



Virtual architectures for future Systems-on-Chip

System-on-Chip (SoC) is taking advantage of the billion transistor capacities provided by deep submicron technologies. The design is undergoing a paradigm shift from gates and functional blocks to the interconnects and communication. On high level, the design relies on virtual architectures describing the functionalities and communication requirements between them. Gradually, the virtual wires are converted to actual physical communications. The hierarchical OSI model plays also a role in the design, describing the on-chip network, data link and physical layers.

Due to problems in keeping the large systems synchronized to high frequency clocks, the preferred architectural scheme is globally asynchronous, locally synchronous (GALS). The synchronous subsystems are based on functional IP blocks, e.g., different types of processors. The communication networks are based on interconnect IP and underlying models of the technology, possibly setting requirements for buffer insertion, signalling schemes, error correction, etc. in the physical layer.

These issues are studied in COMPLAIN (Communication Platform Architectures for Gigascale Integration) project at Tampere University of Technology (TUT) and University of Turku (UTU). This is a part of the larger Finnish SoC infrastructure consisting of national research projects, international co-operation, annual international SoC seminar in Tampere, and a dedicated graduate school TELESOC (Telecommunication System-on-Chip Integration) for PhD studies in SoC. Contact: Prof. Jari Nurmi, TUT jari.nurmi@tut.fi

Virtuaaliarkkitehtuurit tulevaisuuden järjestelmäpiirien perustana

SoC-piirien suunnittelu



Järjestelmäpiirien (System-on-Chip, SoC) suunnittelussa tavoitteena on hyödyntää miljardien transistorien kapasiteettia, jota uudet piirisukupolvet tarjoavat. Suunnittelun ongelmat ovat siirtymässä lohkoista niiden väliseen johdotukseen, eli laajemmin piirinsisäiseen kommunikaatioon. Tämä on aivan uusi suunnittelun lähtökohta.

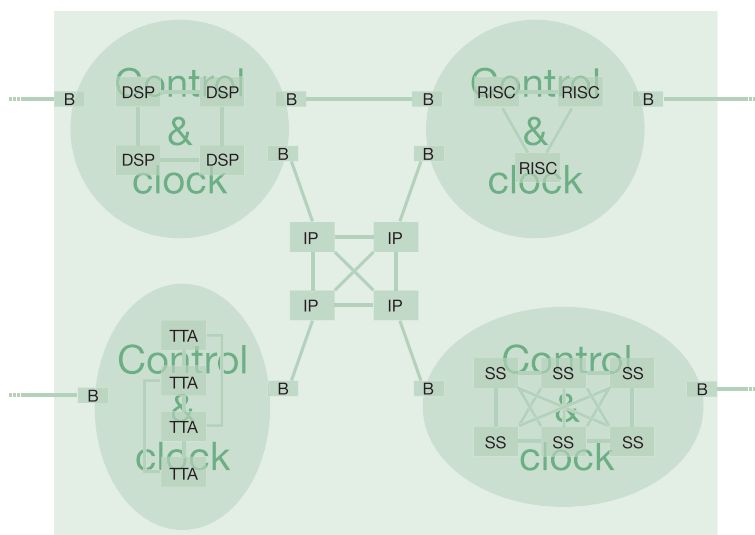
Johdotuksen suunnittelun merkitystä lisäävät entistä tiheämmät piiriteknologiat, suurten digitaalipiirien sisäiset häiriöt ja entisestään kasvavat kellotaajuudet. Suomalaisittain merkittävässä sovelluksessa järjestelmäpiiri toimii usein sovitina yksilön ja globaalisuuden välissä, ja sen ääripäät ovat kehittyneet moniaistikäyttöliittymät ja toisaalla laajakaistainen tiedonsiirto.

Keskittyminen tyyppisovelluksiin mahdollistaa kehitetyn tietämyksen, menetelmien ja arkkitehtuurien uudelleenkäytön. Korkealla tasolla suunnittelun pohjana ovat virtuaaliarkkitehtuurit, joista siirrytään asteittain kohti fyysistä toteutusta. Uusien haasteiden edessä on keskeistä kehittää kansallisen osaamisen tasoa ja infrastruktuuria. Tähän kuuluu väistämättä myös monitieteinen

yhteistyö ja tutkimuksen kansainvälistyminen.

