

<https://doi.org/10.3176/phys.math.1975.4.01>

*B. TAMM*

УДК 681.3

## MUUTUVA STRUKTUURIGA ARVUTISÜSTEEMIDE KASUTAMISE PERSPEKTIIVE

Elektronarvuti kui arvutussüsteemide, aga viimasel ajal ka juhtimis-süsteemide põhikomponent on tänapäeval kõigile hästi tuntud. Tuntud on ka tema funktsionaalne struktuur ning asjaolu, et see on põhimõtteliselt sarnane inimese, looduse täiuslikema infotöötlemiseadme omaga.

1. ja 2. põlvkonna arvutitel oli enamasti üksainus või paar sisendseadet ning mõni üksik väljundseade. Seega oli arvuti kompaktnel, tema struktuuri määras valmistav tehnik (joon. 1). Arvutite universaalsuse tagas asjaolu, et nende riistvara võimaldas realiseerida väga mitmesugust tarkvara.

Kuigi nende arvutite sisend- ja väljundseadmed töötasid vähemalt suurusjärgu võrra aeglasemalt tsentraalprotsessorist ehk keskseadmest, polnud sellest väga suurt häda. Seda peeti õigustatult paratamatuks ning kogu arvuti tervikuna jäi mingi tingliku harmoonilisuse piiridesse. Oldi harjunud, et arvuti sisendile andis programmi operaator, seejärel oodati, kuni ülesanne lahendatud, ning väljundit, kust arvude, teksti, tabelite ja graafikutena ilmusid tulemused, tarbis jälle inimene, enamalt jaolt programmi koostaja. Nii kasutati ja kasutatakse veel praegugi arvuteid tuhandetes arvutuskeskustes. Ka Eestis on selline skeem momendil pea-aegu (kuid mitte enam päris) ainuvalitsev. Kuuekümnendate aastate teine pool ja seitsmekümnendad aastad on aga toonud nii arvutustehnikasse kui ka arvutiteadusse üldse olulisi korrektsioone.

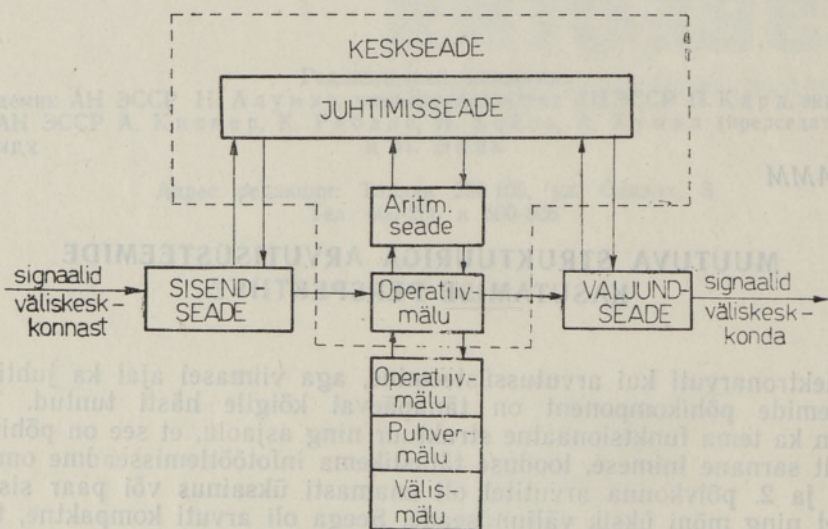
Esimeses lähenduses võiks neid jagada kolme suurde rühma: 1) arvuti on muutunud suurema, reaalsamas kulgevaid protsesse juhtiva tehissüsteemi osaks; 2) arvuti üksikute sõlmede tehniline areng on kulgenud äärmiselt kiiresti, kuid ebaühtlaselt; 3) tarkvara areneb kvalitatiivselt nii makro- kui ka mikroprogrammeerimise suunas.

Vaatleme neid muutusi lähemalt.

