Tampereen yliopisto

Viivi Ohtonen

JÄRJESTELMÄPIIRIT

Kandidaatintutkielma
Informaatioteknologian ja viestinnän tiedekunta
Tarkastaja: Katja Laine
Huhtikuu 2024

TIIVISTELMÄ

Viivi Ohtonen: Järjestelmäpiirit
Kandidaatintutkielma
Tampereen yliopisto
Tieto- ja sähkötekniikan tutkinto-ohjelma
Huhtikuu 2024

Järjestelmäpiiri (System on Chip) tarkoittaa järjestelmää, jonka elektroniset komponentit on sijoitettu yhdelle sirulle siten, että ne muodostavat toimivan systeemin tietylle sovelluskohteelle. Järjestelmäpiirit ovat ajankohtainen aihe, sillä niitä hyödynnetään laajasti eri aloilla viihde-elektroniikasta avaruusteknologiaan. Tämä työ on yleiskatsaus järjestelmäpiireistä, ja se on suunnattu kaikille aiheesta kiinnostuneille ilman aiempaa tietämystä aiheesta. Aihe on laaja, ja työ on rajattu siten, että se ei käsittele tarkasti järjestelmäpiirien valmistamista puolijohde- ja transistoritasolla. Tutkielma keskittyy järjestelmäpiirien suunnitteluvaiheisiin, sovelluskohteisiin, haasteisiin ja tulevaisuudennäkymiin. Millaisissa sovelluksissa järjestelmäpiirejä hyödynnetään? Millaisia haasteita kyseiseen teknologiaan liittyy ja miten niihin on vastattu? Onko järjestelmäpiireillä tulevaisuutta vai onko olemassa kestävämpiä ratkaisuja?

Työ avaa lukijalle järjestelmäpiirien historiaa ja sisäistä arkkitehtuuria. Järjestelmäpiiri sisältää useita rakennusosia, kuten prosessoreja ja muistiyksiköitä. Näitä rakennusosia ja niiden integroimista toisiinsa voi suunnitella monella eri abstraktiotasolla. Suunnittelun alhaisimpana abstraktiotasona pidetään elektroniikkasuunnittelua. Järjestelmäpiiri koostuu puolijohdemateriaalista valmistetusta alustasta ja sen päälle sijoitetuista transistoreista. Transistoreja on miljoonia yhdellä sirulla, ja niiden määrä kasvaa jatkuvasti. Sirun koon minimoimista varten hyödynnetään sähköisten komponenttien ja järjestelmän sisäisen logiikan skaalaustekniikoita. Skaalaus ei kuitenkaan ole mahdollista loputtomiin, mikä tuo omat haasteensa. Monen sirun teknologia (System on Multi-Chip) jakaa suuren järjestelmän usealle sirulle, jotka ovat yhteydessä toisiinsa. Suunnittelun ylimpänä abstraktiotasona pidetään korkean tason synteesiä, jossa suunnittelua tehdään ohjelmointikielillä ja yksityiskohtaiset porttitason ja ajoituksen määritelmät jätetään automaatiotyökalujen vastuulle.

Työ käsittelee järjestelmäpiirien turvallisuusominaisuuksia ja -haasteita. Turvallisuus on alalla ajankohtainen aihe, sillä järjestelmäpiirejä hyödynnetään yhä enemmän esineiden internetin tuotteissa, kuten lääketieteen puettavassa teknologiassa sekä kodin ja ajoneuvojen turvajärjestelmissä, jotka käsittelevät suurta määrää kriittistä dataa. Järjestelmäpiirit tarjoavat ratkaisuja suuren datamäärän turvaamiseen, sillä kriittisissä sovelluksissa datan menettäminen voi johtaa tuhoisiin seurauksiin. Järjestelmäpiirit ovat lisäksi kiinnostava teknologia rahoitusalalla, sillä niitä on hyödynnetty ultranopeassa kaupankäynnissä, jossa osakkeita ostetaan ja myydään automaattisten työkalujen avulla. Järjestelmäpiirit voivat olla osana kaikkea, johon liittyy tietoliikenne, tietojenkäsittely ja laitteen sisäinen ajoitus.

Avainsanat: Järjestelmäpiiri, mikrosiru, sovelluskohtainen piiri, esineiden internet, ohjelmoitava järjestelmäpiiri

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin Originality Check -ohjelmalla.

SISÄLLYSLUETTELO

1. JOHDANTO 1
2. JÄRJESTELMÄPIIRIT YLEISESTI 2
2.1 Kehitys 2
2.2 Arkkitehtuuri 2
2.3 Suunnittelun abstraktiotasot 3
3. SOVELLUSKOHTEET 8
3.1 Tuotesuunnittelun vaiheet 8
3.2 Järjestelmäpiirit osana tuotteita 11
4. HAASTEET 13
4.1 Tekniset haasteet 13
4.2 Turvallisuus 14
5. VERTAILU 17
5.1 System on Multi-Chip (SoMC) 17
5.2 Ohjelmoitavat järjestelmäpiirit 18
6. TULEVAISUUS 20
7. YHTEENVETO 22
LÄHTEET 23

LYHENTEET JA MERKINNÄT

CLK	Clock
CS	Compressed Sensing
DFT	Design for Testability
DPM	Dynamic Power Management
DRAM	Dynamic Random-Access Memory
eDRAM	embedded Dynamic Random-Access Memory
FPGA	Field Programmable Gate Array
HDL	Hardware Description Language
IoT	Internet of Things
NoC	Network on Chip
OSI	Open System Interconnection
P	Prosessori
PROM	Programmable Read-Only Memory
RTL	Register-Transfer Level
SoC	System on Chip
SoMC	System on Multi-Chip
TI	Texas Instruments

1. JOHDANTO

Järjestelmäpiiri (System on Chip, SoC) on kokonaisuus, jossa sähköiset komponentit muodostavat tiettyä sovellusta varten räätälöidyn piirin yhdelle mikrosirulle. [1, s. 1] Yhden sirun teknologia sai alkunsa vuonna 1974 Texas Instruments (TI) -valmistajan 4-bittisestä mikrotietokoneesta, jota ennen sovelluksen eri toiminnallisuuksien integrointi oli mahdollista ainoastaan useita mikrosiruja hyödyntäen. Kehityksen jatkuva tavoite on ollut löytää keinoja lisätä enemmän muistia ja toiminnallisuutta sirulle, kuitenkaan kasvattamatta sen hintaa, piirin pinnien määrää tai pinta-alaa. [2] Tähän tarpeeseen SoC-kehitys on vastannut, ja tämä työ tutkii sen sovelluskohteita, ongelmia ja tulevaisuuden näkymiä. Lisäksi työssä vertaillaan järjestelmäpiiriteknologiaa muihin ratkaisuihin ja punnitaan sen mahdollisuuksia ja heikkouksia. Työn tutkimuskysymykset liittyvät siihen, millaiset tulevaisuudennäkymät järjestelmäpiiriteknologialle on, millaisia ratkaisuja haasteisiin on kehitetty ja millaisissa sovelluksissa järjestelmäpiirejä hyödynnetään.

Luku 2 käsittelee järjestelmäpiirien pääpiirteitä kehityksen, arkkitehtuurin ja suunnittelukäytäntöjen näkökulmasta. Luvussa 3 tutustutaan käytännön sovelluskohteisiin, joissa järjestelmäpiirit ovat osana. Järjestelmäpiirien haasteita tarkastellaan luvussa 4 sekä teknisten, että turvallisuuteen liittyvien haasteiden kannalta. Luku 5 käsittelee kilpailevia teknologioita hyötyihin ja haittoihin keskittyen. Tulevaisuudennäkymiin tehdään katsaus luvussa 6. Yhteenveto tiivistää työn tulokset luvussa 7.

2. JÄRJESTELMÄPIIRIT YLEISESTI

Järjestelmäpiirit näkyvät jokapäiväisessä elämässä. Ne ohjaavat esimerkiksi älypuhelinten toimintaa ja vastaavat tarpeeseen yhä pienemmästä, edullisemmasta ja vähemmän virtaa kuluttavasta elektroniikasta. Järjestelmäpiiri koostuu useasta toiminnallisesta lohkosta, joita ovat esimerkiksi prosessorit, muisti, kontrollerit, virranhallintayksiköt ja sensorit. Lisäksi järjestelmäpiireihin sulautetaan valmis ohjelmisto, minkä vuoksi se voi toimia itsenäisesti ilman ylimääräisiä asennuksia. [3, s. 11, 12] Järjestelmäpiirissä toiminnallisuudet on sijoitettu yhdelle sirulle, kun taas muissa ratkaisuissa systeemi koostuu useasta integroidusta sirusta.

2.1 Kehitys

Aikaisemmin mikrosiruteknologia oli rajoittunut hyvin vähäiseen monimutkaisuuteen, sillä digitaalisuunnitteluun ei ollut tarjolla moderneja elektroniikan suunnitteluautomaatiotyökaluja. Vuosien varrella kehitetyt synteesityökalut ovat helpottaneet monimutkaisten systeemien suunnittelua, sillä niiden avulla käsillä tehtävä työn määrä on vähentynyt. [4, s. 6]

Siruteknologian elinkaari alkoi transistorien keksimisestä. Järjestelmäpiiri koostuu nykypäivänä jopa miljoonista transistoreista sijoitettuna puolijohdemateriaalista valmistetulle sirulle. [3, s. 1] Seuraavana elinkaaren vaiheena pidetään laitteistokuvauskielien (Hardware Description Language, HDL) keksimistä. [4, s. 6] Laitteistokuvauskielillä esitetään laitteiston toimintaa tarkasti pienemmällä abstraktiotasolla kuin ohjelmointikielissä. Kuvauskielien keksimisen jälkeen kehitettiin synteesityökaluja, jotka mahdollistivat siirtymisen manuaalisesta suunnittelusta kehittyneisiin menetelmiin. Tämän jälkeen kehitys ulottui sovelluskohtaisiin integroituihin piireihin ja Mooren lakiin, joka sallii jatkuvasti toinen toistaan monimutkaisempia piirien rakenteita. [4, s. 6] [5, s. 127]

2.2 Arkkitehtuuri

Systeemin arkkitehtuurilla viitataan toiminnallisuuden määritykseen sekä laitteiston toteutukseen. Järjestelmäpiirin arkkitehtuuri määrittelee systeemin rakennuspalikoiden, kuten prosessorien ja muistin välistä yhteyttä ja toiminnallisuutta. Tarkemmin se koostuu systeemin eri osien datareiteistä ja niiden ohjausjärjestelmistä. [1, s. 2, 3] Komponentin datareitti kuvaa bittisisäänmenon ja -ulostulon välistä logiikkaa porttitasolla, kun taas ohjausjärjestelmä huolehtii datapolun multipleksereitä ohjaavien signaalien lähettämisestä.