Alan tutkimusta tehdään Tampereen teknillisen korkeakoulun Digitaali- ja tietokonetekniikan laitoksella useissa projekteissa. Suomen Akatemian rahoittamat GIGATRAN (Flexible On-Chip Communication and Interconnect Schemes for Gigatransistor System-on-Chip) ja COMPAKA (Communication Platform Architectures for Gigascale Integration) kehittävät piirinsisäistä kommunikaatiota useiden prosessorien muodostamissa järjestelmissä pääasiassa pakettiverkkoihin perustuen.

Jälkimmäisen projektin sarsarprojektissa, TEKES-rahoitteisessa COMPLAINissa, on osallistujana myös Turun Yliopiston Elektroniikan ja tietokonetekniikan laboratorio. Laboratorion osaamistaustaan kuuluvat muun muassa asynkroninen tiedonsiirto synkronisten alijärjestelmien välillä ja teknologian fysiisiin ominaisuuksiin perus-



Hierarkkinen GALS-arkkitehtuuri (B on puskuri, SS hierarkkinen alijärjestelmä ja IP kommunikaatio-IP).

A hierarchical GALS-architecture. (B is buffer, SS synchronous subsystem and IP communication-IP).

tuva suorituskykymallinnus. TTKK:n ryhmän osaamisalue on puolestaan DSP- ja RISC-prosessoriarkkitehtuurit ja niihin perustuvat digitaalisen tiedonsiirron järjestelmät.

COMPLAIN-COMPAKA-projektipari on osa suomalais-ruotsalaista EXSITE-ohjelmaa ja ne toimivat yhteistyössä tukholmalaisen KTH:n kanssa. Kaikki kolme yliopistoa ovat myös mukana EU:n rahoittamassa SoC-Mobinet-projektissa (System-on-Chip for Mobile Internet), jossa tutkimustyön lisäksi kehitetään alan opetusta ja täydennyskoulutusta eurooppalaisena yhteistyönä. Suomalaisilla yrityksillä on avoin valtakirja päästä osalliseksi projektiin osarahoittamalla loppu- tai väitöstutkimusta.

TTKK:lta on lisäksi kaksi tekniikan tohtoria post-doc-tutkijana maailmankuulussa Berkeley Wireless Research Centerissä (BWRC) Kaliforniassa ja korkeakoululla on kahdenvälistä yhteistyötä monien eurooppalaisten alan huippuyliopistojen kanssa.

Eräs Suomen SoC-alan näkyvimpiä aktiviteetteja on vuosittainen SoC-seminaari Tampereella, jossa kutsuttuina luennoijina ovat maailmanluokan asiantuntijat. TTKK ja Turun Yliopisto ovat pääroolissa myös uudessa valtakunnallisessa TELESOC-tutkijakoulussa (Telecommunication System-on-Chip Integration), jossa osallistujina on yhteensä 11 laitosta ympäri Suomen. Nämä kaikki ovat osa infrastruktuuria, joka tarvitaan Suomen SoC-osaamisen nostamiseksi alan kansainväliseen kärkeen.

Suunnittelun uudet painopisteet

Järjestelmäpiirin suunnittelussa on jo pitkään pyritty nopeutamaan suunnittelua siirtymällä käyttämään valmiita IP-lohkoja. Niiden käytössä on edelleen useita ainakin osittain ratkaisemattomia ongelmia esimerkiksi lohkojen varmentamisessa ja liittämässä osaksi suurempaa järjestelmää. Suurten järjestelmäpiirin onnistuneessa toteutuksessa ei kuitenkaan ole enää kysymys pelkästään sopivien IP-lohkojen valitsemisesta ja niiden rajapintojen sekä toimintojen sovittamisesta. Ennenkaikkea kysymys on lohkojen välisen kommunikaation varmentamisesta ottaen huomioon suunnitteluun vaikuttavat uudet, piirin fyysisistä ominaisuuksista johtuvat ongelmat.