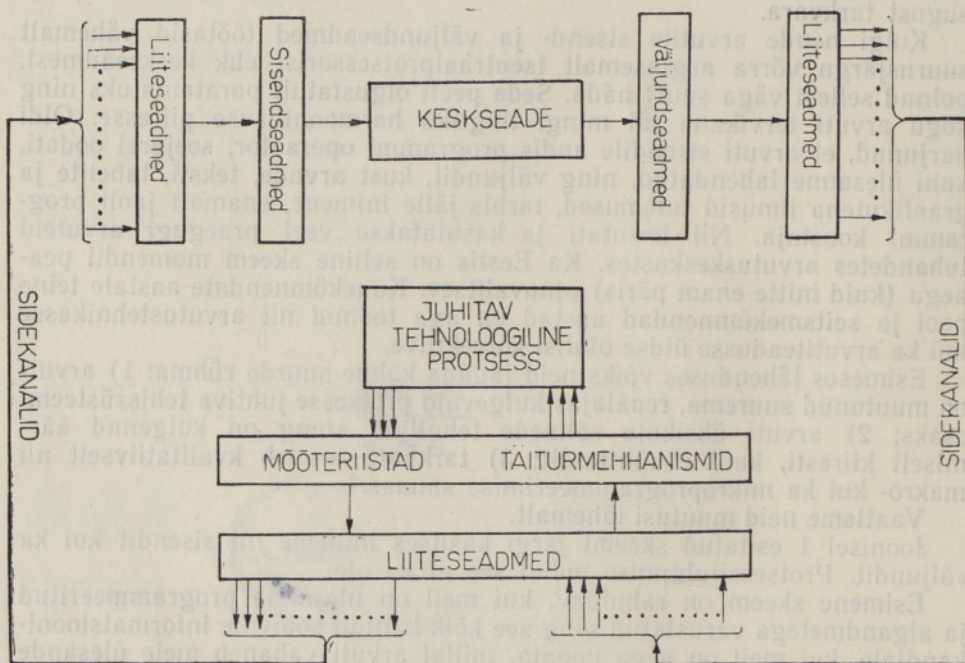
Joonisel 1 esitatud skeemi järgi käsitles inimene nii sisendit kui ka väljundit. Protsessijuhtimise puhul see nii ei ole.

Esimese skeem on rahuldav, kui meil on ülesanne programmeeritud ja algandmetega varustatud ning see kõik kantud sobivale informatsioonikandjale, kui meil on aega oodata, millal arvuti vabaneb meie ülesande lahendamiseks, ja lõpuks, kui meid rahuldavad paberile või magnetlindile salvestatud tulemused, mida me siis reeglina hakkame käsitsi, arvutist sõltumata analüüsima. Kõik see sobib küll paljude arvutuste ja ülesannete lahendamiseks, kuid mitte reaalsamas toimuvate protsesside juhtimiseks, kus iga eelnev sündmus määrab rangelt järgmise nii ajalisel kui ka parameetrisel (joon. 2). Siin pole algandmete täpsed väärtused ette

teada, nad saadakse protsessi juhtimise kestel ning sageli tuleb neid kohe kasutada. Neid ei saada enamikul juhtudel paberile kirjutatult või perforeeritult, vaid protsessi kajastavatelt mõõteriistadelt, protsessi tajuritelt. Neid tuleb muundada ühest füüsilisest suurusest teiseks, kodee-



Joon. 1.



Joon. 2.

rida, edastada arvutile, muundada sisestussobivaks informatsiooniks, arvutada keskseadmes oleva programmi järgi (see on koostatud protsessi matemaatilise mudeli põhjal), tulemused kodeerida ja konverteerida

edastatavaks informatsiooniks, mis sidekanalite kaudu kulgevad tagasi protsessi juures asuvasse liiteseadmetesse. Need omakorda muundavad informatsiooni täiturmehhanismidele sobivateks juhtsignaalideks, mis siis tegelikult juhivad protsessi. See kõik peab toimuma nii kiiresti, et ta rahuldaks juhitava protsessi loomulikku ajalist kulgu, ning peale selle sidusalt (*on-line*).

On kerge näha, et niisuguses juhtimissüsteemis on arvuti ainult üks osa kogu süsteemist. Samuti on selge, et ühe või paari aeglaselt töötava sisend- ja väljundseadmega varustatud arvutiga pole siin midagi peale hakata.

