ tähele panemata. Kui on avatud lüliti «peatus φ puhul», siis jääb masin $\varphi=1$ korral seisma. Kui aga mõlemad need lülitid on suletud, siis signaali $\varphi=1$ korral annab masin juhtimise automaatselt üle pesas 0001 olevale käsule.[1] Sellesse pesasse salvestatakse varem käsk, mis annab juhtimise omakorda üle kindlale programmi osale, mille ülesandeks on ületäitumise põhjuste likvideerimine (näiteks mastaapide muutmine).

Edaspidises kasutame järgmisi tähistusi. Kui pesa aadress on tähistatud tähega a, siis sümbol (a) tähendab selles pesas salvestamisel olevat arvu või käsku. Enne tehte sooritamist summaatoris olnud arvu tähistame sümboliga (s_0), sümboli (s) reserveerime aga summaatoris pärast tehte sooritamist oleva arvu jaoks. Sümbol $(a)_i$ tähendab arvu (a) i-ndat numbrikohta ja (r) registris olevat arvu.

Tabelis 4 on toodud kõik masinas «Ural» täitmisele kuuluvad käsud. Arvud on sealjuures antud kaheksandsüsteemis, s. t. sel kujul, nagu nad tuleb kirjutada plangile. Selgitame vastavaid tehteid veidi lähemalt.

Tehete 01—10 tulemus saadakse summaatoris modifitseeritud pöördkoodis, kusjuures signaal $\omega=1$ saadetakse

[1] Masina mõnedes eksemplarides on juhtimise üleandmine asendatud ühe käsu vahelejätmisega.

Tabel 4

Tehte kood	Tehte nimetus	Käsu kuju	Signaal $\omega=1$	Tehte sisu selgitus
01	Liitmine	01 *a*	$(s)<0$	$(s)=(s_o)+(a)$
02	Summaatorisse saatmine	02 *a*	$(s)<0$	$(s)=(a)$
03	Lahutamine	03 *a*	$(s)<0$	$(s)=(s_o)-(a)$
04	Absoluutväärtuste lahutamine	04 *a*	$(s)<0$	$(s)=\lvert(s_o)\rvert-\lvert(a)\rvert$
05	I liiki korrutamine	05 *a*	$(s)<0$	$(s)=(s)+(r)\cdot(a)$
06	II liiki korrutamine	06 *a*	$(s)<0$	$(s)=(s_o)\cdot(a)$
07	Jagamine	07 *a*	$(s)<0$	$(s)=(s_o):(a)$
10	Märgi omistamine	10 *a*	$(s)<0$	$(s)=(s_o)\cdot\operatorname{sign}(a)$ *
11	Registris oleva arvu nihutamine	11 0	$(s)=0$	(r) nihutamine $(s_o\cdot 2^{17}$ koha võrra; kui $(s_o)<0$, siis paremale)
12	Osa eraldamine	12 *a*	$(s)=0$	$(s)_i=(s_o)_i\wedge(a)_i$
13	Formeerimine	13 *a*	$(s)=0$	$(s)_i=(s_o)_i\vee(a)_i$
14	Võrdlemine	14 *a*	$(s)\neq 0$	$(s)_i=(s_o)_i\eqsim(a)_i$
15	Normaliseerimine	15 *a*	$(s_o)=0$	Pesasse *a* saadetakse (s_o) normaliseeritud kujul, summaatorisse jääb nihutatud kohtade arv

* Sümbol $\operatorname{sign} x$ tähendab arvu x märki, s. t. $x>0$ puhul $\operatorname{sign} x=+1$ ja $x<0$ puhul $\operatorname{sign} x=-1$.

Tehte kood	Tehte nimetus	Käsu kuju	Signaal $\omega = 1$	Tehte sisu selgitus
16	Mälusse saatmine	16 *a*	$(s_0) < 0$	$(a) = (s) = (s_0)$
17	Registrisse saatmine	17 *a*	$(r) = 0$	$(r) = (a)$; $(s) = (s_0)$
20	Arvu saatmine summaatorisse	20 *k*	———	$(s) = k \cdot 2^{-17}$
21	Suunamine	21 *a*	Säilib	Kui eelmise tehte puhul oli $\omega = 1$, siis tuleb täitmisele käsk (a), kui aga oli $\omega = 0$, siis jätkub käskude täitmine loomulikus järjekorras
22	Tingimusteta suunamine	22 *a*	Säilib	Järgmisena täidetakse käsk (a)
23	Võtmesse suunamine	23 *a*	Säilib	Kui võti nr. *k* on sisse lülitatud, siis jäetakse järgmine käsk vahele
24	Tsükli lõpp	24 *a*	———	Juhtimine antakse tagasi käsule (a)
25	Tsükli algus	25 *n*	———	Programmi osa kuni järgmise käsuni 24 *a* kuulub kordamisele
26	Kontroll-liitmine	26 *a*	———	Teostatakse liitmine $(s) = (s_0) + (a)$ ringülekandega, kusjuures signaal φ on automaatselt blokeeritud
30	Käskude muutmine	30 *a*	Säilib	Järgmisele käsule liidetakse arv (a)
31 01 00	Perfolindilt lugemine	31 *a* 01 *c* 00 *b*	———	Perfolindi tsoonis nr. *c* olevad arvud viiakse pesadesse *a*, ..., *b*

Tehte kood	Tehte nimetus	Käsu kuju	Signaal $\omega=1$	Tehte sisu selgitus
31 02 00	Magnetlindilt lugemine	31 a 02 c 00 b	—	Magnetlindi tsoonis nr. c olevad arvud viiakse pesadesse $a, \ldots, b$
31 03 00	Magnetlindile salvestamine	31 a 03 c 00 b	—	Pesades $a, \ldots, b$ olnud arvud viiakse magnetlindi tsooni nr. c
32	Arvu trükkimine	32 0	—	Trükitakse või perforeeritakse arv (s_o)
34	Reavahe trükkimine	34 0	—	Paberilinti trükkimisseadmes nihutatakse ühe rea võrra edasi
37	Peatus	37 a	—	Masin peatub, $(s)=(a)$

juhtimisseadmesse siis, kui see tulemus on negatiivne. Ületäitumine ($\varphi=1$) tekib siis, kui $|(s)|\geqslant 1$. Ilmselt ei saa ületäitumist esineda tehete nr. 02, 04, 06 ja 10 puhul (küll aga tehete nr. 01, 03, 05 ja 07 korral).

Tehte nr. 05 kasutamise korral peab arvestama seda, et üks tegur tuleb registrisse viia tingimata vahetult eelneva tehtega, sest registrisse saab arvu salvestada ainult üheks tehteks. Sama märkus käib ka tehte nr. 11 kohta.

Nihutamise (tehe nr. 11) kasutamine vajab juba lähemat selgitamist. See tehe seisneb registris oleva arvu nihutamises nii mitme koha võrra, kui suur on summaatori esimeses pooles olev arv (korda 2^{17}). Summaatoris oleva arvu märk määrab sealjuures nihutamise suuna: negatiivse arvu korral toimub nihutamine paremale ja positiivse korral vasakule. Nihutamisele kuulub igal juhul ka registris oleva arvu märk, kusjuures nihutamise tulemus fikseeritakse alati summaatoris. Signaal $\omega=1$ antakse siis, kui nihutamise tulemusel summaator sisaldab ainult nulle (olgu märgitud, et nihutamisel vabanevad kohad täituvad nullidega). Nihutamise käsu aadress tehtele mõju ei avalda ja sinna kirjutatakse tavaliselt null.

Loogilised tehted nr. 12, 13 ja 14 teostab masin otsekoodis, kuid tulemus fikseeritakse summaatoris modifitseeritud pöördkoodis. Kui nende tehete puhul kasutatakse lühikesi arvusid, siis masina konstruktsioonist tingituna peab vastava lühikese pesa aadress tingimata olema paarisarvuline.

Olgu näiteks tarvis välja eraldada summaatoris oleva arvu

$$(s_o) = 11{,}011010110 \ldots 0$$

(see on arvu $-0{,}100101001 \ldots 1$ modifitseeritud pöördkood) neli esimest numbrikohta koos märgiga. Selleks tuleb kasutada mingis pesas a salvestatavat abiarvu $(a) = 1{,}111100 \ldots 0$ (plangil tuleb see arv üles kirjutada kujul 76 0000). Käsu 12 a toimel saame siis

$$(s) = 11{,}011011111 \ldots 1$$

(arvu $-0{,}100100000 \ldots 0$ modifitseeritud pöördkood).

Käsk 15 a teostab järgmise tehte: kui summaatoris olev arv on ühest suurem (s. t. eelmise tehte ajal toimus ületäitumine), siis nihutatakse ta numbrikohad ühe võrra paremale; kui aga summaatoris olev arv on absoluutväärtuselt väiksem kui pool, siis nihutatakse ta numbrikohad nii mitme koha võrra vasakule, kui mitu nulli on tema otsekoodis koma ja esimese tüvenumbri vahel. Nihutamise tulemus viiakse igal juhul pesasse a, summaatorisse aga fikseeritakse nihutatud kohtade arv korda 2^{-17} (kui nihutamine toimus paremale, siis märgiga $+$, kui vasakule, siis märgiga $-$). Signaal $\omega = 1$ tekib juhul, kui nihutatav arv võrdus nulliga. Näiteks arvu

$$(s_o) = 11{,}1111001011100100001010011101010100$$

normaliseerimisel käsuga 15 a saame:

$$(a) = 1{,}1101000110111101011000101010110000 0$$
$$(s) = 11{,}111111111111110101111111111111111.$$

Tehted nr. 16 ja 17 ei vaja erilist selgitamist. Rõhutame ainult, et ülekantav arv salvestatakse otsekoodis. Tehte nr. 16 puhul on veel oluline teada, et üleviidav arv säilib ka summaatoris.

Käsu 20 k toimel summaator kustutatakse ja tema kohtadele nr. 8—18 kirjutatakse 11-kohaline kahendarv k, arvu k märk kantakse summaatoris ikka kohtadele nr. 0 ja —1 (arv fikseeritakse summaatoris modifitseeritud pöördkoodis). Näiteks kaheksandarvu —1437 saatmiseks summaatorisse tuleb kasutada käsku 20 5437, mille tulemusel saame

$$(s) = 11{,}1111110011100000111 \ldots 11.$$

Vaadeldavat käsku kasutatakse varem teadaoleva konstandi summaatorisse viimiseks. Näiteks kui pikas pesas nr. 5016 olevat arvu

$$(5016) = 1{,}101110101000 \ldots 0$$

on tarvis nihutada kolme koha võrra vasakule, siis saab seda teostada käskudega

20 0003
17 5016
11 0000,

mille tulemusel saame summaatoris

$$(s) = 11{,}00101011111 \ldots 1.$$

Järgmised viis käsku ei teosta mingeid arvutusi, vaid on määratud ainult masina töö juhtimiseks. Tähtsaimaks neist on suunamiskäsk (tehe nr. 21), mis muudab käskude täitmise loomulikku järjekorda sõltuvalt eelmise tehte tulemusest. Näiteks kui programm sisaldab käsud

. . . .
(0521) = 02 5232
(0522) = 03 5246
(0523) = 21 0557
. . . .

siis juhul, kui vahe (5232)—(5246) osutus mittenegatiivseks (s. t. kui $\omega = 0$), annab käsk (0523) juhtimise üle järgmisele käsule (0524). Kui aga (5232)—(5246) < 0 (sel juhul $\omega = 1$), siis tuleb pärast käsku (0523) täita käsk (0557), s. t. kakskümmend seitse käsku jäetakse vahele.

Käsk 22 a annab juhtimise üle käsule (a) sõltumatult eelmise tehte tulemust iseloomustavast signaalist ω.

Käsu 23 k aadressiks võib olla üks arvudest 0001, 0002, . . . , 0007 vastavalt seitsmele «võtmele» (lülitile) juhtimispuldil. Seda käsku kasutatakse peamiselt masina töö või programmide õigsuse kontrollimisel.

Tsükli alguse ja lõpu käskude (nr. 25 ja 24) mõjul toimub nende vahel paiknevate käskude korduv täitmine. Kordamiste arvu määrab 25 n aadress: käske täidetakse kokku $n+1$ korda (n on kaheksandsüsteemis!). Näiteks kui on tarvis leida pikas pesas 4326 oleva arvu kümnes aste pesasse 4330, siis võib seda teostada käskudega

$$
\begin{aligned}
(0054) &= 02\ 4326\\
(0055) &= 25\ 0010\\
(0056) &= 06\ 4326\\
(0057) &= 24\ 0056\\
(0060) &= 16\ 4330
\end{aligned}
$$

Tsükli käskude korduv täitmine on aga teostatav ka nii, et iga kordamise ajal teatavate käskude aadressid suurenevad ühe või kahe võrra. Sel juhul varustatakse muutmisele kuuluvad käsud märgiga miinus (tehte koodi jaoks määratud kuuest kohast jäi ju üks tegelikult kasutamata, see ongi määratud märgi kujutamiseks). Käsu 25 n aadress n kujutab endast nüüd vahet muutmisele kuuluvate aadresside suurima ja vähima väärtuse vahel. Kui aadresse on igal kordamisel tarvis suurendada kahe võrra (näiteks järjestikuste pikkade pesade kasutamisel), siis tuleb nimetatud vahele liita veel 4000. Programmis kirjutatakse muudetava aadressi kohale tema suurim väärtus (kuigi käskude täitmist alustab masin aadressi vähimast väärtusest). Märgime veel, et käsu 24 a aadressiks võib olla mistahes käsu asukoht käskude 25 n ja 24 a vahelt.

Olgu näiteks tarvis leida pesades 4772, 4774, ..., 5124 oleva neljakümne kuue arvu summa pesasse 5304. Seda arvutust teostavad käsud

$$
\begin{aligned}
(0175) &= 25\ 4130\\
(0176) &= 02\ 4772\\
(0177) &= -01\ 5124\\
(0200) &= 24\ 0177\\
(0201) &= 16\ 5304.
\end{aligned}
$$

Kordamiste arv n on siin määratud järgmiselt:

$$n=5124-4774+4000=4130.$$

Käsu 30 *a* toimel viiakse lühikeses pesas *a* olev arv erilisse käskude registrisse (juhtimisseadmes) ja liidetakse järgnevana täitmisele tuleva käsu aadressiga. Olgu näiteks

$$(a)=0{,}000000000010110110$$

(kaheksandsüsteemis 00 0266) ja järgnegu käsule 30 *a* käsk 06 4132. Siis pärast käsu 30 *a* täitmist on käskude registris (mitte mälus!) käsk 06 4420. Kui aga pesas *a* oli arv

$$(a)=1{,}000000000001010100$$

(kaheksandsüsteemis −00 0124 või 40 0124), siis täidetakse järgmisena käsk 06 4006.

Käsu nr. 32 toimel (aadress selle käsu sisu ei mõjuta) trükitakse summaatoris olev arv. Trükkimine kestab 60 töötakti (0,6 sekundit) ja sõltuvalt vastava lüliti asendist juhtimispuldil toimub kas kaheksand- või kümnendsüsteemis. Trükkimise asemel võib toimuda ka arvu perforeerimine. Sel ajal kui toimub arvu trükkimine, jätkab masin järgmiste käskude täitmist.

Ülejäänud käsud nähtavasti lähemat selgitamist ei vaja, pealegi ei kasuta me neid edaspidises käsitluses.

Arvutusmasinas M-3 teostatavad tehted. Masina mäluks on magnettrummel, mis sisaldab 2048 31-kohalist pesa. Pesade aadressid kirjutatakse üles neljakohaliste kaheksandarvudega 0000—3777 (tarbe korral võib masinasse juurde monteerida veel teise trumli aadressidega 4000—7777).

Arvutusmasin M-3 on kaheaadressiline, käske täidab ta loomulikus järjekorras. Käsu salvestamisel pesasse on kohad jaotatud järgmiselt: esimene koht jääb enamasti kasutamata (sinna kirjutatakse tavaliselt märk +), järgmisel kuuel kohal salvestatakse tehte kood ja ülejäänutel kaks 12-kohalist aadressi. Kirjutatult võib käsk omada näiteks kuju

$$+41\ 3471\ 2634.$$

Tehte koodi kahest kaheksandnumbrist määrab teine üldiselt tehte enda, esimene aga täpsustab seda tehet. Nii tähendab teine number 0 liitmist, 1 lahutamist, 2 jagamist, 3 korrutamist ja 6 loogilist korrutamist. Tabelis 5 on toodud näitena liitmise kõik variandid sõltuvalt tehte koodi esimesest numbrist (täpselt samuti varieeruvad ülejäänud neli tehet).

Tabel 5

Tehte kood	Käsu kuju	Tehte sisu selgitus
00	$+00\ a\ b$	$(r) = (b) = (a_o) + (b_o)$
10	$+10\ a\ b$	$(r) = (a_o) + (b_o)$
20	$+20\ a\ b$	$(r) = (b) = (r_o) + (a_o)$
30	$+30\ a\ 0$	$(r) = (r_o) + (a_o)$
40	$+40\ a\ b$	$(r) = (b) = (a_o) + (b_o)$; (r) trükitakse
50	$+50\ a\ b$	$(r) = \lvert(a_o)\rvert + \lvert(b_o)\rvert$
60	$+60\ a\ b$	$(r) = (b) = (r_o) + (a_o)$; (r) trükitakse
70	$+70\ a\ 0$	$(r) = \lvert(r_o)\rvert + \lvert(a_o)\rvert$

Kirjutis (a_o) tähendab pesas a enne tehte sooritamist olnud arvu, (a) aga sellesse pesasse pärast tehet salvestatavat arvu. Sümbol (r) tähendab registris (aritmeetilises seadmes on kokku 4 registrit, kuid r all on alati mõeldud üht kindlat neist) pärast tehte sooritamist olevat arvu ning (r_o) seal varem olnud arvu.

Käskude niisuguse varieerimise võimalus lubab kasutada väga mitmekesiseid programmeerimismeetodeid ning muudab masina paindlikuks. Kuigi osa käske on tegelikult üheaadressilised (näiteks 30 ja 70), on teised aga peaaegu võrdsed kolmeaadressilistega. Kõike seda arvestades võib arvutusmasina M-3 käskudesüsteemi lugeda üpris õnnestunuks.

Peale nimetatud viie põhitehte on masinas ette nähtud veel terve rida juhtimistehteid (suunamiskäsud, ülekanded, sisendtehted, masina peatamine), mis on koondatud tabelisse 6.

Tabel 6

Tehte kood	Käsu kuju	Tehte sisu selgitus
24	$+24\ a\ b$	$(r) = (b) = (r_o)$; järgmisena täidetakse käsk (a)
64	$+64\ a\ b$	$(r) = (b) = (r_o)$; (r) trükitakse; järgmisena täidetakse käsk (a)
74	$+74\ 0\ b$	$(r) = \lvert (r_o) \rvert$; järgmisena täidetakse käsk (b)
34	$+34\ a\ b$	Kui $(r_o) \geqslant 0$, siis järgmisena täidetakse käsk (b); kui $(r_o) < 0$, siis järgmisena täidetakse käsk (a)
05 15	$+05\ a\ b$ $+15\ a\ b$	$(r) = (b) = (a_o)$
45 55	$+45\ a\ b$ $+55\ a\ b$	$(b) = (a_o)$; (b) trükitakse
07 27	$+07\ 0\ b$ $+27\ 0\ b$	Arv perfolindilt salvestatakse pesasse b
04 ; 14 44 ; 54 17 ; 37 67 ; 77	näiteks $+04\ a\ b$	Masin peatub

11. ÜLESANNETE PROGRAMMEERIMINE

Programmi koostamise käik. Mingi matemaatilise ülesande lahendamiseks elektron-arvutusmasinal tuleb see ülesanne kõigepealt programmeerida, s. t. moodustada käskude jada (programm), mille täitmise tulemusena masin lahendab püstitatud ülesande. Ülesannete programmeerimine kujutab endast küllaltki keerulist ja aeganõudvat tööd ning selle põhjalik selgitamine pole siin võimalik. Järgnevas piirdume ainult programmeerimise lihtsamate põhimõtete selgitamisega mõne elementaarse näite varal. Sealjuures eeldame algul, et ülesande lahendamise käik on juba esitatud arvutamiseks sobiva algoritmi (lahendamiseeskirja) kujul. Siis seisneb programmeerimine lihtsalt selles algoritmis esinevate tehete «tõlkimises masina keelde», s. t. nende üleskirjutamises käskudena. Konkreetsuse huvides kasutame käesolevas paragrahvis arvutusmasina «Ural» käskude süsteemi.