Sovellusten määrän dramaattinen kasvu sekä tiukentuneet vaatimukset järjestelmän tehonkulutukselle, nopeudelle ja pinta-alalle asettavat todellisen haasteen järjestelmäsuunnittelijalle. Markkinoiden hektisyys taas asettaa suunnittelun kesto- ja rajat: moni iteraatiokierroksiin ei ole enää varaa. Tavoite onkin suunnitella järjestelmä toimivaksi mahdollisimman korkealla hierarkiatasolla. Tämän jälkeen suunnittelija valitsee vaatimukset täyttävät IP-lohkot, jotta tarvittava toiminnallisuus ja tiedonsiirto saadaan toteutettua.

Tulevaisuuden järjestelmäpiireissä koko piirille yhteinen kello ei enää ole sopivin ratkaisu, vaan on siirryttävä piiriarkkitehtuureihin, jotka toimivat synkronisesti ainoastaan paikallisten IP-lohkojen tai alijärjestelmien

klustereissa. Piirin yleinen tiedonsiirto perustuu asynkronisiin siirtoprotokollisiin, joihin kuuluu olennaisena osana tiedon siirtäminen paikasta toiseen paketteina sekä pakettien puskurointi synkronisten ja asynkronisten osien rajapinnassa.

Alijärjestelmien sisäisellä toiminnalla tai rakenteella ei ole suurta merkitystä kunhan ne on valittu sopivasti kyseistä sovellusta ajatellen. Näin yksittäinen alilohko voi pohjautua esimerkiksi prosessoriverkkoihin, DSP-, RISC- tai TTA-prosessoreihin tai voi muodostaa oman alijärjestelmistä koostuvan kokonaisuutensa alemmalla hierarkiatasolla. Yksittäinen alijärjestelmä voi yhä useammin muodostaa oman moniprosessoriympäristönsä, jossa on useita suorittimia ja joka käsittää laitteiston lisäksi myös siihen kuuluvan ohjelmiston.

Arkkitehtuureille on myös ominaista hierarkisuus: järjestelmässä on vähintään kaksi hierarkiatasoa, joista alemmalla toimitaan yhteisen kontrollin alaisuudessa mahdollisesti yhteisen kellon ohjaamana, kun taas ylemmällä hierarkiatasolla ei ole yhteistä kontrollia tai synkronointia. Tällaisesta järjestelmästä käytetään nimitystä Globally Asynchronous, Locally Synchronous (GALS). Paikallinen kommunikaation kontrolli on tarpeellinen asynkronisessa tapauksessa, koska siitä tarvitaan fyysisten ominaisuuksien hallintaa. On tarpeen esimerkiksi estää useiden asynkronisten siirtopurskeiden tapahtuminen piiriin tiettyssä osassa yhtäaikaan, koska tämä aiheuttaisi häiriöpiikin tehonsyöttöön. Riippuen lohkojen klusteroinnista ja niiden kontrolloinnista, järjestelmässä voi olla myös huomattavasti useampia hierarkiatasoja.

Keskeinen haaste paikallisista synkronisista yksiköistä koostuvan asynkronisen GALS-järjestelmän toteuttamisessa on kommunikaatio-IP-lohkojen suunnittelu. Näiden liitäntäkomponenttien tehtävänä on siirtää dataa asynkronisesti synkroniselta yksiköltä toiselle. Kyseisten lohkojen tulee olla luotettavia ja suorituskykyisiä erilaisissa ympäristöissä, joissa suorituskykyvaatimukset sekä siirtonopeudet voivat vaihdella suuresti. Tärkeä ominaisuus on myös lohkojen modulaarisuus ja siirrettävyys uusien teknologioihin mahdollisimman pienellä suunnittelupanoksella. Tästä syystä kommunikaatio-IP on rakennettava siten että toiminta on mahdollisimman riippumatonta piiriele-

menttien viiveistä.

Asynkroninen kommunikaatiokanava, jonka avulla data siirretään, perustuu itseajoittuviin (self-timed) kätelysignaaleihin. Kommunikaatiokanavassa datan absoluuttisella siirtoviiveellä ja synkronisten yksiköiden nopeuseroilla ei ole toiminnan kannalta merkitystä, sillä siirtoprosessi on täysin kontrolloitu kätelysignaalien avulla. Johtojen reitityksen helpottamiseksi on kanavissa pyrittävä mahdollisimman pieneen signaalien lukumäärään. Tästä syystä on kiinnitettävä erityistä huomiota käytettävään signaalintipokollaan sekä tehokkaan data- ja kontrollilinjojen multipleksaukseen. Kommunikaatiokanavat systeemin eri väylissä voidaan toteuttaa eri tavoilla, käyttäen hyväksi esimerkiksi perinteistä tai differentiaalista jännitemuotoista tai vaihtoehtoisesti virtamuotoista signaalointia, riippuen siirtoetäisyydestä sekä tarvittavasta siirtonopeudesta.

