



New solutions for SOC design

To control the growing complexity of the System On Chip (SOC) designs, new tools are needed. Platform Based Design (PBD) is seen as a solution and numerous companies have introduced different types of platforms.

Using of platforms reduces costs in design and manufacturing of chips because multiple applications can use same chip and therefore production volumes can be higher. However the platform based design requires new tools and methodologies. As part of the System Chip Inc. Finland project an Information Highway Interface Platform (IHIP) chip has been designed. It is used as an example of platforms and a vehicle for testing new tools and methodologies.

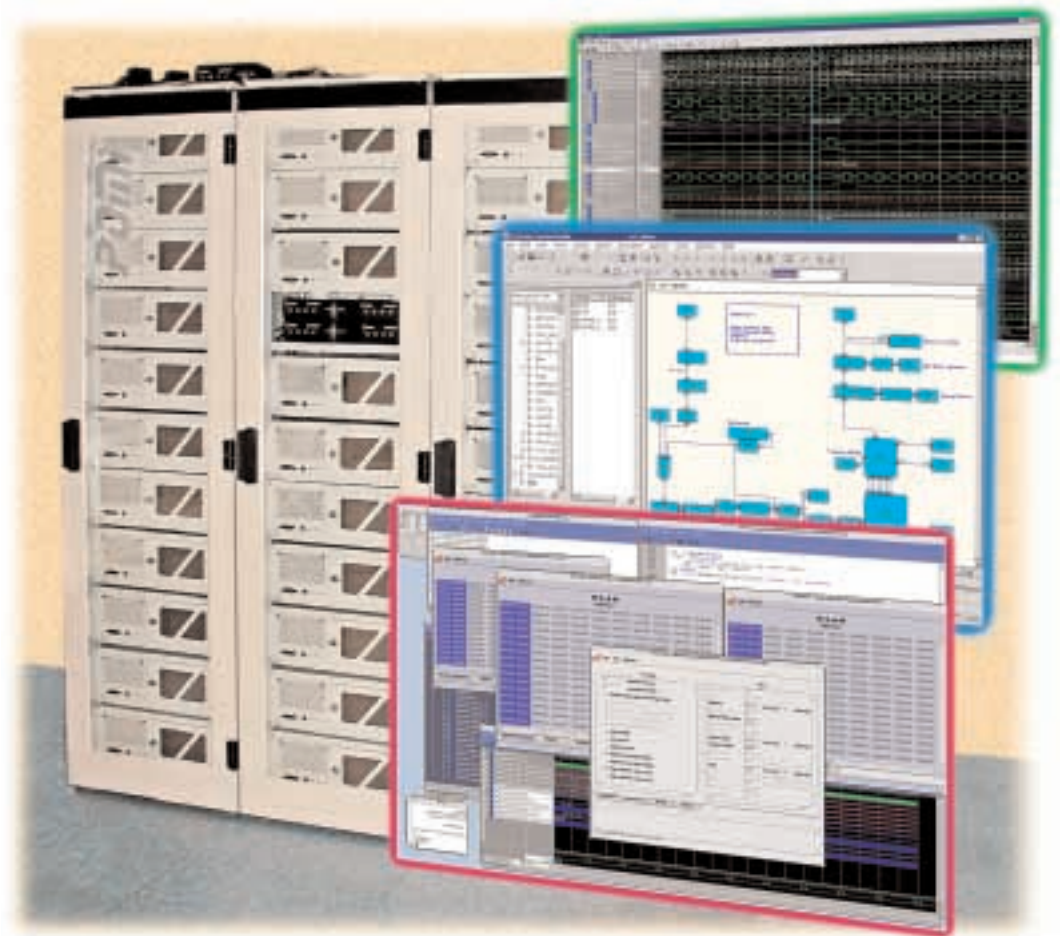
The IHIP consists of a RISC processor, on-chip system bus, configurable shared memory DSP multiprocessor, large on-chip memory block and application specific hardware accelerators. The performance of the configurable shared memory DSP engine is analysed under different digital signal processing load types and it is founded to give a good basis of designing SOC's for multimedia and Internet communications.

The applications used in designing and evaluating the performance of the chip are HiperLAN/2 modem and DVD quality MPEG-2 video decoding. The consortium in the project contains the most significant Finnish companies involved in different stages of virtual component based design.

The SciFi project is a three year effort ending in march 2002. Work is done at VTT Electronics where contact person is project manager Juha-Pekka Soininen (juha-pekka.soininen@vtt.fi). Project is a part of ETX program.

