

# アーキテクチャと形式的検証の協調 による超ディペンダブルVLSI

戦略的創造研究推進事業  
「ディペンダブルVLSIシステムの基盤技術」

東京大学 大学院情報理工学系研究科

坂井 修一（代表者）

五島 正裕

東京大学 大規模集積システム設計教育研究センター（VDEC）

藤田 昌宏

松本 剛史

東京工業大学 大学院情報理工学研究科

吉瀬 謙二

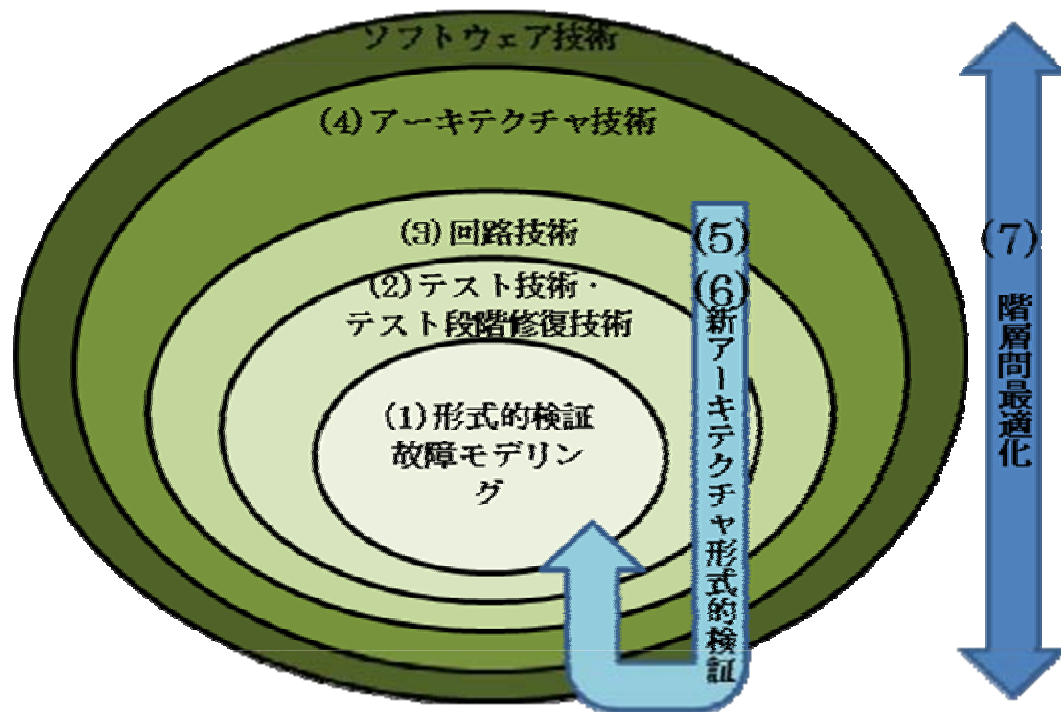
電気通信大学 大学院情報システム学研究科

三好 健文

JST博士研究員

吉田 浩章

# 全体マップ：ディペンダビリティ階層



それぞれの階層で技術開発  
 + 全体を通した最適化  
 + 最新アーキテクチャの検証

Best Effort Design  
 Run Time Recovery

## (1) 形式的検証手法

- 等価性検証ソフトウェア
- ハード・ソフト協調による検証高速化・エミュレータへの応用
- ボトムアップ・トップダウン協調検証
- 算術回路合成・検証・高速化
- 設計解析技術・デバッグ支援

## (2) テスト技術・テスト段階修復技術

- テスト容易化・検証容易化を実現する設計手法
- プログラマブル素子自動挿入

## (3) 回路技術

- タイミング制約緩和回路

## (4) アーキテクチャ技術

- 故障検出・回復機構の提案・実現
- 耐永久故障FPGA
- 耐故障高機能ルータ
- 超ディペンダブルプロセッサ、超ディペンダブルメノコア

## (5)(6) 新アーキテクチャ形式的検証

- ディペンダブルアーキテクチャ技術自体を形式的に検証
- 既存のアーキテクチャ、最新のアーキテクチャを形式的に検証

## (7) 各設計階層間のディペンダビリティ役割分担を最適化

前半3年：方式検討、基本設計、実験システム構築・評価  
 後半2年：プロトタイプ試作と評価、要素技術の統合  
 2010/6/4

# 期待される成果

## ■ VLSIユーザ

- 設計の正しさの向上、リコール減少
- VLSI製作後のバグフィックスや機能修正による利便性向上
- 保証される動作速度の向上
- 宇宙・深海などの環境下での高い信頼性

## ■ VLSI設計・製造者

- 上位で設計の正しさを保ちながら設計の詳細化を行い実装設計につなげる
  - 設計効率一桁向上
- 並列処理・パイプライン処理・キャッシュなどの機構の効率的検証
- 「最悪値の積算」が、「典型値＋回路・アーキテクチャによる補正」によって緩和される

## ■ 成果物・デモ

- 形式検証ツール
  - 等価性検証ツール
  - 上位設計からの製造故障用テスト生成ツール
- テスト段階修復技術
  - インフィールドで論理修正が可能な論理回路生成(論理合成)ツール
- ディペンダブル回路技術
  - 回路(IP)
- ディペンダブルアーキテクチャ技術
  - 要素技術仕様、IP
  - PVTIテストベッド
  - 耐故障テストベッド
- デモ・展示:
  - 形式検証デモ
  - 試作VLSI
  - 超ディペンダブルVLSIテストベッド
- 特許、知財
- 書き物
  - 論文:ジャーナル、国際会議、研究会、全国大会
  - 報告書

# 計画から見た進捗状況

形式的検証	等価性検証ソフトウェア	☆
	ハード・ソフト協調による検証高速化・エミュレータへの応用	☆
	ボトムアップ・トップダウン協調検証	◎
	算術回路合成・検証・高速化	☆
	設計解析技術・デバッグ支援	○
テスト段階の修復	プログラマブル素子自動挿入	○
回路・アーキテクチャ	耐タイミング故障クロッキング	☆
	耐タイミング故障プロセッサ	◎
	耐永久故障FPGA	○
	高機能ルータ	◎
新アーキテクチャ形式検証	超ディペンダブルプロセッサ検証と最適化・ディペンダブルアーキテクチャ一般検証	○

☆: > 100%

◎: = 100%

○: 95~100%

# 成果発表、表彰 (2010.1～2010.5)

---

## ■ 発表 : 38件

- 欧文誌:3件
- 査読付主要国際会議:4件
- 査読付国内会議:3件
- 招待講演 :海外1件
- 口頭講演:海外7件、国内19件
- ポスター: 国内1件

## ■ 受賞 : 7件

- 情報処理学会山下記念賞: 2件
- 情報処理学会システムLSI設計技術研究会優秀論文賞:1件
- 情報処理学会システムLSI設計技術研究会最優秀発表学生賞:1件
- 情報処理学会推奨修士論文:1件
- 情報処理学会推奨卒業論文:1件
- SACSIS2010 優秀ポスター賞:1件

