# von Neumann –arkkitehtuuri ennen ja nyt

Teemu Kerola

Historiallisesti tietokone on laskentakone, joka rakentuu laskennan ohjaamisesta, laskennan kontrolloinnista ja laskennan kohteena olevasta syöteaineistosta. Laskennan tulos on myös syytä saada talteen. Laskennan ohjaaminen tarkoittaa ohjelmaa eli algoritmista käskyjoukkoa, jonka mukaan oikea tulos lopulta saadaan. Syöteaineisto tarkoittaa laskennan kohteena olevien lukujen joukkoa, joille laskenta tällä kertaa halutaan suorittaa. Sekä ohjelman että syötteen tulee olla jonkinlaisessa muistissa laskennan aikana, jotta laskenta voidaan tehdä automaattisesti ilman ulkoista apua. Laskennan tulos talletetaan lopuksi johonkin muistiin. Se, että syötteet luetaan aluksi ulkoiselta laitteelta muistiin ja että tulos voidaan lopuksi tallettaa muistista ulkoiselle laitteelle, ei ole tärkeä itse ratkaisun laskennan kannalta.

Laskennan kontrolli tarkoittaa suorittimen sisäistä rakennetta, jonka mukaan laite toteuttaa annetut käskyt (ohjelman) annetulle syötteelle. Kontrolliin sisältyy tieto siitä, mitä juuri nyt tulisi tehdä ja miten se tehdään eli koneen laskentapiirien ohjaaminen tämänhetkisen toiminnon suorittamiseksi. Muistia oli alkuaan kolmen tyyppistä: ohjelmamuisti, syötemuisti ja tulosmuisti. Laskenta tapahtui erityisten rekistereiden avulla ja laskennan aikana tarvittiin välitulosmuistia.

Kontrollin toteuttamisen tärkeä edelläkävijä oli Joseph-Marie Jacquard, joka vuonna 1801 toteutti kutomakoneensa. Sen voidaan katsoa olevan kaikkien nykyisten sulautettujen järjestelmien esi-isä. Kutomakonetta ohjasivat reikäkortit, joiden avulla kone kutoi annettujen mallien mukaisia kankaita tai koristenauhoja. Automaattista toimintojen ohjausta tuona aikakautena oli monessa muussakin laitteessa, esimerkiksi soittorasioissa tai automaattipianoissa. Kuitenkin juuri Jacquardin kutomakone ja sen ohjauslogiikka on historiallisesti katsoen merkittävä, koska sitä sovellettiin ensin laskimien ja sitten tietokoneiden ohjaamiseen. Jacquardin järjestelmään sisältyi myös erillinen reikäkorttien lävistyslaitteisto, jolla uusia malleja oli helppo ohjelmoida reikäkorteille. Kutomakoneen yhteiskunnallinen merkitys oli myös samaa tasoa kuin tietotekniikan läpimurto pari sataa vuotta myöhemmin. Jo kymmenisen vuotta keksinnön jälkeen niitä oli tuotantokäytössä yli 18000 kappaletta ja niiden aiheuttama rakennemuutos sekä työttömyysongelma kutomateollisuudessa johtivat jopa mellakoihin. Kutomakonetta ohjaavat korttipakat olivat nykyisten ohjelmistojen tapaan arvokasta omaisuutta ja niitä hankittiin kilpailijoilta jopa teollisuusvakoilun avulla.