Asynkroninen rakenne mahdollistaa eri väylissä tapahtuvan datasiirron lomituksen perustuvan tehonkulutuksen piikkien globaalin kontrolloinnin. Tällöin saavutetaan tasaisempi tehonkulutuksen jakauma, josta puuttuvat korkeat piikit. Näin saadaan pienennettyä piiriin kallisarvoista pinta-alaa, sillä piiriin sisäiset nopeita virranvaihteluita suodattavat kondensaattorit, jotka perinteisesti vaativat merkittävän osan piiriin pinta-alasta, voidaan tehdä pienemmiksi. Kontrollointi toteutetaan siten, että kullekin kommunikaatio-IP-lohkoille jaetaan asynkronisesti eriaikaisia datansiirtoikkunoita, joiden sisällä kunkin yksikön datansiirron tulee tapahtua. Datansiirtoikkunoiden jakelutaajuutta eri yksiköille voidaan priorisoida niiden tarvitseman kaistanleveyden perusteella.

Tulevaisuuden integroiduissa digitaalijärjestelmissä fyysikaalinen ja sähköinen mallintaminen tulevat erittäin tärkeiksi. Yksittäisten transistorien ja loogisten elementtien tehonkulutus ja viive pienenevät teknologian kehittyessä, kun taas johtimien, etenkin pitkien, kohdalla kehitys on päinvastainen. Myös tämä pakottaa suuntaamaan suunnittelun painopistettä yhä enemmän logiikan ja toiminnallisten lohkojen suunnittelusta johdinvierokoston ja tiedonsiirtoon käytettävien elementtien suunnitteluun.

Suunnittelun uudet painopisteet vaativat myös suunnittelumenetelmien kehittämistä ja painottamista uuteen suuntaan.

Tulevaisuuden piirisuunnittelua voidaan verrata perinteiseen tietoliikennejärjestelmien suunnitteluun. Tietoliikennejärjestelmien suunnittelusta tutun OSI-mallin mukaisesti suurten järjestelmäpiirien globaali toiminta voidaan jakaa useisiin kerroksiin. Etenkin OSI-mallin kerrokset 1 (physical), 2 (data link) ja 3 (network) ovat laitteistosuunnittelijan kannalta olennaisia. Näiden kerrosten määrittelyt kattavat alijärjestelmien välisessä kommunikaatiossa tietovuon kytkennän ja reitityksen piirin läpi (network-kerros), tiedon esitystavan kahden kommunikaatiosolmun välillä (data link-kerros) sekä kommunikaatiosolmujen välisten fyysisten kytkentöjen signaloinnin ja muiden ominaisuuksien suunnittelun (physical-kerros).

