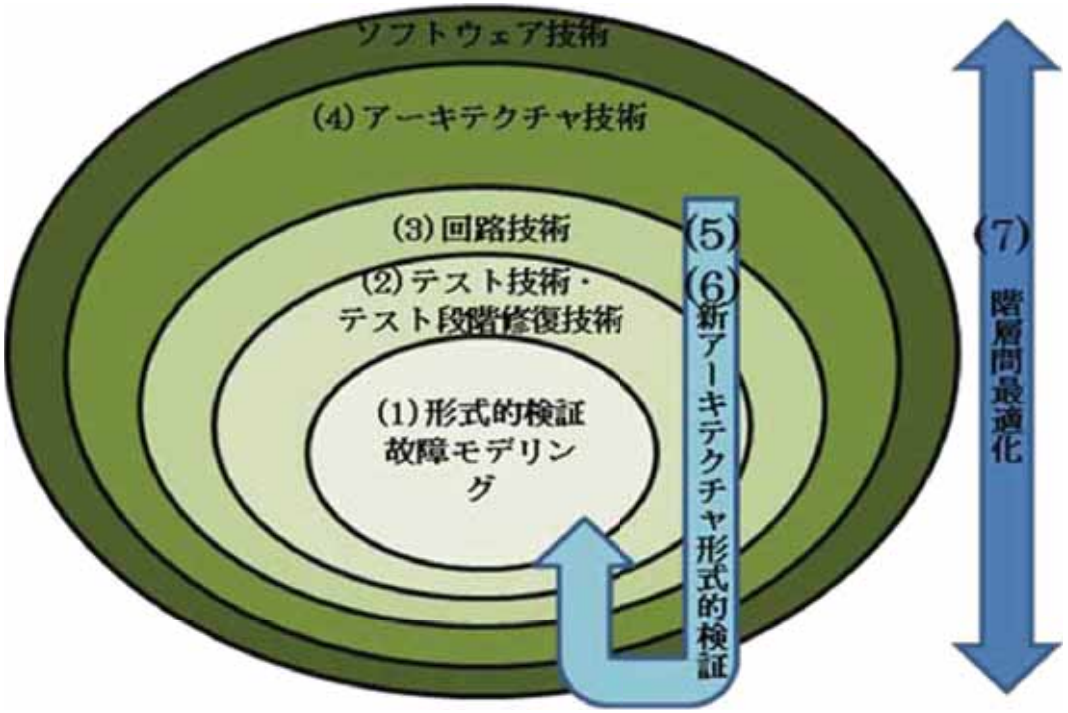
アーキテクチャと形式的検証の協調による超ディペンダブルVLSI

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
東京工業大学　大学院情報理工学研究科　吉瀬　謙二

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

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
2009/4/18

(1) 形式的検証手法

- C/C++言語ベースVLSI高位設計
- 対話・自動ドキュメント化のための要素技術

(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
- フィールドプログラマブル性を部分的に導入可能な合成手法 = テスト段階でデザインミスをとる

(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
- 制御部を含めたFPGA仮想化
- 耐故障高機能ルータ
- 超ディペンダブルプロセッサ

(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

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・深海などの環境でも高い信頼性をもって情報処理ができるようになる
- 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」ことができるようになる
    - 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が効率的に検証できるようになる
  - 「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