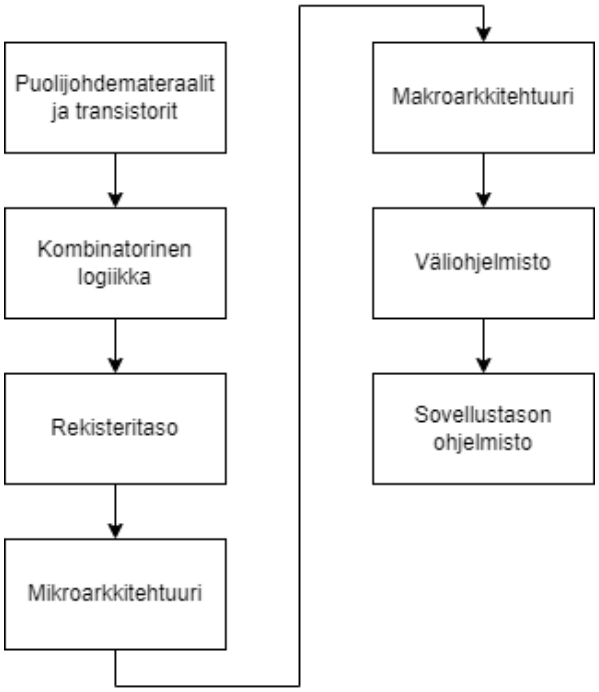
Systeemin arkkitehtuurin laatua määrittää monet tekijät, kuten luotettavuus, saatavuus, skaalautuvuus, joustavuus ja ylläpidettävyys. Luotettava systeemi toteuttaa kaikki käyttäjän vaatimat toiminnot joka hetkellä onnistuneesti. Systeemin saatavuus viittaa sen toiminnalliseen suorituskyvyn osa-alueiden toteutumiseen, kuten riittävään nopeuteen. Skaalautuvuus on tärkeää systeemin kuorman kasvaessa odottamatta, kun taas joustavuus liittyy systeemin muuttuvaan tarpeeseen kuluttaa virtaa ja muistia. [4, s. 8]

Tyypillinen järjestelmäpiiri koostuu useasta osasysteemistä ja komponentista, jotka ovat yhteydessä toisiinsa samalla sirulla. Kaikilla systeemin rakennusosilla on oma tehtävä ja omat toiminnalliset osat.

2.3 Suunnittelun abstraktiotasot

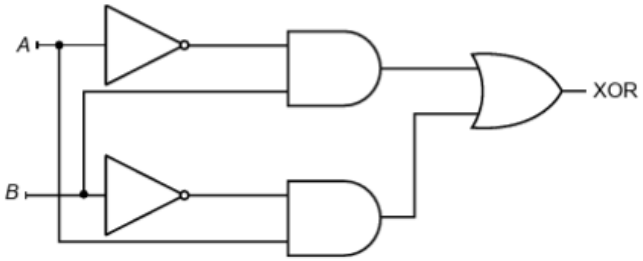
Sulautetut järjestelmät ovat monista komponenteista koostuvia kokonaisuuksia, joiden suunnittelu vaatii paljon aikaa ja resursseja. Järjestelmäpiirien suunnittelussa puhutaan abstraktiotasoista, jotka kuvaavat sitä, kuinka tarkalla tasolla suunnittelua tehdään. Abstraktiotasoista käy hyvin ilmi ohjelmisto- ja laitteistosuunnittelun erot. Ohjelmistosuunnittelussa abstraktiotaso on korkea, ja mitä tarkemmin tarkastellaan laitteiston sisäistä toimintaa, sitä pienempi on abstraktion taso. Olennaista järjestelmäpiirin suunnittelussa on valita, mitkä komponenteista toteutetaan laitteiston (hardware) tasolla ja mitkä ohjelmiston (software) tasolla. [1, s. 5]

Abstraktion alimpana tasona voidaan ajatella olevan puolijohdemateriaalit ja transistorit, joita seuraavat kombinatorinen logiikka, rekisteritaso (Register-Transfer Level, RTL), mikro- ja makroarkkitehtuurien tasot, väliohjelmiston taso sekä lopulta sovellustason ohjelmistosuunnittelu. Tämä seitsemän tason abstraktiomalli tunnetaan nimellä OSI-malli (Open Systems Interconnection). [6, s. 2] OSI-mallia havainnollistaa kuva 1.



Kuva 1. OSI-mallin lohkokaavio.

Kombinatorinen suunnittelu viittaa porttitason suunnitteluun. Tyypillinen esimerkki kombinatorisesta komponentista on summain, jonka ulostulo on yksinkertaisuudessaan sisääntulojen summa. Kombinatorinen logiikka ei tunne muistia, eli edellisellä tapahtumaketjulla ei ole vaikutusta ulostuloon. Portit ovat kombinatorisen suunnittelun kulmakivi. Eri porteilla, kuten AND, OR ja NOT, luodaan logiikka, joka muuttaa sisääntulot ulostuloiksi kyseisen komponentin vaatimusten mukaan. Myös edellä mainittu esimerkki, summain, koostuu porteista, joita pitkin sisäänmenojen bitit muuttavat muotoaan ulostulon biteiksi. [7, s. 47] Kuvassa 2 näkyy XOR-portin toteutus perusportteja hyödyntäen. XOR-portin logiikka tutkii, ovatko sisäänmenot toisistaan poikkeavat.



Kuva 2. XOR-portin toteutus käyttäen NOT-, AND- ja OR-portteja. [7, s. 51]

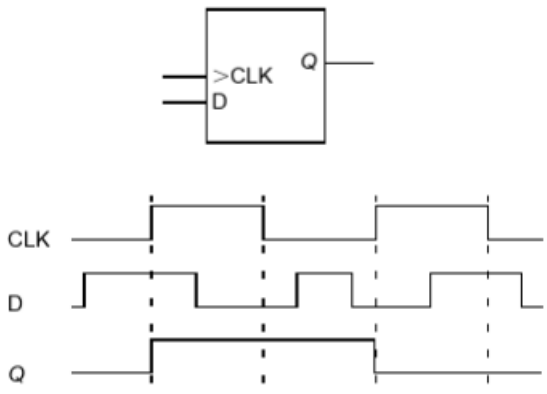
Totuustauluilla kuvataan jokaista sisäänmenon ja ulostulon yhteyttä. Systeemin ja yksittäisten porttien sisäänmeno- ja ulostulobittien määrä voi vaihdella: esimerkiksi AND-porttiin voi mennä useampi sisääntulobitti kerrallaan, mutta kaksi minimissään. Ulostuloja AND-portista tulee vain yksi. Jos kaikki AND-portin sisäänmenot ovat arvoltaan yksi,

ulostulon arvo on yksi. Muissa tapauksissa ulostulon arvo on nolla. OR- ja NOT-portit toimivat eri tavalla: OR-portin ulostulo kertoo, onko vähintään yksi sisäänmenobiteistä arvoltaan yksi, kun taas NOT-portti ottaa vain yhden sisäänmenobitin ja lähettää ulostuloon päinvastaisen bitin.

Kombinatorisen logiikan suunnittelussa on tarkoitus minimoida logiikan määrää, mikä tarkoittaa mahdollisimman vähän portteja. Keinoja minimoida logiikan määrää ovat esimerkiksi toistuvien kaavojen havaitseminen, totuustaulujen tulkitseminen, De Morganin lakien ja Karnaugh:n kartan hyödyntäminen. [7, s. 53 ,329]

Kun suunnittelussa otetaan huomioon systeemin muisti, päästään rekisteritasolle abstraktiossa. Sekventiaalisessa logiikassa tarkastellaan sisäänmenojen ja ulostulojen lisäksi tiloja. Sisäänmenot eivät siis yksin määritä ulostulon arvoja, vaan huomioon on otettava sen hetkinen systeemin tila. [7, s. 53] Siinä, missä kombinatorisella logiikalla voidaan toteuttaa summain, voidaan sekventiaalisella logiikalla toteuttaa esimerkiksi laskuri, joka kasvaa yhdellä luvulla kerrallaan. Tämä on tilojen avulla mahdollista, sillä edellinen luku on aina tiedossa.

Sekventiaalista piiriä ohjaa aina vähintään yksi kello, joka ajoittaa systeemiä tietyllä kellotaajuudella. [7, s. 53] Kellojakson vaihtuessa tapahtuu tilanmuutos. Järjestelmä vaatii rekisterejä, jotka säilyttävät informaatiota. Rekisterit toteutetaan kiikuilla [6, s. 54], jotka säilyttävät yhden bitin ja lähettävät sen eteenpäin. Yleisin rekisterikomponentti on D-kiikku, jonka sisäänmenoja ovat sitä ohjaava kello (CLK) ja sisäänmenobitti D, ja ulostulona on ulostulobitti Q, jonka arvo muuttuu aina kellonjakson vaihtuessa riippumatta sisäänmenobitin muutoksista kellojakson aikana. [7, s. 54, 55] Kuva 3 havainnollistaa D-kiikun lohkokuvaa ja sen sisäistä ajoitusta.