Uudelleenkäyttö komponenteista arkkitehtuureihin

Järjestelmäpiirejä nopeammin ja edullisemmin



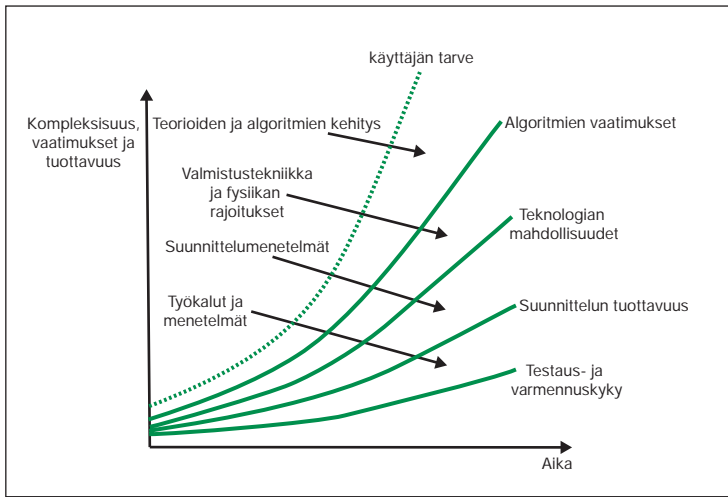
Järjestelmäpiirien suunnittelussa uudelleenkäyttö on ulotettava virtuaalikomponenteista arkkitehtuuritasolle. Monen prosessorin järjestelmäpiireissä pelkkä lohkojen ostaminen valmiina kokonaisuuksina ei riitä, vaan myös lohkojen välisten liitäntöjen ja niiden muodostaman kokonaisuuden täytyy jatkossa olla etukäteen suunniteltu ja testattu paketti.

Piikapasiteetin hyödyntäminen SOC-piirien avulla edellyttää uudelleenkäyttöä, joka ulottuu komponenteista arkkitehtuureihin. Vain hyvin

harvoin tuotteiden volyymit ovat sellaisia, että suunnittelu kannattaa aloittaa puhtaalta pöydältä. Yhä useammin riittää olemassa olevan arkkitehtuuri-

ratkaisun tai konfiguroitavan piirin hyödyntäminen.

Eri tasoiset tuotealustat ovat keino tehostaa tuotekehitystä. Kaupallinen tarjonta on vielä kirjavaa, koska kehitys on nopeaa ja tarpeet hajanaisia. Tehokas hyödyntäminen edellyttää osamista piirisuunnittelusta järjestelmäsuunnitteluun. Alustan valinta tarkoittaa sitoutumista työkaluihin, suunnitteluprosesseihin ja tuotearkkitehtuuriin. Tehdyt päätökset vaikuttavat yrityksen toimintaan koko tuotteen tai tuoteperheen elinkaaren ajan.



Kuva 1. Kompleksisuus on on kasvanut ajan myötä eksponentiaalisesti. Suunnittelu ja erityisesti kiky testata ja varmentaa piirejä rajoittaa yhä selvemmin mahdollisuuksia käyttää kaikki teknologian antamat mahdollisuudet. Teknologian mahdollisuudet eivät nekään riitä käyttäjän mielikuvituksen antamiin tarpeisiin tai edes tunnettujen algoritmien toteuttamiseen.

Complexity has been exponentially expanded. Designing and especially ability test and verify circuits are limiting the possibilities to use all of the technological potential. Technological possibilities cannot still provide means to implement needs that users imagines or even to known algorithms.

Yhä monipuolisemmat järjestelmät voidaan sijoittaa yhdelle integroidulle piirille. Esimerkiksi IBM:n tarjoamassa 0,11 mikrometrin piiprosessissa saadaan yhteen piiriin mahtumaan 40 miljoonaa loogista porttia. Tämä riittää jo erittäin laajan järjestelmän toteuttamiseen yhdellä piirillä. Kun porttien ja järjestelmäpiirin (System-On-Chip, SOC) prosessoreiden määrät kasvavat nelioollisesti, lähivuosina tarvitaan kokonaan uudenlainen lähestymistapa piiriin suunnitteluun.

Piirisuunnittelun uusi suunta

Piirien kompleksisuuden kasvua on pyritty hallitsemaan jakamalla suunnittelu- ja varmennustehävät itsenäisesti suoritettaviin osiin. Aiemmin erikseen testatut virtuaalikomponentit on yhdistetty kokonaiseksi järjestelmäksi yhdelle integroidulle järjestelmäpiirille. Kun virtuaalikomponentit ovat uudelleen käytettäviä, niin tarvittava suunnittelu-aika jää huomattavasti lyhyemmäksi kuin jos vastaava piiri suunniteltaisiin kokonaan perinteisin menetelmin.

