

Inseneri ja tehnoloogia võidujooksust nanomeeterdistsil

ehk tehnoloogia valitsemisest ja usaldamisest

Akadeemilise ettekande põhjal

Eesti Teaduste Akadeemias

10. nov. 2010

USA president Eisenhower hoiatas oma lahkumiskõnes presidenditoolilt, et teadlased ei tohiks muutuda riikliku finantseerimise vangideks, kus lepingud hakkavad asendama intellektuaalset uudishimu.

Kõne esialgses tekstis figureerisid kaks sõna *“engineer”* ja *“engineering”* paralleelselt sõnadega *“scientist”* ja *“science”*, aga lõplikust tekstist olid e-ga algavad sõnad välja jäänud. Möödus pool sajandit ja tehnoloogia tegi hüppe, mille tähendust on täna veel raske mõista. Kuid e-täht ilmus tagasi presidendi teksti. Novembris 2009 kuulutas Obama: *“Scientists and engineers ought to stand side by side with athletes and entertainers as role models”*. See, et sõna *“insener”* oli võrdväärna ilmunud sõna *“teadlane”* kõrvale, on märk, et avalik arusaam tehnoloogiliste muutuste olemusest on samuti muutunud.

Kes keda?

Inimese ja tehnoloogia vahel toimub omapärane võidujooks: kes keda? Uusi vidinaid tuleb peale kiirusel, kus keegi ei suuda enam järel püsida ja kohaneda. Tehnoloogia on inimesest möödunud ja tundub, et *“kätte saada”* teda on lootusetu. Vidinad valitsevad ja inimesest on saanud vidinate järel kõndiv ori. Aga siis teevad insenerid vahespurdi ja näitavad, et nad on võimelised veelgi paremaid vidinaid tegema ehk siis võidujooksus peale jääma.

Vidinate massiivne pealetung tuleb sellest, et üha rohkem tööd on inseneri eest hakanud ära tegema masinad, ehk siis tööriistad, mis insener ise on enda jaoks loonud. Kiirendus tuleb positiivsest tagasisidest. Esimese arvuti projekteeris ja valmistas inimene. Mõne aja pärast võttis ta endale appi selles töös omaenda ehitatud arvuti. Ajapikku muutusid arvutid võimsamaks – kaks pead on ikka kaks pead. Täna projekteerivad ja ehitavad arvutid juba iseseisvalt oma järeltulijaid ja mida võimsamaks arvutid saavad, seda võimekamaid järeltulijaid nad endale loovad. Niiviisi ongi toimunud viimase poolsajandi jooksul tehismaailma evolutsioon: tööriistade omadusi on pidevalt parandatud ja neid parandusi on tööriistad oma järglastele edasi pärandanud. Positiivse tagasisidega haaratud masinprojekteerimine ja automatiseeritud tööstus on meid kandnud märkamatuks intelligentseks muutunud tehismaailma.

Tehisintelligents

Achilleus teadupärast kilpkonna kätte ei saanud. Ta andis suuremeelselt konkurendile pool maad edu ja selle ohvriks ta ka sai. Iga kord, kui Achilleus oli jõudnud punkti, kus hetk tagasi oli asunud kilpkonn, oli viimane jõudnud juba järgmisse punkti ja nii edasi lõpmatuseni, ehk siis kilpkonn oli alati Achilleusest pisut maad ees.

Sama dilemma juures on täna tehnoloogia, mis pürib inimesest targema tehismaailma suunas. Seni ollakse veel veendumusel, et intelligentseks masin ei saa kunagi. Tehisintelligents tähendab, et arvuti oskaks seda, mida seni on osanud üksnes inimene. Seega, et sünniks tehisintelligents, tuleks programmeerida arvutit lahendama ülesannet, mida seni on osanud lahendada ainult inimene. Kui aga ülesanne on arvuti poolt lahendatud, siis sellest hetkest peale ei kuulu ju see ülesanne enam

tehisintelligentsi valdkonda. Niiviisi arutledes, pole tehisintelligents vist tõepoolest mitte kunagi saavutatav.

Samas oskab arvuti juba praegu mängida malet paremini kui inimene. Rääkimata sellest, et arvutid suudavad projekteerida ise oma järglasi. Küsimus on definitsioonis. Oleks vale mõista tehisintelligentsi inimintelligentsi koopiana, vähemalt nii kaua, kuni arvutid töötavad teisel põhimõttel kui inimese aju. Õpetlik on kirjutusmasina leiutamise ajalugu. Algul püüti konstrueerida masinat, mis produtseeriks kirja inimese käekirja kujul. See osutus aga halvaks ideeks. Otstarbekamaks kujunes trükitähtedel põhinev mehaaniline kirjutusmasin. Nii pole ka mõtet taga ajada inimhõimust, mille olemust me üldse halvasti tunneme. Targem oleks hoopis seda ületada. Nii nagu Achilleuski oleks lihtsalt pidanud astuma kilpkonnast üle.