# 協力関係

## ■ チーム内の統合型研究

- ディペンダブル機構の形式的検証
  - タイミングエラー検証回復機構の検証（坂井・五島G＋藤田G）
- ディペンダブル・メニコアの検証（藤田G＋吉瀬G）

## ■ チーム間協力

- 安浦T: カナリアFFの活用法(坂井・五島G)、CADツールの提供(藤田G)等
- 米田T: メニコアVLSI（吉瀬G）

## ■ 国内外研究協力

- 等価性検証ツールの提供・有効性検証(藤田G)
  - NASA, JAXA, Oxford, IIT, U. Florida, New York City College, Bremen Univ.
- プロセッサシミュレータ鬼斬Ⅱ(坂井・五島G)
  - [ディペンダブルプロセッサ全体設計を提供する](#)
- RAMPグループとの協力(坂井・五島G)
  - MIT, CMU, U. Texasなど
- メニコアシミュレータ(吉瀬G)
  - [ディペンダブルメニコア全体設計を提供する](#)

# 出口戦略

---

## ■ 等価性検証ツール

- 国内CADベンダとの連携
- 海外CADベンダへのツール組み込み

## ■ 回路・プロセッサアーキテクチャ

- タイミング故障： CPUベンダへの提供（海外含む）
- 耐故障FPGA： JAXA 斎藤研と相談中

## ■ メモリア

- LSIメーカー、自動車メーカーへの技術移転
  - 領域会議等をきっかけに相談を進めたい

→ 5月20日 総括サイトビジットにて検討

# 今後の展開

---

- 各グループ
  - 要素技術の研究
  - ツール製作、アーキテクチャ設計、プロトタイプ試作
  - 研究協力
  - ツール、シミュレータなどの(一部)公開
- グループ間統合技術
  - ディペンダブル機構(システム)の検証
  - 階層間最適化の検討
- チーム間協力
  - 米田Tと協力してディペンダブル・メニコア技術を集大成
- 出口戦略
  - ツール、アーキテクチャの技術移転



# 形式的検証とテスト段階の修復

---

- 等価性検証フレームワーク FLECの状況
- アーキテクチャ・ディペンダビリティ・機構の検証
- 高性能算術演算回路自動合成の改良
- 上位設計記述によるpost-siliconデバッグフレームワーク
- さらに大規模設計に対して通信を観測するOn-Chip Bufferを自動挿入

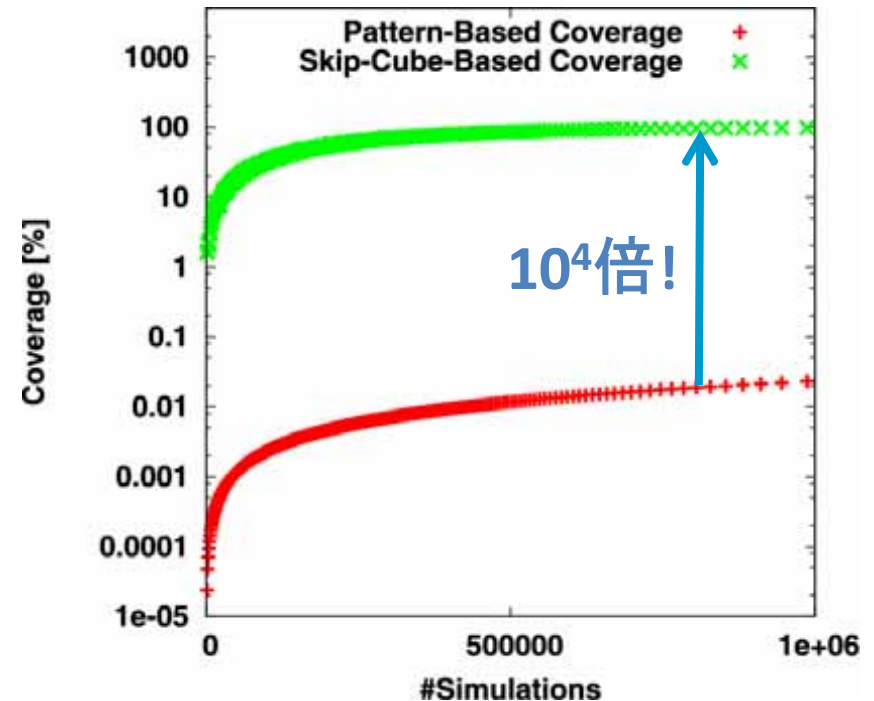
# 等価性検証ツールFLEC

- 企業、大学で評価中(以前と同じ)
  - 実利用への道筋をどう考えるかという段階(後述)
- 准形式的検証(シミュレーションベース検証)の取り込み
  - 従来から提案している「同一の出力が得られる入力の集合を自動的に認識する」技術(日米特許申請中)の利用
- Oxford大学とObservational equivalenceに基づく等価性検証の共同研究を開始
  - ハードウェア設計向けC言語記述
  - 再起呼出しなども含む一般プログラム向け記述
- JAXA開発のシステムレベル設計ツールElegant中の形式的検証ツール(Venus)との融合
  - JAXAから正式に移管を受け、現在統合方法検討中
  - 現在、商品となっているElegantツールには含まれていないが、統合した時点では利用予定
- 算術演算回路検証手法やPost-siliconデバッグなどもFLECフレームワークに順次取込んでいく予定

# 準形式的検証

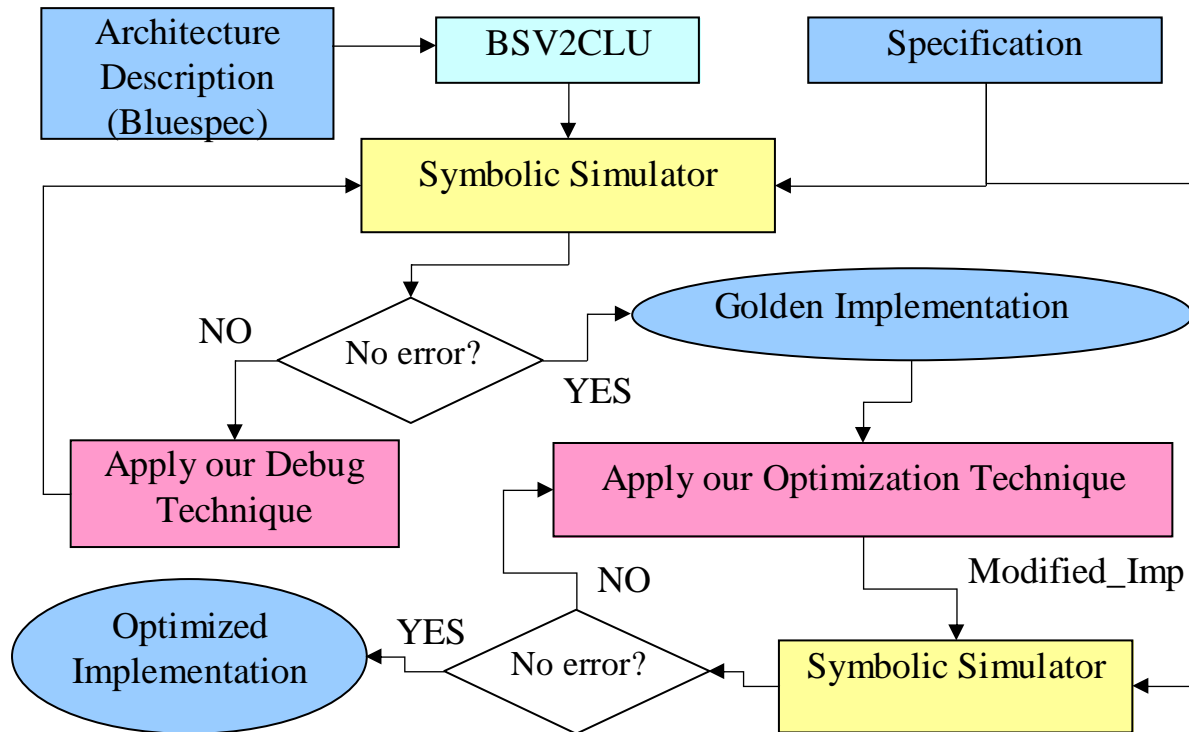
## (知的なシミュレーションベース検証)

