

JEITA

EDA アニュアルレポート 2004

Annual Report on Electronic Design Automation

— 65 nm から 45 nm テクノロジ世代の EDA 技術の進展に向けて —

2005年5月発行

作 成

EDA 技術専門委員会

EDA Technical Committee

発 行

社団法人 電子情報技術産業協会

Japan Electronics and Information Technology Industries Association

目 次

【巻頭言】	1
2004 年度 JEITA/EDA 技術専門委員会 委員一覧	3
略語一覧	5
1. EDA 技術専門委員会の活動	9
1.1 2004 年度 JEITA/EDA 技術専門委員会 事業計画	11
1.2 2004 年度 JEITA/EDA 技術専門委員会ホームページ	16
1.3 2004 年度 JEITA/EDA 技術専門委員会 年間実績・予定表	18
2. 各技術委員会の活動報告	19
2.1 PDM 研究会 (Physical Design Methodology Study Group)	21
2.2 標準化小委員会	24
2.2.1 標準化小委員会	24
2.2.2 IEEE/DASC (電気電子学会/設計自動化標準委員会) IEEE-SA	26
2.2.3 IEC/TC93 (国際電気標準会議/デザインオートメーション)	28
2.2.4 Accellera (設計記述言語の標準化機関)	31
2.2.5 SystemC タスクグループ	33
2.2.6 SystemVerilog タスクグループ	36
2.3 投資効率向上のための情報交換会	41
3. 関連イベント (主催/協賛) 報告	45
3.1 Electronic Design and Solution Fair 2005 (EDSFair2005)	47
3.2 システム・デザイン・フォーラム 2005	60
3.3 ASP-DAC 2005	69
4. 添付資料	77
4.1 PDM 研究会報告	79
4.2 SystemC タスクグループ報告	158
4.3 SystemVerilog タスクグループ報告	216

【巻頭言】

「65nm から 45nm テクノロジー世代の EDA 技術の進展に向けて」 EDA 技術専門委員会 2004 年度 委員長 樋渡 有

21 世紀の最初の decade も早いもので、その半分を通過しようとしている。半導体産業においては、既に 65nm テクノロジーに向けての技術開発が始まっている。また、半導体技術ロードマップの 2003 年版 (ITRS2003) によれば、2010 年には 45nm テクノロジーを適用した半導体の量産が始まると予測されている。90nm から比べて、4 倍の集積度が達成されるわけで、大規模化もさることながら、超微細加工技術に関わる多種多様な設計上の課題が顕在化し、新たに発生する。

EDA 技術専門委員会は、(社) 電子情報技術産業協会 (JEITA) における業界活動組織のひとつとして、電子機器の設計自動化 (EDA : Electronic Design Automation) に関係する様々な活動を行っている。特に、電子機器の機能・性能を決定する LSI 設計技術に係わる活動を、その中心に置いている。

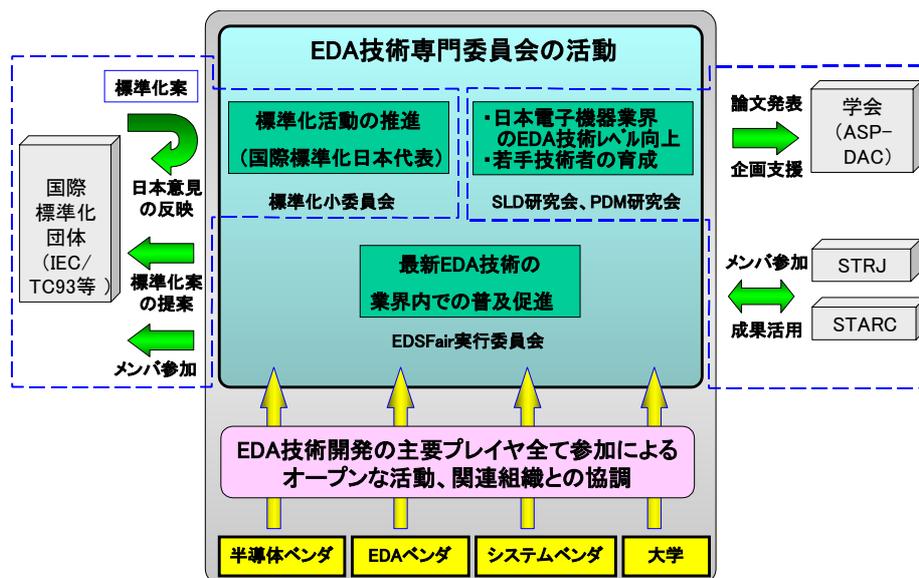
活動の第 1 テーマは、LSI 設計技術に関する動向、関連情報についての調査、検討と課題解決への提案である。システム・オン・チップ化と超微細プロセスに起因する設計課題検討と解決策の提示を目的とする PDM (Physical Design Methodology) 研究会を柱に取り組んでいる。

活動の第 2 テーマは、EDA 技術に関する標準化活動への貢献と関連機関、団体への対応である。IEEE、IEC 等の国際的な標準化活動に対し、標準化小委員会を中心に、技術提案・交流会議などを行なっている。特に電子情報通信学会内に設置されている IEC/TC93 国内委員会/WG2 (ハードウェア設計記述言語) へ委員を当委員会から多数派遣し、強い協力関係を構築している。また、昨年度から SystemC タスクグループと SystemVerilog タスクグループを設置し、両グループ協調して両言語の標準化に、日本としての貢献を行なっている。

活動の第 3 テーマは、EDA 技術の普及、推進のためのイベント実施、支援である。本年度も、2005 年 1 月にパシフィコ横浜において、電子機器の設計技術の総合的な展示会である “Electronic Design and Solution Fair 2005” を開催した。また、EDA 技術者やシステム・LSI 設計者に最新の設計技術、課題などを設計事例とともに紹介するシステム・デザイン・フォーラムを EDSFair 2005 と同時開催した。

さらに、EDA 関連技術の国際学会である ASP-DAC2005 の開催を支援した。

以下に上述した EDA 技術専門委員会の活動と関連団体との関係を示す。



IEC/TC93: 国際電気標準会議/設計自動化

ASP-DAC: Asia South Pacific-Design Automation Conference

STRJ: 半導体技術ロードマップ委員会

STARC: 半導体理工学研究センター

EDA 技術専門委員会活動と関連組織との関係

EDA 技術専門委員会はこれら様々な活動を通じ、日本と世界の電子機器業界のさらなる発展に寄与すべく、21 社約 60 名の業界各社有志・メンバーの自主的な参画により運営実行されている。本冊子「EDA アニュアルレポート 2004」は、EDA 技術専門委員会の 2004 年度活動の年次報告として、上記 3 つの活動テーマそれぞれについて、2004 年度の成果をまとめたものである。Web にも各種報告を掲載しているので、ご覧いただきたい。

(<http://eda.ics.es.osaka-u.ac.jp/jeita/eda.index-jp.html>)

65nm から 45nm のテクノロジー世代に向かって、電子機器が切り拓く素晴らしい未来が広がることを願い、2005 年度も積極的な活動を展開していく。

2004 年度 JEITA/EDA 技術専門委員会 委員一覧

委員長	樋渡 有 (株)	東 芝	セミコンダクター社 システム LSI 設計技術部 企画・管理 部長
副委員長	岡村 芳雄 (株)	ルネサステクノロジ	システムソリューション統括本部 エグゼクティブ
副委員長	相京 隆 (株)	富士通	LSI 事業本部 テクノロジー開発統括部 第三設計技術部 部長
会計監事	灘岡 満 (株)	沖電気工業	SiSC 戦略企画室 技術情報部 部長
幹事	藤波 義忠 (株)	NEC エレクトロニクス	基盤技術開発事業本部 技術企画部 シニア技術戦略プロフェッショナル
同	箱田 俊幸 (株)	三洋電機	セミコンダクターカンパニー LSI-BU 汎用デジタル設計技術部 部長
同	林 正樹 (株)	シャープ	IC 事業本部 要素技術開発センター 第1開発室 室長
同	齋藤 茂美 (株)	ソニー	SSNC DA ソリューション部門 1部 担当部長
同	山口 聖司 (株)	松下電器産業	半導体社 開発本部 先端 LSI 設計 統括グループ 統括グループマネージャー
同	園田 雅彦 (株)	ローム	LSI デザインクオリティ開発部 副部長
委員	安達 徹 (株)	NTT アドバンステクノロジー	先端技術事業本部 CAD システム事業ユニット ユニット長
同	今本 昌浩 (株)	シーケンスデザイン	取締役副社長 日本・アジアオペレーション
同	藤岡 督也 (株)	ジーダット	EDA 営業技術部 部長
同	横川 隆 (株)	図研	SoC 事業部 デザインセンター 部長
同	青木 貫司 (株)	セイコーエプソン	半導体事業部 IC 企画設計部 課長
同	山田 明宏 (株)	凸版印刷	設計本部 ソリューション部長

同	広瀬文保	日本ケイデンス・ デザイン・システムズ社	カスタマー・テクニカル・サービス 本部バイスプレジデント
同	飯島一彦	日本シノプシス(株)	技術本部 本部長
同	瀬谷和宏	丸紅ソリューション(株)	DAソリューション事業部 事業部長補佐
同	小島智	メンター・グラフィックス・ ジャパン(株)	テクニカルセール部 シニアマネージャー
同	藤井浩一	(株) リ コ ー	電子デバイスカンパニー 画像 LSI 開発 C CAD 技術室 課長代理
特別委員	蜂屋孝太郎	NEC エレクトロニクス(株)	第一開発事業本部 基盤開発事業部 最先端設計技術開発グループ シニアデザインエンジニア
同	長谷川隆	富士通(株)	LSI 事業本部 第2システム LSI 事業部 設計技術部 プロジェクト課長
同	浜口加寿美	松下電器産業(株)	半導体社 SLSI 開発本部 第一商品分野開発センター 主幹技師
客員	神戸尚志	近畿大学	理工学部 電気電子工学科 教授
同	今井正治	大阪大学	大学院 情報科学研究科 情報システム工学専攻 教授
同	吉田紀彦	埼玉大学	工学部 情報システム工学科 教授
同	小澤時典	(株)半導体理工学研究センター	技監
同	古井芳春	ソニーセミコンダクタ九州(株)	オフィサー

略語一覧

[1] 団体・組織の名称

Accellera	VI と OVI を統合した、設計記述言語の標準化に関連する活動機関
ANSI	American National Standards Institute 米国の標準化国家機関
ASP - DAC	Asia and South Pacific Design Automation Conference アジア・太平洋地域での EDA 関連の国際学会（1995 年に始まる）
CENELEC	European Committee for Electrotechnical Standardization EC（欧州委員会）の電気電子分野に関する標準化機関
DAC	Design Automation Conference 米国で行われる EDA 関連の国際学会
DASC	Design Automation Standardization Committee IEEE の下部組織で設計自動化に関する標準化委員会
ECSI	European CAD Standardization Initiative 欧州の設計自動化に関する標準化機関
EDIF Div.	Electronic Design Interchange Format Division EIA の下部組織で電子系の情報データ交換規格の検討機関
EIA	Electronic Industries Alliance 米国の電子業界団体(Association を Alliance に改称)
JEITA	Japan Electronics and Information Technology Industries Association 社団法人電子情報技術産業協会（電子業界団体）
ICCAD	International Conference on Computer Aided Design CAD に関する国際学会
IEC	International Electrotechnical Commission 電気電子分野に関する国際標準化機関
IEEE	Institute of Electrical and Electronics Engineers, Inc. 米国の電気電子分野の国際的な学会組織
IPC	Institute for Interconnecting and Packaging Electronic Circuits Industry Association 米国のプリント回路に関する業界組織
ISO	International Organization for Standardization 国際標準化機関
IVC	International Verilog Conference OVI が主催する Verilog HDL 国際学会

JPCA	Japan Printed Circuit Association 社団法人日本プリント回路工業会
OSC	Open SystemC Initiative SystemC の標準化団体
OVI	Open Verilog International Verilog - HDL に関連する技術の標準化と普及推進組織
SEMATECH	Semiconductor Manufacturing Technology Initiative (Consortium) 半導体技術を向上するために始まった米国の官民プロジェクト
Si2	Silicon Integration Initiative 設計環境の整備促進を支援する米国の非営利法人 (旧 CFI)
VASG	VHDL Analysis and Standards Group DASC 傘下の VHDL 標準化に関するワーキンググループ
VITAL	VHDL Initiative Toward ASIC Libraries VHDL ライブラリ標準化団体
VSIA	Virtual Socket Interface Alliance LSI の機能ブロックの I/F 標準化を目指している業界団体

[2] 標準化・規格に関する技術用語

ALF	Advanced Library Format OVI で検討された IP をも含む ASIC ライブラリのフォーマット
ALR	ASIC Library Representation ASIC ライブラリ表現
CALS	Computer Aided Logistics Support (1985) Commerce At Light Speed (1995)
CHDS	Chip Hierarchical Design System SEMATECH が要求仕様を作成した 0.25-0.18um 世代設計システム
CHDStd	Chip Hierarchical Design System technical data CHDS で使用するデータモデルの標準化
DCL	Delay Calculation Language 遅延計算のための記述言語
DPCS	Delay and Power Calculation System IEEE1481 として標準化推進されている遅延と消費電力の計算機構仕様
EDI	Electronic Data Interchange 電子データ交換
EDIF	Electronic Design Interchange Format EIA の下部組織で検討されている電子系の情報データ交換規格

ESPUT	European Strategic Program for Research and Development in Information Technology 欧州情報技術研究開発戦略計画
HDL	Hardware Description Language ハードウェア記述言語
IP	Intellectual Property 流通／再利用可能な LSI 設計資産（本来は知的財産権の意）
JIS	Japanese Industrial Standard 日本工業規格
SDF	Standard Delay Format 遅延時間を表記するフォーマット
SLDL	System Level Design Language システム仕様記述言語
STEP	Standard for the Exchange of Product Model Data CAD の製品データ交換のための国際規格
VHDL	VHSIC (Very High Speed Integrated Circuit) Hardware Description Language IEEE1076 仕様に基づくハードウェア記述言語
VHDL-AMS	VHDL-Analog and Mixed-Signal (Extensions) DASC の中で進められている VHDL のアナログ及びミックスドシグナル システムへの拡張

1. JEITA/EDA 技術専門委員会の活動

1. EDA 技術専門委員会の活動

1.1 2004 年度 JEITA/EDA 技術専門委員会 事業計画

委員会の名称	EDA 技術専門委員会 (Electronic Design Automation Technical Committee)		
委員会の目的	EDAに関連する技術およびその標準化の動向を調査し、その発展、推進を図り、もって国内外の関係業界の発展に寄与する		
委員会の構成	会員会社/委員	21社/21名	
	特別委員	3名	
	客員	4名	
委員会の役員	委員長	: (株) 東 芝	樋渡 有
	副委員長 (正)	: (株) ルネサステクノロジ	岡村 芳雄
	副委員長 (代行)	: 富士通 (株)	相京 隆
	監事	: 沖電気工業 (株)	灘岡 満
	幹事	: NEC エレクトロニクス (株)	柏木 治久
	同	: 三洋電機 (株)	箱田 俊幸
	同	: シャープ (株)	林 正樹
	同	: セイコーエプソン (株)	青木 貫司
	同	: ソニー (株)	齋藤 茂美
	同	: 松下電器産業 (株)	山口 聖司
	同	: ローム (株)	園田 雅彦

活動計画の概要 <別紙-1 参照>

下部組織の設置

- 標準化小委員会	主 査: 岡村 (ルネサステクノロジ)
- SystemC タスクグループ	主 査: 長谷川 (富士通)
- SystemVerilog タスクグループ	主 査: 浜口 (松下電器産業)
- PDM 研究会	主 査: 蜂屋 (NEC エレクトロニクス)
- EDSFair 実行委員会	委員長: 藤波 (NEC エレクトロニクス)

委員会メンバと担当 <別紙-2 参照>

委員会の予算	会費 20,000 円 * 12 月 * 21 社 = 5,040,000 円
委員会の開催	年 3 回程度 (予定日: 4/16 (金)、9/17 (金)、3/18 (金)) 適宜、幹事会を開催する <別紙-3 参照>
担当事務局	JEITA/電子デバイス部

<別紙-1>

活動計画の概要

1. EDA 技術に関する動向、関連情報についての調査、検討と課題解決への提案

- (1) 研究会による技術動向、ニーズ調査
DSM に関する技術課題の明確化と EDA 技術による解決策の検討
→PDM 研究会
- (2) 関連機関、団体、キーパーソン等との合同会議、意見交換、交流
STARC, STRJ 等
特に、STRJ/WG1 との具体的な連携・協同の検討
- (3) 国内外の学会、研究会、イベントへの参加
ASP-DAC2005, DAC2004, IEICE, IPSJ 等
- (4) EDA 環境調査
2004 年度も継続、調査内容の拡張に向けて検討

2. EDA に関する標準化活動への貢献と関連機関、団体への対応

- (1) EDA 設計言語およびモデル標準化のための技術的検討と提案
SystemC, SystemVerilog をそれぞれ 03 年度に新設した標準化小委員会下のタスク
グループにて継続して検討、提案活動を実施
Verilog-HDL, VHDL, アナログ HDL, DPC 等必要に応じて
→ 標準化小委員会で必要に応じて検討
- (2) 国際的な関連機関、団体への参画と標準化活動への協力
IEC/TC93 (特に WG2), Accellera, Si2 等
→ 標準化小委員会にて検討し、対応

3. EDA 技術および標準化の普及、推進のためのイベント実施、支援

- (1) 「EDSFair2005」(横浜)
日本エレクトロニクスショー協会へ運営委託
→ EDSFair 実行委員会
- (2) 各種ワークショップ、講演会の開催
「システム・デザイン・フォーラム 2005」を EDSFair2005 と同時開催

4. 委員会活動の広報

- (1) 活動成果等の各種技術報告書の発行
プロジェクト、研究会による技術報告書、調査報告書等
- (2) 広報パンフレット、アニュアルレポートの発行
- (3) WWW ホームページの公開

<別紙-2>

EDA 技術専門委員会メンバと担当

(2005 年 3 月 現在) (敬称略)

委員長	樋渡 有	(株) 東 芝	EDSFair/ASP-DAC 小委員会主査、 ASP-DAC 支援
副委員長	岡村 芳雄	(株) ルネサステクノロジ	標準化小委員会主査
副委員長	相京 隆	富士通 (株)	システム・デザイン・フォーラム 実行委員長、内規改訂
監事	灘岡 満	沖電気工業 (株)	EDA 環境調査、EDSFair 実行委員
幹事	藤波 義忠	NEC エレクトロニクス (株)	EDSFair 実行委員長
同	箱田 俊幸	三洋電機 (株)	ホームページ、メールシステム
同	林 正樹	シャープ (株)	広報パンフレット
同	青木 貫司	セイコーエプソン (株)	
同	齋藤 茂美	ソニー (株)	アニュアルレポート
同	山口 聖司	松下電器産業 (株)	
同	園田 雅彦	ローム (株)	
委員	安達 徹	NTT アドバンステクノロジー (株)	
同	今本 昌浩	シーケンスデザイン (株)	
同	藤岡 督也	(株) ジーダット	
同	山田 明宏	凸版印刷 (株)	
同	広瀬 文保	日本ケイデンス・デザイン・システムズ社	
同	飯島 一彦	日本シノプシス (株)	
同	瀬谷 和宏	丸紅ソリューション (株)	
同	小島 智	メンター・グラフィックス・ジャパン (株)	標準化小委員会副主査
同	藤井 浩一	(株) リコー	
同	横川 隆	(株) 図研	
特別委員	蜂屋孝太郎	NEC エレクトロニクス (株)	PDM 研究会主査
同	長谷川 隆	富士通 (株)	SystemC タスクグループ主査
同	浜口加寿美	松下電器産業 (株)	SystemVerilog タスクグループ主査
客員	神戸 尚志	近畿大学	元委員長
同	今井 正治	大阪大学	ASP-DAC リエゾン
同	小澤 時典	(株) 半導体理工学研究センター	元委員長
同	古井 芳春	ソニーセミコンダクタ九州 (株)	元委員長
事務局	穂苅 泰明	(社) 電子情報技術産業協会	
同	木暮 英男	(社) 電子情報技術産業協会	

<別紙-3>

2004年度 JEITA/EDA 技術専門委員会 会合予定

年/月	幹事会	技術専門委員会	関連イベント
2004/4	4/16(金) 沖 <ul style="list-style-type: none"> ・2004年度事業計画説明 ・2004年度プロジェクト/研究会計画説明 ・2003年度会計収支と2004年度会計予算説明 ・アニュアルレポート作成状況報告 	4/16(金) 三洋 <ul style="list-style-type: none"> ・2004年度事業計画説明承認 ・プロジェクト/研究会計画承認 ・2003年度会計収支と2004年度会計予算説明承認 	軽井沢 WS (4/26-27) @軽井沢
2004/5			
2004/6	6/18(金) シヤープ 関西 <ul style="list-style-type: none"> ・プロジェクト/研究会進捗報告 ・半導体幹部会報告内容説明 ・委員名簿更新内容確認 ・収支状況報告 		DAC2004 (6/7-11) @San Diego, California
2004/7			<ul style="list-style-type: none"> ・STARC/ASPLA 共催フォーラム (7/12) @パシフィコ横浜 ・DA シンポジウム 2004 (7/21-23) @浜松
2004/8			
2004/9	9/17(金) ローム <ul style="list-style-type: none"> ・プロジェクト/研究会進捗報告 ・収支状況報告 	9/17(金) ソニー 年度中間報告 <ul style="list-style-type: none"> ・プロジェクト/研究会進捗報告 ・収支状況報告 	STARC シンポジウム (9/9-9/10) @新横浜
2004/10			
2004/11	11/19(金) NEC エレクトロニクス <ul style="list-style-type: none"> ・プロジェクト/研究会進捗報告 ・収支状況報告 ・EDSFair用パンフレット作成手順説明 		ICCAD2004 (11/8-11) @San Jose
2004/12			
2005/1	1/21(金) 松下電器産業 関西 <ul style="list-style-type: none"> ・プロジェクト/研究会進捗報告 ・05年度体制協議 ・EDSFair用パンフレット内容 		ASP-DAC2005 (1/18-21) @上海、中国 EDSFair2005 (1/27-28) @横浜

	確認 ・ アニュアルレポート作成 担手順説明 ・ 収支状況報告		
2005/2			
2005/3	3/18 (金) 沖電気工業 (懇親会も開催) ・ 2004 年度プロジェクト／研 究会の年間活動報告 ・ 2004 年度収支状況報告 ・ 2005 年度事業計画説明 ・ 2005 年度プロジェクト／研 究会の年間活動計画説明		DATE2005 (3/7-3/11) @Munich,Germany

(各開催日の社名は、議事録担当を示す)

1.2 2004年度 JEITA/EDA 技術専門委員会ホームページ

1.2.1 目的

電子情報技術産業協会（JEITA）のEDA技術専門委員会の活動状況を公開し、EDA技術の標準化や技術調査に関する理解とご協力をいただくことを目的とする。また、委員会内部の活動状況を記録・掲示し、委員会活動の円滑化を計る。

1.2.2 一般公開用ホームページ

一般公開用ホームページは英語版からスタートし海外からの利用者の利便性が考えられている。日本語版への切り替えも可能である。

- (1) URL : <http://eda.ics.es.osaka-u.ac.jp/jeita/eda/>

大阪大学のご協力により、大阪大学のサーバーにホームページを設置させていただいている。

- (2) エントリーページの構成

日本語版、英語版はそれぞれ次のエントリーで構成されている。

日本語版：

JEITA/EDA 技術専門委員会の紹介

EDA 技術専門委員会の組織と活動概要

関連する機関・会議・展示会

EDA の標準化に関連する国内外の機関

英語版：

Profile (Message from Chairman, Organization)

Projects and Study Groups

Events

Technical Documents

Member Companies

EDA 技術専門委員会のコンタクトアドレス Contact us

- (3) EDA 技術専門委員会の組織と活動概要／Projects and Study Groups

委員会活動と成果のまとめ、下記の研究会・小委員会等の活動状況が紹介されている（英語版は一部）

- ・ 標準化小委員会

下記の過去に終了したプロジェクトの記録も紹介している。

- ・ DMD 研究会
- ・ SLD 研究会
- ・ TAB (Technical Advisory Board)
- ・ DPC プロジェクト
- ・ HDL 技術小委員会
- ・ HDL-IOP プロジェクト
- ・ EDA ビジョン研究会
- ・ 1364HDL プロジェクト
- ・ VHDL プロジェクト
- ・ アナログ HDL プロジェクト

- (4) 関連する機関・会議・展示会／Events

次の関係の深い EDA 関連技術委員会、関連機関、イベントの紹介およびリンクが行われている。

-
- ・ IEEE/DASC（電気電子学会／設計自動化標準化委員会）
 - ・ IEC/TC93（国際電気標準会議／デザインオートメーション標準化技術委員会）
 - ・ Electronic Design and Solution Fair (EDSFair) 2005
 - ・ 過去の会議・展示会
- (5) EDA の標準化に関連する国内外の機関
本委員会に関連のある 17 機関の紹介があり、またそれぞれのホームページへのリンクが行われている。
- (6) EDA 技術専門委員会のコンタクトアドレス
EDA 技術専門委員会、標準化小委員会などのコンタクトアドレスが記載されている。

1.3 2004年度 JEITA/EDA 技術専門委員会 年間実績・予定表

月	EDA 技術専門委員会	
	幹事会	委員会
2004年 4月	4/16 (金) 11-13:30 東京グリーンホテル御茶ノ水 蔦の間	4/16 (金) 14-17 東京グリーンホテル御茶ノ水 蔦の間
5月		
6月	6/18 (金) 関西 13-15 JEITA 関西支部 第2会議室	
7月		
8月		
9月	9/17 (金) 11-13:30 中央大学駿河台記念館 3階 310 会議室	9/17 (金) 14-17 中央大学駿河台記念館 3階 310 会議室
10月		
11月	11/19 (金) 14-17 新高輪プリンスホテル 地下3階 個室「マロニエ」	
12月		
2005年 1月	1/21 (金) 関西 14-17 電子会館 8階会議室	
2月		
3月	3/18 (金) 13-17 総会 JEITA306	

月	小委員会・研究会・関連行事他
2004年 4月	4/2 (金) 14-17 SystemC-TG JEITA314 4/8 (木) 10-12 標準化小委員会 JEITA310 4/23 (金) 13-17 PDM 研究会 報告会 JEITA305
5月	5/7 (金) 14-17 SystemC-TG JEITA310 5/21 (金) 11-17 SystemVerilog-TG JJEITA311 5/21 (金) 13-17 PDM 研究会 (関西) 電子会館ビル 8階会議室 5/31 (月) 15-17 臨時標準化小委員会 ルネサステクノロジ本社 (丸ビル)
6月	6/4 (金) 14-17 SystemC-TG JEITA 314 6/18 (金) 15-17 標準化小委員会 JEITA 関西支部 第2会議室 6/25 (金) 11-17 SystemVerilog-TG JEITA311 6/25 (金) 13-17 PDM 研究会 新高輪プリンスホテル エグゼクティブフロア ローズ
7月	7/2 (金) 14-17 SystemC-TG JEITA 関西支部 第1会議室 7/16 (金) 13-17 PDM 研究会 JEITA 関西支部 第1会議室 7/23 (金) 17:30-19 臨時標準化小委員会 ルネサステクノロジ本社 (丸ビル) 7/28 (水) 11-17 SystemVerilog-TG JEITA311
8月	8/20 (金) 13-17 PDM 研究会 SEMI ジャパン大島ビル 5階 C 会議室
9月	9/3 (金) 13-17 EDA 環境調査 TF 中央大学駿河台記念館 5階 580 会議室 9/3 (金) 13:30-17 SystemVerilog-TG インダストリアルホール 1階 中会議室 9/3 (金) 14-17 SystemC-TG JEITA314 9/7 (火) 11-13:30 標準化小委員会 機械振興会館 6階 68号会議室 9/24 (金) 13-17 (関西) PDM 研究会 JEITA 関西支部 第1会議室
10月	10/1 (金) 14-17 SystemC-TG JEITA314 10/22 (金) 13-17 PDM 研究会 JEITA306
11月	11/12 (金) 13-17 (関西) SystemVerilog-TG 松下電器産業㈱半導体社 長岡京テクノビル 2F 商談 30 11/16 (火) 11-13 標準化小委員会 機械振興会館 5階 516号会議室 11/26 (金) ~27 (土) PDM 研究会 集中審議 (名古屋) クラウンホテル
12月	12/3 (金) 14-17 SystemC-TG インダストリアルホール 4階第2会議室 12/17 (金) 13-17 SystemVerilog-TG サブワーキング会議 航空会館 地下1階 102 会議室 12/22 (金) 13-17 PDM 研究会 JEITA302
2005年 1月	1/11 (火) 11-13 標準化小委員会 機械振興会館 5階 516 会議室 1/26 (水) 11-17 (新横浜) SystemVerilog-TG 日本ケイデンス㈱会議室 1/27 (木) 17:30-18:10 IEEE1666 WG パシフィコ横浜
2月	2/4 (金) 13-17 (関西) PDM 研究会 JEITA 関西支部 第一会議室 2/18 (金) ~2/19 (土) SystemC-TG 集中審議 (焼津) ホテルアンピア松風閣 2/25 (金) 13-17 (関西) PDM 研究会 電子会館 8階会議室
3月	3/1 (火) 11-13 標準化小委員会 機械振興会館 6階 63号会議室 3/2 (水) 11-17 SystemVerilog-TG JEITA 308 3/25 (金) 14-17 SystemC-TG JEITA 306 3/25 (金) 13:30-15:15 PDM 研究会 成果報告会 JEITA 305 3/25 (金) 15:45-17 PDM 研究会 JEITA 305

2. 各技術委員会の活動報告

2. 各技術委員会の活動報告

2.1 PDM 研究会 (Physical Design Methodology Study Group)

2.1.1 目的

130nm プロセス以降のテクノロジー・ノードでは、期待通りに LSI が動作しないケースが増えつつある。この要因は主に物理設計 (RTL もしくはゲートレベルより後の設計) の困難さからくるものであり、65nm、45nm といった次世代テクノロジー・ノードにおける物理設計は益々困難になると予想される。本研究会では以下のような調査や標準化を実施することにより、最先端および次世代のテクノロジー・ノード向け設計環境のスムーズな立ち上げや、若手の設計技術開発者の育成を目的としている。

- 次世代テクノロジー・ノードで新たに発生する諸問題の物理設計へのインパクト調査
- 半導体ベンダーとその顧客の間のインタフェースに関わる設計課題を制御するための、設計ルール・ガイドライン作成
- 次世代テクノロジー・ノードで新たに発生する設計課題を検証するために必要なセルライブラリの標準化
- 上記の検証を行うツールが要件を満たすかどうか判定するための標準ベンチマーク・データ作成

2.1.2 活動内容

2003 年 7 月から活動を開始し、下記のテーマを取り上げて調査やベンチマーク活動を 2005 年 3 月まで実施した。

- チップ内の温度差が信頼性やタイミング設計に与えるインパクトの調査
- I/O 同時動作による電源ノイズの特性、解析モデルの調査
- 配線寄生素子抽出 (LPE : Layout Parasitic/Parameter Extraction) ツールの精度評価用ベンチマーク標準データ作成と要求精度の調査

これら 3 つのテーマを集中的に検討するために 3 つのタスクグループを設置し、2004 年度はそれぞれ次のような活動を行った。

- **オンチップ・サーマル・タスクグループ (On-Chip Thermal Effect Task Group: OCTE-TG)**
SOC におけるオンチップ熱分布のシミュレーションを行い、次の調査を行った。1) 熱勾配がチップ性能 (エレクトロマイグレーション、クロックスキューなど) に与える影響の調査。2) その影響の緩和手法についての具体的方法の調査。3) 温度依存性の強いリーク電流が熱勾配に与える影響の調査。
- **電源ノイズ・タスクグループ (Design for Power Integrity Task Group: DPI-TG)**
IO 系電源ノイズの感度解析を実施し次の調査を行った。1) どのパラメータの影響が大きいかを特定し設計指針を導出。2) SSO 解析に必要な最小限な簡易モデルを調査。3) ジッターなどの設計制約から SSO ノイズを制御するパラメータの制約に落とし込む手法の調査。
- **インターコネクト・モデリング・タスクグループ (InterConnect Modeling Task Group: ICM-TG)**

LPE ツールにおける次のようなモデリングの精度評価基準を作成した。1) 配線密度を均一化し、平坦性を向上させるためのダミーフィルパターン挿入に対する配線容量モデリング。2) CMP プロセスにおいて平坦化を阻害する要因となる、ディッシング、エロージョン、リセスによる配線断面形状モデリングと配線抵抗への影響。

これらの活動で得られた成果は、次のような形態により無償で一般に公開する。

- アニュアルレポート
- JEITA のホームページ
- 関連学会の研究会・カンファレンスで発表や論文誌投稿

成果の詳細は本アニュアルレポートの付録に掲載した。また、次のような学会発表を行った。

- [1] 「フロアプランにおけるオンチップ熱ばらつき解析と対策」、情報処理学会 DA シンポジウム 2004, 2004 年 7 月.
- [2] 「遅延計算およびシグナルインテグリティを考慮した配線寄生容量抽出精度評価」、情報処理学会 DA シンポジウム 2004, 2004 年 7 月.
- [3] "On-chip thermal gradient analysis and temperature flattening for SoC design", ASP-DAC2005, 2005 年 1 月.
- [4] 「ジッタ制約を考慮した IO 同時動作設計ルール提案」、第 18 回 回路とシステム軽井沢ワークショップ、2005 年 4 月.

2.1.3 関連機関の動向

米国の非営利標準化法人 SI2 (<http://www.si2.org/>) は、IEEE1481 の DPCS (Delay and Power Calculation System) 及び DCL を拡張した、OLA (Open Library API) や寄生素子抽出用プロセスパラメータの標準セット (SIPPs: Standard Interconnect Performance Parameters) 等、標準化の推進と普及に取り組んでいる。また、OVI (Open Verilog International) と VI (VHDL International) が合併して発足した Accellera (<http://www.acclera.org/>) では、従来の Verilog や VHDL に加え、上記 DPCS や EDA ライブラリの標準 ALF の IEEE 標準化作業もサポートしている。

一方日本では、あすかプロジェクトが発足し、半導体最先端技術の「壁」を産官学で共同して克服する体制ができている。具体的には、90nm テクノロジー・ノードにおける、システム・オン・チップ設計技術、デバイス・プロセス技術の共同開発が進められている。その実行組織のひとつである半導体理工学研究センター STARC (<http://www.starc.or.jp/>) では、SoC 設計技術開発等、先導開発が行われている。

(4) 参加メンバー

主査	蜂屋 孝太郎	NEC エレクトロニクス(株)
副主査	金本 俊幾	(株)ルネサステクノロジ
委員	岡 宏規	NTT アドバンステクノロジ(株)
同	阿久津 滋聖	沖電気工業(株)
同	石川 博	シーケンスデザイン(株)
同	中林 太美世	シャープ(株)
同	小野 信任	(株)ジーダット・イノベーション
同	太田 敏行	(株)ジーダット・イノベーション
同	室本 栄	日本ケイデンス・デザイン・システムズ社
同	小林 宏行	日本シノプシス(株)
同	奥村 隆昌	富士通 VLSI(株)
同	一宮 敬弘	松下電器産業(株)
同	佐藤 高史	(株)ルネサステクノロジ
同	市宮 淳次	(株)リコー
客員	増田 弘生	(株)半導体理工学研究センター
同	橋本 昌宜	大阪大学

2.2 標準化小委員会

2.2.1 標準化小委員会

(1) 発足の背景とミッション

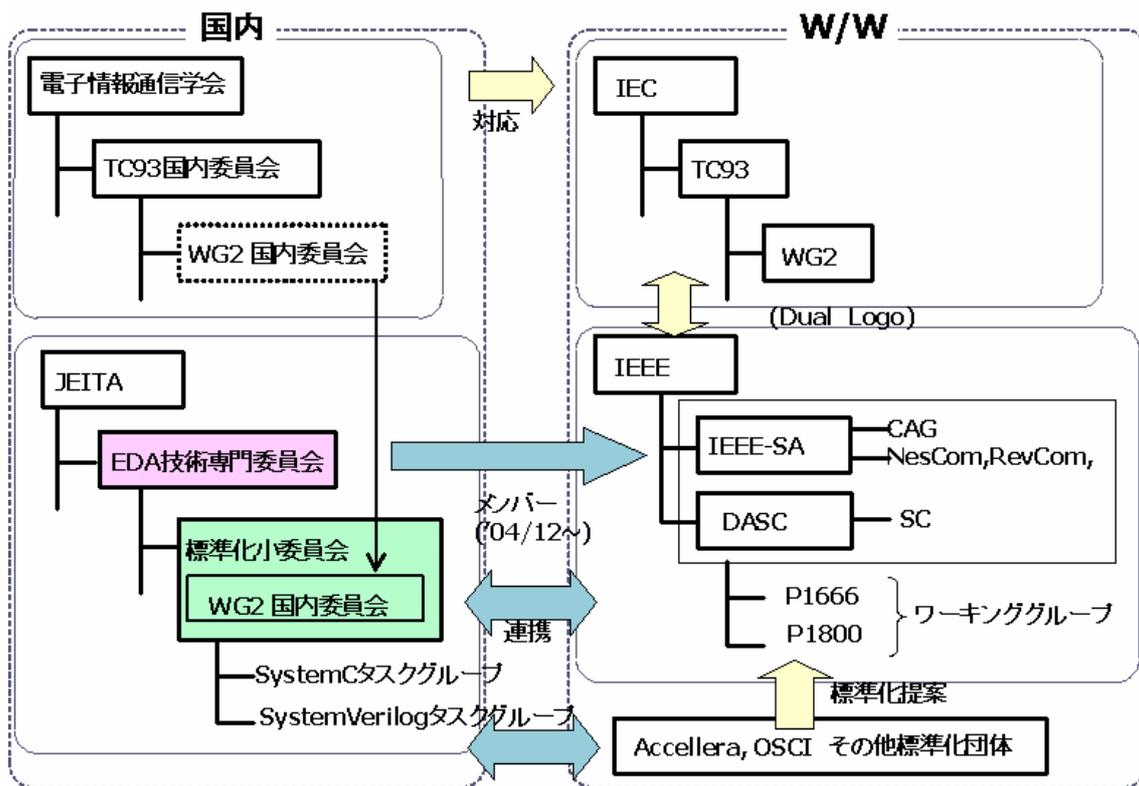
JEITA/EDA 技術専門委員会は標準化活動に関しては 1990 年の EIAJ/EDIF 研究委員会設立に始まり、当初は EDA に関するグローバルな重要課題に対して日本の業界を代表する唯一の機関として、特に設計記述言語の仕様標準化とその啓蒙等に多大な貢献を果たしてきた。近年は活動の中心が設計記述言語の普及定着と環境変化に応じて、先端的設計技術に関する調査・研究等にシフトしてきている。

システム設計メソドロジの革新が進展する中で、標準化は依然として重要なテーマである。標準化関連の活動をより明確に位置づけるため、2000 年 11 月に本小委員会が設立された。

世界的にみれば EDA 関連の標準は IEC (International Electrotechnical Commission) と IEEE (The Institute of Electrical and Electronics Engineers) で議論、制定されてきた。IEC ではデザインオートメーションを議論する TC (Technical Committee) 93、IEEE はコンピュータソサエティの DASC (Design Automation Standards Committee) と SA (Standards Association) である。これまでは IEEE で定められた標準を IEC でも追認するものも多かった。2003 年より議論は IEEE の DASC/SA のワーキンググループでも、標準の制定は IEC と IEEE で同時にできるようになった (Dual Logo)。

国内では IEC の対応機関は電子情報通信学会である。TC 毎に国内委員会があり、電子情報通信学会や JEITA に組織化されている。TC93 とハードウェア設計記述言語関連のワーキンググループ(WG2)の国内委員会は電子情報通信学会にある。

本小委員会は IEC/TC93/WG2 国内委員会を兼ねて活動するという協調体制を 2002 年度に確立した (図参照)。その結果、標準化小委員会の委員が IEC/TC93/WG2 の各種標準化提案を直接審議することができるようになった。



標準化小委員会と他の標準化組織との関係

2003年度には、SystemC および SystemVerilog の標準化を業界として検討・推進する目的で、それぞれタスクグループを発足させた。SystemC は、ますます重要性が認識されているシステムレベルの設計言語のひとつであり、SystemVerilog は IEEE1364 (VerilogHDL) の後継・検証技術の拡張である。両タスクグループも、日本の標準化組織として、海外の関連団体と連携し、言語仕様の専門的な技術検討と改善提案を通じて、標準化へ貢献すること目指して活動を開始した。

本小委員会のミッションは2002年度作成の内規では以下のように定義した。

「本小委員会は EDA および関連技術の標準化に関して、

- ・ 内外の動向を調査、検討し、
- ・ 技術および関連業界の発展に資する提案の必要性を模索し、
- ・ 必要かつ可能な場合には、関係機関に対して提案を行い、
- ・ 内外の標準化関連機関との連携・協調・協力を推進し、
- ・ 特に、デザインオートメーション／設計記述言語（TC93/WG2）WG の活動を支援し、
- ・ また広報活動を行う。」

(2) 2004年度標準化小委員会メンバー（2005年2月現在。敬称略）

主 査 岡 村 芳 雄 (株)ルネサステクノロジ
 副 主 査 小 島 智 メンター・グラフィックス・ジャパン(株)
 委 員 灘 岡 満 沖電気工業(株)

同	樋渡 有	(株)東芝
同	藤波 義忠	NEC エレクトロニクス(株)
同	箱田 俊幸	三洋電機(株)
同	齋藤 茂美	ソニー(株)
同	園田 雅彦	ローム(株)
同	林 正樹	シャープ(株)
同	相京 隆	富士通(株)
同	山口 聖司	松下電器産業(株)
特別委員	河原林 政道	NEC Electronics America, Inc.
同	間佐 五郎	(株)半導体理工学研究センター
同	長谷川 隆	富士通(株)
同	浜口 加寿美	松下電器産業(株)
同	明石 貴昭	日本シノプシス(株)
同	後藤 謙治	日本ケイデンス・デザイン・システムズ社
同	星野 民夫	(株)アプリスター
同	古井 芳春	ソニーセミコンダクタ九州(株)
同	後藤 和永	NEC エレクトロニクス(株)
同	石河 久美子	富士通マイクロソリューションズ(株)
同	蜂屋 孝太郎	NEC エレクトロニクス(株)
客員	今井 正治	大阪大学
同	神戸 尚志	近畿大学

(3) 2004 年度活動

2004 年度には、IEEE の標準化団体のひとつである IEEE-SA の正式メンバとなり、SystemC タスクグループおよび SystemVerilog タスクグループでは言語仕様案に関して国内の意見をまとめて提案するなど活動を本格化させた。また IEC/TC93 の国際会議に出席し JEITA、EDA 技術専門委員会、標準化小委員会の活動状況を紹介した。更に 2005 年 1 月 26 日、27 日には SystemVerilog, SystemC 標準化 WG である IEEE P1800, P1666 会議を横浜で開催、1 月 27 日には EDSFair と併設したシテスム・デザイン・フォーラム 2005 で SystemVerilog, SystemC のユーザフォーラムを主催した。

2.2.2 IEEE/DASC (電気電子学会/設計自動化標準委員会) IEEE-SA

(1) 活動の概要

IEEE は米国に本部を置く電気、電子、情報、などの国際的な学会である。また、この分野の標準化活動を長年にわたり、しかも広範囲に実施している。DASC、SA は IEEE の下部組織として、エレクトロニクス産業における設計自動化関連の標準化活動を行っている。

活動の中心は、標準設計記述言語 (HDL: Hardware Description Language) の VHDL と Verilog HDL に関連する設計と検証であり、タイミング情報、論理合成、算術関数とテストの標準化

に注力している。これら設計言語に関連して、システムレベルまで適用範囲を拡大して、Analog Mixed Signal、ソフトウェアとハードウェア協調設計等の拡張の標準化を検討している。最近では SystemC や SystemVerilog など高位設計技術言語や設計と検証を統合した記述言語の標準化作業が本格化している。

(2) JEITA/EDA-TC との関連

これまでは EDA 技術専門委員会は IEEE/DASC のメンバーとして関連する WG に参加し、標準化案に日本の意見を反映してきた。2004 年 12 月には IEEE-SA の正式メンバーにもなり、IEEE の標準化活動にドラフトレビュー、標準化案の改善の提案、投票を通じて積極的に参加している。

(3) 今までの成果と現在の状況

① 今までの成果

DASC/SA ではこれまでに以下の標準化作業を行っており、そのうちのいくつかは、IEC でも標準として承認されている。

- 1) VHDL (Std-1076)
- 2) VHDL Analog Extensions (Std-1076.1)
- 3) VHDL Math Package (Std-1076.2)
- 4) VHDL Synthesis Package (Std-1076.3)
- 5) VHDL Timing (VITAL) (Std-1076.4)
- 6) Verilog HDL (Std-1364)
- 7) MVL-9 (Std-1164)
- 8) Waveform and Vector Exchange (WAVES) (Std-1029.1)
- 9) The RTL Synthesis Interoperability Standard (Std-1076.6)
- 10) The Delay and Power Calculation Standard (Std-1481)
- 11) The Open Model Foundations Standard (Std-1499)

2004 年度は SystemC, SystemVerilog が IEEE の標準化プロジェクトとしてそれぞれ P1666, P1800 としてスタートした。

② 現在の状況

2005 年 2 月現在の DASC および SA とその傘下の Working Group と Study Group は以下のとおり。

- P1076 Standard VHDL Language Reference Manual (VASG)
 - VHDL-200x: the next revision
 - Issues Screening and Analysis Committee (ISAC)
 - VHDL Programming Language Interface Task Force (VHPI)
- P1076.1 Standard VHDL Analog and Mixed-Signal Extensions (VHDL-AMS)
- P1076.1.1 Standard VHDL Analog and Mixed-Signal Extensions - Packages for Multiple Energy Domain Support (StdPkgs)
- P1076.4 Standard VITAL ASIC (Application Specific Integrated Circuit) Modeling Specification (VITAL)
- P1076.6 Standard for VHDL Register Transfer Level (RTL) Synthesis (SIWG)

- P1364.1 Standard for Verilog Register Transfer Level Synthesis (VLOG-Synth)
- P1481 Standard for Integrated Circuit (IC) Open Library Architecture (OLA) (IEEE1481R)
- P1499 Standard Interface for Hardware Description Models of Electronic Components (OMF)
- P1603 Standard for an Advanced Library Format (ALF) Describing Integrated Circuit (IC) Technology, Cells, and Blocks (ALF)
- P1647 Standard for the Functional Verification Language 'e' (eWG)
- P1666 Standard System C Language Reference Manual (systemc) [cosponsored with IEEE-SA CAG]
- SystemVerilog Working Group
 - P1800 SystemVerilog: Unified Hardware Design, Specification and Verification Language (SV-IEEE1800) [cosponsored with IEEE-SA CAG]
 - P1364 Standard for Verilog Hardware Description Language (IEEEVerilog)
- P1850 Standard for PSL: Property Specification Language (IEEE-1850) [cosponsored with IEEE-SA CAG]

2.2.3 IEC/TC93 (国際電気標準会議/デザインオートメーション)

(1) 活動の概要

IEC は 1906 年に設立された歴史のある国際標準化機関であるが、設計自動化を取り扱う IEC/TC93 は 1992 年に設立された。TC93 の全体会議は毎年開催されており、スイス、英、仏、米、デンマーク、日、英、米、独、伊と開催されてきた。最近では 2002 年 10 月に中国・北京市、2003 年 11 月・2004 年 10 月は米国・Piscataway 市で開催され、各ワーキンググループ (WG) も同時開催されている。

(2) TC93 の組織と参加国

2003 年 3 月現在 IEC の Web サイト (www.iec.ch) によれば、25 カ国が TC93 のメンバーとなっている。IEC のメンバー資格には、P (Participating) と O (Observer) の二種類があるが、P メンバーは 8 カ国、O メンバーが 16 カ国である。P メンバーとしては、日本、中国、ロシア、スペイン、フィンランド、イギリス、米国、チェコが登録されており、O メンバーとして、フランス、オーストラリア、イタリア、ベルギー、オランダ、シンガポール、エジプト、スウェーデン、ウクライナ、ハンガリ、インド、アイルランド、デンマーク、ドイツ、韓国、セルビア・モンテネグロが登録されている。

(3) TC93 の組織とワーキンググループ (WG)

TC93 は 7 つの WG/JWG から構成されている。特に、WG2、WG3、WG6、および WG7 は日本から提案も含め積極的な貢献をしてきた。

WG1 : モデルのハーモナイゼーション : (a) STEP Electrical (ISO 規格) と EDA 標準の整合性の検討、(b) EDIF と AP-210 との整合性の検討

WG2 : ハードウェア設計記述言語 : (a) VHDL 言語仕様、Verilog HDL の整合性等の検討、システム記述言語 (SLDL) も議題に取り上げられてきた。(b) IC delay&power calculation system の検討。日本からの提案 ALR 標準化;IS (国際規格) 化完。現在

は SystemC, SystemVerilog が中心。

WG3 : 設計データ交換表現、(実態として休眠状態)

JWG11 : WG3 と TC52 (IPC 規格) との合同 WG、PCB 関連規格の整合性の検討

WG5 : 標準仕様相互協調の検証・認可技術

WG6 : 再利用可能部品ライブラリ、日・米・欧の各プロジェクト間の仕様整合と連携の検討、日本からは JEITA/ECALS プロジェクトの成果を提案している。IBIS も話題に取り上げられている。最近では Material Declaration (MD) に関する規格案が議論の中心となっている。

WG7 : システムテスト記述言語、ATLAS 言語と他言語の連携仕様の検討。日本からはデジタルシステムへの適用に関する提案をしている。

(4) TC93 国内委員会と主要メンバー (2005 年 2 月現在。敬称略)

・TC93 国際会議

議長：唐津 治夢 (SRI インターナショナル)

・国内専門委員会

委員長：高木 幹雄 (芝浦工業大学)

幹事：神戸 尚志 (近畿大学) *

委員：岡村 芳雄 (ルネサステクノロジ) *、柴田 明一 (JPCA)、高橋 満 (PartsWay)

・WG2 : (ハードウェア設計記述言語)

主査：岡村 芳雄 (ルネサステクノロジ) *

国際コ・コンベンナー、副主査：小島 智(メンター) *

委員：長谷川 隆 (富士通) *、浜口 加寿美 (松下) *

・WG3 : (設計データ交換表現)

主査：神戸 尚志 (シャープ) *

・WG6 : (再利用可能部品ライブラリ)

主査：森 啓 (NEC、国際コ・コンベンナー)

高橋 満 (PartsWay、国際コ・コンベンナー)

・WG7 : (システムテスト記述言語)

主査：山下 寛巳 (ワイマックス)

委員：唐津 治夢 (SRI インターナショナル、国際コ・コンベンナー)

*印は EDA 技術専門委員会からの参加者

(5) Piscataway TC93 国際会議の報告

2004 年の国際会議は 10 月に米国ニュージャージー州 Piscataway 市で開催され、総勢 16 名、日本からは 8 名が出席した。前議長任期満了に伴い後任の推薦を求められ、本年 10 月から日本推薦の唐津治夢氏が議長に就任したことにより昨年度の幹事国、本年の議長と人事が落着し、TC93 への P メンバー国増加、エキスパート確保、活動活性化が最重要課題となった。本年は、参加国が 2 カ国と過去最小であるが、日本からの参加は議長を含め 8 名と最多、昨年より 2 名増加し、再活性化への意欲が高まった。特に韓国が P メンバー参加に積極姿勢であることが報告され(本年は手続きが間に合わず不参加)、今後の発展につなげることが課題である。



WG2 では以下の審議、報告があった。

- ・メンバリスト更新（英 1、米 6、日 5、中 2、独 1、フィンランド 1、IEEE2）
- ・2003 年度 TC93/WG2 議事録承認
- ・2003 年度 活動内容まとめ

昨年来試験的に取り入れている IEC と IEEE のダブルロゴ方式は、着実に成果を挙げており 4 件が最終国際規格案（FDIS : Final Draft International Standard）を通過した。ALF が手続き中

- ・標準化状況の確認

① SystemVerilog (P1800)

P1800WG にて標準化案まとめ中、2005 年末までに IEEE での承認を目指す

② SystemC (P1666)

P1666WG の設置が 11/中旬に承認予定。2005 年末までに IEEE での承認を目指す。

- ・Dual Logo 関連

- ① 投票にかかる時間が IEC（5 ヶ月）は IEEE（2 ヶ月）と比べて長い
- ② レビュー時間も IEC は IEEE と比べて時間がかかり過ぎる
- ③ レビュー中の案の改訂は、IEC は時間のかかる各回のレビュー毎に行われレビュー中に随時改訂が行われる IEEE と比べて時間がかかり過ぎるなど Dual Logo に関わる審議プロセスの問題点が指摘、議論され全体の問題として継続検討となった。

- ・コンソーシアムの活動状況紹介

- ① JEITA/EDA-TCJEITA の新組織、EDA-TC、標準化小委員会、SystemC/SystemVerilogTG の状況を説明
- ② Accellera の状況 設立経緯、ミッション、メンバー、成果（SDF, VITAL, ALF）、現在の状況（SystemVerilog, PSL, OVL, OK）など

- ・今後の開催予定

2005/1 日本、2005/3 DATE、2005/6 DAC、2005/9 日本（TC93 と同時開催）

WG3 は活動が停滞し懸案となっていたが John Messina 氏を新コンビナーとし、ワークス

コープを見直して新 WG とすることとした。93/204/NP で提案された、環境を意識したマテリアルデklarレーション (MD) に関する規格案は、関連する領域も広く、また意識すべき立場、見識も相当多角的な配慮を必要とする広範な活動を目指すべきであるとの議論が行われた。

各 WG とも、XML を用いた記述方式を取り入れて、現在のコンピュータ技術への馴染みの良さを図ろうとする動きが目立つ。こうした流れは記述されるデータの種類のいかんを問わず、類似のハンドラー、アクセラレータの利用を可能とし、規格の処理系を普及させるためにも、合理性にかなっていると見える。ただし、従来の処理系・規格が深く浸透している領域では、現場の順応にはむしろ時間を必要とする可能性が高い。関連団体活動報告として米国からは、Accellera, IEEE SA, DASC, IPC などが中心であった。

次年度開催国として日本が立候補し、2005 年 9 月 5 日の週を提案承認された。奈良を中心に準備中である。

(6) 最近の IEC 規格投票について

2004 年度は下記 4 件の IEEE との Dual Logo の FDIS に賛成投票し成立した。

- 93/194/FDIS IEC 61691-6: VITAL ASIC (Application Specific Integrated Circuit) Modeling Specification (IEEE Std 1076.4-2000)
- 93/193/FDIS IEC 61691-5 : VHDL Language Reference Manual (IEEE Std 1076 -2002)
- 93/192/FDIS IEC 61691-4 : Verilog Hardware Description Language (IEEE Std 1364-2001)
- 93/191/FDIS IEC 61523-3: Standard Delay Format (SDF) for the Electronic Design Process (IEEE Std 1497-2001)

さらに下記 3 件の IEEE との Dual Logo の FDIS に賛成投票した。

- 93/212/FDIS(IEC 62050 Ed 1.0) : Standard for VHDL Register Transfer Level (RTL) Synthesis (IEEE Std 1076.6)
- 93/213/FDIS(IEC 62142 Ed 1.0) : Standard for Verilog Register Transfer Level Synthesis (IEEE Std 1364.1)
- 93/215/FDIS(IEC 62265 Ed 1.0) : Standard for an Advanced Library Format (ALF) Describing Integrated Circuit (IC) Technology, Cells, and Blocks (IEEE Std 1603)

2.2.4 Accellera (設計記述言語の標準化機関)

(1) 活動の概要

設計生産性を改善するには、世界標準とオープンインタフェースに基づいた設計方法論が重要である。Accellera は、Open Verilog International (OVI) と VHDL International (VI) を統合して 2000 年に設立された。設計記述言語を中心に新しい標準の認定、標準の開発およびこれに基づいた新しい設計方法論の普及促進を行っている。

メンバーは、半導体メーカー、システムメーカーと EDA ベンダーから広く参加しており、ハードウェア記述に留まらずシステム記述までを包含する設計記述言語を検討している。

(2) 現在の活動状況

Technical Committee を設置し、設計記述言語の開発と策定を行っている。Technical

Committee は、専門的に検討を進める SubCommittee を下部組織として持つ。また、Accellera で認定した標準は、国際標準として IEEE に提案することが慣例になっており、関連の深い IEEE の Working Group や他の標準団体を支援している。

Accellera Technical Committees

- Formal Verification Technical Committee (PSL)
- Harmony Technical Committee
- Interface Technical Committee
- Rosetta
- SystemVerilog Technical Committee
- Open Verification Library Committee
- Verilog AMS Technical Committee
- OpenKit Initiative Technical Committee

Committee Chair

Harry Foster
Jasper Design Automation
Wolfgang Roething, NEC
Brian Bailey, Mentor
Perry Alexander, Univ. of Kansas
Vassilios Gerousis, Infineon
Brian Bullis, Cisco
Sri Chandrasekaran, Motorola
Nick English

Accellera の本年度の成果は、Accellera 標準を承認したことであり、- Property Specification Language (PSL) 1.1, - Verilog-AMS 2.2 のふたつがある。また、Accellera 標準を IEEE 標準にするために IEEE-SA に提案したことである。SystemVerilog 3.1a は、今後 P1800 で標準のプロセスに乗ることになる。

標準化小委員会は、Associate Member として Accellera に参画しており、継続的に標準活動への参加と情報交換を行っている。本年度は、Accellera の会長他 キーパーソンを招き 2005 年 1 月横浜で技術情報交換を目的とした技術会議を主催した。

(3) メンバー会社

メンバーは、半導体メーカー、システムメーカーと EDA ベンダーが中心となり参加している。EDA 技術専門委員会は、2001 年 4 月から Associate メンバになった。メンバーリストを以下に示す。

Accellera Member List

Corporate Members

- ARM Ltd.
- Artisan Components
- Cadence Design Systems
- Freescale Semiconductor
- IBM Haifa Research Lab
- Intel Corp.
- Magma Design Automation
- Mentor Graphics
- National Semiconductor
- NEC

-
- Novas
 - NVIDIA
 - Real Intent
 - Sun Microsystems
 - Synopsys Inc.

Associate Members

- Aldec Inc.
- Bluespec, Inc.
- JEITA
- Silvaco
- ST Microelectronics
- TNI-Valiosys
- Toshiba

2.2.5 SystemC タスクグループ

(1) 背景

ハードウェア記述言語によるシステム LSI の設計は、VHDL (IEEE 1076) や Verilog-HDL (IEEE 1364) の標準化への JEITA (旧 EIAJ) の貢献とともに広く普及して、産業界で活用されている。一方、半導体の微細化技術は開発がさらに加速され、既に 1000 万ゲート規模の LSI が開発されるに至り、さらに抽象度の高いレベルからの設計が必須となってきた。1990 年代半ばより複数のシステムレベル設計言語の提案が行われ、標準化推進団体が結成されたものもあった。この中で、C++言語を基本とする SystemC は広く半導体メーカ、システムメーカ、EDA ベンダーの賛同を得て、Open SystemC Initiative (OSCI) が結成され、標準化のための言語仕様の策定と整備が進められてきた。

システムレベル設計言語としての要件を備えた SystemC V2.0 のリファレンスシミュレータがまず 2001 年 10 月にリリースされ、その後 2003 年 5 月に言語参照マニュアル (Language Reference Manual, 以下 LRM) が一般公開された。この LRM が 2004 年 11 月に OSCI より IEEE に移管され、IEEE P1666 として正式な標準化活動がスタートしている。それと並行して OSCI にて開発されていた SystemC V2.1 の言語拡張仕様を追加した上で、2005 年度中に標準化を終える見込みである。

(2) 目的

上記のような状況の中で、SystemC は SoC (System on Chip) の開発のためのシステムレベル記述言語のひとつとして既に設計や検証に幅広く使われるようになり、欠くことのできない言語となってきた。設計言語は設計の基本となるもので、この標準化策定に早くから関わることは、産業界にとって次世代の設計手法を構築する上で非常に重要なことである。

本タスクグループは 2003 年 10 月に設置され、日本国内における唯一の SystemC の標準化関連組織として、IEEE P1666 で進められる SystemC 標準作業に対して日本の産業界として意

見を述べ、国内事情・要求事項を取り込んだ形で国際標準化に貢献していく。また、SystemC に関連した調査結果をアニュアルレポートやユーザフォーラム等で積極的に情報発信を行うことで、SystemC を利用した設計手法の国内普及を図り、ひいては日本の産業界の国際競争力を高めることを目指す。

(3) 活動内容

本タスクグループは、次の3つの項目を柱として活動する。

① SystemC 標準化活動

- IEEE P1666 のメンバーとなり、SystemC 言語標準化活動に参画する。具体的には、言語仕様のレビューを行い、追加・修正が必要な点を指摘しフィードバックする。2005 年度中の標準化をめざし貢献する。

② SystemC 技術調査

- SystemC 関連の各種学会、セミナーに参加し、あるいは論文等の分析を行い、世界各国における SystemC の利用状況や設計フローの動向を調査する。
- SystemC の合成サブセットやトランザクションレベルモデリング、及び検証ライブラリといった拡張言語仕様について調査し、標準化の検討を行う。
- 他国では類を見ない、同一組織内に SystemC と SystemVerilog のタスクグループが存在し、共同で技術調査・検討・標準化を進めることが可能であるため、両者を連携したフローの検討や、連携のための技術的課題の先行抽出等を行う。

③ SystemC 普及活動

- EDSFair に併設したシステムデザインフォーラムの 1 プログラムとして SystemC ユーザフォーラムを開催し、SystemC に関連した情報や適用事例の発表の場と位置付け、積極的に情報発信を行い SystemC を利用した設計の普及をはかる。

(4) これまでの成果

2003 年 10 月に発足した後、これまでに次のような成果をあげた。

① SystemC 標準化活動

- OSCI より 2003 年 5 月に一般公開された LRM についてレビューを行い、問題点を 62 件抽出し（うち 46 件については 2003 年度の活動報告書に一覧を記載）、IEEE 並びに OSCI に報告した。
- 2004 年 12 月にリリースされた IEEE 版の LRM (Draft) をレビューし、上述の 62 件の問題点のうち 34 件が改善されていることを確認した。また 16 件については LRM から削除された等で問題がなくなった。また新たに 31 件の問題点が抽出されたため、残る 12 件と合わせて 43 件の問題点を IEEE に報告した。
 - 詳細については、4. 添付資料（4.2 SystemC タスクグループ 2004 年度活動報告）に記載。
- JEITA EDA 技術専門委員会として IEEE-SA の会員となったため、IEEE P1666 ワーキンググループに投票権のあるメンバーとして参加した。IEEE P1666 の会議はこれまでに 2 回開催（2004 年 12 月と 2005 年 1 月）され、本タスクグループより毎回参加している。尚、2005 年 1 月の会議は JEITA EDA 技術専門委員会でホストを行った。

-
- IEEE P1666 は現在次のメンバー会社（団体）が参加している。
 - Cadence (Chair)
 - Calypto
 - MentorGraphics
 - Synopsys
 - JEITA
 - Jeda
 - IEEE P1666 配下に技術サブワーキンググループが設置され、本タスクグループの参加を登録した。この技術サブワーキンググループにて実質的な LRM の編集が行われる。

② SystemC 技術調査

- 2003 年 11 月度に集中審議を行い、本タスクグループ参加各社の SystemC 利用状況について紹介しあい、業界内の現状ステータスについて理解を深めた。（2003 年度の活動報告書に記載）
- 2004 年度には、過去 5 年間に一般に公開されている SystemC 関連の論文や発表資料等 50 件の調査を行い、報告書を作成した。調査対象は、学会 (DAC, DATE 等)、大学関係 (Stanford、UCI 等)、SystemC ユーザグループ等における発表資料とした。
 - 調査結果の一覧とサマリについては、4. 添付資料 (4.2 SystemC タスクグループ 2004 年度活動報告) に記載。

③ SystemC 普及活動

- 2005 年 1 月 27 日に、EDSFair 併設のシステムデザインフォーラムのプログラムの 1 つとして SystemC ユーザフォーラム 2005 を主催した。（本ユーザフォーラムは、過去 4 回に渡り OSCI が主催してきたが、今回より JEITA EDA 技術専門委員会の主催としたものである。）
- フォーラムの内容としては、本タスクグループのメンバー各社より次の 4 つの講演を行っていただいた。
 - 河原林 政道氏 (NEC Electronics America) : 「SystemC の IEEE 標準化状況と最新情報」
 - 高嶺 美夫氏 (ルネサステクノロジ) : 「短 TAT 性能評価プラットフォームとその応用」
 - 塚本 泰隆氏 (リコー) : 「動作合成ツールを使った JPEG (DCT) 回路の設計事例」
 - 柿本 勝氏 (ソニー) : 「動作合成の設計適用事例とその効果」
- 定員 200 名の会場に 250 名弱の聴講者が訪れ、立ち見が出るほどの盛況であり、SystemC に関する関心の高さがうかがわれた。また各講演の後に質問時間を設けたが、それぞれ 1~2 件の質問があり、活発な意見交換の場となった。
- アンケート調査を実施した結果、次のようなコメントが寄せられた。次回以降開催時の参考としたい。
 - 会場が狭い。机が欲しい。休憩時間が短い。

- 発表内容に重なりが多い。プレゼンと配布資料を合わせて欲しい。
- Accellera, OSCI, JEITA に求められる事として、セミナー・ワークショップの定期開催等を通じての日本語による情報発信。
- また OSCI の厚意により前回及び前々回のアンケート調査結果をいただき、今回の調査結果と合わせて聴講者の動向について分析を行うことができた。大まかな傾向としては、主な使用言語は Verilog HDL が相変わらず多数を占めるが、SystemC に関しては様子見の段階から(部分的)使用の段階へ移行しつつあるようだ。また、SystemC がより普及するためには、高位合成などのツールのさらなる整備が必要と思われる。
 - 詳細については、4. 添付資料 (4.2 SystemC タスクグループ 2004 年度活動報告) に記載。

(5) 参加メンバー

主 査	長谷川 隆	富士通(株)
副主査	後藤 和 永	NEC エレクトロニクス(株)
委 員	清水 靖 介	沖電気工業(株)
同	森 井 一 也	三洋電機(株)
同	岡 田 和 久/山 田 晃 久	シャープ(株) ※
同	柿 本 勝	ソニー(株)
同	龍 田 純 一/逢 坂 孝 司	日本ケイデンス・デザイン・システムズ社 ※
同	中 野 淳 二	日本シノプシス(株)
同	今 井 浩 史	(株)東芝
同	竹 村 和 祥	松下電器産業(株)
同	中 村 和 秀	メンター・グラフィックス・ジャパン(株)
同	塚 本 泰 隆	(株)リコー
同	渡 邊 政 志	(株)ルネサステクノロジ
同	河原林政道	NEC Electronics America
客 員	今 井 正 治	大阪大学

(計 15 名、※期中メンバー交代あり)

SystemC タスクグループ URL:

<http://eda.ics.es.osaka-u.ac.jp/jeita/eda/member/std/SystemC/index.html>

2.2.6 SystemVerilog タスクグループ

(1) 背景

SystemVerilog の元となった Verilog HDL (ハードウェア記述言語) は、特定ツールの独自言語として開発されたが、その後、OVI (Open Verilog International) による言語仕様公開を経て、1995 年に IEEE1364 として標準化承認された。その 5 年後、ディープサブミクロン対応の機能を盛り込まれ IEEE1364-2001 として改訂された。JEITA (旧 EIAJ) の EDA 技術専門委員会

では、IEEE1364 の標準化作業が開始されると同時に Verilog HDL 標準化プロジェクトを設置し、継続的に言語仕様の技術検討・国際標準化に貢献してきた。

その後、半導体の微細化技術はさらに加速し 1000 万ゲート規模の LSI が開発されるに至り、一般的に「論理設計工数の 7~8 割が機能検証に費やされているにも関わらず、6 割近い LSI が機能バグ等の問題によりリ・スピニングしている」という報告もある (Collett International)。このような状況において、新しいテストベンチ記述、アサーション/プロパティ記述の必要性が高まり、2000 年以降にいくつかの検証用言語が実用化された。

SystemVerilog は、Verilog HDL に ①デザイン (RTL 設計) のための記述構文 ②検証用言語を追加したものである。新たに検証用言語を持たせたことが大きな特徴であるが、デザインの面でも Verilog HDL に比べ記述量の削減や曖昧性の排除といったメリットがあり、設計品質の向上が期待できる。SystemVerilog の標準化は、SystemVerilogV3.1a が Accellera により制定され、現在、IEEE において IEEE P1800 として標準化が進められている。

(2) 目的

上記の SystemVerilog 標準化において、日本の半導体産業界の要望に沿った形での言語標準化を進めることが、適用容易性を高め、設計品質向上・国際的な競争力確保といった結果につながる。本タスクグループでは、業界各社から参加したエキスパートにより SystemVerilog の技術的な検討を実施し国際標準化に貢献すること、SystemVerilog に関する最新情報の収集と情報発信を目的としている。また、日本国内において SystemVerilog 適用の普及推進にも積極的に取り組む。

(3) 活動内容

① グループの結成と言語習得 (2003 年度)

2003 年 10 月に、SystemVerilog 言語仕様の技術検討・標準化を推進するためのタスクグループとして「SystemVerilog タスクグループ」を結成した。03 年度は、全メンバーが詳細な言語仕様を習得する期間と設定し、内部で言語セミナーを実施するとともに、SystemVerilog の代表的な言語仕様を説明する資料を作成した。

② 国際的な情報収集・標準化組織との連携開始

2004 年 3 月に、米国で開催された国際学会 DVCon (Design & Verification Conference) にメンバーを派遣し、最新技術の情報収集を実施した。また Accellera/SystemVerilog 会議に本タスクグループの主査と副主査が参加し JEITA での活動内容を紹介するとともに今後の連携についてお互いの意識をあわせた。

③ 言語仕様検討と IssueReport の提出

SystemVerilog 言語仕様検討において、より議論を深めるため以下のように、「デザイン」と「検証」の 2 つのサブタスクグループにわかれ、専門的な技術ディスカッションを重ねた。

- ・ SV-DSTG (SV Design Sub TG)
- ・ SV-VSTG (SV Verification Sub TG)

2004 年 8 月に、改善提案を含む 32 件の IssueReport をまとめ、Accellera と IEEE に提出した。本案件は、IEEE P1800-WG の審議対象として DB に登録され、現在審議がなされている。この IEEE P1800-WG Errata 提出案件とその状況については添付資料 4.3 の SV-TG2 に最新情

報を示す。

④ IEEE-SA と IEEE P1800 投票グループへの登録

上記の提案を有効にするため、また、今後の標準化活動における発言権・投票権を得るために、標準化小委員会を IEEE の標準化機関である IEEE-SA メンバに登録した。そして、05 年 1 月に本タスクグループは IEEE P1800 の投票グループのメンバになった。

⑤ 定例会議と IEEE P1800-WG 合同会議

今年度は、5 月、6 月、7 月、9 月、11 月、1 月、3 月に計 7 回の定例会議を開催した。1 月の会議は、IEEE P1800 との合同会議として日本で開催した。本タスクグループメンバと IEEE P1800-WG のキーパーソンが直接議論を行う貴重な機会となった。この合同会議では、EDA 技術専門委員会傘下の標準化小委員会の組織・ミッションや本タスクグループの活動内容について紹介し IEEE P1800-WG のメンバに JEITA の活動に対する理解を深めてもらった。また、P1800 メーリングリストへの参加や投票グループへの登録など、今後の標準化作業がより円滑になされるための環境整備も進み、非常に有意義な会議となった。議事全般に関しては、IEEE P1800 の定例会議として進行したが、多くの情報を得ることが出来た。

これらの定例会議と合同会議以外に、12 月に半導体メーカ会社 5 社だけの打合せをもち、「各社 EDA ツールの SystemVerilog 言語サポート状況」の調査結果について議論した。

⑥ SystemVerilog ユーザフォーラムの開催

2005 年 1 月に、「システム・デザイン・フォーラム 2005」のカリキュラムのひとつとして初めての「SystemVerilog ユーザフォーラム」を開催した。このフォーラムでは、Accellera の Chairman、IEEE P1800-WG の Chairman 代理からそれぞれの標準化状況の説明、また IEEE P1800-WG のメンバの米設計コンサルティング会社より、米国での設計事例として、実用における SystemVerilog 利用メリットについて講演をして頂いた。

本タスクグループからは、SystemVerilog の特徴・利点を広く周知させるために、IEEE P1800 のベースとなっている Accellera SystemVerilog3.1a の言語チュートリアルを実施した。本チュートリアルでは、SystemVerilog の概要・構成を説明するだけでなく、SystemVerilog の普及をより一層促進するためには、より実践的な SystemVerilog の有効性を周知させることが必要と考え、「ユーザの立場からみた言語チュートリアル」との方針で作成した。「ユーザの立場」についてメンバ内で議論した結果、本チュートリアルで説明する構文を次の 2 つの観点で選定した。

A. 機能的に有効と考えられる構文

- ・ 記述の曖昧さの解消
- ・ 可読性,保守容易性,記述量の削減
- ・ EDA ツールの性能向上

B. 標準言語として多くのツールでサポートされている構文

2004/12 末時点での EDA ツールサポート状況

- ・ 全ての分野で現在使える構文
(Lint, Simulation, Synthesis, Equivalence Checker)
- ・ テストベンチとして現在使える構文

(Simulation)

- ・ 早期にサポートを期待する構文

上記の基準から選定された 12 の構文に関して、できるだけサンプル記述を用いて、具体的に有効性を理解できるように配慮しつつ説明を実施した。

本チュートリアルテキストは、添付資料 4.3 の SV-TG3 として掲載する。

⑦ SystemVerilog ユーザフォーラムのアンケート結果分析

以下、本フォーラムにおける受講者アンケートの結果について記す。

【満足度】

	内容	予稿集	業務に役立つか
満足、まあ満足	49.3%	48.4%	50.4%
不満、やや不満	9.9%	9.9%	8.6%

- ・ 全項目で「満足,まあ満足」が 50%前後であり、「不満, やや不満」を大きく上回っている。
- ・ 「不満, やや不満」が全て 10%以下であることより、受講者の満足度は高いと判断できる。

【個別の結果】

A) 今回 SystemVerilog ユーザフォーラムに参加された目的は？（複数回答可）

- ・ フォーラムへの参加目的は、SystemVerilog 技術調査と EDA および標準化動向調査で約 81.7%と多く、導入検討は 12.7%にとどまっている。

B) SystemVerilog での設計・検証環境構築について（複数回答可）

- ・ 設計/検証に使用中は、あわせて 4.5%であり、検討中がまだまだ多い。
- ・ 使用中、検討中ともに、設計と検証への適用は半々である。

C) B) で「適用中」または「検討中」と回答された方へ。SystemVerilog の使用目的は？（複数回答可）

- ・ 使用目的は、テストベンチ、アサーションとあわせると 63.7%と、システムモデリング、RTL 記述の 36.3%を大きく上回った。設計言語よりも、検証言語としての期待が大きいと考えられる。

D) B) で「検討中」と回答された方へ。導入予定時期は？

- ・ 導入予定は、未定が 75.8%と多い。今後の動向を見てから導入を検討するというのか。

E) B) で「必要ない」と回答された方へ。または「検討中」の方へ。

導入の障害となっている理由は何ですか？

- ・ 導入の障害となっているのは、対応ツールが少ないという意見が一番多い。言語の完成度が不十分とあわせると 40.3%である。設計/検証の環境整備の遅れが導入を妨げている大きな要因と考えられる。
- ・ 少数であるが、効果がわからないという意見も出ている。今後も、ユーザフォーラムなどを通して SystemVerilog の有用性を訴えていかなければならない。

F) SystemVerilog をより活用するために充実が必要なものは？（複数回答可）

-
- ・ より充実が必要なもの、1～4位までがツールに関するものである。このことから環境整備が重要であることがわかる。
 - ・ 5位のコーディングスタイルガイドラインのニーズが意外と高い。

【総括】

ユーザフォーラム参加者の間では、SystemVerilog はまだ普及しているとはいえない。導入計画も未定が多く、まだまだ様子を見ている感がある。これは、言語仕様がまだ Fix していないことと、EDA ツールが対応していないことが大きな要因であると推測できる。環境整備が進めば、一気に普及する可能性があると考えられる。また、アサーション、テストベンチといった検証言語としての期待が高く、設計言語と同じ言語で検証も行いたいというニーズの表れであると考えられる。