Tehnoloogiline singulaarsus

Tehnika areng on olnud hämmastav. Veel 15 aastat tagasi ei osanud keegi ette kujutada seda, mida tähendab Internet. Aga see on alles algus. Hoogu on võtmas juba *asjade internet*, mis ühendab tulevikus inimesi ja asju. IT-maailmas levib *pilvandmetöötlus* ehk *pilveraalindus*. See aga tähendab, et juba lähitulevikus hakkavad robotid koosnema kahest tükist: maapealne (roboti keha) ja "pilvepealne" tükk (roboti aju). Roboti sattudes tundmatusse keskkonda, kontakteerub ta ümbritsevate asjadega interneti kaudu, saadab pildi oma ümbrusest ja asjadest seal „taevasse“ (pilveserverite võrku) ja saab tagasi mis iganes infot ta oma ümbruse kohta ka ei vaja. Inimese peaks kõigepealt koolitusele saatma, et ta tundmatus keskkonnas hakkama saaks. Robot aga laeb alla programmi kogu vajaliku tarkusega ja kooli minna pole tal vaja. Homne päev on *pilverobotika* käes ja see on hoopis võimsam värk kui internet. Internetiga võid üksnes suhelda, pilverobotile aga ütled: „Havi käsul, minu soovil!“

Oli aeg, kus üks arvuti teenis sadat inimest. Siis tulid personaalarvutid – igale inimesele tema enda oma. Täna teenivad sada arvutit ühteainust inimest. Mobiiltelefonid, fotoaparaadid, olmetehnika, liftid, rahaautomaadid, targad majad peidavad kõik endas arvuteid. Ühes heas autos abistab juhti sada mikroprotsessorit. Oleme muutunud arvutisõitlasteks, aga märkame oma sõltuvust vaid siis, kui tehnika tõrgub. Ja siit algavadki ohud ja riskid, mis on seotud tehnikaga ja selle edasise arenguga.

Ennustada, mis saab homme niisuguse arengutempo juures nagu täna, on võimatu. Futuristid filosoferivad *singulaarsusest* – see on punkt, kus enam ei tea, millised on laius- ja pikkuskraadid ning mis toimub ümberringi. See on punkt, kus kiirendus võib ulatuda lõpmatusse ja kus enam ei saa aru, kuhu sa kiirened. Matemaatiliseks singulaarsuseks nimetatakse punkti, kus matemaatiline objekt ei ole enam määratud. Nüüd oleme liginemas tehnoloogilisele singulaarsusele, kus tehismaailma arengu ennustamine on võimatu. Ja süüdi on selles elektroonika.

Usalda aga kontrolli

Elektroonikat nimetatakse austavalt innovatsiooni mootoriks. Näiteks auto puhul kuulub 90% innovatsioonist elektroonikale. Auto hinnast moodustab 40% elektroonika ning selle juurde kuuluv tarkvara. Sellest omakorda 60% kulub tarkvara arenduseks ja 40% moodustab riistvara hinna. Ühe koodirea kirjutamine tarkvara loomise protsessis maksab USA sardsüsteemide tööstuse hinnangul 15-40 dollarit. Vastutusrikkamatel aladel hind on kõrgem, näiteks kaitsetööstuses 100 ja kosmoselaevade puhul koguni 1000 dollarit. Tarkvara maht kasvab autotööstuses kolme aastaga neli korda, kusjuures iga kolme kuu tagant tarkvara vahetub – muutub paremaks ja kõrvaldatakse vead.

Elektroonika on keeruline ja nii selle loomisel kui kasutamisel võib esineda vigu ja tõrkeid. Tarkvara puhul tekivad vead selle loomise käigus. On hinnatud, et iga seitsmenda koodirea kirjutamisel teeb tarkvarainsener vea. Riistvara puhul võivad vead tekkida nii projekteerimisel kui ka rikete või häirete tõttu kasutamisel. Elektroonika riist- ja tarkvara testimiseks ja vigade otsimiseks kulub auto valmistamisel 20-25% selle hinnast.

Esimest "ametlikku" arvutiviga mäletatakse aastast 1945, kui üks rumal koiliblikas sattus Harvardi ülikoolis arvuti Mark II relee kontaktidele ja lühistas need. Logiraamatusse kirjutati sellel päeval: "Hey, we found a bug that was a real bug!" Nii olevatki sündinud erialane termin "debug", mis tarkvarainseneride slängis tähendab vigade kõrvaldamist programmis. Koiliblikas oli naljalugu. Tõsised probleemid tulid siis, kui arvutid hakkasid tõsiseid ülesandeid täitma. Näiteks 1995. a. Lahesõjas eksis arvuti ümardamisvea tõttu ning lasi Iraagi raketil tabada USA sõjaväebaasi. Samal aastal jäi Intel Pentium projekteerimisvea tõttu hätta jagamistehtega ja kogu protsessorite seeria tuli tagasi osta. Arvutirike Toyota pidurisüsteemis möödunud aastal, mis seisnes hetkelises süsteemi tundetuses hüdraulika ja elektroonika ümberlülitusel, põhjustas firmale 2 miljardi dollarilise majanduskahju ja 4,5 miljoni auto tagasiostu. Firma aktsiad kukkusid päevaga 22%. Viie aastaga on USAs kahjum arvutivigadest kasvanud viis (!) korda, ulatudes 60 miljardi dollarini aastas.