## Turingin kone ja von Neumann -arkkitehtuuri

Varhaisissa laskentakoneissa oli tyypillistä, että kaikki muistityypit olivat erillisiä ja että ne toteutettiin eri teknologioilla. Ei ollut lainkaan ilmiselvää, mikä olisi paras tapa organisoida muisti. Oli myös epävarmaa, voiko edes olla olemassa yleispätevää automaattista laskentakonetta. Alan Turingin suuri kontribuutio vuonna 1936 oli todistaa teoreettisesti, että tällainen yleispätevä laskentakone voidaan rakentaa ja että sillä voidaan ratkaista kaikki ratkaistavissa olevat ongelmat, kunhan vain muistitilaa ja aikaa on ”riittävästi”. Todistuksessaan käyttämässä Turingin koneessa muisti oli äärettömän pituinen nauha, johon talletettiin sekä ohjelma, syötteet, välitulokset että lopputulos. Juuri tämä yhtenäinen, kaikkeen tarkoitukseen sopiva muisti on nähdäkseni nykyaikaisen tietokoneen perusominaisuus, joka erottaa sen aikaisemmista laskentakoneista. Maailma on täynnä ”ensimmäisiä tietokoneita”, mutta useimmat niistä ovat em. määritelmän mukaan ”vain” nykyaikaisen tietokoneen edeltäjiä. Turingin kone itse ei ole ensimmäinen tietokone, koska se on matemaattinen malli eikä käytännössä toimiva toteutus.

”Suomen ensimmäinen tietokone” ESKO ei ylläolevan määritelmän mukaan ollut tietokone, koska sen käskyt luettiin reikänauhalta, syötteet toiselta reikänauhalta tai magneettirumpumuistista, ja tulos joko kirjoitettiin suoraan sähkökirjoituskoneelle tai kolmannelle reikänauhalle[1]. Se, että ESKO ei yllämainitun näkemyksen mukaan olisi ollut tietokone, ei mitenkään vähennä sen vaikutusta tietotekniikan pioneerityönä Suomessa. Yhtä nurinkurista olisi väittää, että ENIACilla ei olisi ollut merkitystä!

Ensimmäisen toimivan todellisen tietokoneen (EDVAC) suunnittelivat Mauchly ja Eckert Pennsylvanian yliopistossa USA:ssa. Tutkimusryhmää kesällä 1944 konsultoinut John von Neumann kirjoitti suunnitelman loogisesta rakenteesta raportin[2], joka sitten levisi sen ajan tietokoneiden suunnittelijoiden käyttöön. Tässä tietokoneessa on väylällä kytketty suoritin ja muisti, jossa suoritin hakee käskyjä yksi kerrallaan suoritusta varten. Keskeneräisenäkin raportti oli niin selkeä, että sen määrittelemä ”von Neumann” -arkkitehtuuri hyväksyttiin yleisesti järkeväksi tavaksi toteuttaa tietokone. Itse von Neumannin osuus EDVACin suunnittelussa on kiistanalainen, koska EDVAC pohjautuu paljolti Mauchlyn ja Eckertin kokemuksiin aikaisemman ENIAC-koneen kanssa. Toisaalta on selvää, että von Neumann ymmärsi hyvin Turingin todistuksen laajat seuraamukset ja että hänellä oli ainakin hyvä mahdollisuus antaa merkittävä käsitteellinen kontribuutio EDVACin suunnitteluun tuona kesänä. Niin tai näin, Mauchlyn ja Eckertin mielestä ”heidän” arkkitehtuurinsa nimettiin jonkun toisen mukaan.

Mauchlyn ja Eckertin ENIACista saama kokemus osoitti, että ohjelmamuisti tulee toteuttaa samalla teknologialla kuin syötemuistikin. Tällöin ohjelman vaihtaminen on yhtä helppoa kuin syötteen vaihtaminen. Kaiken muistin tulee olla samanlaista, koska muuten esimerkiksi käskyjen lukeminen muita hitaamman teknologian muistista rajoittaa laskentanopeutta merkittävästi. Kun kaikki muistityypit ovat yhteneviä, niin yhden ohjelman tuottamaa tulosta voidaan suoraan käyttää seuraavan ohjelman syötteenä tai käskyinä. Jos muistityypit on toteutettu eri teknologioilla tai edes erillisillä laitteilla, ei tämänkaltainen itseäänmuuttava laskenta ole mahdollista ilman operaattorin (ihmisen) väliintuloa.