Mis puutub juhivatessesse protsessidesse, siis on neid palju ja mitut liiki: pidevad (keemiatööstuse protsessid, näit. tsemendi, tselluloosi ja paberi, plastmassi tootmine, metallurgiatööstuse protsessid jne.), tsükli- lised (keemia-, toiduainete-, metallurgiatööstuse jne. protsessid) ja disk- reetsed (aparaadi-, masina-, kergetööstuse jne. protsessid); omaette kogum protsesse on seotud energia tootmise ja jaotamisega (üldiselt on need pidevad). Tähtsa liigi reaajas kulgevate ja sidusalt juhivate prot- sesside hulgas moodustab teadusliku eksperimendi automaatne juhtimine.

Seega siis, kui 1. ja 2. põlvkonna arvutitel oli üksnes võimsa kalku- laatori osa, muutusid 3. põlvkonna arvutid ühtlasi juhtimisseadmeteks. Tavaliste, nn. universaal-arvutite kõrvale ilmusid juhtimis-arvutid.

Peatume nüüd teisel eespool mainitud põhjuste rühmal, mis soodustas sellist arenemist. Nagu öeldud, oli see arvuti riistvara üksikute osade äärmiselt kiire ja ebaühtlane tehniline arenemine. Milles see seisnes?

Edusammud tahke keha füüsikas, pooljuhtelektroonikas ja magnetooni- kas, samuti paljude täiesti uute tehnoloogiliste võtete ilmumine võimal- dasid tõsta keskseadmete töökiirust veel paari suurusjärgu võrra, mikro- miniatüriseerida kogu riistvara, luua uusi, väga tõhusaid mäluseadmeid ning ka uusi sisend- ja väljundseadmeid (samal ajal suurenes viimaste töökiiruse mahajäämus kogu ülejäänud aparatuuri omast veelgi). Sel- lise arengu tulemustest peab eelkõige mainima räni- ja metalloksüüd- pooljuhtelemente, mille lülitusaeg on kuni 0,02 nanosekundit, s. t. nad võivad teha kuni 50 miljardit lülitust sekundis. Niisugused elemendid, mille geomeetrilised mõõtmed ületavad ainult suurusjärgu võrra suure- mate molekulide mõõtmed, koondatakse keeruliste valmisskeemidena nn. keskmistesse või suurtesse ehk lausintegraalskeemidesse. Need skeemid, mis sisaldavad sageli mõni tuhat aktiivset ja passiivset elementi, on väga kompaktsed ning nende joonmõõtmeid väljendatakse millimeetrites. Näi- teks on uusimate ja praegu ühtlasi väiksemate mikroarvutite kesksead- mete aparatuurse osa mõõtmed  $3 \times 3 \times 3$  mm; pakituna monoliitsesse kaitsekihti, mille välispinnal on üksnes sisendite ja väljundite mikrokle- mid, on kogu arvuti mõõtmed  $0,5 \times 0,5 \times 1,5$  tolli [1,2].

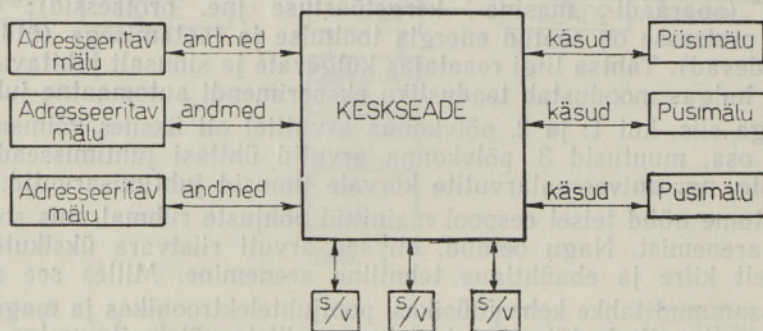
Kui niisuguste väljapaistvate võimete ga keskseadmega sidestada mõni 2. või 3. põlvkonna arvuti mäluseade (magnetketas, ferriitkuup, magnetlint), mille töötsükkel on keskseadme omast paar suurusjärku aeglasem, siis annulleerib see viimase efektiivsuse peaaegu täielikult. Pealegi on niisugune seade võimeline käsitlema suure mahuga kiiret mälu, kuid mainitud mäluseadmete töökiirus langeb mahu suurenedes kiiremini lineaarsest sõltuvusest. Milles on siis väljapääs?