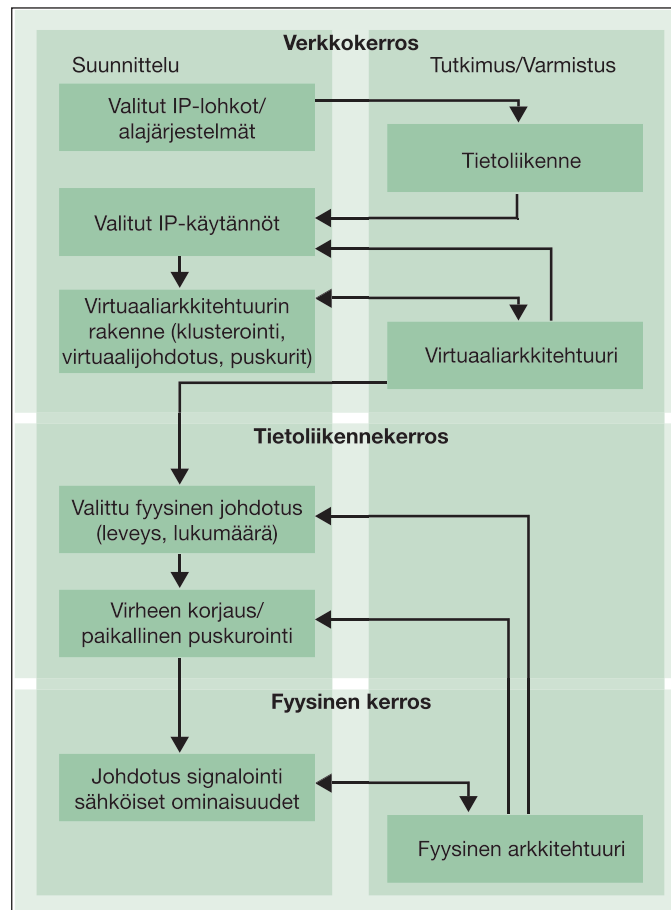
Järjestelmäpiirin suunnitteluvuon

Suurten järjestelmäpiirien suunnittelu alkaa sopivien alilohkojen valinnasta. Lohkot voivat olla peräisin omista aiemmista suunnitteluprojekteista tai ne voidaan ostaa oman yrityksen muilta suunnitteluryhmiltä tai täysin ulkopuolisilta valmistajilta.

Jotta suunnitteluprosessi olisi skaalautuva ja kaikilta osiltaan uudelleenkäytettävä, myös piirin kommunikaatio-osa tulee koota aiemmin suunnitelluista ja varmennetuista IP-lohkoista, kommunikaatio-IP:stä. Tämä tarkoittaa, että myös kommunikaatorakenteet täytyy toteuttaa noudattaen samoja uudelleenkäytön periaatteita, joita tällä hetkellä yritetään soveltaa toiminnallisten IP-lohkojen toteutuksessa.

Piirin arkkitehtuuri suunnitellaan kahdessa vaiheessa. Ensimmäisessä vaiheessa suunnitellaan virtuaaliarkkitehtuuri (platform), jossa valitut alilohkot kytketään toisiinsa virtuaalijohdoilla. Virtuaalijohdot määrittävät ainoastaan piirin kytkentöjen alku- ja loppupisteet ottamatta mitään kantaa fyysisten johdotusten ominaisuuksiin tai reititykseen. Yksittäisen lohkon näkökulmasta tämä on aivan riittävä tieto, sillä lohkolle riittää että tieto siirtyy oikeaan paikkaan oikeaan aikaan.

Virtuaalijohdotuksen lukumäärä määritetään siten, että jokaisen alijärjestelmän kommunikaation tarve voidaan tyydyttää. Suunnittelussa voidaan käyttää apuna aiemmin hyväksi havaittuja virtuaaliarkkitehtuureja ja ainoastaan hienosäätää niitä tarpeen vaatiessa. Virtuaaliarkkitehtuuriin ja virtuaalijohdotuksen tiedon-



Ison järjestelmäpiirin suunnitteluvuon.

A design flow of large System-on-Chips.

siirtokapasiteetista ja lohkojen tiedonsiirron ominaispiirteistä tarvitaan tässä vaiheessa mallit, jotka ovat riittävät, mutta eivät kuitenkaan liian yksityiskohtaiset. Näin on mahdollista arvioida nopeasti kyseisen virtuaaliarkkitehtuuriin käyttäytymisen ja tarpeen vaatiessa muuttaa arkkitehtuuria. Suunnittelun tämä vaihe vastaa järjestelmän mallintamista OSI-mallin kerroksessa 3. Päätettävien asioiden kuuluu alilohkojen ja virtuaalijohdotuksen määrän sekä kytkentöjen lisäksi kommunikaatioverkossa kulkevien pakettien reititys ja hallinta.

Kun virtuaaliarkkitehtuuri on muodostettu ja varmennettu network-tasolla toimivaksi, suunnittelussa siirrytään seuraavaksi alemmalle tasolle. Data link-kerrosta suunniteltaessa päätetään jokaisen virtuaalijohdotuksen ominaisuudet puskuroinnin sekä virheiden havaitsemisen ja korjaamisen osalta. Kasvavien häiriöiden vuoksi ei voida enää olettaa, että tieto siirtyisi perinteiseen tapaan paikasta toiseen ilman tiedonsiirtovirheitä. Tulevaisuudessa joudutaan myös päättämään, onko 100-prosenttisesti virheetön tiedonsiirto tarpeen vai voidaanko hyväksyä

naali tai bittijono saadaan mahdollisimman puhtaana perille vastaanottimeen. Niinikään johtim(i)en fyysinen mitoitus yhdessä toistimien ja, mikäli mahdollista, johtimien lisäämisen ohella edesauttaa signaalin/bitin puhdasta siirtoa kanavassa.