- 形式的検証高速化技術のシミュレーションベース検証への応用
- シミュレーションベース検証のカバレッジ向上
  - 形式的検証高速化技術(日本・米国特許申請中)を応用して、カバレッジ(どれだけの入力に対して検証したか)を劇的に向上可能
  - エミュレータを使用した検証など広い範囲に応用可能
- 形式的手法が適用不可能な設計に対する効果的な手法
  - 規模が大きすぎる
  - 記述スタイルが合わない



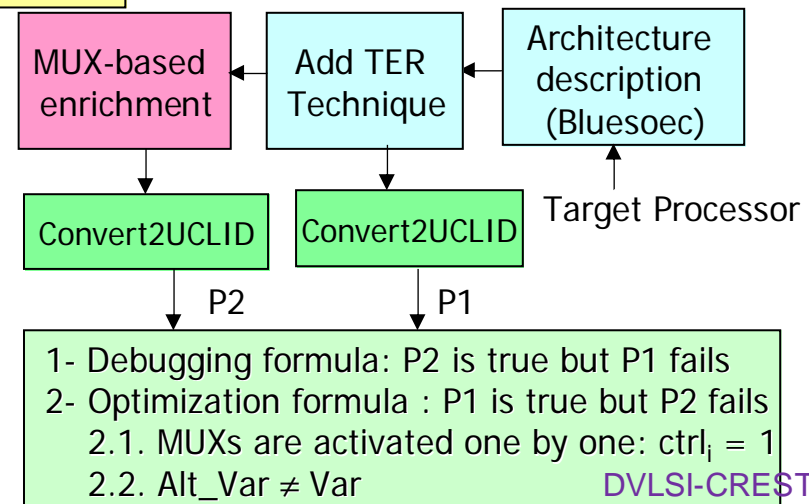
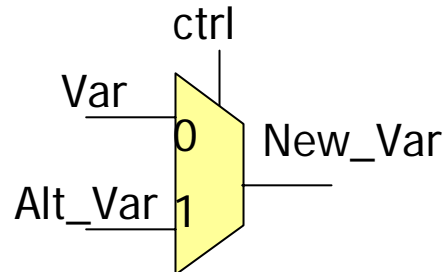
- 16ビット乗算器の検証結果
- 総入力パターン数は $4.3 \times 10^9$
  - 従来技術では $10^6$ のパターンで0.01%のカバレッジしか得られない
  - 提案技術は同パターンでほぼ100%を達成

# アーキテクチャ・ディペンダビリティ・機構の検証： 例：タイミングエラー回復機構（設計）の検証・最適化



- BlueSpec言語に基づくincrementalなプロセッサ設計
- CMUのUCLIDツールを利用した効果的なプロセッサ形式的検証手法を開発

- マルチプレクサ挿入によるデバッグ・最適化
  - P1: with TER but without MUXs
  - P2: with TER and MUXs