①H20年度 研究進捗報告
②H21年度研究計画概要

形式的検証とテスト段階の修復

　グループ間連携：　アーキテクチャ形式検証

回路技術・アーキテクチャ技術

形式的検証とテスト段階の修復

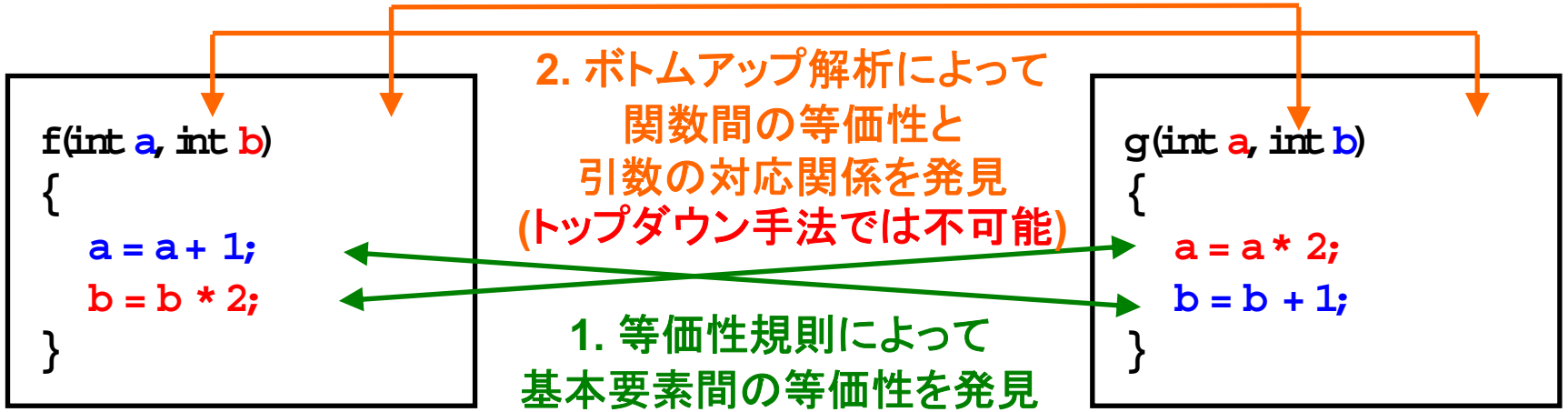
- 形式的等価性検証／解析ツールの開発状況
- ボトムアップ検証
- 算術演算回路の検証・デバッグ手法
- アーキテクチャ形式検証
- 検証における技術的な関連
- テスト・修復容易化のための上位設計解析手法

形式的等価性検証/解析ツールの開発状況

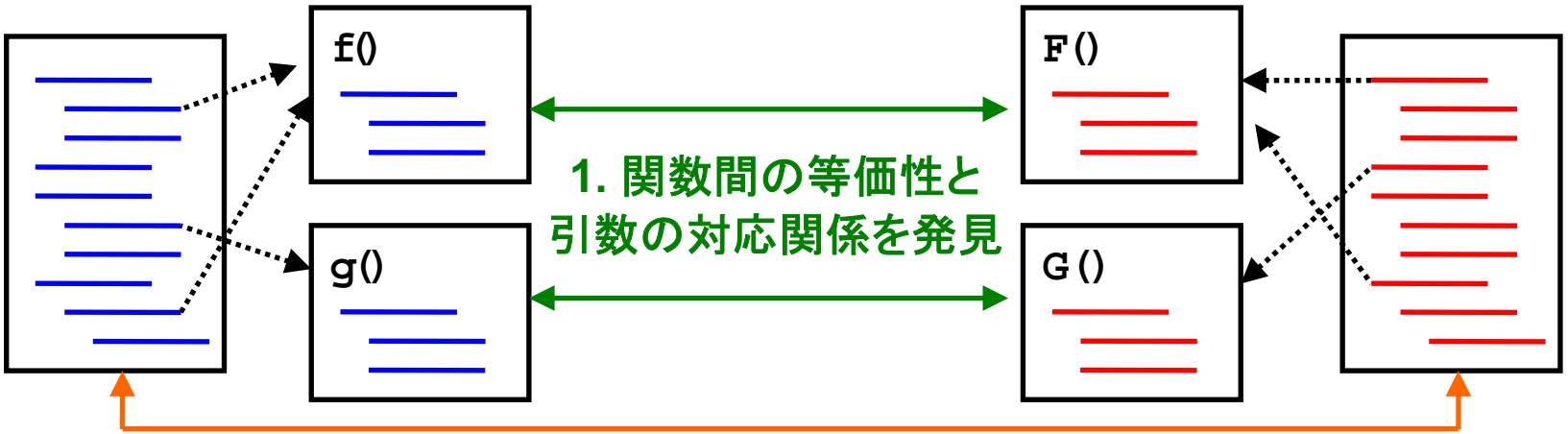
- ツールの特徴と開発状況
  - 拡張システム依存グラフによる統一的な内部表現を持ち、形式的検証手法、テスト容易化のための解析手法、設計理解のための解析手法を実装
  - コード量：C++言語で10万行程度
- 5企業（設計グループ）へツールリリース。内2グループで積極的に評価、その結果のバグ取り
  - 商用高位合成ツールを利用しているグループと、自社ツール、または手動でRTL設計しているグループ
  - 企業からのフィードバック
    - SystemC言語、RTL言語のサポート
    - C言語レベルの等価性検証、特にループ最適化に関する検証
    - 基本検証速度のさらなる高速化
    - マルチメディア系LSIのための算術演算回路検証能力強化（C言語レベル、RTL/回路レベル）
    - 等価性検証以外に対する要求：　C言語レベル設計性能評価、知的シミュレーションパターン生成、シミュレーションベース検証との融合、モデルチェッキングのためのプロパティ自動生成など多数
- 今年度の予定
  - SystemC言語、RTL言語フロントエンドの開発
    - 評価可能な例題の増加が期待できる
  - 実装された各検証手法の協調による検証の高速化
    - 基本手法と後述のボトムアップ検証手法、算術演算の検証手法を組合わせる
  - ループ最適化に対する等価性検証手法の開発
    - HW, SWに関わらず行われており、企業からの要望が強い