Kuva 3. D-kiikku ja havainnollistava esimerkki ajoituksesta. [7, s. 55]

Portit ja muut elektroniset komponentit samassa piirissä muodostavat integroidun piirin, jota kutsutaan myös siruksi. [7, s. 58] Mikroarkkitehtuurin taso tarkastelee kokonaisen järjestelmän yksittäisten komponenttipiirien rakennetta, kun taas makroarkkitehtuuri esittää komponenttien välisiä yhteyksiä korkeammalla tasolla. Integroidut piirit ovat lyhykäisyydessään prosessoreja ja muistisiruja sijoitettuna sulautettuun järjestelmään. [7, s. 59] Prosessorit ovat systeemin aivot, ja ne jaotellaan master- ja slave -prosessoreihin. Slave -prosessorin toiminta perustuu master -prosessorin lähettämiin ohjaaviin signaaleihin.

Usein järjestelmäpiirien prosessorien tärkeimpänä rakennusosana pidetään ALU:a (Arithmetic Logic Unit), joka on vastuussa kaikkien aritmeettisten ja loogisten operaatioiden toimittamisesta. Aritmeettisia operaatioita ovat esimerkiksi yhteen- ja vähennyslasku. Sen sijaan loogisia operaatioita ovat portit, kuten AND, OR ja NOT. ALU on siis osa prosessorin datapolkua, johon kuuluu myös rekisterit. Kontrollipolun signaalit ohjaavat datapolkua esimerkiksi määrittämällä ALU:ssa suoritettavat operaatiot. [4, s. 44, 45, 46]

Korkean tason suunnittelumenetelmien rooli nostaa merkitystään teknologian kehittyessä ja suunniteltavien systeemien monimutkaisuuden kasvaessa. Suunnittelijoiden tehokkuus ja projektien tuottavuus kasvaa siinä missä abstraktiotasokin. Järjestelmäpiirien sovellustason ohjelmistosuunnittelusta käytetään termiä korkean tason synteesi. Korkean tason synteesi jättää suurimman osan laitteiston suunnittelutyöstä käytössä olevan ohjelmiston algoritmien varaan, mikä mahdollistaa laajojen systeemien suunnittelun lyhyessä ajassa. [8, s. 5]

Suunnittelutyökalujen avulla on mahdollista määrittää rekisteritason tarkat datareitit ja ajoitukset, sekä löytää suunnitelmalle sopivin ja tehokkain rakenne. Tietty sovellus voidaan toteuttaa usealla eri rakenteella. Mitä monimutkaisempi koodi on korkealla tasolla, sitä enemmän se kuluttaa resursseja toteutettuna sirulle. Monimutkaisesti toteutettu systeemi on hidas ja sen ylimääräiset komponentit vievät tilaa sirulla. Siksi suunnittelutyökalujen antamat lukuarvot liittyen systeemin toimintaan ovat erityisen tärkeitä, ja hyvällä suunnittelutyökalulla on mahdollista verrata eri toteutustapoja toisiinsa ja kehittää ohjelmointityötä tietojen pohjalta.

Keinoja tehdä systeemin rakenteesta mahdollisimman tehokas on monia. Liukuhihnoitustekniikka (pipelining) mahdollistaa useamman käskyn lähettämisen yhden suorituksen aikana. Systeemin suorituksessa on useita operaatioita, ja liukuhihnoitus mahdollistaa uuden käskyn lähettämisen, kun edellinen käsky on käynyt yhden operaation läpi. Ilman liukuhihnoitusta seuraava käsky odottaisi koko suorituksen ajan. [4, s. 51]

Toinen tapa optimoida suunnitelmaa on rinnakkaisuuden lisääminen. Rinnakkaisuus tarkoittaa sitä, että joitain systeemin operaatioita suoritetaan useampi kerrallaan. [4, s. 53] Tämä parantaa systeemin suoritusaikaa, mutta lisää merkittävästi logiikan määrää, jolloin se vie myös enemmän tilaa sirulta. Liukuhihnoitusta ja rinnakkaisuutta on helppo lisätä korkean tason suunnittelussa ja antaa suunnittelutyökalujen luoda fyysiset muutokset piireihin.

Kaikista korkean tason synteesin hyödyistä huolimatta sen laaja käyttöönotto on ollut hidasta. Suunnittelijoilla on tarve pystyä vaikuttamaan tarkasti systeemin sisäisiin ajoituksiin, ja sen jättäminen ohjelmiston algoritmien varaan ei välttämättä tuo haluttuja tuloksia. [8, s. 5]

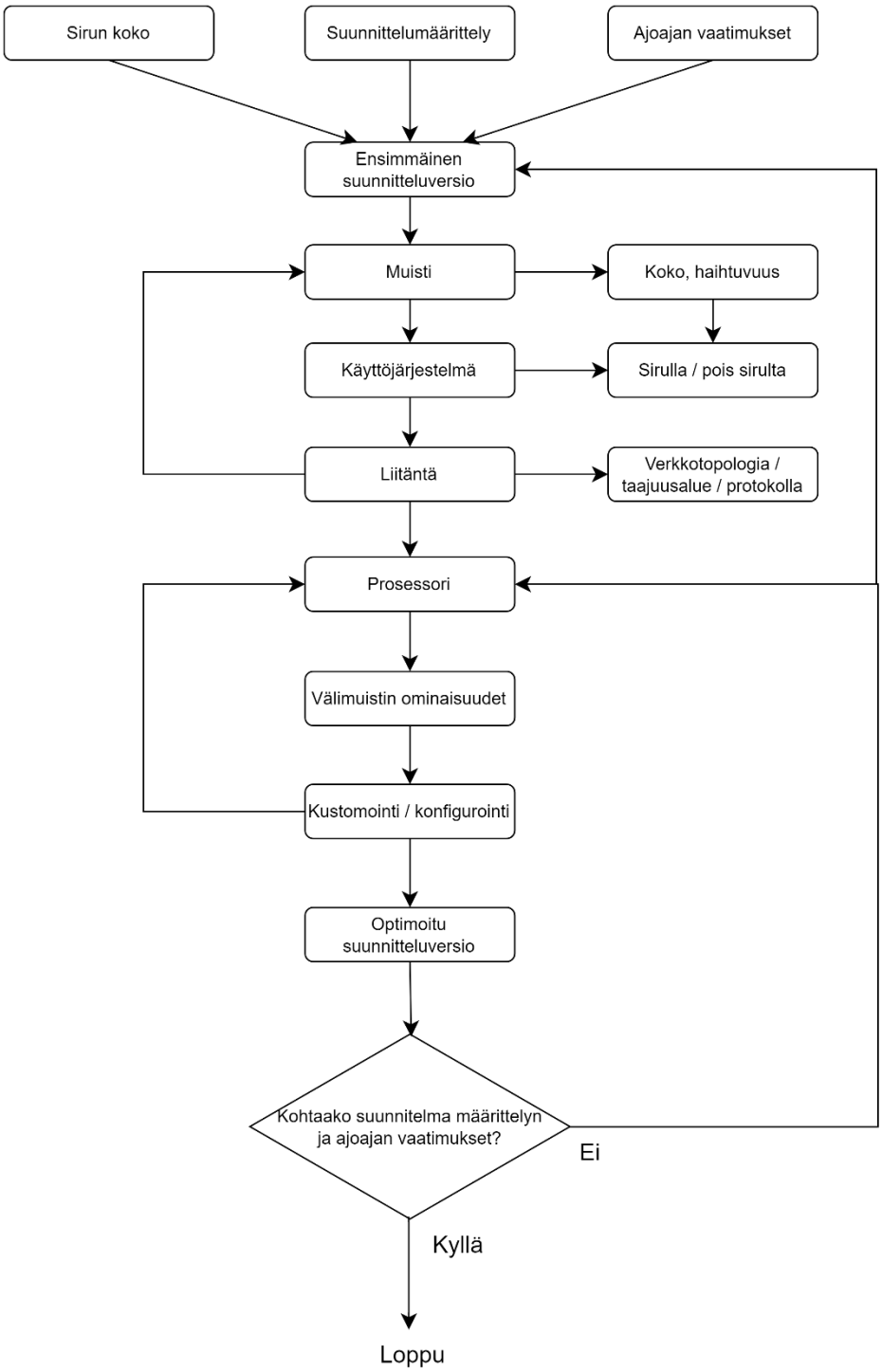
3. SOVELLUSKOHTEET

Järjestelmäpiireillä on suuri rooli teknologian kehityksessä ja entistä kompaktimpien laiteratkaisuiden löytämisessä. Järjestelmäpiirejä hyödynnetään muun muassa viihde-elektroniikassa ja turvallisuusjärjestelmissä, kuten älypuhelimissa ja ajoneuvojärjestelmissä. Prosessi suunnitelmasta valmiiseen tuotteeseen on pitkä ja sisältää useita työvaiheita, joihin palataan monta kertaa ennen sopivan ratkaisun löytämistä.

3.1 Tuotesuunnittelun vaiheet

Sovelluskohtaisen järjestelmäpiirin toteutuksessa on otettava huomioon tehokkuuteen, sirun kokoon, virrankulutukseen, luotettavuuteen ja konfigurointiin liittyvät tekijät, joiden määritteleminen on pohja suunnitelman ensimmäisen version toteutukselle. Ensimmäistä suunnitelmaversiota kehitetään hiljalleen useiden vaiheiden kautta, ja määrittelytekijöihin palataan, mikäli niitä on muutettava sopivan ratkaisun löytämiseksi. [1, s. 246] Kuva 4 havainnollistaa järjestelmäpiirin suunnitteluvaiheet.