Arkkitehtuurien fyysikaalinen mallintaminen

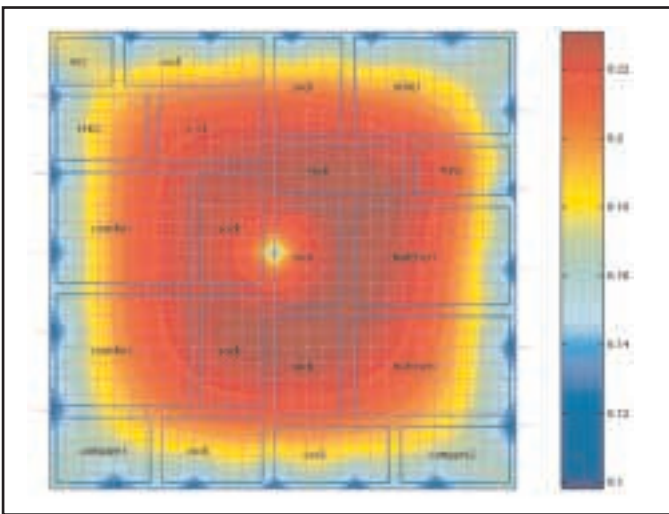
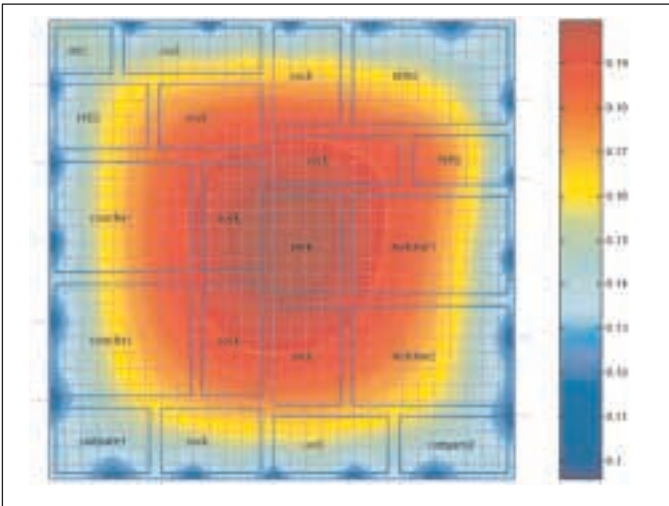
Arkkitehtuurien fyysikaalisella mallintamisella pyritään tarkasti selvittämään, kykeneekö kyseinen arkkitehtuuri suoriutuun tehtävästä sovelluksen vaatimassa ajassa täyttämällä muutkin mahdolliset kriteerit. Jos tämä onnistuu, voidaan arkkitehtuurin sisäisiä rakennelohkoja arvioida ja, ei suunnitella hienojakoisemmin.

Jos esimerkiksi arkkitehtuurissa käytettävien väylien määrä tai väylän leveys bitteinä ei ole täysin kiinnitetty, voidaan tehdä suorituskykyvertailua erilaisilla sanan leveyskilla tai eri määrillä väyliä. Ensimmäinen on siis selvittävä, kykeneekö kyseinen arkkitehtuuri ylipäättään suorittamaan annetun tehtävän tai algoritmin, ja sen jälkeen optimoidaan toiminta arkkitehtuurin sisällä mahdollisimman tehokkaaksi.

Fyysikaalisen mallin pääelementit ovat arkkitehtuurin vaatima pinta-ala, viive ja tehonkulutus. Lisäksi yhä tärkeämpiä suunnittelukriteereitä tulevaisuuden integroiduissa digitaali-järjestelmissä ovat kohina, sekä sisäinen että ulkoinen, ja kellon signaalin viive eri kohdissa piiriä. Nämä kaksi tekijää vaikuttavat merkittävästi arkkitehtuurisuunnitteluun.