Vaikka virtuaalikomponenttien ja muiden lohkojen rajapinnat suunniteltaisiin mahdollisimman vähän liikennettä ja signaaleja sisältäviksi niin väylät, lohkojenvälinen johdotus ja kasvava määrä sulautettua muistia vaativat yhä enemmän käytettäviltä EDA-työkaluilta. Valmiiden lohkojen käyttöä tukevat työkalukokonaisuudet ovat olleet edellytys entistä monimutkaisempien järjestelmien suunnittelemiselle ja varmentamiselle taloudellisesti järkevässä ajassa.

Tuotteiden nopea markkinoille saaminen vaatii kuitenkin valmiimpia kokonaisuuksia järjestelmäpiiriin suunnittelun aloituslähtökohdaksi. Järjestel-

mien monimutkaistuessa virheet suunnittelussa maksavat entistä enemmän, sekä suoraan rahana että välillisesti huonompina suorituskykyinä. Lohkojen ja niiden yhteensopivuuden on oltava etukäteen huolella varmennettua. Suunnittelu on ilman virheitäkin niin kallista, että yhden piiriin valmistusmääriä on lisättävä käyttämällä samaa piiriä samantyyppisissä tuotteissa. Tällöin tuotteet on yksilöitävä ohjelmiston avulla.

Joustavuuden saavuttamiseksi sekä suunnittelussa että tuotteen edelleenkehittämisessä järjestelmäpiiriin täytyy olla konfiguroituvia ja ohjelmoitavia. Lisäksi tarvitaan skaalautuvuutta tuotteen sopeuttamiseksi eri markkinasegmenttien vaatimuksiin. Näillä on merkittävä vaikutus myös suunnittelumenetelmiin ja -prosesseihin jotka muuttuvat kovokeskeisistä ohjelmistokeskeisiksi.

Integrointialustat

Myös suunnittelussa saavutetaan etua käyttämällä samaa suunnitteluympäristöä usean samankaltaisen piiriin suunnittelussa. Integrointialusta on järjestelmäpiiriin suunnitteluympäristö, joka on rakennettu joltain tuotesovellusta varten. Ympäristö sisältää arkkitehtuurisen spesifikaation sekä kokoelman etukäteen valittuja IP-lohkoja jotka on suunniteltu toimimaan yhdessä. Integrointialusta toimii perustana tehokkaalle SOC-suunnittelulle. Se tehostaa uudelleenkäyttöä jota tarvitaan järjestelmäpiiriin sovellusten suunnitteluun ja valmistamiseen ja se on usein räätälöity jollekin kuluttajasovellukselle.

Alustapohjainen suunnittelu (Platform Based Design, PBD) on tehokkainta kun se pohjautuu ohjelmistokeskeiseen top-down-ajatukseen. Kun alusta tarjoaa lähtökohdaksi makrota-

son arkkitehtuurin ja huolehtii komponenttien integroinnista, suunnittelijaryhmä voi aluksi keskittyä sovellusohjelmiston suunnitteluun. Tämän jälkeen ohjelmiston vaatimuksiin riittävä toteutus konfiguroidaan alustalla. Tyypillisesti suunnittelu on joko toimintojen lisäämistä tai entisten toimintojen yhdistämistä.

Alustapohjainen suunnittelu keskittyy yhteen tai useampaan standardoituun väyläarkkitehtuuriin. Kovon suunnittelun tuottavuuden paraneminen saadaan aikaan minimoimalla lohkojen tapauskohtainen liityntöjen suunnittelu ja muokkaaminen. Valmiit standardiliitännät sallivat myös valmiiden lohkospesifisten tuotantotestien käytön rinnakkaisesti.

Integrointialustojen luokittelu

Kun lohkopohjainen suunnittelu tai verifiointi suunnitellaan etukäteen muodostuu integrointialusta (Integration Platform). Integrointialustat jaetaan suunnittelu- (Design/IC Integration), arkkitehtuuri- (Architecture/System Integration) ja valmistusaloitukseen (Manufacturing Integration).

Esimerkkinä Sonics tarjoaa Rapid Integration System -tuotetta suunnittelualustaksi. Siihen sisältyvät väyläratkaisut, ohjelmistot ja standardit, joiden avulla lohkot voidaan kasata kokonaiseksi järjestelmäpiiriksi. Suunnittelualusta on yleiskäyttöisin alustaluokka. Tiettyä suunnittelualustaa voidaan käyttää mihin tahansa sovellukseen johon alustan sisältämät prosessorit, muistit ja väylät soveltuvat. Etuna suunnittelualustassa on joustavuus, mutta joustavuuden hintana on tarve verifioida uudestaan jo valmiit IP-lohkot laitetasolla.