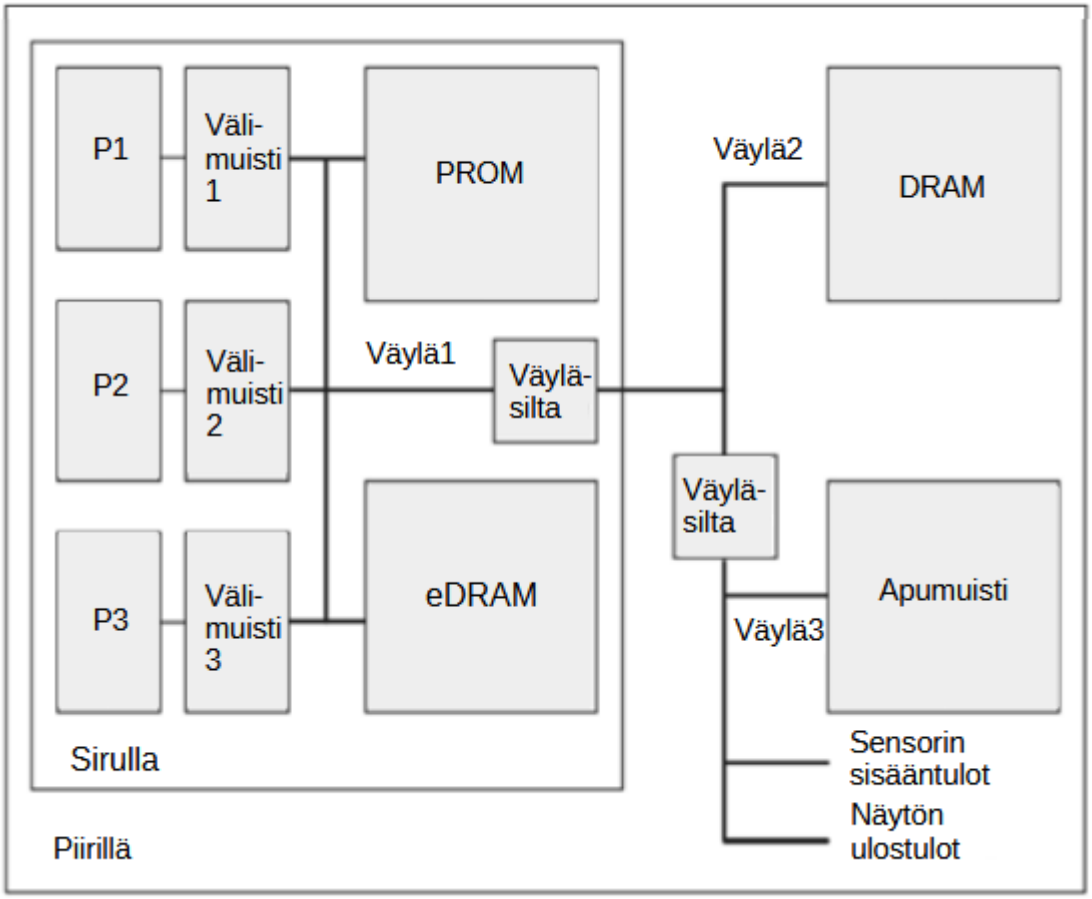
Ensimmäisestä suunnitelmaversiosta edetään muistijärjestelmän suunnitteluun, johon liittyy uusia määritelmiä ja suunnitteluvalintoja. On esimerkiksi valittava, mitkä osat systeemiä sijoittuvat sirulle, ja mitkä piirille sen ulkopuolella. Seuraava vaihe on sisäisten yhteyksien luominen, johon liittyy esimerkiksi aallonpituuden ja protokollan valinta. Voi olla tarpeellista palata vielä muistijärjestelmän kehittämiseen, kunnes paras ratkaisu löytyy. Suunnittelu jatkuu prosessorin ja välimuistin kehittämiseen, minkä jälkeen vielä kustomointiin ja konfigurointiin. Ongelmien ilmentyessä palataan prosessorin, eli systeemin aivojen kehittämiseen. Lopulta päästään optimoituun suunnitelmaversioon ja tarkastellaan, miten hyvin se toteuttaa alkuperäisen määritelmän ja systeemin ajoajan vaatimukset. Parhaassa tapauksessa on päästy suunnitteluprosessin päätökseen, mutta usein palataan vielä prosessorin tai jopa alkuperäisten määritelmien muuttamiseen. [1, s. 247] Suunnitteluprosessi määrittelystä valmiiseen suunnitelmaan kestää noin kuudesta kuukaudesta kolmeen vuoteen riippuen monesta tekijästä, kuten suunnittelutyökaluista, tuotteen monimutkaisuudesta ja siitä, onko suunnitelman lohkoja valmiiksi saatavilla. [3, s. 135]



Kuva 4. Järjestelmäpiirin suunnitteluvaiheet, mukailtu lähteestä. [1, s. 247]

Suunnitteluprosessissa on otettava systeemin toimivan ja tehokkaan rakenteen lisäksi huomioon myös taloudelliset tekijät ja markkinoiden tilanne. Ennen yhtäkään suunnitteluvaihetta tehdään tuotekehityksen budjetti, aikataulutus ja analyysi tuotteen kyvystä kilpailla markkinoilla. Lisäksi kehitetään arvio valmiin tuotteen tehokkuudesta ja myyntihinnasta. [1, s. 249]

Resursseilla on suuri merkitys siihen, millä abstraktion tasolla ja kuinka tehokkailla ohjelmilla tuotteen rakennusosia suunnitellaan. Suunnitelman määrittelyvaatimukset ovat usein lähtöisin asiakkaan toiveista tai vaihtoehtoisesti markkina-analyysistä. Isojen yritysten vaatimukset on usein kirjattu tarkasti ja yksityiskohtaisesti, kun taas pienempien asiakkaiden määritykset tuotteelle voivat olla joustavampia. [1, s. 249] Kuvassa 5 näkyy esimerkki järjestelmäpiirin korkean tason lohkokaaviosta.



Kuva 5. Esimerkki alustavasta korkean tason suunnitelmasta, muokattu lähteestä. [1, s. 250]

Järjestelmäpiirien suunnittelussa jokainen lohko on tarpeen varmentaa, sillä tuote koostuu niin monesta eri osasta, joiden on toimittava juuri oikealla tavalla. Varmennustyöllä pyritään siihen, että valmis tuote toimii heti ensimmäisellä testikerralla. [3, s. 135] Suunnitteluvaiheeseen palaaminen projektin loppuvaiheessa myöhästyttää aikataulua huomattavasti ja aiheuttaa lisäkustannuksia, sillä virhe yhdessä lohkossa voi vaikuttaa ketjureaktiomaisesti kaikkiin sen yhteydessä oleviin lohkoihin. Kustannukset 10-kertaistuvat, mikäli korjaustyötä tehdään suunnittelun jälkeisessä puolijohdekiekon valmistusvaiheessa. Kustannukset jälleen 10-kertaistuvat, jos on ehditty edetä sirun valmistukseen, johon voi upota jopa miljoona dollaria. Sama kaava toistuu tuotekehityksen edetessä

asiakastestaukseen. [3, s. 136] Tuotteen fyysiseen valmistusvaiheeseen ja testaukseen kuluu aikaa noin puoli vuotta olettaen, että tuote suoriutuu testeistä ensimmäisellä kerralla. [3, s. 135]

Varmennus vie suuren osan suunnitteluajasta, minkä vuoksi varmennusteknologiaa on pyritty kehittämään tehokkaammaksi. Varmennuksen tehokkuuden määrittää se, kuinka monta iterointikierrosta suunnitelma vaatii läpäistäkseen testit. Vuonna 2020 tehdyn tutkimuksen mukaan vain 32% uusista järjestelmäpiirituotteista läpäisee käyttötestit ensimmäisellä kerralla. Syy sille, miksi tavoitteista huolimatta varmennus epäonnistuu ennen valmiin tuotteen testausta, on usein vaikeus tunnistaa tuotteen jokainen mahdollinen käyttötilanne. Esimerkiksi älypuhelimessa on niin suuri määrä toimintoja, että niistä jokaisen varmentaminen suunnitteluvaiheessa on suuren työn takana. Järjestelmäpiirien monimutkaisuuden lisääntyessä varmennuksen tehokkuuden tärkeys kasvaa. [3, s. 135, 136]

3.2 Järjestelmäpiirit osana tuotteita

Kaikkialla tietoliikenteessä, tietojen taltioinnissa ja käsittelyssä käytetään perinteisesti järjestelmäpiirejä. Kehittyneet järjestelmäpiirit ulottuvat lääketieteen, viihde-elektroniikan, turvallisuuden, ajoneuvojen ja puolustuksen sovelluksiin. Niitä hyödynnetään lähes kaikissa signaalia, dataa, tietojenkäsittelyä ja tietoliikennettä vaativissa järjestelmissä, kuten esineiden internetin tuotteissa (Internet of Things, IoT). [3, s. 1] Esineiden internet tarkoittaa systeemiä, jossa useampi laite on yhdistetty verkkoon sensoreiden avulla. Laitteet ovat yhteydessä toisiinsa internetin välityksellä, mikä tarkoittaa sitä, että sensoreiden kautta välittyvä data säilötään tietokantaan. [9, s. 1] Muita esimerkkejä järjestelmäpiirien sovelluskohteista ovat suuret datakeskukset, satelliittien tietoliikenne, avaruusteknologia sekä automatisoidut maatalousteknologiat. [3, s. 1, 2]

Järjestelmäpiirien käyttöönotolla on yhteiskunnallisesti tärkeitä vaikutuksia myös tehokkuuden ja taloudellisen hyödyn ohella. Järjestelmäpiirejä on hyödynnetty osana lääketieteen teknologiaa integroimalla niitä osaksi esineiden internetin sovelluksia, millä on positiivisia vaikutuksia potilaiden turvallisuuteen etenkin yksityisyyden suojelun kannalta. Väestön ikääntymisen kannalta kehitys on tarpeellinen myös yhteiskunnan kulujen kannalta. IoT-tuotteet, kuten uusia ratkaisuja terveydenhuollolle tarjoava puettava teknologia tuottavat räjähdysmäisen määrän dataa, mihin ratkaisuna voidaan pitää järjestelmäpiirien kehittämistä osaksi tuotteita. Suuren datan käsittelyyn kuuluu riskejä, ja pahimmassa tapauksessa potilaiden terveyteen liittyvien tietojen katoaminen voi johtaa hengenvaaraan. Järjestelmäpiirit tarjoavat samalla useaan ongelmaan ratkaisun eikä niiden

liittäminen laitteeseen vaadi ylimääräisiä oheislaitteita, mikäli suunnittelu on toteutettu hyvin. [10]

Järjestelmäpiirit voidaan jakaa digitaalisiin, analogisiin ja näiden yhdistelmään. [3, s. 18] Digitaalinen järjestelmä käsittelee dataa diskreetteinä arvoina, kuten bitteinä, kun taas analogisessa järjestelmässä data esiintyy eri amplitudin omaavina pulsseina, kuten ääniaallot. Laajat ja monimutkaiset sovelluskohteet vaativat usein datan käsittelyä analogisen ja digitaalisen yhdistelmänä.