Inseneritöö on vastutusrikas. Talle öeldakse aitähk tema loomingu eest, vahel isegi unustatakse tänada. Harjume ju kiiresti sellega mis on. Aga insenerile ei andestata vigu. Ajalugu on näidanud, et 99% inseneride tööst on olnud see, mille eest ühiskond tänab. Ülejäänud 1% aga tähendab inseneri süüd inimkonna ees. Protsendid ei loe enam midagi, kui mängu tulevad elud – aatomipommid, tuumajaamad...

Seetõttu on inseneri kategooriliseks imperatiiviks – tema töö produkti usaldatavuse (*dependability*) tagamine. Usalda, aga kontrolli. Nii ongi kujunenud arvutisüsteemide loomisel ootamatu vahekord – tervelt 70% projekteerimiskuludest läheb verifitseerimisele, vigade otsimisele ja testimisele.

Moore'i seadus

Rikete põhiallikas elektroonikaskeemides on keerukus ehk siis elementide (transistoride) väga suur arv, mis ulatub miljarditesse. Mikroelektroonika on ime. Sajad miljonid transistorid peavad töötama täiuslikult. Aga looduse vastu ei saa. Mida rohkem on süsteemis elemente ja mida väiksemad on transistoride mõõtmed, seda suurem on tõenäosus, et mõni transistoridest on siiski vigane. Just niisuguse tõenäosuse kasvamise suunas elektroonika praegu areneb.

1965. aastal palus ajakiri Electronics Magazine firma Fairchild'i ühel asutajal Gordon Moore'il ennustada integraalskeemide ehk kiipide arengut järgmiseks 10 aastaks ette. Hinnates senist arengut oma firmas viimase nelja aasta jooksul, kus oli õnnestunud teha transistore järjest väiksemaks tänu fotolitograafia täpsuse tõstmisele, avastas Moore, et transistoride arv oli kahekordistunud iga aasta jooksul. Nii pakkuski ta välja 1000-kordse transistoride kasvu järgmiseks dekaadiks, mis pidi tähendama, et 70ndate keskpaigaks saab ühes kiibis olema 64000 transistori.

Kiipide areng jätkus siiski pisut aeglasemalt ja 10 aastat hiljem Moore korrigeeris ennustust, pakkudes kahekordistumise perioodiks kaks aastat. See osutus nüüd aga liiga pessimistlikuks ja tegelik areng on tänaseni toimunud nii, et transistoride arv kasvab kahekordseks poolteise kuni kahe aasta jooksul. Moore'i seadust on käsitletud kui aksioomi, tegelikult on see vaid rusikareegel ja nii on seda väitnud ka Moore ise. Samas on hämmastav, et see reegel on kehtinud juba 45 aastat, mistõttu Moore'i peetakse tõeliseks prohvetiks. Tänavu Inteli välja lastud 8 tuumalise protsessori kiip *Intel® Xeon® processor 7500* sisaldab 2,3 miljardit transistori. Transistoride tihedus on veel vaid paar kolm korda väiksem kui neuronite tihedus ajus $5 \cdot 10^7 \text{ cm}^{-2}$.

Moore'i seadus on muutunud juhtniidiks uurimis- ja arendustöö planeerimisel elektroonikatööstuses. Paljude elektroonikaga seotud seadmete areng on hästi kooskõlas Moore'i seadusega, näiteks koos transistoride arvu kasvuga kiibis kahanevad transistoride suurus ja hind, kasvavad töökiirus ja mälumaht, isegi pikselite arv fotoaparaatides. Seejuures Moore formuleerides oma ennustust, pidas silmas üksnes transistoride tiheduse kasvu kiibis.

Futuristidel, kes käsitlevad Moore'i seadust laiemas tähenduses, on hoiatav vaade seaduse toimele. Nende arvates tähendab tehnoloogia eksponentsiaalne areng ühiskonna viimist tehnoloogilisse singulaarsusesse ehk siis punkti, kus tehnoloogiline progress muutub silmapilkseks. Niisugusel juhul muutub ühiskonna areng aga ettearvamatuks.