Arvutusmasina «Ural» kui fikseeritud komaga masina puhul lisandub lahendusalgoritmi valikule veel sobivate mastaapide otsimine, mis muudaksid kõik arvud absoluutväärtuselt ühest väiksemaks. Esimeste toodavate näidete puhul me aga eeldame, et sobivad mastaabid on lahendamisel esinevate arvude jaoks juba leitud, nii et ületäitumist ei toimu.

Programmi enda koostamisele asumisel tuleb kõigepealt selgitada tehete sooritamise kõige otstarbekam järjekord. Selle järjekorra selgitamise käigus jaotatakse ülesande lahendamine tavaliselt võimalikult iseseisvateks osadeks — b l o k k i d e k s. Need blokid nummerdatakse, kujutatakse näiteks ristkülikutena ja ühendatakse nooltega, mis näitavad blokkide täitmise järjekorda. Nii saame programmi jaoks nn. b l o k k - s k e e m i. Edasi koostatakse käsud juba iga bloki jaoks eraldi ja ühendatakse nad siis programmiks nooltega määratud järjekorras.

Ka iga bloki programmeerimisel tuleb muidugi silmas pidada tehete otstarbekamat järjekorda. Eeskätt peab siin arvestama järgmisi asjaolusid.

1. Tehete sooritamise järjekord on soovitav valida nii, et eelmise tehte tulemus võimalikult võtaks osa järgmisest tehtest. Sel teel saab enamasti vähendada ebaproduktiivsete tehete 02 (summaatorisse saatmine) ja 16 (mälusse saatmine) arvu. Ühtlasi peab aga jälgima, et ühe ja sama suuruse arvutamine (kui ta valemites esineb mitmes kohas) ei toimuks mitu korda. Niisugusel juhul on kord leitud suuruse hoiulesaatmine loomulikult otstarbekam.

2. Tehete järjekorra valikul tuleb hoolitseda ka selle eest, et arvutustel ei tekiks ületäitumist. Näiteks valemi

$$A+\frac{B\cdot C}{D}-E$$

järgi arvutamisel (siin on eeldatud, et kõik arvud on positiivsed) on soovitav kasutada niisugust tehete järjekorda:

$$B \cdot C; \quad \frac{B \cdot C}{D}; \quad \frac{B \cdot C}{D} - E; \quad \frac{B \cdot C}{D} - E + A.$$

Lähemalt peatume sedalaadi küsimuste juures veidi hiljem.

Enne üksikute blokkide programmeerimist tuleb veel määrata kindlaks ülesande lahendamiseks vajaliku materjali (käskude, algandmete, vahepealsete tulemuste jne.) paigutus masina mälus. Sellega seoses kerkib üles järgmine raskus. Nimelt enne käskude koostamist ei ole võimalik otstarbekalt jaotada mälu, sest pole teada ei programmi käskude arvu ega ka vahepealsete arvutustulemuste salvestamiseks vajalike pesade arvu. Teiselt poolt aga, kui me ei tea, millistes pesades salvestatakse algandmeid ning abiarvusid ja millistesse pesadesse tuleb paigutada arvutustulemused, siis ei saa me ka koostada programmi käske.

Sellest raskusest ülesaamiseks kasutatakse programmeerimisel pesade aadresside asemel esialgu tähelisi sümboleid, mis alles lõpuks asendatakse konkreetsete arvudega. Olgu muide märgitud, et selline tähistusviis hõlbustab ka programmi hilisemat muutmist ja parandamist.

Edaspidises kasutame järgmist tähistusviisi. Programmi käsud paigutame pesadesse $a+0000$, $a+0001, \ldots$, ülesande lahendamiseks vajalikud abiarvud pesadesse $b+0000$, $b+0001, \ldots$, algandmed pesadesse $c+0000$, $c+0001, \ldots$, vahepealsed arvutustulemused pesadesse $d+0000$, $d+0001, \ldots$ ning lõpptulemused pesadesse $g+0000$, $g+0001, \ldots$ Kõik arvulised liidetavad kirjutame siinjuures alati kaheksandsüsteemis. Täheliste liidetavate kohta aga eeldame[1], et nad tähendavad p a a r i s a r v u s i d ja on väiksemad kui 4000. Paarisarvulisuse nõudmine on tarvilik kahel põhjusel: 1) tehete nr. 12, 13 ja 14 korral võib masinas kasutada vaid paarisarvulisi aadresse, mistõttu tähelisest sümboolikast peab aadressi paarisus silma paistma; 2) pikkade pesade aadressid on alati paarisarvulised. Nõue, et tähelised liidetavad oleksid väiksemad kui 4000, on vajalik vahe tegemiseks pikkade ja lühi-

[1] Need eeldused on vajalikud ainult seetõttu, et me järgnevates näidetes kasutame arvutusmasina «Ural» käskude süsteemi.

keste pesade vahel. Nimelt on pikkade pesade aadresse sobiv üles kirjutada kujul $c+4000$, $c+4002$, ... Siin näiteks pikk pesa $c+4002$ koosneb lühikestest pesadest $c+0002$ ja $c+0003$.

Illustreerime kirjeldatud protsessi kõigepealt ühe konkreetse näitega, milles arutlused viime läbi võimalikult üksikasjaliselt.

Näide. Koostame programmi funktsiooni

$$F(x)=\frac{Af(x)+Bx}{f(x)+C}$$

väärtuse arvutamiseks, kus abifunktsioon $f(x)$ on defineeritud valemitega:

$$f(x)=\begin{cases} f_1(x)=Dx^2+Ex, \text{ kui } x\leqslant x_o; \\ f_2(x)=Gx^2+Hx, \text{ kui } x>x_o. \end{cases}$$

Toodud valemites esinevate kordajate A, B, C, D, E, G, H ja konstandi x_o väärtused olgu ette antud [1], kuid argumendi x väärtus selgugu antud valemite kasutamisele eelnevate arvutuste käigus.

Et selle ülesande lahendamisel valemi valik abifunktsiooni $f(x)$ väärtuse arvutamiseks sõltub oluliselt argumendi x suurusest, siis ei saa me seda valikut teostada programmeerimisel, vaid see selgub alles arvutuste käigus. Antud ülesande lahendamist tuleb seetõttu alustada võrratuse $x\leqslant x_o$ täidetuse kontrollimisega ja vastavalt selle kontrolli tulemusele arvutada $f(x)$ kas esimesest või teisest valemist. Me saame järelikult n. ö. h a r u n e v a p r o g r a m m i, mille kokkuvõttes võib jaotada järgmisteks osadeks (blokkideks):

1. Kontrollimine, kas $x\leqslant x_o$ või $x>x_o$.
2. $f_1(x)$ arvutamine.
3. $f_2(x)$ arvutamine.
4. $F(x)$ arvutamine.

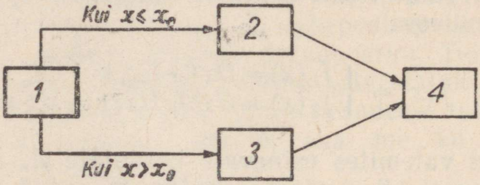
Programmi blokk-skeem on toodud joonisel 71.

Algandmed paigutame järgmiselt:

[1] Antud arvude kohta eeldame, et nad on kõik oma absoluutväärtuselt ühest väiksemad ning et ka arvutuste käigus ületäitumist ei teki. Samuti eeldame, et need arvud on juba teisendatud kahendsüsteemi.

$(c+4000)=A$, $(c+4002)=B$, $(c+4004)=C$,
$(c+4006)=D$, $(c+4010)=E$, $(c+4012)=G$,
$(c+4014)=H$, $(c+4016)=x_o$.

Argument x olgu eelnevates arvutustes[1] saadud pesasse $d+4000$. Vahepealsete tulemuste paigutamiseks kasutame pesasid $d+4002$, $d+4004$, ... Tuleb ainult silmas pidada, et nendesse nn. t ö ö p e s a d e s s e võime uue arvu viia alles siis, kui me enam ei vaja seal varem hoiul olnud arvu. Eksituste vältimiseks on seepärast keeruliste ülesannete puhul kasulik välja märkida tööpesade sisu pärast iga käsu täitmist. Vaadeldava ülesande puhul me seda aga ei tee.



Joonis 71

Asume nüüd üksikute blokkide programmeerimisele.

E s i m e n e b l o k k.

1. käsk: «tuua arv x_o summaatorisse»

$$(a+0000)=02\ c+4016.$$

2. käsk: «lahutada summaatoris olevast arvust x»

$$(a+0001)=03\ d+4000.$$

3. käsk: «kui $x_o-x<0$, siis anda juhtimine üle kolmanda bloki esimesele käsule $(a'+0000)$, $x_o-x\geqslant 0$ korral aga teise bloki esimesele käsule $(a+0003)$»

$$(a+0002)=21\ a'+0000.$$

Et me esialgu ei tea, millistesse pesadesse paiknevad kolmanda bloki käsud, siis kasutame nende jaoks ajutist tähistust $a'+0000$, $a'+0001$, ...

[1] Siin on eeldatud, et arvutused tuleb läbi viia «pikkade arvudega», s. t. üheksakohalise täpsusega.

T e i n e b l o k k.

1. käsk: «tuua summaatorisse argument x»

$$(a+0003)=02\ d+4000.$$

2. käsk: «korrutada summaatoris olev arv arvuga D»

$$(a+0004)=06\ c+4006.$$

3. käsk: «korrutada summaatoris olev arv arvuga x»

$$(a+0005)=06\ d+4000.$$

4. käsk: «viia registrisse argument x»

$$(a+0006)=17\ d+4000.$$

5. käsk: «korrutada registris olev arv arvuga E ja liita tulemus summaatoris oleva arvuga»

$$(a+0007)=05\ c+4010.$$

6. käsk: «viia saadud $f_1(x)$ väärtus hoiule pesasse $d+4002$»

$$(a+0010)=16\ d+4002.$$

7. käsk: «anda juhtimine üle neljanda bloki esimesele käsule $a''+0000$»

$$(a+0011)=22\ a''+0000.$$

Paneme tähele, et selle bloki töötamise tulemusel saame $f(x)$ väärtuse nii summaatorisse kui ka pesasse $d+4002$. Seda on kasulik arvestada järgmise bloki programmeerimisel, et luua seal sama olukord.

Nüüd me saime ühtlasi teada, kuhu on sobiv paigutada kolmanda bloki esimene käsk ja seega võime loobuda ajutisest tähistusest, valides $a'=a+0012$.

K o l m a s b l o k k.

1. käsk: «tuua summaatorisse argument x»

$$(a+0012)=02\ d+4000.$$

2. käsk: «korrutada summaatoris olev arv arvuga G»

$$(a+0013)=06\ c+4012.$$

3. käsk: «korrutada summaatoris olev arv arvuga x»

$$(a+0014)=06\,d+4000.$$

4. käsk: «viia registrisse argument x»

$$(a+0015)=17\,d+4000.$$

5. käsk: «korrutada registris olev arv arvuga H ja liita tulemus summaatoris oleva arvuga»

$$(a+0016)=05\,c+4014.$$

6. käsk: «viia saadud $f_2(x)$ väärtus hoiule pesasse $d+4002$»

$$(a+0017)=16\,d+4002.$$

Juhtimise üleandmist neljanda bloki esimesele käsule pole siin vaja teostada, sest võime lihtsalt valida $a''=a+0020$.

N e l j a s b l o k k.

1. käsk: «liita summaatoris olevale arvule $f(x)$ arv C»

$$(a+0020)=01\,c+4004.$$

2. käsk: «viia $f(x)+C$ ajutiselt hoiule pesasse $d+4004$»

$$(a+0021)=16\,d+4004.$$

3. käsk: «tuua summaatorisse arv $f(x)$»

$$(a+0022)=02\,d+4002.$$

4. käsk: «korrutada summaatoris olev arv arvuga A»

$$(a+0023)=06\,c+4000.$$

5. käsk: «viia registrisse argument x»

$$(a+0024)=17\,d+4000.$$

6. käsk: «korrutada registris olev arv arvuga B ja liita tulemus summaatoris oleva arvuga»

$$(a+0025)=05\,c+4002.$$

7. käsk: «jagada summaatoris olev arv arvuga $f(x)+C$»

$$(a+0026)=07\,d+4004.$$

8. käsk: «viia saadud $F(x)$ väärtus hoiule pesasse $d+4004$»

$$(a+0027)=16\ d+4004.$$

Juhul kui $F(x)$ võtab edasistes arvutustes osa ainult vahetult järgnevast tehtest, siis võib viimase käsu muidugi ära jätta. Kui aga $F(x)$ kujutabki endast arvutuste lõpptulemust, siis tuleb veel lisada programm, mis teisendab ta kümnendsüsteemi ja trükib välja. Kõige lõpuks tuleb sel juhul peatada masin.

Asendades saadud käskudes tähelised aadressid konkreetsete arvudega saamegi meie funktsiooni väärtuse arvutamise programmi lõplikul kujul.

Loomulikult ei ole mõtet kasutada elektron-arvutusmasinat ainuüksi sellise lihtsa funktsiooni väärtuse arvutamiseks. Seepärast eeldame, et tegemist on ühe ulatuslikuma programmi väikese osaga ja valime vastavalt ka koha mälus. Programmi paigutame mälusse alates näiteks [1] pesast nr. 0230, algandmed alates pesast 5300, tööpesadeks aga eraldame mälu viimased pesad. Sellega seoses valime

$$a=0230, \qquad c=1300, \qquad d=3770,$$

mille tulemusena saame programmi, mis on toodud tabelis 7 (algandmete jaoks määratud pikad pesad on seal kirjutatud kahe lühikese pesa näol).

Sellest tabelist ja kogu eelnevast arutelust peaks torkama silma teatav ebaökonoomsus teise ja kolmanda bloki programmeerimisel. Nimelt koosnevad need blokid tegelikult ühesugustest käskudest, erinevus on ainult selles, et teises blokis kasutatakse arvusid D ja E, kuid kolmandas blokis arvusid G ja H. Kas ei saaks nende blokkide ühendamise teel programmi lühendada?

Osutub, et see on võimalik käsu nr. 30 kasutamise teel. Selleks eraldame mingi lühikese pesa, näiteks 1320, erilise ümberadresseerimiskonstandi jaoks (algul viime sellesse pesasse nulli) ja jaotame kogu töö järgmisteks blokkideks:

1. Kontrollimine, kas $x \leqslant x_0$ või $x > x_0$.

[1] Terviklik programm paigutatakse mälusse tavaliselt alates võimalikult väiksema aadressiga pesast (ainult mõned esimesed pesad reserveeritakse erilise sisendprogrammi jaoks).

Tabel 7

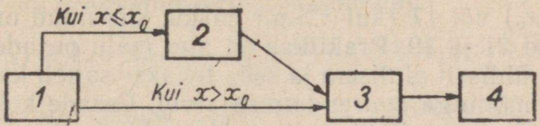
Pesa	Pesas salvestatav käsk	Märkused
0230	02 5316	Esimene blokk
0231	03 7770	
0232	21 0242	
0233	02 7770	Teine blokk
0234	06 5306	
0235	06 7770	
0236	17 7770	
0237	05 5310	
0240	16 7772	
0241	22 0250	
0242	02 7770	Kolmas blokk
0243	06 5312	
0244	06 7770	
0245	17 7770	
0246	05 5314	
0247	16 7772	
0250	01 5304	Neljas blokk
0251	16 7774	
0252	02 7772	
0253	06 5300	
0254	17 7770	
0255	05 5302	
0256	07 7774	
0257	16 7774	
1300 1301	*A*	Lähteandmed
1302 1303	*B*	
1304 1305	*C*	
1306 1307	*D*	
1310 1311	*E*	
1312 1313	*G*	
1314 1315	*H*	
1316 1317	x_0	

2. Ümberadresseerimiskonstandi vähendamine nelja võrra.

3. $f(x)$ arvutamine.

4. $F(x)$ arvutamine.

Blokk-skeem omandab nüüd joonisel 72 toodud kuju.



Joonis 72

Üksikutele blokkidele vastavad programmi osad võib koostada näiteks järgmiselt.

Esimene blokk:

(0230) = 02 5316 (viia x_0 summaatorisse)
(0231) = 03 7770 (arvutada $x_0 - x$)
(0232) = 21 0235

Teine blokk:

(0233) = 20 4004 (viia summaatorisse $-4 \cdot 2^{-17}$)
(0234) = 16 1320 (viia see konstant pesasse 1320)

Kolmas blokk:

(0235) = 02 7770 (viia x summaatorisse)
(0236) = 30 1320 (muuta järgmise käsu aadressi)
(0237) = 06 5312 (arvutada Gx või Dx)
(0240) = 06 7770 (arvutada $f(x)$ esimene liidetav)
(0241) = 30 1320 (muuta järgmise käsu aadressi)
(0242) = 17 5314 (viia registrisse H või E)
(0243) = 05 7770 (arvutada $f(x)$)
(0244) = 16 7772 (viia $f(x)$ hoiule pesasse 7772)

Neljas blokk:

(0245) = 01 5304 (arvutada $f(x) + C$)
(0246) = 16 7774 (viia $f(x) + C$ hoiule)
(0247) = 02 7772 (tuua summaatorisse $f(x)$)
(0250) = 06 5300 (arvutada $Af(x)$)
(0251) = 17 7770 (viia x registrisse)
(0252) = 05 5302 (arvutada $Af(x) + Bx$)
(0253) = 07 7774 (arvutada $F(x)$)
(0254) = 16 7774 (viia $F(x)$ hoiule).

Saadud programm sisaldab 3 käsku vähem kui varemkoostatu. Arvestades veel, et üks pesa tuli nüüd reserveerida ümberadresseerimiskonstandi jaoks, on kokkuhoid seega 2 lühikest pesa. Teiselt poolt oleme aga kaotanud masina tööajas. Nimelt kui tabelis 7 toodud programmi järgi tuleb $F(x)$ väärtuse arvutamiseks täita kas 18 (kui $x \leqslant x_0$) või 17 (kui $x > x_0$) käsku, siis nüüd on vastavad arvud 21 ja 19. Praktikas on aga mälu pesade kokkuhoidmine üldiselt olulisem ja seda tehakse sageli ka masina tööhulga mõninga suurendamise arvel. Pesade kokkuhoidmise vajaduse tingib sisemise mälu suhteliselt väike mahtuvus, sest on äärmiselt oluline, et kogu ülesande lahendamiseks vajalik materjal sinna täielikult ära mahuks.

Tsüklite programmeerimine. Arvutusprotsessi nimetatakse tsükliliseks, kui ta seisneb korduvas arvutamises ühtede ning samade valemite järgi, kusjuures igakordsel arvutamisel kasutatakse uusi lähteandmeid. Niisuguse arvutusprotsessi korduvaid etappe nimetatakse tsükliteks.

Tsüklilise arvutusprotsessi programmeerimisel püüame ka programmi koostada tsüklilisena, s. t. sellisena, milles teatud käskuderühma tuleb täita korduvalt. Kui tsükli kordamiste arv on ette teada (nagu me hiljem näeme, pole see mitte alati nii!), siis saab muidugi koostada ka üksikasjaliselt väljakirjutatud programmi, kuid reeglina tuleb see väga pikk ning võib isegi masina mälusse mitte ära mahtuda.