EDVACin valmistuminen kuitenkin viivästyi vuoteen 1952, koska Mauchly ja Eckert jättivät yliopiston vuonna 1946 ja perustivat oman tietokoneita valmistavan yhtiönsä. Vuonna 1949 Maurice Wilkesin ryhmä sai Cambridgessä valmiiksi EDSACin, joka perustui paljolti von Neumannin artikkeliin ja Wilkesin keskusteluihin Mauchlyn ja Eckertin kanssa. Voidaan siis sanoa, että EDVAC oli ensimmäinen suunniteltu moderni tietokone, mutta EDSAC ensimmäinen valmistunut.

## Toteutusteknologian kehitys

Alkuvaiheessa automaattiset laskimet perustuivat mekaanisiin tai elektromekaanisiin releisiin, eivätkä ne yleensä tästä syystä kunnolla koskaan toimineet. Esimerkiksi Konrad Zusella oli jo vuonna 1936 Berliinissä binäärijärjestelmään perustuva mekaaninen tietokone Z1, jossa kukin bitti oli toteutettu mekaanisen releen avulla. Sodan vuoksi Zusen uraauurtava työ jäi kuitenkin katveeseen, eikä siten vaikuttanut tietotekniikan kehittymiseen Englannissa ja USA:ssa.

Vasta kun muistia ja kontrollilogiikkaa toteutettiin tyhjiöputkilla ilman liikkuvia osia, laskentakoneista saatiin kunnolla toimivia. ENIACissa sekä kontrolli että itse laskennan toteuttavat rekisterit oli toteutettu tyhjiöputkilla, mutta ohjelmamuisti oli johtimilla säädelty kytkinlauta ja

[1] Tietotekniikan alkuvuodet Suomessa, Martti Tienari (toim.), 1993
[2] John von Neumann, First Draft of a Report on the EDVAC, 1945. In W. Aspray and A. Burks, editors, Papers of John von Neumann on Computing and Computer Theory, vol 12 in the Charles Babbage Institute Reprint Series for the History of Computing. MIT Press, 1987.

syöte luettiin reikäkorteilta. EDVACissa rekisterit toteutettiin tyhjiöputkilla, mutta varsinainen keskusmuisti (ohjelma-, syöte-, välitulos- ja tulosmuistit kaikki yhdessä) oli toteutettu elohopeaviiveputkilla. Viiveputken alkupäässä sähköinen bittisignaali muutettiin kiteen avulla värähtelyksi, joka eteni putken toiseen päähän. Siellä toinen kide muutti sen takaisin sähköiseen muotoon, jolloin bitti voitiin joko käyttää tai johtaa takaisin putken toiseen päähän muistiputkeen uudelleen talletettavaksi. EDSACissa sekä keskusmuisti että rekisterit oli toteutettu omilla viiveputkillaan. Rekisteriputki oli tietenkin lyhyempi ja siten nopeampi.

Samaan aikaan 50-luvun alussa viiveputkiteknologian kanssa kilpaili markkinoista katodisädeputkeen (CRT ) perustuva Williams Tube -muistiteknologia. Se perustui CRT:n jälkihehkuun, jota voitiin jonkin aikaa hehkuttamisen jälkeen lukea ja tietenkin aika ajoin hehkuttaa uudelleen. Williams Tube -muistin etuna oli homogeeninen muisti, koska muistin jokainen sana oli käytännössä yhtä nopeasti luettavissa. Huonona puolena oli putkien kallis hinta ja lyhyt kestoikä.

Vuonna 1952 Jay Forrester ja Bob Everett keksivät ferriittirenkaan, joka muutamassa vuodessa valtasi täysin muistimarkkinat. Ferriittirengas on pieni magnetoituva rengas, joita oli tuhansia langoitettu matriisiksi. Kahden osoitejohtimen avulla voitiin valita mikä tahansa ferriittirengas ja se voitiin magnetoida kahteen eri suuntaan, vastaten 0- ja 1-bittejä. Lukemisoperaatio poisti magneettisuuden renkaasta, joten muistipaikan lukemisen jälkeen tieto piti aina kirjoittaa sinne uudelleen. Ferriittirenkailla oli muutama erityisominaisuus, joiden vuoksi niitä käytettiin kylmän sodan sotilassovelluksissa vielä pitkään: magneettisuus säilyi renkaassa ilman sähkövirtaa ja ferriittirengasmuisti ei ollut haavoittuva avaruuden säteilylle eikä mahdollisen ydinräjähdyksen aiheuttamalle elektromagneettiselle pulssille.