Moore'i teine seadus

Tehismaailma rikkalikkuse, tarkuse ja keerukuse kasv on olnud senise katkematu innovatsiooni tulemus, kus võlgname tänu Moore'i seaduse püsivusele. Kui transistori mõõtmete lähenemine füüsikalisele piirile põhjustaks seda, et seadus lakkaks toimimast, katkeks ka kogu innovatsioon ja seiskuks areng.

Kahjuks on aga Moore'i seadusega tõepoolest kaasnenud mitmed negatiivsed faktorid. Koos transistoride arvuga kasvab ka kiibi võimsustarve ja energiatihedus, millega on otseselt seotud füüsikaliste defektide tõenäosuse kasv. Kiipide töösagedus kasvas samuti Moore'i seaduse järgi vastavalt sellele, kuidas kahanesis transistoride mõõtmed. Kiireimate protsessorite töösagedus ulatub praegu gigahertsidesse. Kuid energiatiheduse suurenemine ja kiipide ülekuumenemise oht on sundinud töösagedust piirama. Tulemuseks oli paradoks: transistoride arv küll kasvab, aga kiibi töökiirus väheneb. Just seda me aga elektroonikalt kõige vähem ootame. Transistoride arv kiibis ei huvitagi tegelikult tarbijat. Teda huvitab töökiirus. Kogu innovatsioon, sardsüsteemide kvaliteet, fotokaamerate säritusaeg, arvutimängude atraktiivsus – kõik sõltub eeskätt kiipide töökiiruse kasvust. Ja nüüd äkki – lagi on käes.

Ometi leidsid insenerid väljapääsu. Seni tähendas transistoride töösagedus arvuti töökiirust. Aga see ei pea üldsegi nii olema. Lahendus leiti selles, et kiibi töösagedus lasti küll alla, aga kiipi pandi ühe protsessori asemel mitu lihtsamat, milliste vahel jagati töö paralleelselt ära. Kokkuvõttes, töökiiruse langusele vaatamata arvuti jõudlus hoopis kasvas. Moore'i seaduse "päästis" *paralleelsuse idee*.

Arvutite töökiirust ei iseloomustatagi enam taktsagedusega ehk transistoride töökiirusega, vaid hoopis operatsioonide arvuga sekundis: 1 *flops* tähendab ühte operatsiooni sekundis. Hiljuti püstitati uus maailmarekord arvutite jõudluses: hiina superarvuti *Tianhe 1A* saavutas töökiiruse 2,5 *petaflopsi* ehk $2,5 \cdot 10^{15}$ operatsiooni sekundis.

Paralleelsuse idee tõi uue paradigma tehnoloogiasse – üksikute keeruliste protsessorite asemel hakati projekteerima *mitmiktuumprotsessoreid*, mis koosnevad paljudest lihtsamatest protsessoritest. Seejuures tuumade arvu 2-kordistamine tähendab ka protsessori jõudluse 2-kordistumist. Transistoride arvu 2-kordistamine ühesainsas protsessoris seda ei teeks, küll aga kulutaks rohkem energiat. Nii viisi hakkas toimima ka uus Moore² seadus, mis tähendab tuumade arvu kahekordistumist kahe aasta jooksul.

Uueks väljakutseks elektroonikas on 3D- ehk kihiliste kiipide loomine, kus eri kihtidel realiseeruvad eri funktsioonid: ühel kihil on protsessorid, teisel mälu, kolmandal toimub infovahetus. Klassikalisele horisontaalsele integratsioonile lisandub vertikaalne, mis veelgi avardab tehnoloogilisi võimalusi.

Vahespurtidest

Moore'i seadusele vastandub paradoksaalselt üks teine – Wirth'i seadus, mille üheaegselt Moore'i seadusega formuleeris Niklaus Wirth ja mis väidab, et tarkvara aeglustub kaks korda iga 18 kuuga. Sellega tühistub ühtlasi kogu efekt, mis on saavutatud riistvaras. Näiteks Microsoft Office'i versioon 2007 täidab sama ülesannet mis versioon 2000 kaks korda aeglasemalt, hoolimata elektroonika saavutustest selles ajavahemikus. Seletatav on see sellega, et tarkvara muutub järjest keerukamaks ja universaalsemaks. Algseid lihtsaid funktsioone täiendatakse üha keerukamate võimalustega, mis aga muudavad keerukamaks kogu realisatsiooni ja mille tulemusena algfunktsiooni täitmine aeglustub.

Omaette probleem on tarkvara projekteerimine paralleelsete arvutusprotsesside realiseerimiseks mitmiktuumprotsessoritel, mille lahendamisele on hakatud alles hiljuti mõtlema ja see on uueks väljakutseks tarkvarainseneridele.