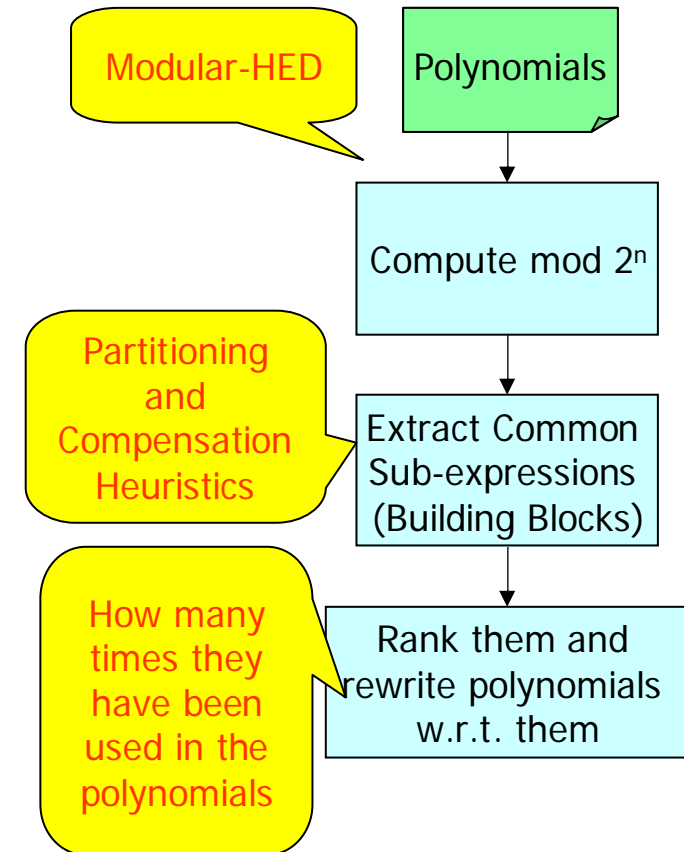


2010/6/4

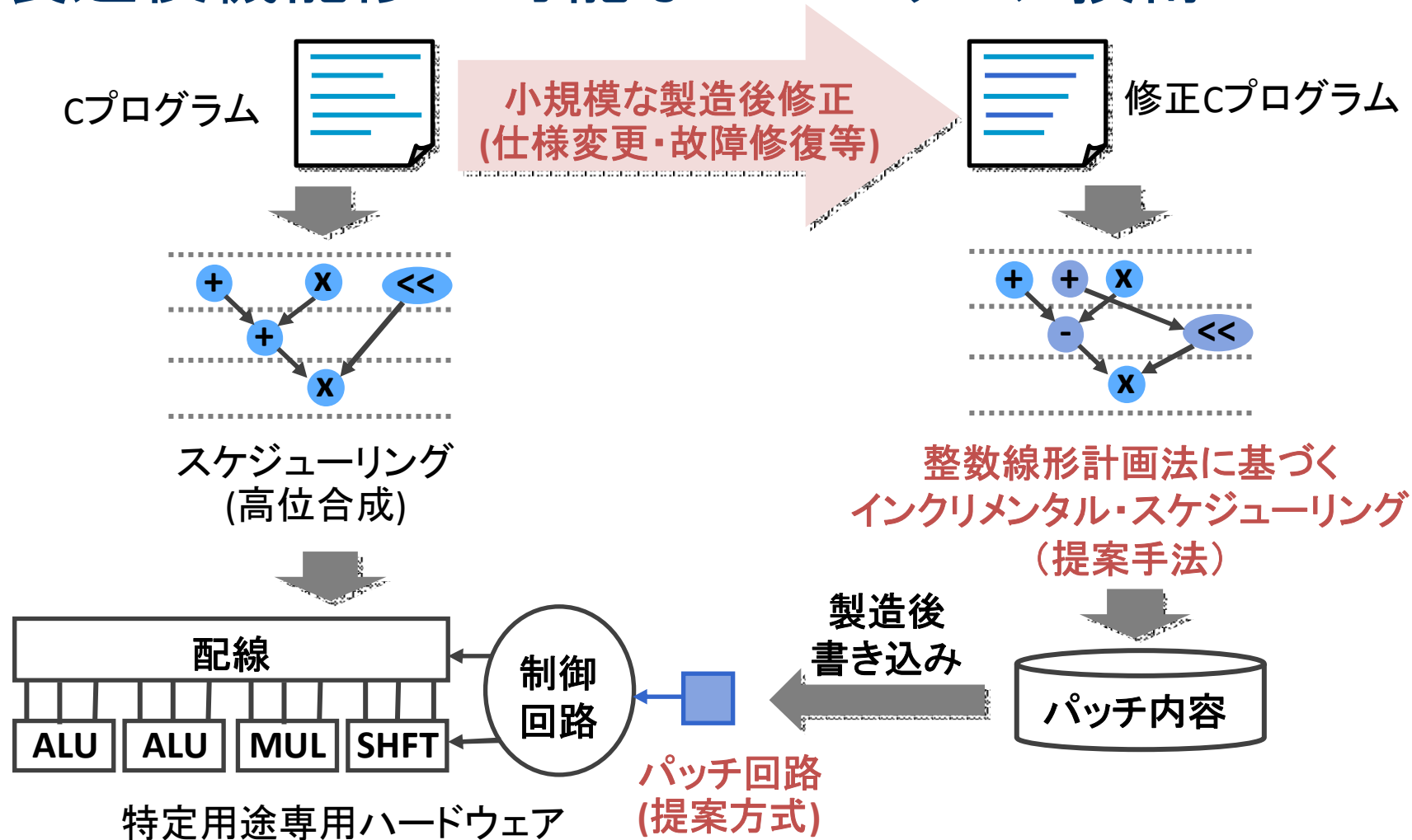
DVLSI-CREST

# 算術演算回路の最適化・検証： より厳密な解法で高品質化

- 多項式の計算において、掛け算と加算の回数を減少させる
- 我々の従来手法(数十%の面積・遅延縮小)
  - 新規開発のModular-HEDを利用して、 $Z_2^n$ 内で最適化
  - 与えられた多項式を  $p_1 \times p_2 + p_3$  と分解して、 $p_3$  を縮小化
- 利点
  - 複数の多変数多項式を同時に最適化
  - Modulo (mod  $2^n$ )を考慮した最適化
  - より強力な 共通式括り出し技術
- さらなる改良
  - 適切な項を追加することで括りだしを強化
  - SMT ソルバーで厳密解(Bremen大学と共同研究)
- 結果
  - 面積・遅延のさらなる数%から十数%の減少



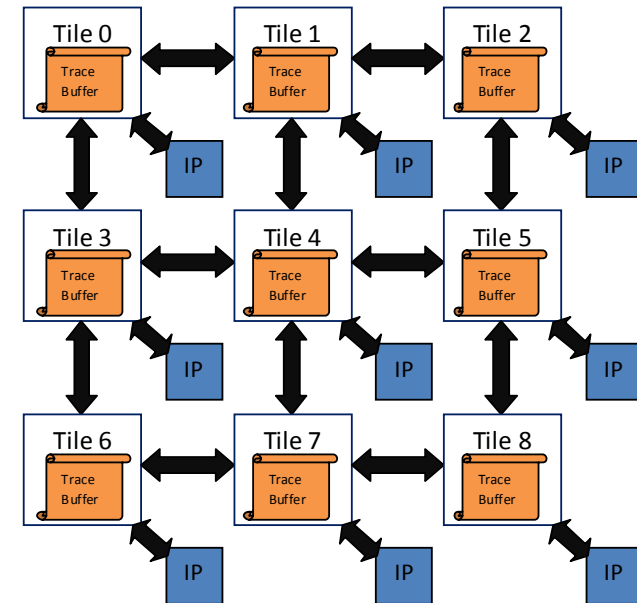
# Post-siliconデバッグ： 製造後機能修正可能なハードウェア技術



回路全体をプログラマブルにするのではなく、制御回路を部分的に変更可能にすることで専用ハードウェア並みの効率・性能を実現

# NoCに対するPost-Silicon デバッグ

- 現実的な仮定に元で、NoCのバグや不具合を検出し、デバッグする技術
  - 分散バッファ、異なる多数のクロックドメイン、部分的に非同期通信
- 従来の手法
  - Global time: 全体を同期するのは無理
  - Global system snapshot:: Global メッセージが必要
- 我々の手法
  - ローカルバッファの情報のみを利用し、グローバルな時間順序関係を推察する
    - グローバルはメッセージは不用
    - 必要に応じて、ローカルに通信
  - バッファには通信イベントの順序のみ保存
    - 時刻は不用
- 米田グループらと協力していきたい



# 実利用へ向けて

- 研究開発した技術での社会での利用
  - 論文などを通じたもの(人任せ)
  - 特許(実際のライセンスはツールでないと難しい)
  - ツールの提供(サポートの問題)
- 次の3つの考え方で努力していく
  - 国内で積極的にEDAツールを開発・販売しているグループとの協調
    - ツールのシナジー効果を中心に柔軟に取り組む
  - 米国検証EDAベンダーへの技術移管
    - 様々な研究成果を柔軟に考える
  - 実際の設計プロジェクトへ参加し、検証を自分たちで行っていく
    - アドバイザーの方の協力も頂いて、企業などに積極的にアピール
- ソフトウェア検証への発展も考慮すべき？
  - ハードウェア・ソフトウェア協調設計を対象



# 「検証とテスト」21年度以降研究成果のまとめ

## 高信頼・高性能計算

高性能カスタム  
算術演算器生成

Overclocking +  
自動エラーリカバリ

- 多項式最適化手法・回路面積／遅延~40%縮小 (DAC09, ICCAD09, 検証もまとめてIEEE TCAD accepted)
- 大規模ドントケア利用による50%回路最適化 (Memocode09)
- タイミングエラーからの自動回復新規回路考案・30%高速化を実証(FPGA10)

## Post-silicon支援

Pre-silicon debug

Post-silicon debug

Post-silicon patch

- C言語、トランザクションレベルpost-siliconデバッグ支援(ICC09, ATS09)
- LUT利用によるpost-silicon修正技術(国内会議)
- 高位記述からの故障検出用ATPG技術・従来比数十倍高速(ASPDAC10, IEEE TCAD May 10)
- 算術演算回路検証／デバッグ技術・従来不可能であった32ビット以上の演算回路に対応(TCAD May 09)
- 通信系に特化した検証手法・従来不可能であったon-chipプロトコル変換器の形式的検証(ISQED10)
- マイクロプロセッサのタイミングエラー自動回復アルゴリズムの検証／最適化・数秒から数時間(ARC10)