Tsüklilise programmi koostamine toimub põhimõtteliselt järgmiselt. Kõigepealt moodustatakse käsud, mis realiseerivad ühes tsüklis ettenähtud arvutused (s. t. tehted, milledest tsükkel koosneb). Et tsükli iga järgneva rakendamise puhul tuleb üldiselt kasutada uusi lähteandmeid, mis erinevad eelmisel rakendamisel kasutatud andmetest, siis võib osutuda vajalikuks koostada käsud, mis muudavad (iga uue kordamise eel) tsükli käskudes lähteandmete aadresse[1]. Seejärel peavad programmis olema käsud,

[1] Kui tsükli kordamisel kasutatavad lähteandmed leitakse tsükli eelmise rakendamise tulemusena (kusjuures edasiste arvutuste käigus meid huvitavad ainult tsükli n-kordse täitmise teel saadavad lõpptulemused), siis igakordse käskude ümberadresseerimise asemel on otstarbekam salvestada lähteandmed kindlatesse pesadesse, kus nad töö käigus asenduvad vahepealsete arvutustulemustega. Seda võtet on kasutatud ka järgnevates näidetes.

mis kontrollivad kordamiste arvu, ning lõpuks suunamiskäsk, mis annab juhtimise tagasi tsükli esimesele käsule või edasistele arvutustele sõltuvalt juba sooritatud kordamiste arvust (s. t. kordamiste arvu kontrollimisel saadud signaalist ω). Arvutusmasina «Ural» puhul lihtsustab tsükli programmeerimist käskude nr. 25 ja 24 olemasolu.

Näitena programmeerime polünoomi

$$p(x)=A_n+A_{n-1}x+A_{n-2}x^2+\ldots+A_1x^{n-1}+A_0x^n$$

väärtuse arvutamise kohal $x=x_0$. Sobiva skeemi arvutuste läbiviimiseks saame, kui kirjutame polünoomi kujul [1]

$$p(x)=(\ldots(((A_0x+A_1)x+A_2)x+A_3)x+\ldots+A_{n-1})x+A_n$$

ning kasutame arvutamisel valemeid

$$\begin{aligned} p_1&=A_0x_0+A_1\\ p_2&=p_1x_0+A_2\\ p_3&=p_2x_0+A_3\\ &\ldots\ldots\\ p_{n-1}&=p_{n-2}x_0+A_{n-1}\\ p(x_0)&=p_n=p_{n-1}x_0+A_n. \end{aligned}$$

Kui veel tähistada $p_0=A_0$, siis omandavad kõik need valemid kuju $p_i=p_{i-1}x_0+A_i$ (kus $i=1, 2, \ldots n$) ning arvutamine taandub selle valemi n-kordsele rakendamisele.

Algandmed paigutame näiteks järgmiselt:

$$(c+4000)=A_0,\ (c+4002)=A_1, \ldots,\ (c+4000+2n)=A_n,$$
$$(d+4000)=x_0.$$

Vahepealsete arvutustulemuste p_i salvestamiseks me eraldi pesa ei reserveeri, vaid kasutame selleks otstarbeks summaatorit.

Kuna $p_0=A_0$, siis peab programm algama suuruse A_0 viimisega summaatorisse. Tsükkel ise koosneb ainult kahest käsust: summaatoris oleva arvu korrutamine x_0-ga ja tulemusele järjekordse A_i juurdeliitmine. Kordamiste arvu reguleerimiseks kasutame käske 25 ja 24.

---

[1] Niisugust võtet polünoomi väärtuste arvutamiseks nimetatakse tavaliselt Horneri skeemiks.

Lõpuks tuleb veel lisada käsk, mis viib vastuse pesasse $g+4000$. Kokku võttes omab soovitud programm (tähelistes aadressides) kuju:

$$
\begin{array}{ll}
(a+0000)=02 & c+4000 \\
(a+0001)=25 & 4000+2n-2 \\
(a+0002)=06 & d+4000 \\
(a+0003)=-01 & c+4000+2n \\
(a+0004)=24 & a+0002 \\
(a+0005)=16 & g+4000.
\end{array}
$$

Kui mingil põhjusel ei ole otstarbekas kasutada tsükli alguse ja lõpu käske[1], siis võib kordamiste arvu reguleerida ka erilise loendaja kasutamise teel. Kordamiste arvu loendajaks kasutatakse lühikest pesa, milles sisalduva arvu väärtust muudetakse iga kordamise puhul kindla arvu ühikute (tavaliselt ühe või kahe) võrra. Kui loendajas olev arv omandab teatava etteantud väärtuse, siis antakse juhtimine tsüklist välja.

Näiteks kui loendaja $b+0000$ algväärtuseks valida null ja pesasse $b+0001$ paigutada lõppväärtus $n \cdot 2^{-17}$, siis pesades $a+0000, \ldots, a+k$ paikneva tsükli n-kordse kordamise saab realiseerida käskudega

$$
\begin{array}{ll}
(a+0000)= \ldots\ldots\ldots & \\
\ldots\ldots\ldots\ldots\ldots\ldots & \\
(a+k+0000)= \ldots\ldots\ldots & \\
(a+k+0001)=20\ 0001 & \\
(a+k+0002)=01 & b+0000 \\
(a+k+0003)=16 & b+0000 \\
(a+k+0004)=14 & b+0001 \\
(a+k+0005)=21 & a+0000.
\end{array}
$$

Kui kirjeldatud tsükkel kuulub mõnda teise tsüklisse, siis loendaja algseisu taastamise võib teostada käsuga

$$(a+k+0006)=16\ b+0000,$$

mis viib loendajasse võrdlemise tulemuse — nulli.

---

[1] Sellise olukorraga on näiteks tegemist sel juhul, kui üks tsükkel sisaldab teist. Siis võib masina konstruktsioonist tingituna ainult ühe neist tsükleist programmeerida käskude nr. 25 ja 24 kasutamisega.

Tsükli kordamiste arvu niisuguse reguleerimise puhul on loendaja väärtust enamasti otstarbekas kasutada ühtlasi ümberadresseerimiskonstandina. Näiteks kui viimativaadeldud polünoomi väärtust on tarvis arvutada mitte ainult ühes punktis x_0, vaid terves reas punktides $(d+4000)=x_0$, $(d+4002)=x_1, \ldots,$ $(d+4000+2k)=x_k$, siis võib seda realiseerida järgmise programmiga:

$(a+0000)=30$	$b+0000$
$(a+0001)=02$	$d+4000$
$(a+0002)=16$	$d+4000$
$(a+0003)=02$	$c+4000$
$(a+0004)=25$	$4000+2n-2$
$(a+0005)=06$	$d+4000$
$(a+0006)=-01$	$c+4000+2n$
$(a+0007)=24$	$a+0005$
$(a+0010)=30$	$b+0000$
$(a+0011)=16$	$g+4000$
$(a+0012)=20$	0002
$(a+0013)=01$	$b+0000$
$(a+0014)=16$	$b+0000$
$(a+0015)=14$	$b+0001$
$(a+0016)=21$	$a+0000$

Siin on eeldatud, et loendajasse $b+0000$ on algul sisse viidud null ja pesasse $b+0001$ arv $(2k+2)\cdot 2^{-17}$. Tööpesaks on selles programmis kasutatud esimest argumenti x_0 sisaldanud pesa (seda võib teha siis, kui x_0 järgnevatest arvutustest enam osa ei võta). Programmi täitmise tulemusel saame vastused pesadesse

$$p(x_0)=(g+4000),\, p(x_1)=(g+4002), \ldots, p(x_k)= \\ =(g+4000+2k).$$

Ligikaudse meetodi valimine. Vaatleme nüüd olukorda, kui programmeeritava ülesande täpse lahendamise käiku määrav algoritm sisaldab tehteid, mida masin ei ole võimeline vahetult sooritama. Neil juhtudel tuleb ülesande lahendamiseks kasutada mõnd ligikaudset (ehk. nn. numbrilist) meetodit, mis võimaldab lahendi leida nõutava täpsusega. See täpsus on enamasti koos ülesandega ette antud, näiteks lubatava vea ülemmäärana.

Ülesande lahendamiseks etteantud täpsusega saab aga reeglina kasutada mitmesuguseid ligikaudseid meetodeid.

Nende meetodite hulgas valiku tegemisel tuleb arvestada elektron-arvutusmasina kui arvutusvahendi iseärasusi (tehete kiire sooritamine ning suhteliselt aeglane andmete sisseviimine masinasse, tehete sooritamine suure kohtade arvuga, sisemise mälu piiratud maht jms.), mis tunduvalt mõjutavad numbriliste meetodite otstarbekuse hinnangut, võrreldes näiteks käsitsi arvutamisega. Nii osutuvad elektron-arvutusmasinal kasutamiseks enamasti eriti sobivateks nn. iteratsioonimeetodid, mis seisnevad otsitava lahendi lähisväärtuse järkjärgulises parandamises.

Näiteks võrrandi $x=f(x)$ lahendamisel hariliku iteratsioonimeetodiga lähtutakse otsitava lahendi mingist alglähendist x_0 ja leitakse järgmised lähendid $x_1, x_2, \ldots$ valemitest [1]

$$x_{n+1}=f(x_n) \quad (\text{kus } n=0, 1, 2, \ldots).$$

Illustreerime ligikaudse meetodi valikut ühe konkreetse näite varal: koostame programmi ruutjuure leidmiseks arvust $A=(d+4000)$.

Ruutjuure $x=\sqrt{A}$ leidmine on samaväärne võrrandi

$$x^2=A$$

lahendamisega. Liites selle võrrandi mõlemale poolele x^2 ja jagades siis suurusega $2x$, saame ekvivalentse võrrandi

$$x=\frac{x}{2}+\frac{A}{2x}.$$

See võrrand omab kuju $x=f(x)$ (kus $f(x)=\frac{x}{2}+\frac{A}{2x}$) ning tema lahendamiseks saab kasutada harilikku iteratsioonimeetodit [2]

[1] Olgu märgitud, et nii saadud lähendite jada $x_1, x_2, \ldots$ koondub võrrandi täpseks lahendiks leiab aset kindlasti siis, kui vaadeldavas piirkonnas $|f'(x)|<1$. Sealjuures osutub koondumine seda kiiremaks, mida väiksem on $|f'(x)|$.

[2] Antud juhul $f'(x)=0{,}5-\frac{0{,}5A}{x^2}$ ning $|f'(x)|<1$, kui vaid $3x^2>A$ (sellise olukorra saavutamiseks me võrrandit teisendamegi). Viimast võrratust tuleb arvestada alglähendi valikul ($x_0=1$ korral on ta rahuldatud).

$$x_{n+1} = \frac{x_n}{2} + \frac{A}{2x_n}.$$

Et arvutusmasinas «Ural» pole võimalik teostada kahega jagamist (arv kaks pole masinas kujutatav), siis tuleb see valem kirjutada kujul

$$x_{n+1} = 0{,}5x_n + \frac{0{,}5A}{x_n}.$$

Alglähendiks valime arvu $x_0 = 1$ (täpsemalt $x_0 = 1 - 2^{-35}$) ning parandame teda toodud valemi abil seni, kuni saavutame olukorra, et kaks järjestikust lähendit erinevad ülimalt viimase numbrikoha poolest, s. t. arvutamise lõpetame siis, kui osutub, et

$$|x_{n+1} - x_n| < \varepsilon = 2^{-34}.$$

Seega antud juhul tsükli kordamiste arv pole ette teada, vaid tuleb määrata arvutuste käigus.

Programmi koostamisele asumisel peab veel arvestama, et juhul kui $A < 0$, pole arvutamine üldse võimalik ja masin tuleb seisma panna. Masina tarbetu töö vältimiseks on veel otstarbekas välja eraldada juhtum $A = 0$, mil vastus on ette teada (kuid viimase valemi järgi arvutamisel tuleks masinal selle vastuse saamiseks sooritada peaaegu 400 tehet).

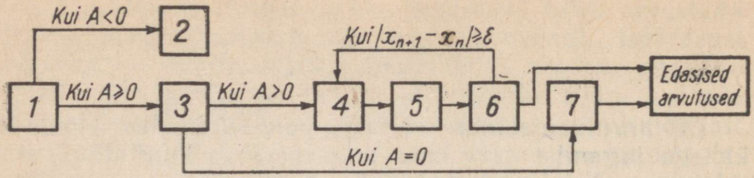
Algandmed paigutame pesadesse

$$(c+4000) = x_0,\ (c+4002) = \varepsilon,\ (c+4004) = 0{,}5,\ (c+4006) = 0$$

(null on vajalik kontrollimiseks, kas $A = 0$). Tööpesad jaotame järgmiselt: pesas $d+4000$ paiknegu juuritav arv A, pesasse $d+4002$ aga salvestame lähendid x_n. Parandatud lähendi x_{n+1} salvestamiseks kasutame pesa $c+4000$, kus algul paikneb alglähend x_0. Sellesse pesasse saame siis ka tulemuse $\sqrt{A}$.

Arvutusprotsessi võib jaotada näiteks järgmisteks blokkideks:

1) kontrollimine, kas $A < 0$,
2) masina peatamine, juhul kui $A < 0$,
3) kontrollimine, kas $A = 0$,



Joonis 73

4) eelmise lähendi viimine pesast $c+4000$ tööpesasse $d+4002$,

5) järgmise lähendi arvutamine pesasse $c+4000$,

6) pesades $c+4000$ ja $d+4002$ olevate lähendite võrdlemine,

7) nulli viimine vastuse pesasse $c+4000$ juhul, kui $A=0$.

Blokk-skeem on toodud joonisel 73.

Blokkide programmeerimisele asumisel on veel kasulik arvestada, et teise bloki asukoht programmis pole oluline. Seepärast püüame ta paigutada nii, et ei oleks tarvis lisada eraldi suunamiskäsku selle bloki vahelejätmiseks. Seda arvestades võib nõutud programmi koostada näiteks järgmiselt:

$(a+0000)=02 \quad d+4000$ (A summaatorisse)
$(a+0001)=21 \quad a+0020$ ($A<0$ puhul 2 blokki)
$(a+0002)=13 \quad c+4006$ (A võrdlemine nulliga)
$(a+0003)=21 \quad a+0021$ ($A=0$ puhul 7 blokki)
$(a+0004)=02 \quad c+4000$ (alglähend summaatorisse)
$(a+0005)=16 \quad d+4002$ (alglähend tööpesasse)

$$\left.\begin{array}{l}(a+0006)=02 \quad c+4004\\(a+0007)=06 \quad d+4000\\(a+0010)=07 \quad d+4002\\(a+0011)=17 \quad c+4004\\(a+0012)=05 \quad d+4002\\(a+0013)=16 \quad c+4000\end{array}\right\}(s)=(c+4000)=0{,}5x_n+\frac{0{,}5A}{x_n}=x_{n+1}$$

$(a+0014)=03 \quad d+4002$ $(x_{n+1}-x_n)$
$(a+0015)=04 \quad c+4002$ $(|x_{n+1}-x_n|-\varepsilon)$
$(a+0016)=21 \quad a+0022$ } arvutuste kordamine või
$(a+0017)=22 \quad a+0004$ } lõpetamine
$(a+0020)=37 \quad d+4000$ ($A<0$ puhul A juhtimispuldile)
$(a+0021)=16 \quad c+4000$ ($A=0$ puhul $\sqrt{A}=0$).

Siin A võrdlemine nulliga toimub loogilise liitmise teel. Käsk $(a+0002)$ annab tulemuseks nulli (ja signaali $\omega=1$) vaid siis, kui $A=0$. See tulemus viiaksegi käsuga $(a+0021)$ vastuse pesasse.

Tabelis 8 on sama programm koos algandmetega toodud numbrilisel kujul, kus näitena on valitud

$$a=0500, \quad c=1410, \quad d=1420.$$

Tegelikkudes arvutustes ruutjuure leidmiseks kasutatav programm erineb siintoodust eeskätt selle poolest, et seal juuritav arv A kõigepealt normaliseeritakse, s. t. asendatakse arvuga A', kus

$$A=A' \cdot 2^{-p}, \quad 0,5 \leqslant A' < 1, \quad p \geqslant 0.$$

Lähtudes ühtlasi täpsemast alglähendist

$$x_0=0,57422A'+0,42578,$$

saab siis suuruse $\sqrt{A'}$ arvutada ülaltoodud valemi kolmekordse rakendamisega. Lõpuks korrutatakse tulemust veel teguriga $2^{-\frac{1}{2}p}$.

Mastaapide valimine. Fikseeritud komaga masina kasutamise puhul tuleb kõik lähteandmed ja arvutusvalemid teisendada sellisele kujule, et nii vahepealsetes kui ka lõpptulemustes arvude absoluutväärtused jääksid ühest väiksemaks. Paljudel juhtudel on seda võimalik saavutada m a s t a a b i k o r d a j a t e valimise (ja arvutusvalemite vastava teisendamise) teel. Olgu näiteks tarvis arvutada funktsiooni

$$y=\frac{\sqrt{(931+9x^2)x}-41}{0,3x+8}$$

väärtus, kui argumendi x kohta on vaid teada, et ta rahuldab tingimust $1 \leqslant x \leqslant 7$. Suhteliselt lihtsalt saab veenduda [1], et niisuguste x väärtuste korral paiknevad selle funktsiooni väärtused $-1,246$ ja $+5,644$ vahel. Seda arvestades tuleb meil nii argumenti kui ka arvutustulemust masinas kujutamiseks näiteks kümme korda vähen-

---

[1] Vaadeldav funktsioon on kasvav. Oma minimaalse väärtuse $(-1,246)$ omandab ta seega punktis $x=1$ ja maksimaalse väärtuse $(5,644)$ punktis $x=7$.

Tabel 8

Pesa	Pesas salvestatav käsk	Märkused
0500 0501	02 5420 21 0520	Esimene blokk
0502 0503	13 5416 21 0521	Kolmas blokk
0504 0505	02 5400 16 5422	Neljas blokk
0506 0507 0510 0511 0512 0513	02 5414 06 5420 07 5422 17 5414 05 5422 16 5410	Viies blokk
0514 0515 0516 0517	03 5422 04 5412 21 0522 22 0504	Kuues blokk
0520	37 5420	Teine blokk
0521	16 5410	Seitsmes blokk
0522	. . . . . .	Edasised arvutused
1410 1411	37 7777 77 7777	$x_0 = 1 - 2^{-35}$
1412 1413	00 0000 00 0002	$\varepsilon = 2^{-34}$
1414 1415	20 0000 00 0000	0,5
1416 1417	00 0000 00 0000	0
1420 1421	A	
1422 1423	Tööpesa	

dada. Tuues sisse mastaabikordaja 10 asendame x ja y vastavalt suurustega $\overline{x}$ ja $\overline{y}$, kus

$$x = 10\overline{x} \text{ ja } y = 10\overline{y}.$$

Arvutusvalem omandab nüüd kuju

$$10\bar{y}=\frac{\sqrt{(931+900\bar{x}^2)\cdot 10\bar{x}}-41}{3\bar{x}+8},$$

ehk

$$\bar{y}=\frac{\sqrt{(93{,}1+90\bar{x}^2)\bar{x}}-4{,}1}{3\bar{x}+8}=\frac{\sqrt{(0{,}931+0{,}9\bar{x}^2)\bar{x}}-0{,}41}{0{,}3\bar{x}+0{,}8}.$$

Kõik siin esinevad arvud on küll ühest väiksemad, kuid valemi rakendatavuseks tuleb veel kontrollida, kas ka kõik vahepealsed arvutustulemused ühest väiksemaks osutuvad. Juurealune avaldis $(0{,}931+0{,}9\bar{x}^2)\bar{x}$ on seda tõepoolest, sest $\bar{x}=0{,}7$ korral omandab ta oma suurima väärtuse 0,9604. Peab aga tähele panema, et sulgudes olev summa võib sellele vaatamata siiski ühest suuremaks osutuda (näiteks $x=0{,}7$ puhul omandab ta väärtuse 1,372). Seetõttu pole vaadeldav avaldis toodud kujul arvutamiseks sobiv ja programmeerimiseks tuleks sulud avada.

Halvem on lugu meie murru nimetajaga, mis $\bar{x}\geqslant\frac{2}{3}$ puhul tekitab ületäitumise. Selle vältimiseks jagame murru lugejat ja nimetajat näiteks kahega, millega meie arvutusvalem omandab lõplikult kuju (sulge juuremärgi all pole nüüd enam vaja avada):

$$\bar{y}=\frac{\sqrt{(0{,}23275+0{,}225\bar{x}^2)\bar{x}}-0{,}205}{0{,}15\bar{x}+0{,}4}.$$

Selle valemi kasutamisel peab silmas pidama, et argument tuleb masinasse viia kümme korda vähendatult ning et masinast saadavat tulemust tuleb tõelise vastuse leidmiseks korrutada kümnega.