Sarnaselt tarkvara projekteerijatele pole tehnoloogia vahespurdile suutnud vastata ka riistvara disainerid ja diagnostikainsenerid. Transistoride organiseerimist kiibis mingi funktsiooni täitmiseks nimetatakse disainiks. Protsessori ehitab transistoridest disainer, kasutades selleks eritarkvara – raalprojekteerimise tööriistu. Paraku on selle tarkvara jõudlus eksponentsiaalses sõltuvuses transistoride arvust kiibis. Mida rohkem transistore, seda aeglasemalt kulgeb projekteerimine või, et kiirendada projekteerimist, jääb efekt transistoride arvu suurenemisest kasutamata. Seda nähtust nimetatakse disaini *produktiivsuse kriisiks*.

Tõsi, vahepeal on siingi toimunud paradigma vahetusi. Näiteks, on levima hakanud nn. “korduvtarvituse” (*reuse*) paradigma, mis tähendab, et kiipe projekteeritakse mitte enam transistoride tasemel, vaid pannakse kokku ostetavatest koostisosadest, mis on projekteeritud eelnevalt kusagil mujal. Paradigmade muutused on mõnevõrra kahandanud tehnoloogia edumaad riistvaradisainerite produktiivsuse ees. Aga paralleelsuse probleem ja uus Moore² seadus on tõsine väljakutse just tarkvara projekteerijatele.

Tehnoloogia areneb kiiremini kui tehnoloogia kasutamise kultuur seda vastu võtta suudab. Selles mõttes on insenerid kaotamas võidujooksu tehnoloogiaga. Mõneti on see heagi – pidurdab meie tormamist singulaarsuse keerisesse. Teiselt poolt, jätkub üha uute väljakutsete sünd tehnikateadustes ning tagaajamine jätkub.

Kriminaalsed transistorid

Nii nagu ühiskonnas oleks parem elada, kui seal poleks kuritegevust, nii võiksime end kergemini ja täielikumalt usaldada tehismaailma meeleva, kui seal ei ringleks kuritegelikke transistore.

Leida üles kriminaalne transistor miljardite hulgast on sama keeruline, kui leida üles piletitähtsuse külastaja keset rahvahulka lauluväljakul. Et otsida kiibis vigaseid transistore tuleks läbi kiibi lasta signaale ja kontrollida, kas need õigesti päralt jõuavad. Et leida katkine transistor, peaks kontrolliv signaal läbima rikkis transistori, et selle mõjul muutuda vigaseks, vigasena kiibist välja jõudma ning sel moel viga ilmutama. Kolm takistust hakkavad siin aga head ideed põrmu viima. Esiteks, kuna transistore on palju (miljardeid), siis kuluks väga palju aega, kuni niisuguste signaalidega kõik transistorid läbi sondeerida jõuaks. Teiseks, on väga raske niisugusi eksperimente signaalidega üldse organiseerida. See oleks sama, kui lasta ühest lauluväljaku äärest rahvahulka sõnum, milles palutakse leida üles kodanik perekonnanimega Kask, uurida ega too pole liiga palju õlut joonud ja saata info politseinikule Kuusk, kes seisab kusagil lauluväljaku teises ääres. Kolmandaks, selle sõnumi edastamisel võib rahvas hakata vigurdama, sest mitte kõik kodanikud ei pruugi asja võtta tõsiselt. Transistoride puhul tähendaks see seda, et kui signaal oligi muutunud vigaseks katkises transistoris, siis on väga raske organiseerida signaali edasist kulgu nii, et viga ei sumbuks kusagil keset transistoride massi.

Need on põhjused, mis teevad raskeks rikete avastamise elektroonikaskeemides, kus on miljardeid transistore. Põhjuseks öeldakse olevat – skeemide keerukus. Aga mida tähendab keerukus?

Diagnostika paradoksid

Kumb on suurem, kas miljard või lõpmatus? Uurime kõigepealt, kuidas testida analoogvõimendit? Tuleks kontrollida, kuidas see võimendab signaale. Võimalike analoogsignaali hulk võimendi sisendis on aga teoreetiliselt **lõpmatu**. Ometigi piisab praktikas sellestki, kui kontrollida, kas seade võimendab ühtviisi näiteks nõrka, keskmist ja tugevat signaali. Teiste sõnadega, piisab vaid kolmest testist veendumaks, et võimendi töötab.

Keerulisem lugu on arvuti summaatoriga. Oletame, et summaator liidab 32-bitiseid arve. Niisuguse summaatori sisendis on seega 64 juheta, millele võib rakendada $2^{64} \approx 10^{19}$ erinevat signaalikombinatsiooni ehk arvupaari. 10^{19} on palju **väiksem kui lõpmatus**. Seega peaks ka digitaalse summaatori testide valik olema lihtsam kui analoogvõimendi puhul. Aga nii see ei ole. Summaatorit ei saa kindlasti testida ainult

kolme arvupaariga. Teiselt poolt, kõigi 10¹⁹ arvupaariga testimine oleks samuti võimatu, sest selleks kulukuks aastaid. Lahendus on kusagil vahepeal. Aga raskustest vahepealse lahendi leidmisel oli juttu eelmises lõigus.