Kun 1990-luvulla pohdittiin järjestelmäpiirien tulevaisuutta, esiin nousi ennustuksia siitä, kuinka paljon matkapuhelimen ominaisuuksia onnistutaan vielä siirtämään yhdelle sirulle. 1999-luvulla matkapuhelinteknologia oli mahdollistanut puhelimen muistin toteutuksen yhdelle sirulle, mutta puhelimen analogisten funktioiden yhdistäminen sirulle oli vasta suunnitteilla. Oli kuitenkin vielä epäselvää, olisiko kehityksellä vaikutuksia kustannuksiin. [11]

Nykypäivänä järjestelmäpiirit mahdollistavat valtavan määrän toimintoja älypuhelimia varten. Älypuhelimissa on erilaisia sensoreita, joiden avulla analogista dataa vastaanotetaan järjestelmäpiirin sisällä. [12, s. 6] Mobiililaitteet tarjoavat jatkuvasti yhä enemmän toimintoja, kuten multimedian suoratoistoa. Toimintojen lisääminen luo vaatimuksia monimutkaisempia järjestelmäpiiriratkaisuja, kuten virranhallintayksikköjen integroimista kohtaan. [13]

Älypuhelimia on kehitetty korvaamaan vanhanaikaiset mekaaniset laitteet, kuten rannekello, kamera ja laskin. Puhelimen esiin ottaminen ja käyttö ei ole kuitenkaan mieluisaa fyysisissä aktiviteeteissa, minkä seurauksena on syntynyt uudestaan tarve puettaville laitteille. Mekaanisten rannekellojen tilalle ovat tulleet älykellot, ja muillekin aina käsien ulottuvilla oleville laitteille on kysyntää. Puettava teknologia voi liittyä muotiin, urheiluun, kodinhoitoon ja terveyteen. Esimerkkejä puettavasta teknologiasta ovat fitness-sensorit sekä älyvaatteet ja -korut. [12, s. 6] Järjestelmäpiirit ovat tärkeä osa puettavan teknologian kehitystä niiden pienen koon ja hinnan vuoksi. Puettavien laitteiden massatuotanto ja halpojen vaihtoehtojen saatavuus markkinoilla riippuu järjestelmäpiirien saatavuudesta. [12, s. 14, 15]

4. HAASTEET

Järjestelmäpiirien käyttöönottoon ja kehitykseen liittyy haasteita, joiden selvittämiseen kaivataan lisää osaajia alalla. Alan trendien ja kehityksen muuttuessa nopeaa vauhtia vaaditaan ammattilaisilta jatkuvaa oppimista ja uusien teknologioiden omaksumista. Elektronisten tuotteiden elinkaari lyhenee jatkuvasti, sillä teknologia kehittyy yhä nopeammin ja uusia parempia ratkaisuja tulee nopeasti edellisten tilalle. [3, s. 10]

4.1 Tekniset haasteet

Mitä monimutkaisempia järjestelmäpiireistä kehitetään, sitä hitaampaa niiden testaaminen ja virheiden korjaaminen on. Kappaleessa 3.1 todettiin, että tuotteen varmennus jokaisen lohkon toteutuksessa on välttämätöntä, jotta valmis tuote toimii ensimmäisellä testikerralla. On kuitenkin mahdotonta varmistaa miljoonien transistorien toimivuus sirulla valmistusvaiheessa. Se, että virheitä ei havaita valmistusvaiheessa ennen sirun integroimista tuotteeseen voi olla tuhoisaa ja aiheuttaa valtavia lisäkustannuksia. Valmistusvaiheen testausmenetelmiä on pyritty kehittämään eri tavoilla. DFT (Design for Testability) -tekniikalla lisätään ylimääräistä logiikkaa suunnitelmaan, jotta suurin osa fyysisistä komponenteista, kuten D-kiikut ja liitännät, voidaan varmentaa. Ylimääräinen testilogiikka kasvattaa järjestelmäpiirin pinta-alaa, mutta vähentää riskejä tuotteen korjauskierroksista koituvista lisäkustannuksista. [3, s. 113]

Suunnittelun haasteisiin liittyy vahvasti yhä monimutkaisemmat kommunikointiprotokollat, jotka määrittävät järjestelmäpiirin sisäisiä yhteyksiä. Verkkosiru (Network-on-Chip, NoC) tarjoaa järjestelmäpiirille standardoidun viestirakenteen lohkojen välistä tietoliikennettä varten. [14, s. 2] Protokollien tulkinta voi olla haasteellista, sillä niiden määritysdokumentit ovat usein pitkiä ja sisältävät epävirallista kieltä. Dokumenttien epämääräisyys tai olennaisen tiedon puuttuminen voi aiheuttaa väärinymmärryksiä suunnittelijoille. Lisäksi järjestelmäpiiriprojekteihin käytetään usein valmiiksi olemassa olevia suunnittelulohkoja, mikä voi olla ongelmallista, mikäli lohkon ja systeemin välisiä integrointivaatimuksia ei ole otettu huomioon. Liitettävän lohkon rajapinnan kuuluu toimia oikein muun järjestelmän kanssa. [13, s. 367]

Virranhallinta on toinen merkittävä ongelma alan kehityksessä. Akun kapasiteetin kuuluisi kasvaa mobiililaitteiden tarjoamien toimintojen määrän kasvaessa. Ongelmaa varten on tutkittu tapoja kehittää järjestelmäpiirin virranhallintaa. Dynaaminen virranhallinta (Dynamic Power Management, DPM) on osa järjestelmäpiirin virranhallintoyksikköä, ja

se hyödyntää systeemin pienen virrankäytön tiloja. DPM-yksikön kehittäminen on ollut avaintekijä puutteellisen akun keston ratkaisemiseen. [15]

Dynaaminen virranhallinta mahdollistaa prosessorin siirtymisen eri tilojen välillä sen ollessa toimettomassa tilassa. Näiden tilojen tarkoitus on säästää virtaa, ja prosessori voi siirtyä esimerkiksi lepotilaan ja syvän levon tilaan riippuen siitä, kuinka pitkän ajan se on toimettomana. Tosin, DPM-toiminnot eivät aina sovellu prosessoreille, sillä DPM-yksikkö voi tulkita väärin prosessorin toimintaa, mikä voi johtaa suorituskyvyn heikkenemiseen. DPM voi tulkita prosessorin olevan toimettomassa tilassa, vaikka todellisuudessa sillä olisi monta käskypyyntöä käsiteltävänä. Eri virransäästötiloilla on eri määritelmä sille, millaisten käskyjen käydessä systeemi herää lepotilasta. Matalan virransäästön tilassa jokainen häiriö herättää systeemin, kun taas syvästä lepotilasta herääminen on mahdollista vain ennalta määritetyillä käskyillä. [15]

DPM-yksikköön liittyy optimointiongelmia, kun järjestelmäpiirien suunnittelu siirtyy kolmiulotteiselle tasolle, ja ytimen tehotiheys kasvaa johtaen sirun ylikuumentumiseen. Optimointiratkaisuja lämmönhallintaa varten voidaan kehittää koneoppimisen avulla. [16]

Virtalähteen kohina on yksi merkittävimpiä ongelmia nykyaikaisissa järjestelmäpiirisovelluksissa. Virtalähteen kohina voi johtaa piirin toimintahäiriöihin, ja se syntyy transistorien kytkennästä aiheutuvien signaaliongelmien, kuten resistiivisen pudotuksen, induktiivisen kohinan ja elektromigraation johdosta. Virtalähteen kohinaa on perinteisesti poistettu erotuskondensaattoreilla, jotka vähentävät jännitteen vaihtelua virtalähteessä. Näiden kondensaattoreiden asettaminen pohjasuunnitelmaan aiheuttaa omia haasteitaan järjestelmäpiirisuunnittelussa. Paras ratkaisu kohinan vähentämiseen olisi sellainen, joka ei aiheuta suuria vaikutuksia sirun muihin toiminnallisiin tekijöihin. [17] [18]

4.2 Turvallisuus

Järjestelmäpiirien tietoturvallisuus on merkittävimpiä haasteita alan kehityksessä. Järjestelmäpiirit ovat ydintekijä esineiden internetin kasvussa, ja siksi niiden merkitys laitteiden välisen yhteyden turvallisuudessa on merkittävä. Esineiden internet kasvaa ennennäkemätöntä vauhtia, ja ulkomaailmaan yhteyden omaavia laitteita on maailmassa arviolta jopa triljoonia vuoteen 2030 mennessä. Nykyajan laitteet, kuten älypuhelimet, puettu teknologia ja käyttötavarat, käsittelevät hyvin henkilökohtaista dataa. Älylaitteiden ollessa arkipäivää lähes jokaisen elämässä on kriittisen tärkeää, että henkilökohtainen data on suojattu ulkopuolisilta. Toisaalta käyttäjät vaativat laitteilta toimintoja, joilla voi olla yhteydessä ulkomaailmaan. [14, s. 1]