Bardeen, Shockley ja Brattain keksivät transistorin jo 1948, mutta vasta 1957 tehtiin ensimmäinen transistoroitu tietokone MIT TX-0. Jack Kilby keksi 1956 toteuttaa useita transistoreita yhdellä mikropiirillä, jolloin päästiin eroon suuresta määrästä epäkäytännöllisiä johtoja. Robert Noyce kehitti mikropiirien nykymallisen tasoon perustuvan valmistusmenetelmän 1959 ja perusti valmistusmenetelmäänsä hyödyntävän Intel-yhtiön Gordon Mooren kanssa 1968. Mikropiireihin perustuva DRAM-muisti valtasi muistimarkkinat 1972 ja siinä tilanteessa olemme edelleen.

Muistin toteutuksessa on aina ollut hierarkiatasoja. Ensimmäisissä koneissa suorittimella oli vain muutama rekisteri, jotka oli toteutettu kunkin ajan nopeimmalla teknologialla (esimerkiksi tyhjiöputkilla tai viiveputkilla). Laskenta tapahtui rekistereissä olevalla datalla, mutta niitä oli myös muutama välituloksia varten. Keskusmuisti oli toteutettu jo sitten selvästi hitaammalla ja halvemmalla teknologialla (esimerkiksi viiveputkilla tai magneettirummulla). Tämä peruslähtökohta muistihierarkialle pätee edelleenkin. Rekisterimuistin koko on edelleenkin vain murto-osa (esim. miljoonasosa) keskusmuistin koosta ja rekisterit ovat edelleenkin kertaluokkaa nopeampia kuin keskusmuisti. Nykyään rekisterit toteutetaan nopeammalla SRAMilla ja keskusmuisti hitaammalla DRAMilla.

## Käytännön ja teorian ristiriita – mutta vain teoriassa

Von Neumann –arkkitehtuurissa konekäskyt haetaan homogeenisesta muistista ja suoritetaan yksi kerrallaan. Näistä molemmista perusperiaatteista on joustettu jo kauan aikaa. Maurice Wilkes kehitti ensimmäisen välimuistiratkaisun vuonna 1965. Sen avulla kurottiin umpeen rekistereiden ja keskusmuistin välistä nopeuseroa. Vaikka nykyisissä koneissa on jo 3 tasoa eri nopeuksista välimuistia, niiden käyttö on edelleen tuntumatonta suorittimen näkökulmasta, koska välimuistissa olevaan tietoon viitataan samalla tavoin kuin keskusmuistiinkin. Konekäskyissä viitataan edelleen ainoastaan homogeeniseen keskusmuistiin. Välimuisti ei siis tuonut loogisesti mitään uutta von Neumann –arkkitehtuuriin, vaan ainoastaan nopeutti käytössä olevaa muistia. Uusimpien

suorittimien konekäskyt tosin sisältävät vihjeitä laitteistolle siitä, minkä tason välimuistiin viitattu tieto kannattaisi ehkä tallettaa.

Virtuaalimuisti antaa mielikuvan, että meillä olisi levymuistin kokoinen ja keskusmuistin nopeuksinen muisti. Tämä on kuitenkin vain harhaa von Neumann –arkkitehtuurin näkökulmasta. Käytännössä tieto joko löytyy keskusmuistista tai sitten ei. Jos tieto pitää hakea levymuistista, ohjelman suoritus keskeytetään samalla tavalla kuin minkä tahansa muunkin keskeytystilanteen vuoksi ja sitä jatketaan samasta kohtaa sitten myöhemmin, kun tieto on saatu kopioitua levyltä keskusmuistiin. Yhden ohjelman suoritusaikana viitattu tieto löytyy siis aina keskusmuistista tavanomaisen von Neumann –arkkitehtuurin mukaisesti. Turingin teorian kannalta muistihierarkialla ei ole väliä, koska muistin nopeus ei siinä ole olennainen tekijä.