Kutistuvat metallijohdot piirillä aiheuttavat sen, että kellon signaali saattaa tulla hyvinkin eri aikaan piiriin eri kohtiin, ja mikäli kyseiset piiriin osat on tahdistettu yhteisen kellon mukaan, alkaa esiintyä ajoituskellon ongelmia. Tehonsyötössä esiintyvä kohina taas saa aikaan toiminnallisten elementtien, kuten loogisten porttien tai yksittäisten transistorien kytketymisen eri aikoihin, mikä aiheuttaa toiminnallisia virheitä.

Jos kohinan taso piirillä ei saa ylittää tiettyä arvoa, täytyy kohinalähde sulkea fyysisesti pois häiriöherkistä piiriin osista (esim. analogiaosat). Jos kohina aiheutuu toiminnasta itsestään, täytyy arkkitehtuuri suunnitella uudelleen. Esimerkiksi tehonsyötön johtimien leveyttä kasvatettaessa tai niiden määrää lisäämällä saadaan jännitteenvaihtelu tehonsyötössä pienennettyä. Tämä vaikuttaa kuitenkin varsinaisten signaalien joh-



Tehonsyötön kohinajännitteen arvo eri kohdissa piiriä eräessä prosessoriarkkitehtuurissa, jolla on toteutettu ATM AIS -solun prosessointia. Punainen väri kuvaa korkeaa kohinajännitteen arvoa ja sininen matalaa arvoa. Ensimmäinen kuva on laskettu wire-bonded PGA-pakkaukselle, jälkimmäinen C4-bonded MLC PGA-pakkaukselle.

The level of random noise caused by power supply at different parts of a processor architecture, that has been used to implement processing in a ATM AIS cell. Red color signifies high level of random noise and blue color signifies low level. The first illustration is calculated for a wire-bonded PGA package and the latter for C4 bonded MLC PGA package.

dottamiseen tarjolla olevaan kapasiteettiin, ja monesti joudutaankin piirin pinta-alaa johdotuksellisista syistä kasvattamaan. Tulevaisuudessa teknologian kehittyessä voidaan todennäköisesti käyttää kokonaisia metallitasoja tehon jakeluun ja tämä pienentää tehonsyöttöverkostolle aiheutuvia ongelmia.

Eräs ongelma koostettaessa aikaisemmin mainittua virtuaaliarkkitehtuuria on siihen käytettävien IP-lohkojen kirjavuus. Eri IP-lohkojen toiminta on varmennettu tietyssä ympäristössä, eikä ole lainkaan varmaa, toimiiko IP-lohko suunniteltavassa virtuaaliarkkitehtuurissa. Jos esimerkiksi kahden IP-lohkon tehonkulutukset ovat suuria ja nämä lohkot kommunikoivat järjestelmätasolla sa-

manaikaisesti, aiheutuu tehonsyöttöverkosto on voimakastakin jännitevaihtelua verkoston yrittäessä antaa tehoa kummallekin lohkolle samaan aikaan. Tämä vaihtelu saattaa aiheuttaa virhetoimintoja muissa IP-lohkoissa.

Arkkitehtuuriratkaisun vaatima pinta-ala riippuu piirillä tarvittavan muistin määrästä, kohinalle sallitusta maksimiarvosta, sisäisen kommunikaation määrästä (väylien määrä/leveys) sekä myös ulkoisen kommunikaation määrästä (I/O-transistoreiden ja liityntänastojen määrä). Käytettäessä CAD-työkaluja työkalun johdotustehokkuus ja eri toiminnallisten lohkojen geometrinen yhteensopivuus vaikuttavat pinta-alaan. Myös tiedon siirtäminen pitkiä matko-

ja piirin osasta toiseen vaikuttaa pinta-alaan. Signaaleja täytyy "puhdistaa" toistimilla, jotka lisäävät pinta-alaa.

Arkkitehtuuriratkaisussa esiintyvät viiveet riippuvat toiminnallisten lohkojen määrästä, niiden keskinäisistä kommunikaatioriippuvuuksista, sovelluksen raskaudesta (tarvitaanko ulkoista kapasiteettia) jne. Jos käytetään valmiita IP-lohkoja, lohkojen sisäinen viive ilmoitetaan joko tietyille teknologiakirjastolle tai sitten ekvivalentteina porttiviiveinä, jolloin absoluuttinen viive riippuu teknologian valinnasta.