Esiteks tuleb üks suur adresseeritav mäluplokk, mida kasutavad 2. ja 3. põlvkonna arvutid, tükeldada paljudeks väiksemateks osadeks, kuid nii, et neist igaühe poole võib eraldi pöörduda. Teiseks, need mälusead- med peavad koosnema elementidest, mille tsükli- aeg on mitu suurusjärku väiksem kui eespool mainitud ketastel, lintidel ja trumlitel, mis on isegi

10–100 korda kiiremad ferriitelementidest. Sellepärast on asutud kasutama pooljuht- ja magnetkilemälusid.

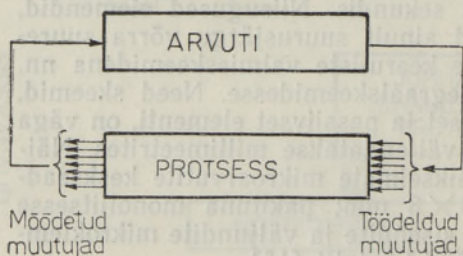
Asjaolu, et nüüd on ühe suure mäluaseadme asemel palju väikese, mõjub loomulikult ka programmeerimisele. Ja esimene asi, mis siit järgneb, on see, et tuleb eraldada lahendatava ülesande andmed (mis muutuavad nii ülesandest ülesandesse kui ka ühe ülesande jooksul) käskudest (mis kujutavad endast tegutsemisjuhiseid ning on märksa püsivama iseloomuga). Esimeste jaoks sobivad uued väikesed adresseeritavad mälad, teiste jaoks püsivamad (nn. ainult-loe-mälad) [3,4].

Kui nüüd meelde tuletada, et selle komplekti harmoonilise töö tagamiseks on vaja suur hulk sisend- ja väljundseadmeid, sest nende töökiirus jääb väga palju maha ülejäänud elementide omast, jõuamegi järgmise põhimõttelise skeemi juurde (joon. 3).



Joon. 3.

Õeldu täienduseks: kõiki mainitud elemente on võimalik kerge vaevaga monteerida mitmekihilistele trükiplaatidele, kuhu kas fotokeemilisel või mõnel muul meetodil on eelnevalt kantud ühendusskeem. Peale sisend- ja väljundseadmete mehhaaniliste osade on kõiki teisi osi võimalik niimoodi komplekteerida ja ka vahetada.

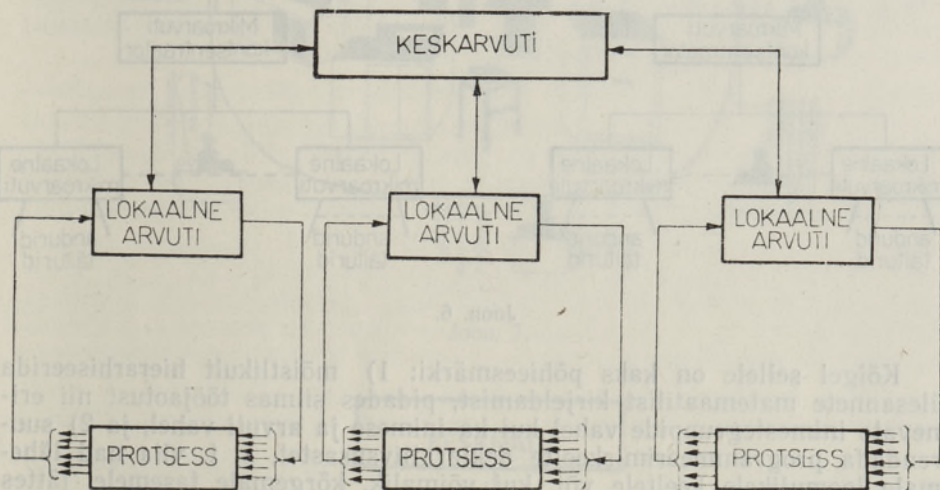


Joon. 4.

Kui meil on mingi protsessi juhtimiseks kasutada suur või keskmine 2. või 3. põlvkonna arvuti, peame seda tegema joonisel 4 esitatud lihtsustatud põhimõtteskeemi järgi. Siin on arvuti vastutav kogu protsessi eest. Kui arvutiga midagi juhtub, laguneb ka protsess. Kõik muutujad tuleb edastada arvutisse ja jälle tagasi. See teeb kommunikatsiooni keeruliseks, mittetöökindlaks, rääkimata sellest, et sidesüsteem läheb kallimaks kui kõik muu. Tarkvara, mis samuti peab rajanema tsentraliseeritud põhimõttel, osutub keeruliseks, kalliks, halvasti käsitletavaks. Süsteem on paindumatu. Kui näiteks tuleb juurde üks juhtimiskontuur (protsess muutub) või muudetakse juhtimisalgoritmi, tuleb oluline osa tarkvarast ümber teha (vahel ka muuta kommunikatsiooniskeemi), mis viib