Paradoks on selles, et ehkki analoogelektronika funktsionaalseks aluseks on *lõpmatu arvude maailm*, siis just digitaalsüsteemide ilmumine, mis toimivad “lihtsamad” *lõplikud arvude maailmas*, tõi teaduse keerukuse mõiste. Keerukust ei määra üksnes hulga suurus, näiteks transistoride arv kiibis, vaid ka hulga elementide omavaheliste seoste iseloom.

Ühes laboris korraldati katse, kus ühele elektronskeemile hakkasid testi koostama insener ja arvuti. Insener sai hakkama veerand tunniga, arvuti töötas kaks ööpäeva ning seejärel loobus. Kuidas siis nii, kogu lootus oli ju pandud sellele, et automatiseerida testide sünteesi – ehk siis luua tarkvara, mis automatiseeriks töökorralduse laulupeol, nii et kordnik Kuusk saaks teada, kas kodanik Kask on liiga palju õlut joonud või ei ole. Paraku jäi nüüd arvuti ülesande keerukuse tõttu jänni.

Arvuti nägi testitavas skeemis miljardit ühesugust transistori. Insener aga tundis ära skeemis tuttavaid funktsionaalseid transistorikooslusi ning oskas testsignaali korraldamisel loovalt ära kasutada skeemi omapära. Arvutit ei olnud õpetatud skeemi omapärasid arvestama. Insener osutus masinast intelligentsemaks.

Nüüd kordus kirjutusmasina loomise viga. Hakati ehitama ekspertsüsteeme, õpetama masinatele samu tarkusi, mida insenerid aastate jooksul olid kogunud, soovides masinaid intelligentsemaks teha. Katse ebaõnnestus. Esiteks kasvas vajaliku eksperttarkuse hulk kiiremini, kui seda suudeti manustada masinale. Seega pidi ikkagi targa masina kõrvale istuma veel targem insener. Nüüd aga kadus peagi inseneridel huvi mängida arvuti “nõuandja” rolli, nad oleksid pigem tahtnud vajutada mingile nupule ja ise kohvi jooma minna...

Edasine areng järgis seda, mis toimub ka ühiskonnas. Kui miski “üle käte” läheb, luuakse seadus, et inimesi vaos hoida. Ka elektroonikasüsteemide valdkonnas hakati looma reegleid ja seadusi, mis kitsendasid disaini vabadust, aga lihtsustasid kontrollimist ja kriminaalsete transistoride üles leidmist.

68 tuhat kanapoega

Üks esimesi elektrilisi seadmeid, mida ma noorpõlves tundma õppisin, oli tester. Testri küljes oli kaks juhet, mille abil sai mõõta pinget, voolu ja takistust. Elektroonika muutus aga keerukamaks ja lihtsad käsitestrid asendusid automaatsete testritega ehk testsüsteemidega. Aga tester sai suhelda kiibiga üksnes väliskontaktide abil, samas kui transistorid ise pugesis üksteise taha kiipide sügavusse. See polnud veel kõige hullem. Testrite keerukus kasvas kiiresti, sest nad pidid testima üha keerukamaid kiipe. Nad muutusid kalliks, makstes kohati miljoneid dollareid. Seetõttu ei saanud neid kuigi sageli moodsamate vastu välja vahetada. Samal ajal aga toimus Moore’i seadus ja kiipide töökiirus üha kasvasid. Testri enda töökiirus hakkas maha jääma testitavate kiipide töökiirusest. Vana kohmakas ja aeglane tester ei suutnud enam järgida kiirelt kepsutavaid transistore ja kriminaalsed elemendid hakkasid habemikku testrit osavalt ära petma. Uusi kiipe oli tarvis testida nende enda töökiirusel, milleks vanemal tehnoloogial põhinevad testrid polnud põhimõtteliselt enam võimelised.

80ndatel töötati firmas *Motorola* välja mikroprotsessorit *Motorola 68000*, milles oli 68000 transistori. Testimisosakonnas olid kibekiired tööpäevad, kuna testprogrammide kirjutamine mikroprotsessorile hakkas käima üle jõu. Osakonnajuhataja võttis stressi maandamiseks vaba nädalavahetuse ja sõitis külla farmerist vennale Texas. Pärast hommikusööki pidi farmer minema söötma kanu. “Palju sul neid tibusid siis on ka?” küsis vend. – “68000,” vastas farmer. “Ohhoo,” mõtles vend ja meenutas, et ka temal on täpselt sama suur hulk hoolealuseid – transistore. “Kuidas Sa neid siis söödad?” tundis ta huvi ja läks vennaga kaasa asja kaema. Ja mis ta siis nägi – vend puistas laiali terad ja kanad hakkasid ise (!)

nohkima... Nüüd tabas väik elektroonikust farmeri venda: "Tõepoolest, miks mitte panna ka transistorid ennast ise (!) testima."