Arkkitehtuuriratkaisun tehonkulutus riippuu toiminnallisten lohkojen sisältämien elementtien määrästä, käyttöjännitteen suuruudesta ja käytettävästä logiikasta, kello-signaalin taajuudesta sekä kytkentäaktiivisuudesta. Kytkentäaktiivisuus kuvaa sitä, kuinka moni elementti toiminnallisissa lohkoissa kytkeytyy samanaikaisesti kello-signaalin saapuessa. Koska kello-taajuudet kasvavat ja toiminnallisten lohkojen koko ja määrä kasvavat, äkilliset suuret teho-piikit alkavat rasittaa tehonsyöttöä ja häiritä herkkiä analogia-osia, jos näitä on integroitu samalle piirille.

Ratkaisuksi tähän ongelmaan sen lisäksi, että käytetään erillisiä käyttöjännitteitä digitaali- ja analogia-osille, on esitetty GALS-ratkaisua. Tämä sallii myös eritaajuisten kello-signaalien käytön eri lohkoissa, mutta ei silti välttämättä kokonaan estä äkillisten tehopiikkien muodostumista. Kaksi eritaajuista kello-signaaliakin saattavat välillä lomittua toistensa kanssa ja aiheuttaa kahden eri lohkon elementeissä tilojen (bittien) muutoksia samanaikaisesti. Tilanne pahenee, jos vielä lohkojen välinen asynkroninen kättelysignaali avaa kommunikaatioyhteyden näiden lohkojen välille, jolloin kommunikaatiosta huolehtivat ajurit lisäävät vielä omalla osuudellaan kokonaistehonkulutusta.

Edellämainitut asiat ovat vahvasti sidoksissa toisiinsa, ja siksi mahdollisimman monipuolinen arkkitehtuurikartointi jo suunnittelun alkuvaiheessa on tärkeää. Järjestelmäpiirin suunnittelu vaatii kaikissa vaiheissa entistä suurempien kokonaisuusien hahmottamista sekä entistä useampien tekijöiden huomioimista ja hallintaa. Järjestelmäpiiri ei ole piiriteknologia tai -menetelmä vaan elämäntapa. ●

Aiheesta enemmän

TTKK:n tutkimusryhmä:
www.cs.tut.fi/~nurmi/

Turun yliopiston tutkimusryhmä: www.physics.utu.fi/ett/

TELESOC tutkijakoulu:
<http://edu.cs.tut.fi/telesoc/>

SoC-seminaari:
www.cs.tut.fi/soc/

SoC-Mobinet:
www.physics.utu.fi/ett/Projects/SocMobinet

BWRC: <http://bwrc.eecs.berkeley.edu/>

Taustat

Kirjoittajat: DI Ilkka Saastamoinen työskentelee TTKK:n Digitaali- ja tietokonetekniikan laitoksella tutkijana "DSP and Digital Communication Systems On Chip" -tutkimusryhmässä.

TkT Jari Nurmi on Tampereen teknillisen korkeakoulun digitaali- ja tietokonetekniikan laitoksen professori ja "DSP and Digital Communication Systems On Chip" -tutkimusryhmän johtaja.

DI Tero Nurmi työskentelee Turun yliopiston elektroniikan ja tietotekniikan laboratoriossa ja tietotekniikan laboratorion johtajana.

TkT Jouni Isoaho on Turun yliopiston elektroniikan ja tietotekniikan laboratorion professori ja johtaja.

FM Pasi Liljeberg työskentelee Turun yliopiston elektroniikan ja tietotekniikan laboratoriossa ja tietotekniikan laboratorion johtajana.

FT Juha Plosila on digitaalitekniikan yliassistentti Turun yliopiston elektroniikan ja tietotekniikan laboratoriossa.

DI Piia Simonen työskentelee TTKK:n Digitaali- ja tietokonetekniikan laitoksella tutkijana "DSP and Digital Communication Systems On Chip" -tutkimusryhmässä.

Yhteystiedot:
jari.nurmi@tut.fi

Tutkimus: COMPLAIN, Communication Platform Architectures for Gigascale Integration

Yhteistyössä: Nokia Mobile Phones, Nokia Networks, Nokia Multimedia Terminals, Telemate, Elektorbit, Nokia Research Center

Teknologiaohjelma: EXSITE