(4) 今後の活動計画

今後も、Accellera や IEEE P1800-WG と連携した標準化活動を展開し、言語仕様策定の議論や投票に参画する。定期的な合同会議開催あるいは Accelle, IEEE P1800-WG の定例会議に代表者を派遣するなど、継続的な情報交換を実施する。また、04 年度の技術検討の結果として提出した Issue Report の案件について継続的にフォローをする。他に、SystemVerilog に関する技術情報を幅広く入手するため、国際学会への代表者参加や、SystemVerilog を利用した設計フローや設計事例についての調査など情報収集に努める。

05 年度は、SystemVerilog の専門用語(英語)に対する標準的な和訳を決めることにより、国内業界での統一的な訳語利用を推進する。普及活動としては、SystemVerilog ユーザフォーラムの第 2 回の開催を計画している。

また、今後、SystemC と SystemVerilog の連携した設計・検証フローの検討や技術的課題についての議論も開始する予定である。

(5) 関連機関の動向

2004 年 5 月に、Accellera による SystemVerilog V3.1a の言語仕様の定義が完了し、6 月に Accellera から IEEE にドネーションされた。同時に発足した IEEE P1800-WG により、このドラフトのレビュー・改定を重ねた。2005 年 3 月時点では IEEE P1800-WG により作成された Draft4 のイニシャル投票が実施されている。本投票は、2005 年 5 月になる予定であり、9 月に IEEE 1800 として承認されるスケジュールで進んでいる。標準化のスケジュールについては、添付資料 4.3 の SV-TG1 に示す。

(6) 参加メンバー

主 査	浜口 加寿美	松下電器産業(株)
副主査	明石 貴昭	日本シノプシス(株)
委 員	湯井 丈晴	(株)沖ネットワークエルエスアイ
同	後藤 謙治	日本ケイデンス・デザイン・システムズ社
	土屋 丈彦	(株)東芝
	千綿 幸雄	富士通(株)
	山元 涉	メンター・グラフィックス・ジャパン(株)
	高嶺 美夫	(株)ルネサステクノロジ
	中谷 英峰	(株)図研

2.3 投資効率向上のための情報交換会

2.3.1 開催経緯

2002年度、2003年度と日本半導体ベンダ各社のEDA環境関連の設計効率・投資効率といった指標の調査を実施してきた。目的は、EDA発展のため、自社EDA環境投資あるいはEDA環境改善の目標設定時のリファレンスデータを得るためである。

しかし、NDA締結を行わない調査であったため、自社の業界内でのポジションがクリアになっても、どこをどのように改善すれば良いのかわかりにくい、具体アクションにつながりにくい、という課題があった。

そこで、改善に向けたアクション開始の一助にすべく、2004年度に投資効率向上のための情報交換会を行うこととした。以下がその開催要領である。

2.3.2 開催要領

(1) 対象テーマ：インフラ

- ・EDAツールやサーバの稼働率向上策
- ・Linux化推進策 など

(2) 目的：投資効率の向上

- ・各社が取り得る具体アクションへのヒントを得る
- ・上記テーマに関してのEDAベンダへの要求項目を取り纏める

(3) 実施に向けた考え方：ギブ & テイク

- ・(参加条件) 自社での対象テーマに関しての工夫/EDAベンダへの要求を発表すること

(4) オープン性：参加社内でクローズ

- ・NDA：締結せず

(5) 実施日：2004年9月3日（金）午後

第一部：各社発表

第二部：EDAベンダへの要求 取り纏め

(6) 参加メンバ：EDA技術専門委員会幹事会メンバ + 発表者

(7) EDAベンダへの要求：

・9月17日（金）のEDA技術専門委員会にて、参加のEDAベンダに伝達報告

情報交換会のテーマをインフラとしたのは、半導体ベンダ共通の課題である点、設計フロー構築のような競争領域ではない点を考慮したためである。また、EDA技術専門委員会にはEDAベンダも委員として参加しているため、対象テーマとしたインフラに関するEDAベンダへの要望も取りまとめ、EDAベンダへの伝達も行うこととした。

2.3.3 開催結果

9月3日に、NECエレクトロニクス、沖電気工業、三洋電機、シャープ、ソニー、東芝、富士通、松下電器産業、ルネサステクノロジの参加を得て開催。以下を発表していただいた。

- (1) インフラ系投資効率の向上に向けた各社の取り組み
 - ① EDAツールやサーバの稼働率向上策
 - ② Linux化推進策
- (2) 上記テーマに関してのEDAベンダへの要求

(1) のインフラ系投資効率の向上に向けた各社の取り組みに関しては、参加会社以外は非公開としたので、本レポートでの内容紹介は割愛するが、各社で採用しているEDAツールのライセンス本数最適化のための基準値の考え方が報告されるなど有意義であった。

(2) のEDAベンダへの要望として挙げられた主なものを以下に記す。

キーワード	内容
OS 対応	Linux 早期サポート
	OS バージョンアップとツールの対応を合わせること
	OS のエンドオブサポート時期の延長
OS 対応情報開示	OS 動作検証結果開示
推奨マシン統一	EDA ベンダ間での推奨マシンの統一
ライセンスモデル多様化	ライセンスのユースモデルの多様化、多線化
対価根拠 明確化	保守費用の対価の内訳の具体化・明確化
ベンチマーク結果開示	各種 OS 間の EDA ツールの性能比較を公表
チェックポイントツールへの対応	チェックポイントツールへの対応、もしくはチェックポイント機能の実装
ツールのマルチスレッド化対応	ツールのマルチスレッド化、かつグリッドシステム（特に LSF）への対応
Flexlm 対応	Flexlm 最新機能の取り込み
ライセンスチェックアウト最適化	利用終了段階でのライセンスの開放
品質向上	新規バージョンリリース前のベンダ QA の充実
日本語マニュアル充実	日本語マニュアル対応

上記 EDA ベンダへの要望事項は、9 月 17 日開催の EDA 技術専門委員会にて EDA ベンダに説明した。

要望事項に対する EDA ベンダでの検討結果を報告してほしいとの意見が半導体ベンダ委員からあったが、委員会という公の場での回答は出来ない旨の意見が EDA ベンダ委員からあり、検討結果の報告は求めないこととした。

3. 関連イベント（主催／協賛）報告

3. 関連イベント（主催／協賛）報告

3.1 Electronic Design and Solution Fair 2005（EDSFair2005）

Electronic Design and Solution Fair（以下 EDSFair）2005 は、2005 年 1 月 27 日（木）～28 日（金）の 2 日間、横浜国際平和会議場（パシフィコ横浜 展示ホール）で開催された。

EDSFair は、1993 年以来、社団法人日本電子機械工業会（EIAJ、現：社団法人電子情報技術産業協会）が開催していた「EDA TechnoFair」と、FPGA/PLD ベンダー各社および大学関係者が組織して実施して来た「FPGA/PLD Design Conference&Exhibit」を 2001 年に統合した展示会である。

この統合から 5 回目を迎えた EDSFair2005 では、さらに展示会としての幅を拡大するべく、EDA 技術はもとより、システム LSI や FPGA/PLD 等の先進デバイス技術、組込みプロセッサ開発環境と組み込み OS 等、さらに、IP や各種設計サービスの分野をも網羅包含する企画を推進したことなどにより、出展者数、小間数とも前回を上回り、盛況のうちに終了した。

なお、今回のキャッチコピーは“「最先端」が「進化」を加速させる。”とし、先端技術の発想やテクノロジーがここに終結されているイメージを追及した。

3.1.1 EDSFair2005 の概要

- (1) 開催期間：2005 年 1 月 27 日（木）～28 日（金）
- (2) 場 所：パシフィコ横浜（展示ホール、アネックスホール）
- (3) 主 催：社団法人 電子情報技術産業協会（JEITA）
協 力：Electronic Design Automation Consortium（EDAC）
後 援：経済産業省、アメリカ合衆国大使館、外国系半導体商社協会（DAFS）、横浜市
協 賛：社団法人 電子情報通信学会（IEICE）、社団法人 情報処理学会（IPSI）
社団法人 日本プリント回路工業会（JPCA）
運 営：日本エレクトロニクスショー協会
- (4) 開催概況
 - ① 入場者数：11,153 名（前年 10,787 名）
 - ② 出展者数：119 社／336 小間（前年 105 社 306 小間）
 - ③ 出展者セミナー：121 セッション、延べ聴講者数 3,431 名
 - ④ スイートルーム：4 社
 - ⑤ ユニバーシティ・プラザ：19 ブース、16 大学 21 研究室
 - ⑥ キーノートスピーチ：聴講者数 286 名
 - ⑦ 併催
 - 第 12 回 FPGA/PLD Design Conference：10 セッション、聴講者数 591 名
 - ・ユーザ・プレゼンテーション：12 プレゼンテーション、聴講者数 120 名
 - ・IP フリーマーケット in EDSFair：21 テーマ、27 日延べ聴講者数 91 名
 - ・有料聴講者特典パネルセッション：聴講者数 34 名
 - ⑧ 同時開催
 - システム・デザイン・フォーラム 2005：6 セッション、聴講者数 983 名

3.1.2 出展構成

- (1) ハードウェア・ソリューション
システム LSI、ASIC/ASSP、MPU/MCU/DSP、FPGA/PLD デバイス、その他
- (2) ハードウェア設計環境 (EDA)
 - ① LSI 設計関連ツール
システムレベル設計 (RTL より高位)、論理設計 (RTL～ネットリスト) 論理検証、アナログ設計・検証ツール、IC レイアウトツール、IC レイアウト検証・解析 IC 信号解析
 - ② PCB 設計関連ツール
回路図作成、PCB レイアウト、PCB 信号解析、部品検索/管理ツール
 - ③ 統合設計 (SiP) 関連ツール
- (3) LSI テスト、計測器
LSI テスタ、デザインフォーテストツール、PCB テスタ、計測器
- (4) IP コア、マクロ
IP コア、セルライブラリ、メモリ
- (5) ソフトウェア・ソリューション
組込み OS、デバイスドライバ、ファームウェア/ミドルウェア
- (6) 組込みプロセッサ開発環境
リコンフィギャブルプロセッサ、ICE、デバッガ、マイコン CASE、コンパイラ/クロスコンパイラ、シミュレータ、ハードウェア/ソフトウェア協調開発環境
- (7) 設計サービス関連
デザインセンタ・デザインサービス、設計コンサルティング、IP 流通サービス、その他
- (8) 設計プラットフォーム
- (9) システム管理ツール
- (10) PR 関連 (出版物、その他)

3.1.3 開会式

1月27日(木)午前9時45分より展示会場入口において開会式を執り行った。

開会式への登壇者は次のとおり。

ご祝辞・テープカット：経済産業省 商務情報政策局 情報通信機器課長 福田秀敬様

ご祝辞・テープカット：アメリカ合衆国大使館 上席商務官 ジョン・フレミング 様

ご祝辞・テープカット：横浜市経済局長 横山 悠様

主催者挨拶・テープカット：株式会社ルネサステクノロジ 代表取締役社長&COO 伊藤 達

開会式終了後、登壇者および関係者による会場一巡が行われ、本年は下記のブースを訪問し、新技術・研究開発等の成果の説明をおこなった。

- ・(株)半導体理工学研究センター ・富士通(株) ・ユニバーシティ・プラザ ・ガイオテクノロジー(株)
- ・シンプリシティ(株) (以上、見学順)

3.1.4 出展者一覧

出展会社名	小間数	出展会社名	小間数
アーム(株)	2	兼松エレクトロニクス(株)	3
(株)アイヴィス	3	(株)キー・ブリッジ	1
IPCORE TECHNOLOGIES	1	キャッツ(株)	1
Artlogic Inc.	...	(株)グラフィン	1
アクテルジャパン(株)	4	コーウェア(株)	8
アジレント・テクノロジー(株)	2	Golden Gate Technology	1
(株)アストロン	1	サイバネットシステム(株)	2
(株)アプライド・シミュレーション・テクノロジー	1	サミット・デザイン・ジャパン(株)	6
(株)アプリスター	1	CQ 出版(株)	1
アルティウム ジャパン(株)	4	シーケンスデザイン(株)	2
(株)アルティマ	1	(株)ジーダット	6
アンソフト・ジャパン(株)	1	SIERRA DESIGN AUTOMATION KK	2
(株)礎デザインオートメーション	1	(株)システム・ファブリケーション・テクノロジーズ	2
伊藤忠テクノサイエンス(株)	9	ジャスパー・デザイン・オートメーション・インク	2
サン・マイクロシステムズ(株)	...	Silicon Integration Initiative (Si2)	1
日本ネットワーク・アライアンス(株)	...	シリコンソーシアム(株)	1
イノテック(株)	9	ステディデザイン(株)	...
バインドキーテクノロジーズ	...	(株)デジアン・テクノロジー	...
ソニックス	...	テスト・リサーチ・ラボラトリーズ(株)	...
チップビジョンデザインシステムズ	...	SILICON DESIGN SYSTEMS, INC.	1
ターゲットコンパイラーテクノロジーズ	...	(株)シルバコ・ジャパン	9
サガンテック	...	シンプリシティ(株)	12
ビーディーエフソリューションズ	...	(株)図研	8
オライダス	...	StarCore LLC	1
ジャズセミコンダクター	...	住商エレクトロニクス(株)	3
トライシーエヌ	...	Averant Inc.	...
(株)イー・イー・ティー・ジャパン	1	Carbon Design systems	...
(株)エスケーエレクトロニクス	2	Aprio Technologies	...
(株)エッチ・ディー・ラボ	4	FishTail Design Automation Inc.	...
NEC エレクトロニクス(株)	10	Pulsic Limited	...
NTT アドバンステクノロジー(株)	4	Xpedion Design Systems, Inc.	...
LSI ロジック(株)	6	AmmoCore Technology, Inc.	...
(株)沖ネットワークエルエスアイ	2	(株)ソリトンシステムズ	4
(株)ダイヘン	2	フォルテ・デザイン・システムズ(株)	2
CAST, Inc.	...	富士通(株)	6

出展会社名	小間数	出展会社名	小間数
巧テクノロジー(株)	1	プラットフォームコンピューティング(株)	1
立野電脳(株)	1	プロトタイピング・ジャパン(株)	2
GIDEL 社	...	VeriSilicon, Inc.	1
Catalyst 社	...	ベリシティ・デザイン(株)	8
タナーリサーチジャパン(株)	4	マクニカネットワークス(株)	1
ディー・クルー・テクノロジーズ(株)	1	マグマ・デザイン・オートメーション(株)	6
デナリソフトウェア(株)	6	丸紅ソリューション(株)	12
Tenison EDA	1	(株)ミッシュインターナショナル	2
テンシリカ(株)	1	三菱電機エンジニアリング(株)	1
(株)電波新聞社	1	メイトリックス ワン(株)	2
TOOL(株)	2	新日鉄ソリューションズ(株)	...
東京エレクトロン デバイス(株)	2	メンター・グラフィックス・ジャパン(株)	25
(株)東芝 セミコンダクター社	4	LIBRARY TECHNOLOGIES, INC	1
日経 BP 社	1	ラティスセミコンダクター(株)	1
日本システムウェア(株)	1	リード・ビジネス・インフォメーション(株)	1
日本イヴ(株)	2	菱洋エレクトロ(株)	2
日本ケイデンス・デザイン・システムズ社	20	(株)ロッキー	1
イノテック(株)	...	合計 119 社 / 336 小間	※会社名 50 音順 / 共同出展は文字下げ表記
日本シノプシス(株)	30		
日本セロックシカ(株)	4		
日本テラ・システムズ(株)	4		
日本トランス EDA(株)	2		
日本ノーベル(株)	1		
ACE Associated Compiler Experts bv	...		
ノバフロー(株)	4		
Novas Software, Inc.	...		
Silicon Canvas, Inc.	...		
(株)半導体理工学研究センター	10		
(株)先端 SoC 基盤技術開発	...		
福華先進微電子股份有限公司	1		

3.1.5 出展傾向

EDSFair2005 は、2004 年の後半からの I T 産業の景気回復に伴い、半導体世界市場も回復傾向となり、前回の開催規模（小間数）を約 10% 増となった。この回復傾向は、2004 年の開催規模を大きく上回り、小間数においては、2003 年の規模をも上回る回復となりました。

	出展者数	小間数
2005 年	119 社	336 小間
2004 年	105 社	306 小間
2003 年	99 社	320 小間

3.1.6 出展者セミナー

1 セッション 45 分間で、30～100 名の適正人数のお客様に向けて集中 PR が行える出展者セミナールームを提供。11 会場・121 セッションで開催し盛況であった。総聴講者数：3,431 名
(前年 116 セッション、総聴講者数 3,087 名)

3.1.7 ユニバーシティ・プラザ

産学の交流を促進するとともに大学機関による研究の成果を発表する場として、今回は 16 大学 21 研究室（19 ブース）が、設計技術に関する研究成果を実演により発表した。（前年 15 大学 20 研究室（19 ブース））

■発表内容／研究室名

- ・カスタム・プロセッサ設計環境 ASIP Meister
大阪大学 大学院 情報科学研究科 情報システム工学専攻 今井 正治 教授 研究室
- ・次世代情報家電・マルチメディア機器向け技術開発
大阪大学 大学院 情報科学研究科 情報システム工学専攻 尾上 孝雄 教授 研究室
- ・ユビキタスメディア応用低消費電力システム LSI
金沢大学 工学部 電気電子システム工学科 吉本 雅彦 教授 研究室
神戸大学 工学部 情報知能工学科 研究室
- ・システム LSI の高性能化、低消費エネルギー化設計技術に関する研究紹介
九州大学 システム LSI 研究センター 安浦 寛人 センター長 研究室
- ・システム LSI のテスト・診断技術
九州工業大学 情報工学部 電子情報工学科 梶原 誠司 教授 研究室
- ・リコンフィギャラブルシステム RICE と C 言語ベース設計環境
九州工業大学 情報工学部 知能情報工学科 佐藤 寿倫 助教授 研究室
九州産業大学 情報科学部 知能情報学科 有田 五次郎 教授 研究室
- ・LUT カスケードの設計システム
九州工業大学 情報工学部 電子情報工学科 笹尾 勤 教授 研究室
- ・リコンフィギャラブルシステム技術と FPGA 応用
熊本大学 工学部 数理情報システム工学科 末吉 敏則 教授 研究室

-
- ・リコンフィギャラブルシステムのバイオインフォマティクスへの応用
慶應義塾大学 理工学部 情報工学科 天野 英晴 教授 研究室
 - ・デジタル LSI の電源系雑音評価技術と適用事例
神戸大学 工学部情報知能工学科 永田 真 助教授 研究室
 - ・システムレベル設計・検証技術
東京大学 工学系研究科電子工学専攻 藤田 昌宏 教授 研究室
 - ・東京大学大規模集積システム設計教育研究センターの活動報告紹介
東京大学 大規模集積システム設計教育研究センター 浅田 邦博 教授 研究室
 - ・次世代モバイルインターネット端末の開発
東北大学 電気通信研究所 21 世紀情報通信研究開発センター
坪内 和夫 センター長、教授 研究室
 - ・FPGA ボードでファミコン
長崎大学 工学部 情報システム工学科 小栗 清 教授 研究室
 - ・組込みシステム向けシステムレベル設計環境
名古屋大学 大学院情報科学研究科 高田 広章 教授 研究室
 - ・多層プリント配線基板設計支援システム MULTI-PRIDE
広島大学 大学院工学研究科 情報工学専攻 渡邊 敏正 教授 研究室
 - ・小面積マルチポートメモリとそれを用いた高性能アーキテクチャの実現
広島市立大学 情報科学部 情報工学科 弘中 哲夫 助教授 研究室
 - ・軽リソースで奥行き感を与える画像生成チップの開発 -基本構成とリソース割り当て-
福井大学 ベンチャー・ビジネス・ラボラトリー
(西門 秀人、吉田 智、高澤 剛、福間 慎治、桜井 哲真)
 - ・LUT カスケードの応用
明治大学 理工学部情報科学科 井口 幸洋 助教授 研究室

3.1.8 キーノートスピーチ

- (1) タイトル：「FPGA の将来とストラクチャード ASIC ビジネスのリスク」
新たなコア・シリコン時代の到来～
- (2) 講 師：アイサプライ・ジャパン株式会社 代表取締役社長 豊崎 禎久 氏
- (3) 日 時：1月27日(木)午前10時30分～午前11時30分
- (4) 場 所：パシフィコ横浜 アネックスホール
- (5) 聴 講：無料
- (6) 聴講者数：286名（前年292名）

いくつかの LSI が開発されている今日において、ASIC、FPGA、ASSP についての競合性などを中心に紹介し盛況であった。

3.1.9 第 12 回 FPGA/PLD Design Conference

- (1) 日 時：1月27日（木）・28日（金）
-

(2) 場 所：パシフィコ横浜 アネックスホール

(3) 聴 講 料：1セッション受講券（1回券）5,000円

5セッション受講券（回数券）15,000円

(4) 聴講者数：10セッション総入場者 591名（前年 605名）

FPGA/PLD Design Conference は、FPGA/PLD に関する最新技術、設計事例、ビジネスおよび将来動向を包括的に知る事ができるコンファレンスとして、特に今回は、様々な設計課題が影響しあうシステム設計において重要となるボーダレスな設計技術、量産デバイスとしての飛躍と脱皮、FPGA の将来像を注目すべきトピックとして講演を充実させ盛況であった。

(5) 各セッション概要

[1月27日（木）]

・セッション1【入門編】 12:50-14:20「HDL とデジタル回路設計 転ばぬ先の基礎」

ミカミ コンサルティング 三上 廉司 氏

・セッション2【解決編】 12:50-14:20「C ベース設計の実態」

設計コンサルタント 鮫島 正裕 氏

・セッション3【入門編】 15:20-16:50「FPGA 回路設計擬似体験」

東海大学 電子情報学部 コミュニケーション工学科 教授 博士（工学） 清水 尚彦 氏

・セッション4【将来編】 15:20-16:50「遂に登場！動的再構成可能なアナログデバイス」

オムロン(株)エレクトロニクスコンポーネンツビジネスカンパニー

セミコンダクタ統括事業部木村桂吾 氏

・セッション5【解決編】 10:00-11:30「デザインヒント ～間違いのない FPGA 設計の進め方～」

三菱電機エンジニアリング(株)鎌倉事業所 電子技術センター 電子応用課

課長 立崎 賢治 氏

[1月28日（金）]

・セッション6【解決編】 10:00-11:30「基板設計におけるノイズ対策事例」

(株)トッパン NEC サーキットソリューションズ 設計技術部 係長（電気特性シミュレーショングループ） 金子 俊之氏

・セッション7【解決編】 12:50-14:20「FPGA を用いた、短 TAT、低コストを実現する設計手法とその応用」

シャープ(株)デジタル家電開発本部 次世代商品開発センター 主事 尾松 智裕 氏

・セッション8【解決編】 12:50-14:20 「高速信号伝送でのトラブル事例とその解決法」

(株)マクニカ 技術開発センター センター長 碓井 有三 氏

・セッション9【解決編】 15:20-16:50 「動き始めた量産 FPGA ～デジタル家電、カーナビ採用事例～」

松下電器産業(株) システム AV ビジネスユニット 購買チーム 中司 昭仁 氏

(株)エイチ・シー・エックス 第二開発部 主任技師 栗原 孝男 氏

・セッション10【将来編】 15:20-16:50「FPGA による数値計算の高速化」

独立行政法人 理化学研究所 戎崎計算宇宙物理研究室 協力技術員 濱田 剛 氏

3.1.10 PGA/PLD Design Conference ユーザプレゼンテーション

- (1) 日 時：1月27日（木）・28日（金）
- (2) 場 所：パシフィコ横浜 アネックスホール
- (3) 聴 講料：無料
- (4) 総聴講者数：120名（前年154名）

12件のプレゼンテーションにより、FPGA/PLD に関する設計事例や応用事例、および FPGA/PLD に関する最新の研究や萌芽的アイデアについての意見交換や議論が活発に行われた。さらに、本年度も聴衆による人気投票により優秀プレゼンテーションを選出し表彰する。

3.1.11 PGA/PLD Design Conference IPフリーマーケット IN EDSFair

- (1) 日 時：1月27日（木）13:50～16:10 プレゼンテーションおよびポスターセッション
1月28日（金）EDSFair2005 会場内にてポスター展示
- (2) 場 所：アネックスホール F204 号室
- (3) 聴 講：無料
- (4) 27日総聴講者数：91名（前年58名）

21件の投稿(前年13件)によるプレゼンテーションとポスターセッション、および、EDSFair2005 会場内でのポスター展示により、IPベンダーやベンチャーが開発した商用IPや、大学や研究室で開発したIPなど、あらゆる設計者が開発したIPについての発表の場とその価値を見極める場として開催され盛況であった。

3.1.12 PGA/PLD Design Conference 有料聴講者特典パネルセッション

- (1) 日 時：1月27日（木）17:30～19:30
- (2) 場 所：アネックスホール F204 号室
- (3) 聴 講：無料 ※FPGA/PLD Design Conference 有料セッション申込者のみ聴講可
- (4) 聴講者数：34名（前年63名）

設計者、デバイスベンダーによるパネルセッションとして開催され、前回にも増し活発な意見交換が行われ盛況であった。

3.1.13 場者数詳細

2005年入場者数

1月27日（木）	晴れ	5,066名
1月28日（金）	晴れ	6,087名
合 計		11,153名

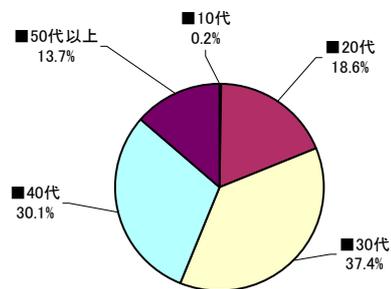
過去の来場者数

開催年	1日目	2日目	合計
2005年	5,066名	6,087名	11,153名
2004年	4,764名	6,023名	10,787名
2003年	5,095名	6,445名	11,540名
2002年	4,324名	6,227名	10,551名
2001年	5,048名	6,839名	11,887名

3.1.14 来場者傾向（全来場者入場登録票アンケート回答 集計結果）

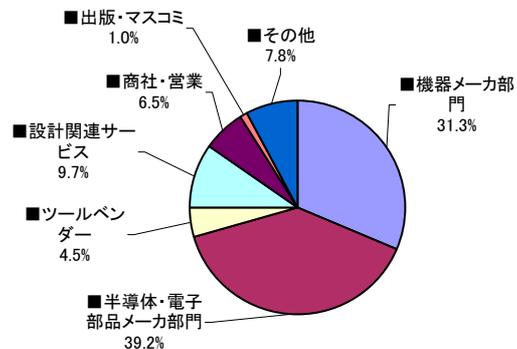
■年代

	2005 (今回)	2004 (前回)
■10代	0.2%	0.1%
■20代	18.6%	20.3%
■30代	37.4%	37.8%
■40代	30.1%	28.7%
■50代以上	13.7%	13.1%



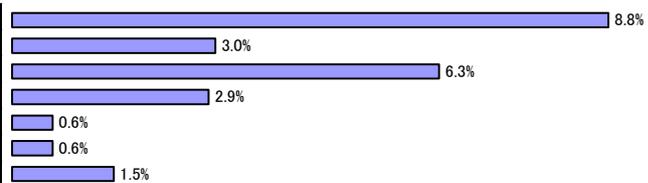
■業種

	2005 (今回)	2004 (前回)
■機器メーカー部門	31.3%	27.6%
■半導体・電子部品メーカー部門	39.2%	42.6%
■ツールベンダー	4.5%	4.2%
■設計関連サービス	9.7%	11.5%
■商社・営業	6.5%	6.4%
■出版・マスコミ	1.0%	0.9%
■その他	7.8%	6.8%

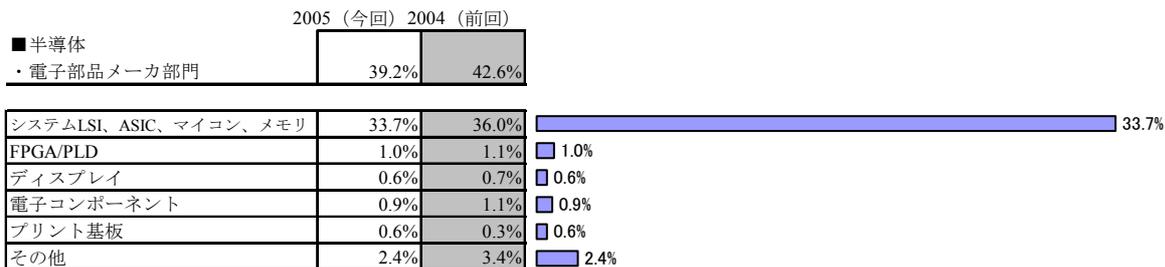


EDSFair2005来場者業種 機器メーカー部門の詳細

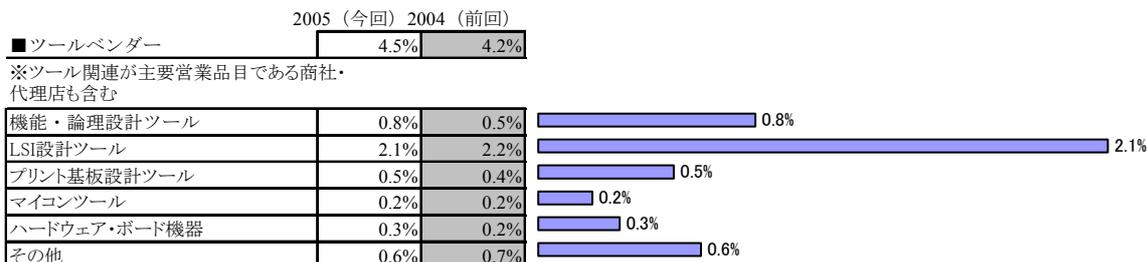
	2005 (今回)	2004 (前回)
■機器メーカー部門	31.3%	27.6%
コンピュータ関連機器	8.8%	6.8%
ネットワーク関連機器	3.0%	2.7%
一般民生機器	6.3%	5.1%
画像処理機器	2.9%	3.2%
医療機器	0.6%	0.6%
アミューズメント	0.6%	0.3%
自動車・輸送機器	1.5%	1.6%



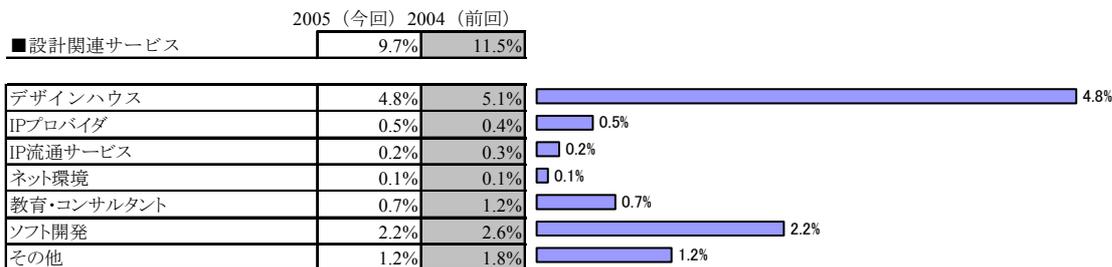
EDSFair2005来場者業種 半導体・電子メーカー部門の詳細



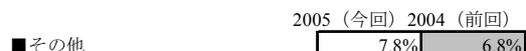
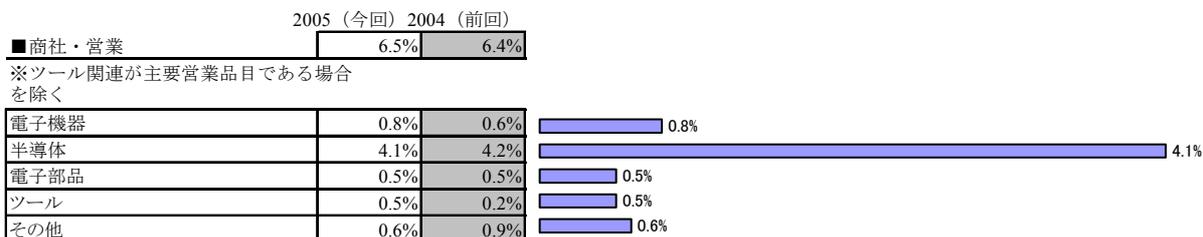
EDSFair2005来場者業種 ツールベンダーの詳細



EDSFair2005来場者業種 設計関連サービスの詳細

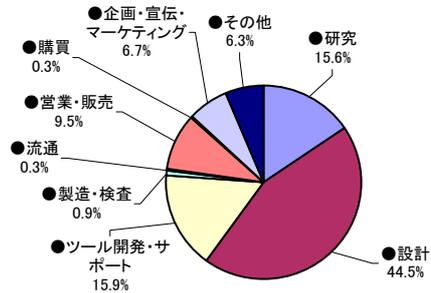


EDSFair2005来場者業種 設計関連サービスの詳細



■職務

	2005 (今回)	2004 (前回)
●研究	15.6%	12.9%
●設計	44.5%	44.1%
●ツール開発・サポート	15.9%	20.8%
●製造・検査	0.9%	0.9%
●流通	0.3%	0.2%
●営業・販売	9.5%	8.4%
●購買	0.3%	0.3%
●企画・宣伝・マーケティング	6.7%	6.1%
●その他	6.3%	6.3%



EDSFair2005来場者職務 研究の詳細

	2005 (今回)	2004 (前回)
●研究	15.6%	12.9%

研究分野	2005 (今回)	2004 (前回)
情報・EDA系	9.8%	7.7%
物理・物性系	0.9%	0.9%
電子応用	2.9%	2.9%
その他	2.0%	1.4%

EDSFair2005来場者職務 設計の詳細

	2005 (今回)
●設計	44.5%

設計分野	2005 (今回)
システムレベル設計・検証	13.3%
機能 (RTL) 設計・検証	8.7%
論理 (ゲートレベル) 設計・検証	2.5%
デジタルIC設計	3.8%
アナログIC設計	2.7%
テスト技術	1.4%
カスタムIC設計	1.7%
レイアウト検証	1.6%
FPGA/PLD設計	1.9%
PCB設計	1.0%
IC Package設計	0.2%
SiP設計	0.1%
リソ/マスク/プロセス/製造	0.6%
IPマクロ	0.6%
装置実装	0.3%
ソフトウェア・ファームウェア設計	2.1%
その他	2.0%

EDSFair2005来場者職務 ツール開発・サポートの詳細

	2005 (今回)	2004 (前回)
●ツール開発・サポート	15.9%	20.8%

ツール開発・サポート分野	2005 (今回)	2004 (前回)
システム設計	3.2%	2.9%
LSI設計	8.9%	12.7%
FPGA/PLD支援	0.9%	1.2%
プリント基板設計	0.8%	0.7%
マイコンツール	0.2%	0.4%
IP流通	0.3%	0.2%
ASP・ネット環境	0.1%	0.2%
教育・教材	0.3%	0.7%
その他	1.2%	1.8%

	2005 (今回)	2004 (前回)
●製造・検査	1.0%	0.9%

	2005 (今回)	2004 (前回)
●流通	7.8%	6.8%

	2005 (今回)	2004 (前回)
●営業・販売	1.0%	0.9%

	2005 (今回)	2004 (前回)
●購買	7.8%	6.8%

	2005 (今回)	2004 (前回)
●企画・宣伝・マーケティング	1.0%	0.9%

	2005 (今回)	2004 (前回)
●その他	1.0%	0.9%

■ご来場目的 (複数回答)

2005 (今回) 2004 (前回)

●展示関連	88.5%	74.7%
●セミナー/カンファレンス	11.5%	25.3%

2005 (今回)

●展示関連	88.5%
-------	-------

システムLSI	9.4%
ASIC・ASSP	6.2%
MPU/MCU/DSP	3.3%
FPGA/PLDデバイス	6.0%
システムレベル設計 (RTLより高位)	6.0%
論理設計 (RTL～ネットリスト)	5.2%
論理検証	5.1%
アナログ設計・検証ツール	3.3%
ICレイアウトツール	3.4%
ICレイアウト検証・解析	3.1%
IC信号解析	1.5%
回路図作成	1.9%
PCBレイアウト	1.2%
PCB信号解析	1.2%
部品検索/管理ツール	0.5%
統計設計 (SiP) 関連ツール	0.9%
LSIテスト	0.9%
デザインフォーテストツール	1.0%
PCBテスト	0.2%
計測器	0.9%
IPコア	2.7%
セイライブラリ	0.8%
メモリ	1.1%
組込みOS	1.6%
デバイスドライバ	0.5%
ファームウェア/ミドルウェア	0.8%
リコンフィギャブルプロセッサ	1.5%
ICE	0.6%
デバッカ	1.0%
マイコンCASE	0.3%
コンパイラ/クロスコンパイラ	0.7%
シュミレータ	3.3%
ハードウェア/ソフトウェア 協調開発環境	2.7%
デザインセンタ・デザインサービス	1.0%
設計コンサルティング	1.1%
IP流通サービス	0.7%
設計プラットフォーム	2.2%
システム管理ツール	1.1%
出版物	0.4%
スイートデモ	1.0%
ユニバーシティ・プラザ	0.5%
その他	1.7%

2005 (今回)

●セミナー/カンファレンス	11.5%
---------------	-------

出展者セミナー	9.4%
キーノートスピーチ	6.2%
第12回 FPGA/PLD Design Conferene	3.3%
システム・デザイン・フォーラム2005	6.0%

3.1.15 EDSFair2005 実行委員会メンバ

委員長	藤波 義忠	NEC エレクトロニクス(株)
副委員長	弘中 哲夫	広島市立大学
副委員長	松沢 洋	日本ケイデンス・デザイン・システムズ社
委員	井上 幸一	イノテック(株)
〃	水尾 学	(株)エスケーエレクトロニクス
〃	灘岡 満	沖電気工業(株)
〃	大沢 敏幸	(株)図研
〃	須合 和成	NTT アドバンステクノロジー(株)
〃	遠藤 裕之	丸紅ソリューション(株)
〃	浅沼 義功	メンター・グラフィックス・ジャパン(株)
〃	水上 明彦	セイコーインスツルメンツ(株)
〃	深井 敦子	(株)ソリトンシステムズ
〃	立石 昭光	(株)東芝 セミコンダクター社
〃	早田 雅彦	日本アルテラ(株)
〃	川原 常盛	コーウェア(株)
〃	勝山 義樹	サイバネットシステム(株)
〃	松本 仁	三菱電機(株)
〃	和田 敦子	日本セロックシカ(株)
〃	佐々木則之	Applied Wave Research Inc
〃	和田 昇平	伊藤忠テクノサイエンス(株)
〃	三井田純美子	マグマ・デザイン・オートメーション(株)
事務局	穂苅 泰明	(社)電子情報技術産業協会
〃	木暮 英男	(社)電子情報技術産業協会

3.2 システム・デザイン・フォーラム 2005

JEITA EDA 技術専門委員会の最新の EDA 技術の業界内の普及・促進活動の一環として当委員会主催、ASP-DAC 2006 実行委員会協賛で「システム・デザイン・フォーラム 2005」を開催した。本フォーラムではシステム・LSI 設計者、EDA 技術者に最新の設計技術、課題など今一番ホットな話題、設計事例の紹介、ディスカッションをした。

EDA 技術専門委員会は、これまで EDA 標準化活動の発表とその一般への普及を図ることを目的とする“EDA 標準化フォーラム”を 1990 年から 1994 年にかけて 4 回開催し、EDA 技術専門委員会の活動に係る内容の発表、討論の場を目的とする“EDA フォーラム”を 1999 年から 2002 年にかけて 2 回開催してきた。昨年は、最新の設計技術、課題を設計事例とともに紹介する“システム・デザイン・セミナー”を EDSFair2004 と同時開催した。

今回は、1 日目に EDA 標準化活動の一環として、SystemVrilog, SytemC のユーザフォーラム、および関連のセッションを行い、2 日目に SoC に関連した設計技術、課題等を設計事例とともに紹介する“システム・デザイン・フォーラム 2005”として 2 日間の日程で、EDSFair2005 と同時

開催した。

日 時：2005年1月27日（木） 12:30～17:00

場 所：パシフィコ横浜 アネックスホール F202

聴講料：無料

定 員：200名／セッション

1月27日のみの協賛：Accellera、OSCI、コーウェア㈱、日本ケイデンス・デザイン・システムズ社、日本シノプシス㈱、フォルテ・デザイン・システムズ㈱、メンター・グラフィックス・ジャパン㈱

内 容：テクニカルセッション（3セッション）

日 時：2005年1月28日（金） 9:30～17:00

場 所：パシフィコ横浜 アネックスホール F204

聴講料：5,250円（消費税込み）／セッション

定 員：100名／セッション

内 容：テクニカルセッション（3セッション）

■セッション1（1月27日 12:30～14:10）

テーマ：SystemVerilog ユーザフォーラム 2005

概要：SystemVerilog は Verilog HDL（IEEE 1364）後継の次世代の設計（実装）／検証用言語として注目を集めており、さらなる設計効率化をもたらす言語として期待されている。本セッションでは SystemVerilog の標準化を進めている Accellera の Dennis Brophy 氏を招き、IEEE 標準化状況および米国での利用状況について解説した。次に JEITA SystemVerilog タスクグループのメンバーがユーザの立場でまとめた SystemVerilog 3.1a に関する言語チュートリアルを行った。また、SystemVerilog を利用した最新の設計事例を紹介した。

オーガナイザ：JEITA SystemVerilog タスクグループ

司 会：浜口 加寿美氏（松下電器産業）

- ① Dennis Brophy 氏（Chairman, Accellera）：SystemVerilog 標準化情報
- ② Oz Levia 氏（Chairman 代理、IEEE P1800）：SystemVerilog 標準化情報
- ③ 土屋 丈彦氏（東芝 セミコンダクター社）：SystemVerilog V3.1a 言語チュートリアル
- ④ Stuart Sutherland 氏（Sutherland HDL, Inc）：SystemVerilog from a User's Perspective

■セッション2（1月27日 14:15～15:15）

テーマ：EDA ベンダーセッション

概要：EDA ベンダー各社より、SystemVerilog および SystemC のサポート状況およびサポートロードマップを紹介した。

司 会：千綿 幸雄氏（富士通）

- ① 中谷 英峰氏（株式会社図研）：

-
- ② 明石 貴昭氏（日本シノプシス）：
 - ③ 柴下 哲氏（メンター・グラフィックス・ジャパン）：
 - ④ 後藤 謙治氏（日本ケイデンス・デザイン・システムズ社）
 - ⑤ 川原 常盛氏（コーウェア）
 - ⑥ 桜井 至氏（フォルテ・デザイン・システムズ）

■セッション3（1月27日 15：20～17：00）

テーマ：SystemC ユーザフォーラム 2005

概要：SystemC は、C 言語ベースのシステムレベル設計言語として既に業界標準として幅広く利用されている。また、待望の IEEE における標準化作業も本格的にスタートしている。本セッションでは SystemC の標準化を進めている OSCI の中核メンバーとして活躍されている NEC Electronics America の河原林氏を招き、IEEE 標準化状況およびロードマップ、米国/欧州での利用状況について解説した。次に各社より SystemC を利用した最新の実践的設計事例を紹介した。

オーガナイザ：JEITA SystemC タスクグループ

司会：長谷川 隆氏（富士通）

- ① 河原林 政道氏（NEC Electronics America）：IEEE 標準化状況および SystemC 最新状況
- ② 高嶺 美夫氏（ルネサステクノロジ）：短 TAT 性能評価プラットフォームとその応用
- ③ 塚本 泰隆氏（リコー）：動作合成ツールを使った JPEG（DCT）回路の設計事例
- ④ 柿本 勝氏（ソニー）：動作合成の設計適用事例とその効果

■セッション4（1月28日 9：30～11：30）

テーマ：最先端 SoC の設計事例

概要：携帯電話、アミューズメント、DVD レコーダ、デジタルスチルカメラといった日本の代表的なエレクトロニクス・システムは、SoC というキーデバイスの中に付加価値が凝縮されている。本セッションでは、このような最先端 SoC の設計事例を紹介した。画像処理の問題や IP 間のデータ転送の問題等を、高性能・低コスト・低消費電力・TAT 短縮の観点から検討した結果としての SoC アーキテクチャのポイント、設計・開発上のチャレンジ、および、それらを解決していくための性能見積もりやシステム検証をはじめとする設計手法について紹介した。

オーガナイザ・司会：広瀬 文保氏（日本ケイデンス・デザイン・システムズ社）

- ① 竹本 卓氏（東芝 セミコンダクター社）：携帯向けメディアプロセッサ SoC “T4G” の設計事例
- ② 服部 俊洋氏（ルネサステクノロジ）：アミューズメント向け AV/グラフィックス SoC “SH7307” の設計事例
- ③ 久松 裕二氏（NEC エレクトロニクス）：DVD レコーダ用 SoC “ μ PD61175” の設計事例
- ④ 飯野 秀之氏（富士通）：デジタルスチルカメラ用画像処理 SoC の設計事例

■セッション5 (1月28日 12:30~14:30)

テーマ：組み込みシステム開発・検証技術の最新動向

概要：日本の民生用エレクトロニクス・システムの主力を占める家電製品、携帯電話、デジタルカメラ、セットトップ・ボックスなどの組み込みシステムには、汎用プロセッサ、DSP、ASIC が含まれているだけではなく、汎用プロセッサや DSP 上で実行されるソフトウェアが含まれており、非常に複雑なシステムとなりつつある。その結果、開発工数の増大と開発期間の長期化が大きな問題になりつつある。

本セッションでは、これらの問題を解決するための新しいシステム開発手法と設計検証技術を紹介した。設計検証技術に関しては、現在注目を集めているアサーションベースの検証技術を中心に解説を行った。次にシステム開発手法として、企業におけるプラットフォームベース設計手法の適用事例を2件紹介した。

オーガナイザ：今井 正治氏 (大阪大学)、中田 恒夫氏 (富士通研究所)

司会：今井 正治氏 (大阪大学)

- ① 東野 輝夫氏 (大阪大学)：設計検証技術の最新動向 - アサーションベース設計
- ② 水野 雅信氏 (松下電器産業)：デジタル家電分野のシステム LSI 検証の取り組み
- ③ 山下 浩一郎氏 (富士通)：組み込みシステム向けプラットフォーム非依存

システム開発環境 SPD

■セッション6 (1月28日 15:00~17:00)

テーマ：チップ・パッケージ・ボード統合設計

概要：LSI の電源電圧低下によるノイズマージンの減少や LSI 間通信の高速化にともない、チップ・パッケージ・ボードのリワーク頻度が増加している。また、リワークを回避するための過剰品質によるコスト増が懸念されている。リワーク件数を削減し、要求性能を満たす範囲でギリギリまでコストを低下させるためには、従来の個別検証では不十分であるため、チップ・パッケージ・ボード統合設計が脚光を浴びつつある。統合設計のための環境開発は、現時点では設計後の最終検証に重点が置かれているが、今後は設計初期段階での協調設計へと移っていこうとしている。本セッションでは、チップ設計、パッケージ設計、ボード設計の各領域の専門家と統合解析の専門家を招き、パネル討論を行った。設計・解析事例の紹介や、課題の明確化、解決手法の提案を通して統合設計のあるべき姿を議論した。

オーガナイザ：JEITA PDM 研究会主査 蜂屋 孝太郎氏 (NEC エレクトロニクス)

モデレータ：浅井 秀樹氏 (静岡大学)

パネリスト：

- ① 村山 敏夫氏 (ソニーイーエムシーエス)
- ② 須藤 俊夫氏 (東芝)
- ③ 渡辺 毅氏 (NEC エレクトロニクス)
- ④ 佐藤 敏郎氏 (富士通)

当日の聴講者数は以下の通り。

	聴講者実数
セッション 1	212
セッション 2	244
セッション 3	265
セッション 4	108
セッション 5	70
セッション 6	84
合計	983

各セッションとも熱の入った発表、討論となり盛況であった。各セッションの状況は下記のとおり。

■セッション 1

SystemVerilog Task Group 主査であり、司会者である浜口 加寿美氏のあいさつにて当セッションは開始した。セッション開始時からほぼ満席に近く、SystemVerilog に対する注目度の高さがうかがえた。Accellera Chairman である Dennis Brophy 氏より「Accellera での標準化と米国ユーザーの声」と題した講演で始まった。Dennis 氏はまず Accellera の組織／目的の説明を行い、そのポジションを明確にした。その後、SystemVerilog に関する概要とその歴史、現状を説明し、最後に米国におけるユーザの

様々な声を紹介した。次に、IEEE P1800 の chairman である Johnny Srouji 氏の代理として Synopsys Inc. の Oz Levia 氏より「IEEE P1800 としての標準化状況」と題した講演を行った。Oz 氏は IEEE P1800 の組織／目的、現在までの標準化作業の経過および状況を説明し、2005 年 9 月に



IEEE 1800 として承認されるだろうと述べた。次に、株式会社東芝の土屋 丈彦氏が「SystemVerilog 3.1a 言語チュートリアル」と題して講演を行った。土屋氏は JEITA SystemVerilog Task Group の委員であり、このチュートリアルは同 Task Group の成果として発表した。当チュートリアルは「SystemVerilog 3.1a Language Reference Manual から各 EDA ベンダーにてサポートされている文法を抽出し、その中でユーザの立場から有用であると思われる記述を Pick Up したものである」と

の方針を最初に示し、現時点で実際にデザインに適用可能な記述を紹介した。また、EDA ベンダーに早期にサポートを期待する記述についても併せて紹介した。土屋氏がポイントをついた解説を行っていたこと、また唯一の日本語での講演であったこともあり、聴講者の反応もよく、盛んにメモを取っていた。最後に Sutherland HDL, Inc. の Stuart Sutherland 氏から「SystemVerilog from a User's Perspective」と題し、ユーザ視点からの SystemVerilog の有効な活用法について講演を行った。Stuart 氏は設計者向け、検証者向けと大きく二つにカテゴリをわけ、それぞれの各項目について従来の VerilogHDL と比較ながら、SystemVerilog を使用することについての効果を示した。最後に Stuart 氏は SystemC と SystemVerilog の競合について言及した。Stuart 氏は「2 言語は役割が異なり、十分すみ分けが可能である」との見解を示し、講演を締めくくった。会場では立ち見の方がでるなど、最後まで盛況であった。

■セッション 2

当セッションはセッション 1:「SystemVerilog ユーザフォーラム」、セッション 3:「SystemC ユーザフォーラム」に対し協賛していただいた 6 社の EDA ベンダーからツールサポート状況等についての講演を行った。まず、株式会社図研の中谷英峰氏から「動き始めた SystemVerilog -SpyGlass の SV 対応の現状とロードマップ」と題し、Atrenta 社 SpyGlass の SystemVerilog サポート状況についての紹介があった。次に日本シノプシス株式会社の明石貴昭氏から「シノプシスの SystemC/SystemVerilog への取組み」と題し、シノプシス社の SystemC/SystemVerilog に対する取組み、ツール対応状況についての説明があった。メンター・グラフィックス・ジャパン株式会社の柴下哲氏から「2 言語を上手に使い分け、全体の設計期間を大幅短縮」と題して講演を行った。柴下氏は SystemC, SystemVerilog のそれぞれの得意分野、使い分け、それに対するソリューションについての解説を行った。次に日本ケイデンス・デザイン・システムズ社の後藤謙治氏より「Incisive による SystemC & SystemVerilog 混在記述の効率的検証」と題し講演があった。後藤氏は SystemC/SystemVerilog 混在環境に対する統合検証ソリューションについて紹介した。コーウェア株式会社の川原常盛氏からは「コーウェアが提供する SystemC ベース統合設計環境」と題し、SoC の設計の問題点、それに対する SystemC を使用した設計検証ソリューションについての説明があった。最後に「SystemC からの動作合成ツール Cynthesizer の紹介」と題し、フォルテ・デザイン・システムズ株式会社の桜井至氏より説明があった。桜井氏はフォルテ社の動作合成ツールを用いた設計フローとサポート状況について講演を行った。以上がセッション 2 の講演内容である。各社とも講演時間が 10 分と大変短く、慌しいセッションとなったが、聴講者が EDSFair のフロアに訪れるきっかけとなったのではと考える。

■セッション 3

最初に河原林政道氏 (NEC Electronics America) より、OSCI の代表として SystemC の IEEE 標準化状況と最新情報の説明があった。IEEE に言語仕様が移管され、順調に標準化が進んでいる旨報告され、また OSCI による最新リリースである SystemC 2.1 やトランザクションレベルモデリングおよび合成サブセットの標準化状況について紹介された。次に高嶺美夫氏 (ルネサステクノロジ) より、短 TAT 性能評価プラットフォームとその応用について説明された。性能評価、機能検

証、ソフトウェア開発への応用と多岐に渡る SystemC の活用事例の紹介があった。続いて塚本泰隆氏（リコー）より、動作合成ツールを使った JPEG（DCT）回路の設計事例の説明があった。ANSI-C から SystemC への書き換えの具体例や、パイプライン合成を行う方法の紹介や、一度 SystemC の動作合成を経験すると RTL 設計に戻れない、という熱のこもった説明もあった。最後に柿本勝氏（ソニー）より、動作合成の設計適用事例とその効果について説明があった。CE 機器向けに必要なとされる設計手法の内、動作合成の重要性について説かれ、それを適用して設計された製品群の紹介もあった。時間が押しているにもかかわらず各発表の後に質問時間を設けたが、それぞれ 1~2 件の質問があり、活発な意見交換の場となった。

■セッション 4

デジタル家電のビジネスは、従来の家電および IT/PC の事業領域を超えた融合のビジネスといわれるように、最先端 SoC の設計事例においても融合が進んでいた。今回紹介された 4 つの設計事例は最終アプリケーションが異なるものの、例えば、画像処理では、動画・静止画・3D といった機能を SoC 内に融合している。かつては独自の HW として進化したであろうアプリケーション群を、一つの SoC にどのように融合させるか？どのようにメモリを始めとする HW 資源を共有して HW を有効利用し、かつ、共有の結果として狭くなる IP 間のデータバンド幅などの性能を確保するか？どのように、HW・SW を切りなおして設計に柔軟性を与え、全体仕様や SW などの再利用性を類似システムにおいて高めていくか、また、個別のアプリのスタンダードに配慮をしていくか？与えられたコストや開発期間を踏まえ、そのためにアーキテクチャとしていかなる決断をしたのか？そのためにシステムアーキテクチャをどのように最適化し検証をしたのか？そのための設計環境の実情は？今回の設計事例はこのような観点から、最先端の取り組みを現場の設計者の生の声で伺うことができ、また、横並びに眺めることができた。先端事情を垣間見るのに、真に有意義であった。

■セッション 5

本セッションでは、家電製品、携帯電話、DSC、STB などの電子システム（組込みシステム）の開発に必要な設計技術と設計検証の最新動向について、3 件の講演を実施していただいた。最初の東野先生の講演は、「設計検証技術の最新動向ーアサーションベース設計ー」と題して行われ、やや研究者向きの内容であった。しかし、2 番目、3 番目の講演である水野氏の「デジタル家電分野のシステム LSI 検証の取り組み」と山下氏の「組み込みシステム向けプラットフォーム非依存システム開発環境 SPD」は、設計現場での経験に基づいたプラットフォーム・ベース設計手法の紹介であり、設計者をターゲットにした講演内容であった。

■セッション 6

パネラー各位による 10~20 分/人のポジショントークで始まった。村山敏夫氏（ソニーイーエムシー）はデジタルビデオカメラなどの商品設計の立場から、チップ・パッケージ・ボード統合解析の目的や、主にボードレベルでの解析例を紹介した。各パーツのモデル入手が困難であることや、パーツの大きさに応じて支配方程式が異なるため分野毎に考え方・意識に差があることを

課題として指摘した。須藤俊夫氏（東芝）からはシグナルインテグリティ（SI）、パワーインテグリティ（PI）、電磁環境適合性（EMI）に関する測定とモデリング手法の説明があった。特に EMI は LSI やボードのみならず、ケーブルや筐体の影響も考慮する必要があり、解析が非常に



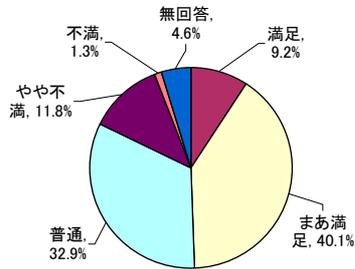
困難であることを指摘した。渡辺毅氏（NEC エレクトロニクス）は EMI ノイズの測定と解析手法を、実例を用いて紹介した。ノイズ励振源である LSI 単体の EMI 測定手法を国際規格化することに成功したことと、EMI 解析用の LSI モデル提供サービスを立ち上げたことで、装置・ボード設計者と LSI 設計者の間のコミュニケーション問題は解決されつつあると述べた。佐藤敏郎氏（富士通）は自社内の統合解析システムとそれを用いた解析事例を紹介した。過去 10 年に渡る内製によるシステム開発が完了し近年製品適用が始まったが、現状では社外製 LSI への適用に難があるため今後は LSI モデルの標準化や、解析システムの外販に力を入れる、という方針を示した。

後半のディスカッションでは会場から 2 件の質問があり、現状の統合解析システムは LSI モデルさえ入手できれば統合解析が可能なレベルに到達していること、LSI 設計者・ボード設計者間のコミュニケーションを容易にするために LSI のノイズ量の一般的な値や目標値が必要であることが確認・議論された。また、モデレータ浅井秀樹先生（静岡大）が、LSI 設計ツールが米国製ツールに席捲されたために国内 LSI 業界が低迷したと推測され、統合解析システムと家電・装置業界も同じ道を歩むのではないかと危惧を示した。しかしパネラーが、現状では米国ベンダー製ツールは実力不足であることが多く統合解析システムは内製ツール中心で構築されている、と反論し、まだ国内製ツールにチャンスがあることを訴えた。

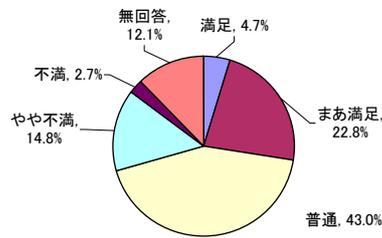
また今回は、次回以降の開催の参考とするために、各セッション終了時に聴講者にアンケートを記入いただいた。アンケートの主な質問事項は以下の通り。

- ・ このフォーラムを知ったきっかけ
- ・ 聴講セッションの満足度
- ・ 聴講セッションの時間配分
- ・ 今後取り上げて欲しいテーマ

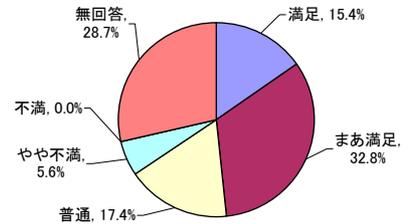
ここでは、各セッションの満足度について以下に示す。アンケートの結果は、次回の開催担当への引継ぎを行い、参考とする。



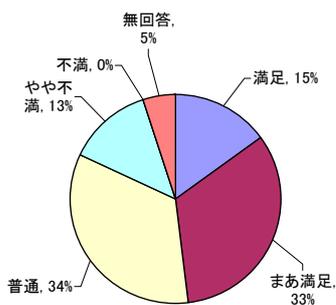
セッション 1
SystemVerilog
ユーザフォーラム 2005



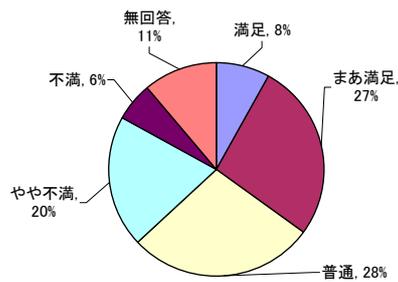
セッション 2
EDA ベンダーセッション



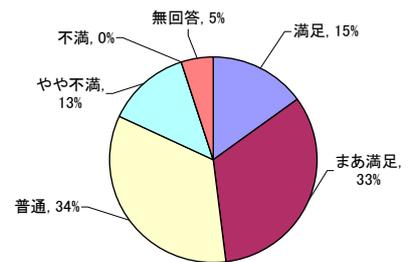
セッション 3
SystemC
ユーザフォーラム 2005



セッション 4
最先端 SoC の設計事例



セッション 5
組み込みシステム開発・検証
技術の最新動向



セッション 6
チップ・パッケージ・ボード
統合設計

各セッションの満足度調査

■WG 委員

主 査：	相 京 隆	富士通(株)
委 員：	藤 波 義 忠	NEC エレクトロニクス(株)
委 員：	今 井 正 治	大阪大学
委 員：	斎 藤 茂 美	ソニー(株)
委 員：	広 瀬 文 保	日本ケイデンス・デザイン・システムズ社
委 員：	蜂 屋 孝 太 郎	NEC エレクトロニクス(株)
委 員：	千 綿 幸 雄	富士通(株)
委 員：	長 谷 川 隆	富士通(株)
オブザーバ：	岡 村 芳 雄	(株)ルネサステクノロジ
オブザーバ：	樋 渡 有	(株)東芝
オブザーバ：	柏 木 治 久	NEC エレクトロニクス(株)
事 務 局：	菊 嶋 隆 史	JESA
事 務 局：	吉 永 文 章	JESA
事 務 局：	小 田 佳 代 子	JESA

3.3 ASP-DAC 2005

3.3.1 はじめに

ASP-DAC (Asia and South Pacific Design Automation Conference) は、VLSI およびシステム LSI の設計技術や設計自動化技術をテーマにしたアジア太平洋地域での最大規模の国際会議である。ASP-DAC は米国で開催されるこの分野のトップ・コンファレンスである DAC (Design Automation Conference)、ICCAD (International Conference on Computer Aided Design) や欧州で開催される DATE (Design, Automation and Test in Europe) とはシスター・コンファレンスの関係にあり、お互いにリエゾンを交換して協力関係を持っている。

ASP-DAC は、電子情報通信学会や情報処理学会などの学会だけでなく、電機メーカーおよび半導体メーカーの業界団体である JEITA (会議開始当時は EIAJ) と EDSFair (会議開始当時は EDAT) の支援のもとで 1995 年に開始された。業界団体である JEITA が ASP-DAC のような国際会議の支援を行っているのは、次のような理由による。電機メーカーや半導体メーカーが国際競争力のある電子製品の開発を行うためには、マーケティングや製品企画だけでなく、大規模・高機能・低消費電力のシステム LSI の最適設計を短期間で行える設計力を持つ必要がある。そのためには、最新の設計自動化技術についての情報収集と研究開発を行う必要がある。一流の国際会議を国内で開催することにより、わが国からより多くの技術者と研究者が参加して最先端の設計技術および設計自動化技術についての情報収集、情報交換などを行うことが可能になる。ASP-DAC はこのような目的を持って始められた。

3.3.2 会議の開催経緯と今後の予定

ASP-DAC の第 1 回目の会議は 1995 年 8 月 30 日から 9 月 1 日にかけて幕張メッセの日本コンベンションセンターで、情報処理分野の国際学会である IFIP (International Federation on Information

Processing) の TC10 WG10.2 および WG10.5 に属する CHDL および VLSI という名称の 2 つの国際会議と並列開催の形で開催された。第 2 回目は 1997 年 1 月に開催され、それ以降毎年 1 月に開催されてきた。ASP-DAC 2005 は 10 回目の開催である。この間、1999 年には香港 (中国) で、2002 年にはバンガロール (インド) でそれぞれ開催された。1999 年以降は、日本で 2 年間開催したあと国外で 1 回開催するというローテーションで運営されている。

今回の ASP-DAC 2005 は、上海 (中国) で 2005 年 1 月 18 日から 21 日までの日程で開催された。2006 年および 2007 年は、横浜 (パシフィコ横浜) で開催される予定である。

3.3.3 ASP-DAC 2005 の概要

ASP-DAC 2005 の概要を表 1 に示す。一般講演としては、32 カ国から投稿された 692 編の論文の中から 280 編が採択され、3 日間にわたって並列の 6 つのトラック (2 日目からは 4 トラック)、35 のセッションで発表された。表 1 からわかるように、2004 年と比較した今回の会議の最大の特記事項は、論文の投稿数が 291 編から 692 編と大幅に増えたことである。

また、大学 LSI 設計のコンテストでは、28 件のデザインが採択され、短時間の発表の後、ポスター展示が行われた。今回初の試みとして、博士課程の学生が自分の研究成果を発表する、Ph.D. フォーラムが開催された (他の国際会議では開催実績あり)。

基調講演のタイトルと後援者を表 2 に、特別セッションのタイトルとオーガナイザを表 3 に、1 月 27 日に開催された有料チュートリアルの概要を表 4 に示す。発表された論文の中から、表 5 に示す 2 本の論文が選ばれ、ベストペーパー賞が授与された。また、デザイン・コンテストに応募した作品の中から、表 6 に示すように、優秀デザイン 2 件が選ばれて表彰された。

また、Sallelite Workshop として、The 3rd Asian University Workshop on Semiconductor Design (北九州市主催、今回初) と Special workshop on compact modeling (ASP-DAC 2004 に続き 2 回目) の二つのワークショップが実施された。

表1 ASP-DAC 2004 と 2005 の比較

会議	ASP-DAC 2004	ASP-DAC 2005	
日時	2004年1月27日(火) ～30日(金)	2005年1月18日(火) ～21日(金)	
会場	横浜市(日本) パシフィコ横浜国際会議場	上海市(中国) Hotel Equatorial	
併設展示会	EDSF 2004	小規模な展示会(12社)	
論文投稿数	291	692	
論文投稿国(地域)数	28	32	
論文採択数(採択率)	147(50.5%)	Regular	99(14.3%)
		Short	86(12.4%)
		Poster	95(13.7%)
		合計	280(40.5%)
キーノートアドレス	2件	3件	
一般講演	35セッション(147編)	35セッション(185編)	
特別セッション (招待講演等)	8セッション	内部チュートリアル	3セッション
		パネル討論	3セッション
		招待講演	1セッション
デザイン・コンテスト	1セッション	1セッション	
ポスターボード	1セッション	4セッション	
Ph.D. フォーラム	—	1セッション	
有料チュートリアル	5件(全日3件、半日2件)	6件(全日2件、半日4件)	

表2 基調講演

講演タイトル	講演者
The development of Integrated Circuit industry in China	Zhenghua Jiang (中国人民会議副議長)
Silicon Compilation: The answer to reducing IC development costs	Rajeev Madhavan (Magma Design Automation)
Design at the end of the Silicon Roadmap	Jan M. Rabaey (UC Berkeley)

表3 特別セッションのタイトルとオーガナイザ

種類	セッション・タイトル	オーガナイザ
パネル討論	Who is responsible for the design for manufacturability?	C. K. Cheng, Stive Lin
	Are we ready for system level synthesis?	Jason Cong, Tony Ma
	EDA market in China	David Chen
エンベディッド チュートリアル	Design for Manufacturability	Lei He, X Sharon Hu
	Leakage Power: Trends, Analysis and Avoidance	Lei He, X Sharon Hu
	Designing Reliable Circuit in the Presence of Soft Errors	Lei He, X Sharon Hu
招待講演	CAD for Microarchitecture Designs (講演 3 件)	Hannah Honghua Yang

表4 Tutorial のタイトル

トラック	種類	タイトル
1	全日	T-1 C-Based Design
2	全日	T-2 power aware design for performance
3	半日	T-3 Automated Macromodeling Techniques for design of complex analog & mixed-signal Integrated Systems
4	半日	T-4 IP:Protection in Semiconductor and VLSI design
5	半日	T-5 Current Practices and Future Directions in high level design verification
6	半日	T-6 Chip-Package co-design

表5 ベストペーパー賞が授与された論文

論文タイトル	著者
“Speed and Voltage Selection for GALS Systems Based on Voltage/Frequency Islands“	Koushik Niyogi, Diana Marculescu (Carnegie Mellon University, USA)
“ The Polygonal Contraction Heuristic for Rectilinear Steiner Tree Construction”	Yin Wang, Xianlong Hong, Tong Jing, Yang Yang (Tsinghua University, China), Xiaodong Hu, Guiying Yan (Chinese Academy of Sciences, China)
“Fast PLL Simulation Using Nonlinear VCO Macromodels for Accurate Prediction of Jitter and Cycle-Slipping due to Loop Non-idealities and Supply Noise”	Xiaolue Lai, Yayun Wan, Jaijeet Roychowdhury (University of Minnesota, USA)

表 6 ベストデザイン賞が授与された設計

タイトル	タイトル
“A Bandwidth Efficient Subsampling-based Block Matching Architecture for Motion Estimation”	Hao-Yun Chin, Chao-Chung Cheng, Yu-Kun Lin, Tian-Sheuan Chang (Chiao Tung University)
“A Low-Power Video Segmentation LSI with Boundary-Active-Only Architecture”	Takashi Morimoto, Osamu Kiriyama, Hidekazu Adachi, Zhaomin Zhu, Tetsushi Koide, Hans Jürgen Mattausch (Hiroshima University)

3.3.4 論文の投稿状況

1997年から2005年の、ASP-DACへの論文投稿数の地域別の推移を図1に示す。ASP-DAC 2002はインドのバンガロールで同時開催されたVLSI Design 2002への投稿論文を含んでいるため、その前後の年よりも投稿数が多い。図1に示すように、1999年（香港開催）以降は投稿論文数が毎年増加傾向にある。海外からの投稿も多くなって来ており、名実ともに設計自動化の分野の国際会議として定着したと言ってよいであろう。表7に、日本からの論文投稿数の推移と全世界から投稿された論文に占める割合を示す。

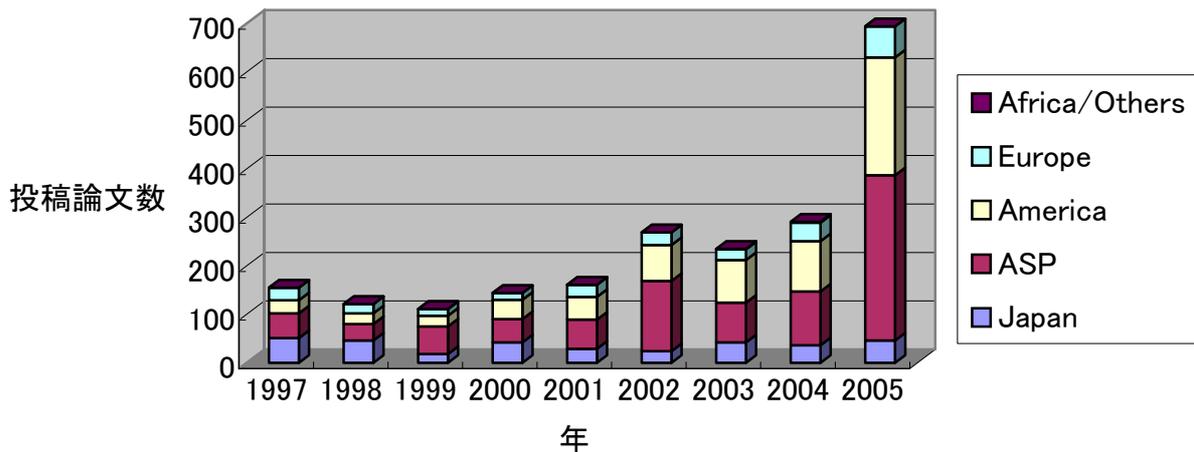


図1 地域別論文投稿数

表7 日本からの論文投稿数と全体に占める割合

年 地域	1997	1998	1999	2000	2001	2002	2003	2004	2005
日本 (割合)	N/A	N/A	N/A	42 (29%)	29 (18%)	24 (9%)	42 (18%)	36 (12%)	46 (6.6%)
全体	N/A	N/A	N/A	144	161	269	235	291	692

次に、研究分野別の論文投稿数および採択論文数を表8に示す。ASP-DAC 2005では、ASP-DAC 2004と同様に、研究分野を11種類に分類して論文の査読と採否の決定を行った。今回論文投稿

数が多かった分野は、分野 6 のパフォーマンス・ドリブン物理設計、分野 11 の最先端設計事例、分野 5 の回路とチップの最適化と設計検証などである。

表 8 分野別の論文投稿数と採択論文数

分野	研究分野	投稿数		2005 年採択数		
		'04	'05	Reg.	Short	Poster
1	System Level Design Methodology	42	65	10	8	8
2	Embedded and Real-Time Systems	22	37	6	4	6
3	Behavioral/Logic Synthesis and Optimization	29	66	9	6	11
4	Validation and Verification for Behavioral/Logic Design	31	50	7	6	7
5	Optimization and Verification in Circuit and Chip	34	74	9	10	10
6	Performance Driven Physical Design	51	103	21	6	17
7	Test Technology and Design for Testability	19	46	8	4	8
8	Analog and RF Circuit Design	29	64	8	8	9
9	Design for Manufacturability (TCAD)	11	23	3	6	0
10	Reconfigurable Systems	8	66	8	8	8
11	Leading-Edge Design Experiments	15	98	10	20	11
	合 計	291	692	99	86	95

3.3.5 登録者の内訳

ASP-DAC への地域別の参加者数の推移を図 2 に示す。また、日本からの参加者の推移を表 9 に示す。中国本土からの参加者は 168 名、台湾からの参加者は 44 名、米国からの参加者は 138 名である。

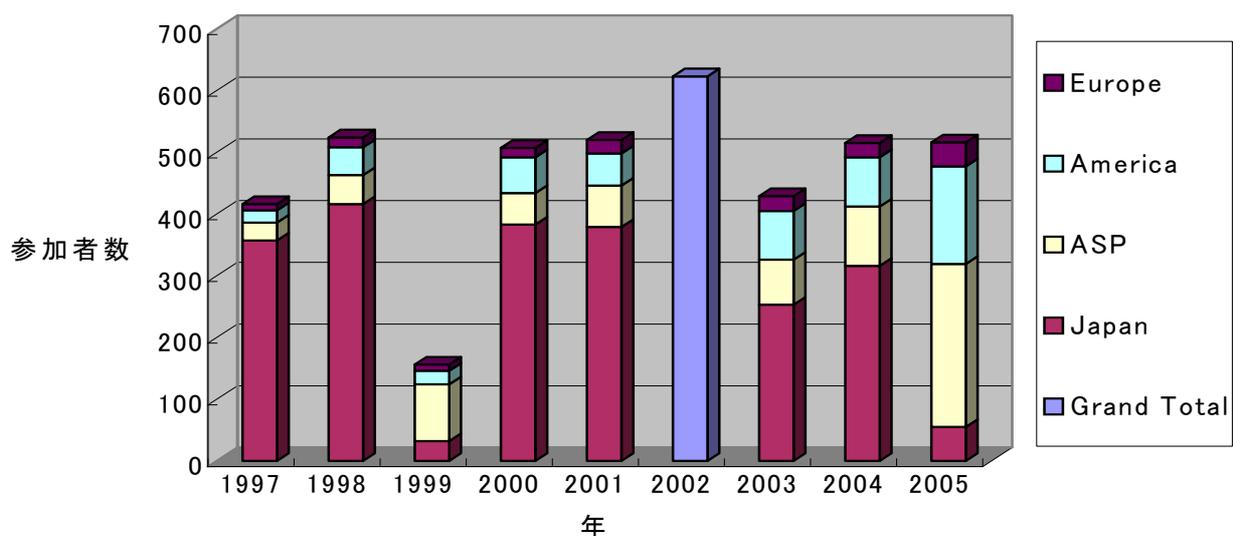


図 2 地域別参加者の推移

表 9 日本からの参加者数と全体に占める割合

年 地域	1997	1998	1999	2000	2001	2002	2003	2004	2005
日本 (割合)	357 (86%)	416 (79%)	32 (21%)	383 (76%)	379 (73%)	N/A	253 (59%)	316 (61%)	55 (11%)
全体	416	524	156	507	520	623	429	515	516

3.3.6 今後の展望

今回、論文数が大幅に増えた。特に 2004 年と比較して論文数が増えた国としては、中国本土 (37 編 ⇒ 209 編)、米国 (97 編 ⇒ 208 編) である。米国からの投稿のかなりの部分は、米国に滞在している中国系の研究者の貢献であると推測される。日本からの論文投稿は、38 編から 46 編に増えたが、全投稿論文数に占める割合としては、12%から 6.6%に低下した。

今回の上海での ASP-DAC の開催は、日本以外での開催としては大成功と言って良いであろう。論文の勧誘能力の点では、日本以上の実力を示したと言ってよい。また、参加者の国内からの動員力の点では日本に劣るものの、海外からの参加者の動員力の点では、日本以上に実力があると言ってよい。今回の成功を踏まえ、5 年後の 2010 年からは、ASP-DAC は日本と中国とで交互に開催される可能性が高い。

4. 添付資料

4.1 PDM 研究会報告

4.1.1 オンチップ・サーマル・タスクグループ (OCTE-TG)

4.1.1.1 活動目的

SoC チップでの消費電力の増加に伴って、発熱の問題が今後懸念される。しかしながら必ずしもオンチップでの発熱による影響は明確になっていない。チップの平均温度の上昇は、配線の EM 制約と密接に関連する。またチップ上での温度のばらつきは、ゲート遅延特性、配線遅延特性と密接に関連し、パスタタイミング、クロックスキューのばらつきへつながらず。