VLSI/Philips RSP7+ -ympä-

ristö (Rapid Silicon Prototyping) sisältää arkkitehtuurialustan yleisiä piiriteitä. Arkkitehtuurialusta on suunnattu tiettyyn sovellusalueeseen. Yleensä noin 95 prosenttia käytettävistä komponenteista sisältyy arkkitehtuurialustaan. Näin voidaan nopeasti evaluoida erilaisia toteutusmahdollisuuksia ja valitun toteutuksen suunnittelu on nopeaa. Tuotteet kohdennetaan eri sovelluksien tarpeisiin yhdellä tai muutamilla kovalohkoilla sekä ohjelmistolla. Arkkitehtuurialustoissa ohjelmistokehitystä tuetaan sisällyttämällä siihen reaaliaikainen käyttäjärjestelmä (RTOS), debuggereita, yhteissimulointimalleja, kääntäjiä, ajureita tai prototyyppiympäristöjä.

Triscend CSOC (Configurable SOC) on kommunikaatiotuotteisiin tarkoitettu valmistusalausta, joka sisältää ARM7 prosessorin lisäksi sulautettua muistia sekä FPGA-lohkon. Lisäksi runsas ohjelmisto sisältää kääntäjiä, debuggereita ja useamman valinnaisen reaaliaikakäyttäjärjestelmän prosessorille ja työkalut FPGA-lohkon suunnittelemissiksi ja ohjelmoimiseksi. Valmistusalausta on nopein tapa saada SOC-piiri markkinoille. Siihen kuuluu valmiiksi suunniteltu piikuvio, jota ei ole tarkoitettu muokata. Tuote muodostetaan ohjelmiston ja FPGA-lohkon konfiguroinnin avulla. Täten valmistusalaustalla toteutetut piirit poikkeavat vain hyvin vähän toisistaan. Jotkut valmistusalaustat mahdollistavat piiriin koon optimoinnin suuria sarjoja tilatessa.

Työkalualustat

Järjestelmäpiiriin suunnittelussa tarvitaan myös tehokkaita työkaluja suunnitteluun ja varmentamiseen. Yleensä yllä käsitellyt alustat määrittävät myös käytettävät EDA-työkalut ja tietokonealustat. Työkalujen käyttöön saadaan lisää tehoa entistä nopeampien prosessoreiden lisäksi käyttämällä rinnakkaisia prosessoreja yhtäaikaan.

Yleisempi tapa on käyttää tietokonefarmeja joissa on riittävä määrä tietokoneita usean käyttäjän tarpeisiin. Farmit ovat tehokkaita suuren suunnittelijaryhmän kanssa. Tällöin kaikille suunnittelijoille ja tehtäville käyvät yhtenäiset alustustiedostot, koska farmin kaikki koneet ovat samanlaisia käyttäjälle päin. Myös EDA-ohjelmistojen kalliit lisenssit tulevat tehokkaampaan käyttöön kun työt menevät automaattisesti jonoon suoritettavaksi välittömästi lisenssien vapauduttua. Sunin

Technical Compute Farm -tuoterpehe on esimerkki tietokonefarmista.

Toinen mahdollisuus prosessorin rinnakkaiseen käyttöön on yhden tehtävän hajauttaminen usealle prosessorille, joko saman tietokoneen sisällä tai useamman koneen klusterina. EDA-ohjelmistot ovat pitkän kehityksen tuloksena perinteisiä yksisäikeisiä. Moniprosessointi ei vielä kukaan onnistu suoraan kaupallisilla työkaluilla, koska mikään EDA-ohjelmisto ei ole säikeistyvä, toisin kuin esimerkiksi monet 3D-grafikkaohjelmistot.

IHIP-piiri ja sen suunnitteluvu

SciFi-projektissa järjestelmäpiirin suunnitteluesimerkinä on käytetty IHIP-piiriä (Information Highway Interface Platform). IHIP on arkkitehtuurialusta, ja sen tarkoitus on antaa käytännön kokemusta järjestelmäpiirin suunnittelun käytännön toteutuksesta, sen tuomista eduista ja ongelmista.

Piirin suunnittelun lähtökohdista on käytetty multimedian ja langattoman tiedonsiirron laskenta- ja muistivaatimusten luokittelua ja arviointia. Luokittelun perusteella on päädytty luonnollisimpaan arkkitehtuuriin, jolla kyseisen tyypin laskenta pystytään suorittamaan. Digitaalinen signaalinkäsittely multimediasa ja langattoman internetin yhteydessä on jaoteltu kahteen perustyyppiin, bittivirtapohjaiseen prosessointiin ja lohkopohjaiseen signaalinkäsittelyyn.