ボトムアップ検証の利点

(1) 内部等価(対応)点の発見



(2) 階層的な探索による効率的な検証



ボトムアップ検証

- 比較する2つの設計が「近い」場合に極めて有効
  - 多くの例題では、大規模設計でも、記述が異なるのは一部に限られる
- 基本手法の提案と例題による実証[1]
  - 変数名の対応を仮定
  - IDCTなどでは、各種最適化結果を数十秒で等価性検証可能
- ランダムシミュレーションによる内部等価点の効率的な絞込み[2]
  - 従来扱えなかった記述の変更にも対応
  - 数分かかっていたものが、数秒程度で処理可能

```
int ex1(int a, int b) {
        return a - b;
}
```

元の設計

```
int ex2(int b, int a) {
        return b - a;
}
```

変数名が変更された場合

```
int ex3(int c, int d) {
        return c - d;
}
```

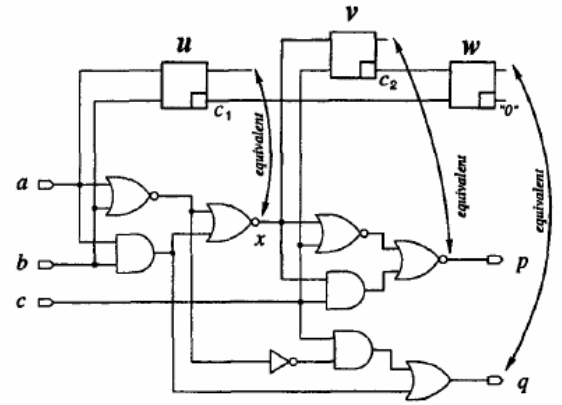
変数の対応が発見されない場合

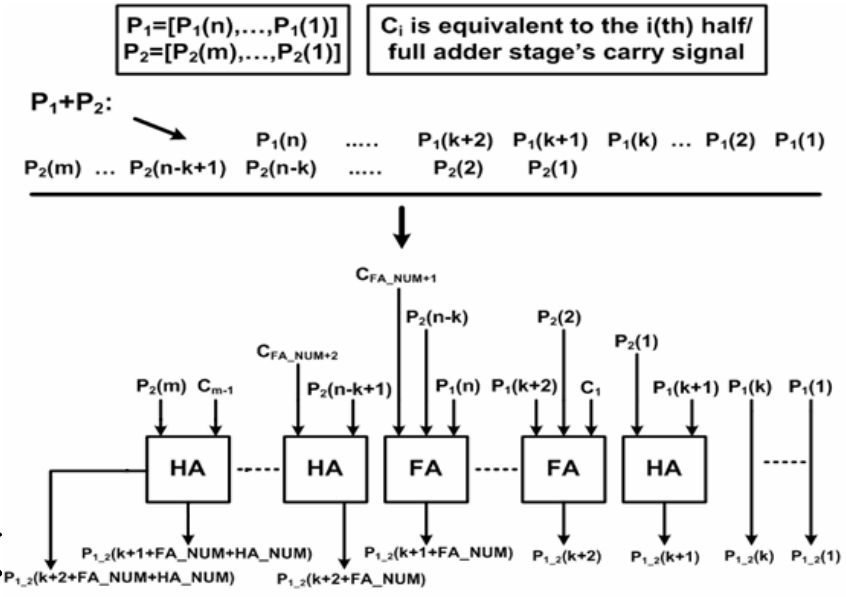
- 今後の予定
  - 形式的検証ツールへの統合・他検証手法との組み合わせ
  - 実用的例題を用いた評価

[1] “Rule-Based Approaches for Equivalence Checking of SpecC Programs,” ACM/IEEE MEMOCODE, 2008
[2] Improving the Accuracy of Rule-based Equivalence Checking of System-level Design Descriptions by Identifying Potential Internal Equivalences,” ISQED, 2009

算術演算回路の検証・デバッグ手法(1)

- 論理回路から半加算器を抽出することで掛算器など算術演算回路の検証を行う[3]
  - 算術演算回路を加算器から構成されるネットワークとして定式化
  - 与えられたRTL/回路とライブラリに定義されたものとの対応を取る
    - 広範囲の掛算器に適用できる部分加算標準形式を考案
    - 64ビット掛算器を数十秒で検証（現在最も高速）
    - X*Y+3*Zなどの一般算術式を計算する組合せ回路も検証可能
    - 自動デバッグへの適用も可能
  - 今後形式的検証ツールと統合していく

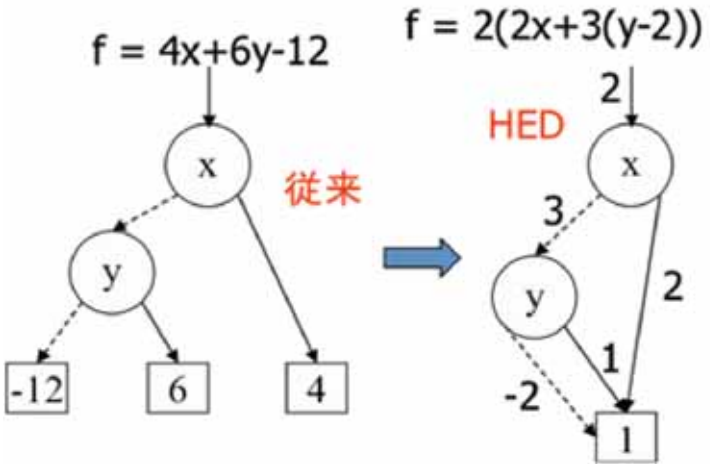




[3] “Arithmetic Circuits Verification without Looking for Internal Equivalences,” ACM/IEEE MEMOCODE08

算術演算回路の検証・デバッグ手法(2)

- 高位設計記述における多項式間やRTL/回路との等価性検証手法[4]
  - 決定グラフHEDを拡張して利用
  - Modulo演算等価性を自動認識
    - $a^2-b^2 = a^2+7b^2 \bmod 2^3$
  - 高位記述からRTL/回路までを統合して扱える初めての手法
  - 高位設計では、従来手法で数時間かけても終了しなかった検証が数分で処理可能