本タスクグループでは、チップ上での熱分布状況を解析するために、ITRS [OCTE-1] を参考に想定した 90nm プロセスで、パッケージを含んだシミュレーションモデルを作成し、温度ばらつきの要因となるパラメータ（メモリとロジックの面積比率、消費電力密度、フロアプラン）と、チップ内の最大温度差の関連性を解析する。これらの解析から、設計段階に於いて、チップの消費電力のプロファイルに従って、温度ばらつきの特性影響を如何に削減するか、フロアプランに対する指針を具体的に提案する。

また、このチップ上の温度差が、チップのパフォーマンス（CLOCK SKEW、EM 特性）へ如何に影響するか定量的な分析を行う。

さらに、温度に対して指数的な増加傾向を持つリーク電流の挙動をシミュレーションモデルへ組み込むことで、設計でどのように考慮すべきかを解析、提案する。

4.1.1.2 現在の研究動向

LSI が消費する電力の増大につれて、LSI の温度は設計の強い制約となってきている。特に、高性能なマイクロプロセッサにおいては、熱のマネジメントが重要な設計項目の一つとなっている。チップの温度を監視するためのセンサを搭載することも珍しくない [OCTE-33、OCTE-34]。

熱による動作不具合を防止するための対策として、例えば次のような方法が提案されている [OCTE-39、OCTE-21、OCTE-22]。

- (1) アーキテクチャ多重化による対策：同じ処理を行う消費電力の異なるブロックを冗長化して用意する。通常は高消費電力ブロックを用いるが、チップ上の温度が過度に上昇した場合には低消費電力ブロックに動作を切り替える（性能は低下する）。
- (2) 電圧と動作周波数の制御：チップ上の温度を監視して温度の上昇に応じて電源電圧や動作周波数を調整する。

通常これらの対策はゲート数の増加につながり、コスト大となる。また、SOC においてはパッケージに割り当てるコストが制限されることが多いため、SOC 設計において熱が問題となるかどうかを見極めることは非常に重要である。しかし、SOC を対象として、温度が問題となるか、また温度の影響がどの程度であるかを定量的に求めた文献はほとんど見当たらない。

以降で検討する内容に関連する温度解析の研究としては、次のようなものが挙げられる。文献 [OCTE-23] では、ごく限られた配線とビアの領域に対して温度分布を正確に求めている。シリコン基盤材料の均一性を仮定してグリーン関数法によりフルチップの温度分布を高速に求める方

法が、文献[OCTE-27]に記載されている。温度分布の高速計算に関する研究としては、文献 [OCTE-28、OCTE-17、OCTE-37] 等を挙げることが出来る。チップ上の温度分布を考慮し、温度分布の平坦化を図るレイアウト設計手法としては、[OCTE-29、OCTE-30] がある。これらは force-directed placement 手法 [OCTE-31] における場の一つとして温度を扱うものである。

4.1.1.3 昨年度の成果

昨年度の結果をまず、以下に整理する。昨年度は、チップのプロファイルを仮定して、シミュレーションを行うための環境作成を行った後、

- 1) チップの構成ブロックによる依存性（特に消費電力の異なる、メモリ、ロジック、IO で分類し、その割合がどのように、最大温度・最大温度差に影響するか。
- 2) 上記の高消費電力のロジックブロックの位置がどう影響するかを解析した。下図にその結果をまとめる。

シミュレーション結果の整理

- 消費電力一定の元で、メモリ領域の割合の増加は温度差 ΔT_{\max} を増加させる
 - 小領域に多くのロジック部の消費電力が集中する
- ロジックがチップの中央よりも角に集中する方が、温度差 ΔT_{\max} が増加する
 - チップ周辺はほとんど断熱である
- ロジックを分割し、温度勾配方向へ移動することで温度差 ΔT_{\max} を緩和できる
 - ブロック分割は、消費電力分布を平坦化する

図 OCTE-2 シミュレーション結果の整理

熱の電気回路との相似性による シミュレーションモデル

- SPICEを使ったシミュレーション
- Active layer cellsに熱源を電流源として接続

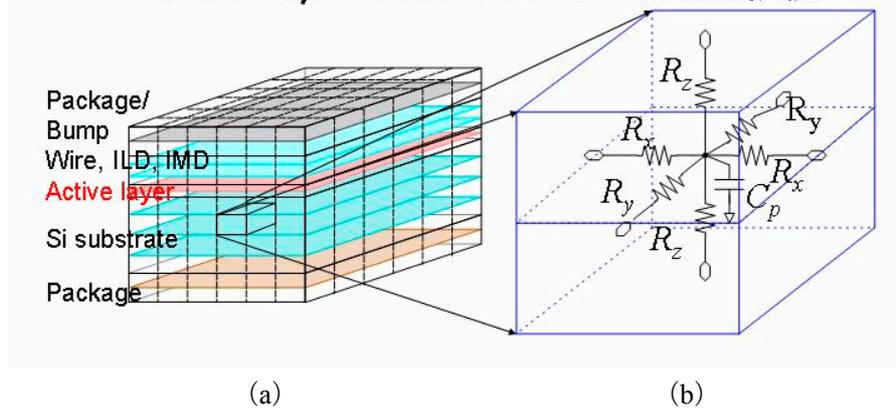


図 OCTE-3 シミュレーションモデル

4.1.1.3.1 解析モデル

本解析ではダイおよびパッケージを直方体の小領域に分割し、これらを相互に熱抵抗で接続することにより熱の拡散をモデル化する。各小領域は熱抵抗と熱容量からなる3次元格子で表わす。小領域への分割の様子を図 OCTE-3 に示す。この図では、BGA パッケージに封止されているダイを x、y 方向にそれぞれ6分割、z 方向に8分割している。ダイはシリコン基板層、デバイス層、および金属配線層からなり、金属配線層の上にはパッケージ材料を介して半田ボールが接続する。また、シリコン基板層の下側を樹脂等からなるパッケージ層により覆う。境界条件として、最上層のボール層と最下層のパッケージ層の両方から周囲温度との熱交換を行うと仮定し、チップおよびパッケージの側面は断熱とする。LSI の大きさ (x、y 方向) を 10 mm 〇、厚さを約 500 μm (ダイ) + 400 μm (パッケージ) とする。配線層数や寸法は、ITRS の 90 nm ノードの数値を参考とする 11) 。分割数は x、y 方向にそれぞれ 16 分割、z 方向に 9 分割とし、パッケージ表面 (図 OCTE-3 (a) の上面と下面) を、25°C に固定する。配線材料として Cu、配線間および配線層間の絶縁材料 (IMD、ILD) として SiO₂ を仮定する。厳密には Cu、SiO₂ の熱抵抗は温度依存性を持つが、ここでは温度依存性を無視し常温の値で近似する。例として用いる LSI の層構成と、層毎の熱特性を表 1 に示す。epoxy はパッケージの材料であるエポキシ樹脂を、sub1~sub4 と sub s の 5 層はシリコン基板を表す。シリコン基板は他の層と比較して厚みが大きいため、sub1~sub4 の 4 層に分割している。基板の表面近傍にはデバイスが作成されているため発熱源となっており、また熱特性が他の層とは異なることから、基板表面を独立した層 sub s としている。配線層と ILD は複数まとめて wire1 と wire2 の 2 層に分割する。それぞれの層は、x、y 方向の配線 2 層と ILD 4 層の計 8 層について別個に等価抵抗・容量を計算し、直列、並列の接続を考慮してまとめている。wire2 の上には、パッケージ樹脂とボールから

なる bump を定義する。

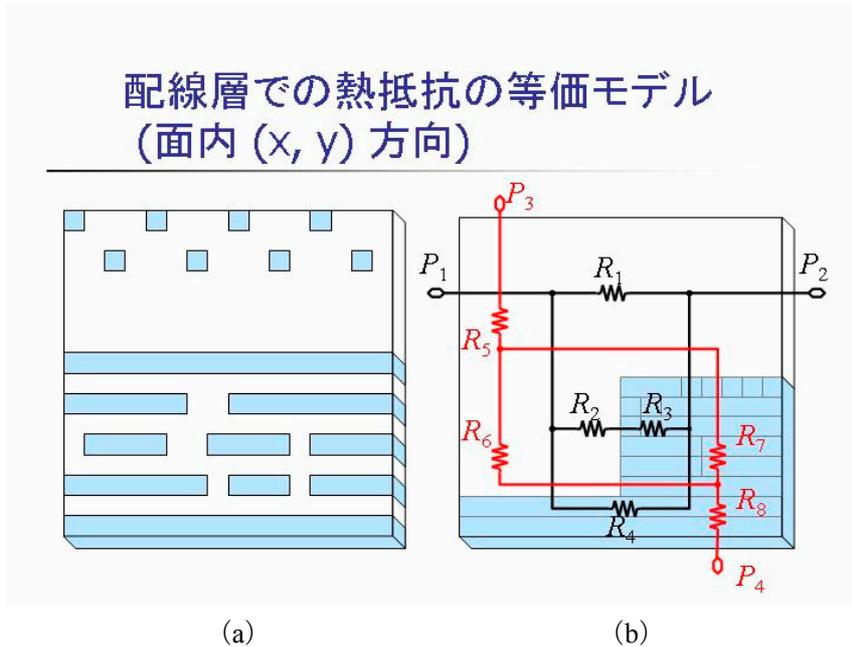


図 OCTE-4 配線層での熱抵抗等価モデル

熱抵抗は対象となる配線層の金属配線をまとめた等価抵抗として計算する。例えば図 OCTE-4 (a) のような配線に対しては、(b) に示すように抵抗値を変えずに並べ変えを行って P1-P2 間の合成抵抗を計算する。この際には、金属配線の熱抵抗が小さい事から、配線が領域を貫通する割合を考慮している (R4)。同一領域の y 方向、および z 方向に対しても同様の計算を行い等価抵抗を決める。熱容量は金属配線と絶縁膜の熱容量の総和から求める。

熱抵抗・容量のパラメータ例

- パッケージ特性(熱抵抗)がチップ温度を決定
- チップの温度時定数はmsのオーダー

Layer	Z-thickness (μm)	R _x (K/W)	R _y (K/W)	R _z (K/W)	C _p (J/K)
Epoxy	200	2.7e4	2.7e4	2.7e3	9.81e-5
Sub1-4	125 each	63.5	63.5	2.54	8.00e-5
Active	2	9.0e3	9.0e3	0.037	1.56e-6
Wire1-2	3.1 each	7.8e3	7.8e3	0.93	5.39e-6
Bump	200	5.0e4	5.0e4	1.0e3	1.69e-5

図 OCTE-5 熱抵抗・容量のパラメータ例

4層（sub1～sub4）に分割するシリコン基板層の熱抵抗と熱容量は、基板が全て純粋な金属シリコンからなるとみなして熱抵抗を求めている。パッケージ材料は通常、ダイを構成する材料と比較して熱抵抗が大きいので、チップの温度を決める主要な因子となる。本例で基準とするパッケージは、等価的なz方向の熱抵抗を約3（K/W）とする。この数値は、ハイエンドMPU等に用いられる熱抵抗の小さいパッケージを、強い送風等による良好な冷却環境下で使用することに相当する。

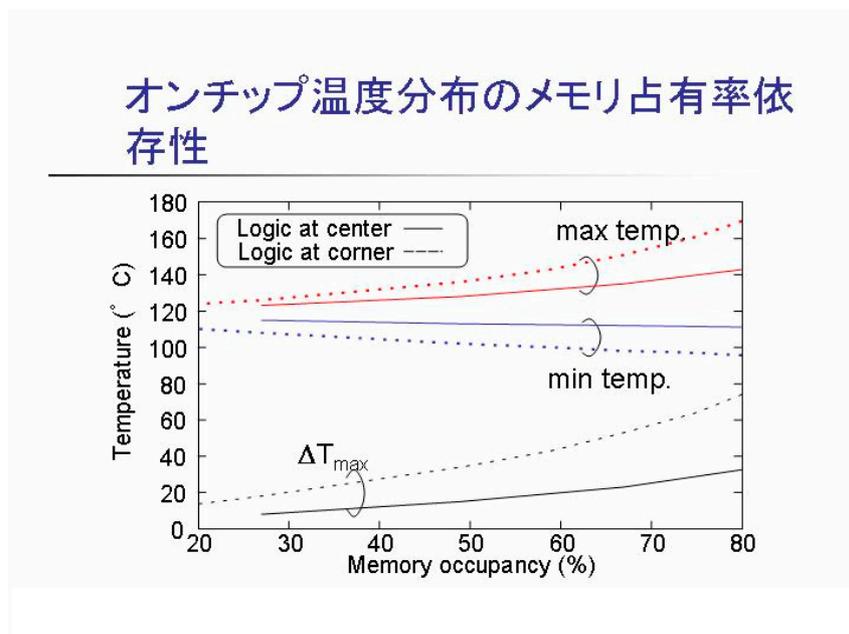


図 OCTE-6 温度分布のメモリ占有率依存性

4.1.1.3.2 回路機能依存性

最近のLSIではメモリを多く搭載する場合がある。メモリとロジックでは消費電力が異なるため、メモリの多寡により温度分布が異なると考えられる。そこでI/O領域を除くチップ上にメモリが占める割合をメモリ占有率 αM として定義し、温度分布に与える影響を確認する。ITRSでは、SoCのメモリ占有率が2003年に約75%、2012年には約93%に達すると予測している(11)。全体的な傾向としてメモリ占有率は増加傾向にあるが、実際にはチップの応用によりメモリ占有率は大きく幅を持つと考えられる。そこでここではメモリ占有率も温度勾配に影響を与えるパラメータと考えて、20～80%に変化させる。

メモリの電力密度の絶対値 ρM を仮定し、占有率によらず ρM 一定とする。ここではメモリの電力を先に決めて、残りを全てロジックの電力として割り付ける。

ロジックを1箇所集中させるC型またはL型の配置、 $\rho M=0.25$ （W/mm²）とし、メモリ占有率 αM ほぼ20～80%の範囲で変化させる時のチップ内温度の最大値と最小値、最大温度差（=最大値 - 最小値）を図OCTE-6に示す。メモリの占有率が大きくロジックの占有率が小さくなるほどチップ内の最大温度、および温度差が増加する。すなわちチップの総消費電力が一定であってもチップの温度分布は異なる場合がある。特にメモリの電力を一定とするため、メモリ占

有率に対してメモリ電力が線形に増加する。このためロジックにおける電力密度はメモリの占有率に対し著しく増大する。その結果、ロジックで観測される最大温度が増加する一方、メモリで観測される最小温度が減少して温度差は急激に大きくなり、温度差は約 75 °C となった。

4.1.1.3.3 フロアプラン依存性

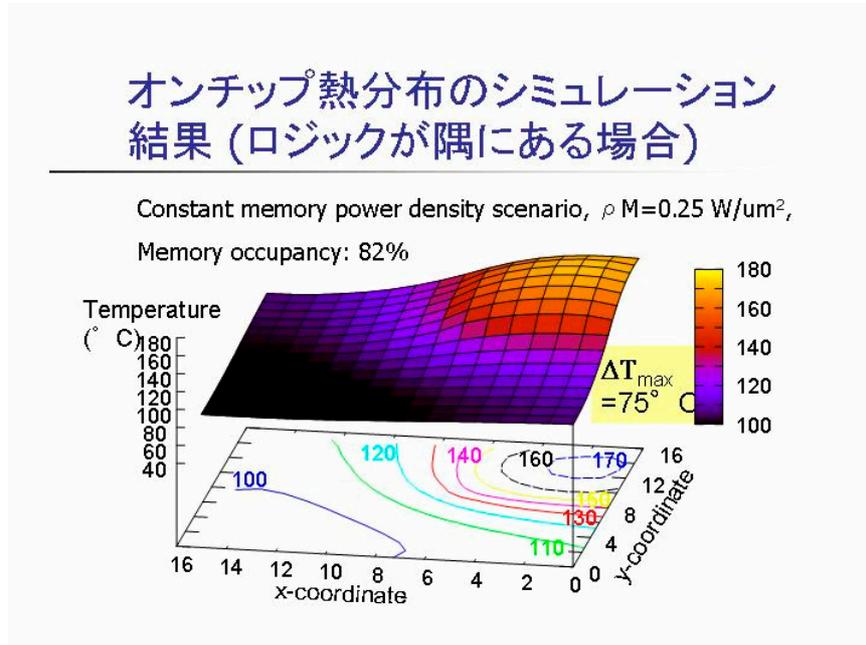


図 OCTE-7 デバイス層の温度分布 (メモリ占有率 80%) L 型

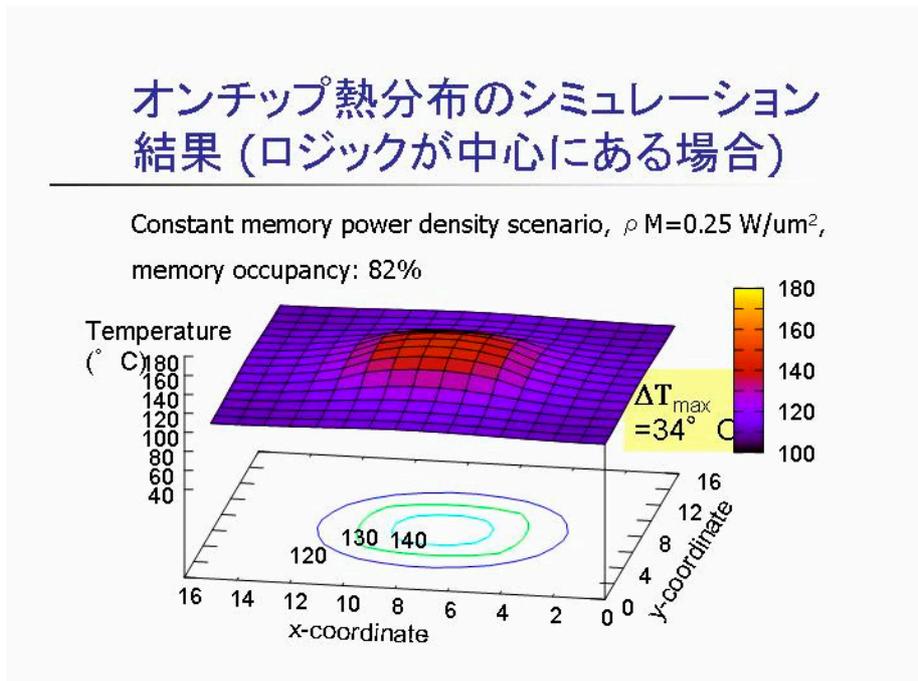


図 OCTE-8 デバイス層の温度分布 (メモリ占有率 80%) C 型

C 型、L 型の配置を出発点として、ロジックがより小さいブロックに分割可能な時に温度勾配を小さくするためのブロック配置について検討する

図 OCTE-6 では、メモリ占有率や電力の設定によらず常に L 型配置の温度差が C 型配置に対して大きい。またチップ内での最大温度も、L 型配置が高い。L 型配置では消費電力の大きいロジックがチップの隅に置かれる。チップの側面は断面積が小さく横方向の熱抵抗を放熱経路として有効に使えないため放熱に対する寄与が小さい。このため、チップの隅に熱源となるブロックを配置すると、温度が上昇すると考えられる。メモリ占有率 80% 時におけるチップ内の温度分布を図 OCTE-8 に示す。

次に、メモリ占有率を約 80 % に固定してその配置や分割を初期フロアプランから変更していく時に、温度分布がどのように変化するかを図 OCTE-9 に示す 3 つのパターンについて分析する。いずれのパターンでも I/O 回路はチップの周辺部に固定する。

パターン (a-1) 初期配置としてロジックを 1 ブロックとしてチップの隅に置く。消費電力とブロック形状を保ったまま、ロジック位置を対角線方向へチップ中央まで 1 格子ずつ移動する。

パターン (a-2) (a-1) によりブロックが中央に移動した状態を初期配置とする。ロジックが 4 つの小ブロックに分割可能であるとして、消費電力と占有面積の合計を同一に保ったまま、分割したブロックを隅に向かって 1 格子ずつ放射状に分散させる。

パターン (b) 初期配置ではロジックを 1 ブロックとしてチップの隅に置く。このブロックがさらに細かい 4 ブロックに分割可能であるとして、消費電力と占有面積の合計を同一に保ったまま、分割したブロックを x、y 軸および対角線方向に 1 格子ずつ同時に分散させる。但し、隅の 1 ブロックだけは制約により移動できないとして、左上隅に固定する。

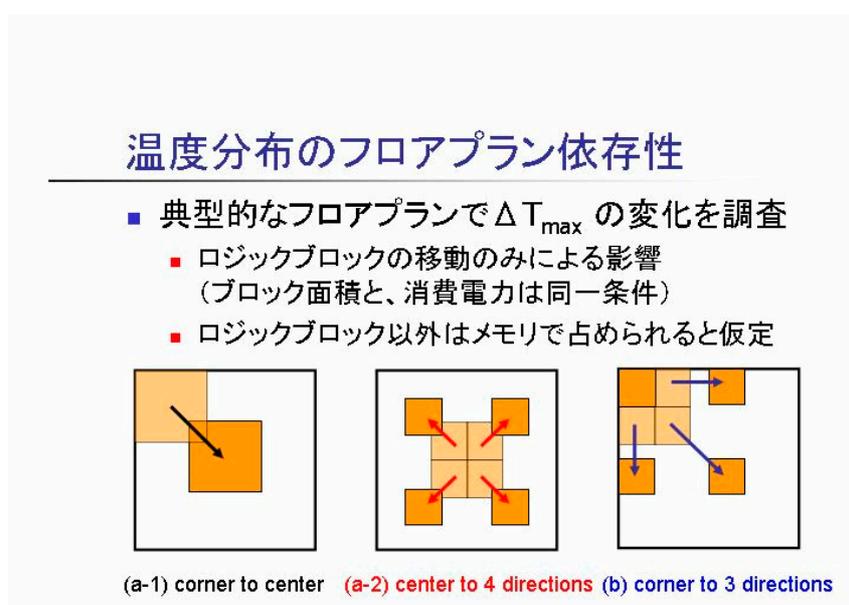


図 OCTE-9 温度分布のフロアプラン依存性

それぞれのパターンについて温度差の変化を図 OCTE-10 に示す。横軸はブロックの移動距離を示しており、x、y 軸方向、又は対角方向に関わらず 1 格子分の移動を 1 目盛としている。(a-1) の終点と (a-2) の始点、および (a-2) の終点と (b) の終点は、移動の結果が同じ配置となるためそれぞれ一致する。グラフより同じロジックを 1 箇所に集める構成をとる場合でも、その配置場所により温度差が異なることがわかる。(a-1) の操作によりチップの隅にロジックを配置する場合には温度差が約 75 °C 生じるが、中央に移動させると約 34 °C となる。さらに電力の大きいロジックを分割し配置を分散すると、チップ内の温度差がより緩和される。4 分割して移動させると温度差は 11 °C まで改善できる。但しこの移動に対しては温度差を最小とする最適配置が存在し、移動量を大きくしてブロックがチップの隅に近付くと温度差は再び大きくなる。温度差が最も改善されるフロアプランでの温度分布を図 OCTE-11 に示す。

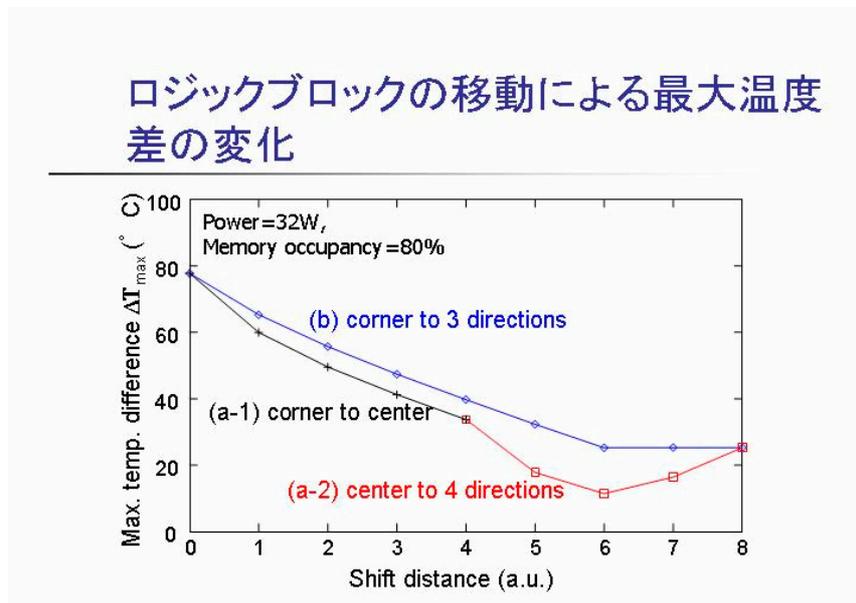


図 OCTE-10 温度分布のフロアプラン依存性

フロアプラン改善後の温度勾配

Constant memory power density scenario, $\rho M=0.25 \text{ W}/\mu\text{m}^2$,
memory occupancy: 82%

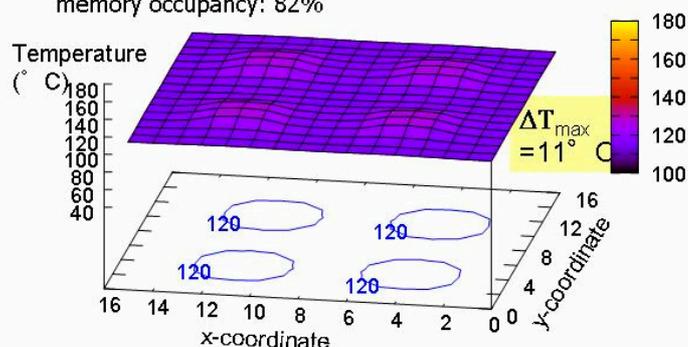


図 OCTE-11 ロジック分割による温度分布の改善結果

4.1.1.4 特性分析

4.1.1.4.1 パッケージ性能との関連

パッケージ性能と熱の関係を定量的に把握するため、各パッケージに対する最大温度差を調査する。パッケージコストとその熱抵抗は一般に反比例の関係にある。パッケージの選択は、LSIの製造コストを構成する重要な要素のひとつとなっており、電力許容量を満たす範囲で、最も安価なパッケージを使用する事が好ましい。

本解析では、 $R\theta=3\text{kW}\sim 27\text{kW}$ の5種類の熱抵抗からなるパッケージを想定する。図 OCTE-12には、想定した5種類のパッケージにおける消費電力と最大温度差の関係を示す。フロアプランはロジックを角に配置する、最大温度差が大きくなる場合を想定している。パッケージの熱抵抗は、そのLSI内部の熱抵抗よりも大きく、パッケージの熱抵抗を設定内で小さくしても最大温度差はほぼ比例して増加している。また、図中の点線は、各消費電力における最適なパッケージを選択したものであり、最大許容温度は $T_{\text{allow}}=120^\circ\text{C}$ としている。

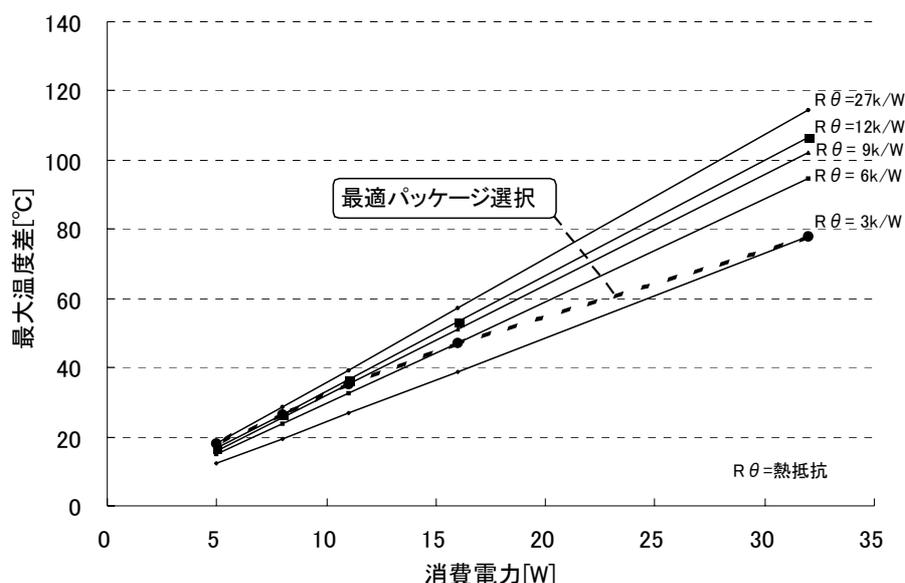


図 OCTE-12 パッケージ性能とチップ消費電力の関係

実際にはこの点線以下のパッケージしか選択出来ない。最大温度差は、各電力でのパッケージの選択差によっても 10°C 以下の改善しかみられない。またチップの消費電力が小さい場合に、極端に温度差が大きくなることはないことがわかる。

4.1.1.4.2 CLOCK SKEW に対する影響分析

本節では、前章でのチップ平面方向の温度分布解析結果を反映したクロックディレイの解析を行う。クロックツリー形状はバランスのとれた対称形状を想定し、SPICE シミュレーションにより各観測点のディレイを計測している。

解析にあたり、クロックツリーモデルを図 OCTE-13 のように仮定した。チップ全体にクロックが供給されており、クロックソース（チップ中央）に対し対象な構造をもつ。配線長はクロック周波数がある程度高速な想定のため最大 $1.25 [\text{mm}]$ としており、それ相当の配線抵抗と配線容量を 4 段の π 型モデルで表している。本モデルでは、グローバルクロックのみで、クロックバッファは最大 6 段を想定している。本グローバルクロックバッファ以降の信号経路はローカルなクロックツリーによりディレイ調整がなされているとする。

観測ポイントは、全 36 の観測ポイントから論理的にスキュー値が 0 となる 9 組に分け、クロックバッファの出力を測定している。それぞれで測定の最も大きな delay 差を本クロックツリーのスキュー値とする。

上記モデルを使用し、各クロックバッファに前節で求めた温度を割り当てて SPICE シミュレーションを行って求めたクロックスキューを図 OCTE-14 に示す。横軸は、フロアプランプランの状態を示しており、縦軸はディレイの絶対値で正規化したスキューである。図中 (a-2) は、ロジックブロックの配置が中心から対象の配置となっており、これは本解析で用いたクロックツリーの構造と同じであるため、スキューは発生しない。(a) または (b) のケースでは、熱勾配が大きい

ほどクロックスキューは大きい。また、本仮定においては、最大で全体ディレイに対して約 10% 程度の差が生じる結果を得た。

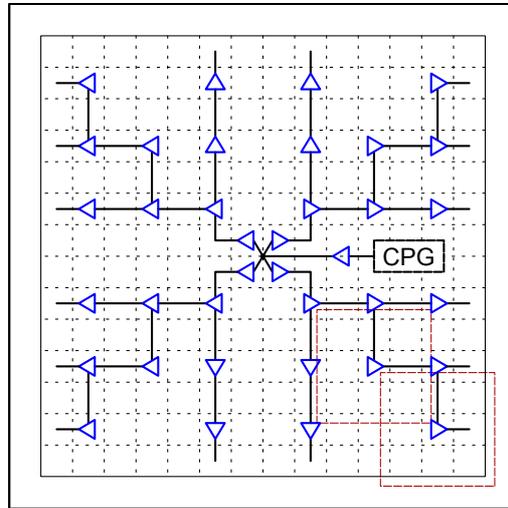


図 OCTE-13 クロックモデル

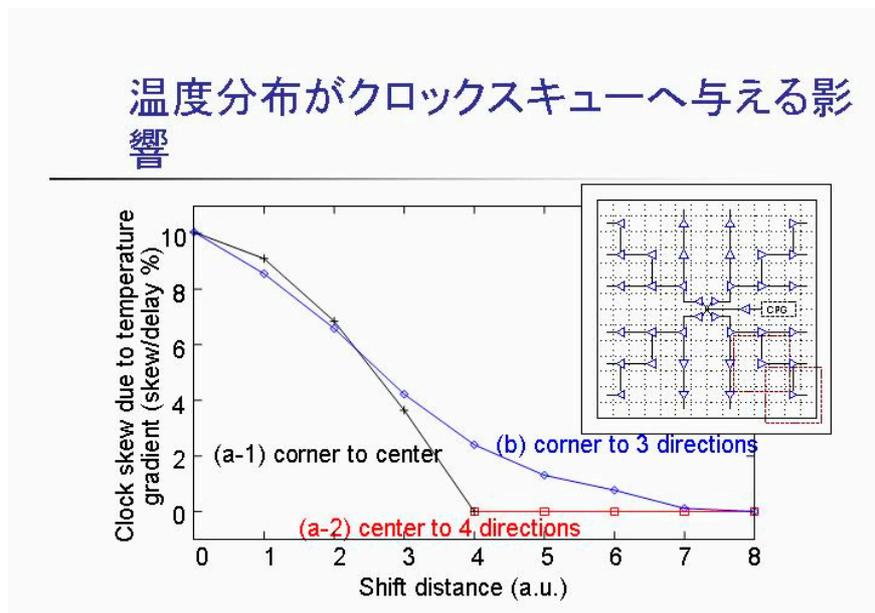


図 OCTE-14 クロックスキューへの温度の影響の解析結果

4.1.1.4.3 EM への影響分析

温度差が配線信頼性に与える影響を調べる。信頼性の指標は、ブラックの式[OCTE-25] $MTTF = A_{j-n} \exp(E/kT)$ で得られる。ここで A 及び E はプロセス及び形状に依存する定数で、K はボルツマン定数である。ここで、 $J_{max}(T_{spec})$ は、ある特定の温度 T_{spec} の温度における最大許容電流密度とし、(ここでは $T_{spec} = 120^{\circ}\text{C}$) $J_{max}(T)$ は、実際の温度 T における実効最大許容電流密度とする。この $J_{max}(T)$ は、上記のブラックの方程式で T における MTTF が温度 T_{spec} に等しくなるようにすることで得られる。

チップの温度が上昇するに従い、実効の J_{max} を減少させる必要がある。その割合を $\gamma_j = J_{max}(T) / J_{max}(T_{spec})$ としてあらわす。 $T > T_{spec}$ であるので、 $\gamma_j < 1$ となる。最大温度が T_{spec} に等しい場合は、 γ_j は 1 となる。

高性能 SoC では、局所的に仕様温度を超えるため、グラフに示すよう、 $\gamma_j < 1$ となる電流密度を守る必要が出てくる。図 OCTE-15 では、すべて同じ消費電力と、ロジック領域の面積を持っているため、フロアプランを改善することで電流密度の制限値を 0.3~0.8 倍緩和できることが分かる。

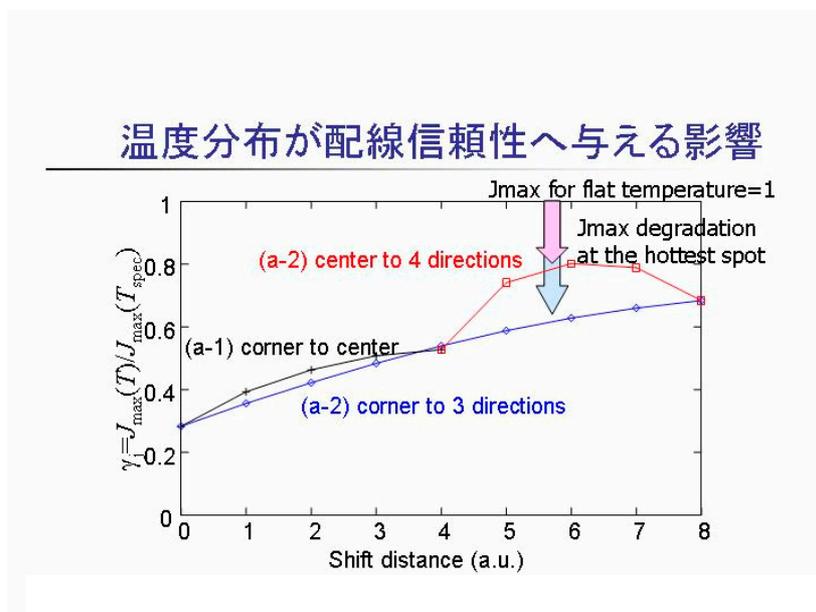


図 OCTE-15 配線信頼性への影響

4.1.1.5 温度差削減のためのフロアプラン配置改善手法

これまでの解析からチップ上の温度差がチップ性能に影響を及ぼすことがわかった。そこで本節では、チップ上の温度差を改善するための手法について検討する。

以下での温度は、全てシリコン基盤表面のデバイス層のセルで定義する。いま T_i をセル番号 i の温度とする。ここで i は集合 S_k の要素であり、集合 S_k はブロック k を構成するセル中のインデックスの集合である。

4.1.1.3 節の結果から、1) ブロックを小さく分割する 2) 小さく分割したブロックを分散して配置する、の 2 点が温度の平坦化に有効であることがわかっている。これらを考慮して、最大温度差 Δt_{max} を削減する一連の基本操作を以下のように提案する (図 OCTE-16)。

-
- step-1. タイミングを第1優先とする最適化を行い初期フロアプランを作る
- step-2. シミュレーションにより温度差を計算する。
- step-3. Δt_{max} があらかじめ決められた目標値以下であるか、または繰り返し回数が事前に設定された上限を超える場合には手続きを終了する。そうでなければ次の step-3 に進む。
- step-4. 全てのブロックについて温度勾配ベクタ vd を x-y 平面上に投影したベクタの総和を求め総和のベクタが指す方向にそのブロックを1ステップ動かして step2 へ戻る。

$$\text{但し } vd = \sum_i \text{grad}Ti$$

チップの熱抵抗はパッケージが支配的であるため、step-3 の勾配の計算では単に z 方向を無視して x-y 平面内への投影で近似できる。以後の解析では、ブロックの移動は隣接する 8 近傍の 1 グリッド刻みとする。

図 OCTE-17 と図 OCTE-18 に、温度の平坦化例を示す。図 OCTE-17 は初期フロアプランであり、チップ内には最大 26.1°C の温度差が有る。図中の矢印は温度勾配ベクトルを示す。このベクトルに従い提案する手続きを 23 回繰り返してブロックの移動を完了した最終的なフロアプランが図 OCTE-18 である。このとき最大温度差は 7.4°C まで低減されている。図には同時に、各ブロックがどの程度移動したかを示してある。ブロックのわずかな移動により効果的に温度差を削減できていることがわかる。このためタイミングに与える影響は軽微であると考えられる。

温度平坦化のための基本操作

1. 初期フロアプラン(温度は考慮しない)
2. 提案する温度シミュレーションによる温度勾配の算出
3. 最大温度差が、目標値内に収まれば終了
4. ブロックに対して温度勾配ベクトルの総和をもとめ、総和が最も大きなブロックをそのベクトルの方向へ移動し、ステップ2へ進む

図 OCTE-16 温度平坦化の基本操作

温度平坦化例 --- 初期フロアプラン

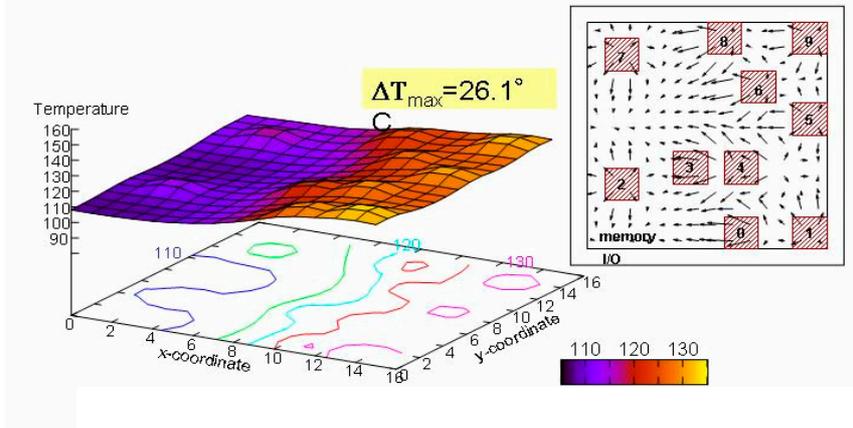


図 OCTE-17 温度平坦化手続きの初期状態

温度平坦化例 --- 最終フロアプラン

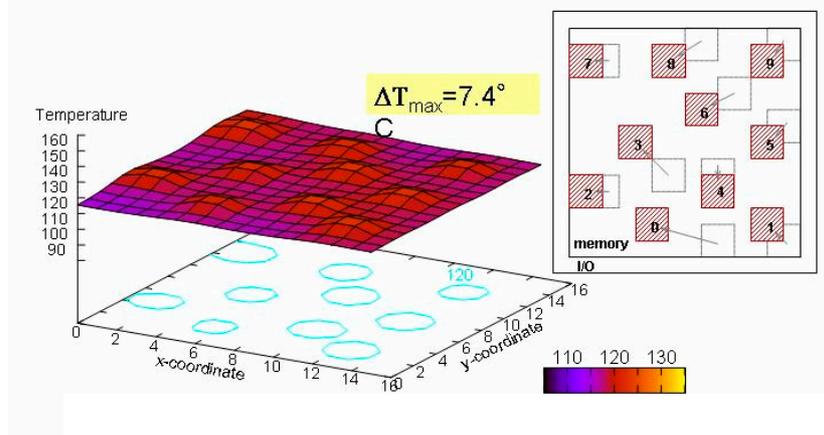


図 OCTE-18 温度平坦化手続きの最終状態

4.1.1.6 リークを考慮した温度解析

4.1.1.6.1 リークモデル

まず、0.1 [um] プロセスを想定した BSIM3 SPICE モデルを用いて、単体トランジスタのサブスレシヨールドリーク電流を求めるシミュレーションを行った。SPICE モデルは、 V_{th} の異なる 2 種類を想定している。その結果を図 OCTE-19 に示す。横軸を温度、縦軸をサブスレシヨールドリー

ク電流としている。よく知られているように、サブスレショールドリーク電流は、温度に対して指数関数的に増加する。

サブスレショールドリーク電流の温度特性 (単体トランジスタ)

BPTM 0.1um BSIM3モデル
Vth(2種類) : Nch/Pch = 0.2/-0.3 or 0.25/-0.2

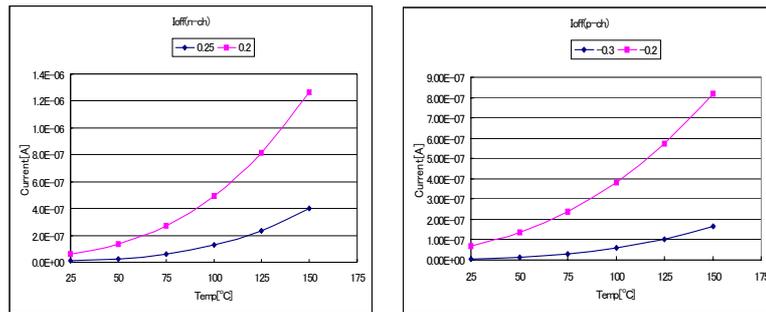


図 OCTE-19 サブスレショールドリークの温度特性 (単体トランジスタ)

次に、ゲートレベル (2 入力 NAND) でのサブスレショールドリーク電流をシミュレーションから求める。2 入力 NAND に対しては、図 OCTE-20 に示す 4 つの入力状態が考えられるため、それぞれの状態でサブスレショールドリーク電流を測定する。

上記 2 入力 NAND でのサブスレショールドリーク電流の温度特性を図 OCTE-21 に示す。論理ゲートでは、入力の State によってサブスレショールドリーク電流は大きく異なることがわかる。特に、縦積みのトランジスタの効果により、サブスレショールドリーク電流は大きく減少する。

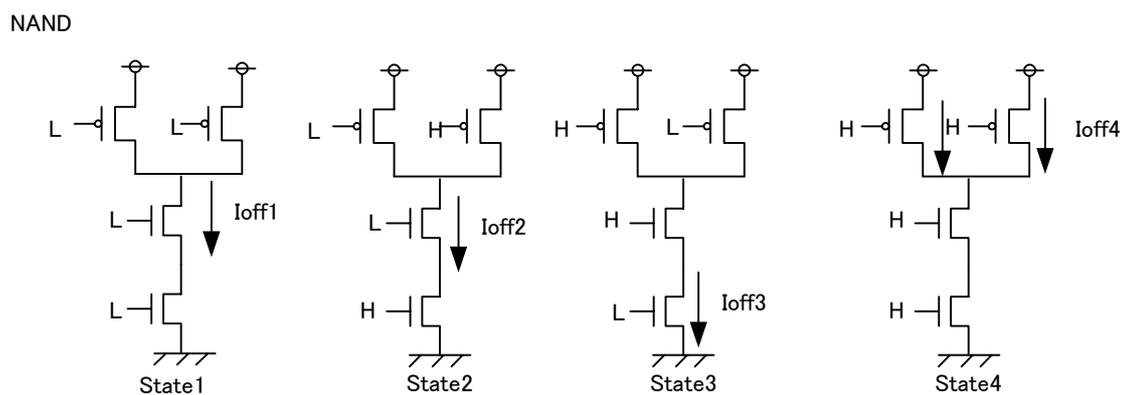


図 OCTE-20 2 入力 NAND のサブスレショールドリーク電流測定モデル

サブスレシヨルドリーク電流の温度特性 (NAND)

- 入力論理状態によりリーク電流は大きく変化
- 縦積みトランジスタによりリーク電流減少
- 但し温度特性は同じため単体トランジスタの式をセルにも適用可能

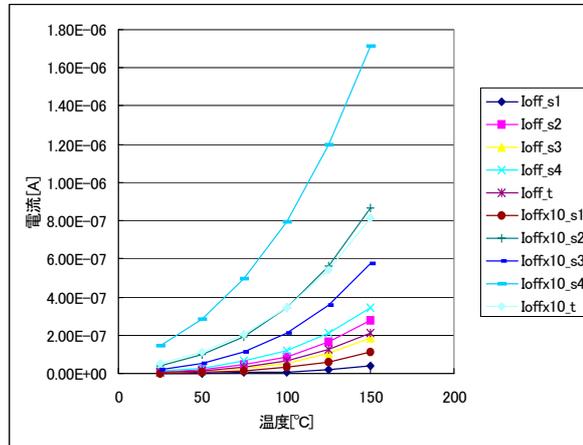


図 OCTE-21 サブスレシヨルドリークの温度特性 (NAND)

リークモデル

- BSIM3に基づくリーク電流モデル式

$$I_{sub} = \mu_0 C_{OX} \frac{W}{L} e^{b(V_{ds} - V_{dso})} v_t^2 \left(1 - \exp\left(\frac{-v_{dd}}{v_t}\right) \right) \exp\left(\frac{-|v_{th}| - v_{off}}{n \cdot v_t}\right)$$

- モデル式の簡略化

$$I_{sub}(T) = \alpha \cdot T^2 \exp\left(-\frac{\beta}{T}\right)$$

- リーク考慮手順

- 単体トランジスタにより β , ブロック電力より α を計算
- リーク電力を回路モデルとして記述し, 回路シミュレータで非線型方程式を解く

図 OCTE-22 単体トランジスタとセルリークの関係

図 OCTE-22 に、リークモデル式を示す。ここでは BSIM3 のサブスレシヨルドリークの式 I_{sub} を温度の関数と見て簡略化する。必要となる最小の係数は α と β の 2 つである。簡略化したモデル式 (簡易リークモデル式) は、一つのトランジスタに対するリークの式であるため、以下の検討が必要となる。

- (1) セルまたはブロックに対する簡易リークモデル式の適用性
- (2) 簡易リークモデル式の、シミュレーションへの組み込み

トランジスタモデルと簡易モデル式によるリーク電力の比較(PMOS単体)

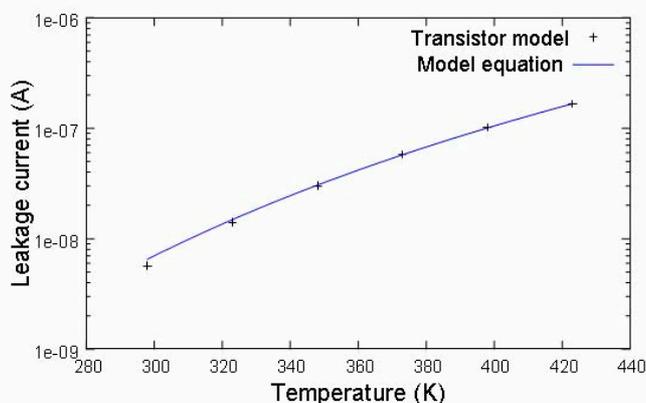


図 OCTE-23 トランジスタモデルと簡易モデルとの電力比較 (PMOS)

図OCTE-23にはPMOSトランジスタモデルのリークと簡易リークモデル式のリーク電流を比較した結果を示す。当然では有るが両者は良く一致する。図OCTE-24は、前節で求めた2入力NANDセルのリーク電流を簡易リーク式によりフィッティングした結果である。点はSPICEにより求めたセルのリーク電流、線は簡易リークモデル式である。2NANDは入力端子の状態によりリーク電流が異なるが、簡易リークモデル式はいずれの状態に対してもリーク電流を良く近似している。NOR等その他のセルについても、モデル式はリーク電流の温度特性を良く近似できることがわかった。

回路ブロックもまた、これら論理セルにより構成されるから、上記「(1)セルまたはブロックに対する簡易リークモデル式の適用性」に関しては、簡易リークモデル式がセルの電流を代表できると考える。本検討では想定するプロセスの単体トランジスタより β を決定し、ブロックの電力を実現するように α を決めることになる。

簡易リークモデル式の精度(NANDセル)

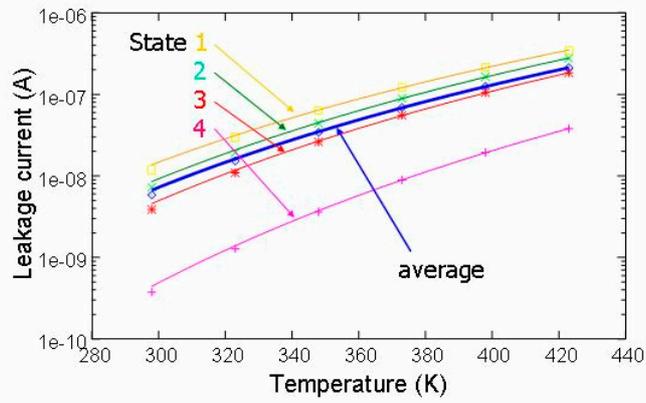


図 OCTE-24 簡易リークモデル式の精度 (NAND)

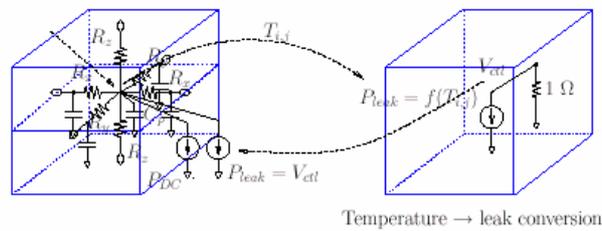
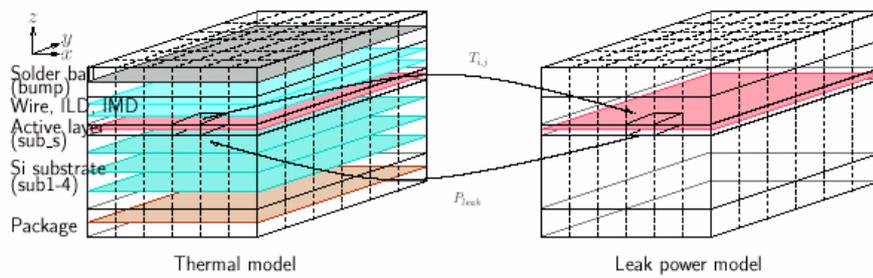


図 OCTE-25 リークを考慮した熱回路モデル

簡易リークモデル式は、図 OCTE-25 に示す構造にてシミュレーションに組み込む。すなわち、左側上段の従来のモデルにおいて、消費電力を表現する電流源を充放電成分とリーク成分に分離して別々の電流源として接続する。充放電成分は直流電流源、リーク成分はその場所の温度を参照する制御電流源となる。SPICE に組み込まれている機能を用いることで、これまで行ってきた解析に対し非常にわずかな修正により、局所的な温度に依存するリーク電力を考慮する解析が可能となる。図 OCTE-26 はチップの温度分布とリーク電力分布の計算例である。

4.1.1.6.2 温度分布を考慮したリーク電力の解析

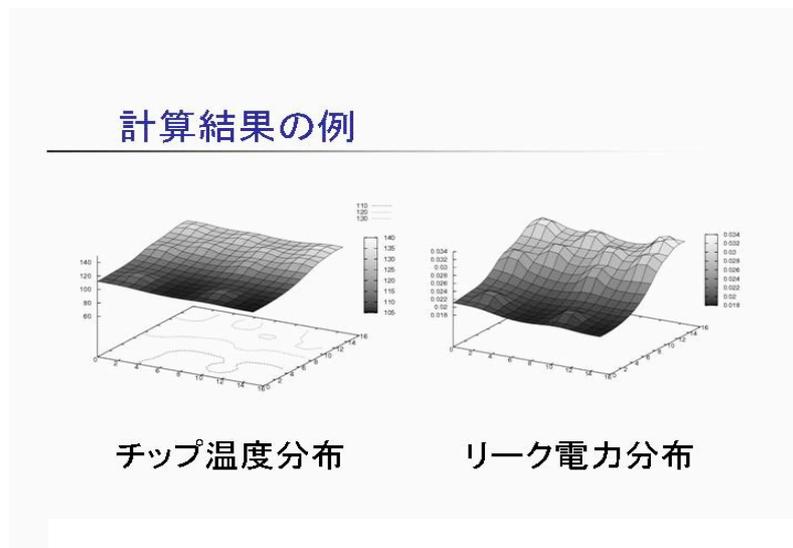


図 OCTE-26 計算結果例

以下、図 OCTE-27 に示すシナリオに従いチップ温度に対するリークの影響を求める。ここでは、リークが総消費電力に占める割合を変化させる。リーク電力の割合は、最大温度（120℃）で定義する。その際には、チップ温度が分布を持たないと仮定する。

図 OCTE-28 にリーク電力割り当ての例を示す。左側は総消費電力が 20W の場合、右側は総消費電力が 32W の場合である。電力割り当ては通常、チップの温度が一様であることを仮定して行うが、これまでの解析から温度は分布を持つ。この分布が消費電力にどのような影響を与えるかを以下で分析する。

熱設計シナリオ(1)

- チップ温度が一律最大(120°C)の条件で, 総消費電力にリークが占める割合 γ を定義
- リークは温度に対して指数的依存性⇒常温では, $\gamma=0$ の場合と比較して)AC電力が減少

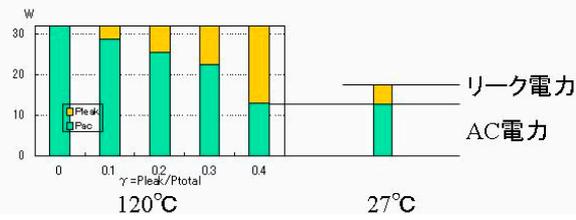


図 OCTE-27 熱設計シナリオ 1

熱設計シナリオ(2)

- チップ上の温度は実際には一律ではない
- 温度分布を考慮すると消費電力はどう変化するか?

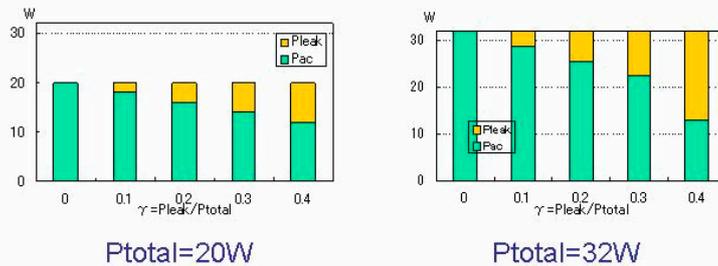


図 OCTE-28 熱設計シナリオ 2

Ptotal=20W時のリークプロファイル

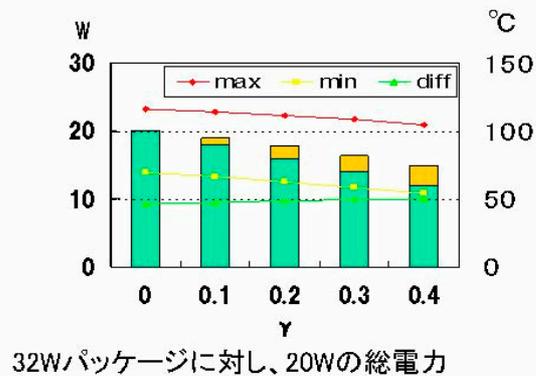


図 OCTE-29 20W パッケージでのリーク

図 OCTE-29 は総消費電力 20W 時のリークプロファイルを示す。32W まで許容できるパッケージに対して消費電力に余裕があるため、また割り当て方法からリーク電力の割合 γ が増えるほど充放電電力が減少することからリーク電力の割合が増すほどチップの平均温度が減少する。

Ptotal=32W時のリークプロファイル

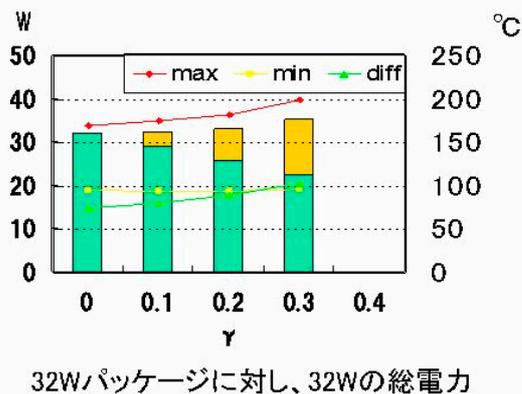


図 OCTE-30 32W パッケージでのリーク

一方、図 OCTE-30 は総消費電力 32W 時のリークプロファイルを示している。仮定した最大温度よりも温度が高い場所が局所的に存在し、またリーク式が温度に対して指数的な依存性をもつために、このケースではリーク割合が増すほどチップ温度は上昇している。リーク割合が 0.4 のケースでは収束値を得ることが出来なかった。これは温度上昇とリーク電力増加の正帰還のためと考える。

4.1.1.6.3 シミュレーション解析結果と考察

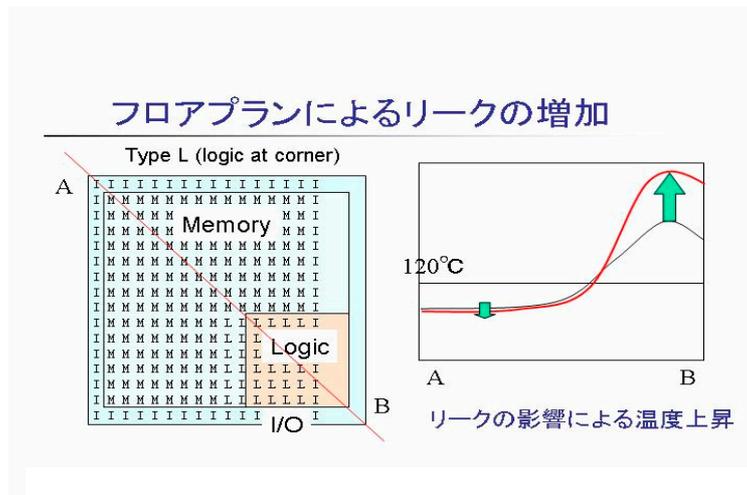


図 OCTE-31 フロアプランでのリーク増加

フロアプランによるリーク増加を、図 OCTE-31 を用いて模式的に考える。左側は今回検討したフロアプラン、右側は A-B 断面に沿った温度分布を示す。リークを考慮しない場合には、細い実線に示すようにロジック部で局所的に温度が上昇する。リークを考慮することで、高温部では大幅にリーク電力が増加する一方、平均温度よりも低い部分では消費電力が減少する。またその変化量はリーク電力の形状から高温部のほうがより大きい。このためフロアプランによりリーク電力が改善できること、および温度が高くなりすぎると正帰還による熱暴走が生じる可能性があることがわかる。

図 OCTE-32 に今回の検討結果をまとめる。

チップにおけるリークプロファイル

- 最大温度(120°C)でリーク割合が実現される電力割り当てについて検討
- 最終的にはleak分が、当初の見積もりほどあがらない。
- リークと温度の関係が指数的であり、最終温度が120°Cまで上昇しないため
- ここで考えた電力分配モデルでは、leak割合が大きいほど、Pacに関して最大消費電力、最大温度の両方について、マージンを持つ
- SIMの有効利用により、上記マージンを削減することで、より多くのゲート(消費電力要素)を搭載できるようになる
- フロアプランによりリークを改善できる可能性有り

図 OCTE-32 リークに関するまとめ

4.1.1.7 活動内容のまとめと今後の課題

2003 年度

- ・ ゲート、配線に対する熱効果の問題整理
- ・ 熱シミュレーションのモデルとツールを提案し問題を確認
- ・ メモリ占有率、フロアプラン形状、メモリの電力モデルをパラメータとして熱勾配との関連性を解析
- ・ フロアプラン改善により熱勾配が改善されることを確認

2004 年度

- ・ 熱がチップ性能に影響を与える領域（消費電力、パッケージ、等）の明確化
- ・ 熱がチップ性能（EM/スキュー）に与える影響を検討
- ・ 上記影響の緩和手法についてより具体的な検討
- ・ 影響予測の精度を向上するためリーク電流等を考慮

以上主な内容を図 OCTE-31 にまとめる。

今後の課題

- ・ チップの外部環境（初期温度）の変化による影響の調査
- ・ フロアプラン段階での電源電圧効果への影響の調査
- ・ 最終レイアウト後の、詳細な熱解析による、他のツール（スタティックタイミング解析、電源変動）との連携（局所的、全体的）
- ・ 温度制御機構を含めたシミュレーション環境の検討

まとめ

- SoCでの熱シミュレーションの提案
- 熱解析での定量的な導出
 - メモリ占有率、フロアプラン、パッケージをパラメータとて解析を行った
 - ロジックが角にある場合に最も熱勾配が大きくなる
 - 温度差 ΔT_{\max} の最大値は32W チップで75 ° Cを超える
- 温度平坦化手法の提案
 - 僅かなフロアプランの変更で ΔT_{\max} を26.1から7.4 ° Cへ削減
- リークの影響の定量的分析
 - 最終限界温度でリークを見積ると過剰見積りになる場合があり、熱シミュレーションにより多くのゲートの搭載が可能になる

図 OCTE-31 成果のまとめ

4.1.1.8 参考文献

- [OCTE-1] INTERNATIONAL TECHNOLOGY ROADMAP FOR SEMICONDUCTORS 2003 EDITION
- [OCTE-2] Clocking Design and Analysis for a 600-MHz Alpha Microprocessor Daniel W. Bailey and Bradley J. Benschneider IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 33, NO. 11, NOVEMBER 1998 1627
- [OCTE-3] Clock Distribution Networks in Synchronous Digital Integrated Circuits PROCEEDINGS OF THE IEEE, VOL. 89, NO. 5, MAY 2001
- [OCTE-4] K. Kanda, et. al. "Design impact of positive temperature dependence on drain current in sub-1-V CMOS VLSIs, " IEEE JSSC, Oct 2001
- [OCTE-5] CMOS VLSI Engineering Silicon-On-Insulator James B. Kuo and Kerwei Su (ISBN:0-7929-8272-2)
- [OCTE-6] Self-Heating Characterization for SOI MOSFET Base on AC Output Conductance
- [OCTE-7] Closed-form Analytical Thermal Model for Accurate Temperature Estimation of Multilevel ULSI Interconnects Ting-Yen Chiang and Krishna C. Saraswat 2003 Symposium on VLSI Circuits Digest of Technical Papers
- [OCTE-8] Analytical Thermal Model for Multilevel VLSI Interconnects Incorporating Via Effect Ting-Yen Chiang, Kaustav Banerje IEEE ELECTRON DEVICE LETTERS, VOL. 23, NO. 1, JANUARY 2002
- [OCTE-9] Analysis of Non-Uniform Temperature-Dependent Interconnect Performance in High Performance ICs Amir H. Ajami, Kaustav Banerjee, Massoud Pedram, Lukas

P.P.P. van Ginneken

- [OCTE-10] Thermal Modeling and Measurement of Large High Power Silicon Devices With Asymmetric Power Distribution Jeffrey Deeney
- [OCTE-11] P. R. Strickland 他、 “The Thermal Equivalent Circuit of a Transistor” IBM Journal of R&D
- [OCTE-12] Compact Modeling and SPICE-Based Simulation for Electrothermal Analysis of Multilevel ULSI Interconnects
- [OCTE-13] Sevgin Oktay、 “Parametric Study of Temperature Profiles in Chips Joined by Controlled Collapse Techniques” IBM Journal of R&D
- [OCTE-14] Warnock 他、 “The circuit and physical design of the POWER4 microprocessor” IBM Journal of R&D
- [OCTE-15] P. Singh 他、 “A power、 packaging、 and cooling overview of the IBM eServer z900” IBM Journal of R&D
- [OCTE-16] S. Sauter、 “Effect of Parameter Variations at Chip and Wafer Level on Clock Skews、 ” IEEE Trans on Semiconductor Manufacturing、 Vol. 13、 No. 4、 Nov. 2000.
- [OCTE-17] T.-Y. Wang and C.-P. Chen、 “3-D Thermal-ADI: A Linear-Time Chip Level Transient Thermal Simulator、 ” IEEE Trans on CAD、 Vol. 21、 No. 12、 December 2002.
- [OCTE-18] R. R. Thompson and H. A. Blum、 “Computer Simulation of the Thermal Environment of Large-Scale Integrated Circuits: Computer Time-Saving Techniques、 ” IEEE Trans. on parts、 hybrids and packaging、 Vol. 7、 No. 4、 pp. 168-172、 December 1971.
- [OCTE-19] S. Boucher、 Computer-Aided Thermal Design of LSI Packages、 IEEE Trans. on Parts、 Hybrids、 and Packaging、 Vol. 8、 No. 2、 pp. 44-50、 June 1972.
- [OCTE-20] H. Sanchez、 B. Kuttanna、 T. Olson、 M. Alexander、 G. Gerosa、 R. Philip、 and J. Alvarez 、 “Thermal management system for high performance PowerPCTM microprocessors、 ” in Proc. IEEE Compcon、 February 1997、 pp. 325-330.
- [OCTE-21] H. Su、 F. Liu、 A. Devgan、 E. Acar、 and S. Nassif、 “Full chip leakage estimation considering power supply and temperature variations、 ” in Proc. ISLPED、 August 2003、 pp. 78-83.
- [OCTE-22] K. Skadron、 M. Stan、 W. Huang、 S. Velusamy、 K. Sankaranarayanan、 and D. Tarjan、 “Temperature-aware computer systems: systems: Opportunities and challenges、 ” IEEE Micro、 vol. 23、 no.6、 pp. 52-61、 Nov.-Dec. 2003.
- [OCTE-23] A. H. Ajami、 M. Pedram、 and K. Banerjee、 “Effects of nonuniform substrate temperature on the clock signal integrity in high performance designs、 ” in Proc. CICC、 2001、 pp. 233-236.
- [OCTE-24] A. H. Ajami、 M. Pedram、 and K. Banerjee、 “Effects of nonuniform substrate temperature on the clock signal integrity in high performance designs、 ” in Proc. CICC、 2001、 pp. 233-236.
- [OCTE-25] J. Black、 “Electromigration - a brief survey and some recent results、 ” IEEE Trans.
-

-
- Electron Devices, vol. ED-16, no. 4, pp.338-347, April 1969.
- [OCTE-26] K. Banerjee, M. Pedram, and A. H. Ajami, "Analysis and optimization of thermal issues in high-performance VLSI," in Proceedings of the ACM International Symposium on Physical Design, 2001, pp. 230-237.
- [OCTE-27] Y.-K. Cheng and S.-M. Kang, "Fast thermal analysis for CMOS VLSI reliability," in Proc. CICC, 1996, pp. 479-482.
- [OCTE-28] Z. Yu, D. Yergeau, and R. Dutton, "Full chip thermal simulation," in Proc. ISQED, 2000, pp. 145-149.
- [OCTE-29] B. Goplen and S. Sapatnekar, "Efficient thermal placement of standard cells in 3D ICs using a force directed approach," in Proc. ICCAD, 2003, pp. 86-89.
- [OCTE-30] B. Obermeier and F. M. Johannes, "Temperature-aware global placement," in Proc. ASP-DAC, 2004, pp. 143-148.
- [OCTE-31] H. Eisenmann and F. Johannes, "Generic global placement and floorplanning," in Proc. DAC, 1998, pp. 269-274.
- [OCTE-32] C.-H. Tsai and S.-M. Kang, "Cell-level placement for improving substrate thermal distribution," IEEE Trans. on CAD, vol. 19, no. 2, pp. 253-266, February 2000.
- [OCTE-33] D. Pham, S. Asano, M. Bolliger, M. Day, H. Hofstee, C. Johns, J. Kahle, A. Kameyama, J. Keaty, Y. Masubuchi, M. Riley, D. Shippy, D. Stasiak, M. Wang, J. Warnock, S. Weitzel, D. Wendel, T. Yamazaki, and K. Yazawa, "The Design and Implementation of a First-Generation CELL Processor", in Proc. ISSCC 2005, 10.2, pp. 184-185.
- [OCTE-34] C. Poirier, R. McGowen, C. Bostak, S. Naffziger, "Power and Temperature Control on a 90nm Itanium-Family Processor" in proc. ISSCC 2005, 16.7, pp. 304-305.
- [OCTE-35] Lei He, Weiping Liao, Mircea R. Stan, "System Level Leakage Reduction Considering the Interdependency between Temperature and Leakage", in proc. DAC2003, 2.3
- [OCTE-36] Wei Huang, Mircea R. Stan, Kevin Skadron, Karthik Sankaranarayanan, Shougata Ghosh, Sivakumar Velusamy, "Compact Thermal Modeling for Temperature-Aware Design" in proc. DAC2004, 51.3.
- [OCTE-37] Peng Li, Lawrence T. Pileggi, Mehdi Asheghi, Rajit Chandra, "Efficient Full-Chip Thermal Modeling and Analysis", in proc. ICCAD2004, 4D-2.
- [OCTE-38] Jason Cong, Jie Wei, Yan Zhang, "A Thermal-Driven Floorplanning Algorithm for 3D ICs", 4C-3.
- [OCTE-39] Design and Implementation of the POWER5 Microprocessor J. Clabes¹, J. Friedrich¹, M. Sweet¹, J. DiLullo¹, S. Chu¹, D. Plass², J. Dawson², P. Muench², L. Powell¹, M. Floyd¹, M. Lee¹, M. Goulet¹, J. Wagoner¹, N. Schwartz¹, S. Runyon¹, G. Gorman¹, P. Restle³, R. Kalla¹, J. McGill¹, S. Dodson¹ ISSCC2004
-

4.1.2 電源ノイズ・タスクグループ (DPI-TG)

4.1.2.1 LSI のトレンドとノイズの影響

近年、プロセスの微細化に伴い、LSI 内の多様なノイズの影響を考慮した遅延設計が不可欠となっている。その要因として、下記に示すように、LSI の高性能化に伴う LSI のノイズ耐性の悪化が挙げられる。

- ・ LSI の高性能化
 - 微細化
 - 高速化
 - 低電圧化
 - 多ピン化
- ・ ノイズ耐性の悪化
 - ノイズレベルに対する信号レベルの相対的な低下
 - 高速動作に伴う di/dt 量の増加
 - 多ピン化に伴う SSN (Simultaneous Switching Noise) の増加

LSI 設計で考慮するノイズの種類は、その発生原因から分類すると、下記が挙げられる。

- ・ コア電源ノイズ
 - LSI 内の素子動作に伴い発生する電源・グランド電位変動。論理的誤動作やジッタ増大の原因となる
- ・ Substrate ノイズ
 - LSI 内の電源分離箇所 (ex. アナログモジュール) に基板を経由して外部 (ex. デジタル部) から流入するノイズ。該当箇所の回路動作に悪影響を及ぼす。
- ・ SSO ノイズ (I/O)
 - I/O 部の同時動作により発生する電源・グランド電位変動。I/O 論理誤動作やジッタ増大の原因となる。
- ・ Crosstalk ノイズ
 - チップ、パッケージ、PCB における信号配線間の容量性・誘導性結合により発生するノイズ。信号伝送信頼性の阻害要因となる。
- ・ EMI ノイズ
 - LSI 動作に伴い、外部に放射される電磁放射ノイズ

4.1.2.2 活動の目的

本タスクグループでは、特に数十ナノ世代の最新のテクノロジーを用いることで顕著となるだろう電源ノイズに焦点を当て、要因分析をし、今後の設計に不可欠な電源ノイズ設計指針を策定することを目的とする。

電源ノイズの問題は、以前より世の中で検討が続けられているが、電源ノイズの主要因とその影響度の関係については明確になっていないのが現状である。さらに、PCB (Printed Circuit Board) や LSI の Core 系内部の電源ノイズ等、個別の分野での評価検討は多々なされているが、それら全

体を一つの系とした、相互の電源ノイズの伝搬や影響度の検討は、殆どなされていない。そこで、それらの系に対して、簡易な等価回路モデルを作成し、電源ノイズ発生源とその影響度の要因の関係を分類整理し、それぞれの因子に対する感度解析を行う。また、要因の感度特性結果から重要度付けをし、電源ノイズ考慮設計手法や電源ノイズ検証に必要なモデルの精度を示す。

今年度は、LSI と LSI パッケージ起因の電源ノイズである SSO (Simultaneous Switching Outputs) ノイズに対する感度解析を行い、要因の重要度付けをするために必要な簡易な電源ノイズ特性評価用モデルの作成、各パラメータの感度の定量化、SSO ノイズを考慮した設計ルール作成、制約変換による設計最適化手法の提案、SSO ノイズ制約からの変換テーブル作成方法の検討、遅延変動制約からの変換テーブル作成方法の検討を行った。

4.1.2.3 SSO ノイズとは

最初に、SSO (Simultaneous Switching Outputs) ノイズに関して簡単に説明する。図 DPI-1 にあるようにオンチップにある出力バッファが、ある制御信号やデータ信号に依って、あるタイミングで同時に動作する系を考えると、出力 I/O バッファがリードフレームやパッケージの伝送線路を駆動する。この際にパッケージのインダクタンス成分などにより、電源、グラウンドラインの電位が変動することを一般的に SSO ノイズと定義する。

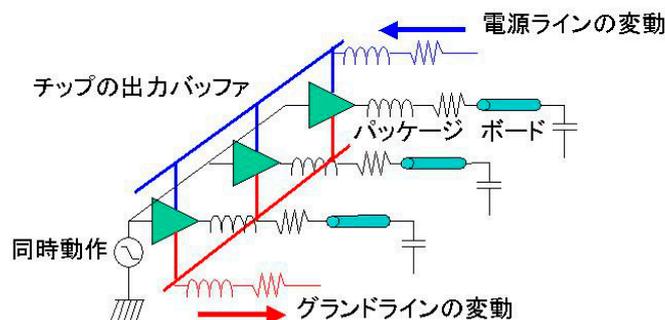


図 DPI-1 SSO ノイズ概略

4.1.2.4 SSO ノイズ解析用の回路モデル

本節では、SSO ノイズ解析用回路モデルの構成を示す。図 DPI-2 にモデリング対象を示した。今回の検討は I/O に特化して行っている。図 DPI-3 に、簡易等価モデルの概要を示す。

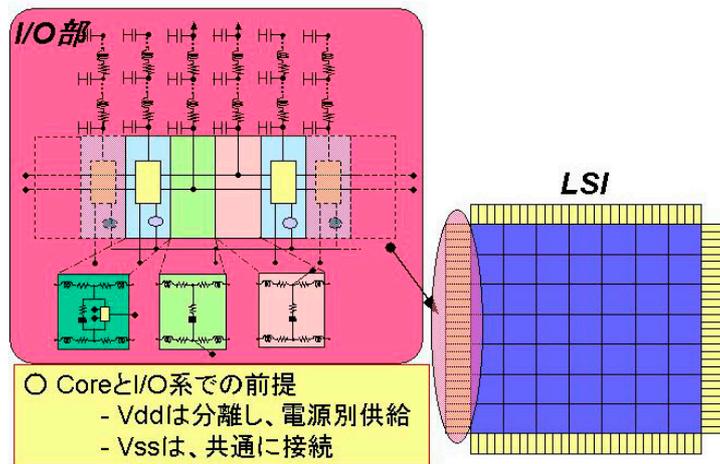


図 DPI-2 SSO ノイズ解析でのモデリング対象

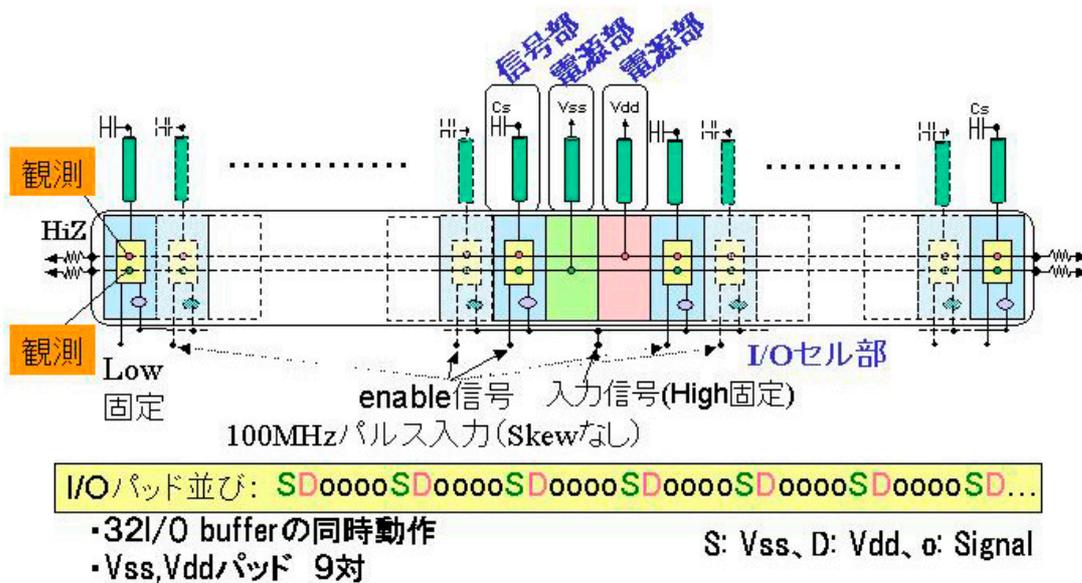


図 DPI-3 SSO ノイズ解析用モデル (全体)