Mastaabikordajate selline valimine pole kaugeltki mitte alati läbiviidav. Pealegi võib väga suurte mastaabikordajate valimine tekitada arvutustel märgatava täpsuse kao. Seepärast kasutatakse eriti «halva iseloomuga» ülesannete programmeerimisel mitmesuguseid võtteid mastaapide automaatseks muutmiseks arvutuste käigus. Kõige universaalsem, kuid kahjuks kõige enam aega nõudev nendest võtetest on liikuva koma kunstlik sissetoomine. Selleks kujutatakse iga arv nagu liikuva komaga masinaski oma mantissi ja järgu kaudu. Mantiss salvestatakse pikka pesasse tavalise fikseeritud komaga arvuna, järk aga mingisse lühikesesse pesasse 2^{17} korda vähendatud kujul. Näiteks arvu

$$-0{,}11011011010001101101001110100000000 \cdot 10^{10110} =$$
$$= -1101101101000110110100{,}11101$$

võib salvestada järgmiselt:

$$(d+4000) = 1{,}11011011010001101101001110100000000$$
$$(d+0002) = 0{,}000000000000010110.$$

Aritmeetiliste tehete sooritamine selliste arvudega toimub eriliste standardsete programmide abil. Olgu arvud $X = m_x \cdot 2^{p_x}$ ja $Y = m_y \cdot 2^{p_y}$ salvestatud pesadesse $m_x = (d+4000)$, $m_y = (d+4002)$, $p_x = (d+0004)$, $p_y = (d+0005)$.

Tehete tulemuse $Z = m_z \cdot 2^{p_z}$ jaoks reserveerime pesad

$$m_z = (d+4006),\ p_z = (d+0010).$$

Siis arvude X ja Y korrutamise teostab programm:

$(a+0000) = 02\ d+4000$ (X mantiss summaatorisse)
$(a+0001) = 06\ d+4002$ (mantisside korrutamine)
$(a+0002) = 15\ d+4006$ (korrutise normaliseerimine)
$(a+0003) = 01\ d+0004$ } (tegurite järkude liitmine)
$(a+0004) = 01\ d+0005$ }
$(a+0005) = 16\ d+0010$ (korrutise järgu salvestamine).

Samade arvude jagamine toimub järgmiselt:

$(a+0000) = 02\ d+4000$ (X mantiss summaatorisse)
$(a+0001) = 07\ d+4002$ (mantisside jagamine)
$(a+0002) = 15\ d+4006$ (jagatise normaliseerimine)
$(a+0003) = 01\ d+0004$ } (järkude lahutamine)
$(a+0004) = 03\ d+0005$ }
$(a+0005) = 16\ d+0010$ (jagatise järgu salvestamine).

Kõige komplitseeritumaks osutuvad liikuva komaga arvude liitmine ja lahutamine. Näiteks liitmise käik sõltub oluliselt liidetavate järkude vahe $\Delta p = p_x - p_y$ märgist ja vajab oma läbiviimiseks abiarve $(d+0011) = 40\,0001$ ning $(d+0012) = 20\,0000$. Programm ise omab kuju:

$(a+0000)=02 \quad d+0004$
$(a+0001)=03 \quad d+0005$ (Δp arvutamine)
$(a+0002)=21 \quad a+0014$ (kui $\Delta p<0$)
$(a+0003)=10 \quad d+0011$ (Δp varustamine miinusega)
$(a+0004)=17 \quad d+0012$
$(a+0005)=11 \quad 0000$ (m_y nihutamine $\Delta p+1$ koha võrra paremale)
$(a+0006)=06 \quad d+4002$
$(a+0007)=17 \quad d+0012$ (m_x nihutamine ühe koha
$(a+0010)=05 \quad d+4000$ võrra ja liitmine)
$(a+0011)=15 \quad d+4006$ (summa normaliseerimine)
$(a+0012)=01 \quad d+0004$ (summa järgu arvutamine)
$(a+0013)=22 \quad a+0023$ (arvutuste lõppu)
$(a+0014)=17 \quad d+0012$
$(a+0015)=11 \quad 0000$ (m_x nihutamine $1-\Delta p$ koha võrra paremale)
$(a+0016)=06 \quad d+4000$
$(a+0017)=17 \quad d+0012$ (m_y nihutamine ühe koha
$(a+0020)=05 \quad d+4002$ võrra ja liitmine)
$(a+0021)=15 \quad d+4006$ (summa normaliseerimine)
$(a+0022)=01 \quad d+0005$ (summa järgu arvutamine)
$(a+0023)=03 \quad d+0011$ (järgu parandamine)
$(a+0024)=16 \quad d+0010$ (summa järgu salvestamine).

Viimaste programmide üksikasjalisema analüüsimise jätame lugejale.

Arvutuste kontrollimine. Ülesande ettevalmistamine lahendamiseks elektron-arvutusmasinal ei piirdu aga ainult selle ülesande programmi koostamisega. Peab nimelt võtma veel tarvitusele abinõud, mis garanteeriksid lahendamise õigsuse, s. t. masinast ülesande lahendamise tulemusena väljatoodavate arvude usaldatavuse.

Kõigepealt tuleb veenduda, et ülesande lahendamiseks koostatud programm ei sisalda ü h t e g i v i g a. Programmi kontrollimine koosneb kahest osast. Esiteks peab selgitama programmi sisulise õigsuse, s. t. et selle programmi täitmise korral masin tõepoolest lahendab püstitatud ülesande.

Teiseks on tarvis kontrollida, kas programm on õigesti kantud perfolindile. Tuleb nimelt arvestada, et isegi üksainus valesti kirjutatud number mõne käsu aadressis või tehte koodis muudab kogu programmi reeglina kõlbmatuks. Näiteks kui suunamiskäsu aadress pole õige, siis võib juhtuda, et see käsk annab juhtimise üle niisugusesse pesasse, kus on salvestamisel tegelikult mitte käsk, vaid arv. Kuna arve ja käske salvestatakse aga pesades ühtemoodi, siis tõlgendab masin seda arvu ikkagi käsuna (s. t. jaotab arvu aadressiks ja tehte koodiks) ning asub teda täitma. Loomulikult ei saa sel juhul arvutuste õigsusest enam juttugi olla.

Programmi õiget kandmist perfolindile kontrollitakse näiteks sel teel, et perforeeritakse ta (erinevate isikute poolt) kaks korda ning võrreldakse tulemusi erilises perfolintide võrdlemise seadmes.

Keerulisemate programmide õigsuse kontrollimiseks kasutatakse muide ka arvutusmasinaid endid. Niisugune kontroll seisneb põhimõtteliselt selles, et masinal lastakse teostada arvutusi kontrollitava programmi üksikute etappide kaupa, kusjuures kasutatakse lihtsustatud lähteandmeid (millede puhul tulemused on kergesti käsitsi leitavad). Programmi sellist kontrollimist nimetatakse tema *silumiseks*.

Edasi tuleb veel kontrollida andmete ja programmi masinasse sisseviimise ning masinas teostatavate arvutuste õigsust. Võib ju juhuslik häire masina töös rikkuda kogu arvutuskäigu või vähemalt moonutada arvutustulemusi. Olgu muide märgitud, et ka kõige hoolikamalt ehitatud elektron-arvutusmasinas esineb üks juhuslik viga keskmiselt iga miljoni tehte kohta. See on muidugi suhteliselt väike vigade arv (vilunud arvutaja teeb käsitsi arvutamisel ühe vea keskmiselt iga saja tehte kohta), kuid arvestades elektron-arvutusmasinate töötamise suurt kiirust ei tohi vigade tekkimise võimalust seega siiski ignoreerida.

Peatumata masina õige töötamise kontrolli teistel võtetel nimetame vaid üht võimalust arvutuste õigsuses veendumiseks. See võimalus seisneb arvutuste kahekordses teostamises. Nimelt lastakse masinal kogu ülesanne või selle üksikud osad kaks korda läbi lahendada ja võrreldakse saadud tulemusi. Nende ühtimise korral loetakse, et ülesanne on lahendatud õigesti. Kahekordse arvutamise tulemuste ühtimine ei garanteeri muidugi täielikult arvutuste õigsust (näiteks kui masinas esineb mingi püsiv häire[1]). On aga väga tõenäoline, et *juhuslike* häi-

---

[1] Püsivate häirete väljaselgitamiseks lastakse masinal aeg-ajalt lahendada nn. *testülesandeid*, millede lahendid on ette teada.

rete poolt tekitatud vead ei anna ühesuguseid tulemusi.

Kahekordne arvutamine ja sellega kaasnev tulemuste võrdlemine võib toimuda näiteks järgmiselt:

Teostagu pesades $a+0000, \ldots, a+k$ paiknev programmi osa arvutusi, millede tulemus saadakse summaatorisse. Vajaliku kontrolli läbiviimiseks lisame programmi käsud:

$(a+k+0001)=16$ d+4000
$(a+k+0002)=14$ d+4002
$(a+k+0003)=21$ a+k+0005
$(a+k+0004)=22$ a+k+0014
$(a+k+0005)=02$ d+4000
$(a+k+0006)=16$ d+4002
$(a+k+0007)=20$ 0000
$(a+k+0010)=03$ d+4004
$(a+k+0011)=16$ d+4004
$(a+k+0012)=21$ a+0000
$(a+k+0013)=37$ 0000
$(a+k+0014)=$edasised arvutused.

Tööpesas $d+4002$ on algul mistahes arv, mille võrdlemine summaatoris oleva arvutustulemusega annab juhtimise esimesel korral üle käsule $(a+k+0005)$ (pole ju tõenäoline, et juhuslik arv oleks täpselt võrdne arvutustulemusega!). Nüüd viiakse arvutustulemus, mis vahepeal oli hoiul tööpesas $d+4000$, üle pesasse $d+4002$. Järgmised neli käsku on vajalikud selleks, et arvutused toimuksid just kaks korda. Nimelt on pesa $d+4004$ valitud nii, et seal oleks salvestamisel mingi positiivne arv, mida vaadeldavates arvutustes tarvis ei ole. Esimesel korral on tehte $(a+k+0010)$ tulemus siis negatiivne ja juhtimine läheb tagasi programmi algusse. Kui aga järgmisel võrdlemisel arvutustulemused ei ühti, siis annab tehe $(a+k+$ $+0010)$ positiivse tulemuse ja masin jääb seisma. Ühtlasi taastub pesas $d+4004$ olnud arv.

Programmeerimise lihtsustamise võimalusi. Elektronarvutusmasin teostab arvutusi üldiselt väga kiiresti, kuid andmete sisseviimine ja tulemuste väljatoomine on tunduvalt aeglasemad operatsioonid. Seetõttu ei ole ökonoomne kasutada niisugust masinat liiga lihtsate (s. t. suhteliselt vähe arvutusi nõudvate) ülesannete lahendamiseks[1]. Keeruliste ülesannete programmeerimine on aga väga tülikas ja aeganõudev töö (selles muide seisnebki tänapäeva elektron-arvutusmasinate peamine puudus). Näiteks kulub isegi suure vilumusega programmeerijatel juba keskmise raskusega ülesande programmeerimiseks mitu nädalat. Programmeerimise lihtsustamiseks kasutatakse mitmesuguseid võtteid.

Üheks loomulikumaks neist on nn. programmide kogu moodustamine, mis seisneb selles, et säilitatakse varemkoostatud programmid (enamasti tähelisel kujul, s. t. kujul, kui tähelised aadressid pole veel asendatud konkreetsete arvudega) ja püütakse uusi programme võimaluse korral koostada juba olemasolevate programmide või nende üksikute osade sobiva ühendamise teel. Näiteks kui polünoomi väärtuse arvutamise programm on kord koostatud ühe ülesande jaoks, siis saab seda ilmselt kasutada ka teiste selliste ülesannete korral, milledes tuleb muuhulgas arvutada polünoomide väärtusi (peab ainult ühtlustama vajalikud aadressid).

Kuid ka sellisest programmide kogust võetud osade ühendamine uueks programmiks osutub ikka veel küllaltki aeganõudvaks tööks. Seepärast on viimasel ajal intensiivselt töötatud mitmesuguste nn. automaatse programmeerimise meetodite loomise alal, mille eesmärgiks on kasutada elektron-arvutusmasinaid endid programmide koostamiseks.

Selles suunas on saadud mitmeid olulisi tulemusi (näiteks programmide kogust võetavate osaprogrammide enam-vähem automaatse ühendamise meetodite väljatöötamise näol), kuid programmeerimise täieliku automatiseerimiseni pole seni veel jõutud.

12. ELEKTRON-ARVUTUSMASINATE KASUTAMISEST

Küberneetika. Eelmises paragrahvis vaatlesime matemaatiliste ülesannete lahendamist elektron-arvutusmasinal. Selgus, et ülesande lahendamiseks tuleb kõigepealt koostada tema lahendamise programm. Programmi koostamine ise koosneb sisuliselt kahest etapist. Esimene neist

[1] Vähem kui miljon tehet nõudva ülesande lahendamine elektron-arvutusmasinal pole enamasti otstarbekas.

seisneb ülesande lahendamise *algoritmi*[1] leidmises, s. t. lahendamise taandamises lõpliku arvu niisuguste tehete järjestikusele sooritamisele, mis antud arvutusmasinas on realiseeritavad. Programmeerimise teisel etapil toimub ainult sellele algoritmile käskude kuju andmine ja käskude ratsionaalse paigutamise selgitamine.

Põhiliseks nendest kahest etapist on esimene — ülesande algoritmeerimine. Nimelt kui ülesande lahendamise algoritm on koostatud, siis vastava programmi koostamine on juba suhteliselt lihtne ülesanne (algoritmi koostab tavaliselt kõrge kvalifikatsiooniga matemaatik, kuid programmi enda väljakirjutamine jääb suhteliselt madalama kvalifikatsiooniga töötaja hooleks). See asjaolu määrabki tegelikult ära, milliseid ülesandeid saab elektron-arvutusmasinaga üldse lahendada: ülesande lahenduskäik peab olema algoritmeeritav.

Selleks et anda mingit ettekirjutust matemaatilistest ülesannetest, mille tegelik lahendamine on jõukohane ainult elektron-arvutusmasinale, toome järgnevalt mõne iseloomulikuma näite.

Maa kunstliku kaaslase või kosmoseraketi viimiseks soovitud orbiidile tuleb suurima täpsusega ette välja arvutada nii väljalaskmise moment, raketile antava kiirenduse muutumise seaduspärasused kui ka kõikide mehhanismide üheaegse töötamise üksikasjaline plaan. Kui neid arvutusi vähemalt põhimõtteliselt saaks teha ka käsitsi (kuigi see võtaks hulk aastaid), siis raketi juhtimine lendamisel, kõikide juhuslike mõjude arvestamine ja lennu korrigeerimine peavad toimuma juba n. ö. sama kiiresti, kui rakett lendab. Niisuguseid arvutusi enam käsitsi teostada pole mõeldav ja seega ainult elektron-arvutusmasinate kasutamine muutis tegelikult võimalikuks kosmoseraketi lennu juhtimise.

Meteoroloogilisi nähtusi kirjeldavad võrrandid osutuvad niivõrd keerulisteks, et nende isegi äärmiselt ligikaudne lahendamine kestab lihtsamate arvutusvahenditega varustatud arvutusbüroos mitu nädalat. Seepärast on arusaadav, et ilmade ennustamise seisukohalt ei oma niisugune lahendamine mingit mõtet. Alles elektron-arvutus-

[1] Algoritm tähendab täpset eeskirja teatud tehete järjestikuse teostamise kohta, millede sooritamisega saab lahendada mingisse kindlasse tüüpi kuuluvaid (matemaatilisi või loogilisi) ülesandeid.

masinate kasutuselevõtmine muutis võimalikuks täiesti täpse ilmade ennustamise. Ennustamise täpsuses võib veenduda igaüks, kes jälgib näiteks Moskva Ilmajaama prognoose [1].

Tööstuste ja transpordi tegevuse ratsionaalse planeerimise probleemi matemaatilise lahendamise võimalusi hakati selgitama tegelikult alles pärast seda, kui kiirelt töötavad arvutusmasinad olid oma haruldasi võimeid juba paljudel aladel veenvalt demonstreerinud. Vastavate uurimuste tulemusel selgus, et nimetatud probleem on sageli tõepoolest sõnastatav matemaatilise ülesandena, mille lahendamine osutub elektron-arvutusmasinale jõukohaseks. Seni on niisuguseid probleeme küll veel suhteliselt vähe lahendatud ja rakendatud, kuid tulemused osutusid enamasti lausa ootamatuteks. Kurioosumina võib näida näiteks selline tulemus, et tootmise ratsionaalse ümberkorraldamisega, ilma olemasolevat masinaparki täiendamata, saab mõne tehase toodangu tõsta 2—3-kordseks (on teada isegi juhtum, millal toodangu sai tõsta kümnekordseks). Kahjuks on aga niisuguste probleemide lahendamised (täpsemalt: praegu kasutatavad lahendusmeetodid) sedavõrd töömahukad, et nad sageli isegi elektron-arvutusmasinatele jõukohased ei ole.

Toodud näidetestki on juba näha, milliseid uusi võimalusi matemaatiliste ülesannete lahendamisel pakuvad kiirelt töötavad arvutusmasinad. Osutub aga, et arvutusmasinas realiseeritavatest tehetest koosnevatele algoritmidele saab taandada ka paljude n. ö. täiesti mittematemaatiliste ülesannete lahendamise. Niisuguste ülesannete väljaselgitamise ning uurimise käigus ongi tekkinud uus teadusala — k ü b e r n e e t i k a, mida võib defineerida kui teadust meid ümbritsevaid nähtusi kirjeldavate algoritmide loomisest ja uurimisest. Nähtuse algoritm kujutab endast tegelikult nende seaduspärasuste täpset kirjeldust, mis j u h i v a d seda nähtust [2]. Seda arvestades kõneldaksegi sageli: küberneetika on teadus meid ümbritsevate nähtuste juhtimisprotsessidest.

Siinjuures ei ole nähtavasti võimalik täpsemalt piirit-

---

[1] Kahjuks Eesti NSV jaoks koostatakse prognoosid esialgu veel ilma elektron-arvutusmasina abita. Sellest ongi tingitud nende prognooside suhteliselt halb täitumine tegelikkuses.

[2] Nimetus «küberneetika» pärineb kreekakeelsest sõnast «kübernetes», mis tähendab «tüürimees, juhtija».

leda nende «meid ümbritsevate nähtuste» klassi, millede algoritmide koostamine võib saada küberneetika uurimisobjektiks. Teaduse ja tehnika arenemisega see klass järjest laieneb. Igatahes koosneb nimetatud klass eeskätt sellistest nähtustest ja toimingutest, millede juhtimine või läbiviimine n. ö. tavalises olukorras nõuab inimese loogilise mõtlemise (intellekti) kaasabi. Näiteks on seni (peale rõhuva enamuse matemaatilistest ülesannetest) õnnestunud algoritmeerida paljude keeruliste tööpinkide ja isegi tervete tehaste (esialgu väiksemate) juhtimine, tänavaliikluse reguleerimine suurlinnades, lennuki juhtimine ka kõige keerulisemates tingimustes, tõlkimine ühest keelest teise, paljude loogiliste mängude mängimine jne. Niipea, kui aga on õnnestunud koostada mingi ülesande lahendamise algoritm, saab selle lahendamise enda juba täielikult automatiseerida, s. t. usaldada ta sobiva konstruktsiooniga arvutusmasina hooleks. Seega kui mingit intellektuaalset protsessi õnnestub algoritmeerida, siis ei vaja ta enam intellekti vahelesegamist ja on mehhaniseeritav.