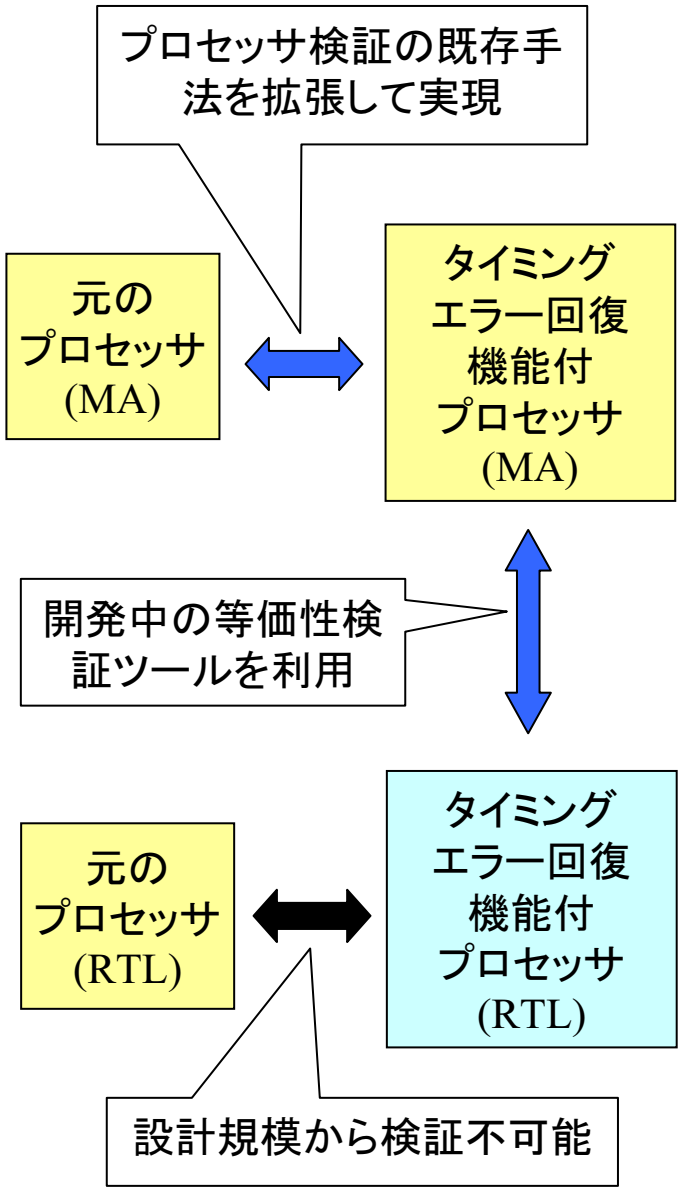
- 算術演算回路の自動デバッグ技術[5]
  - ライブラリの基本回路との対応を自動的に取ることにより、回路中のバグを認識し、自動修正
    - ゲート対応、接続誤りなどであれば、複数バグがあっても、自動的にほぼ完全に修正
  - 動的故障自動回避手法への拡張を検討予定
- 形式的検証ツールとの統合を進める

[4] “A unified framework for equivalence verification of datapath oriented applications,’ IEICE TRANS. INF. & SYST., VOL. E92-D, No. 5 MAY 2009.

[5] “A Formal Approach for Debugging Arithmetic Circuits,” IEEE TRANSACTION ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, VOL 28, NO 5, MAY 2009.