チップ上の Core 領域と I/O 領域の電源分離は次のような形態を想定している。

- Core 電源と I/O 電源は分離して別々に供給
- グランドは Core と I/O とで共通に供給

図 DPI-3 は I/O 部分に着目した、SSO ノイズ解析用モデルの概要を示している。I/O パッドの並びにおける、グランド、電源、信号の順番は可変である。図 DPI-4 は、このモデルのオフチップ部分を詳細に示している。ここで表現されているのは、オフチップのリードフレームやパッケージのインダクタンス、抵抗、容量である。図 DPI-5 は、I/O セル内の詳細モデルを示している。

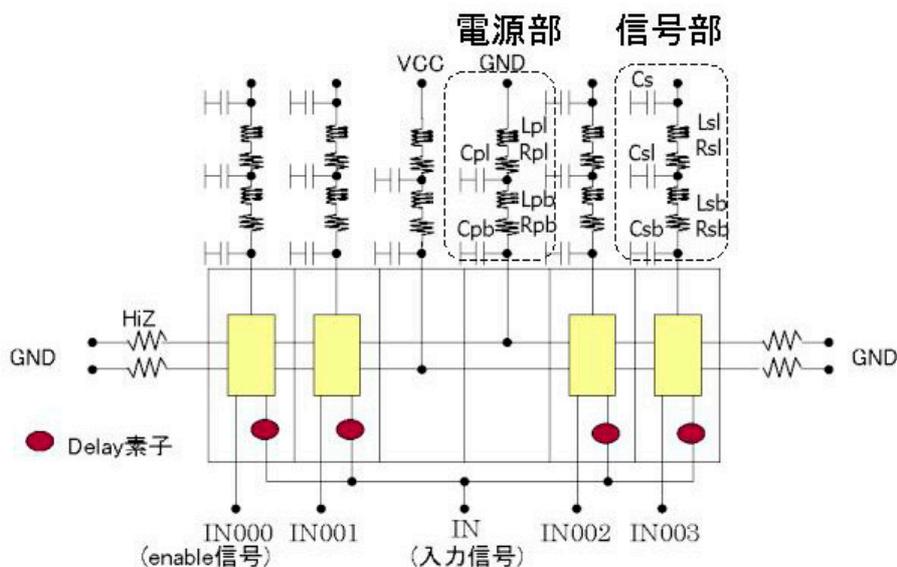


図 DPI-4 SSO ノイズ解析用モデル（電源部、信号部）

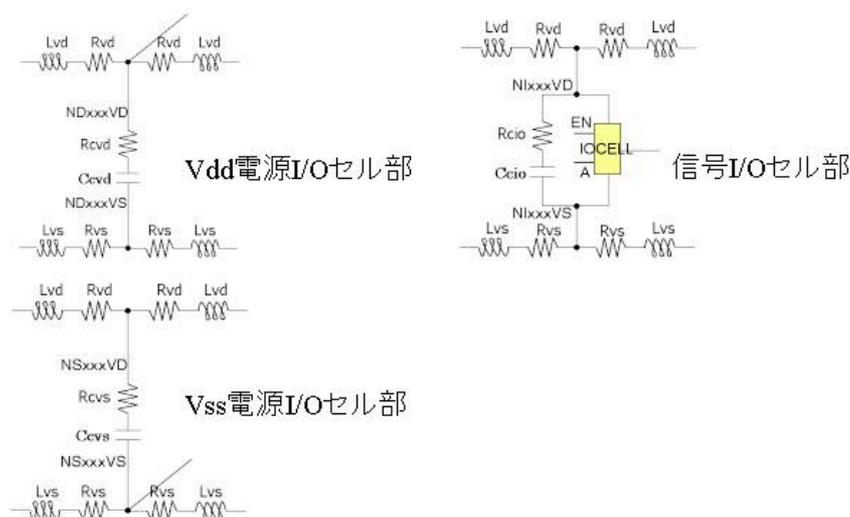


図 DPI-5 SSO ノイズ解析用モデル（I/O セル部）

このシミュレーションモデルは、次の部分から構成されている。

- LSI 内の I/O 領域に含まれている、バッファ部、電源・グランド供給配線部
- パッケージの Bonding Wire 部
- パッケージの Lead Frame 部
- ボード上の配線と受信側の LSI を模擬した負荷容量部

各ブロックの寄生成分の等価回路は RLC のラダー回路になっており、多段タイプを採用している。I/O 系の電源/グランド配線の両端は、高抵抗(HiZ)で終端をしている。信号入力の供給端子は、

すべて共通で、Enable 端子を用いて動作の有無を指定できる回路になっている。さらに、各 I/O バッファの Skew 調整用にバッファの Enable 端子と制御信号源との間に遅延素子を接続しており、それぞれ任意に遅延差の指定ができるようにしている。I/O 系の各 Cell は、図 DPI-5 内上部に示す通り、電源供給配線の寄生成分をモデル化しており、信号 I/O Cell 部は出力バッファが Vdd と Vss 間に接続している。

4.1.2.5 感度解析で使用したパラメータ

表 DPI-1 に今回設定したパラメータの一覧を示す。Lvd, Lvs はオンチップのインダクタンスを表す。Ccio, Ccvd, Ccvs は電源間のデカップリング容量を表す。Cs は負荷容量を表す。Lsb, Lpb, Lsl, Lpl はパッケージのインダクタンスを表す。

ここで、個々のパラメータを大きな系に分けた。オンチップインダクタンスは電源 Rail インダクタンス:Lring、電源間容量:Cd、負荷容量:CL、パッケージ電源インダクタンス:Lpkg と定義している。

表 DPI-1 SSO ノイズ感度解析で使用したパラメータ

パラメータ	範囲	Typ.値	説明	略称
Lvd Lvs	1p, 5p, 10p, 20p, 100p	10p[H]	電源、グランドRail インダクタンス	Lring
Rvd Rvs	-	30m[Ω]	電源、グランドRail抵抗	-
Ccio Ccvd Ccvs	10p, 50p, 100p, 200p, 1n	100p[F]	電源、グランド間容量	Cd
Rcio Rcvd Rcvs	-	3m[Ω]	電源、グランド間抵抗	-
Dy	-	1n[s]	E-element(遅延素子)	-
Cs	1p, 5p, 10p, 20p, 100p	10p[F]	終端負荷容量	CL
Lsb Lpb	100p, 500p, 1n, 2n, 10n	1n[H]	PKG B/Wインダクタンス	Lpkg
Lsl Lpl	100p, 500p, 1n, 2n, 10n	1n[H]	PKG L/Fインダクタンス	
Rsb Rpb	-	0.2[Ω]	PKG B/W抵抗	-
Rsl Rpl	-	0.1[Ω]	PKG L/F抵抗	-
Csb Cpb	-	0.2p[F]	PKG B/W容量	-
Csl Cpl	-	0.5p[F]	PKG L/F容量	-
電源I/O数	5,6,7,10,15(Vss側) 6,7,9,11,16(Vdd側)	9[対]	Vss, Vdd電源I/O対個数	電源数
駆動能力	12, 16, 24	16m[A]	I/Oセル起動能力	駆動能力
伝送路長	1,2,5,10,20,50	0[cm]	信号伝送路長。 Z0=50Ω Td=6ns/m	伝送路長

4.1.2.6 感度解析結果

前記 SSO ノイズ解析用モデルに対し、HSPICE (Synopsys 社製) を用いた過渡解析を実施し、表 DPI-1 に示したパラメータに対する感度解析を行った。本節では、その結果を示す。パラメータ毎の感度を見るため、変化対象とするパラメータのみを変化させるものとし、それ以外のパラメータは、TYP 値で固定する。尚、Vss 測定時のノイズ・ピーク量は、ノイズ・ピーク電圧値を用いるが、Vdd 側測定時のノイズ・ピーク波形は、負極性を持つため、電源電圧から Vdd 側ノイズ・ピーク値を差し引いた値を、ノイズ・ピーク量とする。

- パッケージ電源インダクタンス (Lpkg)

図 DPI-6 にパッケージ電源インダクタンスによる結果を示す。グラフ中には、SSO 数が 1 から 32 各々の場合について、パッケージ電源インダクタンスを 1p、5p、10p、20p、100p [H] と変化させた場合のノイズ・ピーク量を Vss 電源側、Vdd 電源側、個別に示した。

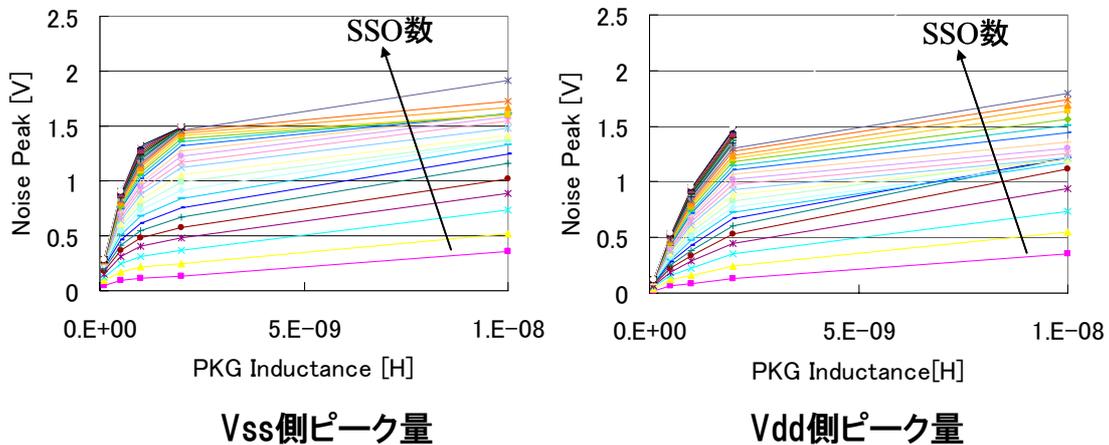


図 DPI-6 パッケージ電源インダクタンス結果

Vss、Vdd 両結果より、パッケージ電源インダクタンスの増加によりノイズ・ピーク量は増加するが、一定値を超えるとその増加量は減少する傾向（飽和傾向）にあることが判る。また、Vss 側と Vdd 側を比較すると、同一の条件では、Vss 側のノイズ・ピーク量の方が大きい。尚、Vdd 側グラフにおいて SSO 数が大きい部分のプロット点が欠落しているのは、シミュレーションの異常終了により結果が得られなかったためである。

- 負荷容量 (CL)

図 DPI-7 に負荷容量に対する結果を示す。パッケージ電源インダクタンスの場合と同様に、SSO 数が 1~32 各々に場合について、負荷容量を 1p、5p、10p、20p、100p[F] と変化させた場合のノイズ・ピーク量を示した。

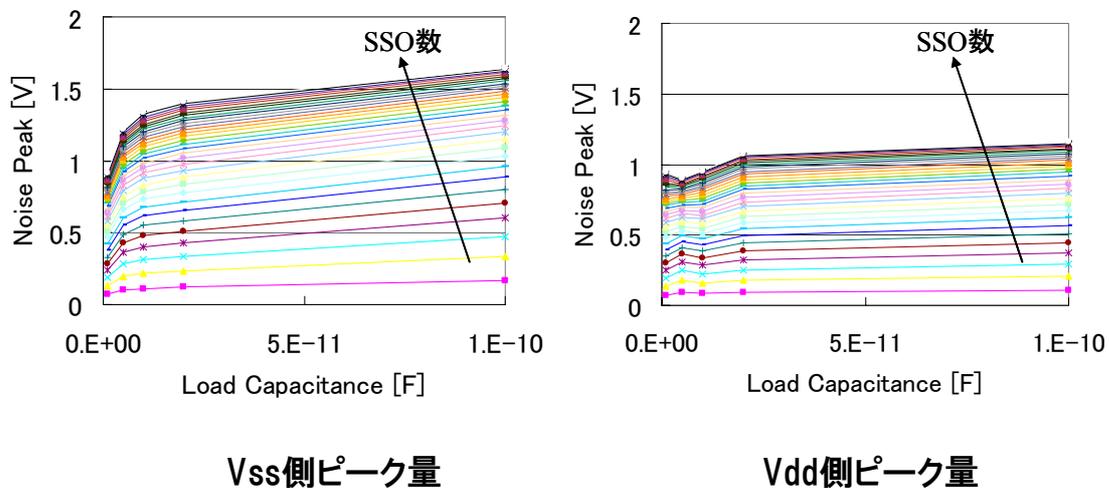


図 DPI-7 負荷容量結果

この結果より、負荷容量の増加に伴うノイズ・ピークの増加は、先のパッケージ電源インダクタンス同様、飽和傾向を持つことが判る。この飽和傾向を VSS 電源側と Vdd 電源側で比較すると、Vss 側では低負荷容量時に既にノイズ・ピーク量が飽和しているため、高負荷容量時のノイズ・ピーク量が Vss 側に比べて 2/3 程度に留まっていることが判る。また、SSO 数毎の測定点の粗密から、SSO 数に対する飽和傾向を持っていることが判る。

- 電源 Rail インダクタンス (Lring)

図 DPI-8 に電源 Rail インダクタンスに対する結果を示す。パッケージ電源インダクタンスの場合と同様に、SSO 数が 1~32 各々の場合について、電源 Rail インダクタンスを 1p、5p、10p、20p、100p[H]と変化させた場合のノイズ・ピーク量を示した。

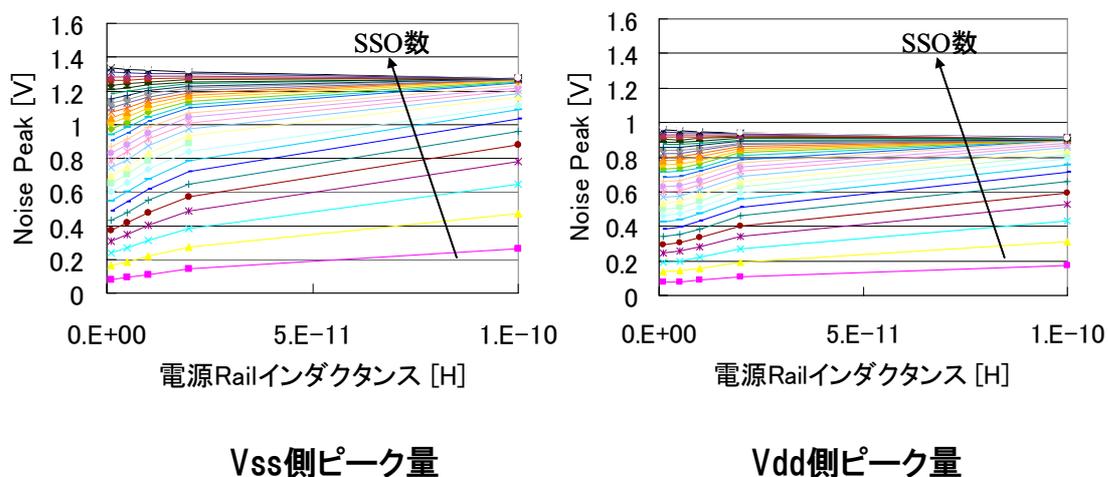
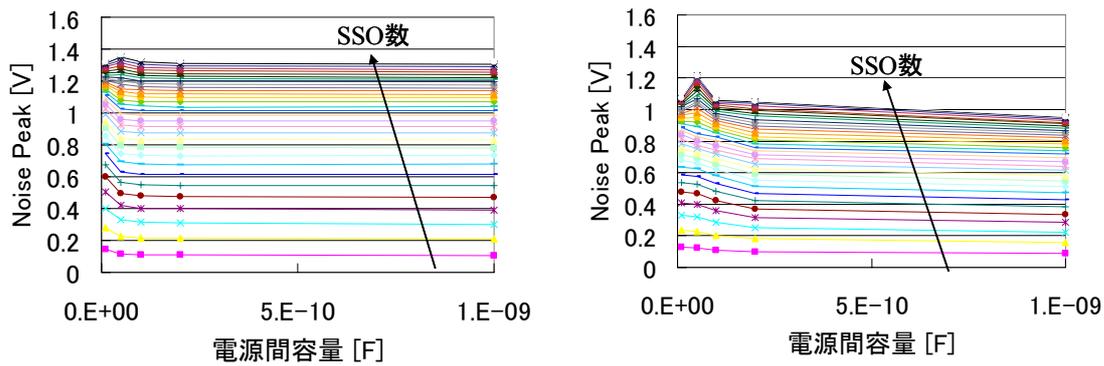


図 DPI-8 電源 Rail インダクタンス結果

この結果より、電源 Rail インダクタンスの増加に伴うノイズ・ピークの増加は、先のパッケージ電源インダクタンス同様、飽和傾向を持つことが判る。ただしその傾向はパッケージ電源インダクタンスの場合と異なり、SSO 数が大きくなると電源 Rail インダクタンスの変化に対する感度はほとんどなくなる。この様子は、Vss 側、Vdd 側共通であるが、Vdd 側でのノイズ・ピーク量は、Vss 側の 2/3 程度に留まっている。

- 電源間容量 (Cd)

図 DPI-9 に I/O 部電源間容量に対する結果を示す。パッケージ電源インダクタンスの場合と同様に、SSO 数が 1~32 各々の場合について、電源間容量を、10p、50p、100p、200p、1n[F]と変化させた場合のノイズ・ピーク量を示した。



Vss側ピーク量

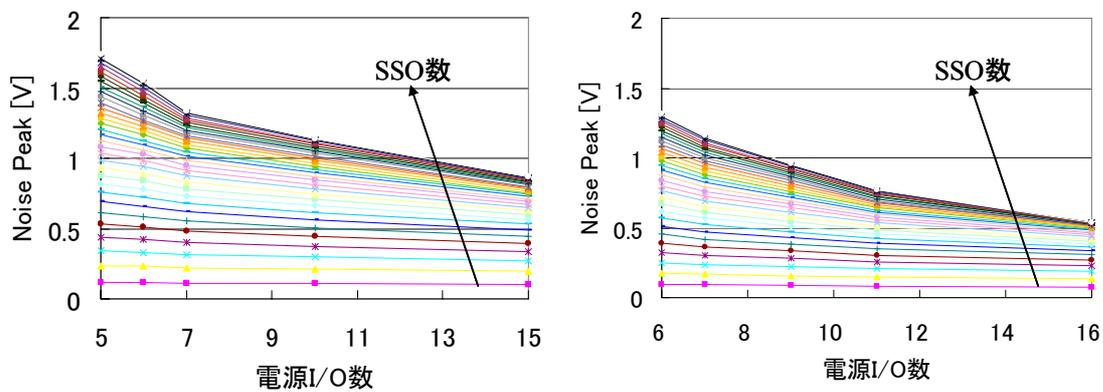
Vdd側ピーク量

図 DPI-9 電源間容量結果

この結果より、電源間容量の増減に関わらずノイズ・ピーク量は変化しないことが判る。この傾向は、Vss 側、Vdd 側、共通であるが、ノイズ・ピーク量を比較すると、Vdd 側でのノイズ・ピーク量は、Vss 側の 2/3 程度に留まっている。

- 電源数

図 DPI-10 に電源数に対する結果を示す。パッケージ電源インダクタンスの場合と同様に、SSO 数が 1~32 各々に場合について、電源・グラウンドのペア数を変化させたときのノイズ・ピーク量を示した。Vss 測定時は 5、6、7、10、15 とペア数を変化させ、Vdd 測定時は 5、7、9、11、16 とペア数を変化させて、電源・グラウンドのペアを等間隔に配置した。



Vss側ピーク量

Vdd側ピーク量

図 DPI-10 電源数結果

この結果より、電源 I/O 数の増加によりノイズ・ピーク量は減少するが、一定値を超えるとその減少量は減少する傾向（飽和傾向）にあることが判る。また、Vss 側と Vdd 側を比較すると、同一の条件では、Vss 側のノイズ・ピーク量が大きいことが判る。

- 駆動能力

図 DPI-11 に I/O セル駆動能力に対する結果を示す。SSO 数が 1、2、4、8、16、32 各々の場合について、I/O 駆動能力を、12、16、24 [mA] と変化させた場合のノイズ・ピーク量を示した。

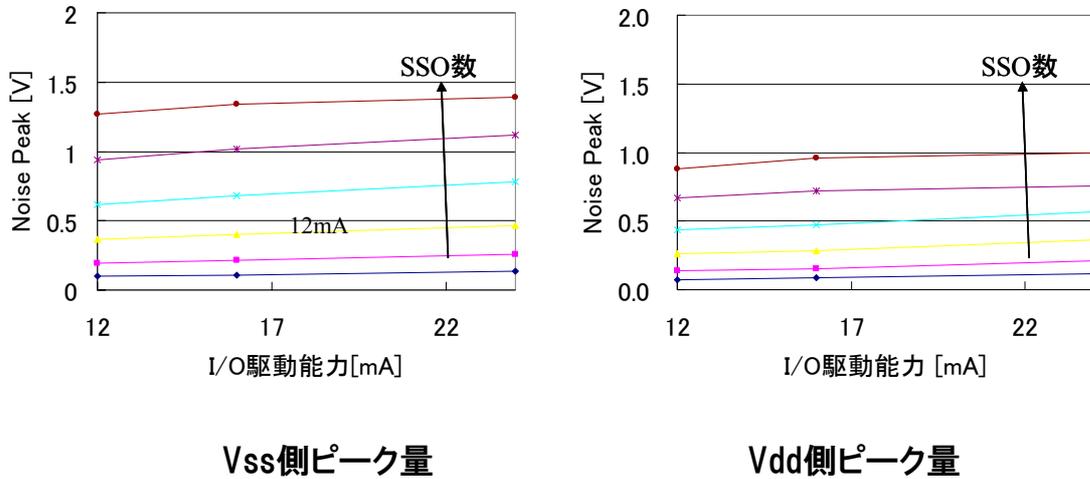


図 DPI-11 駆動能力結果

この結果より、I/O 駆動能力の増加によりノイズ・ピーク量は緩やかに増加することが判る。この傾向は、Vss 側、Vdd 側ともに共通であるが、ノイズ・ピーク量を比較すると、Vdd 側でのノイズ・ピーク量は、Vss 側の 2/3 程度に留まっている。

- 伝送線路長

図 DPI-12 に伝送線路長に対する結果を示す。SSO 数が 1、2、4、8、16、32 各々の場合について、伝送線路長を、1、2、5、10、20、50、100 [cm] と変化させた場合のノイズ・ピーク量を示した。SSO ノイズ解析用モデル上での伝送路の表現は、T 素子を用い、特性インピーダンスを 50 [ohm]、伝播遅延時間を 6n [s/m] とし、図 DPI-4 信号部での負荷容量 Cs 直前に挿入した。また、反射防止用に 50 [ohm] のダンピング抵抗を I/O 出力信号端子直後に挿入した。

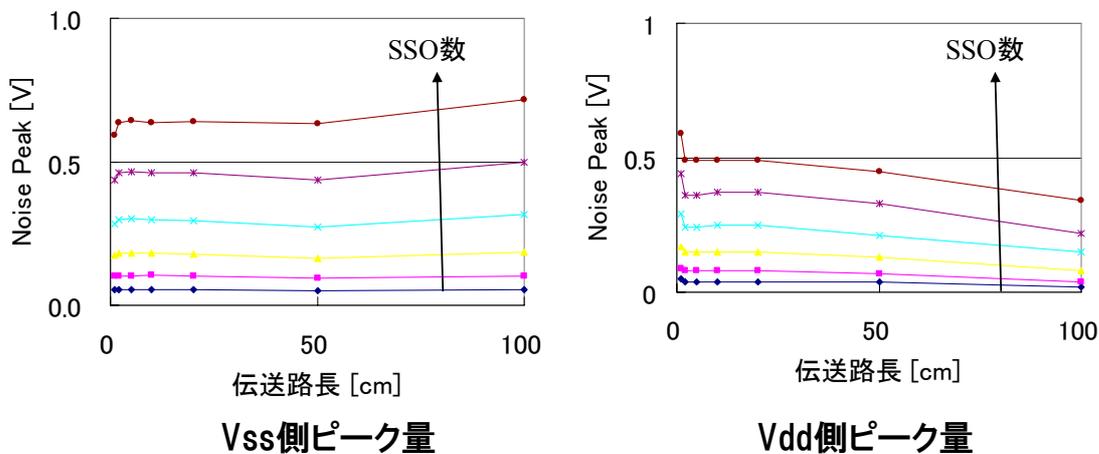


図 DPI-12 伝送線路長結果

この結果より、伝送路長の変化に対するノイズ・ピーク量は、20 [cm] 未満の領域では、ほぼ一定とみなせるが、50 [cm] を超える領域では、Vss 側は緩やかなノイズ量の増加、Vdd 側では緩やかなノイズ量の減少傾向を持っていることが判る。また、Vss 側と Vdd 側のノイズ・ピーク量を比較すると、Vss 側に対し、Vdd 側のノイズ・ピーク量は、2/3 程度に留まる。

4.1.2.7 パラメータ感度の比較

前節では、感度解析結果として SSO ノイズ解析用モデルパラメータのノイズ・ピーク量絶対値に対する影響を示した。本節では、感度解析対象とした各パラメータの変化量に対するノイズ・ピークの変化量を比較する。SSO 数が 8 の場合の比較結果を図 DPI-13 に、SSO 数が 16 の場合の比較結果を図 DPI-14 に示す。

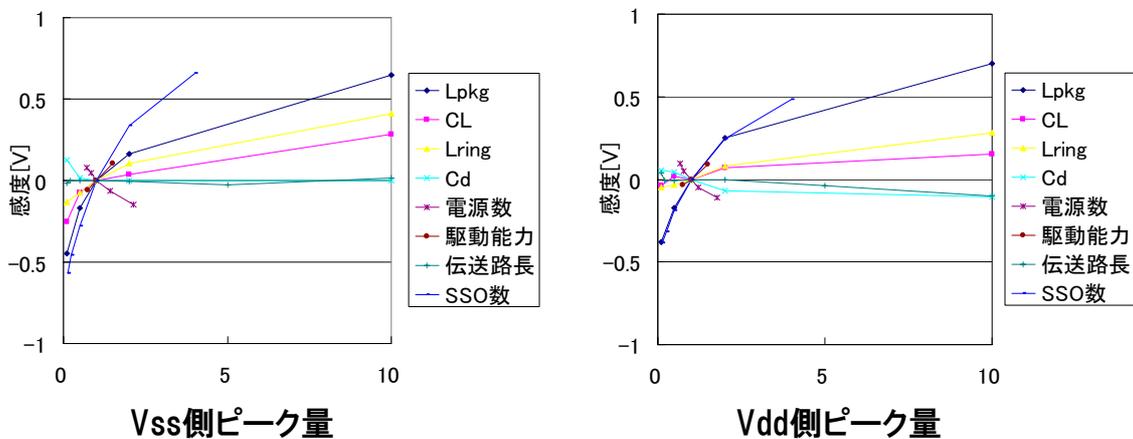


図 DPI-13 パラメータ感度比較結果 (SSO 数=8)

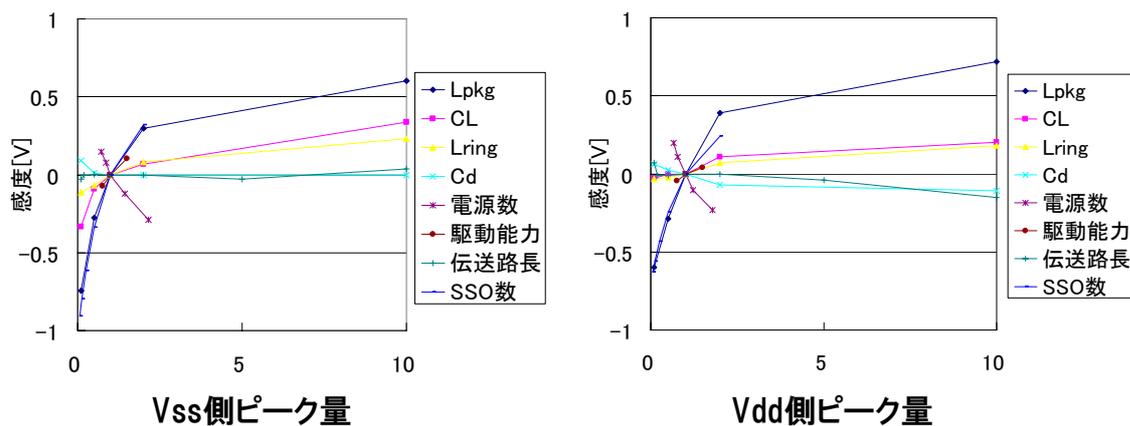


図 DPI-14 パラメータ感度比較結果 (SSO 数=16)

図 DPI-13、図 DPI-14 は、TYP 値を 1 とした場合の、各パラメータ値の変化の割合を横軸に、全パラメータが TYP 値時のノイズ・ピーク量に対するノイズ・ピーク量の変化量[V]を縦軸にとったものであり、グラフの傾きが各パラメータの変化量に対するノイズ・ピーク量の感度を示している。

図 DPI-13、図 DPI-14 を比較すると SSO 数に抛らず各パラメータの感度は、同一傾向であることが判る。Vss 側に着目すると、感度は、SSO 数が最も大きく、次いで I/O 駆動能力、パッケージ電源インダクタンス、電源数、負荷容量、電源 Rail インダクタンスとなる。電源間容量、伝送路長については、目立った感度は得られていない。この内、パッケージ電源インダクタンスと、電源数は、パラメータ値の増加によるノイズ・ピークへの作用は逆方向であるものの、その感度は同等であることが判る。Vdd 側に着目すると、電源間容量、伝送路長について若干の感度が得られている点が Vss 側と異なるものの、SSO 数、パッケージ電源インダクタンス、I/O 駆動能力、電源数の感度が高い傾向は、Vss 側と一致している。

上記結果より、SSO ノイズ・ピーク量を削減するためには、SSO 数の削減、電源 I/O 追加、パッケージ電源低インダクタンス化、I/O 駆動能力低下、負荷容量削減の順で対処するのが効果的であることが判る。

4.1.2.8 感度解析結果に基づく SSO ノイズ量削減のシナリオ

前節では、感度解析結果に基づき各パラメータの SSO ノイズ・ピークに対する感度を不得、ノイズ・ピーク量削減を図る場合の優先順位を示した。図 DPI-15 にその優先度に基づいたノイズ・ピーク量削減のシナリオを示す。このシナリオでは、ノイズ・ピーク量に対する感度が大きいパラメータを優先的に対処し、物理的制約により対処が不可能な場合に次善のパラメータで対処を図っている。

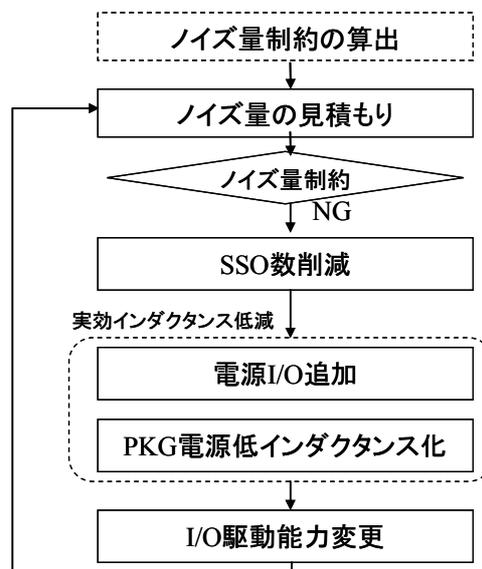


図 DPI-15 SSO ノイズ量削減のシナリオ

しかし、このシナリオでは、各ステップにて物理パラメータの変更と、それによるノイズ量の削減効果の SPICE シミュレーションによる確認を繰り返す必要があり、各ステップの TAT が長いという問題点がある。また、SSO 数削減から I/O 駆動能力変更に至る一連の処理で所望の効果が得られなかった場合には、再度 SSO 数削減から再検討を行うため、各ステップ複数回の実行が必要となる可能性があり、更に TAT の増加が懸念される。また、そもそも、この一連の処理で収束が得られる保障が無い。このため、感度解析の結果挙げられた、SSO 数、電源 I/O 数、パッケージ電源インダクタンス、I/O 駆動能力の各パラメータを、LSI の目標性能に基づいて制御するためのガイドラインもしくは設計ルールが必要となる。次節以降では、この設計ルールの作成例を示す。

4.1.2.9 SSO 設計ルール

SSO ノイズを制御するには、パッケージの電源・グランド系の実効インダクタンスや、バッファの駆動能力といった感度の高いパラメータを制御するのが効果的である。電源・グランドの実効インダクタンスは、パッケージ・タイプや電源数といった設計の初期段階で決まってしまうパラメータでほぼ決まってしまう。したがってここではフロアプラン設計フェーズ以前で適用する、次のような設計ルールを提案する（図 DPI-16）。

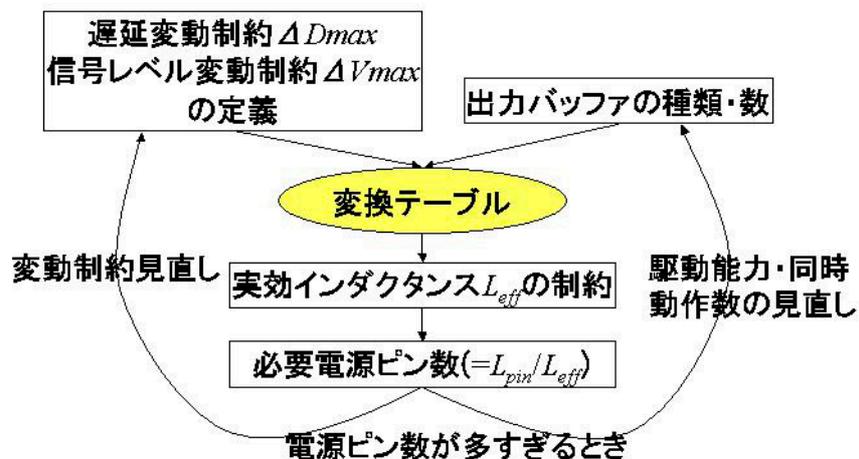


図 DPI-16 SSO 設計ルール

まず、SSO ノイズによって生じる出力バッファの遅延変動と信号レベル変動の制約を定義する（図 DPI-17）。I/O 電源およびグランド電位の変動は、静止しているバッファの出力信号レベルを変動させ、信号遷移しているバッファの遷移タイミングを変動させる。これらの変動による誤動作を避けるためにはこれらを一定値以下に抑える必要がある。

出力バッファの出力信号波形の乱れ → チップ間通信でエラー

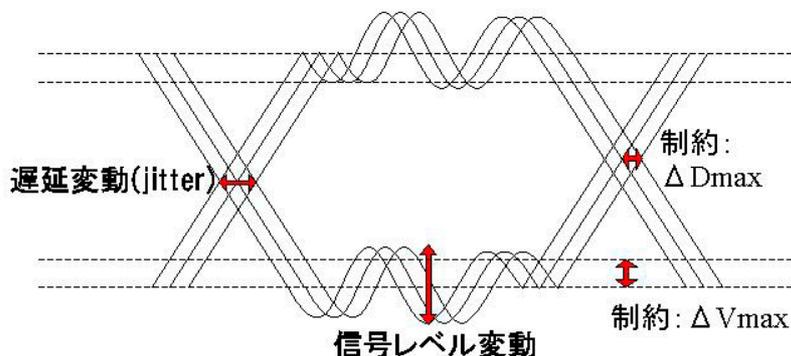


図 DPI-17 SSO ノイズの影響

これらの制約を、予め用意された変換テーブル（もしくはグラフ）を用いて実効インダクタンスの制約に変換する。これをもとに電源ピン数を求め、パッケージ・タイプの見直しを行う。十分な電源ピン数を確保できないなどの理由で実効インダクタンスの制約を満たせない場合には、出力バッファ・タイプや駆動能力を変更したり、信号レベル変動制約を緩和したりすることによって、制約間のトレードオフを解消する。

以下では、制約変換テーブルの作成方法を中心に説明する。

4.1.2.10 駆動能力、実効インダクタンスと SSO ノイズの関係

ここでは SSO ノイズとしてグラウンドバウンス、すなわち図 DPI-18 の a 点の電位変動のみを考え、グラウンドバウンスの簡易式を用いてその特性を述べる。図 DPI-18 の b 点における電位変動（電源バウンス）も同様に扱える。

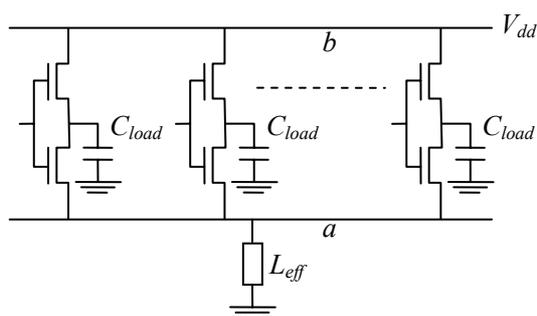


図 DPI-18 SSO ノイズ簡易式導出用回路

N 個の出力バッファの出力信号が同時に High から Low に遷移 (fall) するときを考える。最終段インバータの入力波形は次式で表されるような急峻なランプ波形と仮定し、貫通電流と電源バウンスは無視する。

$$V_{in}(t) = \frac{t}{\tau_r} V_{dd} \quad \text{for } 0 \leq t \leq \tau_r \quad (1)$$

全ての出力バッファについて、 $V_{in}(t)$ は同じと仮定する。また、グラウンドバウンスを V_n で表し、負荷容量 C_{load} は十分大きく最終段インバータの NMOS トランジスタは時刻 τ_r まで飽和領域に留まっておりバッファ出力は High のままだと仮定すると、あるバッファ i の NMOS ドレイン電流 I_i は

$$I_i = k_i \cdot f(V_{in}, V_n) \quad (2)$$

で表される。ここで、 k_i は駆動能力に応じた相対的な係数であり、ここでは相対駆動能力係数と呼ぶ。 $f(V_g, V_s)$ は NMOS トランジスタの飽和領域のドレイン電流を表す式であり、駆動能力によらず一定であると仮定する。グラウンドの実効インダクタンスを L_{eff} とし、抵抗は無視すると、グラウンド電位は次のようにあらわされる。

$$V_n = L_{eff} \cdot \sum_{i=1}^N \frac{dI_i}{dt} \quad (3)$$

式 (2)、(3) より、

$$V_n = L_{eff} \cdot K_n \cdot \frac{df(V_{in}, V_n)}{dt} = F_n \cdot \frac{df(V_{in}, V_n)}{dt} \quad (4)$$

ここで、

$$K_n = \sum_{i=1}^N k_i \quad (5)$$

$$F_n = L_{eff} \cdot K_n \quad (6)$$

と定義した。式 (4) から分かるように、 F_n が一定であればグラウンドバウンス波形が一意に決まる。例えば、

$$f(V_g, V_s) = \beta(V_g - V_t - \gamma V_s) \quad (7)$$

とすると、 V_n は $t = \tau_r$ で最大値 V_{nmax} に到達する [4]。

$$V_{nmax} = F_n \tau_r \beta \left(1 - \exp\left(-\frac{V_{dd} - V_t}{\gamma F_n \tau_r \beta}\right) \right) \quad (8)$$

NMOS トランジスタが飽和領域から線形領域に移る時刻 τ_{sat} を過ぎると、式 (4) は成り立たなくなり、負荷容量 C_{load} 、ドライバ抵抗 R_{drv} 、グラウンド・インダクタンス L_{eff} で形成されるループの減衰振動系となる。図 DPI-19 に V_n の波形の模式図を示す。

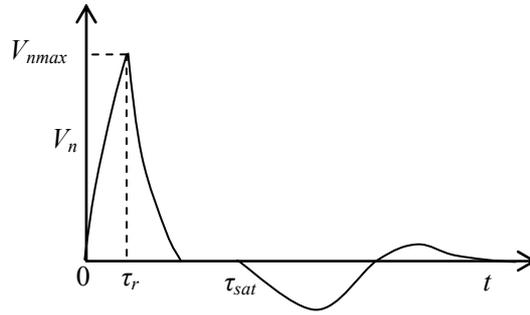


図 DPI-19 グランドバウンス波形

4.1.2.11 SSO ノイズ・ピーク制約の変換テーブル作成

図 DPI-20 の回路とパラメータを用いて SPICE を実行し、パッケージの実効インダクタンスとバッファの駆動能力、SSO ノイズ・ピークとの関係をグラフに示したのが図 DPI-21 の左の図である。図 DPI-21 の右の図は、ノイズ・ピークを一定値に抑えようとしたときに生じる、パッケージ・インダクタンスとバッファ駆動能力との間のトレードオフを示している。

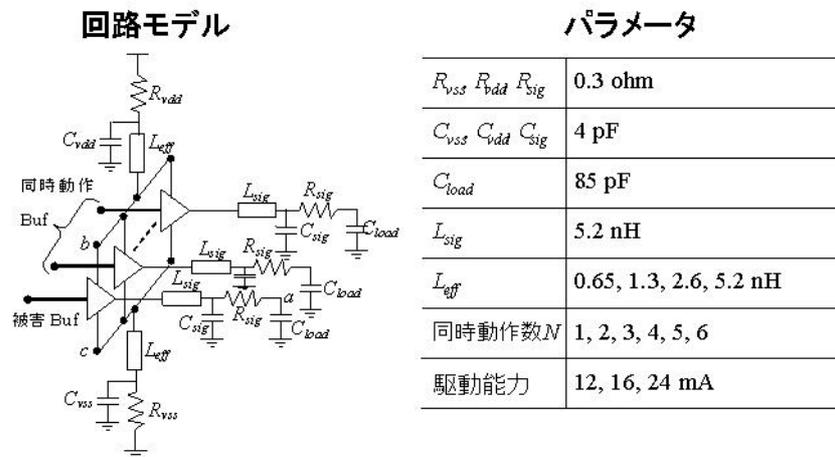


図 DPI-20 変換テーブル作成例

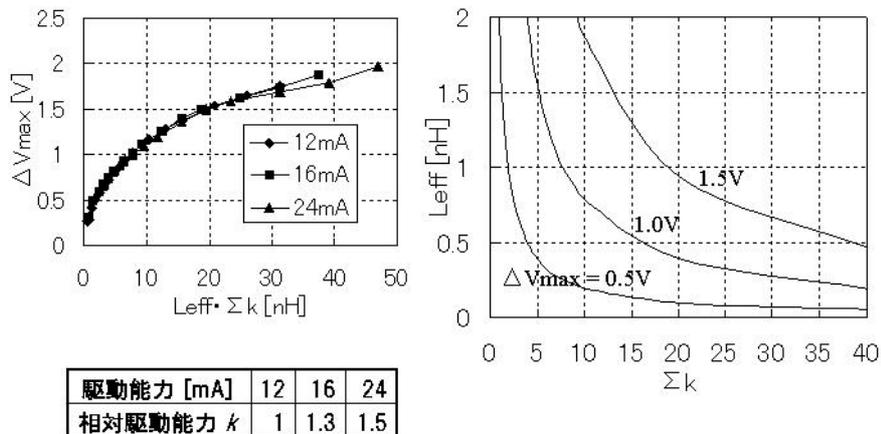


図 DPI-21 ΔV_{max} 変換テーブル

4.1.2.12 遅延変動制約の変換テーブル作成

同様に、パッケージの実効インダクタンスとバッファの駆動能力、遅延変動との関係をグラフにしたのが図 DPI-22 の左のグラフである。図 DPI-22 の右のグラフは、遅延変動を一定値に抑えようとしたときに生じるパッケージ・インダクタンスとバッファ駆動能力との間のトレードオフを示している。

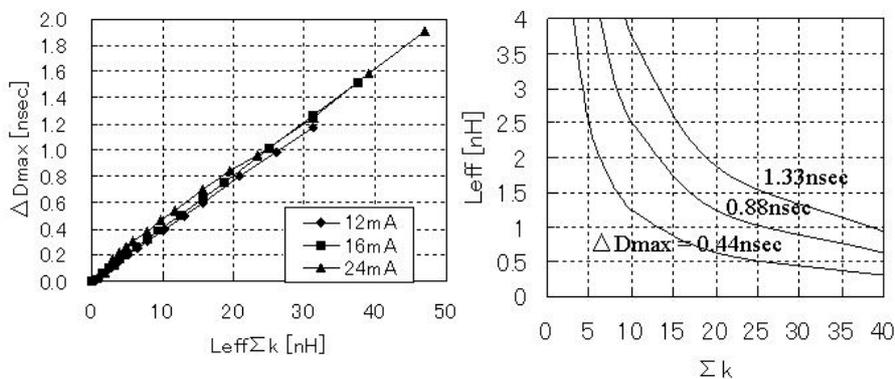


図 DPI-22 ΔD_{max} 変換テーブル

4.1.2.13 まとめ

SSO ノイズ解析用モデルを作成、感度解析を実施。支配的な物理パラメータと、パラメータ感度を得た。その結果より、支配的なパラメータを特定し、遅延変動や信号レベル変動の制約をパッケージ・インダクタンスや設計パラメータの制約に変換するテーブルを予め作成しておき、フロアプラン設計以前の設計フェーズにおいて SSO ノイズを制御する各設計パラメータを最適化する手法を提案した。

今後の課題としては、感度解析で用いた詳細モデルと SSO ルール提案で用いた簡易モデルの相関の調査が挙げられる。また、提案した設計ルールは回路やテクノロジーに依存するので更に網羅

的な検証が必要となる。その他、今回の提案は遅延が増加する場合を検討したが、減少する場合の考慮が必要となってくる。

4.1.2.14 参考文献

- [DPI-1] Y. Yang, A. Thuraijartnam, J. R. Brews, J. L. Prince, “Delay Time Estimate for FAST CMOS Drivers with Noisy Ground Reference” Electrical Performance of Electronic Packaging 1995, pp.43-45, Oct.1995
- [DPI-2] A. Vaidyanath, B. Thoroddsen, J. L. Prince, “Effect of CMOS Driver Loading Conditions on Simultaneous Switching Noise” IEEE Trans. On Components, Packaging and manufacturing Technology ? Part B, Vol. 17, No.4, Nov. 1994
- [DPI-3] Y. Eo, W. R. Eisenstadt, J. Y. Jeong, O.-K.Kwon, “New Simultaneous Switching Noise Analysis and Modeling for High-Speed and High-Density CMOS IC Package Design“ IEEE Trans. On Advanced Packaging, Vol. 23, No.2, May 2000
- [DPI-4] L. Ding, P. Mazumder, “Accurate Estimating Simultaneous Switching Noises by Using Application/On Specific Device Modeling” Proc. of DATE’02, 2002
- [DPI-5] T. Sudo, K. Nakano, J. Kudo, S. Haga, “Characterization of On-Chip Capacitance Effects for I/O Circuits and Core Circuits” Electrical Performance of Electronic Packaging 2001, pp.73-76, Oct. 2001
- [DPI-6] R.J. Evans, D. B. White, R. Pomerleau, A.Scougal, “Control of SSO Noise Through the Use of On-Chip Decoupling Cells” Electrical Performance of Electronic Packaging 2002, pp. 201-204, Oct. 2002
- [DPI-7] 蜂屋、小林、奥村、佐藤、岡、” ジッタ制約を考慮した I/O 同時動作設計ルール の提案” 回路とシステム軽井沢ワークショップ 2005

4.1.3 インターコネクト・モデリング・タスクグループ (ICM-TG)

4.1.3.1 活動目的

配線寄生素子抽出 (LPE) については、これまで種々の高精度化手法が提案されてきたが、その精度範囲を明確にすることが従来以上に求められている。

本年度は、配線遅延、およびクロストークノイズに寄与の大きい配線容量に着目し、90nm 級の先端微細プロセスで設計に用いられている商用 LPE ツールについて、抽出手法に起因する誤差要因の分析を行い、主要因に関する誤差分布を求めた。4.1.3.2 章で詳述する。

また、とくに CMP (Chemical Mechanical Polishing) プロセスを用いる Cu 配線に焦点を当て、以下の 2 点に着目した LPE におけるモデリング手法の比較・評価を行った。

- 1) 配線密度を均一化し、平坦性を向上させるためのダミーフィルパターン挿入に対する配線容量モデリング。
- 2) CMP プロセスにおいて平坦化を阻害する要因となる、ディッシング、エロージョン、リセスによる配線断面形状モデリングと配線抵抗への影響。

それぞれ、4.1.3.3 章、および 4.1.3.4 章でそれらの結果を述べる。

4.1.3.2 抽出手法に起因する誤差要因の分析

プロセスの微細化が進み、配線遅延のばらつきが先端プロセスの優位性を阻害する要因のひとつとなりつつある。また、クロストークノイズなど、シグナルインテグリティ(SI)問題も重要性を増している。

これらの検証に関わる配線寄生素子抽出については、従来から高精度化手法が提案されてきたが、ばらつきやガードバンドを議論する場合、抽出精度を統計的に示すことが必要となる。一方、配線寄生素子抽出値の実ウェア上の値との誤差は、抽出手法に起因する誤差と、配線や絶縁膜の物理寸法などの配線構造パラメータの見積もり誤差に大別でき、それぞれが独立した分布をもつと考えられる。

これまでに、配線構造パラメータの見積もり精度に関する研究はいくつか報告されてきたが、実用されている LPE ツールについて、その抽出手法に起因する誤差を系統的に示した論文はなく、当該誤差要因を考慮して遅延ばらつきを議論することは困難であった。そこで、我々は、配線遅延、およびクロストークノイズに寄与の大きい配線容量に着目し、90nm 級の先端微細プロセスで設計に用いられている商用 LPE ツールについて、抽出手法に起因する誤差要因の分析を行い、主要因に関する誤差分布を求めた。

本章では、ITRS2000update に示された 90nm 相当の SoC 配線構造パラメータを用いて求めた誤差分布を配線遅延、およびノイズ電圧値に換算して示し、誤差要因についての議論を行う。

4.1.3.2.1 先端微細プロセスに対応する LPE ツールの配線容量抽出手法と誤差要因

概ね $0.5\mu\text{m}$ 級プロセスまでは、配線層間、隣接配線間の容量成分を配線幅、配線間隔、膜厚など、2 導体系の配線構造パラメータを変数とする回帰式で表現して配線容量を求めるのが一般的であった。

$0.35\mu\text{m}$ 級以降の微細プロセスになると、2 導体系の配線構造パラメータを変数とする回帰式に対して、3 体目となる別の導体の影響を補正する、いわゆる 2.5-D (2.5-dimensional)、あるいは 2-D/3-B(2-D/3-Bodies)方式の LPE ツールが主流になってきた。同時に、予め想定した配線パターンの組み合わせに対して有限差分法などの手法を用いて配線容量を数値計算し、実レイアウトとのパターン照合により容量抽出を行う IPL (Interconnect Primitive Library) 方式 (図 ICM-1) の LPE ツールが現れ、現在に至っている。なお、回帰式を用いる LPE ツールについても、回帰式の元となる容量値の計算には IPL 方式と同様の数値計算手法を用いている。

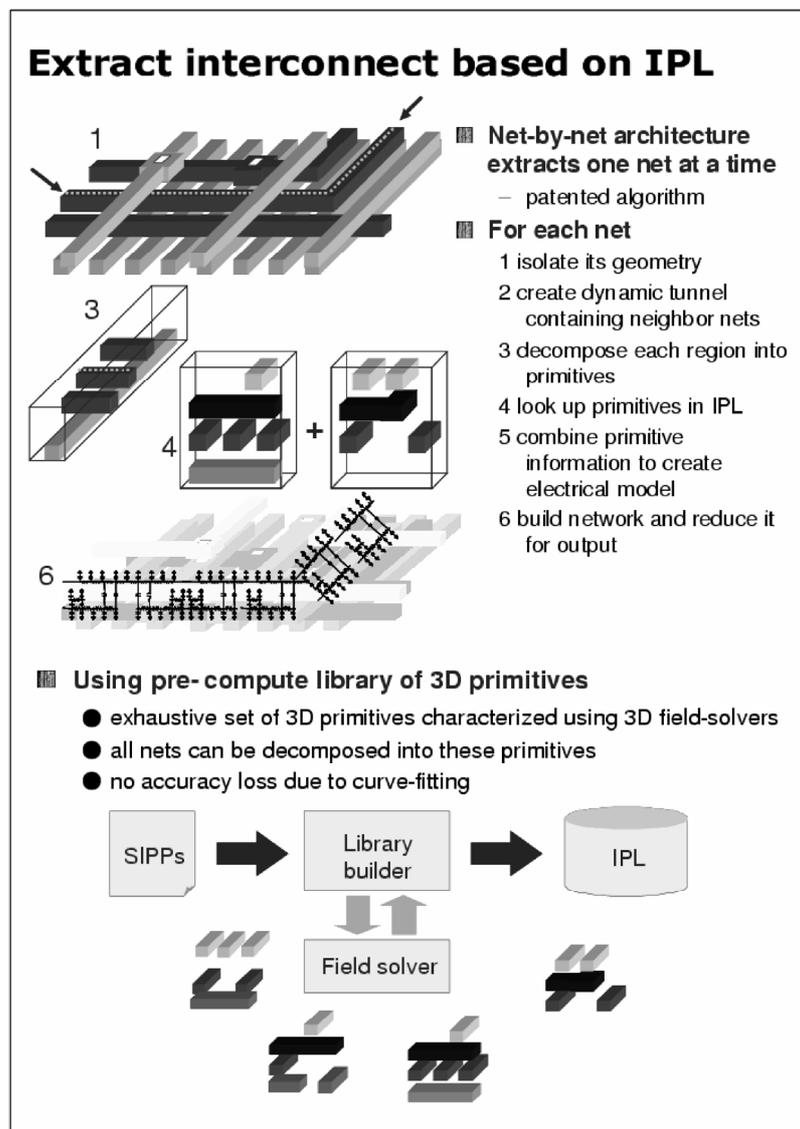


図 ICM-1 IPL (Interconnect Primitive Library) 方式 LPE

以下に、最新の LPE ツールの容量抽出誤差に対する考察を行う。

まず、回帰式を用いる LPE ツールについては、原理的な誤差要因として、回帰誤差が挙げられる。ただし、プロセスの微細化に伴い、想定する配線パターンをより小さい単位とする傾向にあり、実質的には IPL 方式と同等の精度が得られると考えられる。

一方、IPL 方式に関しては、配線パターンの電磁界解析における境界条件の、実レイアウトパターンとの違いが誤差の主因になると考えられる。ここで境界条件とは、電磁界解析の対象となる領域の境界における電場の向き、および強さに対する仮定をさす。境界条件の違いを説明する例として、平行する 3 本の配線を IPL に含まれる配線パターンと想定し、当該配線の外側領域における追加配線の有無による等電位面の変化を図 ICM-2 に示す。図より、容量抽出対象となる 3 本の配線間の電位分布が、追加された外側の配線により変化することが読み取れる。容量は、後述するように予め電位を設定した配線間の電位分布より求めるため、図示した 3 配線間の容量も

追加された外側の配線により変化する。

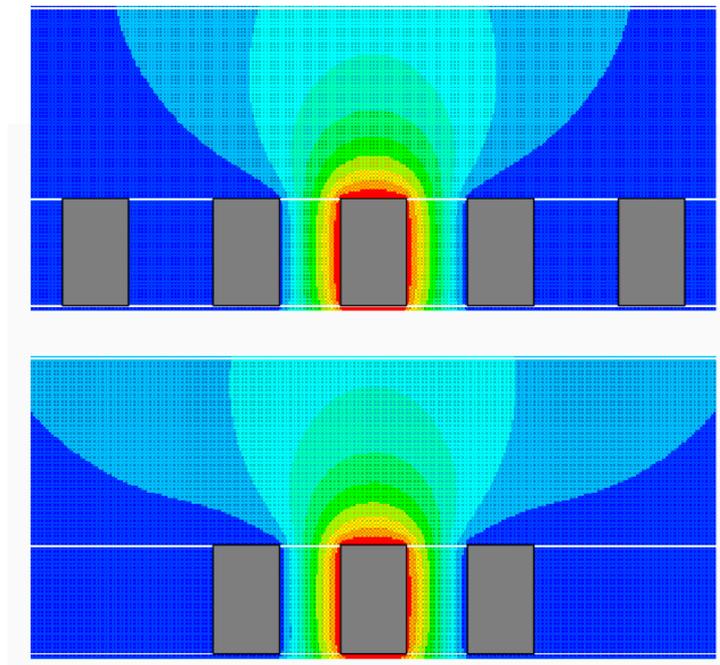


図 ICM-2 追加配線の有無による等電位面の変化

なお、LPE ツールの通常実行においては、実レイアウトパターンに対して直接数値計算を適用するわけではないため、IPL に含まれない想定外のレイアウトパターンに対しては、類似のパターンで近似を行うことになる。そのため、予想外の誤差を生じることも皆無ではない。実際、図 ICM-3 に示すレイアウトパターンを用いた予備的な評価において、図 ICM-4 に示すように比較的大きな誤差を生じるパターンが見受けられた。これらのパターン(Pattern-2、3)は、たとえば長距離のバス配線で省面積と隣接配線間容量低減を両立させる目的から、HVH 方式と呼ばれる自動配置配線の基本配線パターンではなく、上下に隣接する配線層で平行に配線を行った場合出現する。しかし、今回の評価ではこれらのパターンが LPE ツールが想定する範囲外のレイアウトパターンであったと考えられる。最新の LPE ツールでは、このようなケースを可能な限り回避すべく、数多くの配線パターンの組み合わせに対して数値計算を実行するようになっている。ただし、上述した HVH 方式の配線が基本であるため、人手で配線を行ったレイアウトパターンに適用する際にはバリエーションの網羅度に対する注意が必要である。

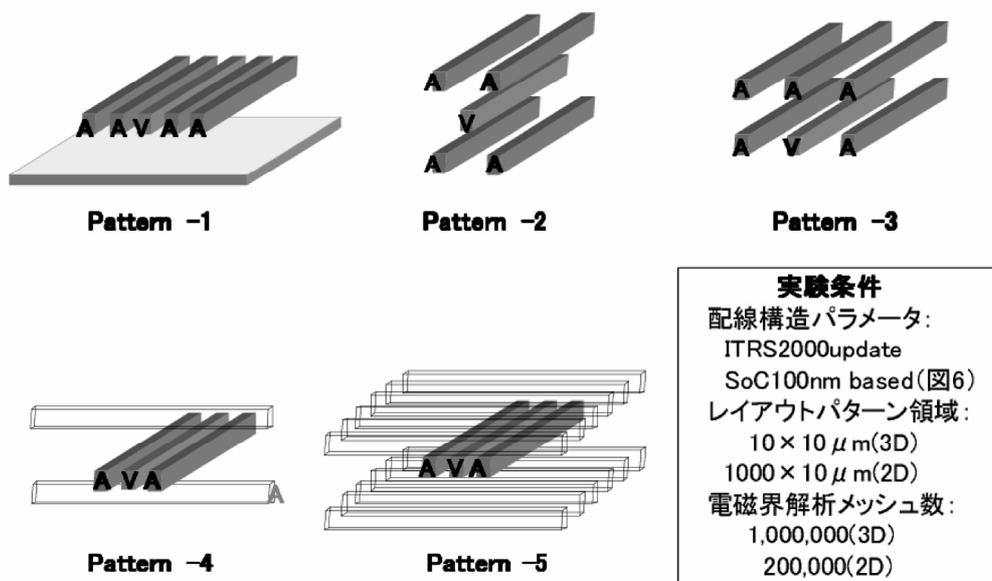


図 ICM-3 予備評価

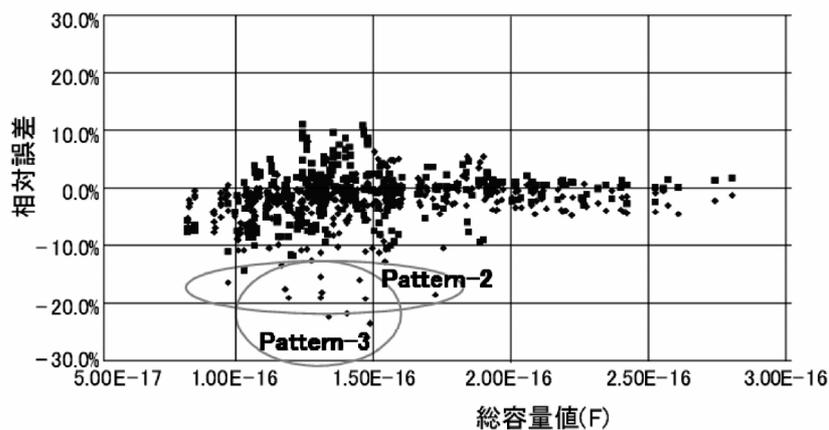


図 ICM-4 予備評価結果

以下の章では、評価対象となる LPE ツールでバリエーションが網羅された配線パターンをテストパターンとして用いる。

本評価では、Sequence 社製 Columbus-AMS、Synopsys 社製 Star-RCXT、Cadence 社製 Fire&Ice QXC を評価対象とした。なお、Columbus-AMS は主にセルキャラクターライズなどに用いるトランジスタレベル LPE を適用対象とする。一方、他の 2 者、特に Fire&Ice QXC は自動配置配線領域、いわゆるゲートレベルの LPE を主な適用対象とする。

4.1.3.2.2 評価パターン

図 ICM-BM1 および図 ICM-BM2 に、評価用のレイアウトパターンを示す。ここでは、境界条件に起因する誤差に着目する観点から、他の誤差要因、とくに IPL に含まれないレイアウトパターンに対する補完誤差が重畳しないパターンを選定した。

図 ICM-BM1 のパターンは、配線端における容量成分を無視すると 2 次元の電磁界解析に帰着できる配線パターン（以下 2D、あるいは 2D パターンと呼ぶ）である。LPE ツール評価用のレイアウトパターンでは、これに合わせて配線端における容量成分が実質的に無視できるよう、配線長を $1000\mu\text{ m}$ とした。また、容量の基準値が電磁界解析の境界条件の影響を受けないよう、領域幅は $10\mu\text{ m}$ とした。

配線構造は、配線層 Layer n の平行 5 配線を上配線層 Upper layer m、および下配線層 Lower layer o のプレートで挟む構造とした。Layer n、Upper layer m、および Lower layer o は、それぞれ $n = [2, 7]$ 、 $m = [n+1, n+2]$ 、 $o = [n-2, n-1]$ の範囲とした。

Layer n の平行 5 配線のうち、中央の配線を評価対象の信号配線とする。平行 5 配線の幅 WV、WAL1、WAL2、WAR1、WAR2、および間隔 SAL1、SAL2、SAR1、SAR2 はすべて独立に、最小値に対して $\times 1$ 、 $\times 2$ 、 $\times 3$ の間で変化させた。なお、それらの組み合わせについては、9 因子 3 水準の直交表 L27 に従い割付けを行った。

一方、図 ICM-BM2 のパターンは、上下の配線層が直交する部分で 3 次元の解析を必要とするレイアウトパターン(以下 3D、あるいは 3D パターンと呼ぶ)である。

配線構造は、配線層が直交する部分の誤差を強調するため、配線層 Layer n の平行 3 配線を直上配線層 Upper layer n+1、および直下配線層 Lower layer n-1 の楕形配線で挟む構造とし、着目配線層 Layer n は、 $n=[2, 7]$ の範囲とした。上下の楕形配線は、それぞれ電氣的に同一ノードとして扱う。

Layer n の平行 3 配線のうち、中央の配線を評価対象の信号配線とする。平行 3 配線の幅 WV、WAL、WAR、間隔 SAL、SAR、上下配線幅 WT、WB、および上下配線間隔 ST、SB はすべて独立に、最小値に対して $\times 1$ 、 $\times 2$ 、 $\times 3$ の間で変化させた。なお、それらの組み合わせについては、2D パターンと同様、9 因子 3 水準の直交表 L27 に従い割付けを行った。

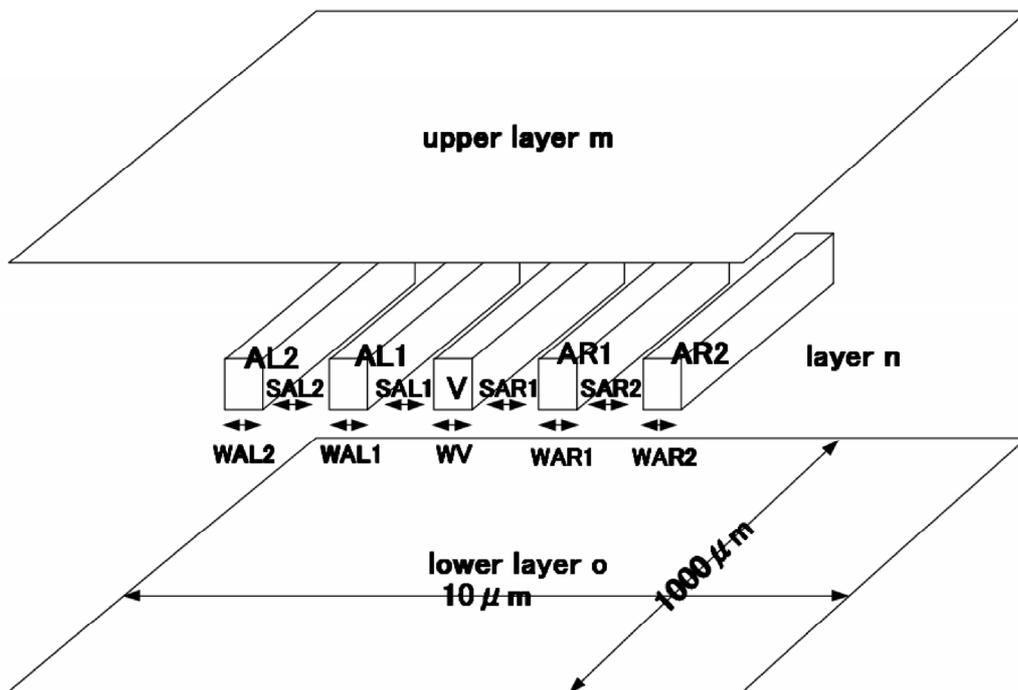


図 ICM-BM1 2D 評価用パターン

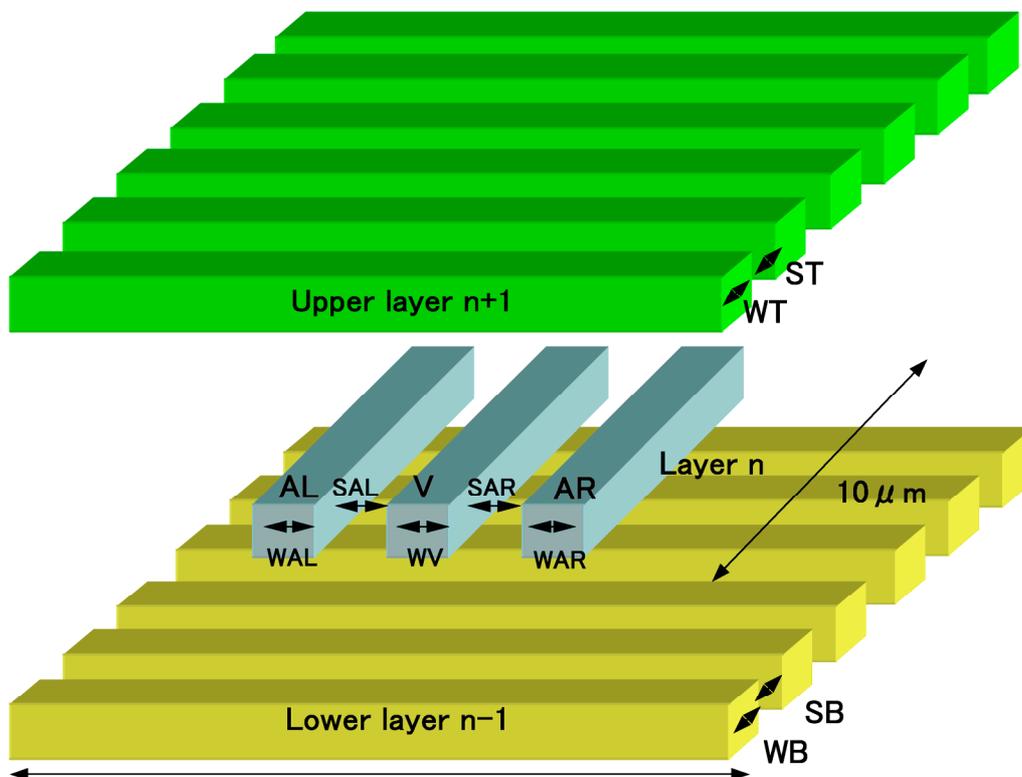


図 ICM-BM2 3D 評価用パターン

4.1.3.2.3 評価結果

(1) 2D パターン

図 ICM-5~9 に、2D パターン評価で得られた誤差分布のグラフを示す。図 ICM-5~7 は遅延値に関する誤差分布、図 ICM-8、9 はクロストークに関する誤差分布である。各ヒストグラムには、実験値の平均 μ と標準偏差 σ に対する $N(\mu, \sigma)$ の正規分布のグラフを併記した。

遅延計算に関しては、配線の総容量値の誤差が遅延時間見積り精度を規定するため、遅延時間換算後の誤差分布（図 ICM-5）に加え、総容量値の誤差分布（図 ICM-6）を示す。

一方、クロストークノイズについては、評価結果として V_{peak} の見積り誤差分布（図 ICM8）に加え、Cc の誤差分布（図 ICM-9）を示す。

2D パターンの評価結果から、以下のことがわかる。

まず、容量抽出誤差の分布は正規分布に近い連続分布になっており、本パターンに対しては境界条件以外の大きな誤差要因は重畳しておらず、境界条件に起因する誤差にも特定の偏りが無いことがわかる。さらに、容量抽出誤差が規定する遅延時間見積り誤差、ノイズ電圧見積り誤差分布とも、同様な分布を呈している。これにより、着目した境界条件に起因する誤差要因に関しては、統計的に評価できることが確認できた。具体的には、遅延時間見積り誤差は $3\sigma=1.13\%$ 、ノイズ電圧見積り誤差は $3\sigma=3.04\%$ となった。なお、参考のため、信号を駆動するドライバを $\times 1$ から $\times 4$ まで変えた遅延時間見積り誤差を示す（図 ICM-7）。

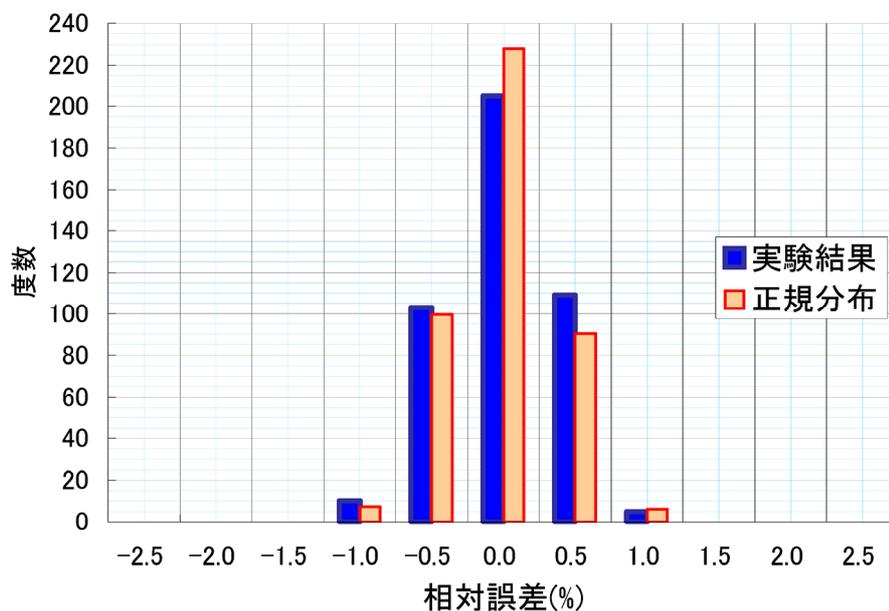


図 ICM-5 遅延誤差分布

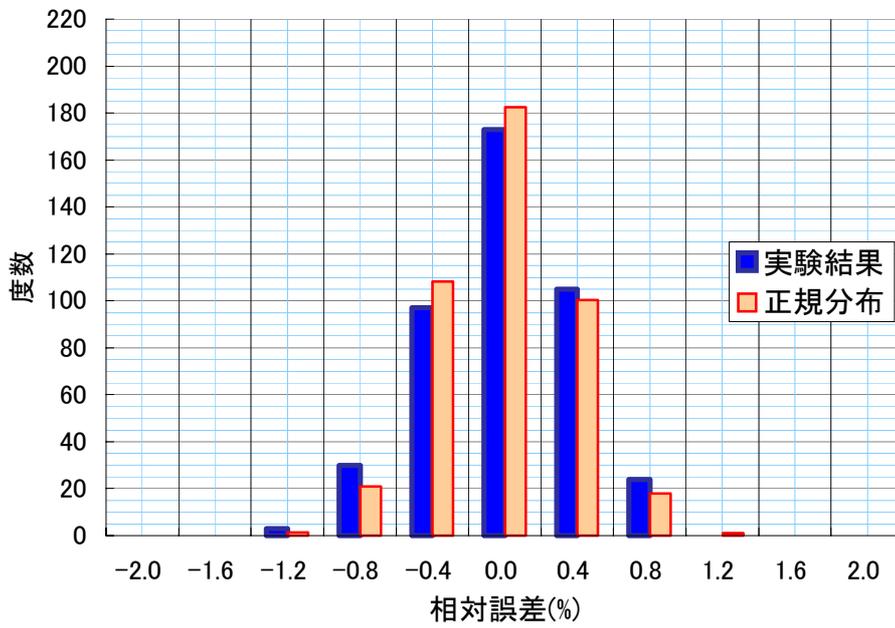


図 ICM-6 総容量誤差分布

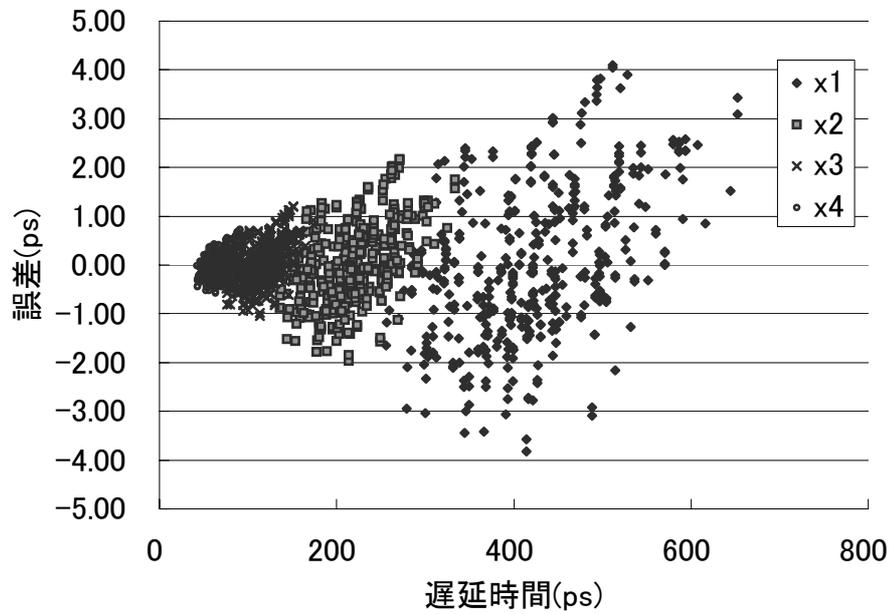


図 ICM-7 遅延誤差分布

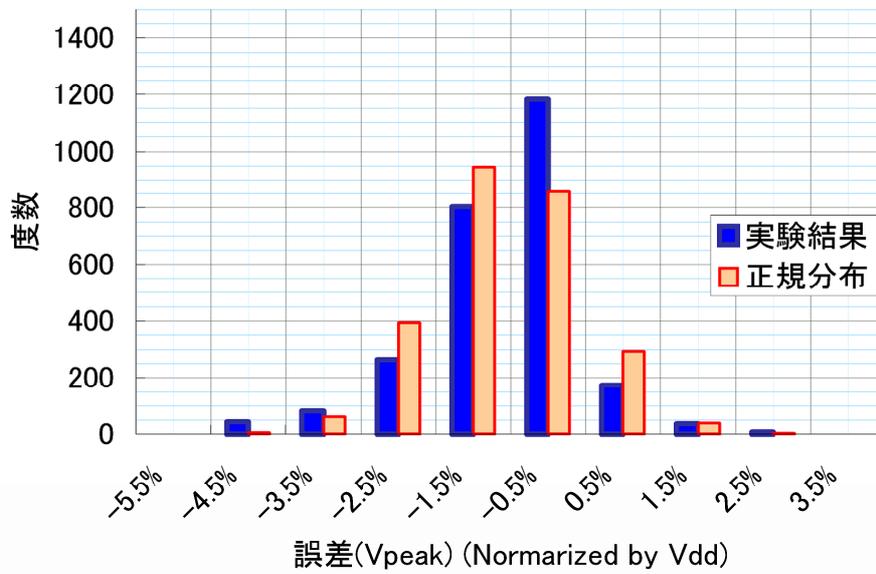


図 ICM-8 クロストークノイズ電圧誤差分布

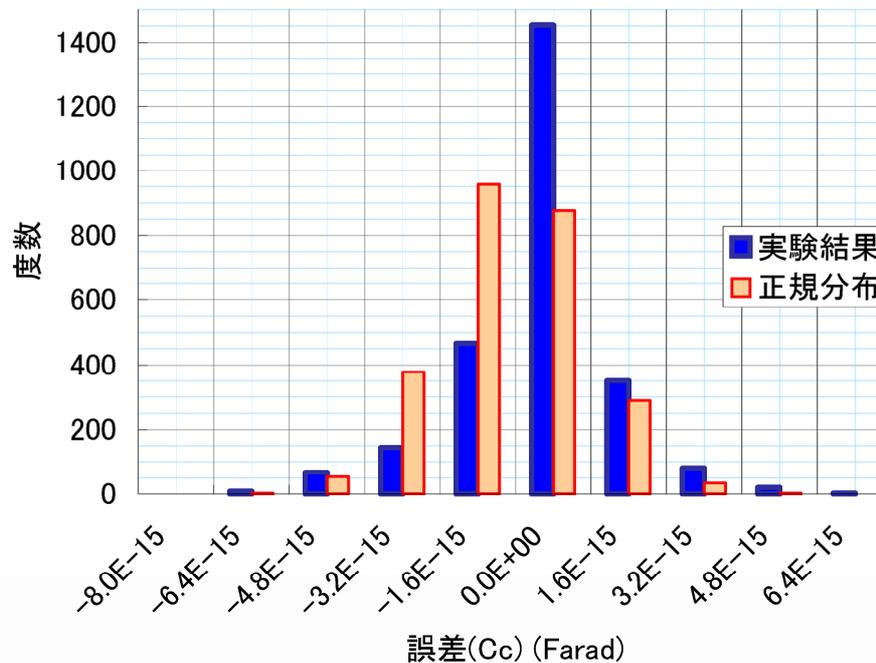


図 ICM-9 カップリング容量誤差分布

(2) 3D パターン

図 ICM-10、11 に、3D パターン評価で得られた誤差分布のグラフを示す。図 ICM-10 は遅延見積もり誤差の分布、図 ICM-11 はクロストーク電圧見積り誤差の分布である。