süsteemi hulgaks ajaks rivist välja. Kaasaja arvutustehnika riistvara on liiga võimas ja liiga kallis selleks, et teda kasutada nii, nagu kasutati ja kasutatakse kalkulaatorarvutit. Arvutused näitavad, et kui ühendada keskmine kaasagne miniarvuti (umbes «Minsk-32» võimsusega) raalreguleerimise otstarbel neljast juhtimisahelast koosnevasse protsessi, on tema keskseade 99,5% ajast töötä [5,6].

Hoopis teine on lugu, kui rajame juhtimise teatud mõistlikule hierarhiale, jaotame ülesanded üksikute juhtimistasemete vahel ning ühtlasi detsentraliseerime juhtimise. Joonisel 5 on juhitava protsessi kolm juhtimiskontuuri üksteisest eraldatud, igaühe andmehõive ja otsene numbriline juhtimine on usaldatud lokaalsele mini- või mikroarvutile ning keskele suuremale arvutile jääb üksnes superviisor- ehk ülajuhtimine (seadesuuruste juhtimine).



Joon. 5.

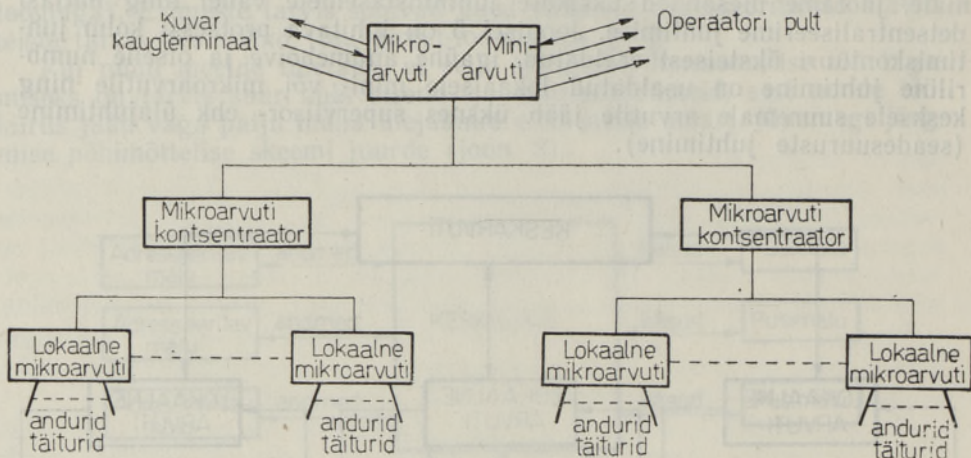
See võrreldamatult paindlikum skeem edastab keskarvutisse ainult tühise osa kogu informatsioonist, selle, mida seal asuvate programmidega arvutamiseks on tõesti tarvis. Seega langeb ta sisend- ja väljundseadmete koormus paljukordselt ja keskarvuti kasutamise täitetur tõuseb. Lokaalsed arvutid on operatiivsemad, ühe väljalangemine mõjub ainult protsessi ühele osale ning superviisorjuhtimist on võimalik ka siis jätkata, niisamuti on protsessi muutudes võimalik kogu arvutisüsteemi kergesti ümber teha. Selline skeem võimaldab detsentraliseerida struktuuri ka ruumiliselt. Nimelt paigutatakse lokaalsed arvutid vahetult neile alluva protsessi juurde, mis lihtsustab saabuvate signaalide muundamist, konverteerimist ja teisi liiteseadmete kompetensi kuuluvaid ülesandeid.

Sageli kasutatakse veelgi rohkemate astmetega struktuure, kus juhtimisfunktsioonid on nii algoritmiliselt kui ka aparatuurselt kindla hierarhia järgi jaotatud. Selliseid nimetatakse detsentraliseeritud intelligentsiga struktuurideks. Üht niisugust kujutab joonis 6.

Kolmas rühm arvutiteadust ja -tehnikat oluliselt mõjustavaid põhjusi käsitleb, nagu algul märgitud, tarkvara ja programmeerimist.