Jo vuonna 1962 Atlas-tietokoneessa käskyjä suoritettiin liukuhihnalla, jossa usea peräkkäinen käsky saattoi olla samanaikaisesti suorituksessa, kukin eri vaiheessa. Käsitteellisesti tästäkään ei tullut ristiriitaa von Neumann –arkkitehtuurin kannalta, koska liukuhihnoitetun koneen tuli käyttäytyä konekäskyjen suorituksen kannalta samalla tavalla kuin, jos käskyjä olisi suoritettu yksi kerrallaan alusta loppuun. Turingin teorian kannalta tämä on tärkeätä, koska Turing-koneessa operaatioita tehdään yksi kerrallaan.

Nykyisissä koneissa tällaista käskytason rinnakkaisuutta on vielä paljon lisää. Useita käskyjä haetaan muistista samalla kertaa ja ne ovat samanaikaisesti suorituksessa (usean käskyn sama vaihe samalla kertaa). Superskalaariarkkitehtuuri tarkoittaa juuri sitä, että yhdellä kertaa voidaan saada valmiiksi useampi kuin yksi käsky. Tämä ei aiheuta käsitteellisiä ongelmia, sillä ainahan voidaan ajatella neljän samaan aikaan muistista haetun konekäskyn muodostavan yhden hyvin ison konekäskyn. Uusimmissa suorittimissa voidaan suorittaa myös usean ohjelman useaa käskyä (ei siis ainoastaan yhden ohjelman useaa käskyä) samaan aikaan, jolloin esimerkiksi yhden ohjelman yhden käskyn aiheuttaman muistiviitteen odotusaikana voidaan suorittaa toisen ohjelman käskyjä. Tästäkään ei ole haittaa käsitteellisesti, koska voimme kuvitella kyseisen suorittimen sisältävän usean virtuaalisen suorittimen ja tällöin kukin virtuaalisuoritin on liukuhihnoitettu von Neumann –arkkitehtuurin mukainen suoritin. Myös käyttöjärjestelmä näkee yhden tällaisen hyperlangoitetun suorittimen asemesta monta virtuaalisuoritinta.

Käytännössä nykyisissä tietokoneissa ei ole enää pitkään aikaan toimittu puhtaasti von Neumannin raportin ja siis Turingin periaatteiden mukaisesti. Käskytason rinnakkaisuus maksimoidaan ja toiminnan oikeellisuus määritellään kuitenkin sen mukaan, että lopputulos ainakin jollain tasolla on sama kuin mitä se olisi tavanomaisen von Neumann –arkkitehtuurin koneessa, joka taas noudattaa Turingin periaatteita. Transmetan Crusoe-suorittimessa (Intel-arkkitehtuurin) konekäskyjä suoritetaan pilkottuna hiukkasen verran sekalaisessa järjestyksessä nopeuden maksimoimiseksi niin kauan kuin sekajärjestys ei haittaa suorituksen oikeellisuutta. Aina silloin tällöin kuitenkin havaitaan, että käskyt olisi pitänyt tehdä oikeassa järjestyksessä. Tällöin viime aikoina suoritettu työ tehdään uudelleen hitaammin, mutta täsmälleen oikeassa järjestyksessä. Kun semanttinen ongelmakohta on saatu ratkottua, suoritus voi jatkua nopeana pienen epäjärjestyksen vallitessa.

Teoriassa von Neumann –arkkitehtuurin periaatteet pätevät edelleen, mutta niitä käytetään nykyään funktionaalisen toiminnan määrittelyn asemesta määrittelemään koneen looginen toiminta. Kaikki von Neumann –arkkitehtuurin mukaiset koneet ovat teoreettisesti yhteneviä Turingin koneen kanssa, joten myös kaikki nykyiset tietokoneet ovat laskentavoimakkuudeltaan samanlaisia Turingin koneen kanssa. Niilläkin voidaan ratkaista kaikki ratkaistavissa olevat ongelmat, kunhan vain muistia ja aikaa on riittävästi.