Nii sündiski digitaalsüsteemide *isetestimise* idee ja *Motorola* 6800st sai esimene elektroonikaseade, kus seda ideed hakati praktiliselt realiseerima. Algul tundus see mõte rohkem fantastikana, aga täna on digitaalsüsteemide isetestimine argipäev. Süsteem jaotatakse tinglikult osadeks – mooduliteks. Moodulite vahele pikitakse suhteliselt lihtsaid juhuslike arvude generaatoreid. Generaatorite signaalidega ergutatakse mooduleid, nende reaktsioonid aga, mis kannavad infot avastatud rikestest, suunatakse spetsiaalsetesse analüsaatoritesse, mis informeerivad avastatud vigadest välismaailma – testinseneri, hooldeinseneri või tarbijat ennast.

Arvutitehnika instituudist

Ülal kirjeldatud probleemid ja väljakutsed on see, mis annab õppetöö tegemise kõrval "leiba" teaduritele ja doktorantidele **Margus Kruusi** juhitud arvutitehnika instituudis. Kuna tehnoloogia liigub kiiresti eest, siis tänane päev pole kunagi sama mis eilne päev. Probleemid vahelduvad ja muutuvad, aga suuresti ka korduvad arenguspiraali eri keerudel. Nii kaua, kuni tehnoloogia pole valmis, ei kao ka laualt disainerite ja diagnostikute "leib".

Prof. **Peeter Ellervee** juhtimisel töötatakse viljakalt keerukate digitaalsüsteemide modelleerimise ja sünteesi valdkonnas. Välja on töötatud tarkvara süsteemide automaatseks sünteesiks kõrgtasandil, mis tähendab seda, et loodava süsteemi tööpõhimõte kirjeldatakse spetsiaalses projekteerimiskeeles ja tarkvara sünteesib sellest kirjeldusest automaatselt seadme struktuuri. Meetodit on kasutatud riistvaralise akseleraatori ehitamisel, mis võimaldas saavutada 200-kordse(!) kiirusevõidu võrreldes tuntud tarkvaraliste lahendustega.

Allakirjutanu juhtimisel on kulgenud uurimistöö digitaalsüsteemide diagnostika valdkonnas. Paljude aastate jooksul on põhitähelepanu olnud suunatud uut tüüpi matemaatilise mudeli välja töötamisele, mis oleks efektiivsem senistest ja rajaks aluse rakendustele süsteemide diagnostika vallas. Töögrupp on olnud teerajajaks spetsiaalse graafiteooria, nn. otsustusdiagrammide mudeli loomisel, olles esimene, kes töötas välja uut tüüpi struktuursed binaargraafid loogikatasandi jaoks ning üldistasid tulemused analoogilise kõrgtaseme mudeli näol. Teooria on saanud aluseks paljude efektiivsete algoritmide, meetodite ja tarkvaratööriistade välja töötamisel. Nii näiteks, kasutades seda graafmudelit, programmeeris **Sergei Devadze** oma doktoritööna rikete simulaatori, mis ületab kiiruselt maailma tipptööstuses kasutatavaid analoogseid simulaatoreid.

Süsteemide verifitseerimise valdkonnas töötab edukalt vanemteadur **Jaan Raik**, kes oma doktoritööna programmeeris unikaalse hierarhilise testigeneraatori **DECIDER**, mille taolisi maailma kommertsturul saada pole võimalik. Praegu koordineerib ta vastutusrikast 7-nda Raamprogrammi europrojekti DIAMOND, kusjuures tema käe all töötavad partneritena sellised tippfirmad nagu IBM ja Ericsson. Jaan Raigi toredaks tulemuseks on ka diagnostikatarckvara **Turbo-Tester**, mis on tänaseks litsenseeritud rohkem kui 100 laboris 45 erinevas riigis. Tema käe all kirjutas oma doktoritöö **Maksim Jenihhin**, kes töötas välja senistest efektiivsemad verifitseerimismeetodid, mis on realiseeritud tarkvarasüsteemina **APRICOT**. Kõik kirjeldatud süsteemid põhinevad instituudis väljatöötatud matemaatikal ehk siis loogika- ja kõrgtaseme graafmudelitel.

Vanemteadur **Artur Jutman** koos oma uurimisgrupiga on teinud suuri edusamme testide programmeerimise automatiseerimisel elektroonikaplaatide testimiseks. Täna käib see töö kogu maailmas veel käsitsi. Instituudi teaduskompetentsi selles valdkonnas on regulaarselt kasutanud Eesti firmad Ericsson, Elcoteq, Testonica Lab ning tippfirma Göpel Electronic Saksamaal. Rootsist Eestisse doktorina naasnud vanemteadur **Gert Jervan** on saanud märkimisväärseid tulemusi digitaalsüsteemide isetestimise ja veakindluse valdkonnas. **Eero Ivask** aga on suutnud tõsta mitmete diagnostikaülesannete lahendamise efektiivsust, kasutades algoritmide paralleliseerimise võimalust. See on äärmiselt oluline

teema, pidades silmas uusi võimalusi, mida annavad mitmiktuumprotsessorite ilmumine tarkvarainseneride käsutusse.