Öeldust me näeme, et elektron-arvutusmasinate kasutuselevõtmisega on oluliselt laienenud ja muutunud tehnika arendamise ja kasutamise eesmärk. Kui varem oli tehnika kasutamise eesmärgiks inimese füüsilise töö asendamine masinate tööga, siis nüüd on lisandunud sellele võimalus anda ka üks osa inimese intellektuaalsest tegevusest üle masinatele. Järelikult mitmesuguste arvutusmasinate (vaadeldavas üldisemas rakenduses on neid õigem nimetada juhtivateks või küberneetilisteks seadmeteks) loomise ülesandeks on vabastada inimene nende intellektuaalsete funktsioonide täitmisest, mis on juba algoritmeeritud ning millede algoritmid on vajalikul määral läbi uuritud, võimaldades tal seega tegelda uute, printsipiaalsemate uurimustega.

Järgnevas vaatlemegi veidi lähemalt mõningaid mittematemaatilist laadi ülesandeid, millede algoritmeerimine ja automatiseerimine on seni õnnestunud.

Juhtivad masinad. Elektron-arvutusmasinate üheks majanduslikult oluliseks rakendusalaks on kujunenud mitmesuguste tootmisprotsesside automaatne juhtimine. Eriti suurt tähtsust omab see tootmisharudes, kus esinevad tervist kahjustavad tingimused või kus masinate töö juhtimine nõuab inimestelt suurt vaimset pingutust ja kiiret ning täpset reageerimist.

Laialdaselt on spetsiaalsed arvutusmasinad rakendamist leidnud näiteks mitmesuguste metalli töötlemise agregaatide (eeskätt frees- ja treipinkide) juhtimisel. Sealjuures kasutatakse üht arvutusmasinat enamasti kas terve rea samasuguste tööpinkide või komplekse automaatliini üheaegseks juhtimiseks.

Eriti efektiivseks on osutunud ka arvutusmasinate rakendamine keemiatööstuses, kus tootmisrežiimi äärmiselt täpsest jälgimisest enamasti täielikult sõltub toote kvaliteet.

Oluline on siinjuures eeskätt just see, et ainult kiire toimega arvutusmasin on võimeline õigeaegselt arvestama vähimaidki kõrvalekaldumisi ettenähtud režiimist ja võtma tarvitusele abinõusid nende kõrvalekaldumiste likvideerimiseks.

Oma põhiosas on tootmisprotsesse juhtivad arvutusmasinad sama ehitusega nagu need, mis on määratud matemaatiliste ülesannete lahendamiseks. Erinevused seisnevad tavaliselt vaid üksikute seadmete väljaehituse astmes ja mahus. Oluliselt muutunud on aga sisend- ja väljundseadmete otstarve ning ehitus.

Juhtiva masina väljundseadmete ülesandeks pole mitte arvutustulemuste trükkimine, vaid juhtiva seadme, näiteks tööpingi, töö vahetu reguleerimine. Seetõttu on tööpinki juhtiva arvutusmasina väljundseadmeks lihtsalt tööpingi juhtimispult. Iga arvutusmasinast väljundseadmesse tulev signaal lülitab nüüd kas sisse või välja tööpingi mingi osa, muudab töökiirust või materjali paigutust. Seega need signaalid sooritavad seda, mida tavalises olukorras peaks tegema tööpinki juhtiv inimene.

Sisendseadmetel on nüüd kaks ülesannet, sest oma töötamiseks vajaliku informatsiooni peab juhtiv masin saama kahest allikast: inimeselt ja juhitavalt seadmelt. Inimeselt saadav informatsioon koosneb tavaliselt masina töö programmist ja mõningatest lähteandmetest (tööpingi juhtimise korral on algandmeteks näiteks töödeldava detaili mõõtmed). Juhitavalt seadmelt tulev informatsioon on vajalik selleks, et arvutusmasin «teaks», kuidas näiteks tööpink reageerib talle väljundseadme kaudu lähetatud signaalidele (nn. tagasiside). Selle informatsiooni ja oma programmi põhjal määrab arvutusmasin juhitava seadme edasise töötamise režiimi. Siinjuures tuleb arvutusmasinal juhtimise käigus saadava informatsiooni alusel mõnikord muuta või täiendada ka iseenda töötamise programmi (sellise parandamise võimalus on programmis muidugi ette nähtud). Viimane asjaolu teeb võimalikuks ka niisuguste tootmisprotsesside juhtimise, mis enne protsessi algamist ei ole täielikult matemaatiliselt kirjeldatavad.

Seda laadi tootmisprotsesside hulka kuulub näiteks kõrgahju töö, mis väga suurel määral sõltub nii ahju kui ka teda teenindava brigaadi «individuaalsetest omadustest». Kõrgahju töö juhtimiseks viiakse elektron-arvutusmasinasse programm, mis kirjeldab kõrgahju tööd, kuid milles terve rida peamisi parameetreid on täpsustamata. Masin ühendatakse kõrgahju kõikide signaal- ja reguleerimisahelatega, kuid esialgu lastakse ahju juhtida siiski vilunud brigaadidel. «Jälgides» brigaadi tööd võrdleb masin seda sisseantud programmiga ja otsustab (programmis antud kriteeriumide alusel), milline töövõte annab paremaid tulemusi. Niimoodi täidab arvutusmasin kõik programmis esinevad lüngad ja valib välja kõige ökonoomsema ning kiirema töörežiimi, mis tagab kõige paremad lõpptulemused. Seejärel antakse kõrgahju töö juhtimine juba arvutusmasinale, millega on garanteeritud range kinnipidamine kõige ökonoomsemast režiimist.

Väga iseloomulikuks juhtivaks masinaks on tänavaliikluse juhtimise automaatseade, mille põhiosaks on samuti elektron-arvutusmasin. See seade võtab arvesse nii seisvate kui ka ristteele lähenevate autode hulga kõigis neljas suunas ja arvutab välja valgusfooride ümberlülitamise kõige optimaalsemad momendid (arvestusega, et transpordil tuleks ristteel võimalikult vähem seista). Niisugused seadmed leiavad juba mitmetes linnades kasutamist ja kõikjal on nad (samuti nagu teised juhtivad masinad) end täielikult õigustanud.

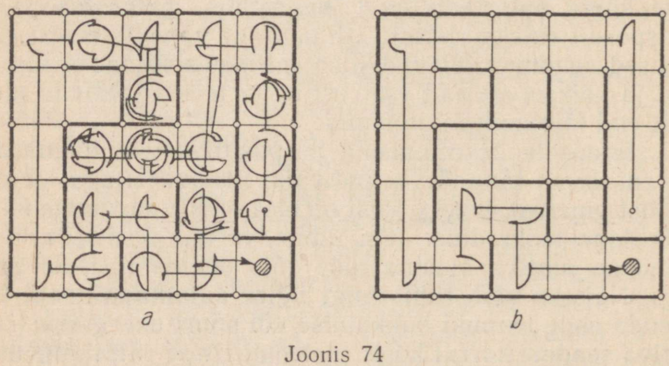
Laialdasele kasutamisele ja positiivsetele tulemustele vaatamata on tänapäeva juhtivatel masinatel aga ka kaks põhilist puudust. Üheks neist on elektriliste seadmete mitteküllaldane töökindlus. On ilmne, et iga ühekordne viga või tõrge juhtiva masina töös võib omada tõsiseid tagajärgi. Näiteks võib tühiseimgi häire automaatpiloodi töös kutsuda esile lennuki hukkumise või suurt energiasüsteemi juhtiva seadme korral kogu süsteemi rivist väljalangemise.

Seepärast tuleb juhtivate masinate töökindluse küsimustele pöörata eriti suurt tähelepanu.

Seni on töökindluse garanteerimiseks kasutatud peamiselt aparatuuri dubleerimist ja mitmesuguseid keerulisi reguleerimismeetodeid, mis muudavad juhtivad masinad aga tarbetult keerulisteks ja kalliteks. Rahuldavamalt võimaldab selle probleemi nähtavasti lahendada töökindlate ning välistele mõjudele vähetundlike pooljuhtdioodide, transistoride, ferromagnetiliste või ferroelektriliste seadmete laialdasem kasutuselevõtmine.

Juhtivate masinate teine oluline puudus seisneb selles, et tööprotsessi juhtimise käigus võib tekkida olukord, millele vastavat reageerimist pole programmis ette nähtud. Kuna kõiki niisuguseid erakordseid olukordi programmis sageli pole võimalik ette näha, siis kasutatakse tavaliselt võtet, et masin jätab protsessi sel korral seisma (kui see võimalik on) ja kutsub erilise signaaliga kohale operaatori. Loomulikult ei saa sellist lahendust aga pidada ammendavaks, pealegi võib esineda olukordi, mille erakordsust masin «taipab» liiga hilja. Seepärast töötataksegi praegu intensiivselt niisuguste algoritmide väljatöötamisel, mis võimaldaksid masinal ka ettenägemata olukordades õigesti (või vähemalt mitte väga valesti) reageerida. Need algoritmid peavad andma masinale seega võime teatud määral «loogiliselt arutleda».

Loogilise mõtlemise imiteerimine. Üheks esimeseks mudeliks, mis imiteeris teatavat loogilist protsessi, oli tuntud ameerika inseneri C. E. Shannoni poolt konstrueeritud «hiir labürindis» (vt. joonis 74). Väliselt koosneb see



Joonis 74

aparaat plaadist, mis on metallvaheseintega jaotatud 25 ruuduks. Osa vaheseinte kõrvaldamise teel saab plaati muuta labürindiks. Labürindi ühte nurka paigutatakse «sööt» (elektrikontakt või rauatükk) ja teise «hiir» (ratastel liikuv püsivmagnet). Plaadi alla paigutatud liikuva magneti abil juhib «hiire» liikumist suhteliselt lihtsa ehitusega arvutusmasin (mille mälu koosneb 110 elektromagnetilisest releest). «Hiir» on varustatud kontaktsarvekestega, mis vaheseinaga kokkupuutumise korral sulgevad vooluahela ja teatavad arvutusmasinale teelolevast takistusest.

«Hiire» ülesandeks on jõuda «söödani». Esimest korda labürinti paigutatud «hiir» saavutab eesmärgi alles pärast pikki eksirännakuid (joonis 74, *a*). Järgmiseks korraks peab aga arvutusmasin eelmise katse tulemused meeles ja juhib «hiire» kõige lühemat teed pidi kohale (joonis 74, *b*). Kui «hiir» asetada labürindi niisugusesse osasse, kus ta varem pole olnud, siis alustab ta uuesti juhuslikku otsingut, kuid olles jõudnud tuntud kohale, jätkab kõige lühemal maršruudil. Labürindi konstruktsiooni muutmise korral püüab «hiir» algul minna vana teed pidi, kuid hiljem «rahuneb» ja otsib taas välja kõige lühema tee.

Kirjeldatud mudel loodi selleks, et uurida võimalusi ehitada niisugust automaat-telefonikeskjaama, mis kaugekõnede andmisel otsiks ühtlasi välja kõige lühema ja vähemkoormatud liini. Niisugusel põhimõttel töötava keskjaama ehitamise võimalikkuse tõestabki tegelikult mudel «hiir labürindis».

Telefoniside korraldamise probleemid dikteerisid veel niisuguse mudeli loomise, milles imiteeritakse «poeskäimist». Selles mudelis on plaadile paigutatud rida lahtreid — «poode», mis on peale järjekorranumbrite varustatud veel eriliste tunnustega. Need tunnused iseloomustavad selles «poes müügil olevaid kaupasid». Mudeli liikuvale osale — «ostjale» antakse ülesandeks «osta» teatavat «kaupa». Selleks külastab ta järjekorras kõiki «poodisid», kuni leiab niisuguse, milles vajalikku kaupa «müüakse». Ühtlasi aga peab mudelit juhtiv arvutusmasin meeles ka kõik muud «kaubad», mis külastatud «poodides» leidusid. Järgmise ülesande saamisel kontrollib masin kõigepealt, kas on juba teada, millisest «poest» nõutud «kaupa» võib saada. Jaatava vastuse korral suunab ta «ostja» lühemat teed pidi õigesse «poodi». Vastasel juhul aga algab otsimise esimesest «poest», milles veel pole käidud.

See mudel loodi niisuguse telefonikeskjaama ehitamise võimaluste uurimiseks, mis arvestaks n. ö. iseenda kogemusi. Nimelt võtab enamik telefoniabonente ühendust peamiselt ainult teatud kindlate korrespondentidega, kuid kasutuselolevad automaat-keskjaamad seda ei arvesta (nõutud korrespondendi väljaotsimine teostatakse kõikide võimalikkude numbrite hulgast). Loomulikult võimaldab aga iga abonendi «lemmikkorrespondentide» tundmine ühenduste andmist märgatavalt kiirendada.

Analoogilisi mudeleid, mis imiteerivad mingit loogilist protsessi või «õpivad kogemustest», on konstrueeritud väga mitmesuguseid. Huvitava klassi moodustavad nende hulgas nn. mängivad masinad. Need on tegelikult arvutusmasinad, mis on varustatud programmiga, mille järgi masin saab mängida inimesega loogilisi mänge, nagu malet, kabet jms. (näiteks on Moskva Polütehnilises Muuseumis välja pandud masin, mis võib iga külastajaga mängida «trips-traps-trulli»).

Vastavalt oma töötamise programmile arvestab niisugune mängiv masin kogu aeg konkreetset olukorda ja valib võimalikkude edasimängu variantide hulgast välja selle, mis tagab talle parima olukorra teatud arvu käikude jooksul. Keerulisemate mängude, näiteks male puhul on vastava programmi koostamisel üheks põhiliseks probleemiks see, kuidas hinnata olukordi malelaual, s. t. milliste formaalsete tunnuste järgi tuleb üht seisu teisest paremaks lugeda. Teiselt poolt aga tuleb välja töötada niisugune mängu strateegia, mis iga vastase poolt valitava strateegia korral võimaldaks saavutada parimaid tulemusi. Male puhul neid küsimusi täielikult lahendada pole õnnestunud.

Niisuguste mängivate masinate konstrueerimisel on ilmselt suur põhimõtteline tähtsus nii elektron-arvutusmasina «võimete» kui ka loogiliste protsesside üldiste seaduspärasuste väljaselgitamise osas.

Automaatne tõlkimine. Üheks inimese intellektuaalse tegevuse alaks, mida elektron-arvutusmasinate kasutuselevõtmine võimaldas mehhaniseerida, osutub tõlkimine ühest keelest teise. Esimene edukas katse kasutada elektronarvutusmasinat tehnilise teksti tõlkimiseks (vene keelest inglise keelde) toimus 7. jaanuaril 1954. a. New Yorgis.

Kuigi esimesed sellelaadilised katsed omasid peamiselt põhimõttelist tähtsust, tõestades automaatse tõlkimise võimalikkust üldse, ei tule saadud tulemuste praktilist väärtust siiski alahinnata (vaatamata sellele, et näiteks ilukirjanduse automaatsest tõlkimisest on veel vara rääkida). Nimelt ilmub igal aastal miljoneid teaduslikke artikleid kõige erinevamates keeltes, milledega tutvumine on teiste maade teadlastele hädavajalik. Selliste artiklite tõlkimise mehhaniseerimine omab järelikult suurt praktilist väärtust. Ühtlasi tuleb muidugi arvestada seda, et teadusliku teksti piiratud sõnavara ja lakooniline vorm muudavad tema tõlkimise suhteliselt kõige lihtsamini automatiseeritavaks.

Nendel põhjustel ongi teadusliku teksti automaatse tõlkimise küsimusi viimastel aastatel paljudes maades laialdaselt uuritud, mille tulemusena vastavad programmid on välja töötatud terve rea keeltepaaride jaoks. Selles suunas saadud tulemusi kasutades on asutud nn. informatsioonimasinate konstrueerimisele. Nende masinate väga suure mahtuvusega ja kiirelt töötavas (enamasti püsiv-) mälus saab salvestada kogu ilmuva teadusliku kirjanduse teatud erialadelt. Sealjuures võib kirjandus esialgselt olla erinevates keeltes. Informatsiooni vastuvõtmisel tõlgib masin selle erilisse «masina keelde», mis on spetsiaalselt selleks otstarbeks loodud ja kujutab endast maksimaalse informatsioonisisaldusega sümboolikat. Mälus salvestatud ja eriala tunnustega varustatud kirjanduse põhjal koostab masin soovitud põhjalikkusega ülevaateid (annab informatsiooni) ühel või teisel kitsamal alal teostatud uurimustest. Informatsiooni väljaandmisel tuleb jälle teostada tõlkimine «masina keelest» soovitud keelde. Seega automaatne tõlkimine (mitmesuguste keelte vahel) moodustab ka informatsioonimasina töös küllaltki olulise osa.

Selleks et elektron-arvutusmasin saaks teostada tõlkimist ühest keelest teise, tuleb tema mälusse viia nii lähtekeelne kui ka tulemuskeelne sõnastik ja tõlkimise programm (tavaliselt moodustavad tulemuskeelne sõnastik ja tõlkimise programm tegelikult ühtse terviku). Lähtekeelse sõnastiku maht ja koosseis oleneb väga suurel määral sellest, millise sisuga tekste kavatsetakse tõlkida. Kuigi enamiku keelte kogu sõnavara ületab 50 000 sõna, piisab näiteks matemaatilise või mõne teise kitsamalt erialase teksti tõlkimiseks vähem kui 2000 sõnast.

Lähtekeelne sõnastik on (enamiku keelte puhul) otstarbekas esitada muutumatute tüvede kujul, kusjuures eraldi osana tuleb lisada veel afiksite (muutelõppude) sõnastik. Peale selle peab sõnastik tavaliselt sisaldama ka enamkasutatavaid idioome.

Sõnastiku mälluviimiseks tuleb tähed muidugi kodeerida kahendarvudega, millega tähestiku järjekord asendub vastavate koodide suurusjärjekorraga. Sealjuures on automaatse tõlkimise puhul otstarbekas kasutada mitte tavalist tähestiku järjekorda, vaid muuta seda vastavalt vaadeldava eriala iseloomule. Nimelt tähed, milledega tõlkimisele kuuluvas kirjanduses sõnad sagedamini algavad, tuleb tähestikus paigutada ettepoole. Sellisel juhul on tõenäoline, et enamik vajalikke sõnu leitakse juba sõnastiku algusest. Näiteks venekeelse matemaatilise kirjanduse puhul osutub sõna algustähtede sageduse (ja seega automaatseks tõlkimiseks sobiva tähestiku) järjekord niisuguseks: п, с, о, в, н, . . .

Tõlgitav tekst antakse elektron-arvutusmasinasse tavaliselt üksikute lausete kaupa ja masin tõlgib iga lause eraldi. Seega kui lause on väljaspool konteksti tõlkimatu (seda juhtub peamiselt ilukirjanduslikku laadi tekstides), siis ei ole masin teda suuteline õigesti tõlkima.

Automaatne tõlkimine algab sellega, et sõnastikust otsitakse vaadeldava lause iga sõna jaoks tüvi ja muutelõpp. Mingi ühe sõna väljaotsimiseks sõnastikust võrreldakse teda sõnastikus olevate tüvedega (nende paiknemise järjekorras) seni, kuni leitakse kõige pikem tüvi, mis vaadeldavas sõnas esineb. Kui koodid on sõnastikku paigutatud kasvamise järjekorras, siis võib võrdlemine toimuda näiteks sel teel, et vaadeldava sõna koodist lahutatakse tüvede koode niikaua, kui vahe muutub negatiivseks. Tüvi, mis andis viimase mittenegatiivse vahe, ongi siis otsitav, kuna aga vahe ise kujutab sõna muutelõpu koodi. Seda võrreldakse nüüd afiksite sõnastikuga samasugusel viisil.

Iga tüvi ja muutelõpp on sõnastikus varustatud teda tõlkiva alamprogrammi järjekorranumbriga, s. t. selle alamprogrammi esimest käsku sisaldava pesa aadressiga. Kui vaadeldav sõna saab omada ainult ühe tähenduse, siis sisaldab niisugune alamprogramm vaid tulemuskeelse vaste ja selle mõningad tunnused (näiteks sõnaliik, sugu, rektsioon, muutumise tüübi number). Mitmetähenduslike sõnade (homonüümide) korral peab vastav alamprogramm määrama tehted, millede abil (teisi sõnu ja nende paiknemist lauses arvestades) saab sobiva vaste (ja selle tunnused) välja valida.