Esimerkki tilanteesta, jossa laitteen on oltava yhteydessä ulkomaailmaan, voisi olla jokin kotiautomaatiojärjestelmä, kuten ulko-oven turvajärjestelmä. Ulko-ovi voi sisältää esimerkiksi liiketunnistimen ja kameran. Liiketunnistin aktivoi kameran, joka lähettää kuvamateriaalia asunnon omistajan älypuhelimeen. Jotta turvajärjestelmä toimii toivotulla tavalla, on laitteiden oltava yhteydessä ulkomaailmaan. Datan täytyy säilyä ja olla päätymättä väärään paikkaan, ja virheet sen käsittelyssä voivat johtaa mihin tahansa identiteettivarkauksista ja yrityssalaisuuksien vuotamisesta jopa ihmisen kuolemaan. [14, s. 2]

Suuren datan tehokas ja turvallinen käsittely on siis hyvin ajankohtainen kehityksen kohde järjestelmäpiiriteknologiassa. Suurta dataa koskevia haasteita on lähestytty kehittämällä sirulle energian optimoinnista vastaavia yksiköitä, jotka pakkaavat datan pienempään kokoon ennen lähettämistä. Datan tehokasta pakkaamista varten on löydetty ratkaisuja, kuten CS (Compressed Sensing) -menetelmä, jolla signaalit voidaan pakata suoraan ilman perinteistä näytteenottoa. Lisäksi henkilökohtaisen datan turvaamista varten on kehitetty tunnistusmekanismeja, joiden toiminta perustuu kuviovertailuun. Kuviontunnistusmekanismi vertaa yksilön antamaa datajälkeä tietokantaan tallennettuihin jälkiin. Tunnistus voi toimia esimerkiksi biometrisin perustein, kuten DNA:n tai kasvojen tunnistamisena järjestelmässä. Kaiken kaikkiaan tunnistusjärjestelmän on tarjottava mahdollisuus helppoon, mutta väärinkäyttöjä ja virheitä ehkäisevään tapaan yhdistää data sen omistajaan. [10]

Turvallisuusuhat voivat sijaita myös suoraan järjestelmäpiirin arkkitehtuurissa. Vaarallisiin toimiin kuuluu valmiiksi käytettäviin lohkoihin tahallisesti lisätyt haitalliset suunnittelumuutokset lohkojen suunnittelijoiden toimesta. Myös huomaamattomat virheet suunnittelussa voivat vaarantaa lohkon turvallisuuden tuotteessa. Lisäksi turvamenetelmien vuotaminen järjestelmäpiirien integrointitahoilta on turvallisuusuhka. Ylituotanto lisää mahdollisuutta virheille ja tuoteväärennökset ovat lisäuhka luotettavuudelle. Varmistustekniikoiden ja turvallisuusarkkitehtuurin on otettava huomioon laaja kirjo tuotannon kompastuskohtia. [14, s. 3]

Järjestelmäpiirien jatkuvaa monimutkaisuuden kasvua ei voi painottaa tarpeeksi haasteista puhuttaessa. Uusien yksiköiden lisääminen sirulle kasvattaa luonnollisesti virheiden todennäköisyyttä. On haasteellista varmistaa laajan systeemin toiminta edes normaalikäytössä puhumattakaan kaikista mahdollisista käyttötilanteista. Systeemin monimutkaisuus on turvallisuusriski, sillä turvaominaisuudet sijaitsevat hyvin hajallaan suunnitelman eri lohkoissa. Lohkojen välistä turvallista viestintää varten on ymmärrettävä hyvin laajasti koko systeemin arkkitehtuuria, tunnistettava luotettavat ja epäluotettavat lohkot sekä analysoitava seurauksia, mikäli yksittäisestä lohkosta vuotaisi tietoa. Systeemin turvavaatimukset ovat myös hyvin tuotekohtaisia, eikä esimerkiksi viihdekäyttöön

suunnatulla laitteella kuulukaan olla yhtä suuret turvallisuusvaatimukset kuin hyvin henkilökohtaista tietoa käsittelevällä puettavalla teknologialla. Tästä syystä ylimääräisiä turvallisuusominaisuuksia on aiheellista harkita valmiiksi suunnitelluille lohkoille tilanteen mukaan. [14, s. 4]

Eri tekniikoita hyödyntäen on pyritty kehittämään turvallisuusvaatimusmäärittelyjä sekä arkkitehtuurin laajaa testaamista hyökkäysten varalta. Lisäksi on tutkittu sopivaa suhdetta turvallisuuden ja toimivuuden välillä. Kehityksestä huolimatta alalla turvaudutaan yhä tietoturva-arkkitehteihin, jotka räätälöivät suojausmekanismeja ja hyökkäystestejä tuotekohtaisesti ilman tarkkaa metodologiaa kehitysvaiheessa. [14, s. 5]

5. VERTAILU

Jotta voi ymmärtää järjestelmäpiirien merkityksen teknologian kehityksessä, on tarkasteltava sen rakennetta verrattuna muihin ratkaisuihin, joissa systeemin toiminnallisuus tapahtuu useassa eri osassa yhden sirun sijaan. Vaihtoehtoisia ratkaisuja SoC-arkkitehtuurille on esimerkiksi usean piirin järjestelmä (System on Multi-Chip, SoMC) ja ohjelmoitavat FPGA (Field Programmable Gate Array) -pohjaiset järjestelmät.

5.1 System on Multi-Chip (SoMC)

Kysyntä uusille ominaisuuksille järjestelmäpiirejä hyödyntävissä laitteissa johtaa epäkäytännöllisiin sirun kokoihin. Tähän ratkaisuna on systeemin jakaminen usealle sirulle, mikä puolestaan tuo mukanaan uusia haasteita arkkitehtuurin suunnittelussa sekä lämpö- ja liitäntäsuunnittelussa. Ennen vuotta 2017 sirulla olevien lohkojen pinta-alan skaalaus mahdollisti niiden koon kutistumisen vapauttaen tilaa uusille lohkoille lisäten sirun pinta-alaa vain hyvin vähän. Mahdollisuus skaalata komponenttien aluetta on vähenemässä ajan myötä, ja tulevaisuudessa tiettyjä yksiköitä ei yksinkertaisesti ole mahdollista skaalata enempää. Pinta-alan skaalauksen hidastuessa ylimääräisten toiminnallisuuksien integrointi siruun rajoittuu. Tätä ongelmaa varten tarvitaan uusi metodologia, jolla systeemin kaikki ominaisuudet voidaan integroida yhteen. [19]

Usean sirun järjestelmä mahdollistaa pienen sirukoon, millä on paljon positiivisia vaikutuksia. Piirin pinta-alan minimoinnin ja järkevämmän tilankäytön lisäksi tekniikalla on positiivinen vaikutus kustannuksiin. Systeemin sirut voidaan pakata eri ulottuvuuksilla: 2D, 2,5D ja 3D. 2D-ratkaisussa sirut ovat vierekkäin, mikä vaatii tilaa leveyssuunnassa. Sirut on yhdistetty toisiinsa yhteisen alustan välityksellä. 2,5D-ratkaisussa sirut ovat yhä vierekkäin, mutta alustalevyn ja sirujen välillä on ylimääräinen yhteyksiä mahdollistava kerros, joka mahdollistaa 2D-ratkaisua tiheämmän liitännän johtaen parempaan kapasiteettiin ja suorituskykyyn. 3D-ratkaisussa sirut pinotaan pystysuunnassa, jotta järjestelmä pysyy pienenä vaakatasossa. Tämä vaatii hyvin tiheän yhteyden sirujen välillä sekä virtajakelun alustalta yläsiruille. Jokaisen ratkaisun välillä on eri suhteita esimerkiksi riskien, kustannusten ja tehokkuuden välillä, joten valinta niiden välillä vaihtelee sovelluskohtaisesti. [19]

Usean sirun tekniikka tuo mukanaan haasteita erityisesti 3D-liitännässä. Sirujen pinoaminen aiheuttaa ongelmia lämmön hajautumiselle tarkoitetun pinta-alan pienentyessä verrattuna 2D-ratkaisuun. On otettava huomioon uudet tarpeet arkkitehtuurissa, kuten

virranhallinnassa. On tutkittava, mitä yksiköitä täytyy sijoittaa eri siruille ja kerroksille, ja minkä sijoittelu riittää yhdelle sirulle. Kun yhdeltä sirulta hajotetaan systeemi usealle sirulle, on suunniteltava sopiva liitäntätapa sirujen välille alkuperäisen viiveen ja kaistanleveyden säilyttämistä varten. Haasteilla on vaikutus suunnitteluketjussa ja määrittelyssä. Valinta yhden ja usean sirun toteutuksen välillä riippuu tuotteen vaatimuksista, kuten järjestelmäpiirin koosta. [19]

5.2 Ohjelmoitavat järjestelmäpiirit

Sulautettujen järjestelmien suunnittelulle on ominaista vahva sovelluskohtaisuus. Jokainen järjestelmäpiiri on rakennettu soveltumaan ennalta määritettyyn tarkoitukseen, ja uusi käyttötarkoitus vaatii uuden suunnitelman ja fyysisen piirin. Ohjelmisto on sulautettu piiriin valmistajan toimesta, eikä käyttäjä pääse siihen käsiksi. Vastakohtana sovelluskohtaiselle piirille on ohjelmoitava piiri, joka pystyy mukautumaan eri käyttötarkoituksiin. On olemassa suhde joustavuuden ja sovelluskohtaisen tehokkuuden välillä, sillä mitä paremmin suunnitelman arkkitehtuuri sopii tiettyyn käyttötarkoitukseen, sitä vaikeampi sitä on muokata vastaamaan jotain muuta. [20, s. 1]