アーキテクチャ検証

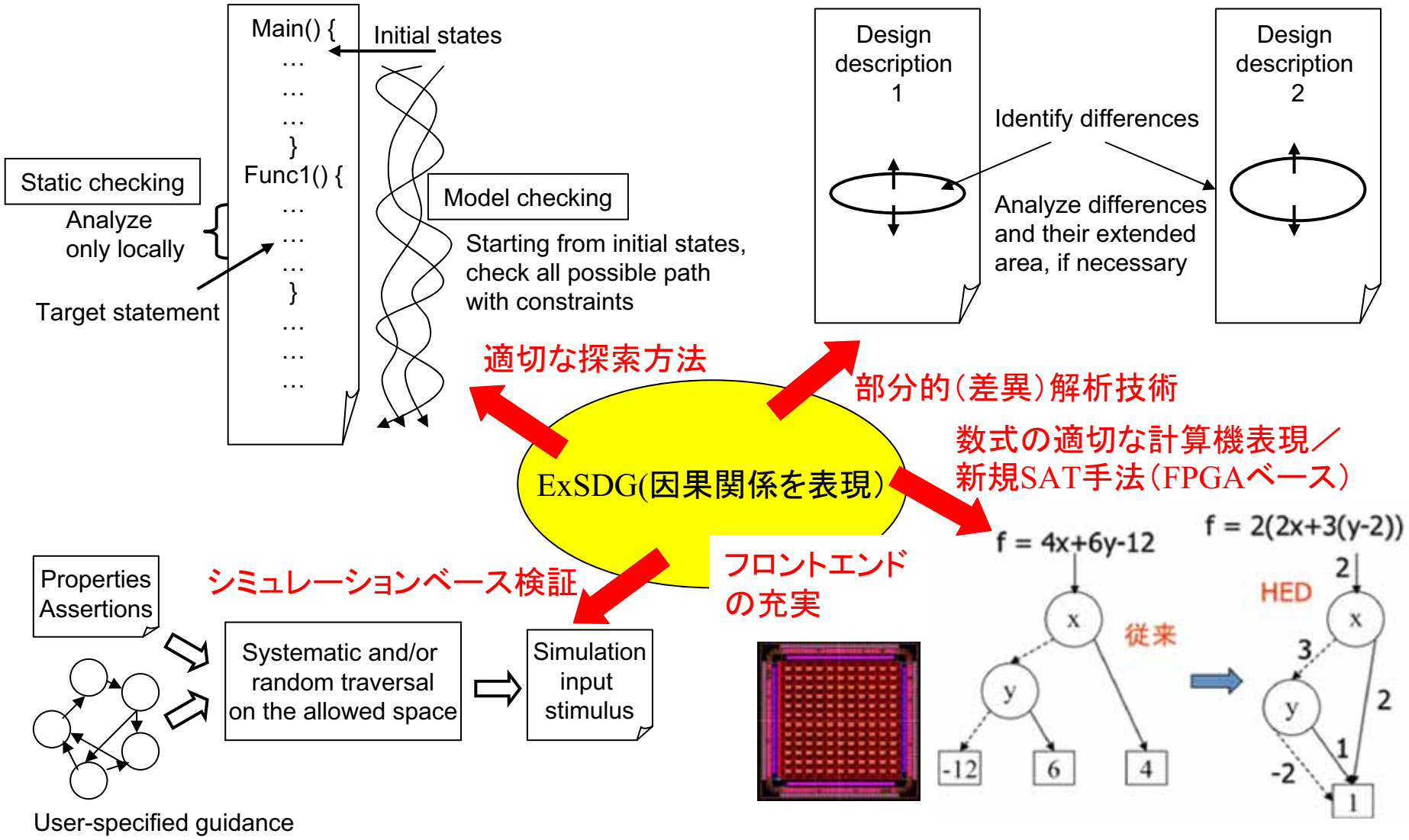
- 2つのプロセッサの動作の等価性を証明する既存手法をベースに開発中の等価性検証ツールと組合せることで、タイミングエラー発生時の自動リカバリー動作を形式的に検証
  - アルゴリズム（マイクロアーキテクチャ）レベルの動作の検証作業
    - タイミングエラーからの回復制御ありとなしの2つの設計の動作の等価性を検証する
      - 検証効率の観点からマイクロアーキテクチャレベルで行う
      - RTLで直接行うことは設計規模から不可能
    - マイクロアーキテクチャレベルとRTL設計との等価性は、開発中の等価性検証ツールで行う
  - まず、仮想的なSuperscalarプロセッサを例として、技術の有効性を実証
    - その後、坂井グループの設計に適用
  - 既存プロセッサを拡張する場合の検証手法として一般化が可能
    - ミシガン大学の提案手法も検証予定
  - 基本検討を終了し、マイクロアーキテクチャレベルの動作記述を作成中