## 形式的検証

等価性検証ツール

アーキテクチャ  
アルゴリズム  
検証・最適化

# 回路技術・アーキテクチャ技術

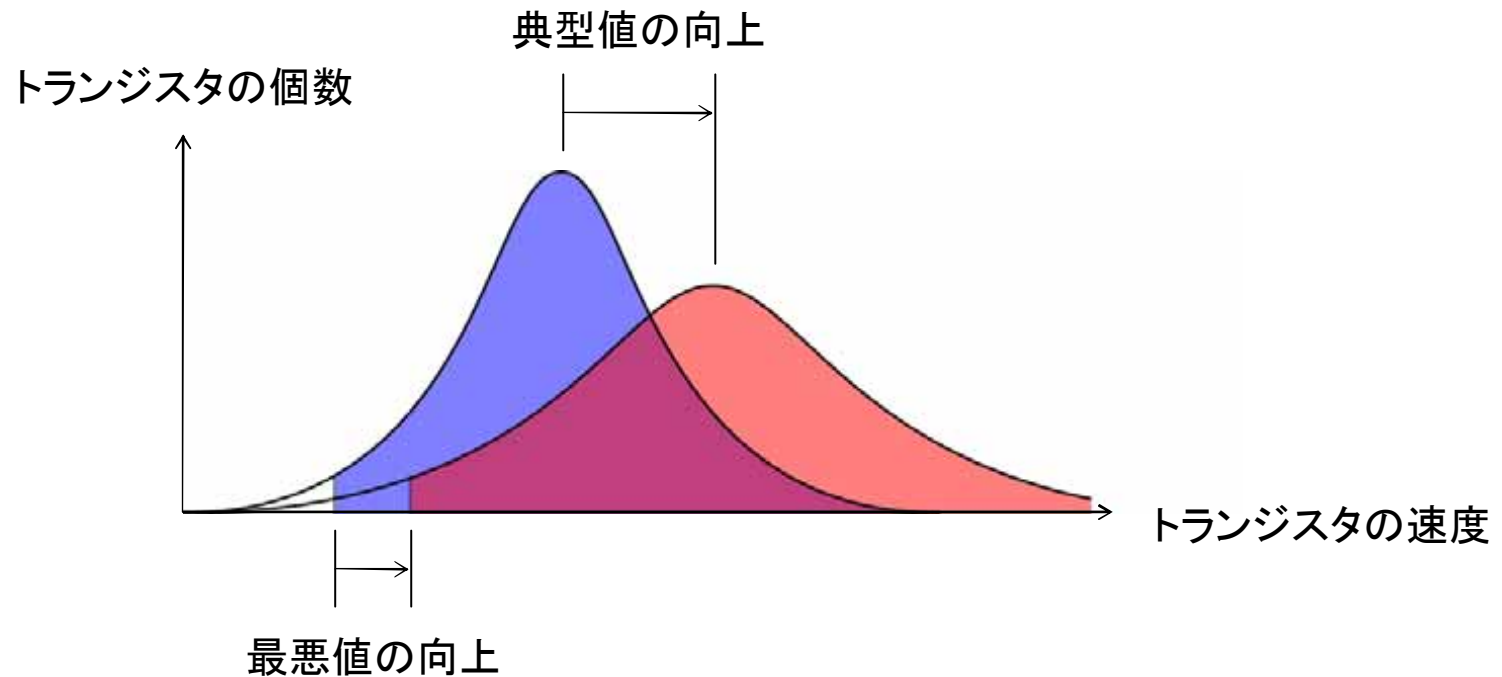
---

- 耐タイミング故障
  - ◆ タイミング制約を緩和するクロッキング方式
  - ◆ 耐タイミング故障スーパスカラ・アーキテクチャ
- 耐永久故障 FPGA アーキテクチャ
  - ◆ 制御回路を含む仮想化：穴のない耐故障性
- ディペンダビリティ向け多機能ルータ
  - ◆ 送受信パケットのレベルで冗長実行を実現

# タイミング制約の緩和 / 動的故障検出・回復

## ■ LSI の製造ばらつき

- ◆ 従来の最悪値に基づく設計は悲観的になり過ぎる

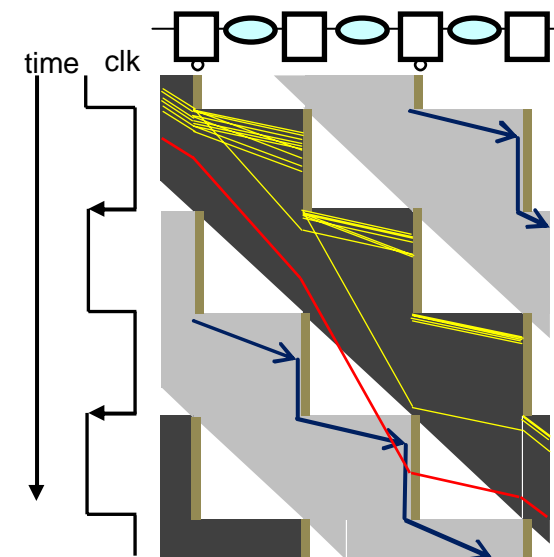
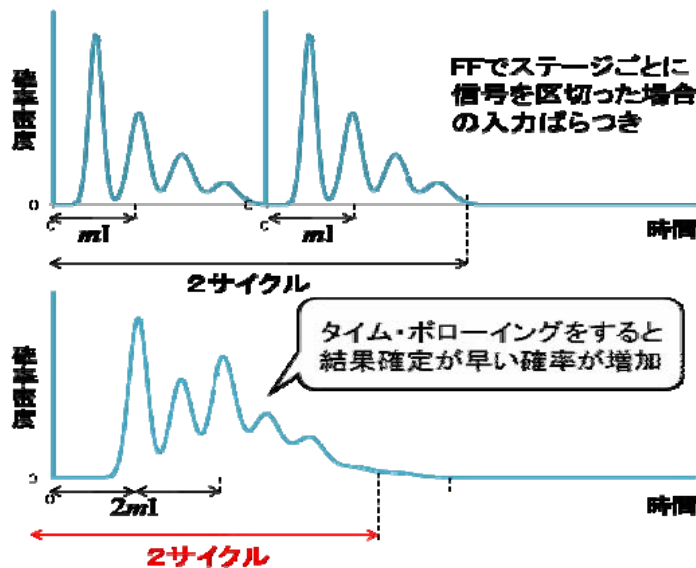
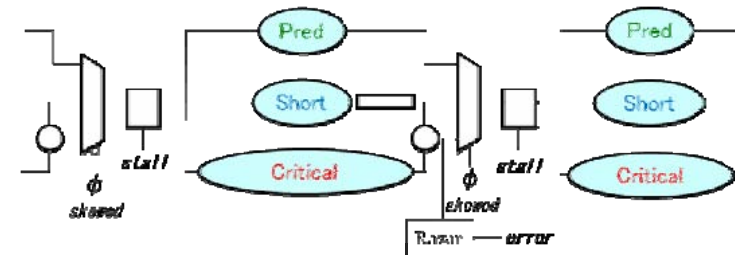


# タイミング制約を緩和するクロッキング方式 (1/2)

	最悪ケース設計	動的故障検出・回復
<p><u>通常</u></p> <p>単層 FF</p> <ul style="list-style-type: none"> <li>● 借金の概念がない</li> <li>● 赤字が出たら暴走</li> </ul>	<p>Razor</p> <ul style="list-style-type: none"> <li>● 動作時に借金可能</li> <li>● 1回でも赤字が出たら破綻・再建</li> </ul>	
<p><u>静的タイム・ボローイング</u></p> <p>二相ラッチ</p> <ul style="list-style-type: none"> <li>● 設計時に借金可能</li> <li>● 動作時に赤字が出たら暴走</li> </ul>	<p><u>動的タイム・ボローイング</u></p> <ul style="list-style-type: none"> <li>● 動作時に借金可能</li> <li>● 赤字が累積したら破綻・再建</li> </ul>	