Bittivirtapohjainen signaalinkäsittely on kyseessä esimerkiksi useiden tiedonsiirtoprotokollien fyysisen- ja linkkikerroksen tapauksessa. Sille on tyypillistä tulevien datapakettien pieni koko (esimerkiksi yksi OFDM-symboli) ja jaksollisuus. Kokonaiskaistanleveys saattaa silti olla suuri, joka aiheuttaa tiukat vaatimukset paketin prosessointiaikaan. Prosessointitarve on jaettu useisiin vaiheisiin, joissa on yleensä pieni määrä laskentatarvetta. Vaiheiden välillä esiintyy segmentointia ja uudelleenjärjestelyä.

Luonnollisin vaihtoehto toteuttamiseen on liukuhinnoitettu rakenne, jossa prosessointivaiheet ovat allokoitu peräkkäisille prosessointielementeille, joko DSP-prosessoreille tai kovokiihdyttimille. Liukuhinnoitettu rakenne vaatii hyvän synkronointimenetelmän prosessorien ja laskentavaiheiden välillä ja se vaatii muisti- ja järjestelmäväyliltä suurta ylläpidettävää

kaistanleveyttä.

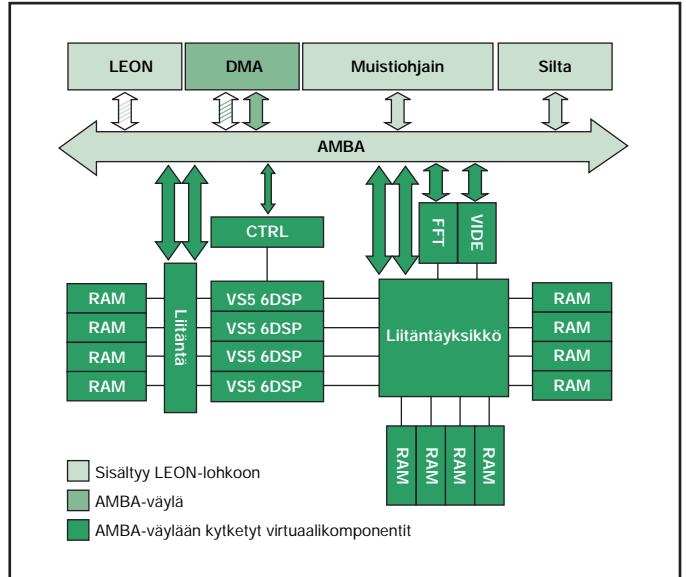
Lohkopohjaisessa signaalinkäsittelyssä tulevien datapaketien koko on suuri ja niitä tulee harvakseltaan. Yksi paketti voi olla esimerkiksi videokoodekissa kokonainen pakkaamaton kehys. Yleensä prosessointiasteita on vähemmän ja ne ovat hitaampia ja monimutkaisempia. Lohkopohjaiseen signaalinkäsittelyyn suunnatuilla arkkitehtuureilla pitää olla suuri muistiavaruus puskuroinnin takia sekä suuri järjestelmä- ja muisti-väylien hetkellinen siirtonopeus johtuen puskurimuotoisesta datan siirrossa. Arkkitehtuurit joilla lohkopohjaista signaalinkäsittelyä tehdään ovat tyypillisesti rinnakkaisprosessoreita.

Internetin IP-pohjaisessa tiedonsiirrossa langattomassa lähiverkossa esiintyy molemmille prosessointityypeille tyypillisiä ominaisuuksia. Fyysinen- ja linkkikerros on tyypillistä bittivirtapohjaista signaalinkäsittelyä, mutta ylempien tiedonsiirto-kerrokset (TCP/IP-paketit) ovat enemmän lohkopohjaista signaalinkäsittelyä vaativaa. IHIP-piirin suunnittelun tavoitteena on ollut kehittää DSP-moniprosessori, joka konfiguroitavan jaetun muistin avulla sopii tehokkaasti sekä bittivirta- että lohkopohjaiseen digitaaliseen signaalinkäsittelyyn.

Esimerkkisovelluksina IHIP-piirin mitoittamisessa on käytetty ADSL- (Asymmetric Digital Subscriber Line) ja HiperLAN/2-modeemeja sekä DVD-tasoisista MPEG2-dekooderia. Esimerkkisovellukset ovat valittu siltä perusteella, että piirin laskentakapasiteetti tarjoaisi mahdollisuuden saman piirin käyttämiseen DVD-tasoisien videon purkamiseen sekä langalliseen tai langattomaan Internet-yhteyteen.