1. ja 2. põlvkonna arvutite ajal programmeeriti peamiselt nii, nagu oli mugav arvutile, kahjuks aga mitte inimesele. Selleks olid muidugi omad paratamatud põhjused. Kuuekümnendail aastail hakkas tarkvara-

teoria laienema eelkõige makroprogrammeerimise suunas, mis kah juba sisaldas detsentraliseeritud intelligentsi sätteid. Assamblerkeeltelt mindi üle protseduurkeeltele, neilt omakorda probleemorientatsiooniga keeltele. Seoses operatsioonisüsteemide tekkimisega eraldusid viimastest nn. tarbijapaketid. 3. põlvkonna arvutitele koostatakse ka integreeritud programmeerimissüsteeme.



Joon. 6.

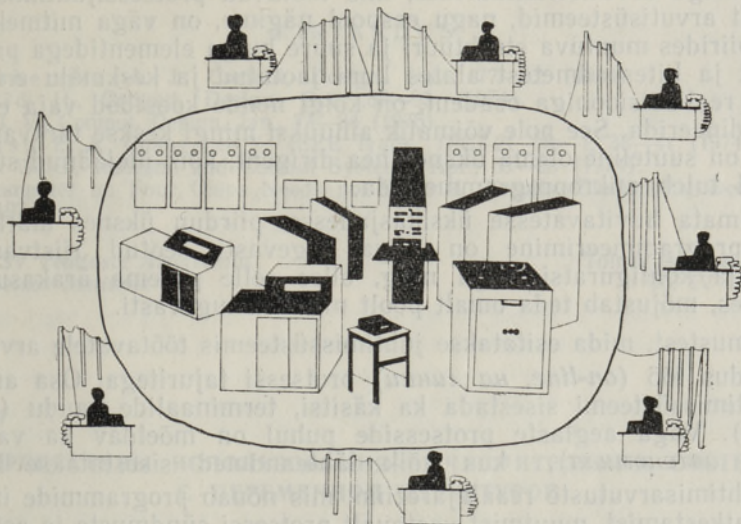
Kõigel sellele on kaks põhieesmärki: 1) mõistlikult hierarhiseerida ülesannete matemaatilist kirjeldamist, pidades silmas tööjaotust nii erinevate inimestegruppide vahel kui ka inimese ja arvuti vahel, ja 2) suurendada programmeerimiskeelete deklaratiivsusastet, s. t. viia nad lähemale loomulikele keeltele või, kui võimalik, kõrgemale tasemele, jättes informatsioonimahukate mõistete ja operatsioonide arvutitele vajalikeks elementaarteheteks lahutamise arvuti enese teha. Seda teevadki teatavasti kõiksugused kompailerid, translaatorid, programmprotsessorid.

Seoses suurte ja keskmiste universaal-arvutite keskseadmete võimsuse ja ka mäluseadmete mahu järsu kasvuga ning sisend- ja väljundseadmete intelligentsemaks muutumisega (programmeeritavad, autonoomse mäluga, paljutasemeliste katkestussüsteemidega, s. t. interaktiivsed seadmed) on võimalik ja otstarbekohane neid protsessorite külge ühendada suurel hulgal — sadu, isegi tuhandeid. See võimaldab ajajaotussüsteemi olemasolu korral multiprogramset tööd ning — kuivõrd mainitud seadmed realiseeritakse kaugterminaalidena — teeb tarbija praktiliselt sõltumatuks keskarvuti asukohast.

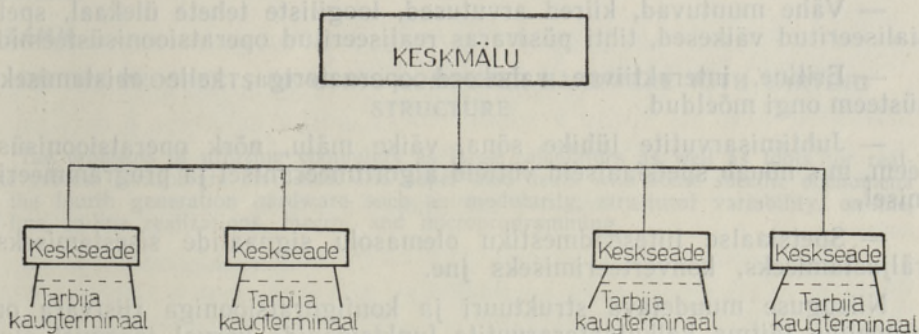
Niisugune on kollektiivse kasutamise arvutisüsteem ja see näib olevat perspektiivsemad suurte ja keskmiste 3. põlvkonna arvutite kasutamise viise (joon. 7), eriti kui arvuti ei tööta sidusalt mõne reaajas kulgeva protsessiga. Kui on kasutada suur ja mitmenivooline mäluseadmete komplekt, on sageli otstarbekas sellega kui struktuuri keskse mooduliga ühendada teatav arv keskseadmeid, millest igaüks on seotud oma tarbijategrupiga (joon. 8). Niisugust arvutistruktuuri ja vastavat töörežiimi nimetatakse multiprotsessoriliseks.