3D パターンは上下の直交配線を含む 3 次元構造となっており、評価対象配線の隣接方向だけでなく、奥行き方向の境界条件の差が誤差要因となるが、誤差分布自体は 2D パターンと

同じく正規分布に近い連続分布になっており、本パターンに対しても境界条件以外の大きな誤差要因は重畳しておらず、境界条件に起因する誤差にも特定の偏りがないことがわかる。遅延時間見積り誤差は $3\sigma=4.74\%$ 、ノイズ電圧見積り誤差は $3\sigma=1.50\%$ となった。

なお、クロストークノイズ電圧見積り誤差分布の 3σ が2Dに比べて減少している。この理由としては上下の配線層を直上下に制限したことにより、総容量値に対する平行配線間容量値の割合が相対的に小さくなったことと、平行配線間容量値を求める際の上下方向の境界条件誤差が緩和されたことが挙げられる。

一方、遅延時間見積り誤差は2Dパターンに比べて増加しているが、これは着目する配線と上下配線層が直交する部分の誤差に起因する。図 ICM-12 に示すように、着目配線と下層配線との間の容量誤差は、着目配線が下層であるほど減少する。これは、電磁界解析と LPE の間で基板面は境界条件が一致しているため、基板面に近いほど誤差が緩和されるためであると考えられる。これを証明するために、図 ICM-13 に示す配線パターンに対し、直交する単独配線間の容量誤差を求めた。本パターンでは、着目配線と直交する下層配線に対し、電氣的に別ノードの隣接配線を配置する。隣接配線を含め、下層配線は3本、5本、7本のバリエーションとした。実験結果を図 ICM-14 に示す。

図 ICM-14 から、評価対象の LPE ツールでは下層配線が5本のパターンに対しほとんど誤差を生じておらず、このパターンが IPL に存在すると考えられる。また、図 ICM-BM2 の配線パターンは下層配線がより多くアレイ状に配置されているため、+方向の誤差を生じていたことがわかる。

さらに、着目配線が基板面に近いほど、誤差が緩和されることが示された。

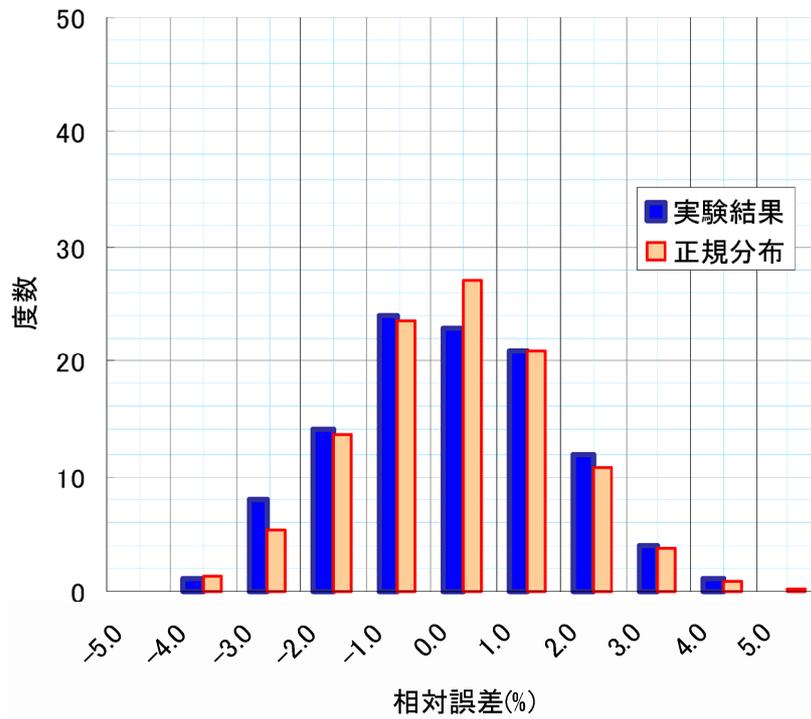


図 ICM-10 遅延換算誤差分布

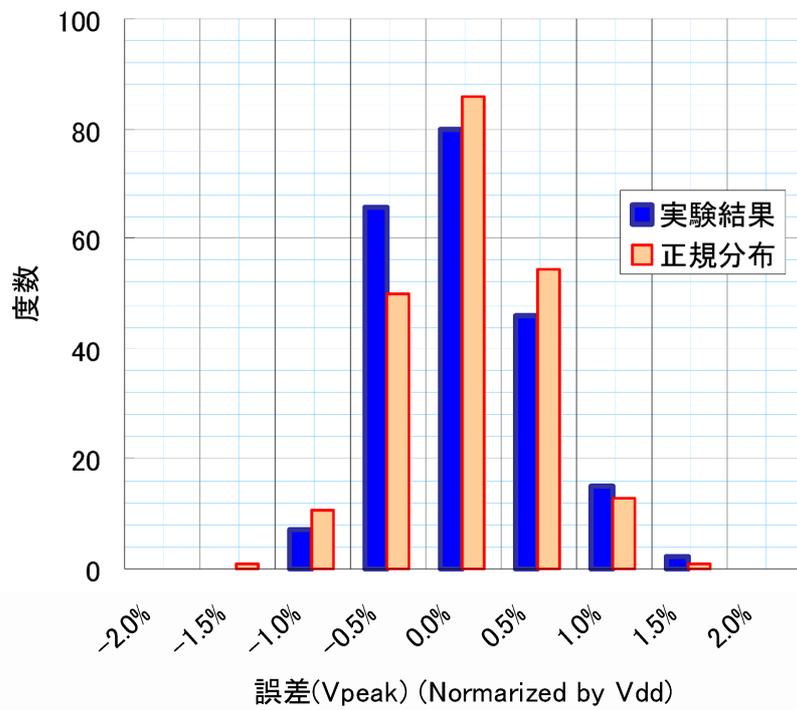


図 ICM-11 クロストーク電圧換算誤差分布

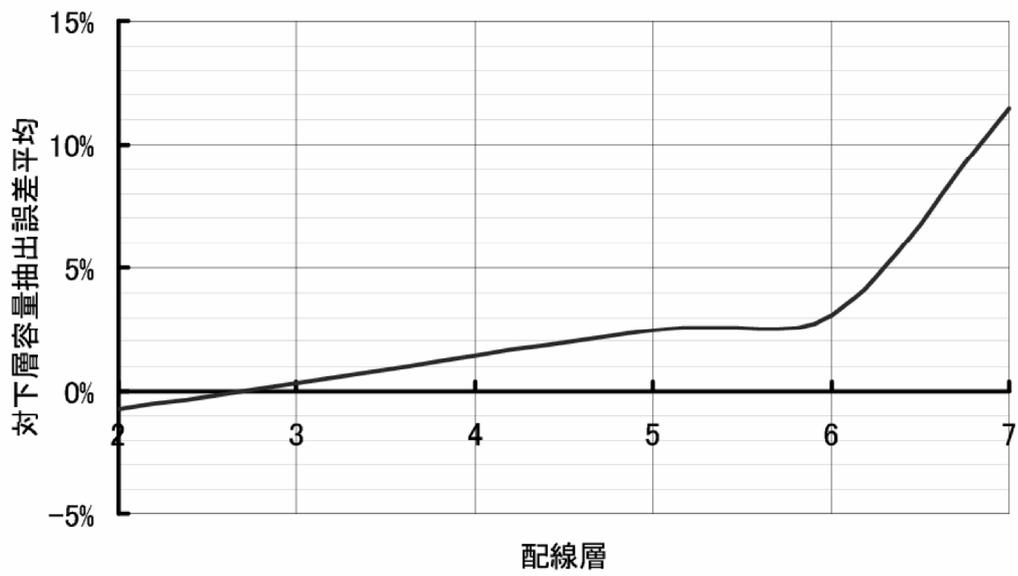


図 ICM-12 着目配線と下層配線との間の容量誤差

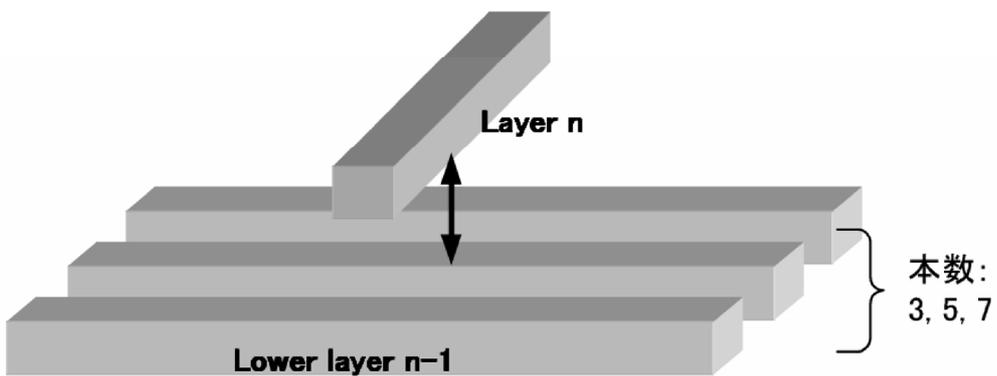


図 ICM-13 誤差の基板距離依存評価用パターン

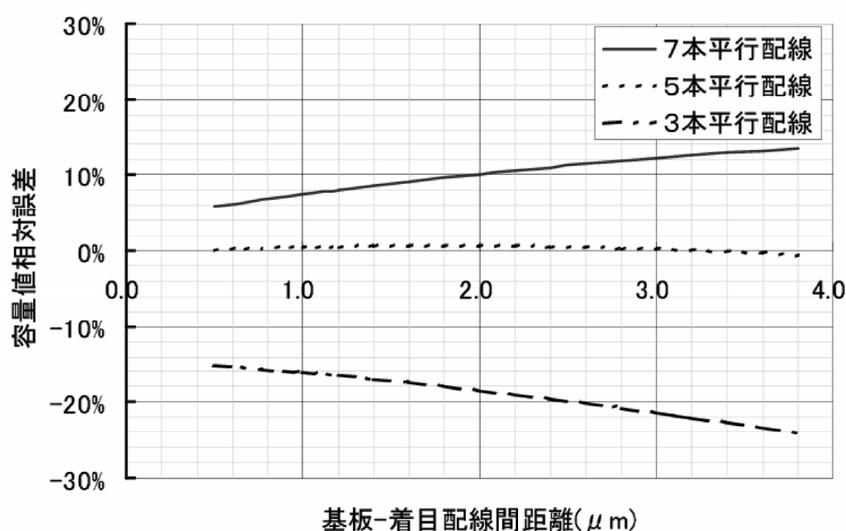


図 ICM-14 誤差の基板距離依存評価結果

4.1.3.3 CMP ダミーメタルフィル考慮容量抽出モデル評価

製造工程において、多層配線プロセスの各層を平坦にするために、CMP (chemical-mechanical polishing) 処理が行われる [ICM-1]-[ICM-3]。その研磨において、配線のメタル材料と絶縁体の材質の違いにより、不均一なメタル分布は高低差を著しくする。平坦性を向上させるために、一般に配線のない領域にはダミー・メタル・フィルを挿入する。このダミー・フィルは物理設計におけるタイミング精度に大きく影響を及ぼす [ICM-4]-[ICM-8]。

ダミー・フィルによるメタル・デンシティや配線容量への影響を考慮したフィリング技術が提案されている [ICM-9]-[ICM-13]。またダミー・フィルを考慮した配線容量抽出方法は、3D フル・シミュレーションによる方法 [ICM-14]、ダミー・フィルを含む絶縁体の等価実効誘電率による方法 [ICM-6]、絶縁膜厚縮約による方法 [ICM-7] 等が提案されている。フル 3D による方法は、高精度であるが処理時間が膨大なため実際の VLSI には計算機リソース (処理時間と使用メモリ) の観点で適用することは難しい。

等価実効誘電率と膜厚縮約による方法はダミー・フィル層を近似するので数%以上の誤差が生じる場合がある [ICM-5]-[ICM-7]。ダミー・フィル・パターンはいくつか提案されている [ICM6]、[ICM-7]、[ICM-10]、[ICM-13] (図 ICM-15)。近年、信号線容量を均一に保つという利点 [ICM-8] で一般に採用されているフィル・パターンは真上 (XY 平面) から見たダミー・メタル形状は正方形に近く、XY 方向に一定にシフトした配置である [ICM-6]、[ICM-7]。

本レポートでは、真上から見たダミー・メタル形状は正方形とし、そのダミー・メタルを XY 方向に均一にシフトした一般的なパターンを対象とする。

層間ダミー・フィルは配線容量に与える影響が大きい層内ダミー・フィルの影響は相対的に小さい [ICM-8]。そこで今回は、層内ダミー・フィルを含まない配線断面を評価対象とする。

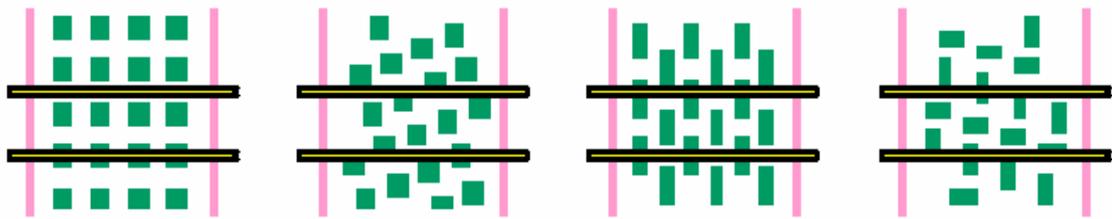


図 ICM-15 ダミー・フィル・パターンのバリエーション

4.1.3.3.1 CMP ダミーメタルフィル考慮容量抽出モデル

CMP ダミーの配線寄生容量に与える影響をモデル化する手法は、図 ICM-16 に示す 3 通りの手法に大別される。図 ICM-17 に、それらのうち配線間/層間の誘電率を変更する手法について提案されているフロー図を示す [ICM-15] [ICM-16]。

今回は LPE ツール評価の観点から、これらのうち LPE ツールが有するダミーを仮想した容量ライブラリ生成を行う手法について、その精度の評価を行った。

・ CMPダミーモデリング手法

モデリング手法	特徴	精度
配線間/層間誘電率変更	従来のLPEに適用可	△
ダミーを仮想した容量ライブラリ	処理データ量は不変	○
実ダミーパターン挿入	処理データ量が膨大	◎

→「ダミーを仮想した容量ライブラリ」手法を評価

図 ICM-16 CMP ダミーモデリング手法

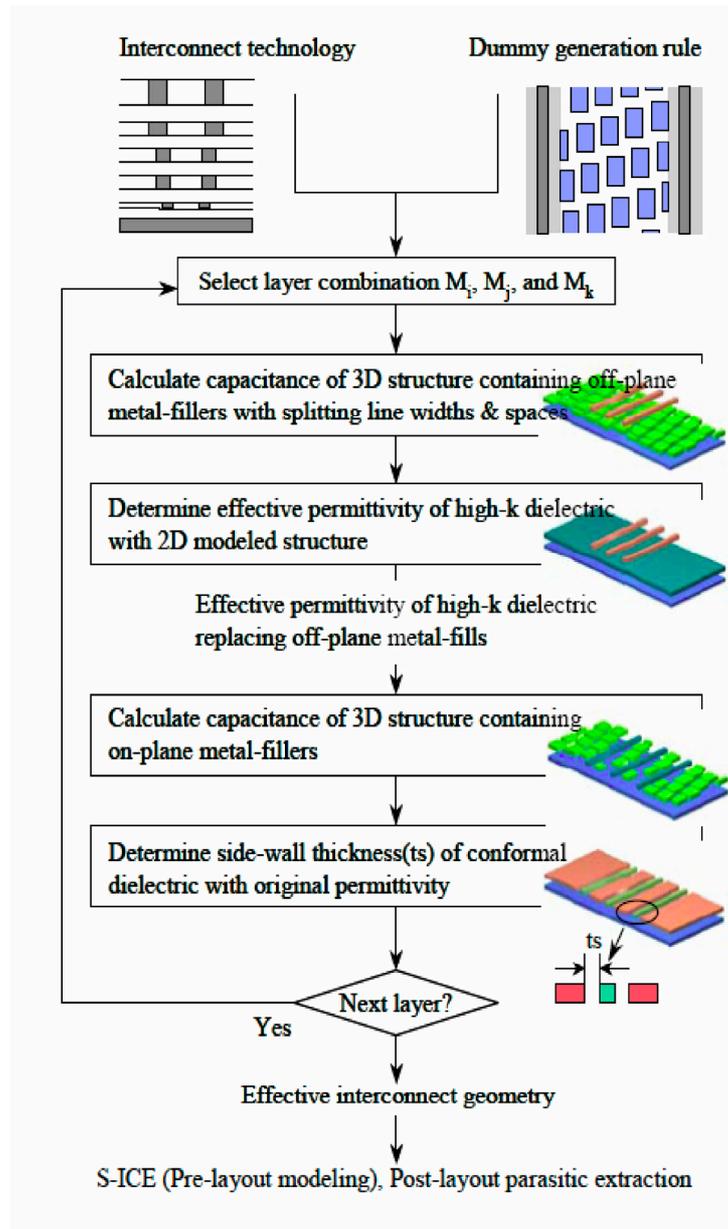


図 ICM-17 配線間／層間の誘電率を変更する手法

4.1.3.3.2 CMP ダミーメタルフィル考慮容量抽出モデル評価

図 ICM-18 に、評価のフロー図を示す。また、図 ICM-19 に、テストデータ仕様を示す。比較的限られた期間での評価となったため、今回は現実的なダミー配置・サイズを考慮したデータバリエーションとした。メタルフィルを議論する上で重要な配線密度については、6%～69%を網羅している。

・ベンチマークフロー

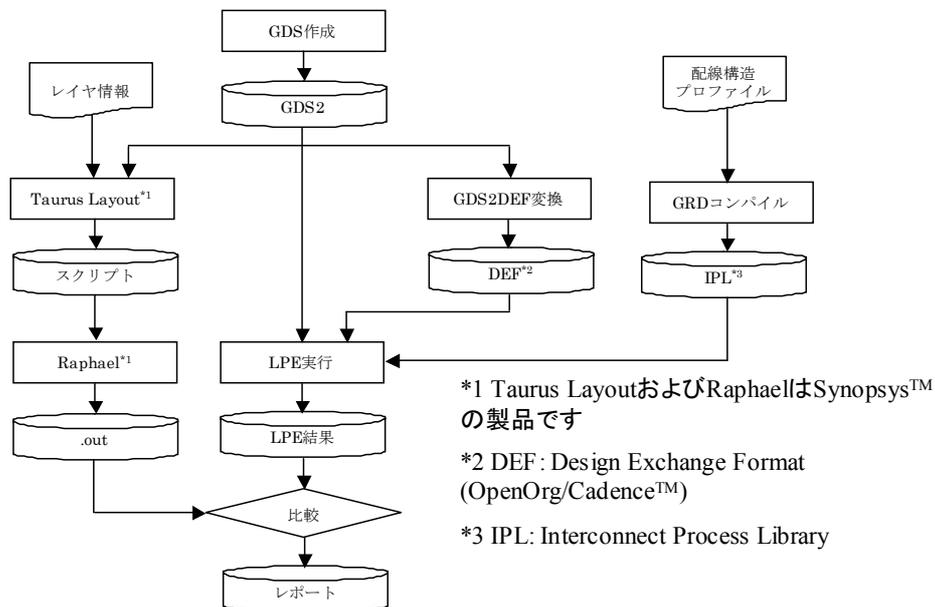


図 ICM-18 ベンチマークフロー

評価結果を図 ICM-22、ICM-23 に示す。ダミーを仮想した容量ライブラリを用いた LPE の精度は、以下のように総括できる。

- ① 格子形状ダミーフィルに対しては高精度（誤差平均 1%未満）
- ② 千鳥格子ダミーフィルに対しては誤差が大きい（今回の構造に対して誤差平均-12%）

また、併せて Raphael で計算した容量値をもとに、ダミーフィルのインパクトを評価した。結果を図 ICM-24 に示す。評価結果より、ダミーフィルを考慮しない場合、容量抽出誤差は最大-23% (@Stitch_3_1)、格子形状でも平均-5%程度の誤差となり、仮想ダミーモデル/実ダミー込み容量抽出が必要であることが判明した。さらに、ダミーを配置したパターン同士の比較から、格子形状より千鳥格子の方が平均 11.7%容量が大きいことが判明。格子形状ダミーフィルを用い、仮想ダミーモデル LPE を行う組合せが精度-TAT-リソースのバランスのよい手法であるといえる。従って、ダミーフィル形状の選定においては、なるべく容量値が小さく、かつ精度のよい仕様を得るために、所望のダミー密度と評価結果のグラフから仕様を決定することを提案する。

• テストデータ仕様

1. 共通仕様

配線層	レイヤ	配線幅[um]	間隔[um]	層間膜	比誘電率
Intermediate	M3-6	0.16	0.16	ILD	3.5
				IMD	2.7
				Passivation	4.0

※ITRS2000Update 100nmプロセス相当

2. リファレンスツールデータ仕様

名称	形状	レイヤ	サイズ[um]	間隔[um]
Array-1	格子オフグリッド	M3/M4/M5	1/3/5	1/2/3
Array-2	格子オングリッド			
Stitch	千鳥格子			

図 ICM-19 テストデータ仕様（プロセス/メタルフィルサイズ）

• テストデータ仕様

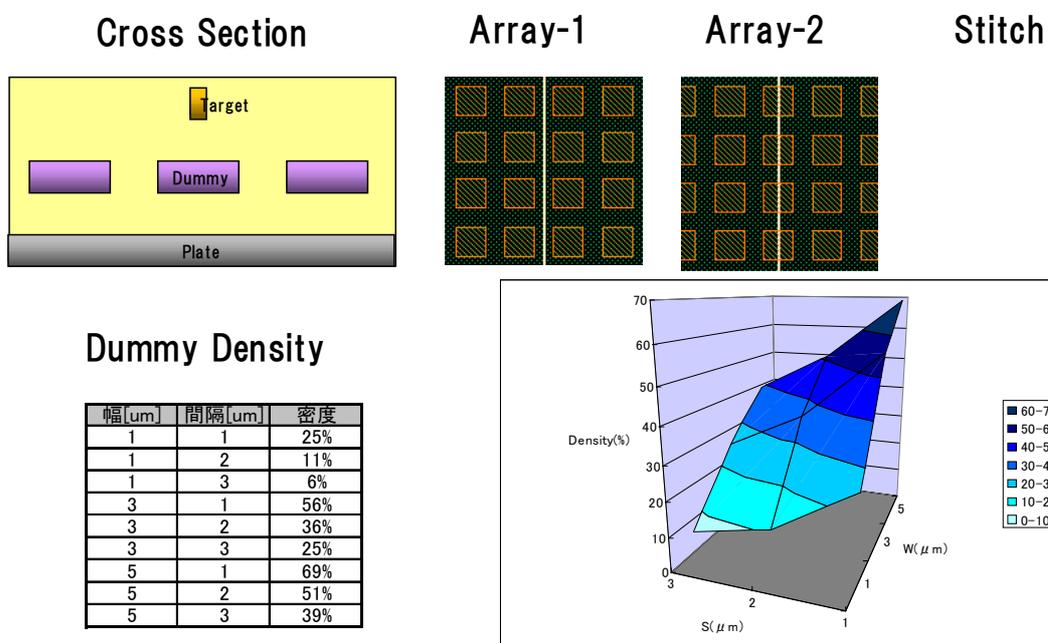
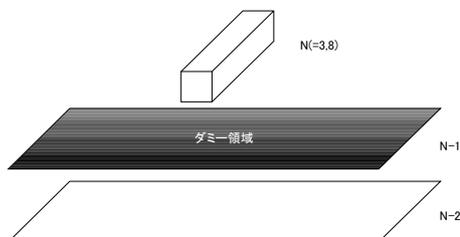


図 ICM-20 テストデータ仕様（メタルフィル諸元）

・ テストデータ仕様

ターゲットツールデータ仕様



- リファレンス用GDSIIデータからターゲット配線およびプレート配線を残しダミーフィルを削除してターゲットデータを作成する
- ダミー領域はLPEライブラリ(IPL)にて仮想ダミーフィル設定を行う
- 仮想ダミーフィル設定はダミーフィルの幅と間隔をパラメータとして定義する

図 ICM-21 テストデータ仕様 (ベンチマーク条件)

・ 評価結果(LPE/Raphael容量比較)

type	size	space	Solver3D	LPE	Error
wo/ dummy			3.26E-15		
array 1 1			3.48E-15	3.35E-15	-3.9%
array 1 2			3.33E-15	3.30E-15	-0.8%
array 1 3			3.31E-15	3.28E-15	-0.8%
array 3 1			3.57E-15	3.47E-15	-2.7%
array 3 2			3.36E-15	3.40E-15	1.2%
array 3 3			3.31E-15	3.35E-15	1.0%
array 5 1			3.64E-15	3.62E-15	-0.6%
array 5 2			3.38E-15	3.46E-15	2.4%
array 5 3			3.31E-15	3.42E-15	3.3%
average(array)			3.41E-15	3.40E-15	-0.1%
array2 1 1			3.36E-15	3.35E-15	-0.5%
array2 1 2			3.31E-15	3.30E-15	-0.4%
array2 1 3			3.30E-15	3.28E-15	-0.8%
array2 3 1			3.42E-15	3.47E-15	1.4%
array2 3 2			3.41E-15	3.40E-15	-0.3%
array2 3 3			3.40E-15	3.35E-15	-1.7%
array2 5 1			3.55E-15	3.62E-15	2.1%
array2 5 2			3.53E-15	3.46E-15	-1.8%
array2 5 3			3.53E-15	3.42E-15	-3.2%
average(array2)			3.42E-15	3.40E-15	-0.6%
stitch 1 1			3.88E-15	3.35E-15	-13.8%
stitch 1 2			3.63E-15	3.30E-15	-9.2%
stitch 1 3			3.57E-15	3.28E-15	-8.2%
stitch 3 1			4.26E-15	3.47E-15	-18.5%
stitch 3 2			3.96E-15	3.40E-15	-14.2%
stitch 3 3			3.66E-15	3.35E-15	-8.5%
stitch 5 1			4.26E-15	3.62E-15	-15.0%
stitch 5 2			4.00E-15	3.46E-15	-13.3%
stitch 5 3			3.84E-15	3.42E-15	-11.0%
average(stitch)			3.90E-15	3.40E-15	-12.4%

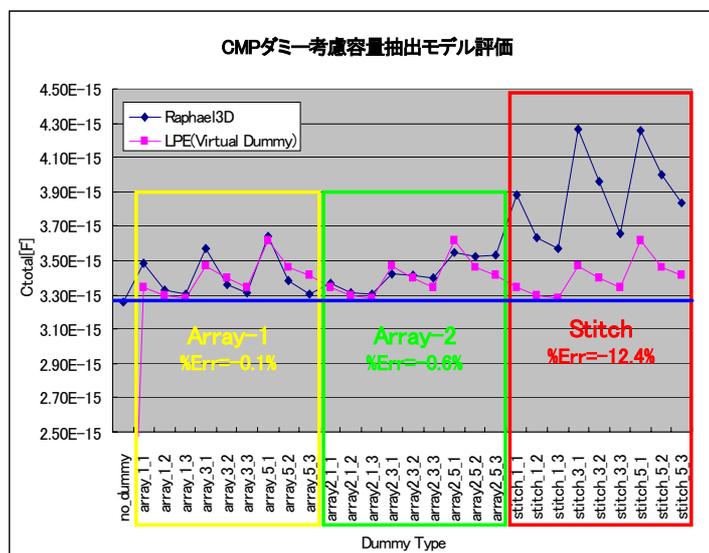


図 ICM-22 ベンチマーク結果 (絶対誤差)

● 評価結果(LPE相対誤差)

type	size	space	Solver3D	LPE	Error
wo/ dummy			3.26E-15	-	-
array 1 1			3.48E-15	3.35E-15	-3.9%
array 1 2			3.33E-15	3.30E-15	-0.8%
array 1 3			3.31E-15	3.28E-15	-0.8%
array 3 1			3.57E-15	3.47E-15	-2.7%
array 3 2			3.36E-15	3.40E-15	1.2%
array 3 3			3.31E-15	3.35E-15	1.0%
array 5 1			3.64E-15	3.62E-15	-0.6%
array 5 2			3.38E-15	3.46E-15	2.4%
array 5 3			3.31E-15	3.42E-15	3.3%
average(array)			3.41E-15	3.40E-15	-0.1%
array2 1 1			3.36E-15	3.35E-15	-0.5%
array2 1 2			3.31E-15	3.30E-15	-0.4%
array2 1 3			3.30E-15	3.28E-15	-0.8%
array2 3 1			3.42E-15	3.47E-15	1.4%
array2 3 2			3.41E-15	3.40E-15	-0.3%
array2 3 3			3.40E-15	3.35E-15	-1.7%
array2 5 1			3.55E-15	3.62E-15	2.1%
array2 5 2			3.53E-15	3.46E-15	-1.8%
array2 5 3			3.53E-15	3.42E-15	-3.2%
average(array2)			3.42E-15	3.40E-15	-0.6%
stitch 1 1			3.88E-15	3.35E-15	-13.8%
stitch 1 2			3.63E-15	3.30E-15	-9.2%
stitch 1 3			3.57E-15	3.28E-15	-8.2%
stitch 3 1			4.26E-15	3.47E-15	-18.5%
stitch 3 2			3.96E-15	3.40E-15	-14.2%
stitch 3 3			3.66E-15	3.35E-15	-8.5%
stitch 5 1			4.26E-15	3.62E-15	-15.0%
stitch 5 2			4.00E-15	3.46E-15	-13.3%
stitch 5 3			3.84E-15	3.42E-15	-11.0%
average(stitch)			3.90E-15	3.40E-15	-12.4%

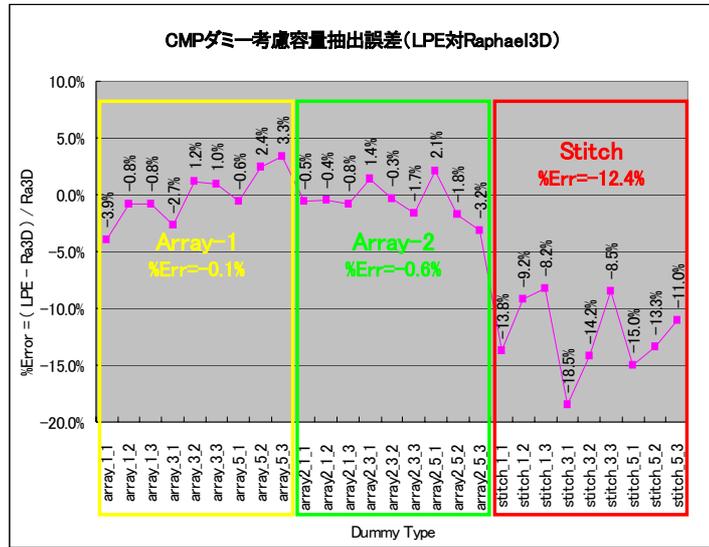


図 ICM-23 ベンチマーク結果 (相対誤差)

● 評価結果(ダミーフィル容量インパクト)

type	size	space	w/ dummy	wo/ dummy	Error
array 1 1			3.48E-15	3.26E-15	-6.4%
array 1 2			3.33E-15	3.26E-15	-2.1%
array 1 3			3.31E-15	3.26E-15	-1.4%
array 3 1			3.57E-15	3.26E-15	-8.6%
array 3 2			3.36E-15	3.26E-15	-3.0%
array 3 3			3.31E-15	3.26E-15	-1.6%
array 5 1			3.64E-15	3.26E-15	-10.5%
array 5 2			3.38E-15	3.26E-15	-3.7%
array 5 3			3.31E-15	3.26E-15	-1.4%
average(array)			3.41E-15	3.26E-15	-4.3%
array2 1 1			3.36E-15	3.26E-15	-3.1%
array2 1 2			3.31E-15	3.26E-15	-1.7%
array2 1 3			3.30E-15	3.26E-15	-1.4%
array2 3 1			3.42E-15	3.26E-15	-4.8%
array2 3 2			3.41E-15	3.26E-15	-4.5%
array2 3 3			3.40E-15	3.26E-15	-4.2%
array2 5 1			3.55E-15	3.26E-15	-8.1%
array2 5 2			3.53E-15	3.26E-15	-7.6%
array2 5 3			3.53E-15	3.26E-15	-7.6%
average(array2)			3.42E-15	3.26E-15	-4.8%
stitch 1 1			3.88E-15	3.26E-15	-16.0%
stitch 1 2			3.63E-15	3.26E-15	-10.3%
stitch 1 3			3.57E-15	3.26E-15	-8.7%
stitch 3 1			4.26E-15	3.26E-15	-23.5%
stitch 3 2			3.96E-15	3.26E-15	-17.8%
stitch 3 3			3.66E-15	3.26E-15	-10.8%
stitch 5 1			4.26E-15	3.26E-15	-23.4%
stitch 5 2			4.00E-15	3.26E-15	-18.5%
stitch 5 3			3.84E-15	3.26E-15	-15.1%
average(stitch)			3.90E-15	3.26E-15	-16.0%

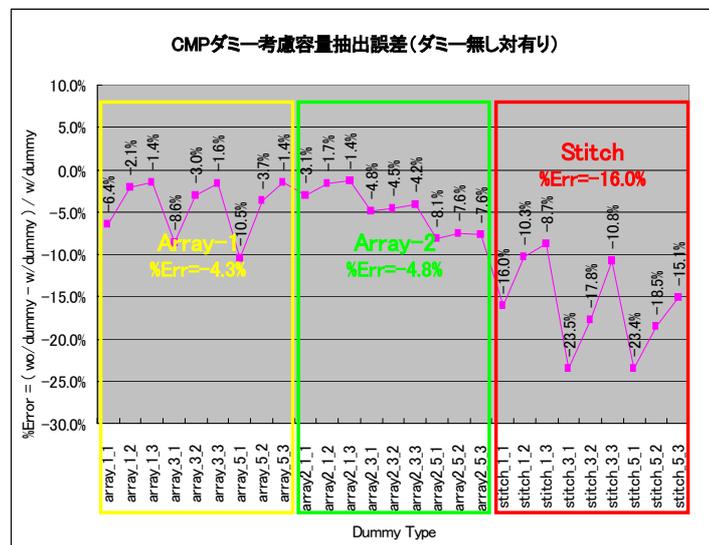


図 ICM-24 ダミーフィル有無の容量比較

4.1.3.4 Cu 配線断面の配線パターン依存性調査

4.1.3.4.1 Cu 配線断面

Cu 配線プロセスの特徴は、絶縁膜上に配線溝を形成し、酸化防止等を目的としたバリアメタル層形成後、電解メッキ等により配線となる Cu を埋め込み・堆積、その後研磨材（スラリ）と研磨パッドを用いた CMP（Chemical Mechanical Polishing）平坦化処理により平坦な配線層を形成するダマシン（象嵌）プロセスである（図 ICM-25）。

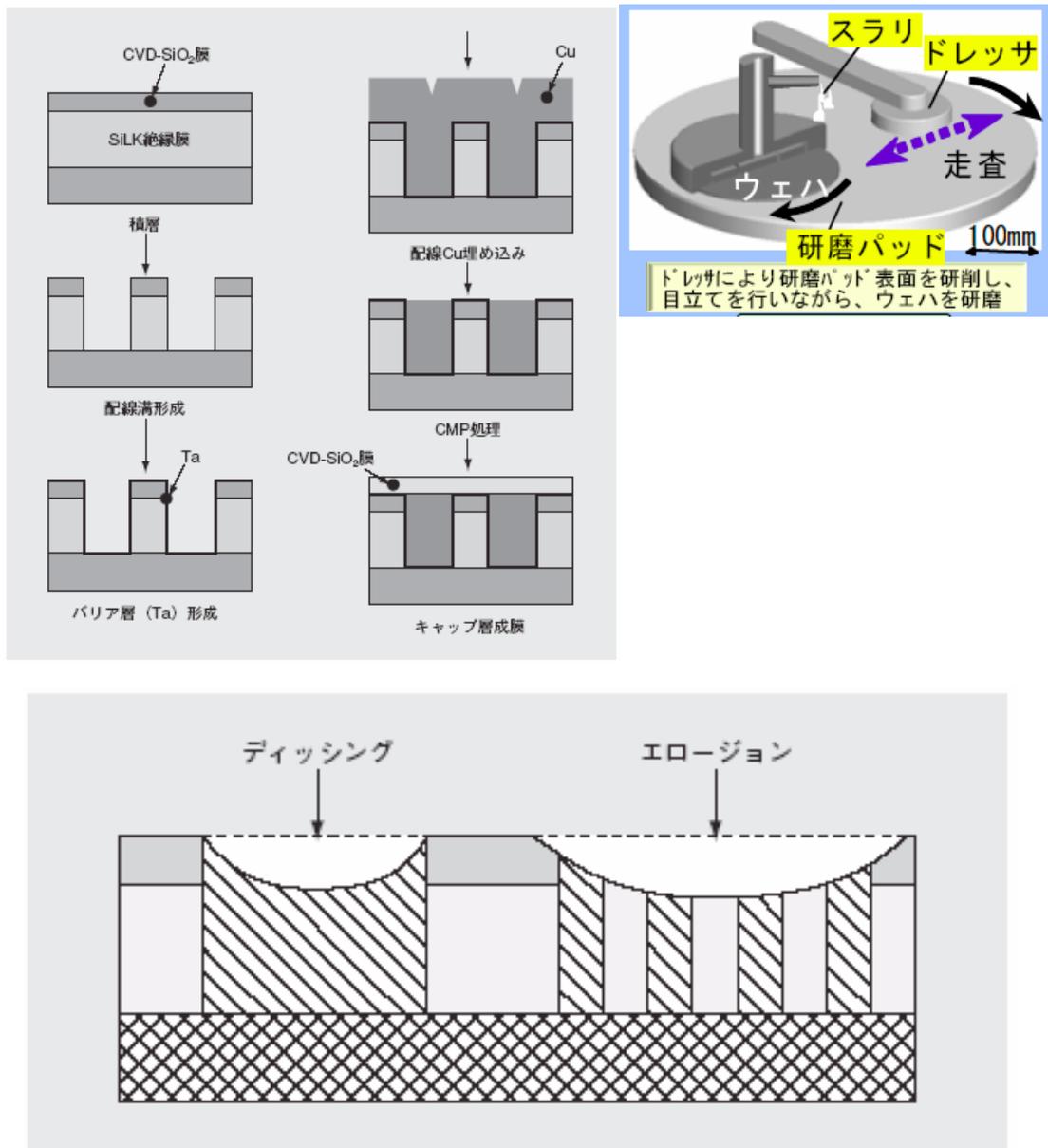


図 ICM-25 Cu ダマシンプロセス

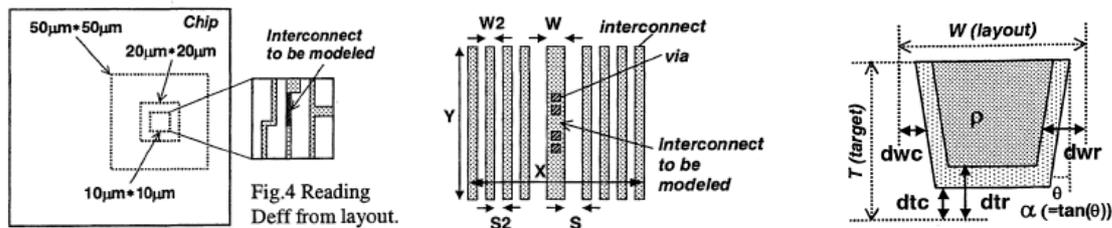
CMP 平坦化を用いるダマシンプロセスであるが、実際にはスラリの Cu、絶縁膜それぞれに対する研磨割合の比（選択比）の違い、および研磨パッドの撓みなどの要因により、ディッシング/エロージョンなど、段差が残る現象を伴う（図 ICM-25）。また、メッキによる Cu の堆積時に初期段差を生じ、これが CMP 工程で解消されない場合も段差残の要因となる。

4.1.3.4.2 シミュレーションモデル

前述した段差残の影響は、主に配線の抵抗率に影響する。そのため、LPE 時に当該現象をモデル化して考慮する手法が提案されているが、それらのシミュレーションモデルは以下に示す 2 種類に大別できる。

(1) 回帰式モデル

図 ICM-26 に示すように、メッキ段差、エロージョン/ディッシング量を配線幅、配線実効密度の回帰式としてモデル化、TEG 測長によりキャリブレーションする手法が提案されている [ICM-17]。



Model Parameters	Unit	Variable	Meaning
dwr	µm	W, (S)	ΔW for resistance
dwc	µm	W, (S)	ΔW for capacitance
dtr	µm	W, S, Deff	ΔT for resistance
dtc	µm	W, S, Deff	ΔT for capacitance
α	-	W	slope of edge
ρ	mΩ·µm	Constant or W	resistivity

dwc:メッキ段差、エロージョン、ディッシング
dtr-dtc, dwr-dwc:バリアメタル厚、
θ:テーパ角

Parameters	Unit	Meaning
W	µm	Width of target interconnect
S	µm	Spacing of target interconnect
W2	µm	Width of neighboring interconnect
S2	µm	Spacing of neighboring interconnect
X	µm	Pattern size
Y		

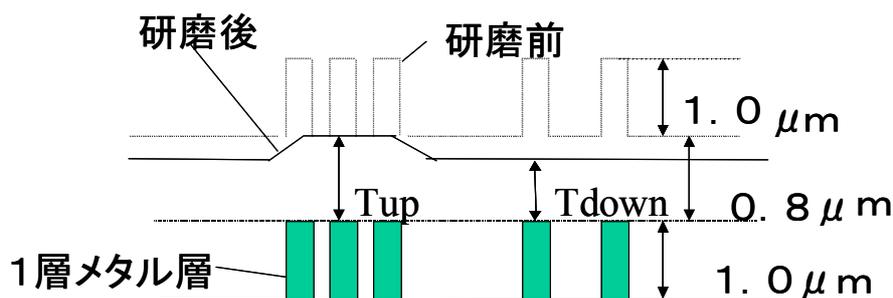
図 ICM-26 回帰式モデル

この手法については、LPE ツールが当該モデルをもち、実効配線密度の計算機能を保有していれば従来のフローを変えることがないのが利点である。一方、配線幅、実効配線密度からメッキ段差、エロージョン/ディッシング量を予測する回帰式を TEG の断面/平面測長値

から求める手法は標準的ではない。さらに、種々の大きさのウインドウ内における配線密度を 1 値の実効密度に縮退することに起因するモデル化誤差が原理的に存在する。そこで、今回はこの点に主眼をおき、後述するプロセスシミュレーションモデルとの比較を行った。

(2) プロセスシミュレーションモデル

研磨パッドの応力応答を FFT を用いて解くプロセスシミュレーション手法が提案されている (図 ICM-27、文献[ICM-18]、[ICM-19]、[ICM-20])。併せて、当該手法には、初期メッキ工程のシミュレーションが含まれている (図 ICM-28)。この手法はより物理的なアプローチであるため、TEG を用いたキャリブレーションに対する親和性が高い。一方、現時点では LPE ツールの外部で平面座標に対する段差のマトリックスを求めるため、当該マトリックスを入力する機能をもった LPE ツールが必要となるほか、配線層ごとにマトリックスを生成しておく必要がある。従って、ここでは回帰式モデルに対するリファレンスとして本モデルを位置づける。



パッドの変形: 応力応答
 応力集中
 研磨速度と応力: プレストンの式

$$\Delta h_i = A \cdot \Delta t [E / \{ h_i - \text{FFT}^{-1} (\text{FFT}(h) \cdot \text{FFT}(f)) \} + P_0]^n / \rho_i^n$$

CMP-Designer-Cu版の特徴

1. 独自のCOGモデルによるCu-CMPの高精度計算
2. 各種モデルによる広範なプロセス対応
 - ・スラリー材料、メッキ条件に対応可能
 - ・各種メッキモデル・非プレストンモデル
 - ・ローカルモデル
3. キャリブレ-ション方法を確立
4. 実測結果との良い一致
 - ・特に1、2ステップのErosion, Dishingがいずれも実測と良く一致する。
5. 高速計算
 - ・FFT、境界処理
6. 操作性向上機能
 - ・半自動ダミー生成、GUI、キャリブレ-ション

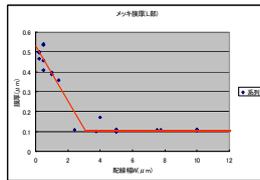
図 ICM-27 CMP プロセスシミュレーションモデル

Cuメッキモデル

1. 実測結果を読み込む
2. 密度と周囲長から計算
3. **配線幅モデル**
4. L&Sモデル:重回帰解析を行う。

配線幅メッキモデル対応:新密度計算モジュール

配線パターン → **ポリゴン**の幅Wを抽出 → 配線幅Wを分類 → 各配線幅での密度計算



新密度ファイル

X座標 Y座標 dens1 dens2 ...

$W < W1$

$W1 < W < W2$

メッキ後の膜厚(Overfill)の実測結果 → 膜厚のW依存性
→ 新密度ファイルを用いてメッキ膜厚計算

図 ICM-28 Cuメッキプロセスシミュレーションモデル

4.1.3.4.3 回帰式-プロセスシミュレーションモデル比較

① CMP工程の周囲パターンの影響評価 (1): 50 μm外のパターンの影響

50 μmの領域の中心にL&Sの配線パターンを形成し、その部分のDishingを計算した。ここで、図ICM-29に示すパターンを用いて50 μmの領域の外に密度83%のダミーパターンを入れた場合と入れない場合のDishingを両手法で評価した。100nm相当のプロセスのCu配線を想定しシミュレーションした結果を図ICM-30に示す。この図は中央部のL&Sパターンの配線幅とDishing量の関係を示している。シミュレーション結果は周辺ダミーパターン有り無しで膜厚0.07 μm(約9%)のDishing量の差が生じ、この値はパターン幅にはあまり関係ないことを示している。回帰式モデルを用いた場合には周辺ダミーパターンの有り無しで全く差が生じない。従って両手法による誤差は約9%である。今後、本結果に対する検討、発展が課題である。

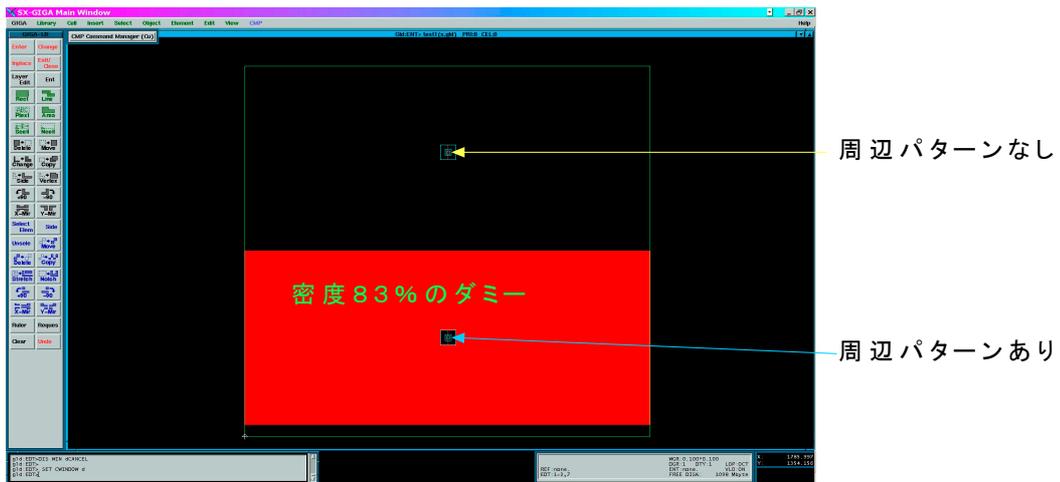


図 ICM-29 比較用のパターンアレイ

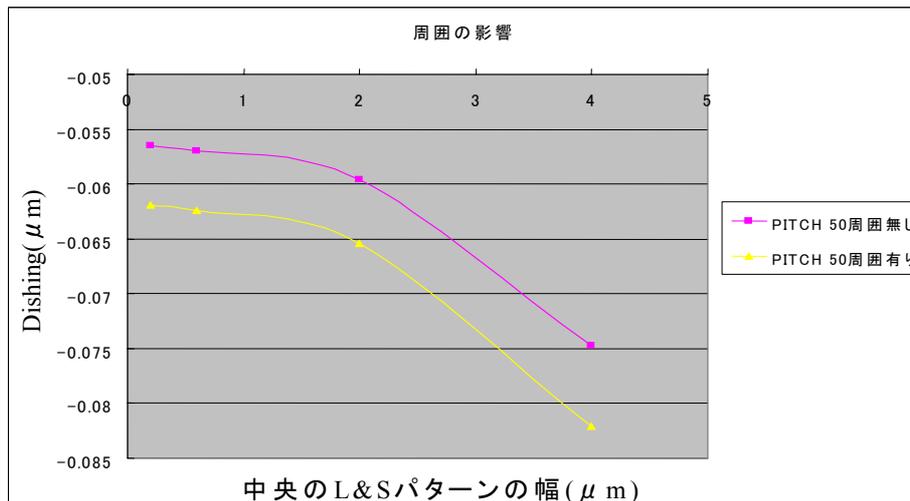


図 ICM-30 シミュレーションによる周辺ダミーパターンの影響

- ② CMP 工程の周囲パターンの影響 (2) : 次に周囲パターンの影響に関する別の事例を示す。図 ICM-31 に、右半分が密度 0%、左半分が密度約 83% (配線アレイで L&S は $0.5 \mu\text{m}/0.1 \mu\text{m}$) のパターンを用いて 100nm 相当のプロセスの Cu 配線を想定した Dishing 分布のシミュレーション結果である。このような曲線はエロージョン曲線と呼ばれ、パターン内の端部(右端から $25 \mu\text{m}$)と中心部での Dishing が 35nm (約 4%) 異なっている。回帰式モデルでは、端部と中心部の差は生じない。①の場合より誤差は小さいが、頻度が高く生じる重要な誤差である。今後、本結果に対する検討、発展が課題である。

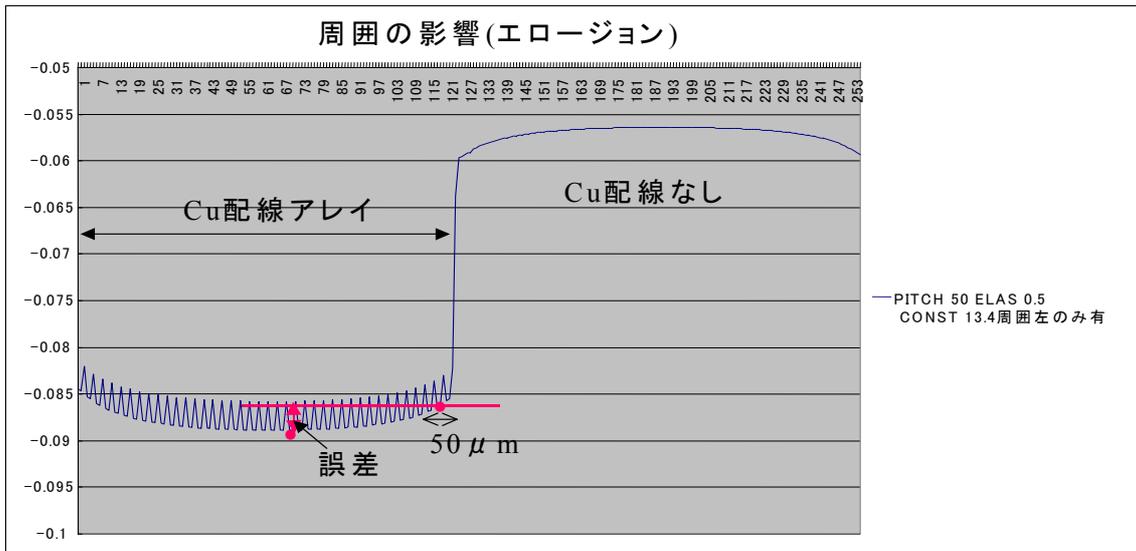


図 ICM-31 シミュレーションによる大きなアレイでの Dishing 分布

③ CMP 工程のウインドウ内でのパターンの影響

50 μm ウインドウ内でのパターンの位置による誤差を図 ICM-32 に示す 2 つのパターンを用い、中央部の Dishing をシミュレーションすることにより求めた。シミュレーション結果、2 つのパターンでの Dishing の差は約 6nm (1%) であった。図 ICM-32 のパターンは 50 μm のウインドウ内でパターンを移動させただけであり、回帰式モデルでは各ウインドウの密度が同じなので Dishing は同じ値である。よって両者の誤差は約 1% であった。今後、本結果に対する検討、発展が課題である。

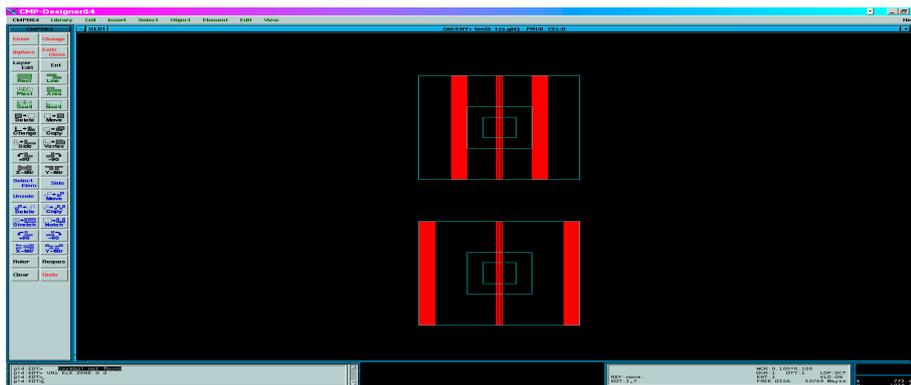


図 ICM-32 ウインドウ内バラツキ評価パターン

④ メッキ工程：メッキ膜厚はパターン依存性があり、この影響を両者で評価した。なお、シミュレーションでの研磨時間は、実際の終点検出器を用いたプロセスと同様に全ての Cu 膜厚がバリアメタル以下になる時間 + α (20%) として行った。図 ICM-33 には、周辺に 0.2 μm

の配線を形成したパターンと周辺配線なしのパターン（いずれも中央の $50\mu\text{m}$ ウィンドウには $2\mu\text{m}$ 配線をしきつめ、①で見られた周辺の影響を除去している）での初期膜厚を、前出（図 ICM-28）のメッキモデルを用いたシミュレーション結果を示す。周囲の $0.2\mu\text{m}$ の配線によりメッキ膜厚は約 $0.3\mu\text{m}$ と周辺配線無しの膜厚より大きくなる。従ってシミュレーションでは周囲の Cu を研磨するために 8 秒間研磨時間を長くしている。その結果 CMP 後の Dishing は図 ICM-34 に示すように、周辺パターン有りの方が約 4% 増加している。回帰式モデルでは $50\mu\text{m}$ ウィンドウ内のパターンが同じであるため差が生じない。したがってここでの両者の差は 4% である。なおメッキ残りは、最新のプロセス、材料を用いた場合には更に大きくなることが知られているが今回は考慮していない。今後、本結果に対する検討、発展が課題である。

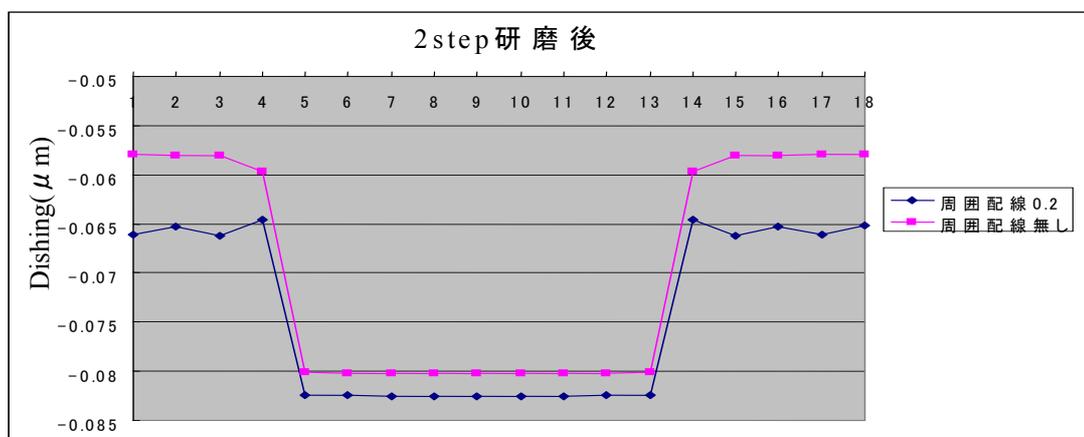


図 ICM-33 シミュレーションによるメッキ後の初期膜厚分布

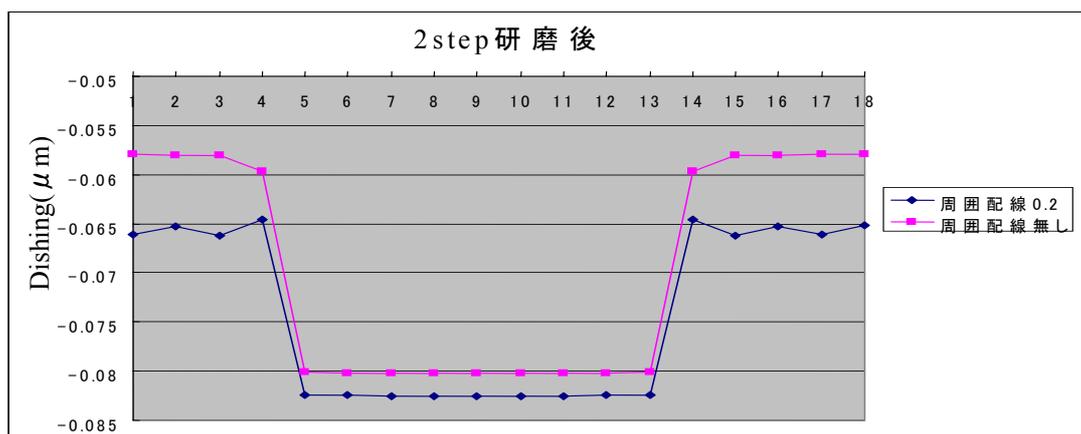


図 ICM-34 シミュレーションによる Dishing 分布

4.1.3.5 まとめ

今回、ICM-TG では以下の 3 項目について評価を行った。

- (1) 配線容量の 3D 効果に対する精度評価、および誤差要因分析。
- (2) Cu 配線プロセスに対する容量モデリング手法評価
- (3) Cu 配線プロセスに対する抵抗（配線段差）モデリング手法評価

今後は、(2)、(3) についての継続的な発展のほか、配線形状ばらつきの遅延への影響評価や、標準ベンチマークデータの策定が課題として考えられる。

以上

4.1.3.6 参考文献

- [ICM-1] D. O. Ouma, D. S. Boning, J. E. Chung, W. G. Easter, V. Saxena, S. Misra, and A. Crevasse, "Characterization and modeling of oxide chemical-mechanical polishing using planarization length and pattern density concepts," *IEEE Trans. Semiconduct. Manufact.*, vol. 15, no. 2, pp. 232-244, May 2002.
- [ICM-2] C. C. Jeng, W. K. Wan, H. H. Lin, M.-S. Liang, K. H. Tang, I. C. Kao, H. C. Lo, K. S. Chi, T. C. Huang, C. H. Yao, C. C. Lin, M. D. Lei, C. C. Hsia, M.-S. Liang, "BEOL process integration of 65nm Cu/low k interconnects," in *Proc. IITC*, pp. 199-201, June 2004.
- [ICM-3] S. Lakshminarayanan, P. J. Wright, and J. Pallinti, "Electrical characterization of the copper CMP process and derivation of metal layout rules," *IEEE Trans. Semiconduct. Manufact.*, vol. 16, no. 4, pp. 668-676, Nov. 2003.
- [ICM-4] P. Zarkesh-Ha, S. Lakshminarayann, K. Doniger, W. Loh, and P. Wright, "Impact of interconnect pattern density information on a 90nm technology ASIC design flow," in *Proc. ISQED*, pp. 405-409, Mar. 2003.
- [ICM-5] W.-S. Lee, K.-H. Lee, J.-K. Park, T.-K. Kim, Y.-K. Park, and J.-T. Kong, "Investigation of the capacitance deviation due to metal-fills and the effective interconnect geometry modeling," in *Proc. ISQED*, pp. 373-376, Mar. 2003.
- [ICM-6] J.-K. Park, K.-H. Lee, J.-H. Lee, Y.-K. Park, and J.-T. Kong, "An exhaustive method for characterizing the interconnect capacitance considering the floating dummy-fills by employing an efficient field solving algorithm," in *Proc. SISPAD*, pp. 98-101, Sep. 2000.
- [ICM-7] K.-H. Lee, J.-K. Park, Y.-N. Yoon, D.-H. Jung, J.-P. Shin, Y.-K. Park, and J.-T. Kong, "Analyzing the effects of floating dummy-fills: from feature scale analysis to full-chip RC extraction," in *Proc. IEDM*, pp. 685-688, Dec. 2001.
- [ICM-8] A. Kurokawa, T. Kanamoto, A. Kasebe, Y. Inoue, and H. Masuda, "Efficient capacitance extraction method for interconnects with dummy fills," in *Proc. CICC*, pp. 485-488, Oct. 2004.
- [ICM-9] B. E. Stine, D. S. Boning, J. E. Chung, L. Camilletti, F. Kruppa, E. R. Equi, W. Loh, S.

-
- Prasad, M. Muthukrishnan, D. Towery, M. Berman, and A. Kapoor, "The physical and electrical effects of metal-fill patterning practices for oxide chemical-mechanical polishing processes," *IEEE Trans. Electron Dev.*, vol. 45, no. 3, pp. 665-679, Mar. 1998.
- [ICM-10] A. B. Kahng, G. Robins, A. Singh, H. Wang, and A. Zelikovsky, "Filling algorithms and analyses for layout density control," *IEEE Trans. Computer-Aided Des. Integrated Circuits & Syst.*, vol. 18, no. 4, pp. 445-462, Apr. 1999.
- [ICM-11] Y. Chen, A. B. Kahng, G. Robins, and A. Zelikovsky, "Area fill synthesis for uniform layout density," *IEEE Trans. Computer-Aided Des. Integrated Circuits & Syst.*, vol. 21, no. 10, Oct. 2002.
- [ICM-12] Y. Chen, P. Gupta, and A. B. Kahng, "Performance-impact limited area fill synthesis," in *Proc. DAC*, pp. 22-27, June 2003.
- [ICM-13] A. Kurokawa, T. Kanamoto, T. Ibe, A. Kasebe, C. W. Fong, T. Kage, Y. Inoue, and H. Masuda, "Dummy filling methods for reducing interconnect capacitance and number of fills," in *Proc. ISQED*, Mar. 2005.
- [ICM-14] O. Cueto, F. Charlet, and A. Farcy, "An efficient algorithm for 3D interconnect capacitance extraction considering floating conductors," in *Proc. SISPAD*, pp. 107-110, Sep. 2002.
- [ICM-15] Keun-Ho Lee, Jin-Kyu Park, Young-Nam Yoon, Dai-Hyun Jung, Jai-Pil Shin, Young-Kwan Park, and Jeong-Taek Kong, "Analyzing the effects of floating dummy-fills: from feature scale analysis to full-chip RC extraction", in *Proc. IEDM*, pp. 31.3.1 - 31.3.4, Dec. 2001.
- [ICM-16] Won-Seok Lee, Keun-Ho Lee, Jin-Kyu Park, Tae-Kyung Kim, Young-Kwan Park, and Jeong-Taek Kong, "Investigation of the capacitance deviation due to metal-fills and the effective interconnect geometry modeling", in *Proc. ISQED*, pp. 373 ? 376, March 2003.
- [ICM-17] K. Yamada, N. Okada, M. Yasuda, N. Oda, "Accurate modeling method for deep sub-micron Cu interconnect", , 2003. *Digest of Technical Papers. Symposium on VLSI Technology*, pp. 111 ? 112, June 2003
- [ICM-18] T. Ohta,.,T. Toda, and H.Ueno" A practical CMP profile model for LSI design application", " in *Proc. SISPAD*, pp. 195 ? 198, Sept. 1999
- [ICM-19] T. Ohta, M. Fujinaga, M.Kimura, T. Wada, and K. Nishi, " A simulation system for capacitance variation by CMP process including defocus effect", in *Proc. SISPAD*,pp. 102 ? 105, Sept. 2000.
- [ICM-20] T. Ohta, and K. Suzuki," A new SP (simultaneous polishing) model for copper CMP process", in *Proc. SISPAD*,pp. 257 ? 260, Sept. 2002
-

4.1.4 2004 年度成果報告会のアンケート結果

報告会開催日時：2004 年 3 月 24 日 13:30-15:10 JEITA605 会議室

参加者：24 名（うち 13 名は PDM 研究会委員）

アンケートの目的

1. 活動成果に対する評価をして頂く
2. 今後検討すべきテーマ、もしくはそのヒントを頂く

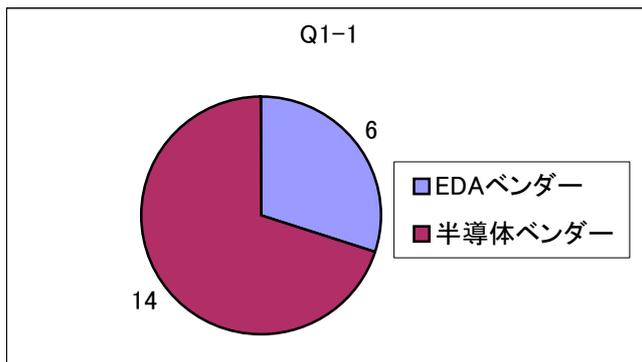
集計結果要約

- ・ おおむね好評でアンケートでの評価も良い。
- ・ 実用的、業務に役立つテーマが好評。ルールへの提示などの要求も強い。
- ・ 自社でカバーできないが気になる分野についても好評。
- ・ 集計には記載していないが、委員として参加していない会社からの参加が目立った。

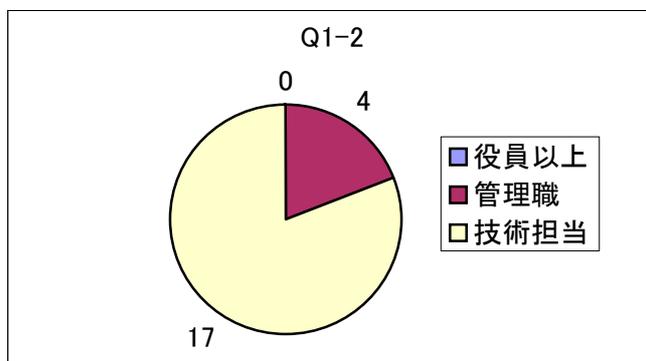
[回答者の業務内容・興味対象]

1. 貴職の業種と役職をお選びください。

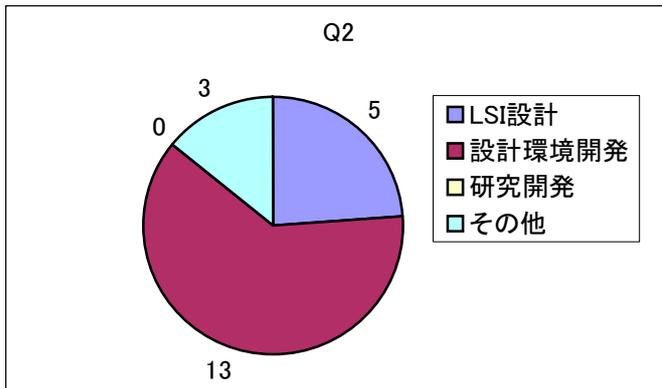
1-1 業種：



1-2 役職：



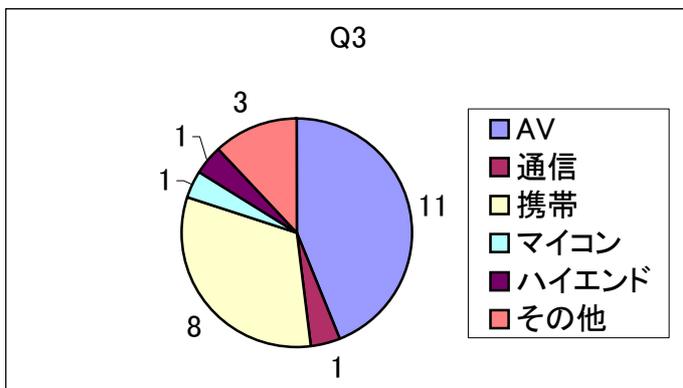
2. ご担当の業務内容を下記からお選びください。



その他：営業技術 1、EMI、EMC 対策 1

3. ご担当の業務で扱うチップの種類のうち主なものをお選びください。

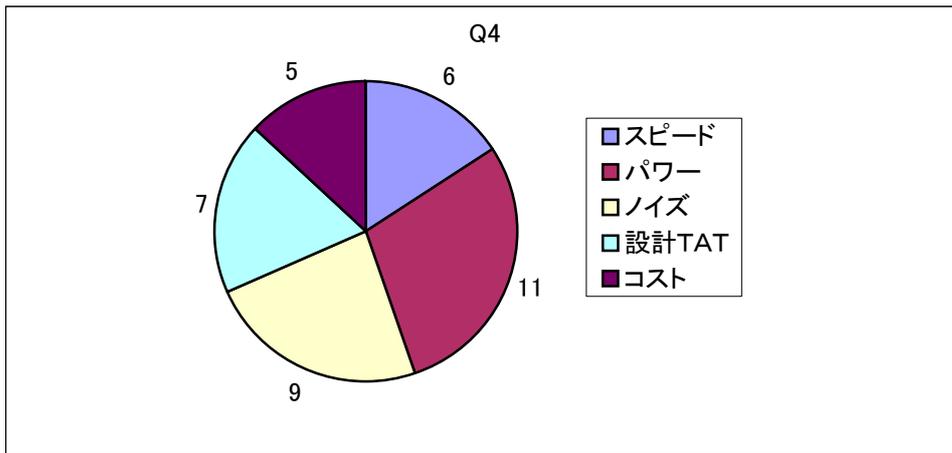
- AV、マルチメディア系
- 広帯域、インターネット系の通信用
- 携帯電話、無線通信系
- マイコン組み込み
- ハイエンド・プロセッサ、チップセット



その他（ASIC1、無記入 2）

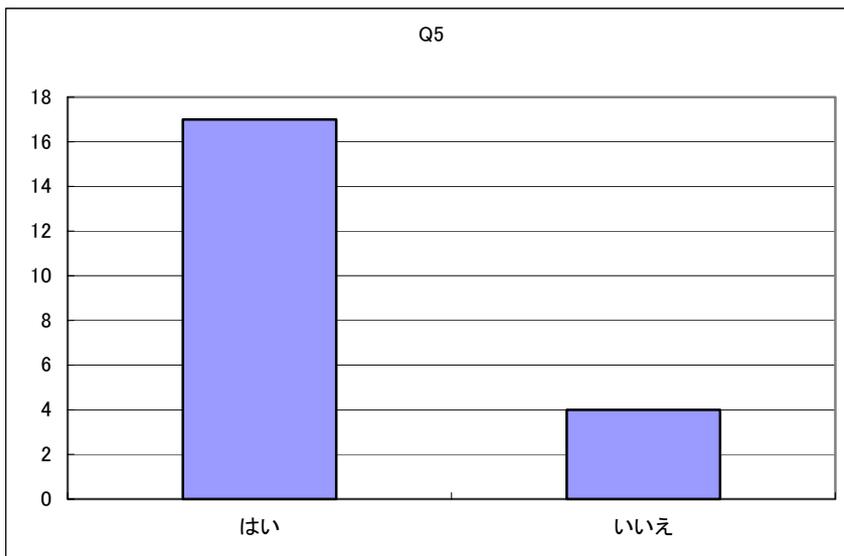
4. 2～3年後の設計課題として、最も重要だと思うものを2つ選択してください。

- スピード（高速動作保証、遅延計算精度・見積もり精度）
- パワー（低消費電力化、EM）
- ノイズ（EMI/EMC 対策、ホットキャリア）
- 設計 TAT
- コスト（チップサイズ、チップ単価）



[成果についての質問]

5. JEITA EDA 技術専門委員会に属する PDM (Physical Design Methodology) 研究会の存在をご存知ですか？

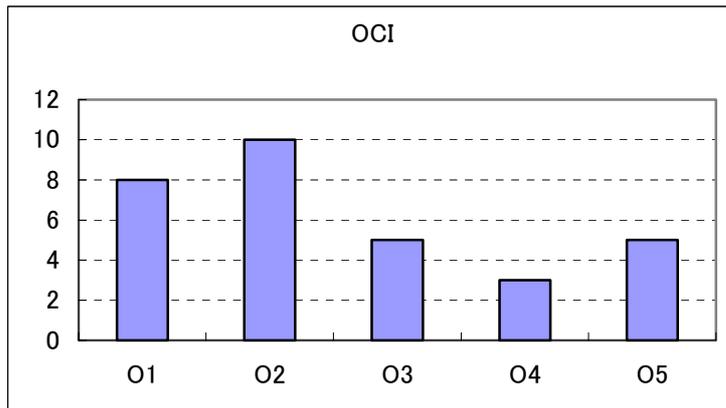


6. PDM 研究会の成果のうち、貴部門の業務で役立つものがありますか？ また、その理由を教えてください。下記の3つのタスクグループ毎に、役に立つ成果の項目番号とその理由、役に立たない成果の項目番号とその理由、をお答えください。また、今後検討して欲しいテーマや課題を教えてください。