FPGA (Field Programmable Gate Array) on ohjelmoitava järjestelmäpiiri, joka koostuu muunnettavista logiikkamoduuleista ja moduulien välisistä yhteyksistä, sekä suurista muisti- ja aritmeettisista lohkoista. [20, s. 1] Nimensä mukaisesti FPGA-piirit ovat alun perin koostuneet sarjasta yksinkertaisia portteja, mutta nykyään ne ovat paljon kehittyneempiä kokonaisuuksia. [21, luku 1] FPGA:t ohjelmoidaan syöttämällä systeemin logiikkaa ja sisäisiä yhteyksiä hallitsevia arvoja bittivirtoina muistiyksiköihin. FPGA:n hyöty suunnittelijalle on vapautuminen yhä monimutkaisempien integroitujen piirien back-end-suunnittelusta, johon kuuluu logiikkasolujen fyysinen sijoittelu, reititys niiden välillä sekä virta-, kello- ja testausrakenteen generointi. FPGA-piirien tärkein ominaisuus on niiden uudelleenkonfiguroitavuus. FPGA-piirien kääntöpuolena on niiden hinta, nopeus ja virrankäyttö, sillä niissä on ylimääräisiä ominaisuuksia, joita kaikki systeemit eivät tarvitse. [20, s. 1, 2]

FPGA-piirejä sovelletaan useisiin tarkoituksiin. Ne ovat kriittinen väline tietoliikenteessä, kuten puhelinmastoissa internetin reitityksessä. Lisäksi FPGA-piirejä hyödynnetään rahoitusalalla ultranopeassa kaupankäynnissä (high-frequence trading), jossa hyödynnetään algoritmeja automaattisessa osakkeiden ostossa ja myynnissä. Ultranopeassa kaupankäynnissä millisekunnit ratkaisevat, ja FPGA-piirit tarjoavat mahdollisuuden kehittää toinen toistaan nopeampia ratkaisuja. Järjestelmäpiirejä, mukaan lukien FPGA-piirejä hyödynnetään paljon maanpuolustusteollisuudessa kuten tutkien sisäisessä signaalinkäsittelyssä. Niillä voi käsitellä tutkakuvia erilaisten matemaattisten suodattimien avulla

nähdäkseen satojen kilometrien päähän. Samalla periaatteella ohjelmoitavia piirejä käytetään infrapunakameroissa havaitsemaan lämpöä valon sijaan. [21, luku 1]

FPGA-piirejä löytyy avaruusteknologiassa, jossa niitä käytetään suojaamaan säteilyn vaikutuksilta laitteisiin ulkoavaruudessa. Myös tekoälyn kehittäjät ovat kiinnostuneet ohjelmoitavista piireistä, sillä niiden avulla voidaan ratkaista monia ongelmia, joihin perinteiset ohjelmointitekniikat eivät riitä. Näihin ongelmiin lukeutuvat esimerkiksi kuvien luokittelu, puheentunnistus ja robotiikan ohjausjärjestelmät. Kaiken kaikkiaan FPGA-piirit ovat hyvä vaihtoehto kaikkeen digitaaliseen elektroniikkaan, jossa suuri kaistanleveys, pieni viive tai korkea prosessointikyky on vaadittu. [21, luku 1]

Lopulta pääsyy sille, miksi FPGA-piirejä ei ole sovelluskohtaisten piirien sijaan kaikkialla, on hinta. FPGA-piirin sisällä on paljon sisäisiä liitäntöjä tukemassa kaikkia mahdollisia toiminnallisuuksia, mikä lisää niiden arvoa. Yksinkertaisessa sulautetussa järjestelmässä hyödyn ja hinnan suhde on tällöin epätasapainossa. FPGA-piirin monimutkaisuutta kuvastaa hyvin se, että niissä on tänä päivänä jopa 50 miljardia transistoria ja 9 miljardia logiikkalohkoa. [3, s. 3] Lisäksi sovelluskohtaiset piirit on helpompi ottaa käyttöön, kun taas FPGA-piirissä kaikki on ohjelmoitava uudestaan ennen sen hyödyntämistä, mikä on hyvin aikaa vievää. Virrankulutus on tärkeä ottaa huomioon pohtiessa ratkaisua FPGA-piirin ja järjestelmäpiirin tai yksinkertaisemman mikrokontrollerin välillä. Sovelluskohtaiseen käyttöön tarkoitetun piirin virrankäyttö voidaan minimoida paremmin, sillä siitä puuttuu ohjelmoitaville piireille ominaisia ylimääräisiä yhteyksiä, jotka kuluttavat virtaa. [21, luku 1]

Vaikka FPGA-piirit eivät ole kilpailukykyisiä hinnan, helppokäyttöisyyden ja virrankulutuksen suhteen, on niillä paljon etuja verrattuna sovelluskohtaisiin piireihin. Etuihin lukeutuu esimerkiksi nopeus ja joustavuus. FPGA-piirit tukevat hyvin laajaa kaistanleveyttä, jolloin ne voivat prosessoida suuria määriä gigabittejä sekunnissa. Niitä käytetäänkin usein videonmuokkauslaitteistoissa, joilla on oltava riittävän laaja kaistanleveys laajoja bittivirtoja varten. Laaja kaistanleveys mahdollistaa datansiirron erilaisista ulkoisista rajapinnoista hyvin nopeasti. [21, luku 1]

6. TULEVAISUUS

Järjestelmäpiiriteollisuus on kehittynyt nopeasti viimeisten vuosikymmenten aikana tehden järjestelmistä toinen toistaan monimutkaisempia, suurempia ja tehokkaampia. Jatkuvasti kehittyvän teknologian ansiosta järjestelmäpiirien kehitysympäristö laajenee jatkuvasti uusilla ominaisuuksilla, mikä mahdollistaa parannettua sirun tehokkuutta ja mallinnusta. [3, s. 8]

Suunnittelijoilta vaaditaan jatkuvasti laajempaa ohjelmisto-osaamista. Logiikkamäärittelyyn ja mallinnukseen käytettävistä laitteistonkuvauskielistä laajassa käytössä on Verilog, VHDL ja SystemVerilog. Vaikka korkean tason synteesi on nostanut suosiotaan, on tärkeää suunnittelijalle ymmärtää rekisteritason yksityiskohtia. Alan osaajilta vaaditaan joustavuutta työskennellä eri ympäristöissä, kuten logiikkasuunnittelussa, synteesissä, ajoitusanalyysissä, fyysisessä piirisuunnittelussa ja FPGA-varmennuksessa. Vaikka FPGA-piirit nähtiin ennen kilpailevana ratkaisuna järjestelmäpiireihin, pidetään niitä nykyään ennemmin täydentävänä työkaluna järjestelmäpiirisuunnittelussa. [3, s. 9]

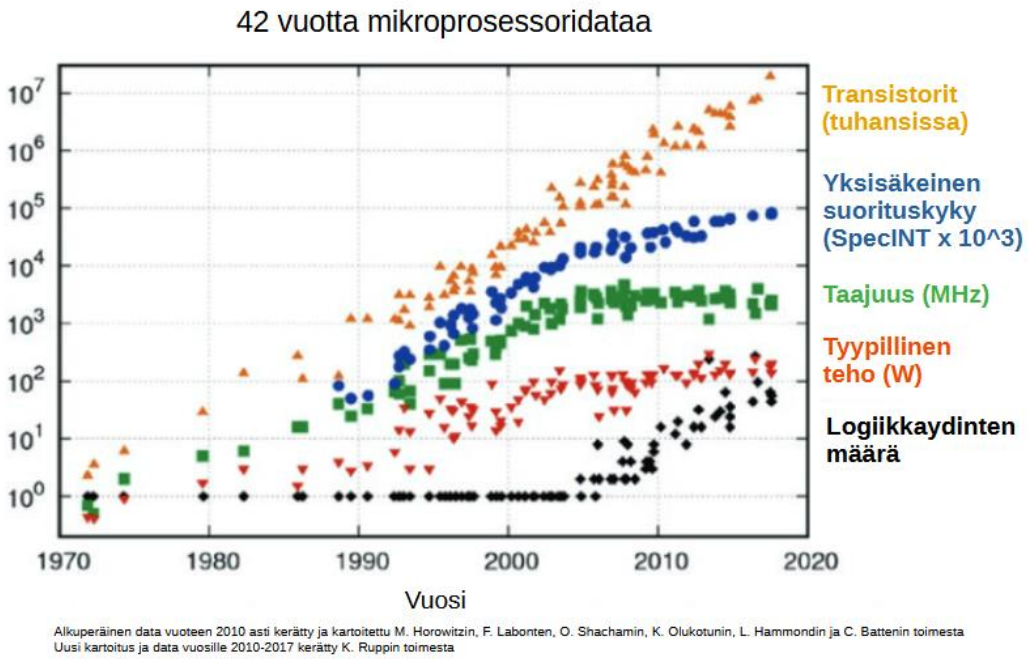
Järjestelmäpiirien monimutkaistuessa tarve uusille suunnitteluohjelmistoille kasvaa jatkuvasti. Synteesiin ja ajoitusanalyysiin käytetään algoritmipohjaisia työkaluja, kun taas piirin komponenttien sijoitteluun ja reititykseen fyysisiä työkaluja. Tulevaisuudessa suunnittelutyökaluissa tullaan hyödyntämään yhä enemmän tekoälyä ja koneoppimista, joiden tarkoitus on tehdä oikeita suunnitteluvalintoja ihmisen puolesta. Tulevaisuudessa työkaluilta odotetaan parempaa tehokkuutta testaamisessa ja virheiden havaitsemisessa, jotta valmis systeemi toimisi ensimmäisellä ajokerralla. [3, s. 9]

Piirien valmistuksen hinta on kasvanut hyvin paljon, mikä korostaa suunnitelman toimivuuden tärkeyttä ensimmäisellä valmistuskerralla. Valmistuksen kustannukset kasvavat, kun piirit ovat entistä skaalatumpia ja sisältävät yhä enemmän vahvistimia pakattuna yhä tiheämmin. Skaalauksen merkitystä alan kehityksessä ei voi korostaa liikaa. Piirejä ei voi skaalata äärettömyyteen, ja tällä hetkellä komponenttien pienentämistä atomikokoon (0,1nm – 0,5nm) pidetään mahdottomana. Vuonna 2019 on valmistunut siru, joka sisältää 2,6 triljoonaa vahvistinta ja 850 000 prosessoriydintä, ja se on suunniteltu 7nm prosessitekniikalla. Tähän verrattuna atomikoon prosessitekniikka on valtavan matkan päässä, mutta osa tutkijoista näkee sen mahdollisena. [3, s. 2, 3]