Nagu eelnevalt näha, tuleb arvuti keskseadmel (ja tema osadel), sõltumata sellest, kas ta töötab reaajas ja juhib sidusalt protsessi või lahendab ajajaotusrežiimis (samal ajal ka vallasrežiimis, *off-line*) teaduslik-tehnilisi, infotöötlemis- või muud ülesandeid, ikka pidevalt suhelda

suure hulga sisendite, väljundite, mälu erinevate osade ja muu aparatuuriga. Nende poole tuleb pöörduda, neid aktiveerida, välja ja ümber lülitada, valmistada ette informatsiooni vastuvõtuks, panna aktiivsesse ooteseisundisse jne. Kõike seda tuleb teha tihti kilohertsiliste sagedustega ja kuni mikrosekunditeni mõõdetava täpsusega.



Joon. 7.



Joon. 8.

2. põlvkonna arvutite lihtsa struktuuri ja aeglase töö tõttu tuli selliseid vajadusi harva ette ja need programmeeriti iga ülesande programmi vajalikkude kohta. 3. põlvkonna arvutisüsteemide puhul oleks niisugune meetod mõttetu. Standardoperatsioonid tuli eraldada tarkvarast. Tekkis uus mõiste — püsivara. See koosneb teatud standardprotseduuridest, mis ei muutu lühema või pikema aja jooksul. Seetõttu nad programmeeritakse ning seejärel realiseeritakse programm aparatuurselt. Seega on püsivara tark- ja riistvara vahepealne.

Kuivõrd püsivaras salvestatud protseduure tuleb arvutil korrata väga palju kordi, on ülimalt oluline, et see toimiks maksimaalse kiirusega ja samal ajal kasutataks minimaalselt arvuti riistvara ressursse. Nende nõudmiste täitmiseks ei piisa programmi kirjutamisest isegi arvutile kõige lähemas keeles, masinkoodis, vaid tuleb laskuda masinkoodi iga tehte

arvutis realiseerimiseks vajalike elementaartehete ja nende taktide tasemele. Niiviisi uut moodi programmeerides on jälle võimalik võita suurusjärke. Selles seisnebki mikroprogrammeerimise olulisi eeliseid.

Niisama hädavajalik on mikroprogrammeerimine detsentraliseeritud intelligentisiga arvutistruktuurides, mis töötavad protsessijuhtimissüsteemides. Et arvutisüsteemid, nagu eespool nägime, on väga mitmekesised, laiades piirides muutuva struktuuri ja suure hulga elementidega protsessi tajuritest ja liiteseadmetest alates kuni jaotatud ja keskmälu erisuguse tehnilise realisatsiooniga osadeni, on kõigi nende koostööd vaja efektiivselt koordineerida. See pole võimalik ainuüksi mingi keskse tarkvara abil. Viimane on suuteline olema üksnes hea dirigent, kõik ülejäänud süsteemi moodulid tuleb mikroprogrammeerida.

Laskumata huvitavatesse üksikasjadesse, piirdun üksnes märkusega, et mikroprogrammeerimine on väga tugevasti seotud riistvara elementide ja konfiguratsiooniga ning, olles selle parema ärakasutamise teenistuses, mõjustab teda omalt poolt niisama tugevasti.

Tingimustest, mida esitatakse juhtimissüsteemis töötavatele arvutitele:

— Sidus töö (*on-line, на линии*) protsessi tajuritega. Osa andmeid võib juhtimissüsteemi sisestada ka käsitsi, terminaalide kaudu (*in-line, в линии*). Väga aeglaste protsesside puhul on mõeldav ka vallastöö (*off-line, автономная*), kus kõik lähteandmed sisestatakse käsitsi.