オンチップサーマル・タスクグループ成果

役に立つ項目

- O1. 熱シミュレーションのモデル化・解析手法
- O2. 温度分布・チップ特性のフロアプラン依存性解析結果
- O3. フロアプランにおける温度分布の平坦化手法
- O4. リーク電流を考慮した熱シミュレーション手法
- O5. リーク電流が温度分布に与える影響の解析結果例



役に立つ理由

○LSI設計会社にいますが、熱に関する検討は具体的な手法についてお手上げだったので、手法が提示されたのは役に立つ。○体系的に理解できた。○知見として参考になる。○熱の影響に対する定量的なイメージ。○熱を気にする必要があるか否かが分かった。○フロアプランの参考になる。○フロアプランによる依存性が分かった。ただしシミュレーションばかりでなく、チップを作成して対策前、対策後の評価もして頂きたい。温度上昇はEM以外にも信頼性の低下を招くため、今後とも続けてください。今の高密度・高集積であるRAMが多く入った chip で（製品レベル）上記の実験評価をしていただきたい。本当に局所的に温度が上昇するのか評価してほしい。○今後リークが大きくなる。フロアプランで変化するというのは面白い。○具体的に方法を示して設計の参考になりそうだったので。○SPICE で解く方法はおもしろいと思いました。○熱の問題は今後重要となる。○実用的に見える。

役に立たない項目（全部つけた人1、O2=1人、O3=1人）

役に立たない理由

32W もの大規模、超消費電力のチップを扱っていないため。

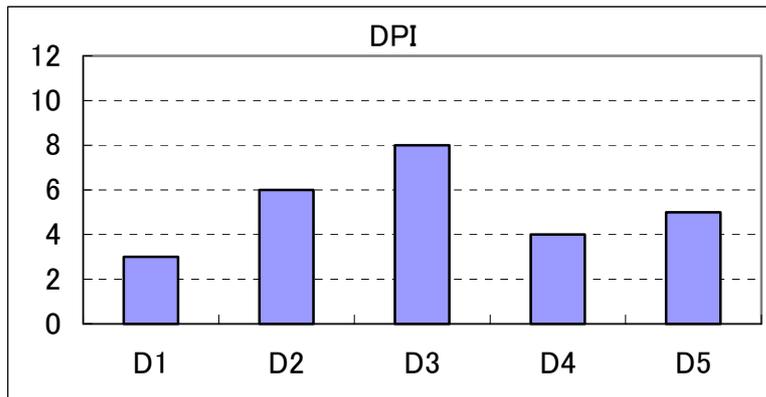
今後検討して欲しいテーマ・課題

○具体的には思いつかないが、設計への FeedBack は弱い気がする。○ ΔT と～何かのパラメータで表されるガイドライン？○実 LSI での検討をお願いします。○温度を検出して自律する機構。○外部環境の影響（風量など）。○実 LSI との相関関係手法。

電源ノイズ・タスクグループ成果

役に立つ項目

- D1. SSO ノイズ解析用モデル
- D2. 上記を用いた感度解析結果
- D3. 制約変換による SSO 設計最適化手法
- D4. SSO ノイズ制約からバッファ駆動能力と実効インダクタンス制約への変換方法
- D5. 遅延変動制約からバッファ駆動能力と実効インダクタンス制約への変換方法



役に立つ理由

○今後、どこに注意していくべきか参考になった。○知見として参考になる。○定量的イメージ。最適化手法。○社内基準値との比較リファレンスとして役に立ちそう。○現存するツールでカバーできない手法の開発。○電源本数が効果ありと判明したと思うから。SSO ノイズモデルの作成で SSO 数から I/O 駆動の順で支配的であることが分かった。○SSO ルールの策定が難しい。○概要しか分からなかったが、今後考えていかないといけないと思えた。○制約の統一化に役立つと思いました。○明確なガイドラインが必要なので。実際に使えそう。

役に立たない項目 (D1=1 人、D2=1 人、D3=2 人)

役に立たない理由

○実状が分かっていないのでノーコメント。○実用(実現)性を加味して評価、順位付けして欲しい。○実測でも評価してコリレーションをとって欲しい。○新しい知見がない。

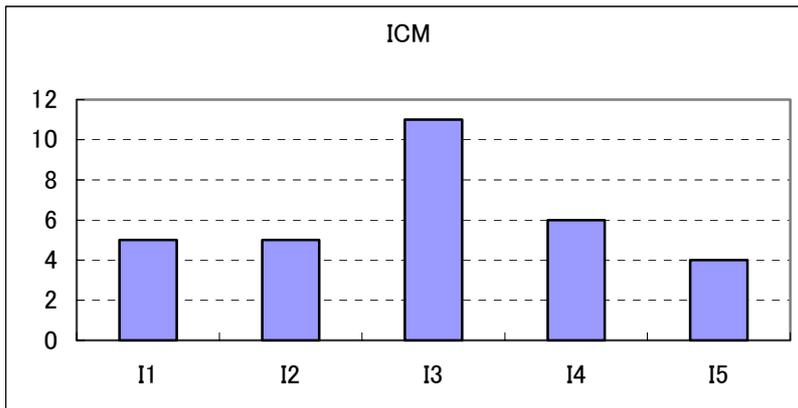
今後検討して欲しいテーマ・課題

○電源ノイズという原因に対して、デジ・アナ干渉、EMI、Drop 遅延を全体最適していくガイドライン。○オン-オフ協調シミュレーションモデル。○I/O と内部を合わせた合成 Tool の作成。○実 LSI へアプライした結果を示して頂けると参考になる。○電源ノイズからの遅延変動。○substrate ノイズ。○SSO ルールをレベルアップして実際に使えるレベルにして欲しい。○D3 の具体化、実効インダクタンスの求め方。

インターコネクトモデリング・タスクグループ成果

役に立つ項目

- I1. オンチップ配線容量の 3D 効果に対する精度評価方法
- I2. 上記を用いた精度評価結果
- I3. CMP ダミーを仮想した配線容量抽出モデルの評価方法
- I4. 上記を用いた評価結果
- I5. CMP プロセスを考慮した抵抗抽出モデルの評価方法



役に立つ理由

○実業務に非常に密着していたため。○業務に直結する評価である。○誤差要因の認識。○評価によりダミーによる値の大きさのΔが大体分かった。○千鳥格子、縦方向の話は参考になりました。○1つ、できればスライドと手持ち資料を合わせて頂きたかった…○社内モデルとの比較、参照用に有用かと。○参考になりました。○モデルの精度向上。○事前に検討しておく必要があり参考になった。○誤差が気になっている。○わかればそれなりの対応が決められる。○ダミーを考慮する必要あり。○自社でやってないので参考になった。)

役に立たない項目 (I2=1人)

役に立たない理由

○容量が何に影響を与えるのか知りたかった。○CMP ダミーや 3D 効果をモデリングした場合とそうでない場合の容量比はどれぐらいになるのか？

今後検討して欲しいテーマ・課題

○評価パターン、ベンチマーク手法公開、標準化。○フローティングダミーが Vcc、Gnd に帯電したことを考慮できるか。○実 LSI へアプライした結果を示して頂けると参考になります。

[その他]

7. 今年度の活動内容とは関係なく、今後検討して欲しいテーマや課題を教えてください。

- 標準化を目指した活動をして欲しい。(ex) 同時動作に対する考え方が同一である (ASIC 設計等で利用できる)。
- Dynamic-Irdrop の遅延へのフィードバック手法。
- OCV 統合モデル→遅延、ノイズ。
- ノイズ対策に関する回路技術。モデリングしたものを、実 LSI へアプライした結果を示して頂けると良いと考えます。
- SSTA 統計的 STA。
- 高速配線モデルの開発。

以上

4.2 SystemCタスクグループ 2004年度活動報告

JEITA EDA技術専門委員会
標準化小委員会

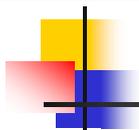
SystemCタスクグループ

JEITA

© Copyright 2005 JEITA, All rights reserved

SystemCタスクグループメンバー

主査	長谷川 隆	(富士通)
副主査	後藤 和永	(NECエレクトロニクス)
委員	清水 靖介	(沖電気)
	森井 一也	(三洋電機)
	岡田 和久/山田 晃久	(シャープ) ※
	柿本 勝	(ソニー)
	龍田 純一/逢坂 孝司	(ケイデンス) ※
	中野 淳二	(シノプシス)
	今井 浩史	(東芝)
	竹村 和祥	(松下電器)
	中村 和秀	(メンター)
	塚本 泰隆	(リコー)
	渡邊 政志	(ルネサステクノロジ)
	河原林 政道	(NEC Electronics America)
客員	今井 正治	(大阪大学)
		(計15名、※期中交代あり)



目次

はじめに

4.2.1 SystemC概要

4.2.2 SystemCタスクグループ設立の背景と目的

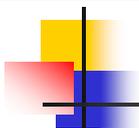
- SystemCタスクグループの活動内容
- SystemC標準化の枠組み

4.2.3 2004年度の成果と2005年度の計画

4.2.4 SystemCユーザフォーラム2005

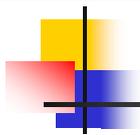
4.2.5 SystemC動向調査報告

4.2.6 IEEE版LRMLレビュー結果

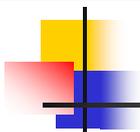


はじめに

- 本書では、SystemCタスクグループ(以下文中ではSC-TGと省略する場合あり)が2003年10月に発足以来行ってきた活動内容及び成果について報告する。

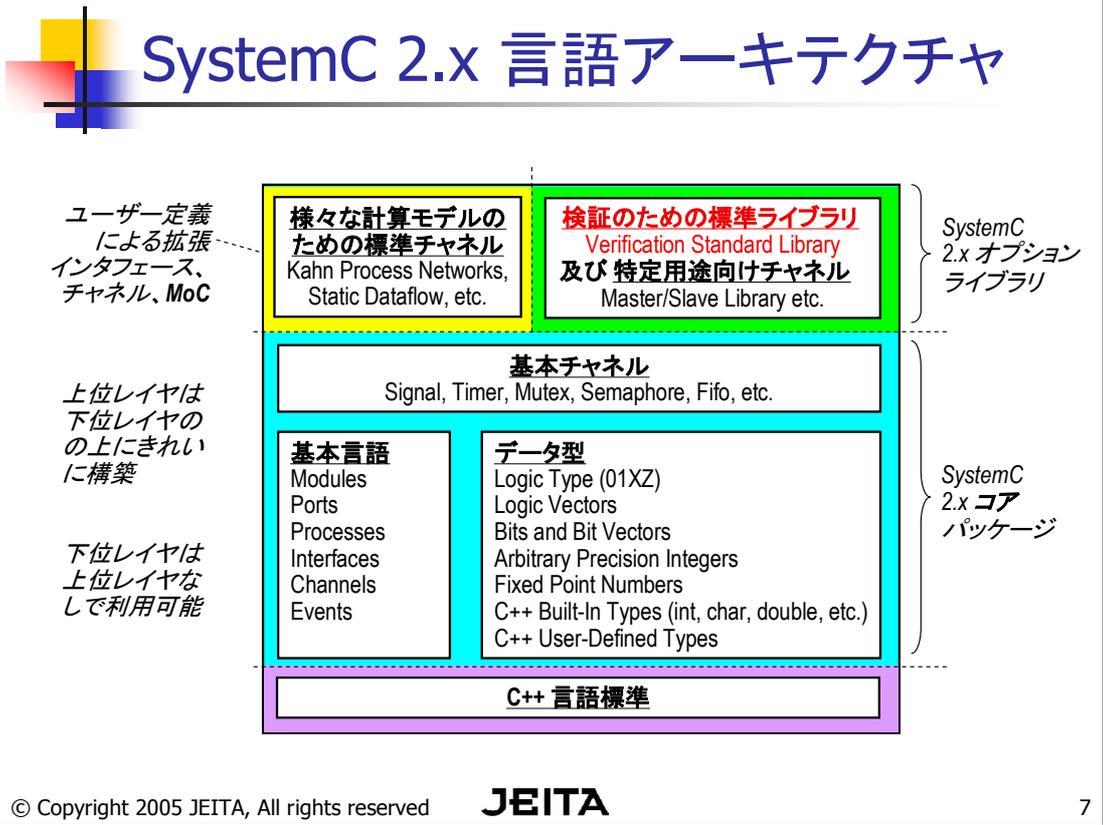


4.2.1 SystemC概要



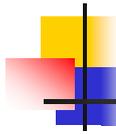
SystemC とは

- C++言語をベースとした、システムレベル設計言語の代表的な言語である
 - Open SystemC Initiative (OSCI)という標準化組織により、言語仕様(LRM)とリファレンスシミュレータが策定され、無償提供されている
<http://www.systemc.org/>
 - 現在は合成サブセットやTLMの標準化案が検討されている
 - 2004年12月にIEEEにLRMが移管され、P1666として標準化がすすめられている
- C++の文法を保持したまま、クラスライブラリの形で以下のような言語拡張がなされている
 - 並列動作を可能とするシミュレーションエンジン(クロック、イベント、等)
 - 抽象化された通信手段(Channels, Interfaces)
 - ハード実装に必要なデータタイプ (固定小数点、固定長ビット、等)
 - 0, 1, Z, X 等の信号値等



4.2.2 SystemCタスクグループ 設立の主旨と目的

© Copyright 2005 JEITA, All rights reserved **JEITA** 8



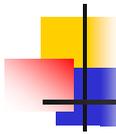
SC-TG設立の背景と目的

■ 背景

- SystemCは、SoC(System on Chip)の開発のためのシステムレベル記述言語のひとつとして、既に幅広く使われてきている。また2004年11月に言語開発元のOSCIよりIEEEに移管され、P1666として正式な標準化活動も開始されており、ますます普及に拍車がかかると予想される。

■ 目的

- 日本国内における唯一のSystemCの標準化関連組織として、OSCIやIEEE P1666ワーキンググループと連携しつつ、日本国内の事情・要求事項を取り込むべくSystemCの国際標準化を進めていく。また、SystemCに関連した調査結果をアニュアルレポートやユーザフォーラム等で積極的に情報発信を行うことで、国内普及を図る。これらにより日本の産業界の国際競争力を高めることを目指す。



SC-TGの活動内容

■ SystemC標準化活動

- IEEE P1666のメンバーとして、SystemC言語標準化活動に参画する(言語仕様のレビュー、追加修正要求等)

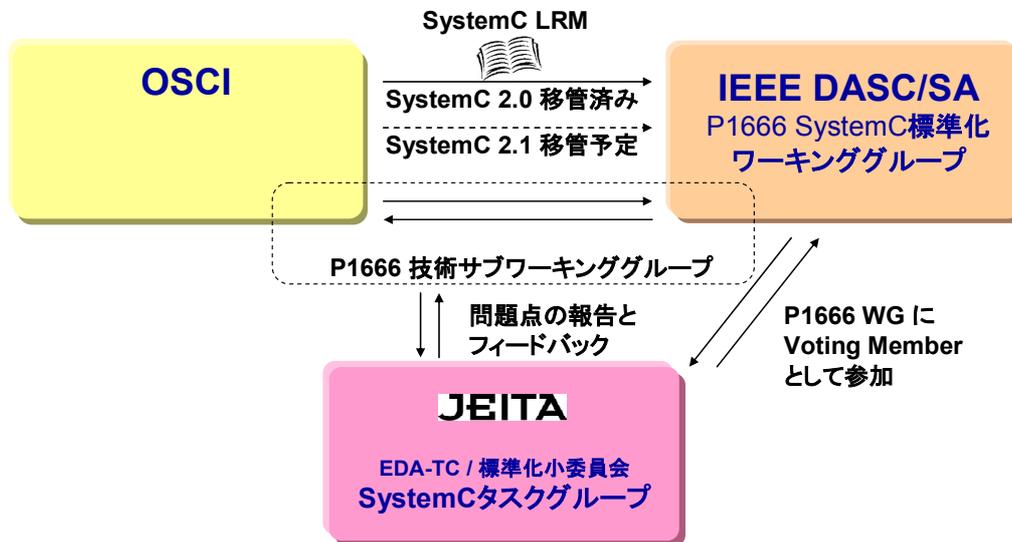
■ SystemC技術調査

- 世界各国におけるSystemCの利用状況や設計フローの動向を調査する
- 合成サブセットやTLM、及び検証ライブラリといった拡張言語仕様について調査し、標準化の検討を行う
- SystemCとSystemVerilogを連携したフローの検討や、連携のための技術的課題の先行抽出等を行う

■ SystemC普及活動

- EDSFIに併設したシステムデザインフォーラムの1プログラムとしてSystemCユーザフォーラムを開催し、積極的に情報発信を行いSystemCを利用した設計の普及をはかる

SystemC標準化の枠組み



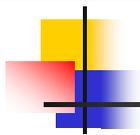
IEEE P1666 メンバー(2005年3月現在)

■ Voting Member

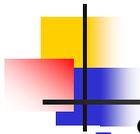
- Cadence Victor Berman (Chair),
Stuart Swan (Tech.SWG Chair)
- Calypto Skip Hovsmith
- MentorGraphics Dennis Brophy
- Synopsys Oz Levia
- JEITA Takashi Hasegawa,
Masamichi Kawarabayashi
- Jeda Eugene Zhang

■ Observer

- Freescale Rob Slater
- Tensilica Grant Martin
- Summit Design Vincent Viteau
- Xilinx Adam Donlin
- Pentum Bodo Parady

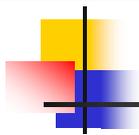


4.2.3 2004年度の成果と 2005年度の計画



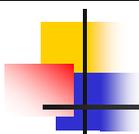
SC-TG 2004年度の成果まとめ

- SystemC標準化活動
 - IEEE P1666の正式メンバーとなり、SystemC言語標準化活動を開始した
 - 2004年12月にリリースされたIEEE版のLRM(Draft)のレビューを行った(詳細については4.2.6 IEEE版LRMレビュー結果に記載)
- SystemC技術調査
 - 過去5年間に一般に公開されているSystemC関連の論文や発表資料等50件の調査を行った(詳細については4.2.5 SystemC動向調査報告に記載)
- SystemC普及活動
 - 2005年1月27日に、EDSF併設のシステムデザインフォーラムの1プログラムとしてSystemCユーザフォーラム2005を開催した。またアンケート調査も実施した(詳細については4.2.4 SystemCユーザフォーラム2005開催報告に記載)

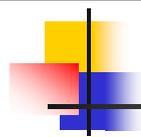


SC-TG 2005年度の活動目標

- SystemC標準化活動
 - 2004年度に報告したIEEE版LRMの問題点についてフォーローを行う
 - 4月にリリース予定のSystemC 2.1の組み込まれたLRMについてレビュー活動を行い、追加修正要求等を行う
- SystemC技術調査
 - SystemC動向調査を継続して実施する
 - 合成サブセットやTLM、及び検証ライブラリといった拡張言語仕様について調査し、標準化の検討を行う
- SystemC普及活動
 - 2004年度の実績を踏まえて、SystemCユーザフォーラムを開催し、積極的に情報発信を行いSystemCを利用した設計の普及をはかる



4.2.4 SystemCユーザフォーラム 2005 開催報告



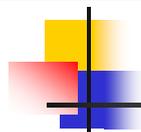
SystemCユーザフォーラム2005概要

- 主催: JEITA EDA技術専門委員会
- 協賛: OSCI、EDAベンダー6社
- 日時: 2005年1月27日 15:10~17:10
- 会場: パシフィコ横浜アネックスホールF202(定員200名)
- 講演内容:
 - 司会: 長谷川 隆(富士通)
 - 河原林 政道氏(NEC Electronics America): 「SystemCのIEEE標準化状況と最新情報」
 - 高嶺 美夫氏(ルネサステクノロジ): 「短TAT性能評価プラットフォームとその応用」
 - 塚本 泰隆氏(リコー): 「動作合成ツールを使ったJPEG(DCT)回路の設計事例」
 - 柿本 勝氏(ソニー): 「動作合成の設計適用事例とその効果」

© Copyright 2005 JEITA, All rights reserved

JEITA

17



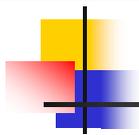
SystemCユーザフォーラム2005を終えて

- 定員200名の会場に250名弱の聴講者が訪れ、立ち見が出るほどの盛況であり、SystemCに関する関心の高さがうかがわれた
- また各講演の後に質問時間を設けたが、それぞれ1~2件の質問があり、活発な意見交換の場となった。
- アンケート調査を実施した結果、次のようなコメントが寄せられた。次回以降開催時の参考としたい。
 - 会場が狭い。机が欲しい。休憩時間が短い。
 - 発表内容に重なりが多い。プレゼンと配布資料を合わせて欲しい。
 - Accellera, OSCI, JEITAに求められる事として、セミナー・ワークショップの定期開催等を通じての日本語による情報発信。

© Copyright 2005 JEITA, All rights reserved

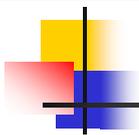
JEITA

18

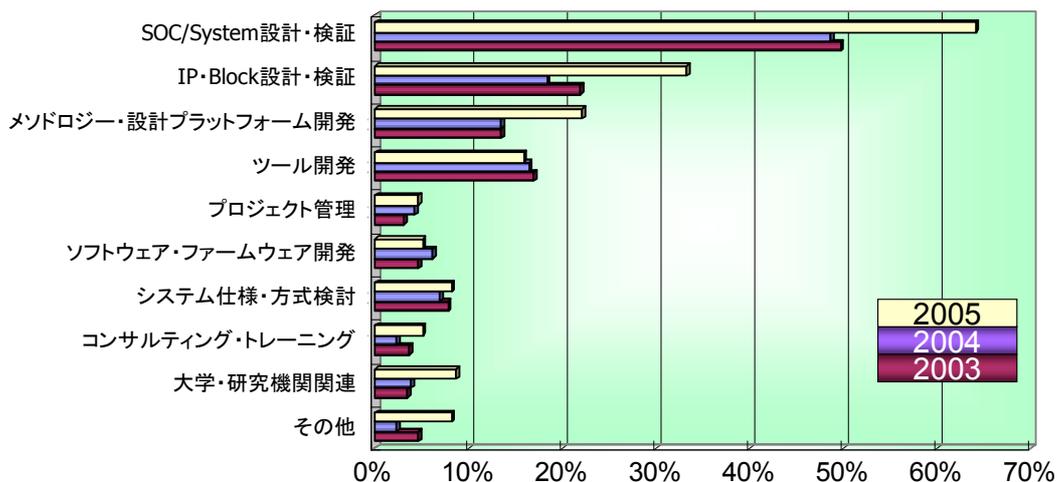


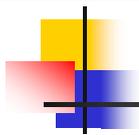
アンケート調査集計結果

- 本ユーザフォーラムは、過去4回に渡りOSCIが主催してきたが、今回よりJEITA EDA技術専門委員会の主催としたものである
- OSCIの厚意により前回及び前々回のアンケート調査結果をいただき、今回の調査結果と合わせて聴講者の動向について分析を行うことができた
 - 大まかな傾向としては、主な使用言語はVerilog HDLが相変わらず多数を占めるが、SystemCに関しては様子見の段階から(部分的)使用の段階へ移行しつつあるようだ
 - また、SystemCがより普及するためには、高位合成などのツールのさらなる整備が必要と思われる
- 次ページ以降に、集計結果をまとめた

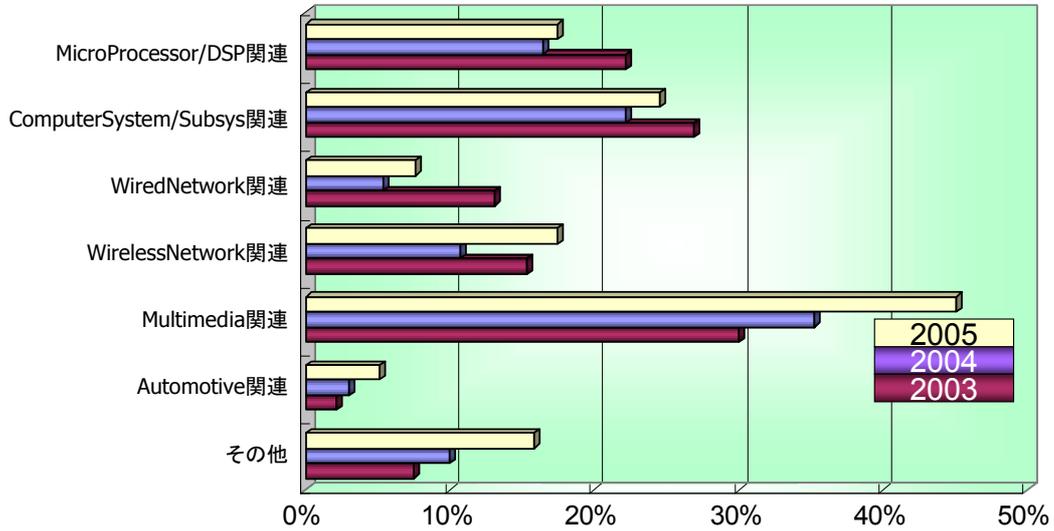


1. ご担当業務またはビジネスは？





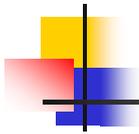
2. ご担当製品アプリケーションは?



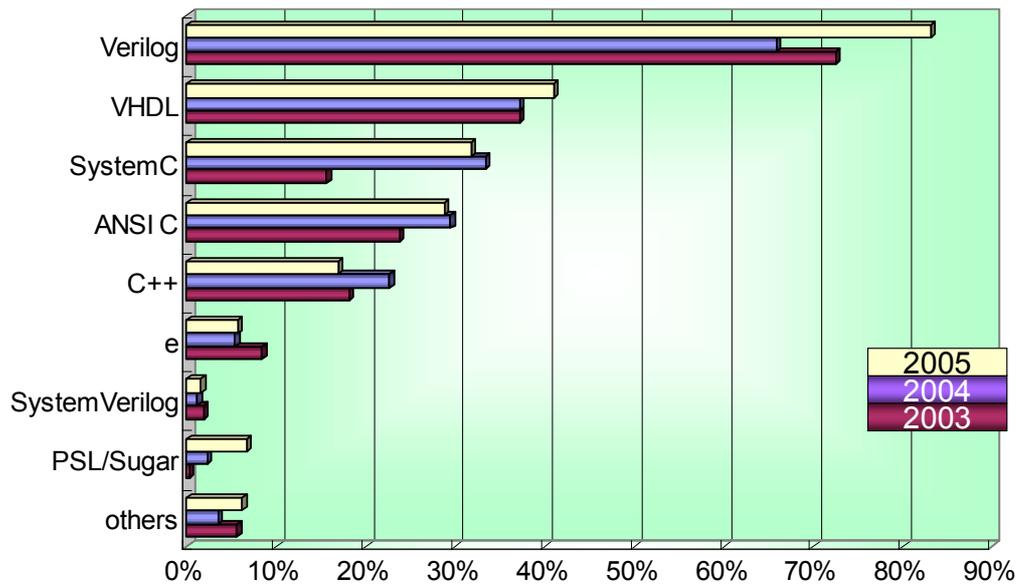
© Copyright 2005 JEITA, All rights reserved

JEITA

21



3. 現在主に使用している言語は?

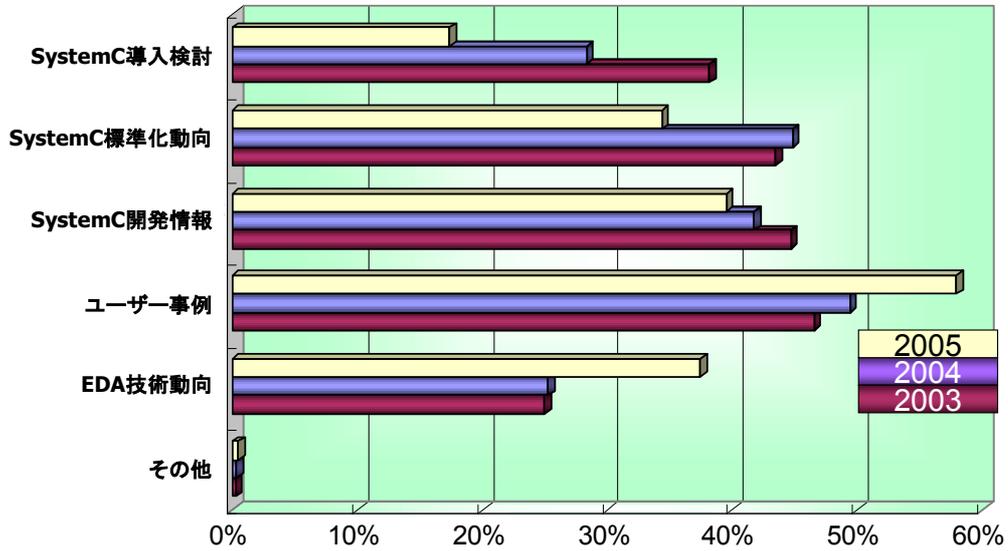


© Copyright 2005 JEITA, All rights reserved

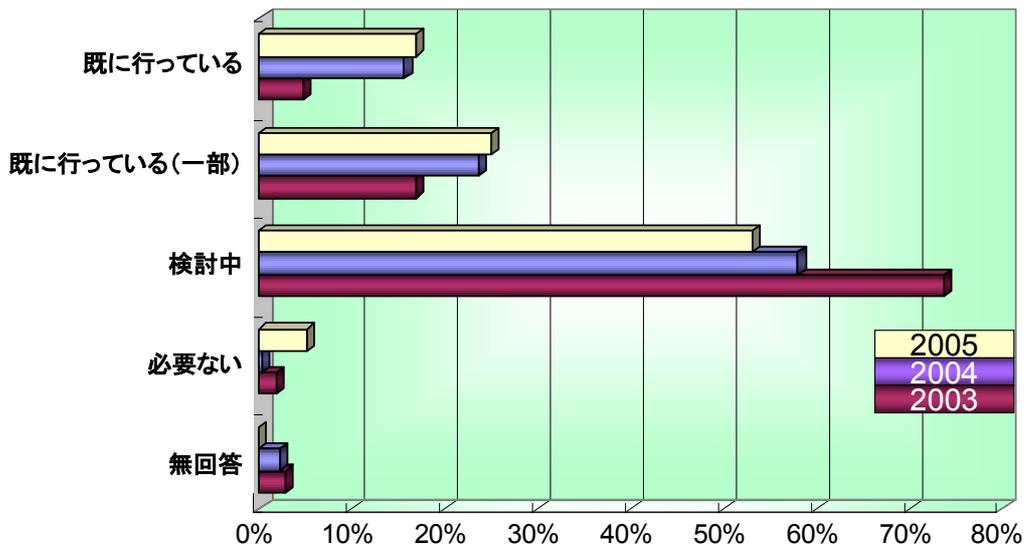
JEITA

22

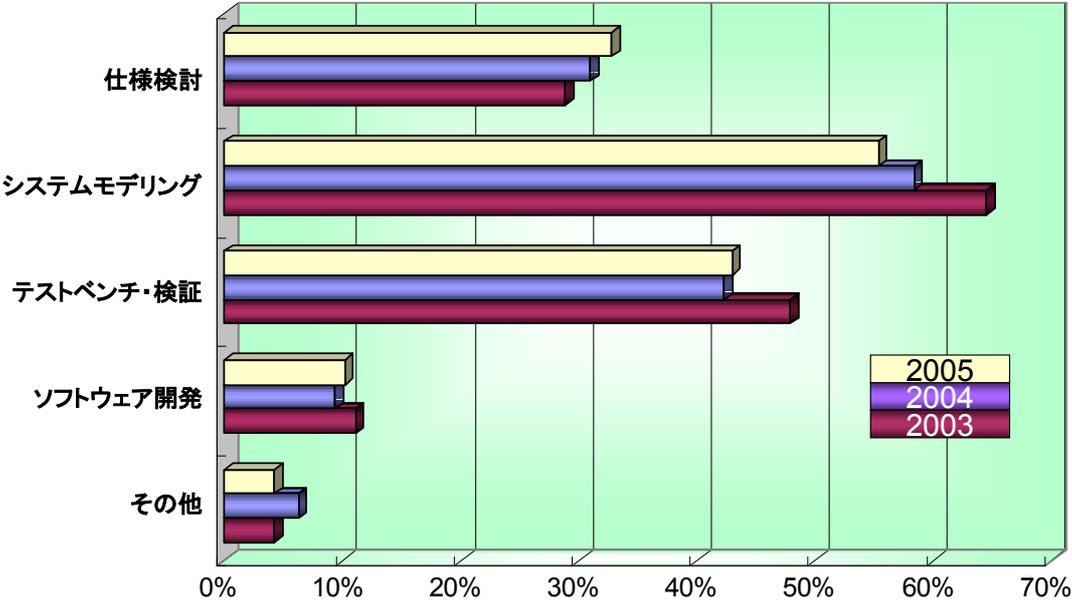
4. SystemCユーザフォーラムに参加された目的は?



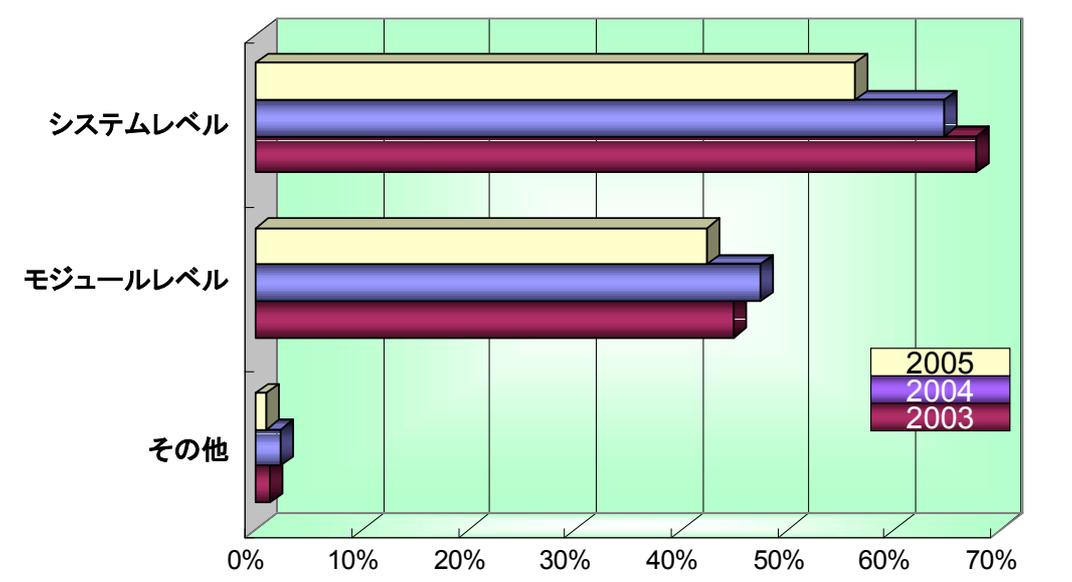
5. SystemCでの設計・検証環境構築について



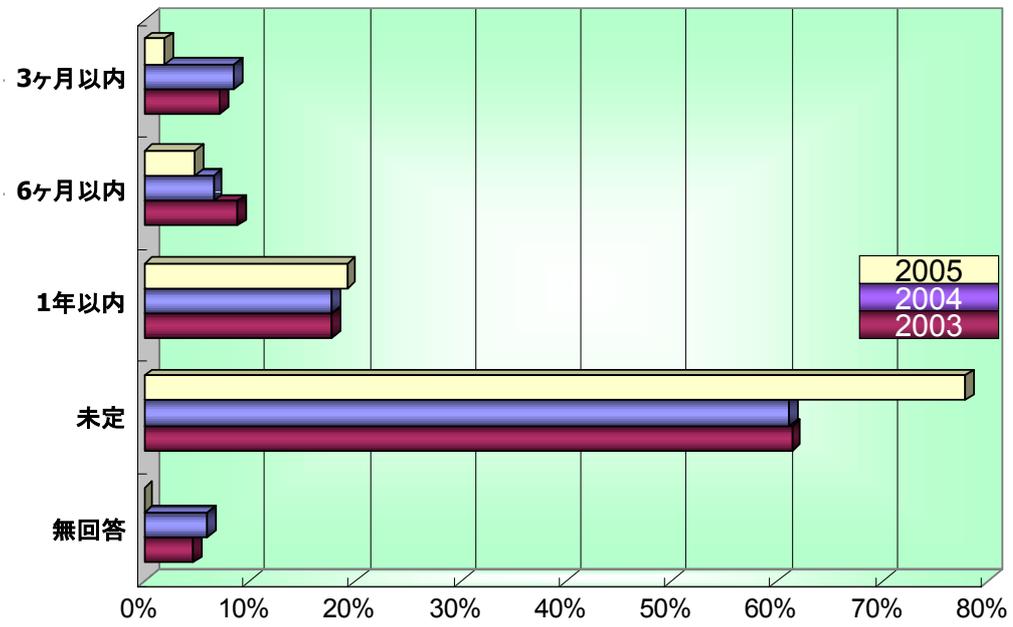
6. 「5」で「既に行っている」または「検討中」と回答された方 a) SystemCの使用目的は?



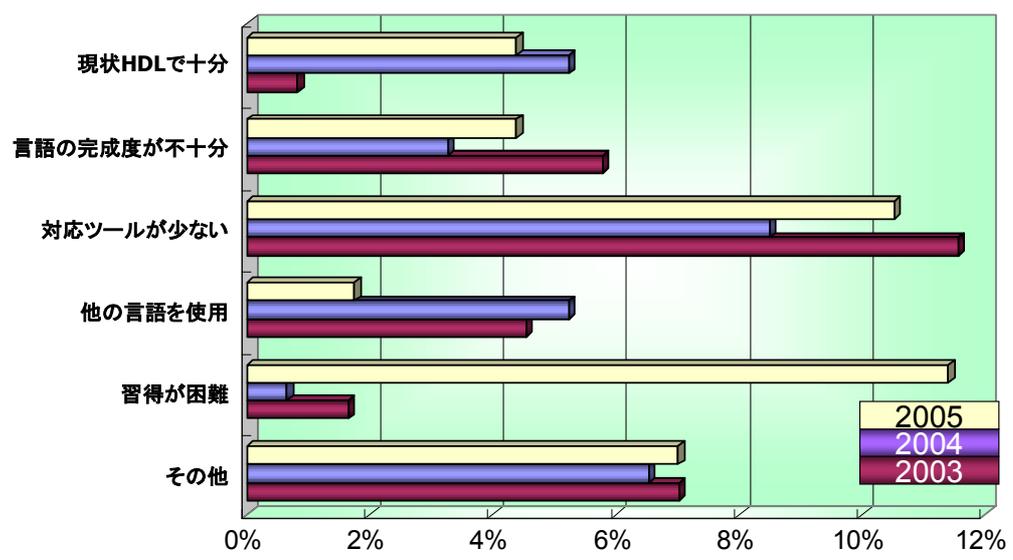
b) SystemCの活用範囲は?



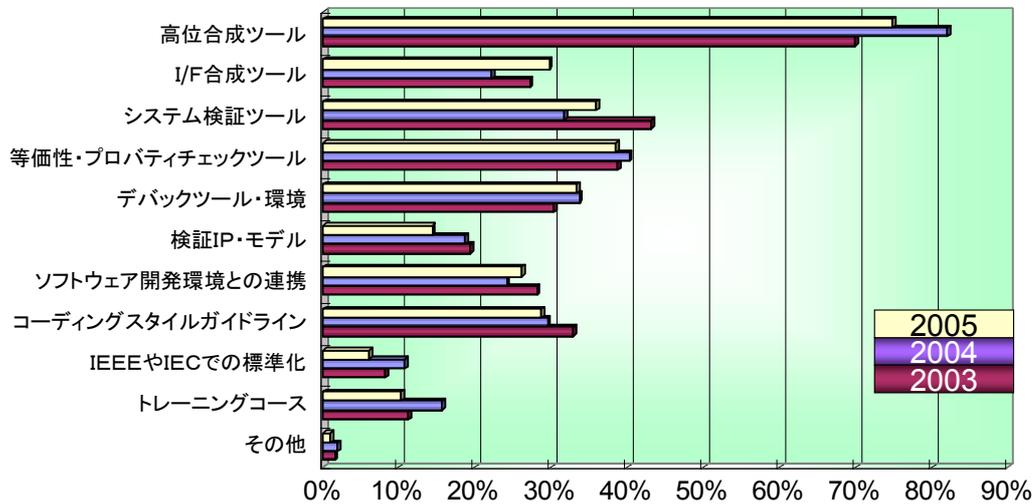
7. 「5」で「検討中」と回答された方へ 導入予定時期は?



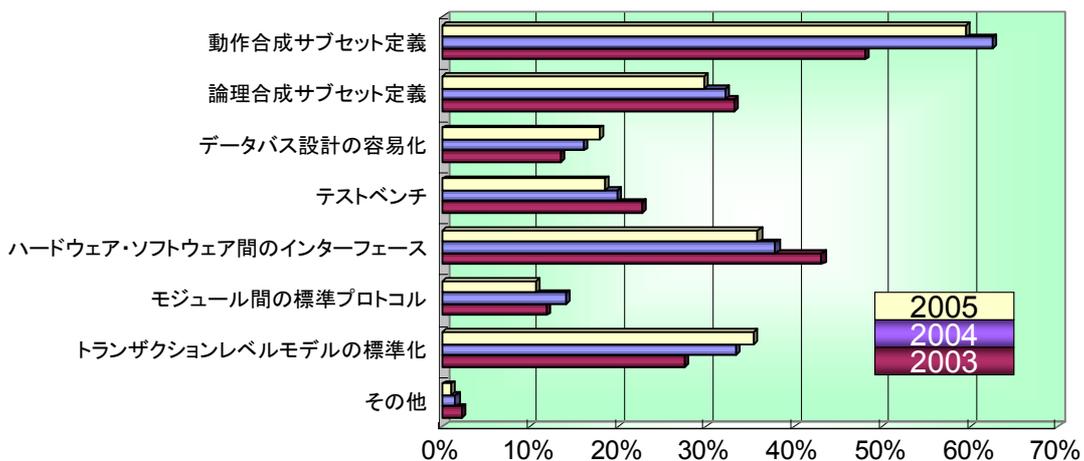
8. 「5」で「必要ない」、「検討中」と回答された方へ 導入の弊害となっている理由は?

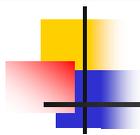


9. SystemCをより活用する為に充実が必要なものとは？

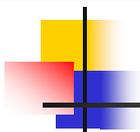


11. 今後SystemCの言語拡張・標準化で期待することは？



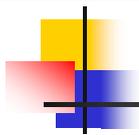


4.2.5 SystemC動向調査報告



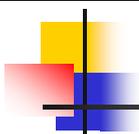
SystemC動向調査の内容

- JEITA SystemC-TGの本年度の活動の1つとして、過去5年間のSystemCに関する論文・発表等を調査し、SystemCの動向調査を行った。
- 調査対象は学会(DAC,DATE等)、大学関係(Stanford、UCI等)、SystemCユーザグループ等の多岐にわたる。
- 今回はその中でも特に興味深い内容をまとめて挙げている。なお過去5年間に調査対象とした為、現在では内容的に多少古いものも含まれている。
- 今後も引き続き同様の調査を行っていく予定である。



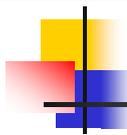
調査内容のサマリ

- 今回取り上げた内容は全部で50件になる。
 - 【カテゴリ別分類】
 - 言語 6件
 - モデリング 16件
 - 検証 5件
 - 性能解析 4件
 - Co-Design/Co-Simulation 9件
 - 合成 7件
 - ツール 2件
 - 情報 1件
 - 【調査ソース別分類】
 - 学会 29件
 - CODES (International Conference on Hardware Software Codesign) 3件
 - DAC (Design Automation Conference) 8件
 - DATE (Design, Automation and Test Conference in Europe) 18件
 - 大学関係 11件
 - Stanford (Leland Stanford Junior University, USA) 3件
 - UCI (University of California, Irvine, USA) 7件
 - Tubingen (University of Tubingen, Germany) 1件
 - ユーザグループ 10件
 - ESCUG (European SystemC User Group Meeting) 7件
 - 日本SCUG (日本 SystemC User Group Meeting) 3件
- カテゴリ別分類では「モデリング」に関する発表が一番多い。また、調査ソース別にみるとDATEやESCUG等の欧州での発表が多い。



SystemC動向調査一覧 (1/5)

No.	学会、大学	タイトル	著者	概要	分類
1	CODES2003	A Modular Simulation Framework for Architectural Exploration of On-Chip Interconnection Networks.	Tim Kogel, Malte Doerper, Andreas Wiefierink, Rainer Leupers, Gerd Ascheid, Heinrich Meyr, Serge Goossens	様々なチップの通信アーキテクチャ検討のために、SystemCの階層チャネルを使ったTLMのフレームワークを構築した。各モデルが持つパラメータ値からパフォーマンスを算出する。	検証
2	CODES2003	Transaction Level Modeling: An Overview	Lukai Cai and Daniel Gajski	曖昧のまま使われているTLMについて、抽象度の面からモデルを厳格に分類し、それらを設計フローの中でどのように進めていくかを提案する。	モデリング
3	CODES2003	Extending the SystemC Synthesis Subset by Object Oriented Features.	Eike Grimpe, Frank Oppenheimer	OODを活かした設計と、それをSystemCで実現するために独自で追加した仕様を紹介。合成試行は、自前で用意したツールで実施している。	合成
4	DAC2002	System on chip design: Automatic generation of embedded memory wrapper for multiprocessor SoC	Ferid Gharsalli, Samy Meftali, Frédéric Rousseau, Ahmed A. Jerraya	メモリアッパーを使用したメモリとプロセッサ間のRTLアーキテクチャ検討手法	性能解析
5	DAC2002	A novel synthesis technique for communication controller hardware from declarative data communication protocol specifications	Robert Siegmund, Dietmar M uller	データ通信プロトコル仕様からの通信制御回路合成手法の紹介	合成
6	DAC2002	Component-based design approach for multicore SoCs	W. Cesário, A. Baghdadi, L. Gauthier, D. Lyonnard, G. Nicolescu, Y. Paviot, S. Yoo, A. A. Jerraya, M. Diaz-Nava	ハイレベルコンポーネントベースのマルチコアSOC設計手法の紹介	Co-Design / Co-Simulation
7	DAC2004	Extending the Transaction Level Modeling Approach for Fast Communication Architecture Exploration	Sundeep Pasricha, Nikil Dutt, Mohamed Ben-Romdhane	CCATB抽象モデルによる高速なコミュニケーション・アーキテクチャ探索を行う為の方法論	モデリング
8	DAC2004	An Area Estimation Methodology for FPGA Based Designs at SystemC-Level	Carlo Brandolesse, William Fornaciari, Fabio Salice	FPGAをターゲットとしたSystemCモデルのエリア見積もり手法	性能解析
9	DAC2004	Performance Analysis of Different Arbitration Algorithms of the AMBA AHB Bus	Massimo Conti, Marco Caldari, Giovanni B. Vece, Simone Orcioni, Claudio Turchetti	クロック・アク્યレートのアムバ AHBバスモデル (SystemC2.0.1) のアービトレーション・アルゴリズムの解析	性能解析
10	DAC2004	Debugging HW/SW Interface for MPSoC: Video Encoder System Design Case Study	Mohamed-Wassim Youssef, Singjoo Yoo, Arif Sasogko, Yanick Paviot, Ahmed A. Jerraya	シーケンシャルなコードをマルチプロセッサSoC上へ実装するデザインフローをビデオエンコーダの実例により紹介	Co-Design / Co-Simulation



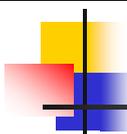
SystemC動向調査一覧 (2/5)

No.	学会、大学	タイトル	著者	概要	分類
11	DAC2004	SUNMAP: A Tool for Automatic Topology Selection and Generation for NoCs	Srinivasan Murali, Giovanni De Micheli	アプリケーションに最適なトポロジーの選択とトポロジ上へのコアのマッピングを自動的に行うSUNMAPツールの紹介	ツール
12	DATE2000	System level design using C++	Diederik Verkest, Joachim Kunkel, Frank Schirmeister	C++を利用した設計言語(SystemC, CynApps/Forte, Ocapi/IMECI)に関する概要紹介	言語
13	DATE2001	Behavioral Synthesis with SystemC	G. Economakos, P. Oikonomakos, I. Panagopoulos, I. Poulakis, G. Papakonstantinou	SystemCベースの動作合成手法	合成
14	DATE2001	A Methodology for Interfacing Open Source SystemC with a Third Party Software	L. Charest, M. Reid, E. Mostapha Aboulhamid, G. Bois	サード・パーティ・ソフトウェアのSystemCへの接続方法	言語
15	DATE2001	SystemCSV : An Extension of SystemC for Mixed Multi-Level Communication Modeling and Interface-Based System Design	Robert Siegmund, Dietmar Müller	インターフェースを導入して、通信と機能記述を分離。抽象度の異なる通信モデルを接続。	モデリング
16	DATE2001	The Simulation Semantics of SystemC	Wolfgang, Juergen Ruf, Dirk Hoffmann, Joachim Gerlach, Thomas Kropf, Wolfgang Rosenstiehl	Watching, signal代入, waitなどを含むSystemC 1.0のシミュレーション・セマンティックの紹介。	言語
17	DATE2001	An Efficient Architecture Model for Systematic Design of Application-Specific Multiprocessor SoC	Amer Baghdadi Damien, Lyonnard Nacer-E. Zergainoh, Ahmed A. Jerraya	マルチプロセッサ・アプリケーションの設計手法の提案。汎用アーキテクチャモデルによって、接合性・柔軟性・可測性の高い手法の提案。	Co-Design / Co-Simulation
18	DATE2001	A Model for Describing Communication between Aggregate Objects in the Specification and Design of Embedded Systems	Kjetil Svarstad, Gabriela Nicolescu, Ahmed A. Jerraya	近年、組み込みシステムの設計において設計の抽象度を上げることが重要になってきているが、同時に通信の重要度を上げることも重要である。この論文では、SystemCの拡張として通信モデルを実装。	言語
19	DATE2001	Network Processors: A Perspective on Market Requirements, Processor Architectures and Embedded S/W Tools	Pierre G. Paulin, Faraydon Karim, Paul Bromley	STMicroelectronicsの新しいネットワークプロセッサの構造と開発環境の紹介。	モデリング
20	DATE2001	Automatic Generation and Targeting of Application Specific Operating Systems and Embedded Systems Software	Lovic Gauthier, Sungjoo Yoo, Ahmed A. Jerraya	アプリケーションに特化したOSとアプリケーション・ソフトウェアの自動生成。	Co-Design / Co-Simulation

© Copyright 2005 JEITA, All rights reserved

JEITA

5



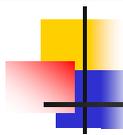
SystemC動向調査一覧 (3/5)

No.	学会、大学	タイトル	著者	概要	分類
21	DATE2001	Simulation-Guided Property Checking Based on Multi-Valued AR-Automata	J'urgen Ruf, Dirk W. Hoffmann, Thomas Kropf, Wolfgang Rosenstiehl	アサーションベースの高速プロパティチェック手法	検証
22	DATE2001	A Framework for Fast Hardware-Software Co-simulation	Andreas Hoffmann, Tim Kogel, Heinrich Meyr	高速HW/SWシミュレーションのためのフレームワークに関する手法。ISS, アセンブラ, リンカ生成	Co-Design / Co-Simulation
23	DATE2001	Mixed-Level Cosimulation for Fine Gradual Refinement of Communication in SoC Design	Gabriela Nicolescu, Sungjoo Yoo, Ahmed A. Jerraya	プロトコルの決まっていない通信からプロトコルの決まった通信へのリファインメント手順	モデリング
24	DATE2002	Automated Concurrency Re-Assignment in High Level System Models for Efficient System-Level Simulation	N. Savoiu, S. Shukla, R. Gupta	C++言語でのシミュレーション高速化(Threadの最適化)に関するものをSystemCを例に説明	モデリング
25	DATE2002	From System Specification To Layout: Seamless Top-Down Design Methods for Analog and Mixed-Signal Applications	R. Sommer, I. Rugen-Herzig, E. Hennig, U. Gatti, P. Malcovati, F. Maloberti, K. Einwich, C. Clauss, P. Schwarz, G. Noessing	SystemC2.0のAnalog-MixedSignalをモデリングするための拡張についての提案	モデリング
26	DATE2003	Transaction-Level Models for AMBA Bus Architecture Using SystemC 2.0	M. Caldari, M. Conti, M. Coppola, S. Curaba, L. Pieralisi, C. Turchetti	SystemC 2.0を用いたトランザクションレベルAMBAバスの検討	モデリング
27	DATE2003	System-Level Power Analysis Methodology Applied to the AMBA AHB Bus	M. Caldari, M. Conti, M. Coppola, P. Crippa, S. Orcioni, L. Pieralisi, C. Turchetti	AHBバスシステムに特化した省電力化に関するガイドラインの検討	性能解析
28	DATE2003	Exploring SW Performance using SoC Transaction-level Modeling	Imed Moussa, Thierry Grellier, Giang Nguyen	VISTAによるSystemC TLMを用いたSW開発環境の紹介	モデリング
29	DATE2003	Systematic Embedded Software Generation from SystemC	F. Herrera, H. Posadas, P. Sánchez, E. Villar	SystemCからのソフトウェアの生成	Co-Design / Co-Simulation
30	Stanford	x pipes Compiler: A tool for instantiating application specific Networks on Chip	Antoine Jalabert (LETI-DSIS), Srinivasan Murali, Luca Benini (Univ. of Bologna), Giovanni De Micheli	特定アプリ用のNetworks on Chip(NoC)を構築するツール。	合成

© Copyright 2005 JEITA, All rights reserved

JEITA

6



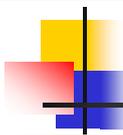
SystemC動向調査一覧 (4/5)

No.	学会、大学	タイトル	著者	概要	分類
31	Stanford	Bandwidth-Constrained Mapping of Cores onto NoC Architectures	Srinivasan Murali Giovanni De Micheli	バンド幅制約の下でコンポーネントをNetworks on Chip(NoC)アーキテクチャにマッピングするアルゴリズム。	合成
32	Stanford	Methodology for Hardware/Software Co-verification in C/C++	Luc Séméria Abhijit Ghosh (Synopsys Inc.)	プロセッサのBus Functional Model。	モデリング
33	Tubingen	Transaction Level Verification and Coverage Metrics by Means of Symbolic Simulation	Prakash M. Peranandam, Roland J. Weiss, Jurgen Ruf, Thomas Kropf	シンボリックシミュレーションにおけるトランザクション検証とカバレッジ	検証
34	UCI	Typing Abstractions and Management in a Component Framework	Frederic J. DoucetSandeep K. Shukla (Virginia Tech) Rajesh K. Gupta (University of California, San Diego)	コンポーネントのポートの型をすべて明示することなくアーキテクチャを構成できる環境。	合成
35	UCI	Transaction level modeling of SoC with SystemC 2.0	Sudeep Pasricha	TLMを使用したeSW開発、性能解析	Co-Design / Co-Simulation
36	UCI	Concurrency in System Level Design: Conflict between Simulation and Synthesis Goal	Nick Savoie, Sandeep K. Shukla, Rajesh K. Gupta	合成用Cモデル記述のシミュレーション速度の向上	ツール
37	UCI	Structural Design Composition for C++ Hardware Models	Vivek Sinha, Frederic Doucet, Chuck Siska, Rajesh Gupta	SystemC上でブロック配置情報を付加する仕組み	言語
38	UCI	Interfacing Hardware and Software Using C++ Class Libraries	Dinesh Ramanatham, Rajesh GuptaRay Roth	SWとHWのインターフェース	Co-Design / Co-Simulation
39	UCI	Transaction Level Modeling in System Level Design	Lukai Cai, Deniel Gajski	トランザクション・レベル・モデリングの定義	モデリング
40	UCI	RTOS Scheduling in Transaction Level Models	Haobo Yu, Andreas Gerstlauer, Daniel Gajski	トランザクション・モデルのRTOSによるスケジューリング	モデリング

© Copyright 2005 JEITA, All rights reserved

JEITA

7



SystemC動向調査一覧 (5/5)

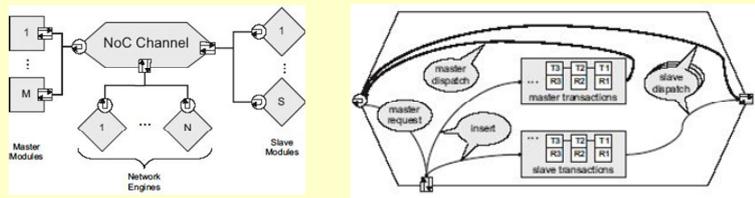
No.	学会、大学	タイトル	著者	概要	分類
41	ESCUG4 2001	Efficient Modeling and Simulation of Data Communication Protocols in Communication-oriented Designs using the SystemC-SV Extension	Robert Siegmund, Dietmar Mueller	SystemC-SV という言語拡張により、抽象度の異なるインターフェースを持つモジュールを自動的に接続し、シミュレーションを行う環境を提供する。	言語
42	ESCUG5 2002	A Method for the Development of Combined Floating- and Fixed-Point SystemC Models	Yves Vanderperren	fx_doubleというクラスを新たに作成し、浮動小数点モデルと固定小数点モデルに対して同じソースコードで済みます。固定小数点条件はプログラムの外部から与える。	モデリング
43	ESCUG7 2003	The SystemC Verification Standard (SCV)	Stuart Swan	SCV(SystemC Verification standard)1.0に関する簡単な紹介。これはSCV1.0のインストールパッケージにも含まれているスライド。	検証
44	ESCUG9 2004	SystemC 2.1 Overview	Stuart Swan	SystemC 2.1の概要。SystemC 2.1でエンハンス項目は、IPモジュール化の容易性向上(sc_argcvt)等の追加)、TLM(Transaction Level Modeling)のしやすさ向上(sc_export等の追加)、その他使いやすさの向上(.read()無しアクセス、異なる型の接続、バートセレクトの扱い等)。	情報
45	ESCUG9 2004	A Functional Coverage Prototype for SystemC-based Verification of Chipset Designs	Robert Siegmund, Ulrich Hensel, Andreas Herrholz, Isa Volt	SystemC環境におけるCoverage-APIのプロトタイプを作成した。AMDのチップセットの検証に適用した。	検証
46	ESCUG9 2004	System Architecture Performance Modeling with SystemC	Soeren Sonntag, Raimar Thudt, Tino Bertram	DSLAM(バケツ中継器)をSystemCを使いシステムレベルでシミュレーション。SystemCの有効性を示すとともにSystemCの問題点を指摘。	モデリング
47	ESCUG10 2004	XML-based Parameterization of SystemC Models	Andreas Doering	XMLを利用してマルチスレッドCPUのSystemCモデルをパラメタライズし、コンフィギュラブルなモデルと、そのモデルを展開する仕組みを作成し、アーキテクチャ評価した。	モデリング
48	日本SCUG 2003	Transaction Level Modeling with SystemC	Thorsten Groetker	SystemCでのトランザクションレベルのモデル手法。AMBA AHB/APBのバスモデル	モデリング
49	日本SCUG 2003	ハードウェア設計とSystemC	長尾文昭	三洋電機セミコンダクターカンパニーでの現状のSystemC設計環境	合成
50	日本SCUG 2003	SoCを利用したSoCシステムシミュレータ開発事例	竹村和祥	SoCシステムシミュレーション環境の開発事例	Co-Design / Co-Simulation

© Copyright 2005 JEITA, All rights reserved

JEITA

8

【1】 CODES 2003

タイトル	A Modular Simulation Framework for Architectural Exploration of On-Chip Interconnection Networks.
著者	Tim Kogel*,Malte Doerper*,Andreas Wieferink*,Rainer Leupers*,Gerd Ascheid*,Heinrich Meyr*,Serge Goossens**
所属	Aachen University of Technology **Coware
概要	様々なチップの通信アーキテクチャ検討のために、SystemCの階層チャンネルを使ったTLMのフレームワークを構築した。各モデルが持つパラメータ値からパフォーマンスを算出する。
内容	<p>通信システムのアーキテクチャ検討を、TLM I/Fで接続されたMaster/NoC(Network on Chip)/Slave/NetworkEngineから構成されるフレームワーク上で行う。NoC Channelは、各要素間の通信制御を行い、Network Engineは、通信に関する転送量情報を渡し、それからシステムのパフォーマンスが算出される。粗い精度ではあるが、システム構成の変更が比較適用に行える構成となっている。</p> 
分類	検証

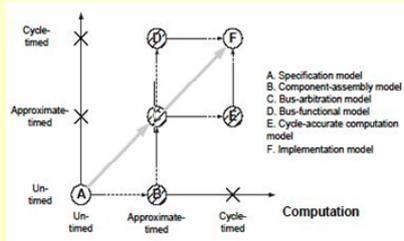
出展: Tim Kogel, "A Modular Simulation Framework for Architectural Exploration of On-Chip Interconnection Networks", CODES 2003

© Copyright 2005 JEITA, All rights reserved

JEITA

9

【2】 CODES 2003

タイトル	Transaction Level Modeling: An Overview
著者	Lukai Cai and Daniel Gajski
所属	University of California, Irvine
概要	曖昧のまま使われているTLMについて、抽象度の面からモデルを厳格に分類し、それらを設計フローの中でどのように進めていくかを提案する。
内容	<p>A: インプリは一切考えず機能だけを実装。プロセス間通信はチャンネルではなく変数で行う B: 並列動作するプロセスと広域メモリから構成され、アンタイムドのメッセージレベル通信を使う C: I/Fがバスとなり時間の概念が入ってくる。但しピンレベルでなくwait()による時間消費 D: I/Fがプロトコルチャンネルになり、厳格な精度となる。 E: 内部はピンレベルであり、ラッパを介し高抽象度I/F通信を行う。 F: I/Fと内部の両方がサイクル精度。所謂RTL</p> <p>各レベルのモデルがライブラリとして保存されることで、仕様変更に対し、各レベルの書き直し/評価を抑えることが出来る。</p> 
分類	モデリング

出展: Lukai Cai and Daniel Gajski, "Transaction Level Modeling: An Overview", CODES 2003

© Copyright 2005 JEITA, All rights reserved

JEITA

10

【3】 CODES 2003

タイトル	Extending the SystemC Synthesis Subset by Object Oriented Features.	
著者	Eike Grimpe, Frank Oppenheimer	
所属	OFFIS Research Institute	
概要	OODを活かした設計と、それをSystemCで実現するために独自で追加した仕様を紹介。合成試行は、自前で用意したツールで実施している。	
内容	<p>OODに沿ったSystemC設計をSystemCに独自のクラスを加え、従来の動作合成ツールがサポートしていなかったOODの設計を、ビヘイビアモデルに合成する。</p> <p>中でもポリモーフィズムとチャネル通信の代替として用いているグローバルオブジェクトについて説明。</p> <p>ポリモーフィズはポインタを使う代わりに、各要素にタグを付けて処理する「タグオブジェクト」を使う。</p> <p>チャネルは合成に向かないという観点から、プロセス間通信にチャネルを用いる代わりにグローバルオブジェクト(データメンバ)を用いる。</p> <p>これらの記述を含むSystemCコードを、自前のツールで合成し、それを現在ある動作合成ツールに与えることに成功している。</p>	
分類	合成	

出展: Eike Grimpe, "Extending the SystemC Synthesis Subset by Object Oriented Features. ", CODES 2003

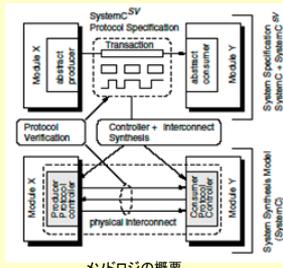
【4】 DAC 2002

タイトル	System on chip design: Automatic generation of embedded memory wrapper for multiprocessor SoC	
著者	Ferid Gharsalli, Samy Meftali, Frédéric Rousseau, Ahmed A. Jerraya	
所属	TIMA laboratory, Grenoble cedex, France	
概要	メモリアッパーを使用したメモリとプロセッサ間のRTLアーキテクチャ検討手法	
内容	<p>マルチプロセッサSoCにおける、メモリアッパーを使用したメモリとプロセッサ間のRTLアーキテクチャ検討手法。メモリアッパーによりフィジカルインターフェースの抽象度を引き上げ、メモリのタイプに依存しないアーキテクチャモデルにより、マルチプロセッサ環境下であっても複数の異なるメモリ検討を容易にし、最終的なアーキテクチャモデルの作成が可能になる。このアーキテクチャモデルが拡張SystemCで書かれる。</p> <p>【ジェネリック・メモリアッパー・アーキテクチャ】</p> <ul style="list-style-type: none"> ・メモリポートアダプタ(MPA) ・フィジカルメモリと内部バス間の接続と制御回路動作 ・コミュニケーションアダプタ(CA) ・バーチャル・モデルと内部バス間の接続(TLM) ・ラッパーバス、アービタ <p>【バーチャル・アーキテクチャ・モデル】</p> <ul style="list-style-type: none"> ・プロセッサ、DSPモデル(チャネルベース) <p>【ラッパー生成自動化によるメモリアッパー検討の向上】</p> <ul style="list-style-type: none"> ・SystemCシミュレーションモデル、VHDL RTL ・ライブラリ(MPA, CA, メモリIP) 	<p>メモリアッパー・アーキテクチャ (シングルポートメモリ例)</p> <p>アーキテクチャ・モデル (シングルポートメモリ例)</p>
分類	性能解析	

出典: Ferid Gharsalli and others, "System on chip design: Automatic generation of embedded memory wrapper for multiprocessor SoC", DAC2002

【5】 DAC 2002

タイトル	A novel synthesis technique for communication controller hardware from declarative data communication protocol specifications
著者	Robert Siegmund, Dietmar Müller
所属	Chemnitz University of Technology, Professorship Circuit and Systems Design
概要	データ通信プロトコル仕様からの通信制御回路合成手法の紹介
内容	<p>一般的な通信プロトコル仕様をSystemCの拡張であるSystemC-SVで記述し、プロトコル仕様からPFG(プロトコルフローグラフ)を作成して、論理合成可能なプロトコルコントローラのプロデューサ、コンシューマを生成するインターフェース合成とハイレベル検証手法。結果の比較にSynopsysのProtocol Compilerを使用。</p> <p>【通信プロトコル仕様の定義】</p> <ul style="list-style-type: none"> SystemC-SV (チェムニッツ大学) コミュニケーションアイテム定義 (トランザクション、メッセージ、フィジカルマップ) トランスミッションプロトコル定義 (シリアル、パラレル、リビート、セレクト) <p>【コントローラフローグラフの生成】</p> <ul style="list-style-type: none"> プロトコルフローアナリシス トランザクションFSMモデル <p>【コントローラ生成とインテグレーション】</p> <ul style="list-style-type: none"> CFGからプロデューサ、コンシューマモデルを生成 コントローラモデルを合成モデルにインテグレート (インターフェースポート>シグナル等) <p>【COSYNEによる自動化】</p> <ul style="list-style-type: none"> VHDL RTLモデルの生成 Protocol Compilerとの比較
分類	合成



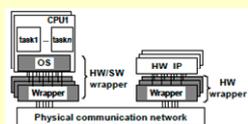
```

SystemC-SVによるプロトコル定義例
(IEC14443)
SV_TRANSACTION(IEC14443Frame) {
  /* parameters */
  SV_Param<sc_uint<8>> framelen;
  SV_ParamArray<sc_uint<8>, 8> data;
  SV_TRANSACTION_CTOR(IEC14443Frame)
  { SV_FROM << "TX"; SV_TO << "RX";
  ...
  /* protocol declaration */
  SV_COMPOSITION(
  SV_SERIAL(
  SoF_M(),
  SV_REPEAT(z,framelen,
  Data_M(framelen,data)),
  EOF_M());
  );
  /* specifications of SoF.Data.EOF left out*/
  SV_PHYMAP(PHYBi) { SV_Param<bool> b;
  SV_SignalRef<bool> XD;
  SV_PHYMAP_CTOR(PHYBi) { ...
  SV_ASSOCIATE(XD-b);
  };
  };
  
```

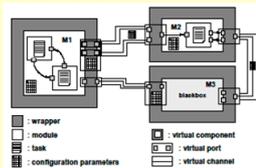
出典: Robert Siegmund and others, "A novel synthesis technique for communication controller hardware from declarative data communication protocol specifications", DAC2002

【6】 DAC 2002

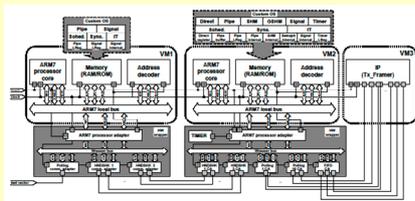
タイトル	Component-based design approach for multicore SoCs
著者	W. Cesário, A. Baghdadi, L. Gauthier, D. Lyonard, G. Nicolescu, Y. Paviot, S. Yoo, A. A. Jerraya, M. Diaz-Nava
所属	TIMA Laboratory, SLS Group STMicroelectronics, AST Grenoble Lab, France
概要	ハイレベルコンポーネントベースのマルチコアSOC設計手法の紹介
内容	<p>拡張SystemCによってシステム仕様を表し、コンフィグレーションパラメータと合わせてバーチャルアーキテクチャモデルを作成する。またハードとソフトのラッパーを自動生成して、システムレベルのコシミュレーションを可能にしている。</p> <p>【バーチャルアーキテクチャモデルの生成】</p> <ul style="list-style-type: none"> バーチャルモジュール、ポート、チャネル コンフィグレーションパラメータ (フィジカルアドレスポート等) <p>【ジェネリックアーキテクチャモデル】</p> <ul style="list-style-type: none"> コミュニケーションネットワークでラッパー間を接続 <p>【HW/SWラッパー自動生成】</p> <ul style="list-style-type: none"> プロセッサアダプタ、チャネルアダプタ、インターナルバス
分類	Co-Design/Co-Simulation



ジェネリックアーキテクチャモデル (マルチコアSoC)



バーチャルアーキテクチャモデル



マルチコアSoCアーキテクチャ

出典: W. Cesário and others, "Component-based design approach for multicore SoCs", DAC2002

【7】 DAC 2004

タイトル	Extending the Transaction Level Modeling Approach for Fast Communication Architecture Exploration
著者	Sundeep Pasricha, Nikil Dutt, Mohamed Ben-Romdhane
所属	Center for Embedded Computer Systems University of California, Irvine, CA Conexant Systems Inc. Newport Beach, CA
概要	CCATB抽象モデルによる高速なコミュニケーション・アーキテクチャ探索を行う為の方法論
内容	<p>SystemC2.0.11によるトップダウン設計手法の中でTLMモデルとBCAモデルのギャップを埋めるモデリング抽象度(CCATB)を定義し、AMBA2.0,3.0のバス・アーキテクチャのトレードオフ解析をBCAの精度を保持し高速に行う環境を構築。 CCATB : Cycle Count Accurate at Transaction Boundaries</p> <p>【バス・モデル高速化の為の特徴】</p> <ul style="list-style-type: none"> - サイクル精度はトランザクションの境界でのみ維持される 1. 計算処理はシミュレーション0近くでまとめて実行される 2. シミュレーション時間はまとめて集められる これがシミュレーションを高速化する - ペリフェラルなどの常に実行すべき動作を持たないIPは動作させない - リソース、時間のかかるthreadの代わりに軽いprocessを使用する - 複数シングルワードトランザクションはバースト内に集約 - 信号同期のオーバーヘッドを解消する為にファンクション・コールを使用 <p>バス・アーキテクチャ : タイミング、プロトコルの詳細を持つTLMチャンネルを拡張して作成 アービタ、デコーダ : CCATBチャンネル・モデルにより実装される</p> <p>【結果】平均、55%高速なシミュレーションを実現(対BCA)</p>
分類	言語

出典 : Sundeep Pasricha and others, "Extending the Transaction Level Modeling Approach for Fast Communication Architecture Exploration" DAC2004

© Copyright 2005 JEITA, All rights reserved

JEITA

15

【8】 DAC 2004

タイトル	An Area Estimation Methodology for FPGA Based Designs at SystemC-Level																								
著者	Carlo Brandolese, William Fornaciari, Fabio Salice																								
所属	Politecnico di Milano - DEI																								
概要	FPGAをターゲットとしたSystemCモデルのエリア見積もり手法																								
内容	<p>動作記述(SystemC)から情報を抽出してパラメータ化し、公式を用いてFPGA(Xilinx VertexII)のエリア(FF数とLUT数)を見積もるメソッドを紹介。メソッドの構造として、特定の合成ツールに依存することなく2つのステージに分かれている。このメソッドの評価に当たってCoCentric SystemC CompilerとMentor Graphics Leonardo Spectrumが使用された。</p> <p>【ハイレベルモデル(HLM)】 動作記述または動作合成ツールの結果から以下情報を抽出</p> <ul style="list-style-type: none"> - コントロールステップ数、コントロール入出力数 - サイズと数(データレジスタ、MUX、オペレータ) <p>HLM</p> <table border="1"> <tr> <td>#CS, #CI, #CO</td> <td>Number of control steps, inputs, outputs</td> </tr> <tr> <td>#REGS</td> <td>Number and size of data registers</td> </tr> <tr> <td>#MUS</td> <td>Number and size of multiplexers</td> </tr> <tr> <td>#OPS</td> <td>Number, type and size of operators</td> </tr> </table> <p>【ローレベルモデル(LLM)】 HLSモデルまたはHDL-RTL(またはゲート)を入力として公式に基づきFFとLUT数を見積る</p> <ul style="list-style-type: none"> - FF数(FSM、データレジスタ) - LUT数(FSM、データレジスタ、MUX、OPS、Glue Logic) <p>LLM</p> <table border="1"> <tr> <td>#FF_{FSM}</td> <td>Number of FFs for FSM registers</td> </tr> <tr> <td>#FF_{REGS}</td> <td>Number of FFs for data registers</td> </tr> <tr> <td>#FF</td> <td>Overall number of FFs</td> </tr> <tr> <td>#LUT_{FSM}</td> <td>Number of LUTs for FSM registers</td> </tr> <tr> <td>#LUT_{MUX}</td> <td>Number of LUTs for Multiplexers</td> </tr> <tr> <td>#LUT_{OPS}</td> <td>Number of LUTs for operators</td> </tr> <tr> <td>#LUT_{GLUE}</td> <td>Number of LUTs for glue logic</td> </tr> <tr> <td>#LUT</td> <td>Overall number of LUTs</td> </tr> </table> <p>計算式を算出する為のメソッド</p> <p>計算式の例</p> $\#LUT_{FSM} = [1.99 \cdot \#CS - 0.24 \cdot \#CI + 1.50 \cdot \#CO - 9.97]$ $\#LUT_{MUX} = \sum_{i=0, \dots, 255} [(0.68 \cdot \log_2(i) - 0.14) \cdot w(i)]$ $\#LUT_{GLUE} = 0.19(\#LUT_{FSM} + \#LUT_{OPS} + \#LUT_{MUX} + \#LUT_{REGS})$	#CS, #CI, #CO	Number of control steps, inputs, outputs	#REGS	Number and size of data registers	#MUS	Number and size of multiplexers	#OPS	Number, type and size of operators	#FF _{FSM}	Number of FFs for FSM registers	#FF _{REGS}	Number of FFs for data registers	#FF	Overall number of FFs	#LUT _{FSM}	Number of LUTs for FSM registers	#LUT _{MUX}	Number of LUTs for Multiplexers	#LUT _{OPS}	Number of LUTs for operators	#LUT _{GLUE}	Number of LUTs for glue logic	#LUT	Overall number of LUTs
#CS, #CI, #CO	Number of control steps, inputs, outputs																								
#REGS	Number and size of data registers																								
#MUS	Number and size of multiplexers																								
#OPS	Number, type and size of operators																								
#FF _{FSM}	Number of FFs for FSM registers																								
#FF _{REGS}	Number of FFs for data registers																								
#FF	Overall number of FFs																								
#LUT _{FSM}	Number of LUTs for FSM registers																								
#LUT _{MUX}	Number of LUTs for Multiplexers																								
#LUT _{OPS}	Number of LUTs for operators																								
#LUT _{GLUE}	Number of LUTs for glue logic																								
#LUT	Overall number of LUTs																								
分類	性能解析																								

出典 : N. Savoie et al. "Automated Concurrency Re-Assignment in High Level System Models for Efficient System-Level Simulation", Proceedings of Date 2002 pp.875-881, Paris, France, March 2002