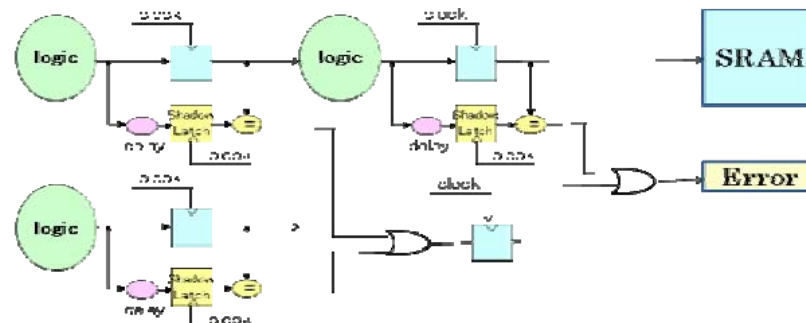
# タイミング制約を緩和するクロッキング方式 (2/2)

- 入力ばらつき
  - ◆ 実効遅延は入力により大きく変動する
- 入力ばらつきを逆に利用
  - ◆ 遅延小の頻度が高い： 普段は高クロック
  - ◆ 稀にタイミング故障： 検出・回復



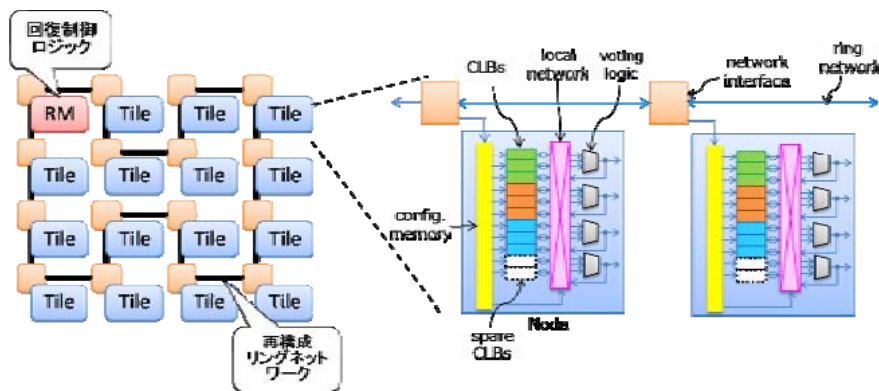
# 耐タイミング故障スーパスカラ・アーキテクチャ

- 基本的な考え方:
  - ◆ アーキテクチャ・ステートの保護
  - ◆ ロールバック
- 既存手法:パイプライン・フラッシュによるロールバック
  - ◆ 制御系の故障に対応できない ⇒ 単純なスカラ・プロセッサのみ
- 提案手法:リセットによるロールバック
  - ◆ 制御系を含むあらゆる故障に対応 ⇒ スーパスカラ・プロセッサにも対応可能
- 進捗:
  - ◆ 新規設計開始した高性能スーパスカラ・プロセッサへ実装中



# 耐永久故障 FPGA アーキテクチャ

- 基本: TMR (Triple Modular Redundancy) + DPR (Dynamic Partial Reconf)
  - ◆ 多数決で故障を検出 → 動的再構成(再構成中も動作は継続)
- 提案:
  - ◆ 多数決回路など ⇒ 粗いルール of 専用回路で
  - ◆ 再構成を行うマネージャ ⇒ ユーザ・ロジックで
- メリット:
  - ◆ あらゆる箇所の故障を回復可能
  - ◆ 追加 HW わずか: 通常用途 FPGA = 高信頼用途 FPGA

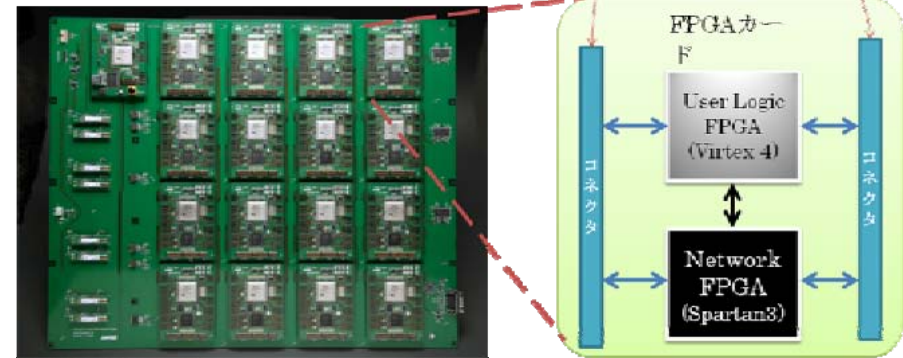


	通常用途時	高信頼用途時										
User Logic Voting Unit (ICAP)	<table border="1"> <tr><td>Software</td></tr> <tr><td>User Logic</td></tr> <tr><td>Hard-Wired Logic</td></tr> <tr><td>Config Subsystem</td></tr> </table>	Software	User Logic	Hard-Wired Logic	Config Subsystem	<table border="1"> <tr> <td>Software</td> <td>RM</td> </tr> <tr> <td>Voting User Logic</td> <td>RM</td> </tr> <tr><td>Hard-Wired Logic</td></tr> <tr><td>Config Subsystem</td></tr> </table>	Software	RM	Voting User Logic	RM	Hard-Wired Logic	Config Subsystem
Software												
User Logic												
Hard-Wired Logic												
Config Subsystem												
Software	RM											
Voting User Logic	RM											
Hard-Wired Logic												
Config Subsystem												
Hard-Wired Voting Unit	<table border="1"> <tr><td>Software</td></tr> <tr><td>User Logic</td></tr> <tr><td>Hard-Wired Logic</td></tr> <tr><td>Config Subsystem</td></tr> </table>	Software	User Logic	Hard-Wired Logic	Config Subsystem	<table border="1"> <tr> <td>Software</td> <td>RM</td> </tr> <tr> <td>User Logic</td> <td>RM</td> </tr> <tr><td>Hard-Wired Logic</td></tr> <tr><td>Config Subsystem</td></tr> </table>	Software	RM	User Logic	RM	Hard-Wired Logic	Config Subsystem
Software												
User Logic												
Hard-Wired Logic												
Config Subsystem												
Software	RM											
User Logic	RM											
Hard-Wired Logic												
Config Subsystem												