— Juhtimisarvutuste reaalarajarežiim, mis nõuab programmide initsieerimist, katkestamist, muutmist vastavalt protsessi sündmuste ja astronoomilise aja kulgemisele. Reaalarajarežiim tingib juhtimisarvutite peamised eripärad: prioriteetide skaalaga katkestussüsteemi, protsessi juhtimise programmide omapärase struktuuri ja reaalaraja operatsioonisüsteemi.

— Vähe muutuvad, kiired arvutused, loogiliste tehete ülekaal, spetsialiseeritud väikesed, tihti püsivaras realiseeritud operatsioonisüsteemid.

— Eriline, interaktiivne vahakord operaatoriga, kelle abistamiseks süsteem ongi mõeldud.

— Juhtimisarvutite lühike sõna, väike mälu, nõrk operatsioonisüsteem, mis nõuab spetsiaalseid võtteid algoritmeerimisel ja programmeerimisel.

— Spetsiaalse liiteseadmestiku olemasolu signaalide sisestamiseks, väljastamiseks, konverteerimiseks jne.

Niisuguse muudetava struktuuri ja konfiguratsiooniga riistvara on võimeline täitma kalkulaatorarvutite funktsioonide kõrval järgmisi olulisi automaatjuhtimissüsteemi operatsioone:

— Andmehõive: andurite skaneerimine, signaalide piiride kontroll, lineariseerimine, normaliseerimine, filtreerimine, ühikute teisendamine, kodeerimine, salvestamine. Töösagedus: msec, sec, min, tund, ööpäev.

— Raalreguleerimine: protsessi parameetrite numbriline stabiliseerimine tavalise analoogreguleerimise asemel. See võimaldab rakendada mitmedimensioonilisi, muutuva struktuuriga regulaatoreid; eriti tähtis on see keerukate objektide puhul. Töösagedus: msec, sec, min.

— Superviisorjuhtimine: protsessi optimaalse režiimi arvutamine ja realiseerimine kas automaatselt protsessijuhtimissüsteemi väljundseadmete kaudu või operaatori kaudu, kes peab protsessiga sidet puldi vahendusel. Töösagedus: min, tund, ööpäev.

— Dialog: operaatori teenindamine puldi, näit. kuvari kaudu, ka nn. *in-line*-andmehõive. Töösagedus: min, tund.



— Andmeedastus: formaatide ja koodide teisendamine, liinide ja seadmete kontroll, ajastamine, andmete õigsuse kontroll, ülekande juhtimine. Töösagedus: sec, msec.

Autor tänab Raul Tavastit väärtusliku konsultatsiooni eest artikli koostamisel.

#### KIRJANDUS

1. Weissberger A. J., *Comput. Design*, Nov., 77—82 (1974).
2. Schmid H., *Comput. Design*, Oct., 87—93 (1974).
3. Sell J. V., *Comput. Design*, Jan., 77—84 (1975).
4. Parrish E. A., Chang Lee Y., *IEEE Trans.*, IECI-21, No. 1, 38—41 (1974).
5. Rudd W. G., *Instrum. and Control Systems*, Apr., 63—66 (1974).
6. Microcomputer on Four Chips Needs no External Circuits, *Electronics*, Dec., 33—35 (1974).

Eesti NSV Teaduste Akadeemia  
Küberneetika Instituut

Toimetusse saabunud  
16. IV 1975

#### Б. ТАММ

#### ПЕРСПЕКТИВЫ ИСПОЛЬЗОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ С ПЕРЕМЕННОЙ СТРУКТУРОЙ

Рассматриваются вопросы использования ЭВМ как больших калькуляторов и как средств управления процессами в реальном масштабе времени. Освещается модульность, переменность структуры, новые режимы работы, макро- и микропрограммирование с точки зрения их использования в вычислительной технике четвертого поколения.

#### В. ТАММ

#### ON THE PROSPECTS OF USING COMPUTER HARDWARE WITH VARYING STRUCTURE

The problems of utilizing computers as large calculators as well as tools for real-time process control are discussed. The paper also deals with some specific phenomena of the fourth generation hardware such as modularity, structural variability, on-line, off-line, in-line realizations, macro- and microprogramming.