© Copyright 2005 JEITA, All rights reserved

JEITA

16

【9】 DAC 2004

タイトル	Performance Analysis of Different Arbitration Algorithms of the AMBA AHB Bus																																																																																																																																																						
著者	Massimo Conti, Marco Caldari, Giovanni B. Vece, Simone Orcioni, Claudio Turchetti																																																																																																																																																						
所属	DEIT, Universita Politecnica delle Marche, via Breccie Bianche, I-60131, Ancona, ITALY																																																																																																																																																						
概要	クロック・アクキュレートAMBA AHBバスモデル(SystemC2.0.1)のアービトレーション・アルゴリズムの解析																																																																																																																																																						
内容	<p>検証対象となるアービトレーション・アルゴリズムは、4通りがあり、6つのマスタ2種類のトラフィックを与える事でバス・パフォーマンスを計測し、SJF with waiting time controlのアルゴリズムが一番良い結果となる事を検証(Wait時間短いスイッチング・アクティビティ低い)</p> <p>【アービトレーション・アルゴリズム】 Priority, Priority with waiting time control, Short Job First Short Job first with waiting time control</p> <p>【トラフィックの与え方】 High : 集中的にバスを使用(8,16ビット・バーストのランダム・シーケンス) Low : まばらにバスを使用(シングル,4ビット・バーストのランダム・シーケンス)</p> <p>【マスターのプライオリティ】 M1 high ... M5 low</p> <p>【他の考察】 同精度のVHDL AHBバス・モデルでもシミュレーションを行っており、SystemCデザインの方が50倍高速であった。 SystemCとVHDLシミュレーション環境の相違点から同じ条件ならSystemCの方がより高速な結果になるであろうと推測している SystemC : 5マスター + アービトレーション・アルゴリズムが複雑 VHDL : 2マスター + Priority with break</p>	<table border="1"> <thead> <tr> <th>Master #</th> <th>M1</th> <th>M2</th> <th>M3</th> <th>M4</th> <th>M5</th> <th>default</th> </tr> </thead> <tbody> <tr> <td>Traffic type</td> <td>High</td> <td>High</td> <td>Low</td> <td>Low</td> <td>Low</td> <td>-</td> </tr> <tr> <td colspan="7" style="text-align:center">Priority</td> </tr> <tr> <td>Max wait (clock cycles)</td> <td>15</td> <td>17</td> <td>83</td> <td>481</td> <td>602</td> <td>-</td> </tr> <tr> <td>Average wait (clock cycles)</td> <td>0.46</td> <td>0.46</td> <td>4.74</td> <td>5.88</td> <td>6.40</td> <td>-</td> </tr> <tr> <td>Bus use (%)</td> <td>27.7</td> <td>28.5</td> <td>9.0</td> <td>10.2</td> <td>9.8</td> <td>14.8</td> </tr> <tr> <td colspan="7" style="text-align:center">Priority with waiting time control</td> </tr> <tr> <td>Max wait (clock cycles)</td> <td>26</td> <td>40</td> <td>89</td> <td>96</td> <td>98</td> <td>-</td> </tr> <tr> <td>Average wait (clock cycles)</td> <td>0.47</td> <td>0.75</td> <td>5.68</td> <td>5.38</td> <td>5.9</td> <td>-</td> </tr> <tr> <td>Bus use (%)</td> <td>32.0</td> <td>30.4</td> <td>10.4</td> <td>11.2</td> <td>10.4</td> <td>5.6</td> </tr> <tr> <td colspan="7" style="text-align:center">Short Job First</td> </tr> <tr> <td>Max wait (clock cycles)</td> <td>85</td> <td>494</td> <td>20</td> <td>24</td> <td>22</td> <td>-</td> </tr> <tr> <td>Average wait (clock cycles)</td> <td>0.80</td> <td>1.89</td> <td>1.45</td> <td>1.98</td> <td>1.84</td> <td>-</td> </tr> <tr> <td>Bus use (%)</td> <td>30.3</td> <td>29.5</td> <td>10.6</td> <td>9.9</td> <td>9.7</td> <td>9.9</td> </tr> <tr> <td colspan="7" style="text-align:center">Short Job First with waiting time control</td> </tr> <tr> <td>Max wait (clock cycles)</td> <td>82</td> <td>95</td> <td>32</td> <td>41</td> <td>41</td> <td>-</td> </tr> <tr> <td>Average wait (clock cycles)</td> <td>1.25</td> <td>1.49</td> <td>1.98</td> <td>2.63</td> <td>2.78</td> <td>-</td> </tr> <tr> <td>Bus use (%)</td> <td>33.5</td> <td>32.6</td> <td>12.1</td> <td>10.7</td> <td>10.7</td> <td>0.3</td> </tr> </tbody> </table> <p>異なるアービトレーション・アルゴリズムのバスパフォーマンス</p> <table border="1"> <thead> <tr> <th rowspan="2">Arbitration algorithm</th> <th colspan="3">Normalized Switching activity</th> </tr> <tr> <th>ADDR</th> <th>DATA</th> <th>CONTROL</th> </tr> </thead> <tbody> <tr> <td>Priority</td> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>Priority with wait control</td> <td>1.00</td> <td>0.97</td> <td>0.85</td> </tr> <tr> <td>SJF</td> <td>1.03</td> <td>0.98</td> <td>0.86</td> </tr> <tr> <td>SJF with wait control</td> <td>1.02</td> <td>0.95</td> <td>0.78</td> </tr> </tbody> </table> <p style="text-align:right">スイッチング・アクティビティ</p>	Master #	M1	M2	M3	M4	M5	default	Traffic type	High	High	Low	Low	Low	-	Priority							Max wait (clock cycles)	15	17	83	481	602	-	Average wait (clock cycles)	0.46	0.46	4.74	5.88	6.40	-	Bus use (%)	27.7	28.5	9.0	10.2	9.8	14.8	Priority with waiting time control							Max wait (clock cycles)	26	40	89	96	98	-	Average wait (clock cycles)	0.47	0.75	5.68	5.38	5.9	-	Bus use (%)	32.0	30.4	10.4	11.2	10.4	5.6	Short Job First							Max wait (clock cycles)	85	494	20	24	22	-	Average wait (clock cycles)	0.80	1.89	1.45	1.98	1.84	-	Bus use (%)	30.3	29.5	10.6	9.9	9.7	9.9	Short Job First with waiting time control							Max wait (clock cycles)	82	95	32	41	41	-	Average wait (clock cycles)	1.25	1.49	1.98	2.63	2.78	-	Bus use (%)	33.5	32.6	12.1	10.7	10.7	0.3	Arbitration algorithm	Normalized Switching activity			ADDR	DATA	CONTROL	Priority	1	1	1	Priority with wait control	1.00	0.97	0.85	SJF	1.03	0.98	0.86	SJF with wait control	1.02	0.95	0.78
	Master #	M1	M2	M3	M4	M5	default																																																																																																																																																
Traffic type	High	High	Low	Low	Low	-																																																																																																																																																	
Priority																																																																																																																																																							
Max wait (clock cycles)	15	17	83	481	602	-																																																																																																																																																	
Average wait (clock cycles)	0.46	0.46	4.74	5.88	6.40	-																																																																																																																																																	
Bus use (%)	27.7	28.5	9.0	10.2	9.8	14.8																																																																																																																																																	
Priority with waiting time control																																																																																																																																																							
Max wait (clock cycles)	26	40	89	96	98	-																																																																																																																																																	
Average wait (clock cycles)	0.47	0.75	5.68	5.38	5.9	-																																																																																																																																																	
Bus use (%)	32.0	30.4	10.4	11.2	10.4	5.6																																																																																																																																																	
Short Job First																																																																																																																																																							
Max wait (clock cycles)	85	494	20	24	22	-																																																																																																																																																	
Average wait (clock cycles)	0.80	1.89	1.45	1.98	1.84	-																																																																																																																																																	
Bus use (%)	30.3	29.5	10.6	9.9	9.7	9.9																																																																																																																																																	
Short Job First with waiting time control																																																																																																																																																							
Max wait (clock cycles)	82	95	32	41	41	-																																																																																																																																																	
Average wait (clock cycles)	1.25	1.49	1.98	2.63	2.78	-																																																																																																																																																	
Bus use (%)	33.5	32.6	12.1	10.7	10.7	0.3																																																																																																																																																	
Arbitration algorithm	Normalized Switching activity																																																																																																																																																						
	ADDR	DATA	CONTROL																																																																																																																																																				
Priority	1	1	1																																																																																																																																																				
Priority with wait control	1.00	0.97	0.85																																																																																																																																																				
SJF	1.03	0.98	0.86																																																																																																																																																				
SJF with wait control	1.02	0.95	0.78																																																																																																																																																				
分類	性能解析																																																																																																																																																						

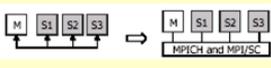
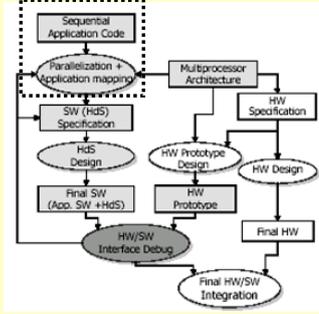
出典: Carlo Brandolesse and others, "An Area Estimation Methodology for FPGA Based Designs at SystemC-Level", DAC2004

© Copyright 2005 JEITA, All rights reserved

JEITA

17

【10】 DAC 2004

タイトル	Debugging HW/SW Interface for MPSoC: Video Encoder System Deign Case Study	
著者	Mohamed-Wassim Youssef, Sngjoo Yoo, Arif Sasogko, Yanick Paviot, Ahmed A. Jerraya	
所属	System Level Synthesis Group, TIMA Laboratory, Grenoble, France	
概要	シーケンシャルなコードをマルチプロセッサSoC上へ実装するデザインフローをビデオエンコーダの実例により紹介	
内容	<p>アプリケーションをマルチプロセッサSoC上へ実装する為のデザインフローが図1になる。このフローの中でシーケンシャルなアプリケーション・コードをMPI(Message Passing Interface)のAPIにより並列処理可能なコードに分割する。この時、分散されたタスクのコミュニケーションを担う、MPIプリミティブがSystemCにより実装されている。点線部分のParallelizationは人手により行われる為、並列化されたコードの正当性を検証する為、MPICHシミュレータとMPI/SCシミュレーション・モデル(SystemC)が使用されている。</p> <p>【パラレル化】 シーケンシャル・コードのシミュレーションからプロファイルを行いボトルネック部分を洗い出し3つのタスクに分ける (S1,S2,S3 : Mは他の処理を行う)</p>  <p>【MPIプリミティブの利用】 3つのタスク処理がコミュニケーション出来る様にMPIプリミティブによりPoint-to-Pointコミュニケーションを実装</p> 	 <p>図1. MPSoCデザインフロー</p>
	分類	Co-Design/Co-Simulation

出典: Mohamed-Wassim Youssef and others, "Debugging HW/SW Interface for MPSoC: Video Encoder System Deign Case Study", DAC2004

© Copyright 2005 JEITA, All rights reserved

JEITA

18

【11】 DAC 2004

タイトル	SUNMAP: A Tool for Automatic Topology Selection and Generation for NoCs
著者	Srinivasan Murali, Giovanni De Micheli
所属	Computer Systems Lab Stanford University Stanford, CA-94305, USA
概要	アプリケーションに最適なトポロジの選択とトポロジ上へのコアのマッピングを自動的に行うSUNMAPツールの紹介
内容	<p>トポロジの選択がコミュニケーション・ディレイ、パワー消費、エリアなどに影響を及ぼす、SUNMAPは最適なトポロジを自動選択する機能を備える。(図1: Phase1,2)</p> <p>【SystemCとの関係】 SUNMAPは、xpipesCompilerとxpipesアーキテクチャを使用する事で、ネットワークコンポーネントのSystemCモデルを生成する(図1: Phase3)</p> <p>【SUNMAP適用事例】 SUNMAPを6つのコアを持つDSPフィルタデザインへ適用(図2) 1. パタフライ・トポロジのネットワーク・コンポーネントが自動生成(図3) されDSPのNoCデザインがSystemCのサイクル・アキュレート、シグナル・アキュレートによりシミュレーション 2. 他のトポロジの選択とシミュレーション結果からレイテンシを計測</p> <p>結果: パタフライ・トポロジがレイテンシ最小(図4)</p>
分類	ツール

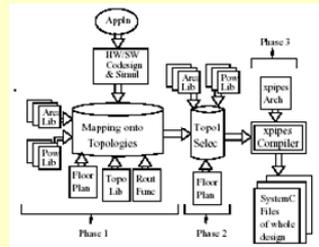


図1. SUNMAPのデザインフロー

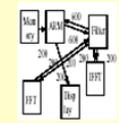


図2. DSPフィルタデザイン

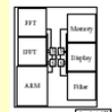
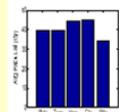


図3. パタフライ・フロアプラン



出典: Srinivasan Murali and others, "SUNMAP: A Tool for Automatic Topology Selection and Generation for NoCs", DAC2004

【12】 DATE 2000

タイトル	System level design using C++
著者	Diederik Verkest, Joachim Kunkel, Frank Schirmermeister
所属	IMEC, Synopsys, Cadence Design Systems
概要	C++を利用した設計言語(SystemC, CynApps/Forte, Ocapi/IMEC)に関する概要紹介
内容	<p>システムレベル設計においてC++を利用した設計言語 (SystemC, CynApps/Forte, Ocapi/IMECなど) が有効であることを説明(チュートリアル)</p> <ul style="list-style-type: none"> ・C++のクラスライブラリを利用してビット表現やデータタイプ、FSMの記述が可能 ・システム記述、アーキテクチャ記述、RTL記述、各抽象度間のリファインメントが可能 ・IPベースの設計にはC++のオブジェクト指向の考え方が有効 <p>ただし課題として、論理合成時のポインタやオブジェクト指向(仮想関数や多重継承)の扱いがある。</p>
分類	言語

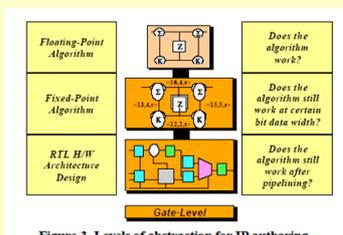


Figure 3. Levels of abstraction for IP authoring
システムレベルの設計抽象度

出展: D. Verkest et al. "System level design using C++" Proceedings of Date 2000 pp.74-81, Paris, France, March 2000

【13】 DATE 2001

タイトル	Behavioral Synthesis with SystemC
著者	G. Economakos, P Oikonomakos, I. Panagopoulos, I. Poulakis, G. Papakonstantinou
所属	National Technical University of Athens Department of Electrical and Computer Engineering
概要	SystemCベースの動作合成手法
内容	<p>SystemCベースの動作合成手法を用いることにより、新しいソフトウェアや既存のソフトウェアからハードウェア設計を始めることが可能となります。また、RTLコードを、Veirlog、VHDL及びSystemCで生成することにより、合成前後の結果を同じシミュレーション環境で検証することが可能となります。さらには、将来SystemCからの合成ツールがでてくれば、論理合成までもSystemCベースで行うことが可能となります。</p> <p>AGを用いた演算子のスケジューリング ↓ レジスタのスケジューリング ↓ 内部記述 Pattern Text Generator(PTG) ↓ SystemC、VHDL、Veirliog RTL記述生成</p> <p>【基礎理論】 動作合成のスケジューリング手法 AG(Attribute Grammer computational model) D.E. Knuth. Semantics of context-free languages. Mathematical Systems Theory 2(2):127-145, 1968 【応用理論】 Eli W.M.Waite. An executable language definition. ACM SIGPLAN Notices.28(2):21-40,1993 【結果】 AGを用いた計算量の少ない推定方法を応用することにより、高速に動作合成を行うことができた。</p>
分類	合成

出展: G.Economakos etc. "Behavioral Synthesis with SystemC"
Proceedings of Date 2001 pp.21-25, Munich, Germany, March 2001

© Copyright 2005 JEITA, All rights reserved



21

【14】 DATE 2001

タイトル	A Methodology for Interfacing Open Source SystemC with a Third Party Software
著者	L. Charest, M. Reid, E.Mostapha Aboulhamid , G. Bois
所属	Univ. of Cincinnati
概要	サード・パーティ・ソフトウェアのSystemCへの接続方法
内容	<p>SystemC外部のソフトウェアから内部信号を表示する場合、signalのリストを生成することとsignalの値の更新情報を取得することが必要となる。これを実現するため、sc_sginal_base内にnotify_interfaceを実装し、update時に必要な情報を取得できるようにした。</p> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 20px;"> <p>Graphical User Interface</p> <pre>my_sc_interfacec +update_signal()</pre> </div> <div style="border: 1px solid black; padding: 5px;"> <p>SystemCの推奨構造</p> <pre>sc interface -bound_instance:sc_interface* +update_signal() +get_bound_instance():const sc_interface* +is_bound():bool +bind() sc simcontext -signal_to_update:sc_signal_base** +time_stamp():double +crunch() +initial_crunch() sc_signal #update() +read():class T +notify_interface() sc_signal_base +update() +notify_interface()</pre> </div> </div> <p>【結果】 QTベースのGUIを接続し、 波形表示に成功。</p> <p>【推奨】 SystemCにこの構造を持たせることにより より用意に外部ソフトウェアが接続可能。</p>
分類	言語

出展: L. Charest etc. "A Methodology for Interfacing Open Source SystemC with a Third Party Software"
Proceedings of Date 2001 pp.16-20, Munich, Germany, March 2001

© Copyright 2005 JEITA, All rights reserved



22

【15】 DATE 2001

タイトル	SystemCSV : An Extension of SystemC for Mixed Multi-Level Communication Modeling and Interface-Based System Design																																			
著者	Robert Siegmund, Dietmar Müller																																			
所属	Chemnitz University of Technology Professorship Circuit and Systems Design																																			
概要	インターフェースを導入して、通信と機能記述を分離。抽象度の異なる通信モデルを接続。																																			
内容	<p>通信と機能記述を分離するという概念は、多言語でも導入されている。ここでは、SystemCにインターフェースを導入したSystemCSVを提案する。</p> <p>SC_INTERFACE : インターフェース インターフェース・アイテム ・トランザクション・アイテム : 双方向トランザクション ・メッセージ・アイテム : 単方向トランザクション ・PHYMAPアイテム : 物理信号マッピング</p> <p>COMPOSITION : 結合 SERIAL : 逐次 PARALLEL : 並列 REPEAT : 繰り返し SELECT : 常に1つだけ選択 composite() decompose()</p> <p>【結果】 無線スマートカードをモデル化。</p> <table border="1"> <thead> <tr> <th colspan="7">抽象度</th> </tr> </thead> <tbody> <tr> <td>転送</td> <td>ABM</td> <td>ABM</td> <td>ABM</td> <td>RTL</td> <td>RTL</td> <td></td> </tr> <tr> <td>読み書き</td> <td>ABM</td> <td>ABM</td> <td>RTL</td> <td>RTL</td> <td>RTL</td> <td></td> </tr> <tr> <td>インターフェース</td> <td>FL</td> <td>SL</td> <td>SL</td> <td>SL</td> <td>SL</td> <td></td> </tr> <tr> <td>時間/フレーム</td> <td>1.3ms</td> <td>0.10s</td> <td>0.15s</td> <td>0.3s</td> <td>0.21s</td> <td></td> </tr> </tbody> </table> <p>ABM : Abstraction Model RTL : Register Transfer Level FL : Frame Level SL : Signal Level 時間は1000フレームの平均</p>	抽象度							転送	ABM	ABM	ABM	RTL	RTL		読み書き	ABM	ABM	RTL	RTL	RTL		インターフェース	FL	SL	SL	SL	SL		時間/フレーム	1.3ms	0.10s	0.15s	0.3s	0.21s	
抽象度																																				
転送	ABM	ABM	ABM	RTL	RTL																															
読み書き	ABM	ABM	RTL	RTL	RTL																															
インターフェース	FL	SL	SL	SL	SL																															
時間/フレーム	1.3ms	0.10s	0.15s	0.3s	0.21s																															
分類	モデリング																																			

出展: Robert Siegmund etc. "SystemCSV: An Extension of SystemC for Mixed Multi-Level Communication Modeling and Interface-Based System Design"
 Proceedings of Date 2001 pp.26-33, Munich, Germany, March 2001

© Copyright 2005 JEITA, All rights reserved



23

【16】 DATE 2001

タイトル	The Simulation Semantics of SystemC
著者	Wolfgang , Juergen Ruf, Dirk Hoffmann, Joachim Gerlach, Thomas Kropf, Wolfgang Rosenstiehl
所属	C-LAB/Paderborn University University of Tuebingen
概要	Watching、signal代入、waitなどを含むSystemC 1.0のシミュレーション・セマンティックの紹介。
内容	<p>SystemCは新しいシステムレベル言語であるが、シミュレーション・合成・フォーマル検証などのアプリケーションにはSystemCの正確なセマンティックが必要です。</p> <p>【ABM】 Abstract State Machinesは言語の動きを示すのに適しています。 この論文ではABMによってSystemCシミュレーション・カーネルの動きを表現します。全15のABMで動作を表しています。</p> <pre> graph TD A[イベント生成] --> B[プロセス実行] B --> C[CTHREADスケジュール] C --> D[信号更新] D --> E[イベントチェック] E --> B E --> F[CTREAD実行] F --> G[時間更新] G --> H[クロック更新] H --> B E -- イベントなし --> F </pre>
分類	言語

出展: Wolfgang etc. "The Simulation Semantics of SystemC"
 Proceedings of Date 2001 pp.64-70, Munich, Germany, March 2001

© Copyright 2005 JEITA, All rights reserved



24

【17】 DATE 2001

タイトル	An Efficient Architecture Model for Systematic Design of Application-Specific Multiprocessor SoC
著者	Amer Baghdadi Damien Lyonnard Nacer-E. Zergainoh Ahmed A. Jerraya
所属	TIMA Laboratory
概要	マルチプロセッサ・アプリケーションの設計手法の提案。汎用アーキテクチャモデルによって、接合性・柔軟性・可測性の高い手法の提案。
内容	<p>現在マルチプロセッサシステムを設計するためのアプローチは数多く提案されているが、それらは汎用性に欠けている。ここでは、接合性・柔軟性・可測性の高い汎用アーキテクチャモデルを用いた手法を提案します。</p> <p>【検証】 アプリケーション(システム記述)やHW/SWシミュレーションはSystemC。</p> <p>【汎用モデル】 通信I/Fやプロセッサモデルなど汎用モデルを使用。</p> <p>【アーキテクチャ生成】 パラメータ設定からアーキテクチャモデルを生成</p> <p>【結果】 IS95システムのアーキテクチャ設計トータル42時間で終了。</p>
分類	Co-Design/Co-Simulation

出展: Amer Baghdadi Damien Lyonnard Nacer etc. "An Efficient Architecture Model for Systematic Design of Application-Specific Multiprocessor SoC" Proceedings of Date 2001 pp.55-63, Munich, Germany, March 2001

© Copyright 2005 JEITA, All rights reserved



25

【18】 DATE 2001

タイトル	A Model for Describing Communication between Aggregate Objects in the Specification and Design of Embedded Systems
著者	Kjetil Svarstad, Gabriela Nicolescu, Ahmed A. Jerraya
所属	SINTEF Telecom and Informatics Signal Processing and Systems Design group TIMA Laboratory, SLS group
概要	近年、組み込みシステムの設計において設計の抽象度を上げることが重要になってきているが、同時に通信の重要度を上げることも重要である。この論文では、SystemCの拡張として通信モデルを実装。
内容	<p>この論文では、通信の抽象度をService、Message、Driver、RTLと4段階に分ける。RTLは信号レベルの通信、Driverは信号レベルへの相互接続、MessageではRPCを用いたチャネル通信、Serviceにおいては、接続方法やタイミングなども抽象化した通信と定義する。</p> <p>【試行】 ここでは、UMLで使用を定義し、それに基づいて機能記述、抽象度の高いServiceレベルの通信モデルを作成、さらに動作レベルのピンレベルモデルを作成してみた。</p> <p>【結果】 名前による接続を用いたServiceレベルのモデルは、機能モデルと比較して2-3倍の記述量となった。これは、C++のテンプレートを使用していることとC++で記述するためのクラスメンバなどの宣言の部分があるため、純粋な記述量の増加量ではないとの考察。 また、動作レベルの記述は、その3-4倍の記述量となった。</p>
分類	言語

出展: Keitil Svarstad etc. "A Model for Describing Communication between Aggregate Objects in the Specification and Design of Embedded Systems" Proceedings of Date 2001 pp.77-85, Munich, Germany, March 2001

© Copyright 2005 JEITA, All rights reserved



26

【19】 DATE 2001

タイトル	Network Processors: A Perspective on Market Requirements, Processor Architectures and Embedded S/W Tools
著者	Pierre G. Paulin, Faraydon Karim, Paul Bromley
所属	STMicroelectronics
概要	STMicroelectronicsの新しいネットワークプロセッサの構造と開発環境の紹介。
内容	<p>ネットワークプロセッサにおいては、外部からのパケットの処理に主眼がおかれています。一般に1つのパケットが到着したときに他のパケットも到着している場合は少ないので、新しいSTのネットワークプロセッサはデータキャッシュユーティリティの多くをなくしています。その代わりに、メモリ・コプロセッサ・アクセラレータ間で高速にデータをやりとりできるように、レイテンシを少なくするため、パイプライン・並列処理・マルチスレッド構造が取り入れられています。</p> <p>【組み込みソフト開発環境】 FlexWare ・FlexCC : Cコンパイラ ・FlexSim : ISS ・FlexGsb : ソースデバッガ ・FlexPerf : パフォーマンス解析ツール</p> <p>【Cコンパイラ】 【結果】 ネットワークパケットの処理時間 OC-192 52ns OC-768 13ns</p>
分類	モデリング

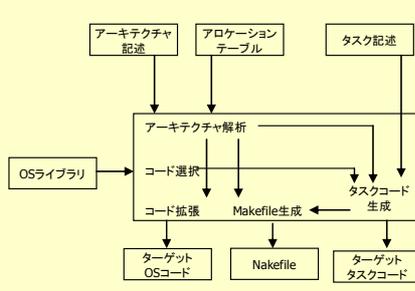
出展: Pierre G etc. "Network Processors: A Perspective on Market Requirements, Processor Architectures and Embedded S/W Tools"
 Proceedings of Date 2001 pp.420-429, Munich, Germany, March 2001

© Copyright 2005 JEITA, All rights reserved



27

【20】 DATE 2001

タイトル	Automatic Generation and Targeting of Application Specific Operating Systems and Embedded Systems Software
著者	Lovic Gauthier, Sungjoo Yoo, Ahmed A. Jerraya
所属	SLS Group, TIMA Laboratory
概要	アプリケーションに特化したOSとアプリケーション・ソフトウェアの自動生成。
内容	<p>アプリケーションに合わせてOSとアプリケーションSWを記述するのは大変な作業です。この論文では、非常に小さく柔軟性のあるOSカーネルに、階層構造を含むシステム表記、モジュール間の通信、モジュールの動作、モジュールのプライオリティなどの条件から、アプリケーションに合わせたOSとSWコードを自動生成するテクノロジーを紹介します。</p> <p>【結果】 1245行のSystemCからなるToken Ringシステムに適用。 3つの68000プロセッサを適用したところ、制御用のSWと残りの2つのプロセッサにタスクが均等に分配された。 生成されたコードは797行のC(OS部分)と1.86KB及び1.62KBのアセンブリコードがそれぞれ生成。 この手法により、非常に柔軟にシステム構成を変更することが可能となる。</p> 
分類	Co-Design/Co-Simulation

出展: Lovic Gauthier etc. "Automatic Generation and Targeting of Application Specific Operating Systems and Embedded Systems Software"
 Proceedings of Date 2001 pp.679-685, Munich, Germany, March 2001

© Copyright 2005 JEITA, All rights reserved



28

【21】 DATE 2001

タイトル	Simulation-Guided Property Checking Based on Multi-Valued AR-Automata
著者	Jürgen Ruf, Dirk W. Hoffmann, Thomas Kropf, Wolfgang Rosenstiel
所属	Institute of Computer Engineering, University of Tübingen
概要	アサーションベースの高速プロパティチェック手法
内容	<p>既存のプロパティチェックはテストベンチに埋め込むようになっているが、この論文ではHDLコード内に埋め込む方法を検討する。また、メモリ使用量は考慮せず、高速に動作するプロパティチェックに関して検討する。</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>アサーション</p> <p>↓</p> <p>FLTL (Finite Linear Temporal Logic)</p> <p>↓</p> <p>AR-Automata</p> <p>【結果】 10~100セル、50000~100000サイクルシミュレーションで、5~30%程度のオーバーヘッド。</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p>SystemC記述 ... Sc_assert("F[20] req"); ...</p> <div style="display: flex; justify-content: space-between;"> <div style="border: 1px solid black; padding: 2px;">シミュレーション カーネル</div> <div style="border: 1px solid black; padding: 2px;">AR-automata チェッカ</div> </div> <div style="display: flex; justify-content: center; margin-top: 10px;"> <div style="border: 1px solid black; border-radius: 50%; padding: 5px; text-align: center;">automata データベース</div> <div style="margin: 0 10px;">←</div> <div style="border: 1px solid black; padding: 2px;">AR-automata 合成</div> <div style="margin: 0 10px;">→</div> </div> </div> </div>
分類	検証

出展: Jürgen Ruf etc. "Simulation-Guided Property Checking Based on Multi-Valued AR-Automata"
Proceedings of Date 2001 pp.742-748, Munich, Germany, March 2001

© Copyright 2005 JEITA, All rights reserved



29

【22】 DATE 2001

タイトル	A Framework for Fast Hardware-Software Co-simulation								
著者	Andreas Hoffmann, Tim Kogel, Heinrich Meyr								
所属	Integrated Signal Processing Systems (ISS), RWTH Aachen								
概要	高速HW/SWシミュレーションのためのフレームワークに関する手法。ISS、アセンブラ、リンカ生成								
内容	<p>LISAによるソフトウェア記述とSystemCによるハードウェア記述により、早期に高速なHW/SWシミュレーション環境を構築。また、LISAからソフトウェアシミュレータを含む設計環境を生成。</p> <p>【ハードウェアシミュレーション】 SystemC内でハードウェアシミュレーションをする場合には、HDLにアブストラクト階層を持たせて接続。</p> <p>【仮想プロトタイプ】 C++とHDLなどの接続には、アブストラクト・インターフェースでC++インターフェースをピンに変換、HDLシミュレータのCインターフェースによって接続。</p> <p>【LISA】 プログラマブル・アーキテクチャ記述言語</p> <div style="display: flex; justify-content: center; align-items: center; margin-top: 10px;"> <div style="border: 1px solid black; padding: 5px; margin-right: 20px;">LISA</div> <div style="display: flex; flex-direction: column; align-items: center;"> <div style="width: 1px; height: 10px; background-color: black; margin-bottom: 5px;"></div> <div style="border: 1px solid black; padding: 5px; margin-right: 20px;">ISS</div> </div> <div style="display: flex; flex-direction: column; align-items: center;"> <div style="width: 1px; height: 10px; background-color: black; margin-bottom: 5px;"></div> <div style="border: 1px solid black; padding: 5px;">アセンブラ・リンカ</div> </div> </div> <div style="margin-top: 20px;"> <p>【結果】</p> <table border="1"> <thead> <tr> <th>モデル</th> <th>速度 [kCycle/s]</th> </tr> </thead> <tbody> <tr> <td>VHDL (位相精度)</td> <td>0.27</td> </tr> <tr> <td>SystemC (マクロサイクル精度)</td> <td>98.5</td> </tr> <tr> <td>LISA (ARM7) (位相精度)</td> <td>4500</td> </tr> </tbody> </table> </div>	モデル	速度 [kCycle/s]	VHDL (位相精度)	0.27	SystemC (マクロサイクル精度)	98.5	LISA (ARM7) (位相精度)	4500
モデル	速度 [kCycle/s]								
VHDL (位相精度)	0.27								
SystemC (マクロサイクル精度)	98.5								
LISA (ARM7) (位相精度)	4500								
分類	Co-Design/Co-Simulation								

出展: Andreas etc. "A Framework for Fast Hardware-Software Co-simulation"
Proceedings of Date 2001 pp.760-765, Munich, Germany, March 2001

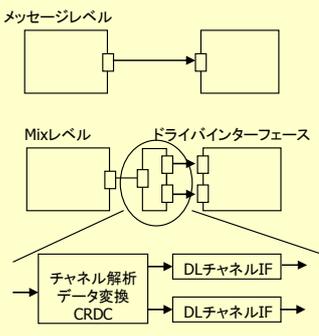
© Copyright 2005 JEITA, All rights reserved



30

【23】 DATE 2001

タイトル	Mixed-Level Cosimulation for Fine Gradual Refinement of Communication in SoC Design
著者	Gabriela Nicolescu, Sungjoo Yoo, Ahmed A. Jerraya
所属	SLS Group, TIMA Laboratory
概要	プロトコルの決まってない通信からプロトコルの決まった通信へのリファインメント手順
内容	<p>3つのレベルメッセージレベル、ドライバレベル、RTLを定義。メッセージレベルからドライバレベルへのリファインメントについて検討。</p> <p>【MLとDLの接続】 MLとDLの接続には、インターフェースを変換するモジュールアダプタを挿入。アダプタはチャンネルを解析してデータを変換するモジュールとDLのIFに接続するモジュールで構成。</p> <p>【ケース・スタディ】 IS-95システムをSDLを用いてMLでモデル化。DLにはSystemCを使用。送信側はSDLのまま受信用側だけSystemC DLとし、IPCベースのCoSimulationで混在シミュレーションを実行。データを正しく受け取ることができ、音声波形を表示できたことを確認。</p>
分類	モデリング



出展: Gabriela Nicolescu et al. "Mixed-Level Cosimulation for Fine Gradual Refinement of Communication in SoC Design" Proceedings of Date 2001 pp.754-759, Munich, Germany, March 2001

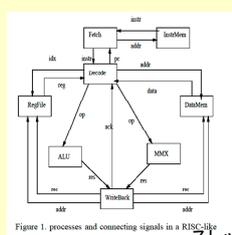
© Copyright 2005 JEITA, All rights reserved



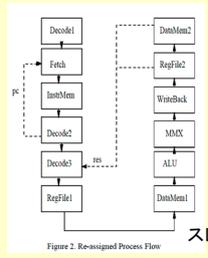
31

【24】 DATE 2002

タイトル	Automated Concurrency Re-Assignment in High Level System Models for Efficient System-Level Simulation
著者	N. Savoieu, S. Shukla, R. Gupta
所属	UCI
概要	C++言語でのシミュレーション高速化(Threadの最適化)に関するものをSystemCを例に説明
内容	<p>Thread記述に関してPOSIX準拠のマルチスレッドプログラミングで高速処理ができるように sc_(c)thread/sc_methodのスレッド処理を自動的に最適化する。利点は、SystemCでモデルを書く人(特にハードウェア設計者やシステム設計者)がソフトウェア技術であるマルチスレッドプログラミングの知識がなくても高速なシミュレーションが可能のようにソースコードを自動的に変更すること。効果としては、2倍程度まで高速化できた。</p>
分類	言語



自動スレッド最適化



出展: N. Savoieu et al. "Automated Concurrency Re-Assignment in High Level System Models for Efficient System-Level Simulation", Proceedings of Date 2002 pp.875-881, Paris, France, March 2002

© Copyright 2005 JEITA, All rights reserved



32

【25】 DATE 2002

タイトル	From System Specification To Layout: Seamless Top-Down Design Methods for Analog and Mixed-Signal Applications
著者	R. Sommer, I. Rugen-Herzig, E. Hennig, U. Gatti, P. Malcovati, F. Maloberti, K. Einwich, C. Clauss, P. Schwarz, G. Noessing
所属	Infinion, Siemens, FhG IIS
概要	SystemC2.0のAnalog・MixedSignalをモデリングするための拡張についての提案
内容	<p>アナログ、ミックスドシグナルのトップダウンフロー (ANASTASIA+プロジェクト) の中で、記述言語とシミュレーションモデルに関する提案。アナログ機能として線形システム (伝達関数など) を時間領域と周波数領域をモデル化するための SystemC2.0を拡張している</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Figure 9: Examples for SytemC MoC's and proposed extensions</p> <p>SC2.0の拡張提案</p> </div> <div style="text-align: center;"> <pre> SDF_MODULE(sc_gnd, pd0) { sdf_inport<double> INPUT; sc2_sdf_inport<sc2::ADMS_LITE> sdf_outport<double> OUTPUT; void init() { double wpsst; wpsst=2.0*M_PI*F0; A0(0)=1.41*wpsst; B(0)=K; } void sig_gnd() { if(ADMS_LITE) OUTPUT=LTF(A1,B1,S,Hc,Hd,INPUT); else OUTPUT=LTF(A0,B0,S,Hc,Hd,INPUT); } } </pre> <p>Figure 10: Transfer function embedded in dataflow block with event driven control input port</p> <p>伝達関数の記述</p> </div> <div style="text-align: center;"> <pre> elec_wire w1, w2, w3, w4, w5, tr; elec_gnd gnd; double Rp1=60.0, Rp2=40.0; double Cp=1e-12, Lp=1e-3; //parameter V = vsrc(w1.gnd,2.0); R rp1 (w2,w3,Rp1); R rp2 (w2,w3,Rp2); C cp (w2.gnd,Cp); L lp (w3,h1,Lp); CCVS i2v (w5.gnd,w1,w4,1.0); //Current Controlled Voltage </pre> <p>Figure 12: C++ network example</p> <p>RLC回路の記述</p> </div> </div>
分類	モデリング

出展: R. Sommer et al. "From System Specification To Layout: Seamless Top-Down Design Methods for Analog and Mixed-Signal Applications", Proceedings of Date 2002 pp.884-891, Paris, France, March 2002

© Copyright 2005 JEITA, All rights reserved

JEITA

33

【26】 DATE 2003

タイトル	Transaction-Level Models for AMBA Bus Architecture Using SystemC 2.0
著者	M. Caldari*, M. Conti*, M. Coppola**, S. Curaba**, L. Perialisi*, C. Turchetti*
所属	University of Ancona
概要	SystemC 2.0 を用いたトランザクションレベルAMBAバスの検討
内容	<p>SystemC 2.0を用いたトランザクションレベルでAMBA 2.0のAHB及びAPBバスをモデル化。シミュレーションの効率化のため、ダイナミックセンシビリティを用いてスLEEP動作を制御。</p> <p>【構造】 バスはsc_interfaceを継承したインターフェースとsc_channelを継承したクラスを実装。継承したクラス内にバス・アクセスIF用のメソッドを実装。</p> <p>【ステート形式】 通信手順はPSM (Program State Machine) と呼ばれるステートマシン形式で実装。</p> <p>【結果】 Ultra60を使用 AHB SystemC RTL 3 K cycle/s AHB SYSTEMC TLM 300 K cycle/s</p> <div style="text-align: center;"> </div>
分類	モデリング

出展: M. Caldari etc. "Transaction-Level Models for AMBA Bus Architecture Using SystemC 2.0" Designers forum of Date 2003 pp.26-31, Munich, Germany, March 2003

© Copyright 2005 JEITA, All rights reserved

JEITA

34

【27】 DATE 2003

タイトル	System-Level Power Analysis Methodology Applied to the AMBA AHB Bus																
著者	M. Caldari*, M. Conti*, M. Coppola**, P. Crippa*, S. Orcioni*, L. Perialisi*, C. Turchetti*																
所属	University of Ancona																
概要	AHBバスシステムに特化した省電力化に関するガイドラインの検討																
内容	<p>近年のSoCはIPの利用や省電力化が重要な要因となってきた。この論文ではIPを利用する環境における電力の解析や最適化について検討する。</p> <p>【汎用性】 電力解析モデルは汎用性を高くすることが重要。右図のようなタイプが考えられる。また電力解析しない場合はスキップできるようにする。</p> <p>【実装】 Activityクラスを定義し、ここからbit_change_count及びstore_activityメソッドを実行してデータを収集。</p> <p>【結果】</p> <table border="1"> <tr> <td>IDLE_HO_IDLE_HO</td> <td>96.5uJ</td> <td></td> </tr> <tr> <td>IDLE_HO_WRITE</td> <td>0.5uJ</td> <td></td> </tr> <tr> <td>READ_WRITE</td> <td></td> <td>417.7uJ</td> </tr> <tr> <td>READ_IDLE_HO</td> <td></td> <td>9.6uJ</td> </tr> <tr> <td>WRITE_READ</td> <td></td> <td>315.3uJ</td> </tr> </table>	IDLE_HO_IDLE_HO	96.5uJ		IDLE_HO_WRITE	0.5uJ		READ_WRITE		417.7uJ	READ_IDLE_HO		9.6uJ	WRITE_READ		315.3uJ	
	IDLE_HO_IDLE_HO	96.5uJ															
IDLE_HO_WRITE	0.5uJ																
READ_WRITE		417.7uJ															
READ_IDLE_HO		9.6uJ															
WRITE_READ		315.3uJ															
分類	性能解析																

出展: M. Caldari etc. "System-Level Power Analysis Methodology Applied to the AMBA AHB Bus" Designers forum of Date 2003 pp.32-39, Munich, Germany, March 2003

【28】 DATE 2003

タイトル	Exploring SW Performance using SoC Transaction-level Modeling	
著者	Imed Moussa, Thierry Grellier, Giang Nguyen	
所属	TNI-Valiosys, France	
概要	VISTAによるSystemC TLMを用いたSW開発環境の紹介	
内容	<p>SW設計者はサイクル精度のSoCモデルを必要としない。VISTA上でプラットフォームを入力、そこからSystemC 2.0 TLMのシミュレーションモデルを生成。SWモデルをリンクしてSW検証環境として使用。</p> <p>【アブストラクト通信】 チャンネルはシンプルバスとほぼ同じIFの汎用チャンネルに属性をつけて特定用途のバスに対応。</p> <p>【結果】 MPEG符号化モデルを作成。10000バスアクセス/sで実行。0.5実時間を20秒で実行。</p>	
	分類	モデリング

出展: Imed Moussa etc. "Exploring SW Performance using SoC Transaction-level Modeling" Designers forum of Date 2003 pp.120-125, Munich, Germany, March 2003

【29】 DATE 2003

タイトル	Systematic Embedded Software Generation from SystemC	
著者	F. Herrera, H. Posadas, P. Sánchez, E. Villar	
所属	TEISA Dept., E.T.S.I. Industriales y Telecom., University of Cantabria	
概要	SystemCからのソフトウェアの生成	
内容	<p>SystemCシミュレーション・カーネルとRTOSの動きは非常に近いものである。並列動作とTHREADの制御がタスクの制御と同様と考えられる。このためSystemCの実行制御であるwaitやnotiryをRTOSの動作に置き換えることで、SystemCをRTOSモデルに置き換えることが可能となる。この論文では、SystemCのモジュールをRTOSに置き換える手順を考察する。</p> <p>【RTOS生成】 SC2RTOS.hファイルでSystemCのマクロやクラスをRTOS動作として定義。 HWとして動作する場合にはsystemc.hを使用し、SWとして動作する場合にはSC2RTOS.hを使用。</p> <p>【結果】 SystemCでシステム記述を行い、そのうちSWにしたい部分だけをSC2RTOS.hを用いてRTOSモデルとして使用。記述の変更なしでHW/.SWの切り替えが可能となった。</p>	
分類	Co-Design/Co-Simulation	

出展: F. Herrera etc. "Systematic Embedded Software Generation from SystemC" Proceedings of Date 2003 pp.142-149, Munich, Germany, March 2003

© Copyright 2005 JEITA, All rights reserved



37

【30】 Stanford Univ.

タイトル	× pipesCompiler: A tool for instantiating application specific Networks on Chip	
著者	Antoine Jalabert (LETI-DSIS), Srinivasan Murali, Luca Benini (Univ. of Bologna), Giovanni De Micheli	
所属	Stanford University	
概要	特定アプリ用のNetworks on Chip(NoC)を構築するツール。	
内容	<p>将来のシステムのコンポーネント間はBUSIによる接続ではなく、ネットワークによる接続となる。これをNetworks on Chip (NoC)アーキテクチャと呼ぶ。コンポーネント間の接続はhomogeneousではなく、アプリケーションに依存してheterogeneousである。このツールは、コンポーネントの接続情報から、Network InterfaceとSwitchを用いたheterogeneousな接続を生成する。Network InterfaceとSwitchはSystemCで記述され、ライブラリに登録されている。NoCアーキテクチャのブロック図とMPEG4 Decoderの生成されたフロアプランの例を示す。</p>	
分類	合成	

出典: A. Jalabert, S. Murali, L. Benini, G. De Micheli, "× pipesCompiler: A tool for instantiating application specific Networks on Chip", Stanford University

© Copyright 2005 JEITA, All rights reserved



38

【31】 Stanford Univ.

タイトル	Bandwidth-Constrained Mapping of Cores onto NoC Architectures
著者	Srinivasan Murali Giovanni De Micheli
所属	Stanford University
概要	バンド幅制約の下でコンポーネントをNetworks on Chip(NoC)アーキテクチャにマッピングするアルゴリズム。
内容	<p>プロセッサ、DSP、メモリー等のコンポーネント(コアと呼んでいる)をMesh-based NoCアーキテクチャにマッピングする際に重要なことは、コンポーネント間の通信のバンド幅制約を満たしつつ、通信の遅延が最小になるようにマッピングすることである。マルチメディア・アプリケーションでは特にバンド幅制約を満たすことが重要となっている。このようなマッピングを実現するアルゴリズムを提案。SystemCで記述されたDSP Filterを例題とし、サイクル精度でシミュレーションを行い、アルゴリズムが有効であることを示した。</p>
分類	合成

出典: S. Murali, G. De Micheli, "Bandwidth-Constrained Mapping of Cores onto NoC Architectures", Stanford University

© Copyright 2005 JEITA, All rights reserved

JEITA

39

【32】 Stanford Univ.

タイトル	Methodology for Hardware/Software Co-verification in C/C++
著者	Luc Séméria Abhijit Ghosh (Synopsys Inc.)
所属	Stanford University
概要	プロセッサのBus Functional Model。
内容	<p>HW-SW Co-Simulationの目的はアーキテクチャ検証と性能確認。アーキテクチャ探索(アンタイムド)のためモデルは高速であることが要求される。また、アーキテクチャ決定後、HWはRTL記述にリファインされ、SWはターゲット・プロセッサ用のコードにリファインされる。このモデルで性能検証(サイクル精度)をするために、HWはRTL、プロセッサ・モデルはISSが使用される。これら2つの抽象度で使うことのできるプロセッサのBFM(Bus Functional Model)を作成した。SystemCで実装。</p> <p>BFMの特徴は以下のとおり。</p> <ul style="list-style-type: none"> • Memory Mapped I/O • 割込み処理登録機能 • ブロッキング/ノンブロッキング・アクセス • メモリを内部接続可能 <p>BFMモデルとHW、SWとの接続関係を下図に示す。</p>
分類	モデリング

出典: L. Séméria, A. Ghosh, "Methodology for Hardware/Software Co-verification in C/C++", Stanford University

© Copyright 2005 JEITA, All rights reserved

JEITA

40

【33】 Univ. of Tübingen

タイトル	Transaction Level Verification and Coverage Metrics by Means of Symbolic Simulation		
著者	Prakash M. Peranandam, Roland J. Weiss, Jurgen Ruf, and Thomas Kropf		
所属	Department of Computer Engineering University of Tübingen		
概要	シンボリックシミュレーションにおけるトランザクション検証とカバレッジ		
内容	<p>検証の課題である機能のカバレッジの計算方法に対する提案である。トランザクション・レベルでのモデリングにおいて、トランザクションを1機能に対応してモデリングされているとすると、そのトランザクションに対するカバレッジによって機能カバレッジを計算することができる。この論文では、TLMをMSM(Message Sequences Model)のシンボリック・シミュレーションとして評価する。</p>		
	<p>右図ベンディングマシンのトランザクションでは、 get_coffee, get_money, put_coffee get_money, get_coffee, put_coffee はどちらでも可である。</p>		<p>Linear-Time Temporal Logic(LTL)の論文による表記を用いて、入力メッセージの順序を定義し、それらのカバレッジを計算することで、機能カバレッジを測定している。</p>
分類	検証		

出展: Prakash M. Peranandam etc. "Transaction Level Verification and Coverage Metrics by Means of Symbolic Simulation" Department of Computer Engineering University of Tübingen

© Copyright 2005 JEITA, All rights reserved
JEITA
41

【34】 Univ. California, Irvine

タイトル	Typing Abstractions and Management in a Component Framework		
著者	Frederic J. Doucet Sandeep K. Shukla (Virginia Tech) Rajesh K. Gupta (University of California, San Diego)		
所属	University of California, Irvine		
概要	コンポーネントのポートの型をすべて明示することなくアーキテクチャを構成できる環境。		
内容	<p>アーキテクチャ探索を行う際は、HWとSWをプログラミング言語を使ってモデリングする。アーキテクチャの含まれるコンポーネントのすべてのポートの型 (unsigned int, 32ビットのビットベクタ等) を指定する必要の無い、フレームワークを提供する。型指定の無いコンポーネントについては、ライブラリから適切な型を持つコンポーネントのC++オブジェクトが選ばれる。この機構をC++の集約 (aggregation) とComponent Integration Language(スクリプト言語) を使って実現している。C++オブジェクトの持つ情報をスクリプト言語に受け渡すために、独自のBALBOA Interface Description Languageを利用している。C++オブジェクトはSystemCで記述している。ライブラリに登録されているコンポーネントの例としては、2つの実装方法を持つ加算器 (浮動小数点型と16ビット整数型)。フレームワークの概念図を下に示す。</p>		
分類	合成		

出典: F. J. Doucet, S. K. Shukla, R. K. Gupta, "Typing Abstractions and Management in a Component Framework", Univ. of California, Irvine

© Copyright 2005 JEITA, All rights reserved
JEITA
42

【35】 Univ. California, Irvine

タイトル	Transaction level modeling of SoC with SystemC 2.0	
著者	Sudeep Pasricha	
所属	STMicroelectronics Ltd	
概要	TLMを使用したeSW開発、性能解析	
内容	<p>SystemC2.0でモデリングしたトランザクション・レベル・モデル(TLM)を使用して、以下に適用</p> <ol style="list-style-type: none"> TLMsによるSW開発 <ul style="list-style-type: none"> ・100KHzの高速シミュレーション ・アーキテクチャが決まれば、RTLを待たずに直ぐにSW開発が開始可能 (例:RTL設計完了の6ヶ月前からSW開発を開始) TLMsによるアーキテクチャ探索 <ul style="list-style-type: none"> ・タイムドのTLMsを使用したアーキテクチャ探索 ・シミュレーション時間が短いので、いろいろなアーキテクチャを試せる ・デザインの変更が比較的容易なので、HW/SWのトレード・オフ解析なども可能 ・RTLとの精度比較では、15%以内のずれに収まった 	<p><EASY platform with ISS/eSW></p>
分類	Co-Design/Co-Simulation	

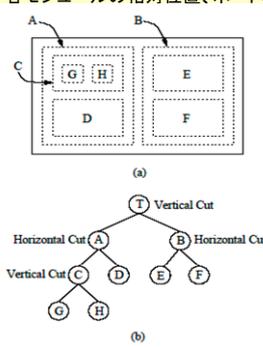
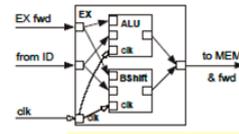
出典: Sudeep Pasricha, etc. "Transaction level modeling of SoC with SystemC 2.0", Univ. of California, Irvine, 2002

【36】 Univ. California, Irvine

タイトル	Concurrency in System Level Design: Conflict between Simulation and Synthesis Goal	
著者	Nick Savoivu, Sandeep K. Shukla, Rajesh K. Gupta	
所属	UC Irvine	
概要	合成用Cモデル記述のシミュレーション速度の向上	
内容	<ul style="list-style-type: none"> ・合成用に記述されたSystemCモデルを、マルチ・プロセッサ上でより高速にシミュレーションできるモデルへ自動変換するツールを紹介 ・各スレッド記述を単純に各プロセッサに割り当てただけでは、シミュレーション速度はむしろ遅くなる。 ・そこで、合成用SystemCデザインを読み込んで解析し、APU (Atomic Process Unit) へ分解する。それぞれのAPUをスケジューリングし、マルチ・プロセッサ上でより効率的に並列動作可能なモデルに変換することにより、シミュレーション速度の高速化を実現する。 ・上記を自動化するツールを開発 ・簡単なDMAデータ転送のデザインを使って実験し、シミュレーション速度が速くなることを確認 	<p><変換フロー></p>
分類	ツール	

出典: Nick Savoivu, etc. "Concurrency in System Level Design: Conflict between Simulation and Synthesis Goal", Univ. of California, Irvine, 2002

【37】 Univ. California, Irvine

タイトル	Structural Design Composition for C++ Hardware Models
著者	Vivek Sinha, Frederic Doucet, Chuck Siska, Rajesh Gupta,
所属	UC Irvine
概要	SystemC上でブロック配置情報を付加する仕組み
内容	<p><ICSP (Incidence Component Structure Project) ></p> <ul style="list-style-type: none"> ・構造レイアウト、配線、フロアプラン情報等を記述可能 ・高い抽象度から上記のような情報を検討していくことにより、後工程での配線の混雑を減らし、配線遅延やブロック間のタイミング要求を満たせるようになる ・各モジュールの相対位置、ポートの場所を追記できるよう、SystemCのクラス・ライブラリから拡張  <pre> struct stage_ex : public icsp_module { alu alu; // sub-modules: both EX stage functional bshift bsh; icsp_import< bool > clk; icsp_import< t_id_ex_reg > id_ex_in; icsp_import< t_ex_mem_reg > ex_mem_in; icsp_ouport< t_ex_mem_reg > ex_mem_out; // constructor. stage_ex(const char * NAME) // Initialize base module: // 1. rectangle partition : icsp_module(name, ICSP_SPLIT_VERTICALLY, icsp_corder(icsp_comp(alu), icsp_comp(bsh)), // 2. Port placement. icsp_porder(icsp_ppair(clk, ICSP_LOWER_LEFT), icsp_ppair(id_ex_in, ICSP_UPPER_LEFT), icsp_ppair(ex_mem_in, ICSP_MID_LEFT), icsp_ppair(ex_mem_out, ICSP_UPPER_RIGHT)) ... </pre> 
分類	言語

出典: Vivek Sinha, etc. "Structural Design Composition for C++ Hardware Models", Univ. of California, Irvine, 2001

© Copyright 2005 JEITA, All rights reserved

JEITA

45

【38】 Univ. California, Irvine

タイトル	Interfacing Hardware and Software Using C++ Class Libraries
著者	Dinesh Ramanatham, Rajesh Gupta Ray Roth
所属	UC Irvine CynApps Inc
概要	SWとHWのインターフェース
内容	<ul style="list-style-type: none"> ・SystemCを使って、ソフトで実行していた処理の一部をHWモデル化する。 ・HW/SW共に同じC言語で記述することで、両者間の移行を容易にする ・その際に、HWと通信する手段として、ハンドシェイク(Request/Done)などによるハードウェア・ドライバを作成する。 ・HWの処理部分はSWコードをそのまま流用し、場合によってはパイプライン化を行う。 <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px; width: 30%;"> <p><SWコード></p> <pre> int main() { data_structure data; F1 (&data); F2 (&data); F3 (&data); } </pre> </div> <div style="border: 1px solid black; padding: 5px; width: 30%;"> <p><HW呼び出し></p> <pre> int sc_main() { sc_signal<bool> request, done; data_type data; sc_clock clock("clk", 1, 0.5, 0.0); hardware = new hardware("h", clock.pos()); request, done, &data); F1(&data); request.write(1); do { sc_run(1); } while(!done); request = 0; F3(&data); } </pre> </div> <div style="border: 1px solid black; padding: 5px; width: 30%;"> <p><HWコード></p> <pre> class hardware : public sc_sync { public hardware(char *h, sc_clock_edge &m_clk, const sc_signal &m_req, sc_signal &m_done, data_structure &m_data) : sc_sync(h, m_clk) { void entry { if (m_req) { F2(m_data); m_done.write(1); } } }; </pre> </div> </div>
分類	Co-Design/Co-Simulation

出典: Dinesh Ramanatham, etc. "Interfacing Hardware and Software Using C++ Class Libraries", Univ. of California, Irvine, 2000

© Copyright 2005 JEITA, All rights reserved

JEITA

46

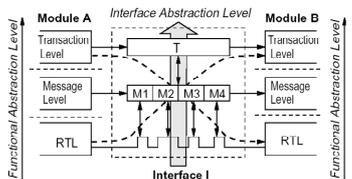
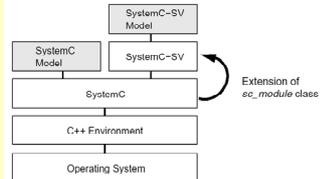
【39】 Univ. California, Irvine

タイトル	Transaction Level Modeling in System Level Design		
著者	Lukai Cai, Deniel Gajski		
所属	UC Irvine		
概要	トランザクション・レベル・モデリングの定義		
内容	<p>・4種類のトランザクション・レベルを定義(右図のグレー部分) <応用例> 1. システム・リファインメント(1 → 2 → 3 → 4 → 6) 2. アーキテクチャ探索(3 → 4 or 5 → 3ヘフィード・バック) 3. IP評価(3 → 5) <実適用> 1. 新規デザイン 仕様モデルから、システム・リファインメントを行う。 この時作成した各PEモデルは、ライブラリとして登録しておく。 2. 既存デザインのアップ・グレート(プラットフォーム・ベース) 仕様モデルを作成後、既存のデザインを再利用し、追加機能部分だけ見積もり値を使ってシミュレーション。既存部分は正確な値がわかっているため、精度は高くなる。</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>1. Specification model</p> <p>2. PE-assembly model</p> <p>3. Bus-arbitration model</p> </div> <div style="text-align: center;"> <p>4. Time-accurate communication</p> <p>5. Cycle-accurate computation</p> </div> </div> <p style="text-align: center;">6. Implementation model</p> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="text-align: center;"> <p>PE-assembly model</p> </div> <div style="text-align: center;"> <p>Bus-arbitration Model</p> </div> <div style="text-align: center;"> <p>Time-accurate Communication</p> </div> <div style="text-align: center;"> <p>Cycle-accurate computation</p> </div> </div>		
分類	モデリング		

【40】 Univ. California, Irvine

タイトル	RTOS Scheduling in Transaction Level Models																																
著者	Haobo Yu, Andreas Gerstlauer, Daniel Gajski																																
所属	UC Irvine																																
概要	トランザクション・モデルのRTOSによるスケジューリング																																
内容	<p>・RTOSの機能を提供するモデルを作成し、各PEをSWタスクとして実装 ・スケジューリングの方法を変えてシミュレーションし、最適なスケジューリング方法を決定 ・ポコーダ・モデルで、ラウンド・ロビン、エンコーダ>デコーダ、デコーダ>エンコーダのスケジューリングで実験 ・デコーダ>エンコーダのスケジューリングを採用</p> <div style="display: flex; justify-content: space-around;"> <div style="width: 45%;"> </div> <div style="width: 45%;"> </div> </div> <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th></th> <th>Lines of code</th> <th>Sim. time</th> <th>Context switches</th> <th>Transcoding delay</th> </tr> </thead> <tbody> <tr> <td>Unsched.</td> <td>11,313</td> <td>27.3s</td> <td>0</td> <td>9.7ms</td> </tr> <tr> <td>Roundrobin</td> <td>13,343</td> <td>28.6s</td> <td>3262</td> <td>10.29ms</td> </tr> <tr> <td>Encod>decod</td> <td>13,356</td> <td>28.9s</td> <td>980</td> <td>11.34ms</td> </tr> <tr> <td>Decod>encod</td> <td>13,356</td> <td>28.5s</td> <td>327</td> <td>10.30ms</td> </tr> <tr> <td>Impl.</td> <td>79,096</td> <td>5h</td> <td>327</td> <td>11.7ms</td> </tr> </tbody> </table>				Lines of code	Sim. time	Context switches	Transcoding delay	Unsched.	11,313	27.3s	0	9.7ms	Roundrobin	13,343	28.6s	3262	10.29ms	Encod>decod	13,356	28.9s	980	11.34ms	Decod>encod	13,356	28.5s	327	10.30ms	Impl.	79,096	5h	327	11.7ms
	Lines of code	Sim. time	Context switches	Transcoding delay																													
Unsched.	11,313	27.3s	0	9.7ms																													
Roundrobin	13,343	28.6s	3262	10.29ms																													
Encod>decod	13,356	28.9s	980	11.34ms																													
Decod>encod	13,356	28.5s	327	10.30ms																													
Impl.	79,096	5h	327	11.7ms																													
分類	言語																																

【41】 ESCUG4 2001

タイトル	Efficient Modeling and Simulation of Data Communication Protocols in Communication-oriented Designs using the SystemC-SV Extension
著者	Robert Siegmund, Dietmar Mueller
所属	Chemnitz University of Technology, Chemnitz, Germany
概要	SystemC-SV という言語拡張により、抽象度の異なるインターフェースを持つモジュールを自動的に接続し、シミュレーションを行う環境を提供する。
内容	<p>SV_INTERFACE, SV_TRANSACTION, SV_MESSAGE, SV_COMPOSITION 等のキーワードを拡張し、インターフェース定義を行う。各モジュールからはそれら呼び出して通信を行う。抽象度が異なる場合には、定義に従い自動的に COMPOSITION/DECOMPOSITIONを行って通信する。</p> <p>これらのCOMPOSITION/DECOMPOSITION回路は合成可能なSystemCコードで生成可能なので、真のインターフェース合成と言える。</p> <p>拡張ライブラリを被せる形なので、既存デザインとの共存も可能。</p>  
分類	言語

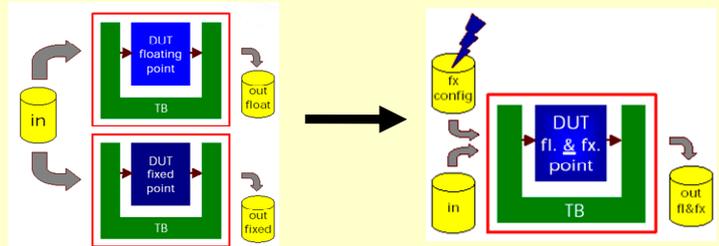
出展: ESCUG(European SystemC Users Group Meeting) Oct.2001

© Copyright 2005 JEITA, All rights reserved

JEITA

49

【42】 ESCUG5 2002

タイトル	A Method for the Development of Combined Floating- and Fixed-Point SystemC Models
著者	Yves Vanderperren
所属	Alcatel
概要	fx_doubleというクラスを新たに作成し、浮動小数点モデルと固定小数点モデルに対して同じソースコードで済みます。固定小数点条件はプログラムの外部から与える。
内容	<p>通常、固定小数点モデル化では、浮動小数点モデルで記述されたプログラム中のdoubleをsc_fixedに書き換える。しかしこの書き換えが面倒なので、新たにfx_doubleというクラスを作成し、浮動小数点、固定小数点ともにこのfx_doubleを使用。これにより、ソースコードを一元化できる。なお、固定小数点モデルの整数部桁数等の条件は、プログラムの外部から与える。</p> 
分類	モデリング

出展: ESCUG(European SystemC Users Group Meeting) March.2002

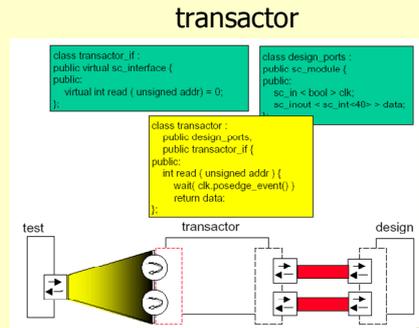
© Copyright 2005 JEITA, All rights reserved

JEITA

50

【43】 ESCUG7 2003

タイトル	The SystemC Verification Standard (SCV)
著者	Stuart Swan
所属	Cadence
概要	SCV(SystemC Verification standard)1.0Iに関する簡単な紹介。これはSCV1.0のインストールパッケージにも含まれているスライド。
内容	<p>SCV1.0でできることは以下の通り。</p> <ol style="list-style-type: none"> 1.トランザクションの宣言および記録 2.トランザクション・レベルからピン・レベルへの変換 (transactor) <ul style="list-style-type: none"> → これはSystemCのインターフェースとポートを使って従来手法により実装する 3.ランダムパターン生成 <p>Veraや'e'言語の持つ多くの特徴はSCVではなく、C++やSystemCの機能で実現可能。</p>
分類	検証



出展: ESCUG(European SystemC Users Group Meeting) March.2003

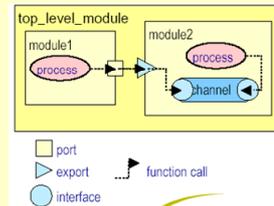
© Copyright 2005 JEITA, All rights reserved

JEITA

51

【44】 ESCUG9 2004

タイトル	SystemC 2.1 Overview
著者	Stuart Swan
所属	OSCI
概要	SystemC 2.1の概要。SystemC 2.1でエンハンス項目は、IPモジュール化の容易性向上 (sc_argcv()等の追加)、TLM(Transaction Level Modeling)のしやすさ向上 (sc_export等の追加)、その他使いやすさの向上 (.read()無しアクセス、異なる型の接続、パートセレクトの扱い等)。
内容	<p>SystemC 2.1でのエンハンス項目についての説明。</p> <ol style="list-style-type: none"> 1. IPモジュール化の容易性向上 <ul style="list-style-type: none"> ・新規関数sc_argc(), sc_argv()の追加 <ul style="list-style-type: none"> → sc_main()の引数取得が容易になる。 IP等をコマンドラインから制御可能になる。 ・新規コールバック関数の追加 <ul style="list-style-type: none"> → sc_main()以外でも呼び出し可。 before_end_of_elaboration() start_of_simulation(), end_of_simulation() 2. TLMのしやすさ向上 <ul style="list-style-type: none"> ・sc_exportの追加や動的プロセス生成 3. その他 <ul style="list-style-type: none"> ・.read(), .write()メソッド無しにアクセス可。 ・パートセレクトした値をテンポラリ変数に入れる必要なし。 ・異なる型を接続可。
分類	情報



出展: ESCUG(European SystemC Users Group Meeting) Feb.2004

© Copyright 2005 JEITA, All rights reserved

JEITA

52

【45】 ESCUG9 2004

タイトル	A Functional Coverage Prototype for SystemC-based Verification of Chipset Designs	
著者	Robert Siegmund, Ulrich Hensel, Andreas Herrholz, Isa Volt	
所属	AMD Dresden Design Center, Dresden, Germany	
概要	SystemC環境におけるCoverage-APIのプロトタイプを作成した。AMDのチップセットの検証に適用した。	
内容	<p>sr_cover, sr_toggle_coverage等のキーワードを拡張し、カバレッジ測定に有用なAPI群のプロトタイプを構築した。よくあるラインカバレッジではなく、データやパラメタ等のカバレッジを測定するもの。ランダム検証と組み合わせると、質の高い検証を行うことができる。</p> <p>次のような機能を持つ。</p> <ul style="list-style-type: none"> ・カバレッジ対象とカバレッジデータベースの定義 ・トグルカバレッジ ・クロスカバレッジ ・カバレッジリポート <p>カバレッジ結果の分析手法も提案されており、カバレッジホルの調査を行うことができる。SystemC 2.0にコンパチブル。</p>	
分類	検証	

出展: ESCUG(European SystemC Users Group Meeting) Feb.2004

© Copyright 2005 JEITA, All rights reserved

JEITA

53

【46】 ESCUG9 2004

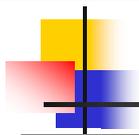
タイトル	System Architecture Performance Modeling with SystemC	
著者	Soeren Sonntag, Raimar Thudt, Tino Bertram	
所属	Infineon	
概要	DSLAM(パケット中継器)をSystemCを使いシステムレベルでシミュレーション。SystemCの有効性を示すとともにSystemCの問題点を指摘。	
内容	<p>DSLAM(Digital Subscriber Line Access Multiplexer: DSL回線とギガビットイーサ間のパケット中継器)をSystemCにより抽象度の高いモデルと、やや抽象度を下げたモデルを作成し、シミュレーション。抽象度の違いによるシミュレーション速度差が報告されている。今回のモデルリングにおいてはSystemC 2.0.1の問題点を2つ指摘。</p> <ol style="list-style-type: none"> 1. 複数のイベントをキューにためることができない。 <pre> event.notify(10, SC_NS); event.notify(20, SC_NS); wait(event); // 時刻10nsに戻ってくる(時刻10nsにイベント発生) wait(event); // 戻ってこない(時刻20nsにはイベントは発生しない) </pre> 2. プロセスは、自分をトリガしたイベントがどのイベントかがわからない。 <p>対処方法はあるが、SystemCのカーネル側で対応したほうが効率的。</p> <pre> sc_port<int, 512> in; //512チャンネルの多重ポート sensitive << in; wait(1); // どのチャンネルにイベントが発生したのか? </pre> 	
分類	モデリング	

出展: ESCUG(European SystemC Users Group Meeting) Feb.2004

© Copyright 2005 JEITA, All rights reserved

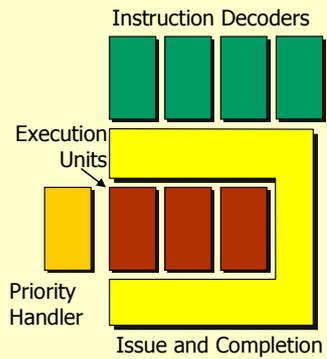
JEITA

54

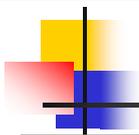


【47】 ESCUG10 2004

タイトル	XML-based Parameterization of SystemC Models
著者	Andreas Doering
所属	IBM Research Lab. Zurich, Switzerland
概要	XMLを利用してマルチスレッドCPUのSystemCモデルをパラメタライズし、コンフィギュラブルなモデルと、そのモデルを展開する仕組みを作成し、アーキテクチャ評価した。
内容	<p>次のような項目についてパラメタ化を行った。</p> <ul style="list-style-type: none"> ・スレッドの数、パイプラインのタイプと数 ・レイテンシ、容量、命令分布、分岐予測、発行ポリシー <p>同一のスレッドを複数インスタンスする際の問題点：</p> <ul style="list-style-type: none"> ・各スレッドがそれぞれ独立のIndexを持つ必要がある。 ・パラメタを渡す仕組みが必要。 <p>これらを解決するために、SC_THREAD_I マクロを作成した。</p> <pre>#define SC_THREAD_I(func,i) char strgbuf[30]; ¥ printf(strgbuf,#func "_%d",i); ¥ declare_thread_process(func ## _handle, ¥ strgbuf, SC_CURRENT_USER_MODULE, func)</pre>
分類	モデリング

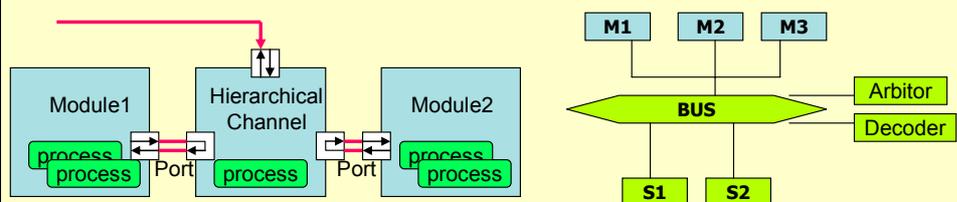


出展: ESCUG(European SystemC Users Group Meeting) Oct.2004



【48】 日本SCUG 2003

タイトル	Transaction Level Modeling with SystemC
著者	Thorsten Groetker
所属	Synopsys GmbH
概要	SystemCでのトランザクションレベルのモデル手法。AMBA AHB/APBのバスモデル
内容	<p>SystemCのコミュニケーションメカニズムについて説明。Module間のコミュニケーションは、Channelで実現。Moduleは、Portを介して、Channelと接続される。Portは、Interface Method Calls(IMC)を使って実現されている。Channelは、改造構造も実現可能。</p> <p>メッセージレイヤーと、Transactionレイヤー、Transferレイヤーの3つの抽象モデルがあるが、このうちの2種類のTransaction Levelモデルを紹介。1つは、Generic Transaction Level Communication。もう一つは、AMBAのAHB/APBプロトコルのトランザクションモデリング。AMBAのAHB/APBのプロトコルモデルが、burst_read(), burst_write()のBlockingのAPIと、request(), has_grant(), init_transaction(), set_data(), response()のNon-BlockingのAPIで実現。基本は、OSCIのsimulation環境にある、Simple Busモデル。</p>
分類	モデリング



出展: 日本SCUG(日本 SystemC Users Group Meeting) 2003

【49】日本SCUG 2003

タイトル	ハードウェア設計とSystemC
著者	長尾文昭
所属	三洋電機セミコンダクターカンパニー
概要	三洋電機セミコンダクターカンパニーでの現状のSystemC設計環境
内容	<p>■SystemCを選んだ理由 (1) ユーザ定義の信号型 (2) 固定小数点型 (3) C++/Cの通常プログラムとの組み合わせ (4) インターフェースの抽象記述</p> <p>■動作合成ツール評価 動作合成ツールの評価を500程度程度の信号処理回路で評価。記述は各社の入力フォーマットにあわせて修正し、熟練者が時間をかけて実現できるレベルの設計制約を設定して、合成。制約条件を満たしたツールがある。使い方の工夫で運用可能と判断し、新規開発機種で適用開始。</p> <p>■SystemCからVerilogへの変換ツール SystemCからVerilog-RTLへの変換ツールを自社開発(合成機能はなし)。SystemCのmethodプロセスとCthreadプロセスに対応。型合成や、型変換を自動で行う。</p> <p>■現在のSystemC設計環境 デザイン入力とデバッグの統合環境を利用。シミュレータはリファレンスシミュレータ他、高速シミュレーション環境も予定。動作合成は、実用レベルのものも登場してきたが、入力構文がHDLレベルを超えないため、OSCI合成WGIに期待</p> <p>■将来への設計環境の希望 (1) アルゴリズム検証用モデルをリファレンスとし、専用プロセッサIP、専用ICEを含めた、協調検証環境 (2) モジュール、インターフェース、ユーザ定義型それぞれに対応したハードウェア合成環境。合成用構文の標準化。ハードウェア合成ルールチェッカ、合成前後の等価性チェックツール。</p>
分類	合成

出展: 日本SCUG(日本 SystemC Users Group Meeting) 2003

© Copyright 2005 JEITA, All rights reserved

JEITA

57

【50】日本SCUG 2003

タイトル	Socを利用したSoCシステムシミュレータ開発事例
著者	竹村和祥
所属	松下電器半導体社開発本部
概要	SoCシステムシミュレーション環境の開発事例
内容	<p>マルチメディア処理LSIのシステムシミュレータをSoCで開発。目的は、①システムアーキテクチャ性能検証。②HWリファレンスモデルの作成。③SWチューニング。開発期間は6ヶ月。コードサイズは30K行。精度はサイクル。一部擬似モデル。効果として、①は複数アプリSWを実行しながらのバス調停方式の有効性をES前に実証 50KHz ②RTL検証でのランダム検証でのリファレンスモデルとして活用 ③SWデバッグを接続しての、MW、アプリケーションSWのチューニングを実行</p> <p>The diagram illustrates a SystemC SoC architecture. At the top, components include '内製プロセッサ' (In-house Processor), 'DMAC', 'ARM (擬似モデル)' (ARM Simulation Model), and 'HWエンジン (擬似モデル)' (HW Engine Simulation Model). These are connected to a central 'AHB' bus. Below the AHB bus are 'IF' (Interface) blocks and '共有メモリ' (Shared Memory) blocks for 'SRAM' and 'SDRAM'. Further down, 'INTC' (Interrupt Controller), 'ブリッジ' (Bridge), and 'TIMER' are connected to an 'APB' bus. At the bottom, 'AHBスレーブ' (AHB Slaves) are connected to the APB bus. A legend on the right indicates that blue boxes represent 'サイクル精度' (Cycle Accuracy) and pink boxes represent '擬似モデル' (Simulation Model).</p>
分類	Co-Design/Co-Simulation

出展: 日本SCUG(日本 SystemC Users Group Meeting) 2003

© Copyright 2005 JEITA, All rights reserved

JEITA

58

4.2.6 IEEE 版 SystemC V2.0.1 LRM レビュー活動 サマリー

2005.03

■背景

昨年末、IEEE 内に、P1666 のもと SystemC 標準化の SG が開始され、標準化対象となる、新たな SystemC 2.0.1 LRM ドラフトが、WEB (<http://www.eda.org/systemc/>) にて公開された。

2003 年度、本 TG 内では OSCI 版の SystemC 2.0.1 LRM のレビューを行い、その結果を issue report として、IEEE/DASC に提出したが、この最新版についても再度 LRM のレビューを行うこととした。

■LRM レビュー活動

レビュー方法としては、SystemC2.0.1 LRM を基に、各社、各自でレビューし、その結果を持ち寄り、集中審議における議論を通して、整理・まとめを行った。その際に、以下の 3 点に留意して、レビューを行った。

1. 旧 LRM (OSCI 版) のレビューをした際に発見した ISSUE が改善されているかどうか
 2. 新たな問題点がないかどうか
 3. 旧 LRM にあって新 LRM で落ちている点はないか、また新 LRM で追加された点は何か
- 整理・まとめにおいては、前回と同じく、VHDL IR Template の分類、記述方法を参考とした。

■トータル件数 43 件

IEEE 版 LRM に対する新規 issue report : 31 件

再度 issue report するもの : 12 件

参考 :

旧 LRM(OSCI 版 LRM)に対する issue report :	62 件
IEEE 版 LRM で修正された issue :	34 件
仕様削除、説明削除、実装依存等で問題なくなったもの :	16 件
再度 issue report が必要なもの :	12 件

■分類別件数

(1) Language Definition Problem	33 件
(2) Terminology, Grammar and Typographical Errors	8 件
(3) Language Deficiencies and Modeling Problem	1 件
(4) Request	1 件

■全体としての傾向

昨年度レビューを行った OSCI LRM と比べた場合、章立てが大きく変更され、内容的にも新しい内容となっている。全体的に、読みやすく、かつわかりやすくなっているというのが、多くの委員の感想。また、前回の本 TG が指摘した ISSUE の多くも修正済み (50 件/62 件) であった。

一般のユーザ向けというよりは、シミュレータの実装者を読者として想定したスタイルで記述

されている。このためか、シミュレーションのアルゴリズム、Elaboration、モジュール階層などの説明も、前回と比べて追加されている。

分類別では、Language Definition Problem に関する ISSUE が、最も多かった。

■ 今後

IEEE SystemC 2.0.1LRM に関する ISSUE リストを、IEEE SystemC SG に改善提案として、提出する。4月に、SystemC 2.1 に対する LRM が完成予定であり、本 TG でもレビューを行う予定である。

■ 参考資料

- － VHDL IR Template
- － SystemC 2.0.1 LRM (旧)
- － Draft Standard for SystemC (IEEE PXXX?/D2.0.1, October 18 2004)
- － SystemC 2.0.1 reference simulator ソースコード

4.2.6.1 IEEE 版 LRM ISSUE LIST (一覧表)

JEITA Issue #	Classification	Summary	LRM section	Affiliation	OSCI-LRM Issue #
1	Language Definition Problem	interface, discrepancy between 3.1.4 and B.2.29	3.1.4	Fujitsu Limited	
2	Terminology, Grammar, and Typographical Error	definition error of sc_event operator=	5.7.2	NEC Electronics Corporation	
3	Terminology, Grammar, and Typographical Error	The explanation for to_seconds() includes a typographical error.	5.8.4	NEC Electronics Corporation	
4	Language Deficiencies and Modeling Problems	sc_event_finder should support multiple parent ports	5.5	NEC Electronics Corporation	50
5	Terminology, Grammar and Typographical Errors	A typo is in description.	5.12.1	OKI Electric Industry Co.,Ltd.	
6	Language Definition Problem	Explain typedef declaration.	5.15.2	OKI Electric Industry Co.,Ltd.	
7	Language Definition Problem	"sc_attr cltn(const sc_attr cltn&);" differs from the reference	5.15.2	OKI Electric Industry Co.,Ltd.	
8	Language Definition Problem	About deletion of get_data_ref() and get_new_value()	6.4.2	OKI Electric Industry Co.,Ltd.	
9	Request	Add the example of update().	6.6.4	OKI Electric Industry Co.,Ltd.	
10	Language Definition Problem	About deletion of get_data_ref()	6.7.2	OKI Electric Industry Co.,Ltd.	
11	Language Definition Problem	About the reason of an addition of start_time() and	6.7.2	OKI Electric Industry Co.,Ltd.	
12	Language Definition Problem	About deletion of a sc_bit class	7	OKI Electric Industry Co.,Ltd.	
13	Language Definition Problem	About deletion of sc_set default time unit	5.8	OKI Electric Industry Co.,Ltd.	
14	Terminology, Grammar, and Typographical Errors	Indentation	7.4.2.2	Sony Corporation	
15	Language Definition Problem	Add the instruction of referenced section for operators.	7.5.5, 7.5.6	Cadence Design Systems, Japan	
16	Language Definition Problem	What is meaning "// NOT YET IMPLEMENTED".	7.5.9.2	Cadence Design Systems, Japan	
17	Language Definition Problem	Insufficient explanation of constructor.	6.21.2	Toshiba Corporation	20
18	Language Definition Problem	Insufficient explanation of constructor.	6.22.2	Toshiba Corporation	24
19	Language Definition Problem	Constructor with a double-type argument is not declared explicit.	7.5.5.2	Toshiba Corporation	6
20	Language Definition Problem	There are assignment operators, which don't have the corresponding constructors.	7.5.5.2	Toshiba Corporation	7
21	Language Definition Problem	Constructor with double-type argument is not declared explicit.	7.5.6.2	Toshiba Corporation	11
22	Language Definition Problem	There are assignment operators, which don't have the corresponding constructors.	7.5.6.2	Toshiba Corporation	12
23	Language Definition Problem	Insufficient explanation of explicit conversion.	7.6.4.6	Toshiba Corporation	33
24	Terminology, Grammar and Typographical Errors	Wrong inequality	7.7.1	Toshiba Corporation	
25	Terminology, Grammar and Typographical Errors	Wrong range of vlue for a unsinged fixed-point format	7.7.1	Toshiba Corporation	
26	Language Definition Problem	fixed-precision fixed-point is not mentioned	7.7.7.1	Toshiba Corporation	
27	Language Definition Problem	Lack of an explanation of a member function	7.7.7.2	Toshiba Corporation	
28	Terminology, Grammar and Typographical Errors	Wrong range of singed representation	7.7.9.1	Toshiba Corporation	
29	Language Definition Problem	Insufficient explanation of sc_max function.	8.4.2	Toshiba Corporation	48
30	Language Definition Problem	Insufficient explanation of sc_min function.	8.4.3	Toshiba Corporation	49
31	Language Definition Problem	There're no explanation about "to_int,to_long,to_uint,to_ulong" in "Explicit type conversion"	7.6.2.5 7.6.3.6 7.6.4.6 7.6.7.6	Nihon Synopsys Co. Ltd	
32	Terminology, Grammar and Typographical Errors	In explanation literal "I" is used, but in the table literal "i" is used.	7.6.2.6	Nihon Synopsys Co. Ltd	
33	Language Definition Problem	There's no example of bit select in "7.6.7 Bit-selects" section.	7.6.7	Nihon Synopsys Co. Ltd	
34	Language Definition Problem	There's no example of part select in "7.6.8 Part-selects" section.	7.6.8	Nihon Synopsys Co. Ltd	
35	Language Definition Problem	There's no example of part select in "7.6.9 Concatenations"	7.6.9	Nihon Synopsys Co. Ltd	
36	Language Definition Problem	There's no explanation about "scan, print".	7.6.2, 7.6.3, 7.6.4, 7.6.6, 7.6.7, 7.6.8, 7.6.9	Nihon Synopsys Co. Ltd	
37	Language Definition Problem	There's no explanation about "length".	7.6.3, 7.6.4, 7.6.6, 7.6.8	Nihon Synopsys Co. Ltd	
38	Language Definition Problem	More detail explanation about a difference between sc_fixed and	7.7.3.1	RICOH COMPANY,LTD	
39	Language Definition Problem	Macro DECL_BIN_OP(/,div) is expanded	7.7.10.2 7.7.11.2 7.7.12.2	RICOH COMPANY,LTD	
40	Language Definition Problem	add the comment of the difference of sc_pvector and vector	8.1	Renesas Technology Corp.	59
41	Language Definition Problem	The lack of explanation what methods should be defined	8.1	Renesas Technology Corp.	
42	Language Definition Problem	add the comment of the difference of sc_string and string	8.2	Renesas Technology Corp.	58
43	Language Definition Problem	The lack of explanation of some methods of sc_string class	8.2	Renesas Technology Corp.	