# テストベッド



DVFS 機能を持つ  
FPGA テストベッド



耐永久故障 FPGA  
テストベッド



# 超ディペンダビリティ支援高機能ルータ

高機能ルータを核として送受信パケットのレベルで冗長実行を実現するシステムの開発

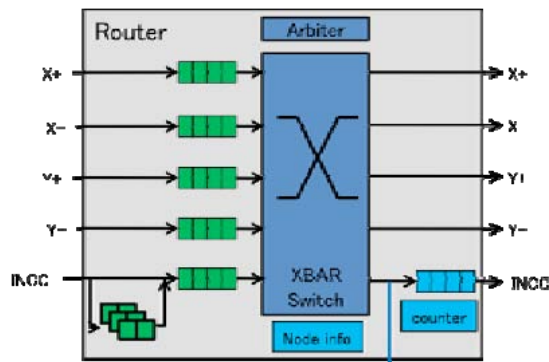
## ● 高機能ルータアーキテクチャ

冗長実行自動支援のためのパケットの複製、  
同一性検出、マージの機能を実現

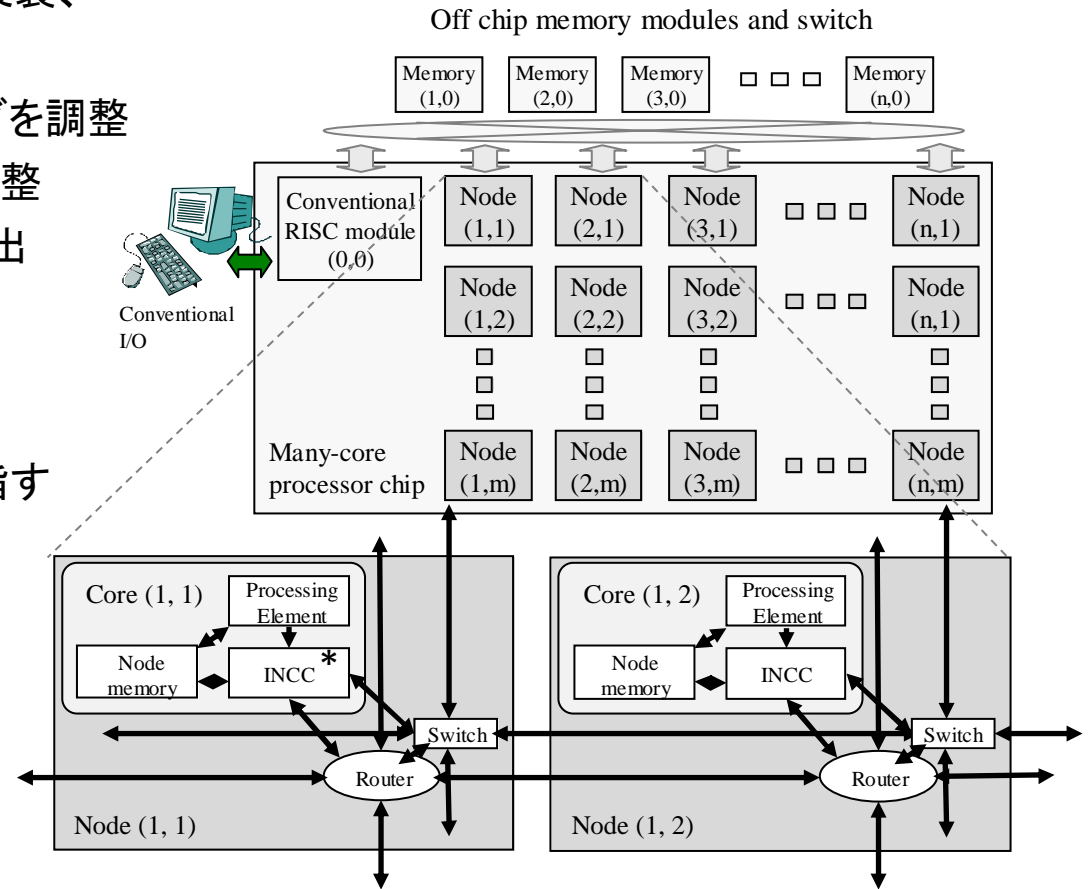
- ルータでパケットを受信するタイミングを調整
- ルータでパケットを送信する宛先を調整
- ルータでパケットを比較してエラー検出

## ■ SmartCoreシステム

多数のコアと高機能ルータによって  
ディペンダビリティ向上と速度向上を目指す



ルータアーキテクチャ



\* Inter Node Communication Controller

高機能ルータをもつマルチコアシステムアーキテクチャ

# 超ディペンダビリティ支援高機能ルータグループ成果

## ■ 高機能ルータアーキテクチャの開発

佐藤真平ほか：“SmartCoreシステムによるメニーコアプロセッサの信頼性向上手法”，情報処理学会研究報告2009-ARC-186, pp. 1-6 (January 2010).

佐藤真平ほか：“メニーコアプロセッサのオンチップネットワーク性能を向上させるSmartCore システム”，先進的計算基盤システムシンポジウムSACSIS2009論文集, (May 2009).

## ■ マルチコアシミュレータの改良

- 高機能ルータアーキテクチャの評価のための機能追加

Koh Uehara et al: “A study of An Infrastructure for Research and Development of Many-Core Processors”, Workshop on Ultra Performance and Dependable Acceleration Systems (UPDAS), (Dec. 2009)

## ■ 評価のためのFPGAシステムの整備

- ベースシステム: ScalableCoreの試作
- SmartCoreシステムシミュレーション環境の構築について検討

高前田伸也ほか：“メニーコアアーキテクチャのHW評価環境ScalableCoreシステムの開発”，先進的計算基盤システムシンポジウムSACSIS2010論文集, (May 2010).

Shinya Takamaeda et al: “ScalableCore : High-Speed Prototyping System for Many-Core Processors”, IEEE Symposium on Low-Power and High-Speed Chips COOL Chips XII, (Apr. 2009).

Shinya Takamaeda et al: “ScalableCore : The Concept of Practical and Low-Cost Prototyping System for Many-Core Processor Research and Educations”, The 4<sup>th</sup> Workshop on Architectural Research Prototyping (WARP 2009), (June 2009)

## ■ マルチコア開発支援ソフトウェア

- タスクのコア配置最適化手法の検討

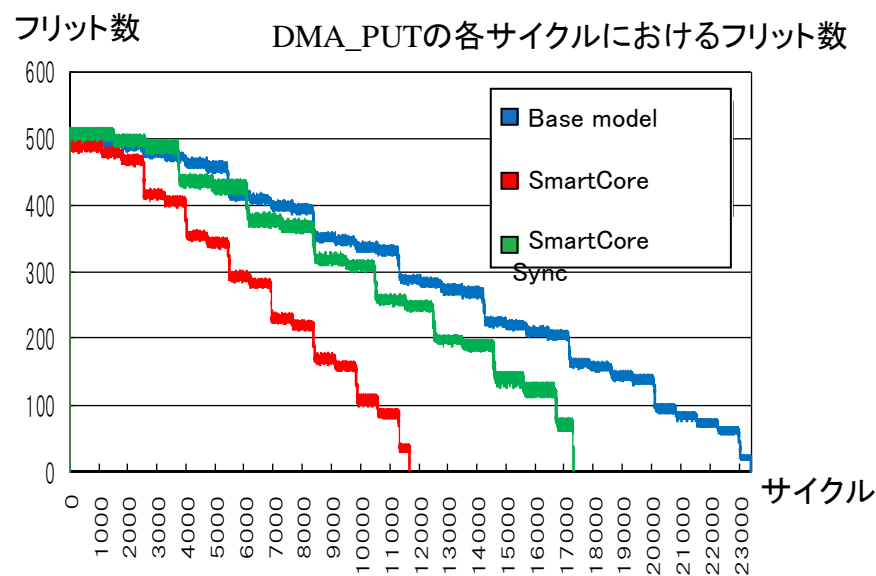
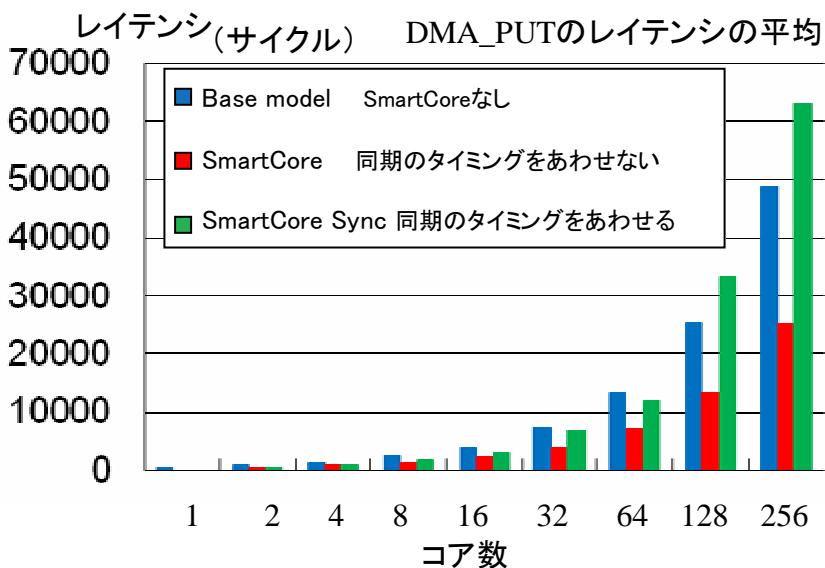
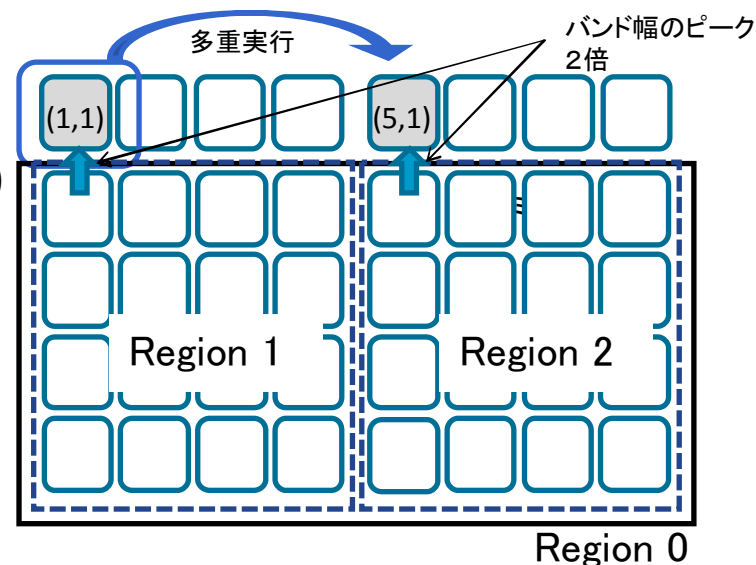
佐野伸太郎ほか：“メニーコアプロセッサの性能向上を達成するパターンに基づいたタスク配置手法”，先進的計算基盤システムシンポジウムSACSIS2010論文集, (May 2010).