Seega automaatse tõlkimise esimesel etapil leitakse tõlgitava lause iga sõna (täpsemalt: selle tüve ja muutelõpu) jaoks vastava tõlkimiseeskirja (programmi) järjekorranumber (aadress).

Tõlkimise teine etapp seisneb lähtekeelse lause grammatilises analüüsis ja grammatiliselt õige tõlke moodustamises (esimesel etapil saadud andmete põhjal). Töö käik sõltub nüüd oluliselt sellest, milliseid grammatilisi meetodeid tõlkimisel kasutatakse. Nimelt on konkreetsete keelte normatiivsed grammatikad oma rõhuvas osas koostatud nii, et neid automaatse tõlkimise puhul vahetult rakendada ei saa. Põhjuseks on asjaolu, et grammatika reegleis kasutatakse väga sageli rangelt defineerimata mõisteid, mida (või millede mõtet) ainult selgitatakse; need reeglid on enamasti kirjeldavat laadi. Automaatse tõlkimise jaoks tulevad grammatika reeglid aga esitada rangelt formaliseeritud kujul: mõtteliste või sisuliste seoste kasutamine on kas täiesti lubamatu või vähemalt äärmiselt ebasoovitav.

Kogu grammatilist laadi informatsiooni hangib arvutusmasin automaatsel tõlkimisel vaid sõnade kirjapildist ja nende paiknemisest lauses. Selle informatsiooni hankimise käik (tõlkimise programm) tuleb esitada täpselt fikseeritud reeglite kujul, kusjuures arvutusmasinate kahendehitust arvestades peab iga reegel seisnema valikus kahe alternatiivse võimaluse vahel.

Täielikku formaalset grammatikat pole seni veel ühegi keele jaoks loodud, vaid on läbi aetud olemasolevate (normatiivsete) grammatikate osalise formaliseerimisega. See põhjustabki tõlkimismeetodite suure mitmekesisuse, mistõttu lähema ülevaate andmine pole käesolevas raamatus võimalik [1]. Märgime vaid, et käesoleval ajal teostatakse intensiivseid uurimusi mitte üksnes kirjalike tekstide, vaid ka kõne tõlkimiseks, kusjuures masin peab tõlke välja andma nii trükitult kui ka kuuldava kõnena.

Füsioloogilised mudelid ja robotid. Laialdast rakendamist hakkavad küberneetika meetodid viimastel aastatel

[1] Kasutatavate tõlkimismeetodite kohta vt. näit. Л. Ю. Панов. Автоматический перевод. Москва, 1958.

leidma ka füsioloogias ja psühholoogias, võimaldades selgitada inimorganismi mõningate alade mehhanismi ja üldisi seaduspärasusi. Näiteks on elektron-arvutusmasinate abil püütud modelleerida üksikuid närvihäireid (nende tõenäoliste põhjuste väljaselgitamiseks). On loodud masinaid isegi näiteks südame- ja närvihaiguste diagnoosimiseks. Samasse suunda kuuluvad ka uurimused inimese hävinud või nõrgenenud meeleorganite asendamiseks kunstlike vahenditega, kuigi eriti silmapaistvaid tulemusi sel alal seni veel saavutatud ei ole.

Suuremaid tulemusi on saadud inimese mõningate normaalsete psühholoogiliste funktsioonide modelleerimise alal. Nii on elektron-arvutusmasinate abil õnnestunud imiteerida tingrefleksi kujunemist. Selleks paigutatakse miniatuurne arvutusmasin näiteks elektrimootori abil liikuvasse mudelisse, mis on varustatud tundliku puhvriga. Kui mudel satub liikumisel vastu tõket, siis muudab arvutusmasin tema liikumise suunda nii, et mudel tõkkest mööduks (samuti nagu mudeli «hiir labürindis» puhul). Mudel varustatakse veel mikrofoniga ja iga kord enne tõkke vastu põrkamist antakse tugev helisignaal. Veidi aja pärast hakkab mudel juba ainuüksi helisignaali mõjul «tõkkest mööduma», kusjuures tõket ennast ei tarvitse ees ollagi.

Analoogilisel viisil on modelleeritud ka teisi lihtsamaid füsioloogilist laadi protsesse, nagu toidu otsimine, põgeneja püüdmine, tagaajajate eest põgenemine jne. Siia võib lugeda ka ülalkirjeldatud «poeskäimise» imiteerimise.

Suurt huvi on alati äratanud inimesele nii vormi kui ka funktsioneerimise poolest sarnaste masinate — robotite ehitamine. Sellel teemal on kirjutatud palju fantastilisi jutustusi ja romaane, kuid alles küberneetika annab mõningaid (esialgu küll kasinaid) võimalusi nende fantaasiate muutmiseks tegelikkuseks.

Inimest imiteeriva mudeli ehitamisel on põhiküsimuseks teda juhtiva programmi koostamine. Loomulikult ei saa see programm olla terviklikult etteantav, vaid funktsioneerimise käigus peab mudelit juhtiv arvutusmasin oma töö programmi enamasti ise koostama (sõltuvalt tingimustest, millesse mudel satub). Seega esialgu tuleb masinasse sisse anda tegelikult programmi koostamise programm. Selliste programmeerivate programmide loomine on seni õnnestunud vaid lihtsamatel juhtudel.

Üheks suuremaks raskuseks vaadeldavate mudelite loomisel on ka keskse juhtimisseadme konstrueerimine. Kui soovitakse mudelis imiteerida suurt osa inimese funktsioonidest, siis omandab juhtimisseade niivõrd komplitseeritud kuju, et seda koos suuremahulise mäluga pole võimalik mudelisse äragi paigutada, rääkimata teistest konstrueerimisel kerkivatest raskustest.

Lihtsamaid funktsioone täitvaid roboteid saab siiski ka praegu juba ehitada ning on tehtudki katseid roboti kasutamiseks näiteks «katselendurina».

Eeskätt robotitega seoses ongi lääneriikides viimase kümne aasta vältel küberneetika ümber palju lärmi tehtud (sealjuures peaaegu eranditult just mitteasjatundjate poolt). Küberneetikast räägitakse kui «teaduste teadusest», mis suudab kõik ja mille arengul pole mingeid piire. Väidetakse, nagu oleks küberneetika eesmärgiks inimeste asendamine masinatega, nagu hakkaksid Maakeral tulevikus valitsema robotid ning inimesed saavad eksisteerida vaid seni, kuni sellised «kunstlikud inimesed» nende olemasolu sallivad.

Kuidas niisugusesse kõmusse suhtuda? On ju küberneetilise tehnika edasiarendamise eesmärgiks luua seadmeid, mis suure usaldatavusega suudaksid realiseerida üha keerulisemate ülesannete lahendamist. Ülesannete keerulisuse mõõdupuuks kasutatakse tavaliselt seda ülesannet lahendada või algoritmeerida suutva inimese kvalifitseeritust (kõneldakse ka «algoritmi kvalifitseeritusest»). Kas ei ole seega tulevikus võimalik luua algoritme, mis oleksid «kvalifitseeritumad» igast inimesest, ehitada masinaid, mis oleksid inimesest «targemad»?

Nendele küsimustele vastamiseks tuleb selgitada, millega piirdub sarnasus arvutusmasina ja aju töötamises ja milles peituvad nende peamised erinevused.

Arvutusmasina ü k s i k u t e elementide ja ajurakkude (neuroonide) töötamises on nähtavasti suur sarnasus, kuid arvutusmasin ja aju tervikuna on kahtlemata hoopis erineva ehitusega. See erinevus on nii kvantitatiivset kui ka kvalitatiivset laadi ning pealegi toimub nende töötamine sisuliselt erinevatel põhimõtetel. Kui masin lahendab loogilist või matemaatilist ülesannet, siis on tema töötamine vaid formaalselt (oma tulemuste poolest) analoogiline inimese aju vastava tegevusega. Malet mängivat masinat näiteks võiks võrrelda vaid inimesega, kes enne järjekordse käigu tegemist vaatab läbi kogu olemasoleva malealase kirjanduse. Siin võidaks muidugi väita, et tegemist on vaid programmi ebaratsionaalse koostamisega masina jaoks. Asi pole aga mitte ainult programmi ratsionaalsuses, vaid eeskätt arvutusmasina töötamispõhimõttes. Loogiliste ülesannete masinas lahendamise meetodiks on ju valiku teostamine kahe alternatiivse võimaluse vahel. Inimene aga malet mängides või loogilist ülesannet lahendades reeglina seda meetodit ei kasuta.

Lõpuks tuleb inimese ja masina võrdlemisel arvestada seda, et masin töötab vaid etteantud instruktsiooni — programmi järgi. Paljud teadlased töötavad praegu selle kallal, et lihtsustada programmi koostamist. Nad püüavad saavutada olukorda, et masin täidaks oma funktsioone võimalikult minimaalse esialgse informatsiooni järgi. Loota aga, et masinad hakkaksid töötama ilma neile lahendamisele kuuluvate ülesannete kohta mingit informatsiooni andmata (s. t. et masinad kogu vajaliku informatsiooni ise muretseksid), ei näi olevat mingit põhimõttelist võimalust.

Seega suudab masin lahendada vaid neid ülesandeid, mida inimene tal lahendada käsib, s. t. ülesandeid, millede algoritmi (või algoritmi koostamise algoritmi) inimene on juba tuletanud. See muidugi ei tähenda, nagu ei saaks arvutusmasin lahendada ülesandeid, mis inimesele endale pole jõukohased. Koguni vastupidi. Elektron-arvutusmasinatega lahendatakse peaaegu eranditult ainult niisuguseid ülesandeid, mis (peamiselt oma suure töömahu tõttu) inimesele ei ole jõukohased. Selles pole aga põhjust imestada, sest enamikul masinatest on ju sama omadus (näiteks auto sõidab kiiremini kui inimene käia suudab jne.). Vaadeldavas mõttes on masinad järelikult inimesest «targemad». Ei tohi aga unustada, et niisuguseid ülesandeid lahendab masin siiski ainult juhul, kui inimene talle vastava käsu annab.

Muidugi võib masin mõne ülesande lahendamisel teha rohkem, kui inimene tal v a h e t u l t teha käsib, kuid lõppude lõpuks täidab ta siiski vaid inimese käsku. Väita, et masin (kas või tulevikus) suudab tegutseda nagu iseseisva mõistusega olend, pole seega millegagi põhjendatud.

Järelikult: masin võib teha kõike seda, mida õnnestub algoritmeerida. Millega aga piirdub algoritmeeritavate ülesannete hulk, sellele pole printsipiaalselt võimalik vastata (kuigi saab näidata konkreetseid ülesandeid, mida pole võimalik algoritmeerida). Ei saa ette ära näidata, kui kaugele võib arendada küberneetilisi seadmeid, samuti nagu ei saa ette näha inimmõistuse arenemise piire. Samal põhjusel pole ka mingit alust karta, et tulevikus teevad masinad ära kogu inimese intellektuaalse töö ja inimmõistus mandub.

MÕNINGAD KODUMAISED JA VÄLISMAISED ELEKTRON-ARVUTUSMASINAD

Elektron-arvutusmasinate laialdast seeriatootmist on seni suudetud organiseerida vaid neljas riigis: NSV Liidus, Ameerika Ühendriikides, Saksa Föderatiivses Vabariigis ja Inglismaal. Masinaid on valmistatud küll ka teistes riikides (Prantsusmaal, Tšehhoslovakkias, Saksa Demokraatlikus Vabariigis, Šveitsis, Kanadas, Jaapanis ja mujal), kuid enamasti ainult väikeste seeriate või üksikeksemplaridena mõne firma või teadusliku asutuse poolt oma tarbeks, kusjuures kasutatakse sageli välismaisi eeskujusid.

Nõukogude Liidus toodetavad elektron-arvutusmasinad on välja töötatud peamiselt Moskva uurimisinstituutide poolt ning ei kujuta endast ühegi välismaise eeskuju koopiat. Nad on valmistatud sajaprotsendiliselt kodumaa tööstuse poolt, kodumaistest detailidest ja materjalidest.

Seeriaviisiliselt toodetakse meil universaalseid elektron-arvutusmasinaid «Ural», M-3 (väikeste masinate klassi kuuluvad) ja «Strela» (suurte masinate klassi kuuluv) ning mitmesuguseid enam või vähem spetsialiseeritud elektron-arvutusmasinaid, nagu näiteks «Pogoda». Peale nimetatud kolme tüübi on vähemal arvul valmistatud veel universaalseid elektron-arvutusmasinaid M-2, БЭСМ, «Kiev».

Nõukogude matemaatiliste masinate tööstus arendab välja järjest uusi täiuslikumaid arvutusmasinate tüüpe. Arengusuunaks on peamiselt ferromagnetiliste ja ferroelektriliste elementide kasutamine mäluseadmeis, pooljuht- ning magnetiliste lülitavate elementide kasutamine loogiliste skeemide koostamisel ning trükitud skeemide tehnoloogia rakendamine tööstuses. Nende abinõude tulemusena saab võimalikuks kiirete, väikesegabariidiliste, töökindlate ja ökonoomsete arvutusmasinate laialdane kasutuselevõtmine kõigil teadusaladel ning rahvamajanduses.

1. ELEKTRON-ARVUTUSMASIN «URAL»

See väikeste masinate klassi kuuluv elektron-arvutusmasin on sobiv kasutamiseks konstruktsioonibüroodes, projekteerimisinstituutides ning kõrgemates õppeasutustes. Masina põhiosaks on aritmeetiline seade ja temaga vahetult seotud operatiivmälu (magnettrummel). Välised mäluseadmed (magnetofon ja fototransmitter) on operatiivmäluga ühendatud aritmeetilise seadme kaudu.

Aritmeetiline seade opereerib 35-kohaliste kahendarvudega, milles koma on fikseeritud kõige vasakpoolsema koha ette. Kolmekümne kuues kahendkoht kujutab arvu märki (vt. § 4). Informatsioon antakse mälust aritmeetilisse seadmesse 9-kohaliste kahendarvude kaupa. Sisendnihutaja (9-kohaline register) kannab need numbrirühmad üle 42-kohalisse registrisse, kus viimased kuus kohta täituvad vaid korrutamisel ja tagavad korrutise suurema täpsuse.

Register on ühendatud summaatoriga klappide abil, mis vastavalt vajadusele annavad registrist summaatorisse üle kas arvu otse- või pöördkoodi. Summaatoriga on ühendatud väljundregister, mis on samuti nagu sisend-nihutajagi 9-kohaline. Väljundregistri kaudu salvestatakse informatsioon magnettrumlile. Aritmeetilise seadmega on ühendatud ka kohalik juhtimisseade.

Aritmeetilises seadmes toimuvad operatsioonid on jäigalt sünkroniseeritud magnettrumlilt tulevate sünkroniseerimisimpulsside abil. Kõik tehted toimuvad ühe takti (10 msec) kestel, välja arvatud normaliseerimine (2 takti) ja jagamine (4 takti).

«Urali» käskudesüsteemiga tutvusime juba varem (§ 10). Meenutame vaid, et tegemist on üheaadressilise masinaga.

Mäluseadmed. Operatiivmäluks on magnettrummel, kuhu informatsioon salvestatakse 1024 pesasse. Informatsioon on paigutatud trumlile tihedusega 3 impulssi millimeetril. Tööolukorras möödub helipea alt 280 000 impulssi sekundis, helipeas indutseeritav signaal on 0,7—0,8 V ning õhupilu trumli ja helipea vahel umbes 25 mikroni. Magnetofoniga tuttav lugeja taipab, milliseid tehnilisi raskusi on meie tööstus siin edukalt ületanud.

Väliseks mäluks on magnetofon. 35-millimeetrisele magnetofonilindile salvestatud informatsioon jaotatakse erimärkidega tsoonideks. Iga tsooni alguse ette on perforeeritud tsooni number, mida fikseerib fotoelement. Ühel lindil on paralleelselt 2 tsooni. Tsooni pikkus on meelevaldne, kuid mitte üle 1000 arvu (seega võib üks tsoon täita praktiliselt terve operatiivmälu). Informatsioonivahetus välise ja operatiivmälu vahel toimub automaatselt vastava käsu esinemisel programmis. Käsus on näidatud ära informatsiooni ülekande suund, tsooni number lindil ja operatiivmälu vastavad aadressid. Üldse on lindil kuni 255 tsooni. Informatsiooni tihedus lindil on 0,8 impulssi millimeetril ja lindi liikumise kiirus 0,8 m/sec.

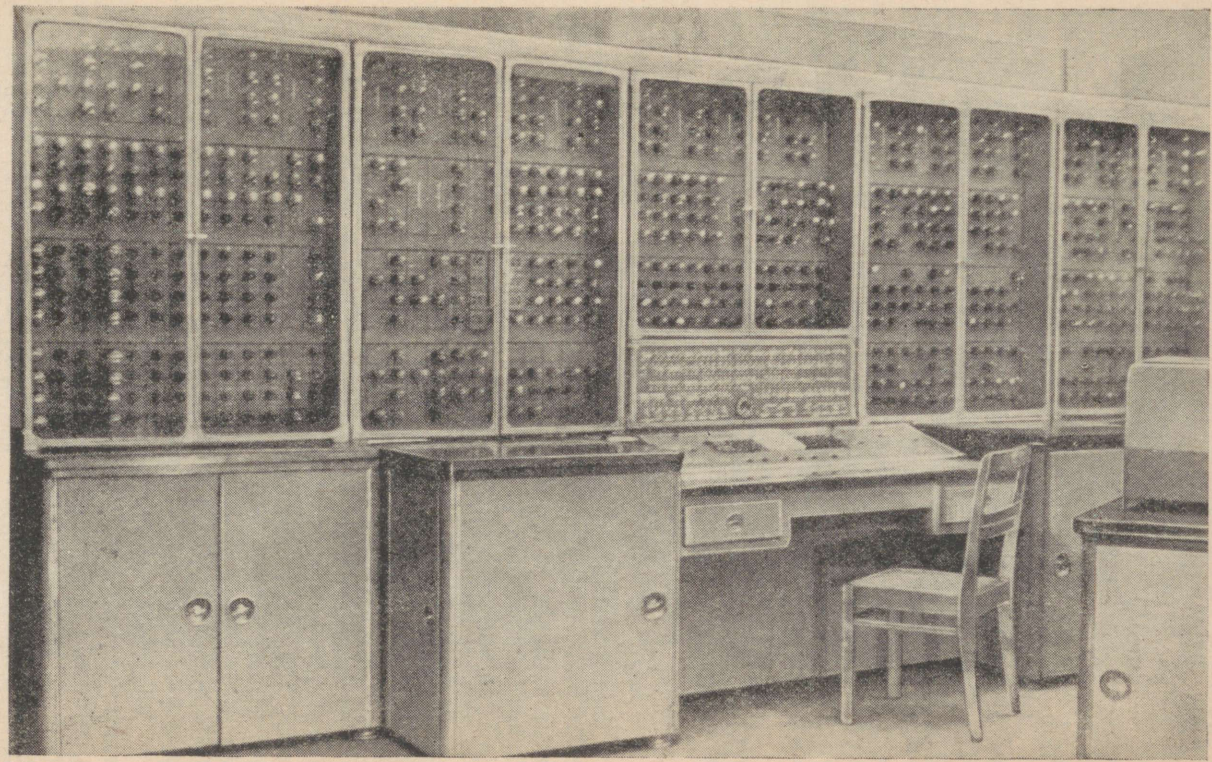
Paralleelselt magnetofoniga kasutatakse veel perfolintmäluseadet (sisendseade). Siin on informatsioon perforeeritud tavalisele kinofilmile (läbipaistmatule). Tsoonide arv on 127 ning tööpõhimõte põhiliselt sama. Informatsioonivahetus välise mälu ja operatiivmälu vahel kulgeb läbi aritmeetilise seadme summaatori, mistõttu arvutused tuleb katkestada. Arvutusmasinat sünkroniseerivad sel perioodil impulsid, mida antakse lindiveomehhanismide veovõllilt.