Quo vadis, teadus?

Maailm ja selle olemus on muutumas. Kõik uus meid ümbritsevas tehismaailmas, kogu innovatsioon sünnib eeskätt tänu arvutite võimsusele, mis aga pidevalt kasvab. Robotid võtavad inimeselt töö. Kommunikatsioon asendab liikumise, kaob vajadus autode järele. Inimese ja masina vaheline piir hägustub, infotehnoloogia ühendatakse implantaatide abil vahetult inimese aju ja närvisüsteemiga. Inimmõistus nihkub üha rohkem väljaspoole inimkeha. Professionaalne teave muutub aegunuks selle hankimise hetkel. Teaduskraadide tähendus hakkab seisnema üksnes eneseaktualiseerimises ning kaotab peaaegu täielikult kommertsiaalse tähenduse... Niiviisi ennustatakse tulevikku.

Singularity Institute of Artificial Intelligence tegeleb singulaarsuse uurimisega ja on seadnud endale kaks ülesannet: fookuseeruda üldisele tehisintelligentsile ja fookuseeruda ohutusele. Viimane on eriti oluline. Aga kõik see pole uus: arvutiinseneride kogukonnas on üks hulk uurijaid, kes juba mõnda aega tegelevad süsteemide usaldatavuse (*dependability*) ja ohutuse probleemidega, kus eesmärgiks on seatud, et süsteemide projekteerimisel peab "*usalduse projekteerimine*" (*design for dependability*) käima käsikäes kogu projekteerimise endaga. Tuleviku robotite maailmas seisneb põhiline oht mitte robotite võimalikus mässus, vaid loodusressursside väärkasutuses robotite endi arengu nimel inimkonna arvel. Praegu aga ohustab inimene ennast ise, näiteks üritades energia jätkusuutlikkuse nime all kasutada põllumaad energiametsade kasvatamiseks nisu asemel. Kasvava ohu allikaks on tuumaelektrijaamad.

Kuhu liigub siis tehnoloogilise singulaarsuse rajal teadus? Tundub, et kolme fundamentaalse avastusega – *aatom, geen ja arvuti* – on põhilised mateeria, elu ja arvutamise seadused ära tunnetatud. Romantiline *teaduslike avastuste periood (age of discovery)* on asendumas pragmaatilisema *teaduse rakendamise perioodiga (age of mastery)*. Innovatsiooniliidriteks on saamas mitte tehniliste uuenduste loojad, vaid need, kes tegelevad rakenduste kontekstiga. Selline arengutendents on isenesest soodus Eesti teadusele ja tööstusele. Meil ei ole vaja üüratu kalleid uurimis- ja tootmiskeskondi, fundamentaalset tööstust, mis on suunatud tehniliste uuenduste loomisele – uut tüüpi tehnoloogiate väljatöötamisele. Vaja on üksnes seista kahe jalaga maas, tunnetada ühiskonna vajadusi ning avastada uusi olemasoleva tehnoloogia rakendusvõimalusi.

Meid ümbritsev tehiskeskond on täis arvuteid, kes inimest teenindavad. Teenindamisvõimalustel pole aga piire, nii nagu muinasjutuski: havi käsul, minu soovil. Havi rollis on sardsüsteemid ja nende loomine olekski tänapäeval loomupärane teaduse ja tehnoloogia väljund – lõpmatu innovatiivsuse allikas.

Avaraid kommertsperspektiive pakub järgmine Interneti põlvkond – tarkade asjade internet.

Innovatsiooni võtmeks on tarkvara, ehk õppinud ja haritud inimese aju. Hallollust meil Eestis jätkuks, kui vaid haridus õiget rada käiks. Arvutite valdkonnas on tark- ja riistvara piirid ähmastumas.

Rekonfigureeritavate FPGA-tehnoloogiate puhul on programmeeritavad mõlemad – nii tark- kui ka riistvara. Kuna Eestis ei ole kallist mikro- ja nanoelektronika tööstust, siis võtame kätte ja hoopis programmeerime just niisuguse riistvara nagu soovime.

Kuna tarkvarale taandatud töö nõuab üksnes ajuresursse, siis IT-alane kõrgharidus peaks olema Eesti tähtsaim prioriteet, kuna mõjutab mistahes teisi teaduse või elu valdkondi. Aga parimaks küberkaitseks ühel väikesel maal oleks IT-alane riiklik sõltumatus.

Raimund Ubar

Arvutitehnika instituudi professor