# SmartCoreシステムによるプロセッサ間の通信バンド幅向上

## バンド幅向上

Region0中のノードからNode(1,1)へのアクセス  
バンド幅を向上するためNode(1,1)の動作を、Node(5,1)  
で多重実行

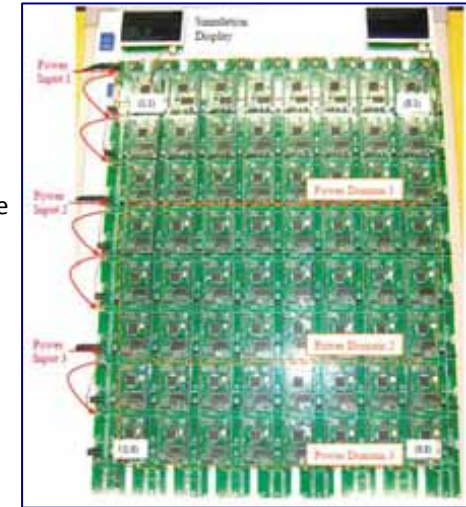
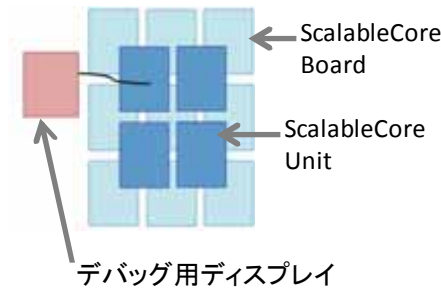
Region1中のノードはNode(1,1)へ、Region2中の  
ノードはNode(5,1)へアクセスすることにより、  
バンド幅がピークで2倍



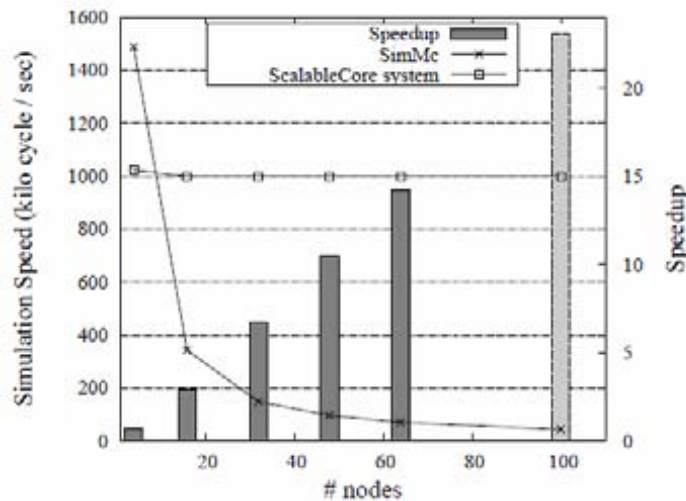
# 超ディペンダビリティ支援高機能ルータ評価環境

## FPGAシステムによる評価環境 ScalableCoreシステム

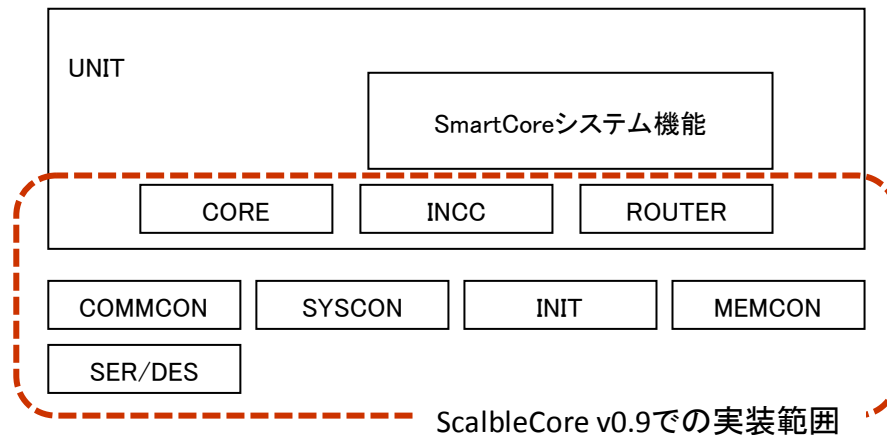
- SmartCoreシステム評価のベースシステムとして
- ScalableCore v1.0
  - Unit間の通信速度: 45Mbps
  - SWシミュレータと同じプログラムの実行が可能
  - プログラムは起動時にFWにより配布
- SWシミュレータとの比較による評価



ScalableCoreシステムを構成するユニットと8x8の構成例



SWシミュレーション環境とのシミュレーション時間の比較  
100コアのシミュレーション時に23倍の高速化を実現可能



ScalableCoreをベースとする  
SmartCoreシステム評価環境の構築モデル

# 高機能ルータ：研究項目

---

- 高機能ルータアーキテクチャの改良
  - 冗長実行をおこなうコアの個数や構成を動的に変更する際のオーバーヘッドの低減
  - ソフトウェアシミュレータを用いた提案アーキテクチャの評価
- ScalableCoreシステムの改良および機能追加
  - 大規模なFPGAを搭載
  - 100枚程度のFPGAボードを接続した大規模のFPGAシステムを構築
  - 高機能ルータアーキテクチャを搭載するルータ、コア等を実装し、提案方式の実現可能性の検証および性能評価を実施
- アーキテクチャ検証を形式的に行う手法の検討