Sisend- ja väljundseadmed. Arvude sisseviimine masinasse toimub ainult perfolindilt. Üksikuid arvusid saab masinasse viia ka juhtimispuldilt. Sisendseadmete kiirus (lindilt) on 4500 arvu minutis. Vigade leidmiseks perfolindilt kasutatakse võrdlejat, mis saab töötada kahes režiimis (kahe perforeeritud lindi võrdlemine ja perfolindi võrdlemine operaatori poolt klaviatuuril valitud arvudega).

Väljundseadmeks on trükkija, mis võib trükkida arve (või käske) magnettrumlilt ainult summaatori kaudu. Kuna ühe arvu trükkimine vältab 60 takti, siis selleks, et võimaldada arvutuste jätkamist ka trükkimise ajal, omab trükkija eraldi registri (külma katoodiga tiratronidel). Niipea kui arv on üle antud trükkimisregistrisse, jätkuvad arvutused. Kui uus trükkimiskäsk esineb enne 60 takti möödumist, siis peab terve arvutusmasin ootama trükkimisregistri vabanemist.

Teiseks väljundseadmeks on väljundperforaator, mis perforeerib 150 arvu minutis (trükkija 100 arvu minutis). Arvutustulemuste kõrval trükitakse sageli kontrolliks ära ka täidetud programm.

Juhtimisseade ja pult. Juhtimisseade dešifreerib käsud ja juhib vastavate sõlmede tööd automaatselt. Puldile on välja toodud lülitid masina käsitsi juhtimiseks ning kla-



Joonis 75.

viatuur üksikute arvude viimiseks masinasse. Puldil asuvatelt neoonlampidelt võib lugeda (kahendsüsteemis), millised arvud asuvad summaatoris või aritmeetilise seadme, juhtimisseadme ja trükkija registrites. Puldil on veel erilised signaallambid, mis süttivad summaatori ületäitumise ja suunamiskäskude puhul. Ühtlasi asub puldil kell. Masina üldvaade on toodud joonisel 75.

Elektron-arvutusmasin «Ural» vajab ca 60—70 m² põrandapinda. Järelevalve ja remondi hõlbustamiseks koosneb masin pistikute abil lahtivõetavatest blokkidest (mis igaüks sisaldavad ühe lambi koos juurdekuuluvate detailidega). Arvutusmasinat toidetakse elektromehaaniliselt stabiliseeritud 3-faasilise võrguvooluga (220 V). Vajalikud alalispinged saadakse võimsatelt germaanium-pooljuhtalaldajatelt. Elektronlampide pikema tööea tagamiseks on anoodpinge vaid 140 V.

Vaadeldava arvutusmasina uues mudelis on peale muude täiustuste tunduvalt tõstetud töötamiskiirust (kuni 6000 tehet sekundis).

2. ELEKTRON-ARVUTUSMASIN M-3

Seeriatootmisele on antud ka (mitmete organisatsioonide poolt endale varem üksikeksemplaridena ehitatud) arvutusmasin M-3. Kuigi ta on tunduvalt väiksem ning 2 korda aeglasem kui «Ural», osutuvad tema tööjõudlus ja kasutamisvõimalused umbes samadeks.

M-3 on ehitatud standardsetest mittedefitsiitsetest raadiodetailidest; ta sisaldab umbes 17 500 raadiodetaili, sealhulgas 4000 pooljuhtseadist ja ligi 800 raadiolampi. Jooteühenduste arv on ligikaudu 200 000—300 000.

Aritmeetiline seade koosneb kolmest põhiregistrist ja ühest abiregistrist 30-kohaliste kahendarvude salvestamiseks (31. koht sisaldab märgi). Abiregister on vajalik niisuguste tehete puhul, mis sisaldavad liitmist (sisuliselt on sellisteks kõik neli aritmeetilist tehet, sest lahutamine toimub kui täiendkoodi liitmine). Abiregistris moodustatakse ja salvestatakse liitmisel tekkivat ülekannet.

Registrites teostatavaid tehteid juhib kohalik programmjuhtimisseade, andes iga tehte puhul välja rea pingenivoosid ning üksikimpulsse, mis juhivad registreis toimuvaid protsesse — nihutamist (vasakule ja paremale), osakorrutiste liitmist jne. Kohalik programmjuhtimisseade on ühendatud tsentraalse impulsside jaotusseadmega. Jaotusseade kutsub mälust välja käsu, dešifreerib selle, määrab kindlaks teostatava tehte, kutsub mälust välja vajalikud arvud, paigutab need registritesse ja annab juhtimise seejärel tehteimpulsiga üle kohalikule programmjuhtimisseadmele. Kui kohalik programmjuhtimisseade on saatnud tehte lõppu tähistava impulsi, siis asub jaotusseade järgmise käsu otsimisele.

Käsud on arvutusmasinas M-3 kaheaadressilised, kusjuures tehte tulemus käsu kujust sõltuvalt kas jääb aritmeetilise seadme ühte registrisse, trükitakse või saadetakse mälusse tagasi (käsu teise aadressi järgi). Käskude süsteem sisaldab 48 käsku, nende hulgas ka neli tehet arvude absoluutväärtustega. Koma on fikseeritud vasakule arvu ette; kui arvutuste käigus tekib ühest suurem arv, siis arvutusmasin peatub.

Mälu. Operatiivmäluks on 2048 pesaga magnettrummel. Trumli läbimõõt on 216 mm ja ta pöörleb kiirusega 3000 pööret minutis. Informatsiooni tihedus trumlil on 3,2 impulssi millimeetril, kusjuures trummel on jagatud 31 kanaliks. (32. kanal sisaldab markerid, 33. kanal aga nn. nullimpulsi, mida kasutatakse mäluseadme õige töö kontrollimiseks). Mäluseadme töötakti keskmine kestus on 10 millisekundit.

Välise mäluseadmena kasutatakse viiekanalilist perfolinti ning standardset telegraafitransmitterit, mis ühtlasi on sisendseadmeks. Väljundseadmeks on teletaip PTM, mis trükib tulemused mitmes tulbas paberilehele. Et teletaipi töötamiskiirus on ainult 330 märki minutis, siis on ka väljundkiirus väike — 30 arvu minutis.

Juhtimispuldile on välja toodud aritmeetilise seadme ühe registri ning juhtimisseadmete registrite signalisatsioon. Samuti asuvad seal lülitid üksikute arvude, käskude, aadresside jne. käsitsi valimiseks ning seadmed masina juhtimiseks. Aritmeetilise seadme registrite signalisatsioon asub vahetult aritmeetilises seadmes ning on nähtav läbi klaasuste, mis katavad kapi esikülge. Puldil olev masina automaatse peatamise lüliti peatab masina kas mingi ettenähtud käsu või aadressi kasutamise korral.

Toiteseade koosneb muundusagregaadist — kolmefaasilisest asünkroonmasinast, sünkroongeneraatorist ja alalisvoolugeneraatorist, millede pinged on stabiliseeritud (viimasel elektronregulaatoriga). Sünkroongeneraator (sagedusega 200 Hz) toidab elektronlampide kütteahelaid. Kõrgema sageduse kasutamine vähendab trafode ja alaldajate filtrite gabariite. Masina käivitamisel tõstetakse küttepingeid aeglaselt nullist alates, samuti viiakse nad väljalülitamisel aeglaselt nullini.

Elektron-arvutusmasin M-3 annab laialdasi võimalusi edasiarendamiseks. Tema sõlmede tegelik konstruktsioon lubab tõsta töökiiruse kuni 2000 tehteni sekundis, kuid kitsaskohaks on mäluseade magnettrumlil. Masina konstruktsioon võimaldab (vaid väikeste muudatuste tegemisega) täiendada arvutusmasinat 2048-pesalise ferriitmäluseadmega. Kuna M-3 on asünkroonne (s. t. puudub sünkroonimpulsside generaator ning järgmine tehe algab kohe pärast eelmise lõpetamist), siis kiirendab iga üksiku sõlme tegutsemiskiiruse kasv kogu masina tööd. Suurema kiiruse saavutamiseks vajab masin peale ferriitmäluseadme veel sisendseadmeks fototransmitterit ja väljundseadmeks kiirtrükkijat.

3. ELEKTRON-ARVUTUSMASIN «STRELA»

Elektron-arvutusmasinaid «Strela» toodeti pikema aja vältel, kuid nüüd on nad juba tehniliselt vananenud ja toodangust maha võetud. Kõik olemasolevad «Strela» tüüpi arvutusmasinad töötavad aga siiski edukalt edasi.

Elektron-arvutusmasinat «Strela» võib lugeda suurte arvutusmasinate hulka, kuna ta teostab 2000—3000 tehet sekundis, sisaldab 8000 elektronlampi, 50 000—60 000 pooljuhtseadist, tarvitab 140 kW võimsust. Selliste suurte arvutusmasinate kasutamine on end ka majanduslikult täiesti õigustanud, sest ühe miljoni tehte maksumuseks tuleb neil ca 30 rubla. Samal ajal vajab arvutaja miljoni tehte teostamiseks (aritmomeetriga) 3 aastat ning tema palgaks kulub 30 000 rubla.

Arvutusmasin «Strela» on paralleelse toimega, sünkroniseerimisimpulssidega jäigalt sünkroniseeritud. Operatiivmälu on elektrostaatiline, potentsialoskoop-tüüpi katoodkiiretorudega, sisaldab 2047 pesa (valikuajaga 20 μsec ja salvestusajaga 40 μsec). Peale selle kasutatakse veel dioodmälu — 256 pesa konstantide ning 240 pesa enamtarvitatavate alamprogrammide salvestamiseks. Dioodmälusse masina töötamise ajal midagi salvestada ei saa.

Väline mälu koosneb kahest (mõne eksemplari puhul ka kolmest) spetsiaalsest magnetofonist. Neis kasutatakse 125 mm laiust magnetofonilinti, kuhu informatsioon on salvestatud 43 paralleelse kanalina. Peale selle sisaldavad kolm kanalit veel tehnilistel põhjustel vajalikke märke. Informatsiooni paiknemise tihedus lindil on 1,4 impulssi millimeetril. Pöördumise korral välise mälu poole ei katke arvutused masina muudes osades. Magnetofonilint on jaotatud 253 tsooniks, igasse tsooni saab salvestada kuni 2000 arvu.

Arvude kujutamist arvutusmasinas «Strela» vaatlesime juba II peatükis. Käskude süsteem sisaldab kokku 22 (kolmeaadressilist) käsku aritmeetiliste ja loogiliste tehete sooritamiseks ja 16 suunamiskäsku. Selline suur suunamiskäskude arv võimaldab otseselt üle minna dioodmälusse salvestatud alamprogrammidele. Erinevalt mõningatest teistest masinatest on siin veel käsud, mis võimaldavad aritmeetilisi tehteid sooritada tervete arvuderühmadega.

Informatsioonivahetus arvutusmasinaga toimub vaid perfokaartide vahendusel. Programmi ja algandmete perforeerimine ning perforeeritud kujul saadud tulemuste trükkimine toimub täiesti lahus arvutusmasinast.

Aritmeetiline seade on arvutusmasinal «Strela» omapärane: iga erinev tehe teostatakse seadme erinevas osas ning iga detail omab vaid ühe funktsiooni (nii on masinas näiteks eraldi summaator, korrutaja jne.). See suurendab muidugi masina maksumust ja teeb töö ebakindlaks, kuid lihtsustab vigade avastamist. Konstruktiivselt koosneb masin 2—9-lambilistest blokkidest. Kuigi «Strela» tehnilise külje suhtes võib esitada õigustatud pretensioone, on masin matemaatiku seisukohalt laitmatu ja üks täiuslikumaid.

4. ELEKTRON-ARVUTUSMASIN M-2

Elektron-arvutusmasinat M-2 on seni valmistatud vaid üksikeksemplarides. Maksumuse ja gabariitide poolest võiks teda lugeda keskmise suurusega arvutusmasinaks, kuid kiiruse ja kasutusvõimaluste poolest on ta vaid vähe erinev suurtest arvutusmasinatest.

Aritmeetiline seade opereerib nii liikuva kui ka fikseeritud komaga arvudega. Masina registrid sisaldavad 33 kahendkohta ja algebralise märgi (34. kohal). Fikseeritud komaga töörežiimis on koma esimese koha ees ja kasutatavad kahendarvud on 33-kohalised. Liikuva koma režiimis sisaldavad 6 esimest kohta arvu järgu ja 27 järgmist mantissi (vt. § 4). Varemvaadeldud masinatega võrreldes on siin arvu märgi kujutamine teistsugune: «+»=1 ja «–»=0.

Arvutusmasin M-2 on kolmeaadressiline. Käsu kujutamisel on pesa jaotatud järgmise skeemi kohaselt:

$$\underbrace{1|0|0|1|1|1|1|1|0|0}_{\text{1. aadress}}|\underbrace{0|0|1|1|1|1|0|1|0|0}_{\text{2. aadress}}|\underbrace{0|1|1|0|0|1|0|1|1|0}_{\text{3. aadress}}|\underbrace{1|0|1|0}_{\text{tehte kood}}$$

Käsu üleskirjutamisel jaotatakse iga aadress kolmeks numbriterühmaks (2, 4 ja 4 numbrit) ning kirjutatakse ühe neljand- ja kahe kuueteistkümnendnumbrina. Kuueteistkümnendsüsteemi numbrimärkideks kasutatakse tavalisi numbreid ja tähestiku esimest kuut tähte ($a=10$, $b=11$, $c=12$, $d=13$, $e=14$, $f=15$). Näiteks viimasel skeemil toodud käsk omab üleskirjutatult kuju

$$2\ 7c\quad 0\ f4\quad 1\ 96\quad a$$

(neljandnumbrite eraldamiseks kuueteistkümnendnumbritest jäetakse nende vahele tavaliselt väike vahe).

Tabelis 9 on toodud mõned näited arvutusmasina M-2 käskudest.

Mäluseadmed on arvutusmasinas M-2 kolme tüüpi. Operatiivmälu pesade nr. 512—1023 ehitamiseks on kasutatud tavalisi katoodostsillograafi katoodkiiretorusid (tüüp 13-ЛО-37), mille ekraani välisele poolele on kondensaatori moodustamiseks kleebitud peen vaskvõrk. Võrgul on metallikihiga võrreldes see eelis, et ta võimaldab informatsiooni ekraanil visuaalselt kontrollida. Operatiivmälu pesad nr. 0—511 on ehitatud magnettrumlil[1].

Väliseks mäluks on ümberehitatud magnetofon, milles informatsioon salvestatakse tavalisele lindile (600 m linti salvestab 50 000 arvu). Informatsioonivahetus võib toimuda vaid elektrostaatilise mälu ja magnetofoni vahel.

Sisendseadmeks on kas foto- või elektromehaaniline transmitter. Esimene on eelistatavam, sest ta töötab ca 60

[1] Viimasel ajal on elektron-arvutusmasinates M-2 elektrostaatilisi mäluseadmeid asendatud ferriitmäluseadmetega.

Tabel 9

Tehte kood	Käskude kuju	Tehte sisu
0	0 00 0 00 *A* 0	Juhtimise tingimusteta edasiandmine käsule (*A*)
0	0 00 0 02 *A* 0	Juhtimine antakse üle käsule (*A*) ja ühtlasi minnakse üle liikuva komaga töötamise režiimi
0	0 00 0 0c *A* 0	Juhtimine antakse üle käsule (*A*) ja ühtlasi minnakse üle fikseeritud koma ning kahekordse täpsusega töötamise režiimi
6	*A* *B* *C* 6	$(B)<(C)$ puhul täidetakse järgmisena käsk (*A*), vastasel juhul jätkub käskude täitmine loomulikus järjekorras
c	*A* *B* *C* *c*	$(A)=(B)-(C)$

korda kiiremini. Väljundseadmed on samasugused nagu arvutusmasinal *M*-3.

Elektron-arvutusmasin *M*-2 paistab eriti silma oma ökonoomsuse poolest. Ta sisaldab vaid 1879 kõige tavalisemat elektronlampi, neist üle kahe kolmandiku tüüpi 6H8C. Kiiruse poolest on *M*-2 võrdne arvutusmasinaga «Strela» (2000—3000 tehet sekundis). Huvitav on märkida, et näiteks liitmine (tehte kood *d*) kestab fikseeritud koma puhul ca 462,5 μsec ja liikuva koma puhul 537,5—1387,5 μsec; jagamine (tehte kood 8) aga on vastavalt 1700 μsec ja 325—1512,5 μsec.

5. ELEKTRON-ARVUTUSMASIN БЭСМ

Suurim ja täiuslikum kodumaine arvutusmasin on БЭСМ [1]. Selle arvutusmasina esimene eksemplar on ehitatud ja töötab juba aastaid NSV Liidu Teaduste Akadeemia Täppismehaanika ja Arvutustehnika Instituudis Moskvas.

[1] БЭСМ = Быстродействующая Электронная Счетная Машина.

БЭСМ on suure töötamiskiirusega (kuni 13 000 tehet sekundis) ja mitte liialt kogukas (sisaldab umbes 5000 elektronlampi). Käsud on kolmeaadressilised. БЭСМ on liikuva komaga masin, mille pesa sisaldab 39 kahendkohta. Et astronoomilised arvutused nõuavad sageli suuremat täpsust, siis võib masina lülitada ümber ka kahekordse täpsusega töörežiimi (suhteline täpsus 10^{-18}).

Operatiivmäluna kasutatakse 1024 aadressiga ferriitmälu. Peale selle on masinas veel magnettrummel ning neli magnetofoni (kokku üle 120 000 arvu salvestamise võimalusega). Sisendseadmeks on fototransmitter (20 arvu sekundis), väljundseadmeks elektromehaaniline trükkija (15 rida sekundis) ja fototrükkija (200 arvu sekundis).

Arvutusmasina БЭСМ baasil on välja töötatud tüüp БЭСМ-2, mis kõigi näitajate osas ületab tunduvalt esialgse variandi.

*

Vaatleme järgnevalt mõningaid Ameerika Ühendriikide firmade poolt toodetavaid arvutusmasinaid. Kirjeldatavad suured arvutusmasinad ei paku meie toodanguga võrreldes midagi uut, kuigi on tunda, et masinate konstruktsiooni on mõjutanud järgmised asjaolud: 1) firmad püüavad teaduslik-tehniliseks otstarbeks ehitatud masinaid muuta kasutatavateks ka arvepidamises ja vastupidi; 2) orientatsioon arvepidamisele sunnib kasutama lisaseadmeid pidevaks arvutuste õigsuse kontrolliks ning suuri välise mälu seadmeid. Sellest tingituna on Ameerika Ühendriikide arvutusmasinates eriti põhjalikult välja arendatud kõik välisseadmed (väline mälu, sisend-väljundseadmed) ja ses osas on esialgu märgata veel meie mõningat mahajäämust.

6. ELEKTRON-ARVUTUSMASIN «UNIVAC»

Elektron-arvutusmasinaid «Univac» valmistab firma «Remington Rand Inc.» osakond «Eckert-Mauchly». Seeriatootmist alustati 1951. a., 1955. a. oli valmis või lõpetamisel 25 eksemplari.

Arvude kujutamine. Arvutusmasin «Univac I» on paralleel-järjestikuse toimega ja opereerib kahendsüsteemis kodeeritud numbrite või tähtedega. Üks arv koosneb

12 märgist. Iga märk on esitatud seitsmekohalise kahendarvuna, kus kuus esimest kohta tähendavad kas kümnendnumbrit, tähte või kirjavahemärki. Seitsmes koht on kontrollmärk, milleks valitakse kas 0 või 1 nii, et märgi (kui seitsmekohalise kahendarvu) ristsumma oleks paaritu. Märkide vahele jäetakse tühjad kohad. Üldse tuleb iga arv kujutada 91-kohalises pesas, mis informatsioonisisalduse poolest on ekvivalentne 72-kohalisega.

Kui masin töötleb numbrilist informatsiooni, siis esitatakse see kümnendsüsteemis, kusjuures iga kümnendkoht on kujutatud kahendarvuna (kokku 11 kümnendkohta ja märk). Sellest järeldub, et arvutusmasinas «Univac I» on informatsiooniga täidetud vaid 41,8% pesa kahendkohtadest (kui masinat kasutatakse automaatseks tõlkimiseks, siis on informatsioonisisaldus suurem).