IHIP-arkkitehtuurialusta

IHIP-piiri koostuu toimintaa ohjaavasta RISC-prosessorista, DMA-yksiköstä, järjestelmäväylästä, neljän DSP-prosessorin ja kahden kovokiihdyttimen ja konfiguroitavan jaetun muistin muodostamasta DSP-moottorista sekä suuresta piirille sisällytetyistä muistilohkosta (kuva 2). DSP-moottori koostuu neljästä suomalaisen VLSI Solutionin VS56000 DSP-ytimeistä. DSP-moottorissa on lisäksi kaksi kovolohkoa, toinen Viterbi-dekoodaukseen ja toinen FFT:n (Fast Fourier Transform) ja IFFT:n (Inverse FFT) laskemiseen. Käyttämällä kovolohkoja ohjelmitavien prosessorien lisäksi saadaan piirin koko-



Kuva 2. SciFi-projektissa kehitetään IP-lohkoihin perustuvaa mukautuvaa SOC-laskenta-alustakonseptia, joka mahdollistaa erityyppisten sovellusten suorittamisen.

IP-block based adaptive SOC processing platform is being developed in SciFi project that enables processing of different types of applications.

naislaskentakapasiteettiä tietyissä tilanteissa nostettua huomattavasti.

IHIP-piiri on tähdätty tulevaisuuden CMOS ASIC -prosesseille, sen toteuttaminen nykyisillä 0,13 mikrometrin prosesseilla voisi olla juuri ja juuri mahdollista. Kellotaajuus piirille on suunniteltu olevan 200 megahertsiä, ja sen ohjelmitava suorituskäyky kovokiihdyttimiä laskematta on noin miljardi operaatiota sekunnissa. Suunnittelu on tehty siten, että käytettävät komponentit ovat olleet mahdollisuuksien mukaan jo olemassa olevia, kaupallisia virtuaalikomponentteja ja suunnittelussa avainasemassa on ollut komponenttien valinta ja niiden yhteenliittäminen. Tällöin suunnittelun painopiste on voitu pitää riittävän tehokkaan arkkitehtuurin suunnittelussa ja valittujen sovellusten toiminnallisuuden jakamisessa.

IHIP-piirin DSP-moottori

Piirin ohjauksesta vastaa Leon RISC-prosessori. Sillä on vastuu ajettavien ohjelmien lataamisesta DSP-prosessoreille, DSP-moottorin jaetun muistin konfiguroinnista, DSP-prosessorien pysäyttämisestä ja käynnistimisestä, keskeytysten generoinnista ja DMA-yksikön käskyttämisestä. Sillä on myös pääsy DSP-moottorin jaettuun muistiin. Valtaosan piirillä ajettavista sovelluksista pyörittää neljän DSP-prosessoriytimen, FFT- ja Viterbi-kovolohkojen sekä konfiguroitavan jaetun

muistin muodostama DSP-moottori.

Jaettu muisti koostuu kahdeksasta 64 kilosan muistipankista ja muistipankit on jaettu kahteen DSP-ytimien kahta erillistä datamuistiväylää varten. Jokaisen DSP-ytimen kummankin muistiväylän 64 kilosan muistiavaruus ja jaetun muistin muistipankkien muistiavaruus on jaettu 16 kilosan segmentteihin, joiden yhdisteleminen on konfiguroitavissa (kuva 3).

Mikä tahansa DSP-ytimen loogisista muistisegmentistä voidaan konfiguroida vastamaan mitä tahansa jaetun muistin muistipankin fyysisistä segmentteistä. Sillä rajataan prosessorien pääsyä muistiväylille pienemmissä osissa kuin perinteisillä ratkaisuilla. Näin yhtäaikaisten prosessorien odotustiloja aiheuttavien muistiviittausten määrä voidaan minimoida. Lisäksi konfiguraatiomuutokset mahdollistavat nopeat muistisiirrot prosessorien välillä ja niillä voidaan keinotekoisesti lisätä jokaisen DSP-ytimen suhteellisen pientä muistiavaruutta.