Tulevaisuudessa sirun kustannuksia ja siten myös tuotteen kustannuksia yritetään saada alas uudelleenkäyttämällä valmiita lohkoja. Lohkojen turvallinen uudelleenkäyttö vaatii strategiasuunnittelua, ja turvallisuus on otettava huomioon. Turvallisuuden

kehittäminen on ajankohtaista myös korkean tason suunnittelussa, jossa suunnitelman yksityiskohtainen arkkitehtuuri ja ajoitus jää suunnittelutyökalun vastuulle. [4, s. 15]

Tulevaisuuden trendeihin järjestelmäpiirialalla lukeutuu monia kehityskohteita, jotka liittyvät järjestelmäpiirien monimutkaisuuden hallinnan lisäksi ihanteellisen suorituskyvyn, tehon ja pinta-alan tavoitteluun, sekä suunnittelumetodologiaan. [3, s. 2] Vuosikymmenten varrella systeemien nopeudessa on tapahtunut ilmiömäisiä muutoksia, joita kuva 6 havainnollistaa. Kuvassa näkyy nopeuden, vahvistintiheyden, tehon ja logiikkaydinten määrän muutokset 42 vuoden ajalta. Muutokset ovat vaatineet jatkuvaa kehitystä suunnittelumetodologiassa. Tämä on ollut mahdollista suunnittelutyökalujen kehityksen ansiosta. Valinta eri suunnittelumetodologioiden välillä riippuu vaihtuvista tekijöistä, kuten tehokkuudesta, kustannuksista ja tuotantomäärästä. Suunnittelumenetelmiä ovat esimerkiksi räätälöity, solukeskeinen ja taulukkorakenteinen suunnittelu. Monimutkainen järjestelmäpiiri voi hyödyntää kaikkia edellä mainittuja menetelmiä. [3, s. 7, 8]



Kuva 6. 42 vuotta mikroprosessorien kehitystä, muokattu lähteestä. [3, s. 6]

Teknologisen kehityksen vaiheista elämme älyllistämisen aikakaudessa, joka on seurausta esineiden internetin aikakaudesta. Näinä vuosina teknologialla kehitetään kaikista käyttämistämme laitteista älykkäitä koneoppimisen ja tekoälyn avulla. Tämän kaiken on mahdollistanut integroitujen piirien teknologia, joka alkoi vain muutaman vahvistimen yhdistämisestä 1960-luvulla, ja joka on kehittynyt tähän pisteeseen monen vaiheen kautta mahdollistaen laajoja systeemeitä ja verkkoja luotuna yhdelle sirulle. [4, s. 15]

7. YHTEENVETO

Työssä tutustuttiin järjestelmäpiireihin perusteellisesti alan kirjallisuuden ja tutkimusten pohjalta. Järjestelmäpiirit ovat sovelluskohtaisia piirejä, jotka mahdollistavat koko järjestelmän toiminnan yhdelle sirulle. Järjestelmäpiirien kehitys koostuu monesta vaiheesta ja suunnitteluun on monta abstraktista lähestymistapaa. Nykyään suunnittelu tapahtuu harvoin kombinatorisen logiikan tasolla, kun taas korkean tason synteesin hyödyntäminen automaatiotyökaluilla on jatkuvassa nousussa. Järjestelmäpiirejä on kaikkialla viihde-elektroniikasta lääketieteeseen ja avaruusteknologiaan, joten niillä on valtava yhteiskunnallinen merkitys, minkä vuoksi työ painottaa turvallisuuden ja testauksen tärkeyttä suunnitteluprosessissa. Testausmenetelmien lisäksi järjestelmäpiirien jatkuvasti kasvava monimutkaisuuden taso luo omat haasteensa, joita työssä käsiteltiin. Haasteiden jatkumona työssä vertailtiin sovelluskohtaisia piirejä ohjelmoitaviin piireihin ja yhden sirun piirejä usean sirun piireihin. Vaihtoehtoiset teknologiat eivät kuitenkaan tarkoita uhkaa, vaan ennemmin tukevia ratkaisuja räätälöimään oikeanlaisia piirejä mahdollisimman moneen eri käyttötarkoitukseen. Sekä suunnittelumenetelmien että sirun ominaisuuksien valintaan vaikuttavat niin taloudelliset kuin laitekohtaisetkin vaatimukset.

Järjestelmäpiirien tulevaisuus on täynnä potentiaalia ja uusia haasteita. Kuluttajat vaativat uusia ominaisuuksia älylaitteilta, jotka ovat jatkuvasti enemmän osana arkea. Lisäksi maanpuolustus- ja lääketiede vaativat uusia ratkaisuja esimerkiksi suuren datan käsittelyssä. Tästä syystä laitteiden prosessorien sisältämä transistorien määrä kasvaa eksponentiaalisesti vuosikymmenittäin. Sirun komponenttien määrän kasvattamisesta huolimatta sirun kokoa on onnistuttu minimoimaan komponenttien ja sisäisen logiikan skaalaustekniikoiden myötä. Skaalaus ei kuitenkaan ole mahdollista äärettömyyteen, ja on ennustettu, että lähivuosina skaalaus hidastuu ja lopulta päättyy. Tämän myötä ylimääräisten toiminnallisuuksien integroiminen yhdelle sirulle kasvattamatta samassa suhteessa sirun kokoa tulee olemaan mahdotonta ilman vaihtoehtoista metodologiaa.

Ala vaatii hyvällä ongelmanratkaisutaidolla varustettuja suunnittelijoita, jotka ovat valmiita omaksumaan jatkuvasti uusia automaatiotyökaluja ja suunnittelumenetelmiä. Tekoäly ja koneoppiminen tarjoavat logiikkalohkojen uudelleenkäytön lisäksi apua suunnittelijoille, kun monimutkaisten järjestelmien suunnittelu ei ole enää mahdollista ihmisaivoille.

LÄHTEET

[1] Flynn, M. J. & Luk, W. (2011) *Computer system design: system-on-chip*. 1st edition

[2] Raghunathan, K. R. (2021) History of Microcontrollers: First 50 Years. *IEEE MICRO*. [Verkossa] 41 (6), 97–104.

[3] Chakravarthi, V. S. (2022) *A practical approach to VLSI System on Chip (SoC) design : a comprehensive guide*. Second edition. Cham, Switzerland: Springer International Publishing.

[4] Chakravarthi, V. S. & Koteshwar, S. R. (2023) *System on chip (SOC) architecture : a practical approach*. [Verkossa]. Cham, Switzerland: Springer.

[5] Iniewski, Krzysztof. (2013) *Embedded systems hardware, design, and implementation*. Hoboken, N.J: John Wiley & Sons, Inc.

[6] Henkel, J. (2021) *Dependable embedded systems*. 1st edition 2021. Jörg. Henkel & Nikil. Dutt (eds.). [Verkossa]. Cham: Springer Nature.

[7] Ganssle, J. G. (2008) *Embedded hardware*. Amsterdam ; Elsevier/Newnes.

[8] Coussy, P. & Morawiec, A. (2008) *High-Level Synthesis: from Algorithm to Digital Circuit*. 1. Aufl. [Verkossa]. Dordrecht: Springer Netherlands.

[9] Márquezm Fausto Pedro García (ed.) (2021) *Internet of Things*. [Verkossa]. London: IntechOpen.

[10] Djelouat, H. et al. (2018) System-on-Chip Solution for Patients Biometric: A Compressive Sensing-Based Approach. *IEEE sensors journal*. [Verkossa] 18 (23), 9629–9639.

[11] Silcott, G. et al. (1999) SOCs drive new product development. *Computer (Long Beach, Calif.)*. [Verkossa] 32 (6), 61–66.

[12] Mishra, S. (2015) *Wearable Android : Android wear & Google Fit app development*. 1st edition. Hoboken, New Jersey: Wiley.

[13] Sudeep Pasricha, N. D. (2008) *On-Chip Communication Architectures - System on Chip Interconnect*. 1st edition. Vol. v.Volume-. San Diego: Elsevier.

[14] Bhunia, Swarup. et al. (eds.) (2017) *Fundamentals of IP and SoC Security Design, Verification, and Debug*. 1st ed. 2017. [Online]. Cham: Springer International Publishing.

[15] You, D. & Chung, K.-S. (2014) Dynamic power management for embedded processors in system-on-chip designs. *Electronics letters*. [Verkossa] 50 (18), 1309–1310.

[16] Narang, G. et al. (2023) Dynamic Power Management in Large Manycore Systems: A Learning-to-Search Framework. *ACM transactions on design automation of electronic systems*. [Verkossa] 28 (5), 1–21.

[17] Moustakas, K. & Noulis, T. (2022) Wide band differential on-chip monitoring of system on chip power supplies noise. *International journal of electronics and communications*. [Verkossa] 154154311-.

[18] Mitra, P. & Sarkar, A. (2020) Soft Computing Techniques Based CAD Approach for Power Supply Noise Reduction in System-on-Chip. *Journal of electronic testing*. [Verkossa] 36 (5), 665–669.

[19] C. Patrick et al., "From System-on-Chip (SoC) to System on Multi-Chip (SoMC) architectures: Scaling integrated systems beyond the limitations of deep-submicron single chip technologies," 2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits), Honolulu, HI, USA, 2022, 1-2

[20] Nurmi, Jari. (ed.) (2007) *Processor Design System-On-Chip Computing for ASICs and FPGAs*. 1st ed. 2007. [Verkossa]. Dordrecht: Springer Netherlands.

[21] Russell Merrick (2023) *Getting Started with FPGAs*. No Starch Press.