Mäluseadmed. «Univac I» mälu koosneb seitsmest elavhõbedahoidlast, milledes igaühes informatsioon ringleb mööda 18 kanalit. Need 126 kanalit on kasutatud järgmiselt: 100 kanalit informatsiooni salvestamiseks, 7 kanalit temperatuuriregulaatorite tarbeks, 6 kanalit sisendseadmete jaoks, 6 kanalit väljundseadmete jaoks, 6 kanalit tagavaraks ja 1 kanal eriotstarbeks. Iga kanal mahutab kümme arvu, seega koosneb operatiivmälu 1000 pesast, millest igaüks sisaldab kas ühe arvu või kaks käsku. Üldse mahutab operatiivmälu 720 000 kahendnumbrit; mälust otsimise aeg on keskmiselt 222 µsec.

Lisamäluseadmena omab «Univac I» veel üks kuni kümme erilist magnetofoni «Uniservo», mis ühtlasi on osaliselt ka sisend- ja väljundseadmeteks. Need võimaldavad samaaegselt jätkata arvutusi, kirjutada ühele lindile, lugeda teiselt ja kerida ümber kolme linti. Magnetofonilint pole siin tavaline, tema aluseks on värvilise metalli sulam, mida katab magnetiline kate. Lindil on paralleelselt 8 kanalit — kuus neist sisaldavad informatsiooni, üks kontrollmärke ja üks markereid. Informatsioon kantakse «Uniservolt» operatiivmälusse üle blokkidena, igas blokis 60 arvu. Kogu ülekandeprotsess toimub programmis vastava käsu esinemisel automaatselt: käivituvad mootorid, lint liigub edasi kiirusega 2,5 meetrit sekundis ja pidurdub. 60 arvu ülekandmiseks kuluv aeg (koos käivituse ja pidurdusega) on 0,104 sekundit.

Kokku sisaldavad 10 magnetofoni 1,2 miljonit arvu (s. t. 109,2 miljonit kahendnumbrit). Kui aga arvestada veel seda, et magnetofoni lindirullid on vahetatavad, siis pole välise mälu maht üldse piiratud.

Trükkimisseadmeteks on kiirtrükkija (600 rida, à 130 märki minutis) ja üks või rohkem kirjutusmasinat «Uniprinter». Kiirtrükkija trükib tulemused tulpadena kuni 68 cm laiusele paberiribale, kasutades kokku 51 erinevat sümbolit (arvud, märk + ja –, ladina tähestiku tähed ja kirjavahemärgid). «Uniprinter» on tavaline elektrikirjutusmasin, mis trükib 10 märki sekundis, dešifreerides impulsse magnetofonilindilt.

Lisasisendseadmena kasutatakse elektrikirjutusmasinat «Unityper», mis üheaegselt teksti trükkimisega paberile kannab informatsiooni magnetofonilindile. Erilistele klahvidele vajutades saab vea korral kustutada vale märgi, arvu või terve 60-arvulise bloki. Informatsiooniga täidetud magnetofonilint juhitakse võrdlejasse, mille klaviatuuri abil teine masinakirjutaja kannab informatsiooni veel kord lindile. Lahkumineku esinemise korral ei saa ta klahvi vajutada ning peab selgitama, milles oli viga.

Arvutusmasina komplekti kuulub veel perforaator, mis võimaldab kanda perfokaartidele magnetofonilindil olevat informatsiooni (3,5 minuti jooksul perforeeritakse 450 m pikkuselt lindilt informatsioon 1800-le perfokaardile), ja seade informatsiooni ülekandmiseks perfokaartidelt magnetofonilindile.

Põhiline informatsiooni töötlemine toimub keskses arvutusseadmes. Informatsioon juhitakse sinna vaid operatiivmälust, mis omakorda on ühendatud kiirtrükkijaga, magnetofonidega ja nende kaudu «välismaailmaga».

«Univac I» töötamiskiirus pole kuigi suur: ta teostab sekundis kuni 710 liitmist-lahutamist, 330 korrutamist või 210 jagamist. Aadresside sobiva valikuga programmeerimisel saab tema kiirust aga tõsta. Üldiselt on masina kiirus määratud sünkroniseerimisimpulsside sagedusega — 2250 kHz.

«Univac I» on täisautomaatse programmjuhtimisega, kusjuures suur väline mälu võimaldab salvestada kõik alamprogrammid, mis tunduvalt vähendab programmeerija tööd. Näiteks sisaldab väline mälu programme vajalikumate funktsioonide arvutamiseks kiirelt koonduvate ridade abil; iteratsioonivalemeid; programme, mis võimaldavad selgitada, mitu lähendust või rea liiget tuleb arvutada.

Suur osa (30%) arvutusmasina «Univac I» detaile on kasutatud korrasoleku automaatseks kontrollimiseks. Aritmeetiline seade ja juhtimisseade on dubleeritud. Ülalmainitud paaritut ristsummat kontrollitakse igal sammul ja see võimaldab ära hoida vead informatsiooniülekandel masina ühest osast teise (samuti üksikute numbrikohtade muutumist mälus).

Arvutusmasina «Univac I» tootmise ja ekspluateerimise kogemused võimaldasid asuda täiuslikuma arvutusmasina väljalaskmisele (1957. a, alates). Selleks on «Univac II», mille konstruktsiooni kallal töötas aastaid firma «Remington Rand» 3 osakonda. Põhilised erinevused arvutusmasinast «Univac I» on järgmised. Elavhõbemälu on asendatud 2000-pesalise ferriitmäluga, mida soovi korral saab laiendada kuni 10 000 pesani. Välise mälu maht ja kiirus on kaks korda suuremad (ainult informatsiooni lindil paiknemise suurema tiheduse arvel), «Uniservode» arv on 10 asemel 16. Arvutusmasin tervikuna on kaks korda kiirem oma eelkäijast.

7. ELEKTRON-ARVUTUSMASIN «UNIVAC-SCIENTIFIC»[1]

Seda arvutusmasinat, tüübinimetusega ERA 1103, toodab firma «Remington Rand» teine osakond «Engineering Research Associates Div». Üldse esineb temast kolm varianti: ERA 1103, ERA 1103 A ja ERA 1103 B. Erinevalt teistest «Univac»'idest on ta rohkem kohandatud teaduslik-tehnilisteks arvutusteks. Temaga on võimalik lahendada ülesandeid ajalise muutujaga ja anda tulemusi välja mingis etteantud ajamastaabis. Eeskätt on see masin aga arvestatud rakendamiseks lennuki- ja raketitööstuses. Masina hind on umbes 1 miljon dollarit.

Arvutusmasin ERA 1103 opereerib 35-kohaliste kahendtäisarvudega (36. koht tähistab märki + või —). Mälu on elektrostaatiline, valmistatud nii 1024 kui 4096 pesaga (alates 1954. aastast valmistatakse seda tüüpi masinaid vaid ferriitmäluga). Peale operatiivmälu on olemas veel magnettrummel, millele on paigutatud 4096 pesa.

Väline mälu on tüüpide ERA 1103 ning ERA 1103 A ja B puhul erinev. Viimastel on selleks standardsed «Uniservod». Tüübid A ja B erinevad omavahel selle poolest, et B omab ka neli liikuva komaga tehet.

[1] Scientific — teaduslik (ingl. k.)

Kuna ERA 1103 opereerib vaid täisarvudega, siis on aritmeetiline seade ületäitumise vältimiseks (eeskätt korrutamise puhul) ehitatud 72-kohalisena. Kõik 40 käsku on valitud masinate ekspluateerijate soove arvestades. Nii on näiteks tunduvalt lihtsustatud erikäskude abil maatriksite korrutamist.

Võrreldes arvutusmasinaga «Univac II» on ERA 1103 tüüpi masinad aeglasemad ja umbes 3 korda vähem ruumi nõudvad (110 m²). Puudub eriaparatuur korrasoleku automaatseks kontrollimiseks: korrasolekut kontrollitakse profülaktiliselt, sünkroniseerimisimpulsside sageduse tõstmisega 500 kHz-lt 750 kHz-le. Puudub ka eraldi sünkroniseerivate impulsside generaator, mille asemel kasutatakse markerimpulsside põhisageduse (125 kHz) neljandat harmoonilist.

8. ELEKTRON-ARVUTUSMASINAD IMB SEERIA 700

Firma IBM[1] on võimsaim USA elektron-arvutusmasinaid tootev ettevõte. Juba pikemat aega on tuntud IBM väikesed arvutusmasinad, tabulaatorid jne. Nüüd toodetakse aga ka nii väikesi (seeria 600) kui suuri (seeria 700) elektron-arvutusmasinaid.

Seeriasse 700 kuuluvad tüübid 701, 702, 704, 705 ja 709. Kõikide nende aluseks on tüüp 701, mille esimesed eksemplarid töötavad 1953. aastast alates. 1954. a. järgnes tüüp 702, 1956. a. tüübid 704 ja 705 ning 1957. a. tüüp 709. Tüüp 701 on kasutatav büroodes arvepidamise automatiseerimiseks, kuid samuti ka teaduses ja tehnikas. Tüüp 704 on ette nähtud ainult teaduslik-tehniliseks otstarbeks, tüübid 702 ja 705 aga vaid büroodes kasutamiseks. Masinate hind ulatub 1,5 miljoni dollarini (tüübi 709 puhul isegi üle 2,5 miljoni dollari).

Tüübi 701 põhiosaks on aritmeetiline seade koos juhtimisseadmega. Sellega on vahetult ühendatud üks või kaks elektrostaatilise mälu blokki (à 2048 pesa). Operecritakse 36-kohaliste kahendarvudega. Tüübi 704 puhul on elektrostaatiline mälu asendatud sama mahuga ferriitmäluga. Otseselt operatiivmäluga on ühendatud 4 magnettrumlit, mis sisaldavad kokku 10 000 (tüüp 701) või 20 000 (tüüp 704) pesa. Kolmandaks mäluliigiks on väline mälu 7-kanalilisel magnetofonilindil. Kogu kasutatav magneto-

[1] IMB = International Business Machine Co.

fonide komplekt võimaldab salvestada 40 või 324 miljonit kahendnumbrit (vastavalt tüüpidele 701 või 704).

Arvutusmasinad seeriast 700 on üheaadressilised. Arvutusmasina IBM-701 kiirus on 2000 tehet sekundis, aga (nii väidab firma) oskusliku programmeerimise puhul saab selle tõsta 14 000 käsuni sekundis. Tüüp 704 on täiuslikum ja kiirem — oskusliku programmeerimise puhul ulatuvat tema kiirus 41 700 käsuni sekundis. Ka osutub ta kasutamisel paindlikumaks, sest aritmeetiline seade ei võta sisend- ja väljundprotsessidest osa (erinevalt tüübist 701). Sisend-väljundseadmeteks on nendes masinates peale magnetofonide veel perfokaartide kompaja, kiirtrükkija ja perforaator.

9. ELEKTRON-ARVUTUSMASIN «STRETCH»

Elektron-arvutusmasinat «Stretch» ehitab firma IBM juba alates 1956. a. novembrist (Ameerika Ühendriikide Aatomienergia Komisjoni tellimisel), kuid seni pole ta veel lõplikult valmis. Selle masina kohta on teada järgmist: uute elementide kasutuselevõtuga, kõigi seadmete üheaegse paralleelse tööga ja väga läbimõeldud käskude süsteemiga loodetakse tema kiirust tõsta umbes 100—200 korda (võrreldes seni ehitatud arvutusmasinatega). Arvutusmasin koosneb kolmest sektsioonist:

1) sisend-väljundsektsioon koos mäluseadmete-magnetofonidega;

2) järjestikune eelarvutusseade algandmete eeltöötlemiseks;

3) aritmeetiline seade koos juhtimisseadmega.

Eelarvutusseade võib töötada nii kahend- kui ka (kahendarvude abil kodeeritud) kümnendsüsteemis, nii fikseeritud kui ka liikuva komaga. Kiirus on suur, nimelt kestab liitmine 2—3 μsec, korrutamine — 5—15 μsec, loogilised tehted — 2 μsec.

Aritmeetiline seade peab töötama veelgi suurema kiirusega — keskmiselt 2 miljonit liitmist-lahutamist või 0,5 miljonit korrutamist sekundis.

Operatiivmälu täieliku töötakti kestus on 0,5 μsec (812 pesa), välise mälu töötakt — 4 μsec (1 miljon pesa). Välises mälus on informatsioon salvestatud magnetilise kattega ketastele. Arvutusmasinas «Stretch» on kasutatud hulgaliselt pooljuhtseadmeid, mis töötavad rahuldavalt sagedusel 10 MHz.

KIRJANDUS

A. Üldine

Х. Рутисхаузер, А. Шпайзер, Э. Штифель. «Электронные цифровые вычислительные машины с программным управлением». Журнал «Вопросы ракетной техники» № 2, 3, 4, 5. 1952 г.

А. И. Китов. Электронные цифровые машины. Изд-во «Советское радио». Москва, 1956.

С. А. Лебедев. Электронные вычислительные машины. Изд-во АН СССР, Москва, 1956.

Н. А. Архангельский, Б. И. Зайцев. Автоматические цифровые машины. Изд-во физ.-мат. литературы, Москва, 1958.

И. А. Полетаев. Сигнал (о некоторых понятиях кибернетики). Изд-во «Советское радио», Москва, 1958.

Ф. В. Майоров. Электронные вычислительные машины и их применение. Военное изд-во Мин. Обороны СССР, Москва, 1959.

Н. Кобринский. В. Пекелис. Быстрее мысли. Изд-во «Молодая гвардия», Москва, 1959.

А. И. Китов, Н. А. Криницкий. Электронные вычислительные машины, Изд-во СССР, Москва, 1959.

E. C. Berkeley, L. Wainwright. Computers — Their Operation and Applications. New York, 1956.

B. Spetsiaalne

М. Уилкс, Д. Уилер, С. Гилл. Составление программ для электронных счетных машин. Изд-во иностранной литературы, Москва, 1953.

Синтез электронных вычислительных и управляющих схем. Изд-во иностранной литературы, Москва, 1954.

«Пути развития советского математического машиностроения». Доклады конференции, часть I, II, III. Москва, 1956.

Ф. Б. Майоров. Электронные цифровые вычислительные устройства. Госэнергоиздат, Москва-Ленинград, 1957.

Быстродействующая вычислительная машина М-2. Под редакцией чл.-корр. АН СССР И. С. Брука. Изд-во технико-теоретической литературы, Москва, 1957.

В. Д. Моисеев. Автоматические вычислительные машины и их применение на железнодорожном транспорте. Трансжелдориздат, Москва, 1957.

Р. К. Ричардс. Арифметические операции на цифровых вычислительных машинах. Изд-во иностранной литературы, Москва, 1957.

Б. М. Каган, Т. М. Тер-Микаэлян. Решение инженерных задач на автоматических цифровых вычислительных машинах. Госэнергоиздат, Москва-Ленинград, 1958.

Вычислительные машины (СЕАК и ДИСЕАК) Национального бюро стандартов США. Машгиз, Москва, 1958.

Э. Бут, К. Бут. Автоматические цифровые машины. Изд-во физ.-мат. литературы, Москва, 1959.

J. Millmann, H. Taub. Pulse and Digital Circuits. New York, 1956.

R. K. Richards. Digital Computer Components and Circuits. New York, 1957.

SISUKORD

Каазик Юло Янович, Синисоо Марк Альфредович,
Салум Хенн Лудвигович

ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ

На эстонском языке

Эстонское Государственное Издательство
Таллин, Пярнуское шоссе, 10

*

Toimetaja L. A b o Tehniline toimetaja Ü. L a u l

Korrektor H. N a s s a r

Ladumisele antud 9. I 1960. Trükkimisele antud 11. IV 1960. Paber 54×84, 1/16. Trükipoognaid 12,25. Formaadile 60×92 kohaldatud trükipoognaid 10,05. Arvutuspoognaid 12
Trükiarv 2000. MB-02445. Tellimise nr. 422.

Hans Heidemanni nim. trükikoda, Tartu, Ülikooli 17/19.

Hind rbl. 4.90

3—5

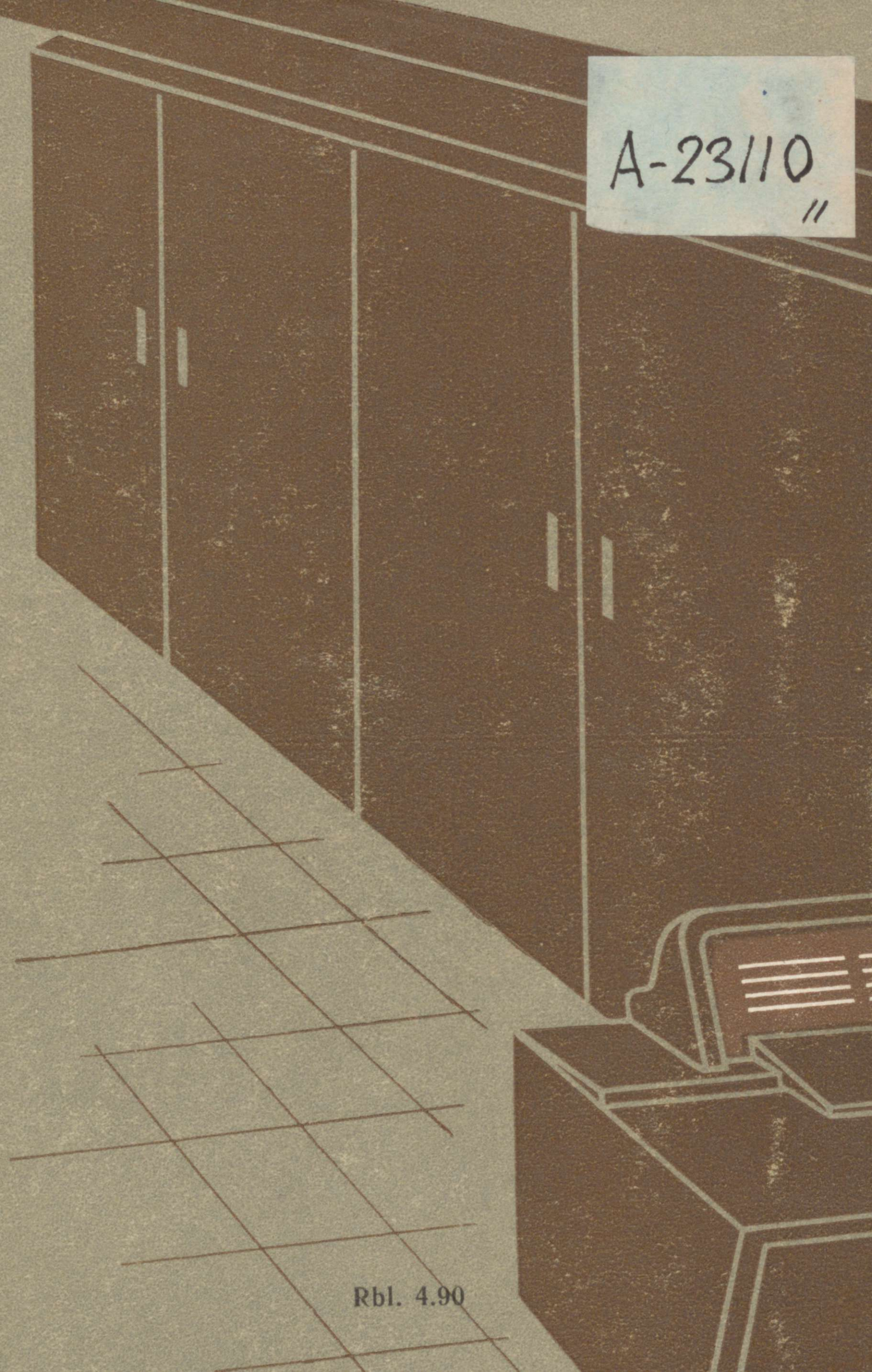
Lähemal ajal ilmuvad:

N. Wiener

Küberneetika. Tõlge inglise keelest. 180 lk.

E. Künnap

Automaatreguleerimine. 260 lk.



Rbl. 4.90