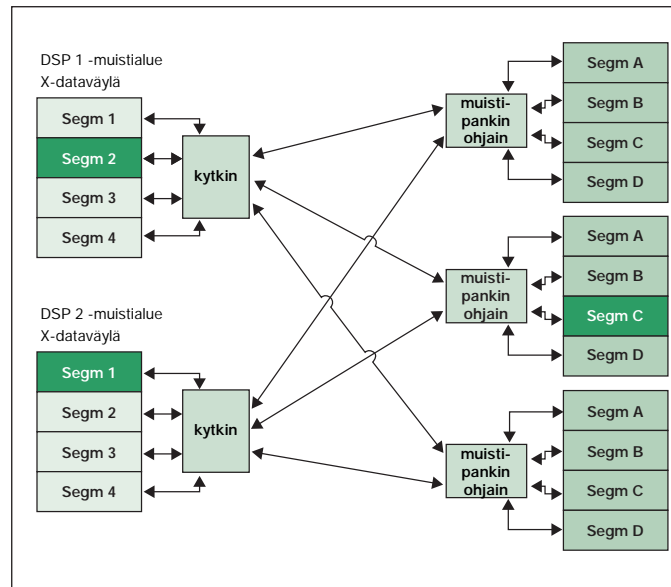
Konfigurointia varten muistiohjaimessa on kaksi rekisteripankkia, joista toinen sisältää käytössä olevan konfiguraation. Toinen rekisteripankki voidaan ladata häiritsemättä DSP-moottorin toimintaa järjestelmäväylän kautta Leon RISC-prosessorin toimesta tai DMA-yksikköä käyttämällä. Konfiguraatiomuutos vie vain yhden kellojakson.

Konfiguroitavuus jaetussa muistissa mahdollistaa DSP-moottorin ja kovokiihdyttimien

toimimisen luonnollisesti aiemmin esiteltyjen signaalinkäsittelytyyppien ajoalustana. Peruskonfiguraatit jaetulle muistille on liukuhihnoitettu konfiguraatio, rinnakkainen konfiguraatio ja itsenäinen konfiguraatio. Liukuhihnaksi konfiguroitu DSP-moottori toimii siten, että osa DSP-ytimen muistiavaruudesta on konfiguroitu I/O-puskuriksi edelliseen tai seuraavaan liukuhihnan vaiheeseen.

Kovokiihdyttimet ovat lisäksi kytketty jaettuun muistiin siten, että ne voidaan liittää saumattomaksi osaksi liukuhihnaa. HyperLAN/2-modeemin toiminnallisuus muodostaa kaksi liukuhihnoitettua rakennetta, toisen vastaanottimessa ja toisen lähettimessä. Rinnakkaisessa konfiguraatiossa jokainen DSP-ydin käyttää ainakin osittain yhteistä muistipankkia muiden DSP-ydinten kanssa.

Konfiguraatio muistuttaa perinteistä MIMD-moniprosessorijärjestelmää (multiple instruction, multiple data) ristiin-kytketyllä muistiohjaimella. MPEG2-videon dekodauksessa osa prosessointivaiheista on rinnakkaisesti kahdeksi peräkkäiseksi rinnakkaiseksi konfiguraatioksi. Itsenäisessä konfigu-



Kuva 3. Muistin konfiguroimismahdollisuudet IHIP-piirissä. Esimerkissä DSP 1:n toinen muistisegmentti ja DSP 2:n ensimmäinen muistisegmentti on konfiguroitu osoittamaan samaan fyysiseen jaetun muistin segmenttiin.

Configuration of shared memory in IHIP chip. In this example, second memory segment from DSP 1 and first memory segment of DSP 2 are mapped to a same physical memory segment in the shared memory.

raatiossa jokainen DSP-ydin toimii itsenäisesti ajana omaa ohjelmaansa ja käyttäen suurimaksi osaksi omaa muistiaan. Käyttämättömät DSP-ytimet voidaan sulkea, mikä vähentää virran kulutusta piirissä.

IHIP-piirin simulointimalli

Virtuaalikomponentit on suunniteltu siten että niiden liittynät ovat standardoituja rajapintoja, kuten esimerkiksi VSIA:n (Virtual Socket Interface Alliance), VCI:n (Virtual Component Interface) tai ARMin AMBA-väylän (Advanced Micro-controller Bus Architecture) liittäntä. Koska yksittäisten lohkojen kompleksisuus ei kasva samassa suhteessa kuin koko piirin kompleksisuus, tarjoaa virtuaalikomponenttipohjainen suunnittelu mahdollisuuden käyttää hajautettua simulointialustaa piirin funktionaalisuuden varmentamisessa.

Kun monimutkaista piiriä ajetaan yhdellä tehokkaalla työasemalla, kasvaa simulointiaika epäkäytännöllisen suureksi. Eri-tyyseen suurien piirien tapauksessa hajauttamalla piirin simulointi usealle työasemalle saadaan simulaationopeuteen parannusta vaikka kommunikaatioon ja synkronointiin kuluva aika onkin merkittävä. Hyötyä saadaan erityisesti porttitason simuloinneissa. Virtuaalikomponenttipohjaiset piirit tarjoavat standardoituine rajapintoineen valmiin hajautusrajapinnan.