JEITA Issue Number: 1

Classification: language definition
Language Version: IEEE PXXX?/D2.0.1, October 18 200
LRM Sections: 3.1.4 Specific technical terms
Summary: interface, discrepancy between 3.1.4 and B.2.29
Related Issues:
Relevant LRM Sections:
Key Words and Phrases: interface, interface proper
Current Status:
Superseded By:

Date Submitted: 2005/02/18
Author of Submission: Takashi Hasegawa
Author's Affiliation: Fujitsu Limited
Author's Post Address:
Author's Phone Number:
Author's Fax Number:
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem**3.1.4 Specific technical terms**

<snip>

An interface is a class derived from the class `sc_interface`.
An interface proper is a class derived from the class `sc_interface` but not derived from the class `sc_object`.

Annex B.2

2.29 interface: A class derived from class `sc_interface` but not derived from class `sc_object`. An interface declares the set of methods to be implemented within a channel, and called via a port. An interface contains pure virtual function declarations, but no function definitions and no data members.

JEITA Issue Number: 2

Classification: Terminology, Grammar, and Typographical Errors
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Sections: 5.7.2 Class definition
Summary: definition error of `sc_event operator=`
Related Issues:
Relevant LRM Sections:
Key Words and Phrases: `sc_event operator=`
Current Status:
Superseded By:

Date Submitted: 2004/02/06
Author of Submission: Kazunori Goto
Author's Affiliation: NEC Electronics Corporation
Author's Post Address: 1753, Shimonumabe, Nakahara-Ku, Kawasaki, Kanagawa 211-8668, Japan
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

The definition of `operator =` has two "operator". This is a typographical error.

[Error]

```
class sc_event
{
...
    sc_event& operator operator=( const sc_event &);
};
```

[Correct]

```
class sc_event
{
...
    sc_event& operator=( const sc_event &);
};
```

JEITA Issue Number: 3

Classification: Terminology, Grammar, and Typographical Errors
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Sections: 5.8.4 Functions and operators
Summary: The explanation for `to_seconds()` includes a typographical error.
Related Issues:
Relevant LRM Sections:
Key Words and Phrases: `sc_event`
Current Status:
Superseded By:

Date Submitted: 2004/02/06
Author of Submission: Kazunori Goto
Author's Affiliation: NEC Electronics Corporation
Author's Post Address: 1753, Shimonumabe, Nakahara-Ku, Kawasaki, Kanagawa 211-8668, Japan
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

The explanation for `to_seconds()` includes a typographical error.

[Error]

```
double to_double() const;
double to_seconds() const;
These functions shall return the underlying representation of the time value, first converting the value to a double in the case of to_double and to_seconds, and then also scaling the resultant value to units of 1 second in the case of to_seconds.
```

[Correct]

```
double to_double() const;
double to_seconds() const;
These functions shall return the underlying representation of the time value, first converting the value to a double in the case of to_double, and then also scaling the resultant value to units of 1 second in the case of to_seconds.
```

JEITA Issue Number: 4

Classification: Language Deficiencies and Modeling Problems
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Section: 5.5.3 `sc_event_finder_t`
Summary: `sc_event_finder` should support multiple parent ports
Related Issues:
Relevant LRM Sections:
Key Words and Phrases: `sc_event_finder`
Current Status:
Superseded By:

Date Submitted: 2004/02/06
Author of Submission: Kazunori Goto
Author's Affiliation: NEC Electronics Corporation
Author's Post Address: 1753, Shimonumabe, Nakahara-Ku, Kawasaki, Kanagawa 211-8668, Japan

Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

Current sc_event_finder doesn't support multiple parent ports. This is not good in case of modeling a transaction bus.
When there is connection to multiple parent ports in case of tlm port, current sc_event_finder can only find the first parent port even though it is connected to other parent ports.

JEITA Issue Number: 5

Classification: Terminology, Grammar and Typographical Errors

Language Version: IEEE PXXX?/D2.0.1, October 18 2004

Summary: A typo is in description.

Related Issues: none

LRM Sections: 5.12.1 Discription

Key Words and Phrases: sc_object

Current Status:

Superseded By:

Date Submitted: 2004/02/28

Author of Submission: Yasuyuki Shimizu

Author's Affiliation: OKI Electric Industry Co.,Ltd.

Author's Post Address: 550-1 Higashiasakawa-cho Hachioiji-shi,
Tokyo 193-8550, Japan

Author's Phone Number: +81-426-62-6190(ex.4626)

Author's Fax Number: +81-426-64-1256

Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

"5.12.1 Description" has description of "an sc_object."
However, this is the error of "a sc_object."

Proposed Resolution

Please correct a typo.

JEITA Issue Number: 6

Classification: Language Definition Problem

Language Version: IEEE PXXX?/D2.0.1, October 18 2004

Summary: Explain typedef declaration.

Related Issues: none

LRM Sections: 5.15.2 Class definition

Key Words and Phrases: sc_attr_cltn, elem_type, typedef

Current Status:

Superseded By:

Date Submitted: 2004/02/28

Author of Submission: Yasuyuki Shimizu

Author's Affiliation: OKI Electric Industry Co.,Ltd.

Author's Post Address: 550-1 Higashiasakawa-cho Hachioiji-shi,
Tokyo 193-8550, Japan

Author's Phone Number: +81-426-62-6190(ex.4626)

Author's Fax Number: +81-426-64-1256

Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

In "5.15.2 Class definition", It is declared as follows,

```
public:
typedef sc_attr_base* elem_type;
typedef elem_type* iterator;
typedef const elem_type* const_iterator;
```

Why does it make a typedef declaration of iterator and const_iterator using elem_type?

For instance, as follows,

```
public:
typedef sc_attr_base* elem_type;
typedef sc_attr_base* iterator;
typedef const sc_attr_base* const_iterator;
```

Proposed Resolution

Please tell us the reason for making a typedef declaration of iterator and const_iterator using elem_type.

JEITA Issue Number: 7

Classification: Language Definition Problem

Language Version: IEEE PXXX?/D2.0.1, October 18 2004

Summary: "sc_attr_cltn(const sc_attr_cltn&);" differs from the reference simulator.

Related Issues: none

LRM Sections: 5.15.2 Class definition

Key Words and Phrases: sc_attr_cltn, vector, constructor

Current Status:

Superseded By:

Date Submitted: 2004/02/28

Author of Submission: Yasuyuki Shimizu

Author's Affiliation: OKI Electric Industry Co.,Ltd.

Author's Post Address: 550-1 Higashiasakawa-cho Hachioiji-shi,
Tokyo 193-8550, Japan

Author's Phone Number: +81-426-62-6190(ex.4626)

Author's Fax Number: +81-426-64-1256

Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

By the reference simulator of OSCI, "sc_attr_cltn(const sc_attr_cltn&);" is public.
However, in this chapter, it is private.

Proposed Resolution

Isn't there any problem by changing into private from public?

JEITA Issue Number: 8

Classification: Language Definition Problem

Language Version: IEEE PXXX?/D2.0.1, October 18 2004

Summary: About deletion of get_data_ref() and get_new_value()

Related Issues: none

LRM Sections: 6.4.2 Class definition

Key Words and Phrases: sc_signal, get_data_reg(), get_new_value()

Current Status:

Superseded By:

Date Submitted: 2004/02/28

Author of Submission: Yasuyuki Shimizu

Author's Affiliation: OKI Electric Industry Co.,Ltd.

Author's Post Address: 550-1 Higashiasakawa-cho Hachioiji-shi,
Tokyo 193-8550, Japan

Author's Phone Number: +81-426-62-6190(ex.4626)

Author's Fax Number: +81-426-64-1256

Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

There were `get_data_ref()` and `get_new_value()` in the reference simulator of OSCI at the member function of `sc_signal`.
But, `get_data_ref()` and `get_new_value()` have vanished from the member function in this chapter.

Proposed Resolution

Is there no influence to the simulator or modeling even if it deletes from a member function?

JEITA Issue Number: 9

Classification: Request
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
Summary: Add the example of `update()`.
Related Issues: none
LRM Sections: 6.6.4 Member functions
Key Words and Phrases: `sc_buffer`, `update()`
Current Status:
Superseded By:

Date Submitted: 2004/02/28
Author of Submission: Yasuyuki Shimizu
Author's Affiliation: OKI Electric Industry Co.,Ltd.
Author's Post Address: 550-1 Higashiasakawa-cho Hachioji-shi,
Tokyo 193-8550, Japan
Author's Phone Number: +81-426-62-6190(ex.4626)
Author's Fax Number: +81-426-64-1256
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

The explanation of `update()` of "6.6.4 Member functions" is still indefinite.

Proposed Resolution

Please add description of the concrete example about operation to the explanation of `update()` of "6.6.4 Member functions".

JEITA Issue Number: 10

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
Summary: About deletion of `get_data_ref()`
Related Issues: none
LRM Sections: 6.7.2 Class definition
Key Words and Phrases: `sc_clock`, `get_data_ref()`
Current Status:
Superseded By:

Date Submitted: 2004/02/28
Author of Submission: Yasuyuki Shimizu
Author's Affiliation: OKI Electric Industry Co.,Ltd.
Author's Post Address: 550-1 Higashiasakawa-cho Hachioji-shi,
Tokyo 193-8550, Japan
Author's Phone Number: +81-426-62-6190(ex.4626)
Author's Fax Number: +81-426-64-1256
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

`get_data_ref()` existed in the reference simulator at the member function of `sc_clock`.

However, there is not `get_data_ref()` in "6.7.2 Class definition".

Proposed Resolution

Why was `get_data_ref()` deleted from the member function of `sc_clock`?

JEITA Issue Number: 11

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
Summary: About the reason of an addition of `start_time()` and `posedge_first()`.
Related Issues: none
LRM Sections: 6.7.2 Class definition
Key Words and Phrases: `sc_clock`, `start_time()`, `posedge_first()`
Current Status:
Superseded By:

Date Submitted: 2004/02/28
Author of Submission: Yasuyuki Shimizu
Author's Affiliation: OKI Electric Industry Co.,Ltd.
Author's Post Address: 550-1 Higashiasakawa-cho Hachioji-shi,
Tokyo 193-8550, Japan
Author's Phone Number: +81-426-62-6190(ex.4626)
Author's Fax Number: +81-426-64-1256
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

`start_time()` and `posedge_first()` are declared as the member function of "6.7.2 Class definition".

However, `start_time()` and `posedge_first()` do not exist in the member function of the `sc_clock` class of a reference simulator.

Proposed Resolution

What is purpose of `start_time()` and `posedge_first()` defined?
Moreover, for what kind of case are these functions used?

JEITA Issue Number: 12

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
Summary: About deletion of a `sc_bit` class
Related Issues: none
LRM Sections: 7. Data types
Key Words and Phrases: `sc_bit`, `data type`
Current Status:
Superseded By:

Date Submitted: 2004/02/28
Author of Submission: Yasuyuki Shimizu
Author's Affiliation: OKI Electric Industry Co.,Ltd.
Author's Post Address: 550-1 Higashiasakawa-cho Hachioji-shi,
Tokyo 193-8550, Japan
Author's Phone Number: +81-426-62-6190(ex.4626)
Author's Fax Number: +81-426-64-1256
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

There is no description about `sc_bit` which existed in LRM of OSCI.

Proposed Resolution

Why was the `sc_bit` class deleted from the data type?

JEITA Issue Number: 13

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
Summary: About deletion of `sc_set_default_time_unit`

Related Issues: none
LRM Sections: 5.8 sc_time()
Key Words and Phrases: sc_time, sc_set_default_time_unit, time unit
Current Status:
Superseded By:

Date Submitted: 2004/02/28
Author of Submission: Yasuyuki Shimizu
Author's Affiliation: OKI Electric Industry Co.,Ltd.
Author's Post Address: 550-1 Higashiasakawa-cho Hachioji-shi,
Tokyo 193-8550, Japan
Author's Phone Number: +81-426-62-6190(ex.4626)
Author's Fax Number: +81-426-64-1256
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

There is no description about sc_set_default_time_unit which existed in LRM of OSCI.

Proposed Resolution

Why was sc_set_default_time_unit deleted?

JEITA Issue Number: 14

Classification: Terminology, Grammar, and Typographical Errors
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
Summary: Indentation
Related Issues:
Relevant LRM Sections: 7.4.2.2 Class definition
Key Words and Phrases: // comment line
Current Status:
Superseded By:

Date Submitted: 2005/02/18
Author of Submission: Masaru Kakimoto
Author's Affiliation: Sony Corporation
Author's Post Address: Shinagawa Seaside North Tower,
4-12-3, Higashi-Shinagawa, Shinagawa-ku,
Tokyo 140-0002 JAPAN
Author's Phone Number: +81-3-6834-5120
Author's Fax Number: +81-3-6834-5109
Author's Net Address: masaru.kakimoto@jp.sony.com

Description of Problem

Indentation error in following lines.
P133 // prefix and postfix increment and decrement operators
P134 // explicit conversion to character string

Proposed Resolution

Add indentation.

JEITA Issue Number: 15

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
Summary: Add the instruction of referenced section for operators.
Related Issues:
Relevant LRM Sections: 7.5.5 sc_bigint 7.5.6 sc_biguint
Key Words and Phrases: sc_bigint, sc_biguint, sc_signed, sc_unsigned
Current Status:
Superseded By:

Date Submitted: 2005/02/18
Author of Submission: Takashi Ohsaka
Author's Affiliation: Cadence Design Systems, Japan
Author's Post Address: 3-17-6 Shin-Yokohama, Kouhoku-ku,
Yokohama 222-0033, Japan
Author's Phone Number: +81-45-475-8430
Author's Fax Number: +81-45-475-6331
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

Both sc_bigint and sc_biguint sections have the description that sc_signed and

sc_unsigned are base classes. However a user have to associate operators (arithmetics, bitwise and comparisons) that are available from table of base classes. I think to add instruction for this is more useful for customer.

Proposed Resolution

Add the instruction for operators table in base class.

In sc_bigint : Specified operators in base class as described in 7.5.3.7.

In sc_biguint : Specified operators in base class as described in 7.5.4.7.

JEITA Issue Number: 16

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
Summary: What is meaning "///NOT YET IMPLEMENTED".

Related Issues:
Relevant LRM Sections: 7.5.9.2 Class definition
Key Words and Phrases: sc_signed_concref sc_signed_concref_r
Current Status:
Superseded By:

Date Submitted: 2005/02/18
Author of Submission: Takashi Ohsaka
Author's Affiliation: Cadence Design Systems, Japan
Author's Post Address: 3-17-6 Shin-Yokohama, Kouhoku-ku,
Yokohama 222-0033, Japan
Author's Phone Number: +81-45-475-8430
Author's Fax Number: +81-45-475-6331
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

I understand this feature is not implemented yet in this version. But I seem this feature is "a future plan". If so, LRM does not have to include a plan. Or will this feature be added until fixed IEEE LRM? I want to know mean of this description.

Proposed Resolution

If this is meaning a future plan, remove this comments from class definitions and only add description about "not implemented" outside class definitions.

JEITA Issue Number: 17

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Section: 6.21.2 Class definition (page 105)
Summary: Insufficient explanation of constructor.
Related Issues:

Relevant LRM Sections:
Key Words and Phrases: `sc_fifo_in_if`, constructor, protected
Current Status:
Superseded By:

Date Submitted: 2005/02/22
Author of Submission: Hiroshi Imai
Author's Affiliation: Toshiba Corporation
Author's Post Address: 580-1, Horikawa-Cho, Saiwai-Ku, Kawasaki,
212-8520, Japan

Author's Phone Number:
Author's Fax Number:
Author's Net Address: `eda-sctg@eda.ics.es.osaka-u.ac.jp`

Description of Problem

Explain reasons why the constructor without argument is declared protected.
See the following code.

Protected Constructor
`sc_fifo_in_if()`;

JEITA Issue Number: 18
Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Section: 6.22.2 Class definition (page 106)
Summary: Insufficient explanation of constructor.
Related Issues:
Relevant LRM Sections:
Key Words and Phrases: `sc_fifo_out_if`, constructor, protected
Current Status:
Superseded By:

Date Submitted: 2005/02/22
Author of Submission: Hiroshi Imai
Author's Affiliation: Toshiba Corporation
Author's Post Address: 580-1, Horikawa-Cho, Saiwai-Ku, Kawasaki,
212-8520, Japan

Author's Phone Number:
Author's Fax Number:
Author's Net Address: `eda-sctg@eda.ics.es.osaka-u.ac.jp`

Description of Problem

Explain reasons why the constructor without argument is described as a protected constructor.
See the following explanation.

Protected Constructor
`sc_fifo_out_if()`;

JEITA Issue Number: 19
Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Section: 7.5.5.2 Class definition (page 171)
Summary: Constructor with a double-type argument is not declared explicit.
Related Issues:
Relevant LRM Sections:
Key Words and Phrases: `sc_bigint`, constructor
Current Status:
Superseded By:

Date Submitted: 2005/02/22
Author of Submission: Hiroshi Imai

Author's Affiliation: Toshiba Corporation
Author's Post Address: 580-1, Horikawa-Cho, Saiwai-Ku, Kawasaki,
212-8520, Japan

Author's Phone Number:
Author's Fax Number:
Author's Net Address: `eda-sctg@eda.ics.es.osaka-u.ac.jp`

Description of Problem

Explain reasons why constructor with a double-type argument is not declared explicit.
See the following codes.

```
//Constructors  
sc_bigint( double v );  
explicit sc_bigint( const sc_fxval& v );  
explicit sc_bigint( const sc_fxval_fast& v );  
explicit sc_bigint( const sc_fxnum& v );  
explicit sc_bigint( const sc_fxnum_fast& v );
```

JEITA Issue Number: 20
Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Section: 7.5.5.2 Class definition (page 171)
Summary: There are assignment operators, which don't have the corresponding constructors.

Related Issues:
Relevant LRM Sections:
Key Words and Phrases: `sc_bigint`, constructor, assignment operator,
`sc_int_base`, `sc_uint_base`

Current Status:
Superseded By:

Date Submitted: 2005/02/22
Author of Submission: Hiroshi Imai
Author's Affiliation: Toshiba Corporation
Author's Post Address: 580-1, Horikawa-Cho, Saiwai-Ku, Kawasaki,
212-8520, Japan

Author's Phone Number:
Author's Fax Number:
Author's Net Address: `eda-sctg@eda.ics.es.osaka-u.ac.jp`

Description of Problem

Although there are assignment operators with a `sc_int_base`- and `sc_uint_base`-type argument, there aren't constructors with a `sc_int_base`- and `sc_uint_base`-type argument.
Does this cause any problem?
See the following codes.

```
//Assignment Operators  
sc_bigint<W>& operator = ( const sc_int_base& v );  
sc_bigint<W>& operator = ( const sc_uint_base& v );
```

JEITA Issue Number: 21
Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Section: 7.5.6.2 Class definition (page 173)
Summary: Constructor with double-type argument is not declared explicit.

Related Issues:
Relevant LRM Sections:
Key Words and Phrases: `sc_biguint`, constructor
Current Status:
Superseded By:

Date Submitted: 2005/02/22
Author of Submission: Hiroshi Imai
Author's Affiliation: Toshiba Corporation
Author's Post Address: 580-1, Horikawa-Cho, Saiwai-Ku, Kawasaki,
212-8520, Japan

Author's Phone Number:
Author's Fax Number:
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

Explain reasons why a constructor with double-type is not declared explicit.
See the following codes.

Constructors

```
sc_biguint( double v );  
explicit sc_biguint( const sc_fxval& v );  
explicit sc_biguint( const sc_fxval_fast& v );  
explicit sc_biguint( const sc_fxnum& v );  
explicit sc_biguint( const sc_fxnum_fast& v );
```

JEITA Issue Number: 22

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Section: 7.5.6.2 Class definition (page 173)
Summary: There are assignment operators, which don't
have the corresponding constructors.

Related Issues:
Relevant LRM Sections:
Key Words and Phrases: sc_biguint, constructor, assignment operator
Current Status:
Superseded By:

Date Submitted: 2005/02/22
Author of Submission: Hiroshi Imai
Author's Affiliation: Toshiba Corporation
Author's Post Address: 580-1, Horikawa-Cho, Saiwai-Ku, Kawasaki,
212-8520, Japan

Author's Phone Number:
Author's Fax Number:
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

Although there are assignment operators with a sc_int_base- and
sc_uint_base-type argument,
there aren't constructors with a sc_int_base- and sc_uint_base-type
argument.

Does this cause any problem?
See the following codes.

Assignment Operators

```
sc_biguint<W>& operator = ( const sc_int_base& v );  
sc_biguint<W>& operator = ( const sc_uint_base& v );
```

JEITA Issue Number: 23

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Section: 7.6.4.6 Explicit type conversion (page 197)
Summary: Insufficient explanation of explicit conversion.
Related Issues:
Relevant LRM Sections:
Key Words and Phrases: sc_lv_base, explicit conversion, dontcare, high
Z

Current Status:
Superseded By:

Date Submitted: 2005/02/22
Author of Submission: Hiroshi Imai
Author's Affiliation: Toshiba Corporation
Author's Post Address: 580-1, Horikawa-Cho, Saiwai-Ku, Kawasaki,
212-8520, Japan

Author's Phone Number:
Author's Fax Number:
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

In the explanation of explicit conversion, the behavior of a logic vector
including X or Z is not explained.
See the following explanation.

Explicit Conversion

```
int to_int() const ;  
long to_long() const ;  
unsigned int to_uint() const ;  
unsigned long to_ulong() const ;
```

JEITA Issue Number: 24

Classification: Terminology, Grammar and Typographical
Errors
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Section: 7.7.1 Fixed-point representation (p217)
Summary: Wrong inequality

Related Issues:
Relevant LRM Sections:
Key Words and Phrases: fixed-point representation, word length, integer
word length
Current Status:
Superseded By:

Date Submitted: 2005/02/10
Author of Submission: Hiroshi Imai
Author's Affiliation: Toshiba Corporation
Author's Post Address: 580-1, Horikawa-Cho, Saiwai-Ku, Kawasaki,
212-8520, Japan

Author's Phone Number:
Author's Fax Number:
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

Interpretation of fixed-point representation is explained in three cases.
Inequality in the first case is wrong.
It should be "wl < iw!".

JEITA Issue Number: 25

Classification: Terminology, Grammar and Typographical
Errors
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Section: 7.7.1 Fixed-point representation (p218)
Summary: Wrong range of value for a unsigned
fixed-point format

Related Issues:
Relevant LRM Sections:
Key Words and Phrases: unsigned fixed-point format, range of values
Current Status:
Superseded By:

Date Submitted: 2005/02/10
Author of Submission: Hiroshi Imai
Author's Affiliation: Toshiba Corporation

Author's Post Address: 580-1, Horikawa-Cho, Saiwai-Ku, Kawasaki,
212-8520, Japan

Author's Phone Number:

Author's Fax Number:

Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

The range of values for a unsigned fixed-point format is wrong.
It should be "[0, 2^{iw1} - 2^{^{-(wl-iwl)}}]".

JEITA Issue Number: 26

Classification: Language Definition Problem

Language Version: IEEE PXXX?/D2.0.1, October 18 2004

LRM Section: 7.7.7.1 Reading parameter settings (p224)

Summary: fixed-precision fixed-point is not mentioned

Related Issues:

Relevant LRM Sections:

Key Words and Phrases: fixed-precision fixed-point, parameter settings

Current Status:

Superseded By:

Date Submitted: 2005/02/10

Author of Submission: Hiroshi Imai

Author's Affiliation: Toshiba Corporation

Author's Post Address: 580-1, Horikawa-Cho, Saiwai-Ku, Kawasaki,
212-8520, Japan

Author's Phone Number:

Author's Fax Number:

Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

This section explains parameter setting of fixed point object.
Fixed point object consists of fixed-precision fixed-point object
and limited-precision fixed-point object.
But in the original LRM, only limited-precision fixed-point object is
mentioned
as follows:

The following functions are defined for every limited-precision fixed-point
object and shall return its current parameter settings (at run-time).

It should be as follows:

The following functions are defined for every fixed-precision fixed-point
object
and limited-precision fixed-point object and shall return its current
parameter settings (at run-time).

JEITA Issue Number: 27

Classification: Language Definition Problem

Language Version: IEEE PXXX?/D2.0.1, October 18 2004

LRM Section: 7.7.7.2 Value attributes (p225)

Summary: Lack of an explanation of a member function

Related Issues:

Relevant LRM Sections:

Key Words and Phrases: fixed-precision fixed-point, value attributes,
value()

Current Status:

Superseded By:

Date Submitted: 2005/02/10

Author of Submission: Hiroshi Imai

Author's Affiliation: Toshiba Corporation

Author's Post Address: 580-1, Horikawa-Cho, Saiwai-Ku, Kawasaki,
212-8520, Japan

Author's Phone Number:

Author's Fax Number:

Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

An explanation of a member function, const sc_fxval_fast value() const ; is
missing.

JEITA Issue Number: 28

Classification: Terminology, Grammar and Typographical
Errors

Language Version: IEEE PXXX?/D2.0.1, October 18 2004

LRM Section: 7.7.9.1 Overflow modes (p228)

Summary: Wrong range of signed representation

Related Issues:

Relevant LRM Sections:

Key Words and Phrases: signed representation, the lowest (negative)
number, the highest (positive) number
range

Current Status:

Superseded By:

Date Submitted: 2005/02/10

Author of Submission: Hiroshi Imai

Author's Affiliation: Toshiba Corporation

Author's Post Address: 580-1, Horikawa-Cho, Saiwai-Ku, Kawasaki,
212-8520, Japan

Author's Phone Number:

Author's Fax Number:

Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

The explanation of range of signed representation is wrong.
It should be "-2^{n-1} (=MIN) <= x <= (2^{n-1} - 1) (=MAX)".

JEITA Issue Number: 29

Classification: Language Definition Problem

Language Version: IEEE PXXX?/D2.0.1, October 18 2004

LRM Section: 8.4.2 sc_max (page 317)

Summary: Insufficient explanation of sc_max function.

Related Issues:

Relevant LRM Sections:

Key Words and Phrases: sc_max, >=

Current Status:

Superseded By:

Date Submitted: 2005/02/11

Author of Submission: Hiroshi Imai

Author's Affiliation: Toshiba Corporation

Author's Post Address: 580-1, Horikawa-Cho, Saiwai-Ku, Kawasaki,
212-8520, Japan

Author's Phone Number:

Author's Fax Number:

Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

The class T must have ">=" member function to use sc_max function.
Explain the necessary condition of the class T.
See the following explanation.

template <class T>

```
const T sc_max( const T& a , const T& b ) { return (( a >= b ) ? a : b );}
```

JEITA Issue Number: 30

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Section: 8.4.3 sc_min (page 317)
Summary: Insufficient explanation of sc_min function.

Related Issues:
Relevant LRM Sections:
Key Words and Phrases: sc_min, <=
Current Status:
Superseded By:

Date Submitted: 2005/02/11
Author of Submission: Hiroshi Imai
Author's Affiliation: Toshiba Corporation
Author's Post Address: 580-1, Horikawa-Cho, Saiwai-Ku, Kawasaki,
212-8520, Japan

Author's Phone Number:
Author's Fax Number:
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

The class T must have "<=" member function to use sc_min function.
Explain the necessary condition of the class T.
See the following explanation.

```
template <class T>  
const T sc_min( const T& a , const T& b ) { return (( a <= b ) ? a : b );}
```

JEITA Issue Number: 31

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Section: 7.6.2.5, 7.6.3.6, 7.6.4.6, 7.6.7.6
Summary: There're no explanation about "to_int, to_long,
to_uint, to_ulong" in "Explicit type
conversion".

Related Issues: None
Relevant LRM Sections: 7.6.2.5, 7.6.3.6, 7.6.4.6, 7.6.7.6 Explicit type
conversion
Key Words and Phrases: to_int, to_long, to_uint, to_ulong
Current Status:
Superseded By:

Date Submitted: 2005/2/17
Author of Submission: Junji Nakano
Author's Affiliation: Nihon Synopsys Co. Ltd
Author's Post Address:
Author's Phone Number:
Author's Fax Number:
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

There're no explanation about "to_int,to_long,to_uint,to_ulong" in "Explicit
type conversion" 7.6.2.5, 7.6.3.6, 7.6.4.6, 7.6.7.6. These explanations are
summarized in 7.2.8, so we should add "reference to 7.2.8 about to_int,
to_long, to_uint, to_ulong" description.

JEITA Issue Number: 32

Classification: Terminology, Grammar and Typographical
Errors
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Section: 7.6.2.6 Bitwise and comparison operators

Summary: In explanation literal "l" is used, but in the table
literal "l" is used.

Related Issues: None
Relevant LRM Sections:
Key Words and Phrases: comparison operations table
Current Status:
Superseded By:

Date Submitted: 2005/2/17
Author of Submission: Junji Nakano
Author's Affiliation: Nihon Synopsys Co. Ltd
Author's Post Address:
Author's Phone Number:
Author's Fax Number:
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

In explanation literal "l" is used, but in the table literal "l" is used.

JEITA Issue Number: 33

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Section: 7.6.7
Summary: There's no example of bit select in "7.6.7
Bit-selects" section.

Related Issues: None
Relevant LRM Sections: 7.2.4
Key Words and Phrases:
Current Status:
Superseded By:

Date Submitted: 2005/2/17
Author of Submission: Junji Nakano
Author's Affiliation: Nihon Synopsys Co. Ltd
Author's Post Address:
Author's Phone Number:
Author's Fax Number:
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

There's no example of bit select in "7.6.7 Bit-selects" section. The example
is in "7.2.4 Bit-select" section. So we should add "reference to 7.2.4"
description.

JEITA Issue Number: 34

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Section: 7.6.8 Part-selects
Summary: There's no example of part select in "7.6.8
Part-selects" section.

Related Issues: None
Relevant LRM Sections: 7.2.5
Key Words and Phrases: Part-select
Current Status:
Superseded By:

Date Submitted: 2005/2/17
Author of Submission: Junji Nakano
Author's Affiliation: Nihon Synopsys Co. Ltd
Author's Post Address:
Author's Phone Number:
Author's Fax Number:
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

There's no example of part select in "7.6.8 Part-selects" section. The example is in "7.2.5 Part-select" section. So we should add "reference to 7.2.5" description.

JEITA Issue Number: 35

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Section: 7.6.9 Concatenations
Summary: There's no example of part select in "7.6.9 Concatenations" section.
Related Issues: None
Relevant LRM Sections: 7.2.6"
Key Words and Phrases: Concatenations
Current Status:
Superseded By:

Date Submitted: 2005/2/17
Author of Submission: Junji Nakano
Author's Affiliation: Nihon Synopsys Co. Ltd
Author's Post Address:
Author's Phone Number:
Author's Fax Number:
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

There's no example of part select in "7.6.9 Concatenations" section. The example is in "7.2.6 Concatenations" section. So we should add "reference to 7.2.6" description.

JEITA Issue Number: 36

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Section: 7.6.2 , 7.6.3, 7.6.4, 7.6.6, 7.6.7, 7.6.8, 7.6.9
Summary: There's not explanation about "scan, print".
Related Issues: None
Relevant LRM Sections:
Key Words and Phrases: scan, print
Current Status:
Superseded By:

Date Submitted: 2005/2/17
Author of Submission: Junji Nakano
Author's Affiliation: Nihon Synopsys Co. Ltd
Author's Post Address:
Author's Phone Number:
Author's Fax Number:
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

There's no explanation about "scan, print" in "7.6.2 sc_logic, 7.6.3 sc_bv_base, 7.6.4 sc_lv_base, 7.6.6 sc_lv, 7.6.7 Bit-selects, 7.6.8 Part-Selects, 7.6.9 Concatenations"
Add explanation about "scan" and "print".

JEITA Issue Number: 37

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Section: 7.6.3, 7.6.4, 7.6.6, 7.6.8
Summary: There's no explanation about "length"
Related Issues: None
Relevant LRM Sections:

Key Words and Phrases: length

Current Status:
Superseded By:

Date Submitted: 2005/2/17
Author of Submission: Junji Nakano
Author's Affiliation: Nihon Synopsys Co. Ltd
Author's Post Address:
Author's Phone Number:
Author's Fax Number:
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

There's no explanation about "length" in "7.6.3 sc_bv_base, 7.6.4 sc_lv_base, 7.6.6 sc_lv, 7.6.8 Part-Selects"
Add explanation about "length".

JEITA Issue Number: 38

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Section: 7.7.3.1 Fixed precision fixed-point types
Summary: More detail explanation about a difference between sc_fixed and sc_fix

Related Issues:
Relevant LRM Sections:
Key Words and Phrases: sc_fixed,sc_fix
Current Status:
Superseded By:

Date Submitted: 2005/02/18
Author of Submission: Yasutaka Tsukamoto
Author's Affiliation: RICOH COMPANY,LTD
Author's Post Address: 3-2-3 Shin-Yokohama, Kouhoku-ku, Yokohama 222-8530, Japan
Author's Phone Number: +81-45-477-1447
Author's Fax Number: +81-45-477-1766
Author's Net Address: yasutaka.tsukamoto@nts.ricoh.co.jp

Description of Problem

There is no detail explanation about a difference between sc_fixed and sc_fix.

Proposed Resolution

A difference between sc_fixed and sc_fix should be described like 6.8.3 of OSCI LRM.

JEITA Issue Number: 39

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Sections: 7.7.10.2 7.7.11.2 7.7.12.2
Summary: Macro DECL_BIN_OP(/,div) is expanded

Related Issues:
Relevant LRM Sections:
Key Words and Phrases: sc_fxnum,sc_fxval,sc_fxval_fast
Current Status:
Superseded By:

Date Submitted: 2005/02/18
Author of Submission: Yasutaka Tsukamoto
Author's Affiliation: RICOH COMPANY,LTD
Author's Post Address: 3-2-3 Shin-Yokohama, Kouhoku-ku, Yokohama 222-8530, Japan
Author's Phone Number: +81-45-477-1447

Author's Fax Number: +81-45-477-1766
Author's Net Address: yasutaka.tsukamoto@nts.ricoh.co.jp

Description of Problem

The Macro DECL_BIN_OP(/,div) is expanded at the class definition in 7.7.10.2, 7.7.11.2 and 7.7.12.2.

Proposed Resolution

These expanded descriptions should be deleted.

JEITA Issue Number: 40

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
LRM Sections: 8.1 sc_pvector
Summary: add the comment of the difference of sc_pvector and vector

Related Issues:
Relevant LRM Sections:
Key Words and Phrases: sc_pvector,vector
Current Status:
Superseded By:

Date Submitted: 2004/02/09
Author of Submission: Masashi Watanabe
Author's Affiliation: Renesas Technology Corp.
Author's Post Address: 20-1, Jousuihon-cho, 5chome, Kodaira-shi, Tokyo 187-8588, Japan
Author's Phone Number: +81-42-320-7300(ex.4372)
Author's Fax Number: +81-42-327-8195
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

In SystemC library there is sc_vector class, and Standard Template Library(STL) includes vector class too. I think sc_vector class differs somewhat from vector class. So, it is necessary that there is a difference of them in LRM and when do we use sc_pvector class not vector class.

Proposed Resolution

Add the comment why sc_pvector was created.

JEITA Issue Number: 41

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
Summary: The lack of explanation what methods should be defined

Related Issues:
Relevant LRM Sections: 8.1 sc_pvector
Key Words and Phrases: sc_pvector template required method
Current Status:
Superseded By:

Date Submitted: 2005/02/17
Author of Submission: Masashi Watanabe
Author's Affiliation: Renesas Technology Corp.
Author's Post Address: 20-1, Jousuihon-cho, 5chome, Kodaira-shi, Tokyo 187-8588, Japan
Author's Phone Number: +81-42-320-7300(ex.4372)
Author's Fax Number: +81-42-327-8195
Author's Net Address: watanabe.masashi@renesas.com

Description of Problem

As sc_pvector is template class, user can specify the original class to template argument. examples..

```
sc_pvector<my_class> sc_pvector_inst;
```

And original class must define some methods, which are used by sc_pvector to define its method. Thus please add the comment about what method should be defined in original class.

Proposed Resolution

Add the definition about methods which user must define

JEITA Issue Number: 42

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
Summary: add the comment of the difference of sc_string and string

Related Issues:
LRM Sections: 8.2 sc_string
Key Words and Phrases: sc_string,string,STL
Current Status:
Superseded By:

Date Submitted: 2004/02/09
Author of Submission: Masashi Watanabe
Author's Affiliation: Renesas Technology Corp.
Author's Post Address: 20-1, Jousuihon-cho, 5chome, Kodaira-shi, Tokyo 187-8588, Japan
Author's Phone Number: +81-42-320-7300(ex.4372)
Author's Fax Number: +81-42-327-8195
Author's Net Address: eda-sctg@eda.ics.es.osaka-u.ac.jp

Description of Problem

In SystemC library there is sc_string class, and Standard Template Library(STL) includes string class too. I think sc_string class differs somewhat from string class. So, it is necessary that there is a difference of them in LRM and when do we use sc_string class not string class.

Proposed Resolution

Add the comment why sc_string was created.

JEITA Issue Number: 43

Classification: Language Definition Problem
Language Version: IEEE PXXX?/D2.0.1, October 18 2004
Summary: The lack of explanation of some methods of sc_string class

Related Issues:
Relevant LRM Sections: 8.2 sc_string
Key Words and Phrases: sc_string method
Current Status:
Superseded By:

Date Submitted: 2005/02/17
Author of Submission: Masashi Watanabe
Author's Affiliation: Renesas Technology Corp.
Author's Post Address: 20-1, Jousuihon-cho, 5chome, Kodaira-shi, Tokyo 187-8588, Japan
Author's Phone Number: +81-42-320-7300(ex.4372)
Author's Fax Number: +81-42-327-8195
Author's Net Address: watanabe.masashi@renesas.com

Description of Problem

The following methods are in OSCI LRM, but not in IEEE LRM.

```
sc_string( const char* s, int n );  
friend sc_string operator + ( const char* s, const sc_string& t );  
sc_string substr( int first, int last ) const;  
operator const char*() const;  
operator const char*() const;  
static sc_string to_string(const char* format, ...);  
template<class T> sc_string& fmt(const T& t);  
sc_string& fmt(const sc_string& s);  
int pos(const sc_string& sub_string) const;  
sc_string& remove(unsigned index, unsigned length);  
sc_string& insert(const sc_string& sub_string, unsigned index);  
bool is_delimiter(const sc_string& str, unsigned index) const;  
bool contains(char c) const;  
sc_string uppercase() const;  
sc_string lowercase() const;  
static sc_string make_str(long n);  
void set( int index, char c );  
int cmp( const char* s ) const;  
int cmp( const sc_string& s ) const;  
void print( ostream& os = cout ) const;
```

Proposed Resolution

Add those declaration and explanation

添付資料 SVTG-1

SystemVerilog IEEE標準化状況

JEITA SystemVerilog Task Group

JEITA



IEEEにおけるSystemVerilog標準化の過程

- 2004/05 AccelleraによるSystemVerilog3.1a言語仕様定義完了
- 2004/06 AccelleraからIEEEへSystemVerilogのdonation
- 2004/06 IEEE P1800(SystemVerilog)発足
- 2004/10 P1800 draft1作成完
- 2004/11 P1800 draft2作成完
- 2005/01 P1800 draft3作成完

JEITA

(2)



IEEEにおけるSystemVerilog標準化の今後の予定

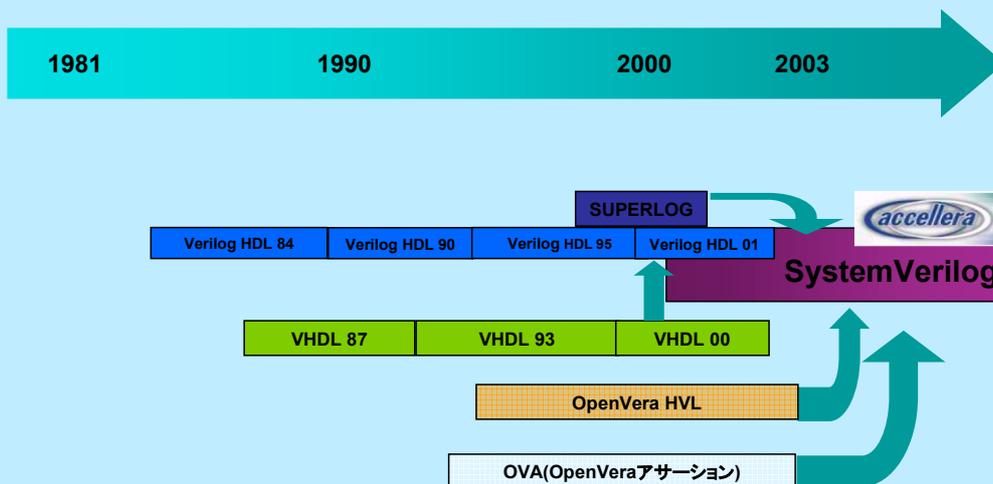
- 2005/02 P1800 draft4作成完
- 2005/03 P1800 draft4による initial投票
- 2005/05 P1800 draft5 による本投票
- 2005/09 IEEE P1800承認
- 2005/10 IEEE Std 1800-2005 出版

JEITA

(3)



SystemVerilogの成り立ち



JEITA

(4)



IEEE P1800 WG Errata 提出案件の状況

JEITA SystemVerilog Task Group

JEITA



Agenda

- IEEE P1800 WG Errata 提出案件の概要
- JEITA SV-TG 提出案件の検討状況
- 未決案件の詳細

JEITA

(6)



IEEE P1800 WG Errata 提出案件の概要

- P1800 Errata
 - SystemVerilog 3.1a Language Reference Manual に対するすべての訂正、追加提案をデータベース化
 - 各提案に対する検討状況、責任者、検討経過、結論等を閲覧可能
- P1800 Errata の状況 ('05/3/2 現在)

Status	Open	Resolved	Closed	Total
new	55			55
feedback	11			11
acknowledged	34			34
assigned	34			34
closed			246	246

Statusの意味

新規
関係者からのフィードバックあり
修正案が承認済み
担当者をアサイン済み
修正案がドラフトに反映済み

- JEITA SV-TG より提案した Errata の状況
 - '04/8/4、32件のErrata を P1800 へ提案 (次頁参照)
 - '05/3/2 時点の処理状況

new	5	} 詳細後述
feedback	1	
acknowledged	3	… ドラフトへ反映された時点でそれを確認する
closed	23	… 反映を確認済み

JEITA

(7)



JEITA SV-TG 提出案件の検討状況

Id	Summary	Status
126	Jeita 1: Typo in 17.13	closed
127	Jeita 2: Cover statement handling in formal verification undefined	new
128	Jeita 3: typo in Section 17	closed
129	Jeita 4: typesetting in 17	acknowledged
130	Jeita 5: Typos in 17	acknowledged
131	Jeita 6: Intersect may be confusing in section 20	closed
132	Jeita 7: strong intersect needed	closed
133	Jeita 8: assuring randomization	closed
134	Jeita 9: 4 state values as constraints	new
135	Jeita 10: typo	closed
136	Jeita 11: unclear assertion semantics	closed
137	Jeita 12: iff issues	closed
138	Jeita 13: unclear behavior of default sequence	feedback
139	Jeita 14: unclear how to change value in covergroup	closed
140	Jeita 15: typo in section 20	closed
141	Jeita 16: Typo in section 20	closed
142	Jeita 17: BNF typo	closed
143	Jeita 18: rewrite #2 step in section 2?	closed
144	Jeita 19: can step be used beyond clocking blocks?	closed
145	Jeita 20: Max uses for time unit?	closed
146	Jeita 21: Simple explanation in section2	closed
147	Jeita 22: Enum clarity needed	closed
148	Jeita 23: mixing of signed/unsigned	closed
149	Jeita 24: Change variable names in examples in 3.16	closed
150	Jeita 25: Example typo in 4.2	closed
151	Jeita 26: optional rule to specify size in 4.2	new
152	Jeita 27: Another option for size	new
153	Jeita #28: typo in 28	closed
154	Jeita 29: Dual Data Rate needed in always ff	new
155	Jeita 30: In the Index, check isn't on page 199	acknowledged
156	Jeita 31: In Index, issue with blocking and DPI imports	closed
157	Jeita 32: typo in 5.6	closed

JEITA

(8)



未決案件の詳細(1)

- ID:127, Jeita 2: Cover statement handling in formal verification undefined, new
 - Cover文の取り扱いについて、動的検証に関しては明確に定義されているが、静的検証に関しては定義されていないことを指摘。
 - Status: new (enhancement request)
 - Assigned to SV-AC
 - 最新の検討状況: This errata, 127 as Coverage for static verification was voted to be resolved not-a-bug. However, to keep it as an enhancement request, in September 13, 2004 meeting.
 - 特に強くプッシュせず、進行を見守る
- ID:134, Jeita 9: 4 state values as constraints, new
 - 制約付きランダム生成において、2値(0/1)だけでなく4値(0/1/X/Z)を使用可能とすることの有効性を提案。
 - Status: new (enhancement request)
 - 議論された形跡なし
 - 改めて問い合わせる

JEITA

(9)



未決案件の詳細(2)

- ID:138, Jeita 13: unclear behavior of default sequence, feedback
 - Coverpoint文において、default sequenceとしてカウントされる事象の定義が不明確であることを指摘。
 - Status: feedback
 - Assigned to SV-EC (Errata)
 - 最新の検討状況: This point may be discussed a bit, because the number of possible unspecified sequences is pretty large and the coverage engine needs to keep track of them even if the sequence is not a prefix of the specified one. For instance if $4 \Rightarrow 5$ is specified, then any of $0=1$, $9 \Rightarrow 10$ etc needs to be tracked in default sequence. Also it is not clear whether $9 \Rightarrow 10$ and $9 \Rightarrow 10 \Rightarrow 11$ will be counted as two in the default bin. Ideally, it should be mandatory for a default sequence to have the same starting point as any of the specified sequences, to be counted. So for this example where specified sequence is $4 \Rightarrow 5 \Rightarrow 6$, the default sequences to be counted would be $4 \Rightarrow 6$, $4 \Rightarrow 5 \Rightarrow 7$ etc. A sequence like $12 \Rightarrow 14$ is not counted in default.
 - 進行を見守る

JEITA

(10)



未決案件の詳細(3)

- ID:151, Jeita 26: optional rule to specify size in 4.2, new
 - unpacked 配列の範囲指定方法の修正と拡張を要求している
 - Status: new
 - 現ドラフトでは [size] が [0:size-1] として解釈されるが、これは C 言語の書式と逆になるため混乱を招きかねない
 - この問題を解決するため、[size] を [size-1:0]、[!size] を [0:size-1] とする変更を提案している
 - Status が assigned に代わった時点で、担当者と連絡をとる
- ID:152, Jeita 27: Another option for size, new
 - 前項 ID:151 について、別の修正案を提案している
 - Status: new
 - [size>] で [size-1:0] とし、[size<] で [0:size-1] とする
 - Status が assigned に代わった時点で、担当者と連絡をとる

【補足 (ID:151,152共通)】 C言語の書式は LRM が正しく、JEITA 側の指摘に誤解があった。上記は誤解を含む提案内容をそのまま記載している

JEITA

(11)

SystemVerilog

未決案件の詳細(4)

- ID:154, Jeita 29: Dual Data Rate needed in always_ff, new
 - always_ff の記述方法の拡張、あるいは、LRM への明記を要求している
 - Status: new
 - event_expression に posedge あるいは negedge しか指定できないため、DDR(Dual Data Rate) をモデリングできない
 - あるいは、posedge/negedge を省略した場合に両エッジとなることが明記されていない
 - そこで、下記2つの修正案を提案している
 - “edge” (1995の予約語) で両エッジとなるように拡張
例: always_ff @(edge clk, negedge rst)
 - posedge/negedge を表記しない場合に両エッジとなることを明記
例: always_ff @(clk, negedge rst)
 - Status が assigned に代わった時点で、担当者と連絡をとる

JEITA

(12)

SystemVerilog

SystemVerilog 3.1a 言語チュートリアル

JEITA SystemVerilog Task Group

JEITA



Agenda

- 本言語チュートリアルの方針
- SystemVerilogについて
- ユーザの立場からみたSystemVerilogの利点
- 言語チュートリアル
 - SystemVerilog言語構成図
 - 全ての分野で現在使える構文
 - テストベンチとして現在使える構文
 - 早期にサポートを期待する構文
 - 予約語一覧
- Summary

JEITA

(14)



チュートリアル作成メンバー

- 「JEITA EDA技術専門委員会／標準化小委員会
傘下の SystemVerilogタスクグループ」
SystemVerilogの国際標準化活動に日本から参画

- メンバー 9社(半導体ベンダー:5、EDAベンダー:4)

(株)沖ネットワークエルエスアイ、(株)図研、
(株)東芝、日本ケイデンス・デザイン・システムズ社、
日本シノプシス(株)、松下電器産業(株)、
メンター・グラフィックス・ジャパン(株)、
富士通(株)、(株)ルネサステクノロジ

(注)五十音順

JEITA

(15)



本言語チュートリアルの方針

- ユーザの立場からみたSystemVerilog3.1aの言語
チュートリアル



SystemVerilog 3.1a Language Reference Manual
(586ページ)

メンバーによる有用なSV(SystemVerilog)構文の抽出
(JEITA発行 “EDAアニュアルレポート2003” 参照)

EDAツール
サポート状況調査



・有用な機能
・EDAサポート状況
の観点からのSV言語紹介
(2004/12末 現在)

JEITA

(16)



本言語チュートリアルの方針

- 調査対象 EDAツール(バージョン)

2004/12末 現在

	Atrenta	Cadence	Mentor	Synopsys
Lint	SpyGlass 3.5.2-SV			LEDA 4.1
Simulation		IUS 5.4	ModelSim 6.0	VCS 7.2
Synthesis				DesignCompiler 2004.06-SP2-4
PropertyCheck				
EquivalenceCheck		Conformal 5.0		Formality 2004.06-SP3

(注)本リストに載っていないツールや各ツールの詳細に関しては各ベンダーにお問い合わせください。

JEITA

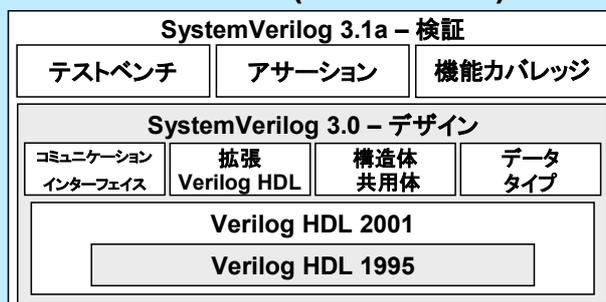
(17)



SystemVerilogについて

Verilog HDL(IEEE1364)の拡張仕様

HDVL (HDL + HVL)



- デザインと検証のために言語を統合
 - HDVL (Hardware Description and Verification Language)

JEITA

(18)



ユーザの立場からみたSystemVerilogの利点-1

- **記述の曖昧さの解消**
 - Ex. `always_comb/latch/ff`
`unique/priority case/if,`
`logic, bit`
- **可読性,保守容易性,記述量の削減**
 - Ex. 構造体 `struct` , 列挙型 `enum`,
モジュール接続 `.name/.*` ワイルドカード
多次元配列, `interface, do_while`
- **EDAツールの性能向上**
 - DPI

JEITA

(19)



ユーザの立場からみたSystemVerilogの利点-2

◆設計者の本音



EDAツールに依存しない言語が使いたい・・・
SVはIEEE P1800として標準化作業中

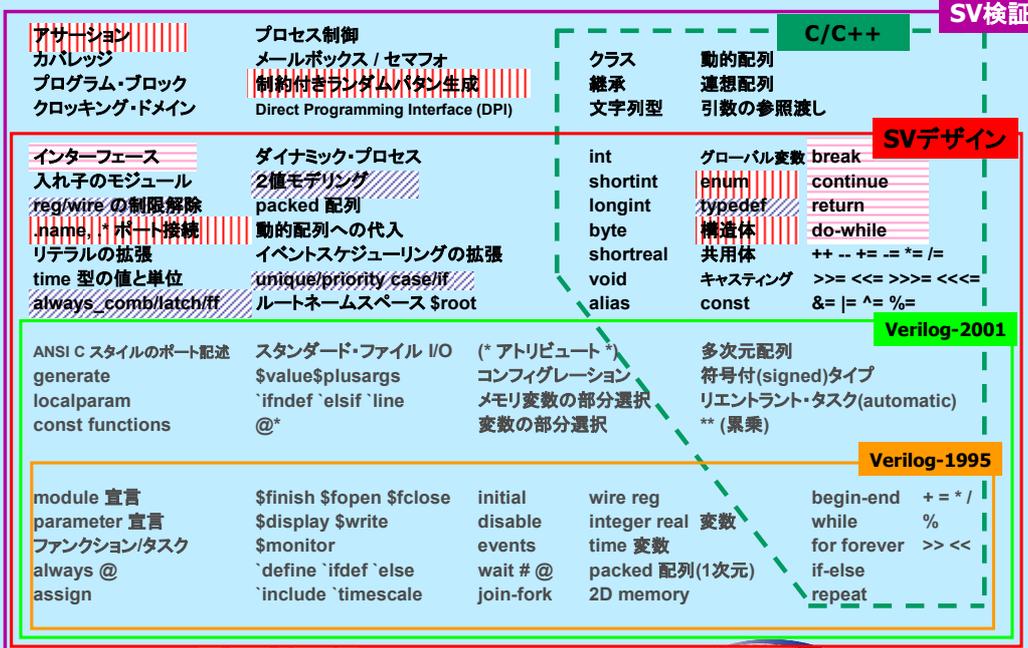
JEITA

(20)



言語チュートリアル SystemVerilog言語構成図

全分野 ベンチ 早期期待



JEITA

(21)

SystemVerilog

言語チュートリアル

- **全ての分野で現在使える構文**
(Lint, Simulation, Synthesis, Equivalence Checker)
 - ① logic, bit
 - ② always_comb/latch/ff
 - ③ unique/priority case/if
 - ④ typedef
- **テストベンチとして現在使える構文**
(Simulation)
 - ⑤ do_while, break, continue
 - ⑥ interface
- **早期にサポートを期待する構文**
 - ⑦ 構造体 struct
 - ⑧ 列挙型 enum
 - ⑨ 多次元配列
 - ⑩ モジュール接続 .name/. *
 - ⑪ アサーション
 - ⑫ 制約付きランダムパターン生成 constraint

JEITA

(22)

SystemVerilog

● 全ての分野で現在使える構文

- **logic 4値**
 - **reg, wire**の区別なく記述可能
 - **wire**として使用する場合, ドライバを一つに限定
 - ⇒ レースコンディションが起きないモデリング
 - リントツールでチェック可能(早期検出)
- **bit 2値**
 - シミュレーションの高速化
 - 初期値が0と規定されている
 - ⇒ すべての2値シミュレーションで同一動作

JEITA

(23)



● 全ての分野で現在使える構文

- **always_comb/latch/ff**
 - 設計者が意図する回路を明示的に指定できる
 - ツール認識に依存しない回路が得られる
 - **always_comb, latch**はセンシティビティリストが不要
 - 詳細な規定によりツール間の不整合を防ぐ
 - initial, always**のすべての手続き的ブロックがアクティブになった後、時間ゼロにおいて自動的に1回実行される
 - ⇒ 時間ゼロでの出力値は入力値に一致する
 - 従来の**always**記述で@イベントがトリガされず
 - シミュレーションロックしていた問題を解決する

JEITA

(24)



● 全ての分野で現在使える構文

● unique/priority case/if

- case/if文にて条件処理を順次/並列の何れかの解釈に指定
- ツールベンダー依存のプラグマを排除。シミュレーション、合成で統一した解釈を実現
 - full_case, parallel_case
 - RTL/GATE間のシミュレーションミスマッチ防止

```
unique case (sel)
  3'b001 : muxo = a;
  3'b010 : muxo = b;
  3'b100 : muxo = c;
endcase
```

```
priority if (sel0) q = 3'b000;
else if (sel1) q = 3'b011;
else if (sel2) q = 3'b110;
else if (sel3) q = 3'b111;
```

JEITA

(25)



● 全ての分野で現在使える構文

● unique/priority case/if

SystemVerilog定義	従来のプラグマ定義
unique case	// parallel_case full_case
unique case + default	// parallel_case
priority case	// full_case
priority case + default	// none
unique if	
priority if	

- caseにdefault:行が無い場合は、full_caseの扱い
- unique caseはparallel_caseの解釈と同一

JEITA

(26)



● 全ての分野で現在使える構文

● ユーザ定義データタイプ `typedef`

- 可読性の向上
ユーザにとって意味のある型を定義できる
- 保守容易性の向上
記述部を変更せず、宣言部を変更するだけで定義を変更できる

`typedef`を用いたワード定義の例

```
typedef logic [31:0] word_t;  
word_t A, B;
```

1ワードを32ビットとして定義

```
typedef logic [63:0] word_t;  
word_t A, B;
```

1ワードを64ビットとして定義

JEITA

(27)



● 全ての分野で現在使える構文

● `do_while`, `break`, `continue`

- C言語ライクな構文制御が可能
- 可読性の向上, 記述量の削減

```
initial begin  
do  
begin  
n = n++;  
if ( (n%3) == 0 ) continue;  
foo = myfunc(n);  
if (foo == 22) break;  
end  
while (foo != 0);  
...  
end
```

`continue` に
よってループの
最初へ戻り繰り返す

`break` に
よってループの外
へ抜ける

JEITA

(28)

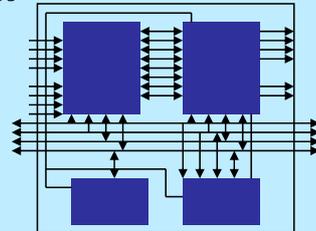


● テストベンチとして現在使える構文

● interface

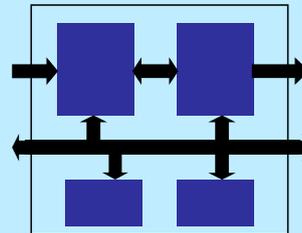
● Verilog HDL 1995/2001 モジュール・ポート接続

- モジュール間の詳細な端子接続を知る必要がある
- 入出力の端子群に変更があった場合記述の変更が煩雑
- ポート宣言を多くのモジュールで何度も繰り返す必要がある



● SystemVerilog interface接続が可能

- モジュール間の接続を一つに束ねて行う
- 接続の内容の定義はモジュール定義からは独立して行う
- 記述量の削減に多大な効果
- ポートの追加・削除に伴うファイル修正作業を軽減



JEITA

(29)

SystemVerilog

● テストベンチとして現在使える構文

● interface

```

module memMod( input logic req,
bit clk,
bit start,
logic [1:0] mode,
logic [7:0] addr,
inout wire [7:0] data,
output bit gnt,
bit rdy );
logic avail;
always @(posedge clk) a.gnt <= a.req & avail;
...
endmodule

module cpuMod(
input bit clk,
bit gnt,
bit rdy,
inout wire [7:0] data,
output bit req,
bit start,
logic [7:0] addr,
logic [1:0] mode );
...
endmodule

module top;
logic req, gnt, start, rdy; // req is logic not bit here
logic clk = 0;
logic [1:0] mode;
logic [7:0] addr;
wire [7:0] data;
memMod mem(req, clk, start, mode, addr, data, gnt, rdy);
cpuMod cpu(clk, gnt, rdy, data, req, start, addr, mode);
endmodule
    
```

interfaceを使わない場合

```

interface simple_bus;
logic req, gnt;
logic [7:0] addr, data;
logic [1:0] mode;
logic start, rdy;
endinterface: simple_bus

module memMod(simple_bus a, input bit clk);
logic avail;
always @(posedge clk) a.gnt <= a.req & avail;
...
endmodule

module cpuMod(simple_bus b, input bit clk);
...
endmodule

module top;
logic clk = 0;
simple_bus sb_intf();
memMod mem(sb_intf, clk);
cpuMod cpu(.b(sb_intf), .clk(clk));
endmodule
    
```

interfaceを使う場合

SystemVerilog

● 早期にサポートを期待する構文

● 構造体 **struct**

- C言語に似た, 簡潔な宣言
 - `typedef` を用いて, データ型名を指定可能
 - 構造体の全体に対し, 一括して値を代入可能

- 可読性, 保守容易性, 記述量の削減

```
struct {
    bit [7:0] opcode;
    bit [23:0] addr;
    data_word data;
} instruction; //名前付き構造体

instruction IR;

IR = {6, 377, 45}; // 構造体への代入
```

JEITA

(31)



● 早期にサポートを期待する構文

● 列挙型 **enum**

- Verilog HDL 1995/2001
`parameter`定数宣言, もしくは``define`マクロ宣言
 - 値に名前をつけるにとどまる

- SystemVerilog
`enum`宣言

- 値の集合を定義, 初期値設定, 範囲チェック可能
- 高抽象化モデリング

記述量の削減に加え、
可読性, 保守容易性の向上

```
parameter RED =0,
           YELLOW =1,
           GREEN = 2;

reg [1:0] traffic_light;
```

Verilog HDL 1995/2001

```
enum {red, yellow, green} traffic_light;
```

SystemVerilog

JEITA

(32)



● 早期にサポートを期待する構文

● 多次元配列

- 可読性, 保守容易性, 記述量の削減

多次元化

```
reg [7:0] mem0[0:1023];
reg [7:0] mem1[0:1023];
reg [7:0] mem2[0:1023];
reg [7:0] mem3[0:1023];
reg [31:0] q;

always @(posedge clk) begin
  if(ce) begin
    if(we[3]) mem3[a] <= d[31:24];
    if(we[2]) mem2[a] <= d[23:16];
    if(we[1]) mem1[a] <= d[15:8];
    if(we[0]) mem0[a] <= d[7:0];
  end
end

always @(posedge clk) begin
  if(ce)
    q <= {mem3[a], mem2[a],
mem1[a], mem0[a]};
end
```

Verilog HDL 1995/2001

```
reg [3:0][7:0] mem[0:1023];
reg [3:0][7:0] q;

always @(posedge clk) begin
  if(ce) begin
    if(we[3]) mem[a][3] <= d[3];
    if(we[2]) mem[a][2] <= d[2];
    if(we[1]) mem[a][1] <= d[1];
    if(we[0]) mem[a][0] <= d[0];
  end
end

always @(posedge clk) begin
  if(ce)
    q <= mem[a];
end
```

同一変数を
Byte単位で
ハンドリング

SystemVerilog

JEITA

(33)

SystemVerilog

● 早期にサポートを期待する構文

● モジュール接続 .name/. * ワイルドカード

- Verilog HDL 1995/2001 でのモジュール接続:

- 順序接続
- 名前接続

```
module my_chip (input wire clock, reset);
  wire [63:0] a, b, c, d;
  alu u1 (clock, a, b, reset, c, d);
  alu u2 (.ck(clock), .reset(reset), .a(a), .b(b), .c(c), .d(d));
```

ポート宣言の順序で接続
(誤りやすい記述)

名前同士で接続
(冗長な記述)

- SystemVerilog で拡張された接続:

- .name を用いて同一端子名省略
- . * ワイルドカード

```
module my_chip (input wire clock, reset);
  wire [63:0] a, b, c, d;
  alu u1 (.ck(clock), .reset, .a, .b, .c, .d);
  alu u2 (.ck(clock), .*);
```

.name によって接続される端子とネット名
が同一のときに記述を削減できる

. * は同一名の端子とネット
を自動的に接続する

JEITA

(34)

SystemVerilog

● 早期にサポートを期待する構文

● アサーション

● アサーション検証とは

“回路動作(プロパティ)を定義し、その正当性を評価”
“評価したかどうかをカバレッジデータとして記録”
する仕組み

● SystemVerilog Assertionの利点

- テンポラル表現の導入による簡潔なプロパティ記述、記述量の削減
- デザイン記述言語とアサーション記述言語の一体化による設計情報の保守容易性向上と再利用性の向上
- アサーション・チェック用モジュールや PLI コールが不要

JEITA

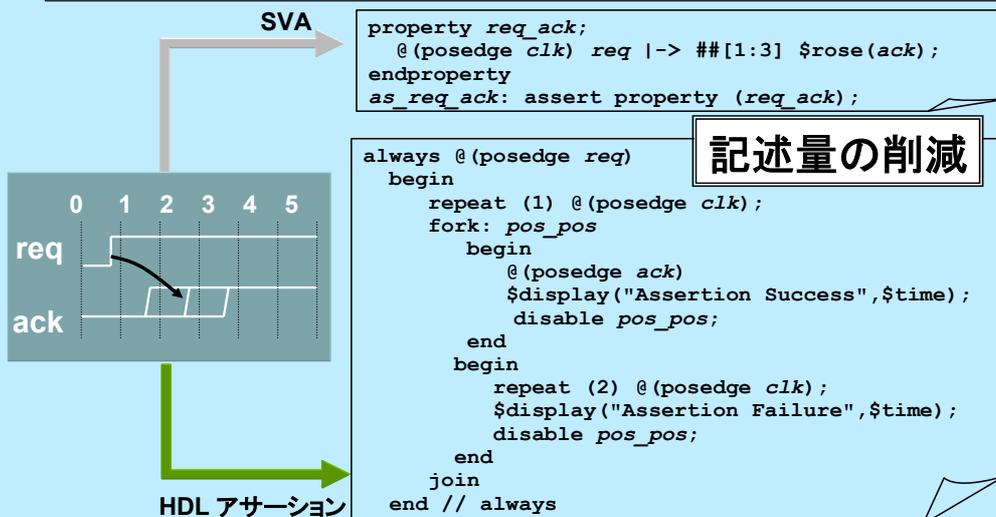
(35)



● 早期にサポートを期待する構文

● アサーション

reqの立ち上がり後、1~3サイクル以内にackが立ち上がる



記述量の削減

JEITA

(36)



● 早期にサポートを期待する構文

● アサーション

- アサーションの構成(あくまでも一例)

```
property プロパティ名;  
    クロック定義 シーケンス -> シーケンス;  
endproperty
```

```
アサーション名: assert property (プロパティ名);
```

```
property req_ack;  
    @(posedge clk) req -> ##[1:3] $rose(ack);  
endproperty
```

```
as_req_ack: assert property (req_ack);
```

JEITA

(37)



● 早期にサポートを期待する構文

● 制約付きランダムパターン生成 constraint

- テストベンチ言語を設計言語に統合



```
class Bus;  
    rand bit [15:0] addr;  
    rand bit [31:0] data;  
    constraint word_align { addr[1:0] == '2b00; }  
endclass  
  
//addrの低位2bitが'00のBusオブジェクトを50個生成  
Bus bus = new;  
repeat(50)  
begin  
    integer result = bus.randomize();  
end
```

JEITA

(38)



● 予約語一覧(220)

Verilog HDL
1995 (102)

Verilog HDL
2001 (21)

SystemVerilog
3.0-3.1a (97)

always	for	output	supply0	automatic	alias	endclass	join_none	string
and	force	parameter	supply1	cell	always_comb	endclocking	local	struct
assign	forever	pmos	table	config	always_ff	endgroup	logic	super
begin	fork	posedge	task	design	always_latch	endinterface	longint	tagged
buf	function	primitive	time	endconfig	assert	endpackage	matches	this
bufif0	highz0	pull0	tran	endgenerate	assume	endprogram	modport	throughout
bufif1	highz1	pull1	tranif0	generate	before	endproperty	new	timeprecision
case	if	pulldown	tranif1	genvar	bind	endsequence	null	timeunit
casex	ifnone	pullup	tri	incdir	bins	enum	package	type
casez	initial	rcmos	tri0	include	binsof	expect	packed	typedef
cmos	inout	real	tri1	instance	bit	export	priority	union
deassign	input	realtime	triand	liblist	break	extends	program	unique
default	integer	reg	trior	library	byte	extern	property	var
defparam	join	release	trireg	localparam	chandle	final	protected	virtual
disable	large	repeat	vectored	noshowcancelled	class	first_match	pure	void
edge	macromodule	rnmos	wait	pulsestyle_ondetect	clocking	foreach	rand	wait_order
else	medium	rpmos	wand	pulsestyle_onevent	const	forkjoin	randc	wildcard
end	module	rtran	weak0	showcancelled	constraint	iff	randcase	with
endcase	nand	rtranif0	weak1	signed	context	ignore_bins	randsequence	within
endfunction	negedge	rtranif1	while	unsigned	continue	illegal_bins	ref	
endmodule	nmos	scalared	wire	use	cover	import	return	
endprimitive	nor	small	wor		covergroup	inside	sequence	
endspecify	not	specify	xnor		coverpoint	int	shortint	
endtable	notif0	specparam	xor		cross	interface	shortreal	
endtask	notif1	strong0			dist	intersect	solve	
event	or	strong1			do	join_any	static	

(39)

Summary

- “有用な機能”、“EDAサポート状況”の2つの観点から抽出されたSystemVerilogの言語紹介
 - 12の構文に関して説明を実施
 - 全ての分野で現在使える構文
 - テストベンチとして現在使える構文
 - 早期にサポートを期待する構文
 - SVで追加される予約語を含んだ一覧の説明を実施

JEITA

(40)



EDAアニュアルレポート2004

2005年5月発行

禁無断転載

発行 社団法人 電子情報技術産業協会 電子デバイス部
〒101-0062
東京都千代田区神田駿河台 3-11
三井住友海上駿河台別館ビル
TEL 03-3518-6430 FAX 03-3295-8725

印刷・製本 有限会社 ウイード
〒162-0041
東京都新宿区早稲田鶴巻町 546
TEL 03-3513-5751 FAX 03-3513-5752