IHIP-piiri koostuu osista jot-

ka ovat eri tarkkuudella ja kielellä määriteltyjä. Prosessoreista on olemassa käskykantasimulaattoreita RTL-mallien (Register Transfer Level) lisäksi, väylistä voi olla tapahtuma-pohjaisia malleja kellon- ja bitintarkkojen mallien lisäksi sekä muisteista voi olla käyttäytymis- tai porttitason malleja. Näiden mallien yhdisteleminen vaatii simulaattoreiden välille kirjoitettuja sovelluskohtaisia liittymiä. Simulaattorien väliset standardit rajapinnat, kuten OMI (Open Model Interface) tarjoavat kuitenkin mahdollisuuden automatisoida liittymä- ja hajautuskoodin generoinnin käyttäen jotain olemassa olevaa hajautusprotokollaa kuten MPI (Message Passing Interface).

IHIP-piirin simulointimalli on päätetty rakentaa Beowulf klusterin päälle. Beowulf on Linux-käyttöjärjestelmällä varustetuista PC-koneista rakennettu tehokas rinnakkaislaskentaympäristö. VHDL ja porttitason simulaatioihin tarkoitus on käyttää Mentorin ModelSim simulaattoria, SystemC-malleihin SystemC kirjaston mukana tulevaa simulaatioydintä, muistien simulointeihin Denalin Memory-Modelleria sekä lisänä prosessorien käskykantasimulaattoreita. Tarkoitus on mitata liittämiseen ja hajautukseen tarvittavan ohjelmakoodin tekemiseen tarvittavaa työmäärää, hajautetun simulaation suorituskykyä ja koko hajautetun simulaation konseptia.

SciFi-projektissa on havaittu,

että perusteellinen sovellusalueen analysointi on edellytys onnistuneelle tuotealustan määrittelylle. IHIP-piiriin kehitetty konfiguroitua muistiarkkitehtuuri on suorituskykyanalyysissä osoittautunut kehityskelpoiseksi lähtökohdaksi eri tyyppisten DSP-sovellusten suorittamiseen. Toisaalta kokemukset virtuaalikomponenttien käytöstä ja suunnittelun- ja varmennuksen hajauttamisesta osoittavat, että työkaluissa ja menetelmissä on vielä kehittämistä. SciFi-projektissa on ollut mukana keskeisimmät SOC-alueella Suomessa toimivat yritykset. Yksi selvimistä tuloksista onkin ollut, että tulevaisuuden SOC-piirit edellyttävät entistä laajempaa ja syvempää yhteistyötä.

Aiheesta enemmän

IP organisaatiot:

www.vsia.org
EDA-työkalut:
 www.mentor.com
 www.synopsys.com
 www.cadence.com
 www.sun.com/desktop/suntcf/
 www.coware.com/
 IBM:n mikroelektronikka-
 osasto:
 www.chips.ibm.com/
Tuotealustapiirit:
 www.triscend.com/
 www.semiconductors.philips.com/technology/velocity/
 www.sonicsinc.com/
Suomalaista IP-osaamista:
 www.fincitec.fi
 www.hantroc.com
 www.vlsi.fi
 www.elektrobit.fi
 www.smartech.fi

Taustat

Kirjoittajat: Juha-Pekka Soinen, erikoistutkija, VTT Elektronikka, juha-pekka.soinen@vtt.fi
 Jussi Roivainen, tutkija, VTT Elektronikka, Jussi.Roivainen@vtt.fi
 Antti Pelkonen, tutkimusharjoittelija VTT Elektronikka, Antti.Pelkonen@vtt.fi

Tutkimus: System Chip Inc, Finland

Yhteistyössä: ABB Corporate Research, DSLBit, Elektrobit, Fincitec, Hantro Products, Nokia Mobile Phones, Nokia Research Center, Nokia Networks, Synopsys Finland, VLSI Solution, VTT Elektronikka

Teknologiaohjelma: ETX

Sanasto

SOC	System-On-Chip
IP	Intellectual Property
VC	Virtual Component
PDB	Platform Based Design
RTOS	Real Time Operating System
OFDM	Orthogonal Frequency Division Multiplexing
DSP	Digital Signal Processing
IHIP	Information Highway Interface Platform
FFT	Fast Fourier Transform
IFFT	Inverse FFT
MIMD	Multiple Instruction, Multiple Data
VSIA	Virtual Socket Interface Alliance
VCI	Virtual Component Interface
RTL	Register Transfer Level
OMI	Open Model Interface
MPI	Message Passing Interface
VHDL	Very high speed integrated circuit Hardware Description Language