MA: マイクロアーキテクチャ

検証における技術的な関連

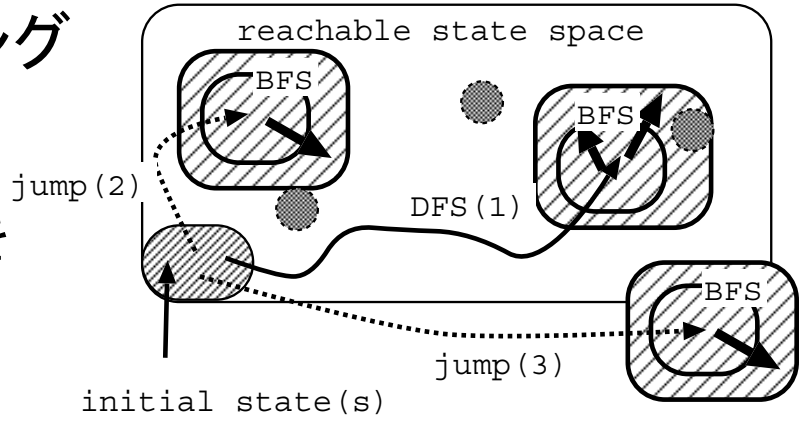
- ベースとなるのは、形式的な設計解析技術と内部表現（ExSDG）
- 各種技術を適切に組合せて利用し、ツール化



テスト・修復容易化のための上位設計解析手法

- 記号・具体混合シミュレーション[6]
  - 上位設計において、指定された条件を満たす入力パタンを生成する
    - テストにおけるテストパタンとしても利用できる
  - 従来の固定値を用いた具体シミュレーションと，記号式を用いた記号シミュレーションを同時に実行
  - 分岐カバレッジを向上させ，アサーション違反を起こす入力値を探索
  - SpecCで3000行程度の記述に対して
    - 未初期化変数の値の参照，配列オーバーラン等の検出に成功
    - 固定値入力で10サイクル程度の記号シミュレーションに成功

- 反例やバグトレース解析のための実行時間を考慮した動的スライシング技術を開発
  - C言語レベルでバグの解析が可能
  - 今後実験を重ね、反例から自動修復を行う手法の確立を目指す
    - バグ位置の推定手法、バグ部分を正しい設計に置換する手法が必要



[6] “An Interactive Verification and Debugging Environment by Concrete/Symbolic Simulations for System-level Designs,” Asian Test Symposium, 2008

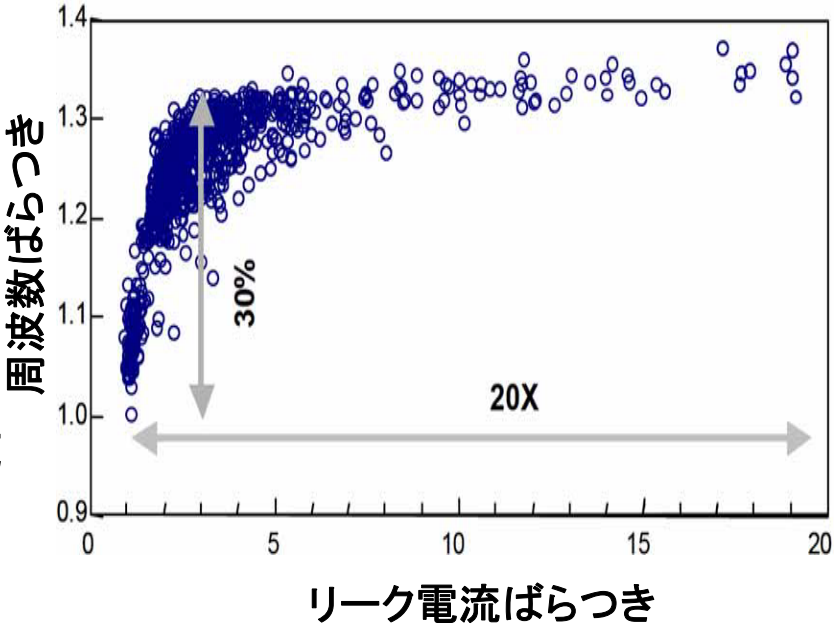
回路技術・アーキテクチャ技術

- タイミング制約を緩和するクロッキング方式
- 故障耐性を持つスーパスカラプロセッサ
- 耐永久故障 FPGA 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

タイミング制約の緩和

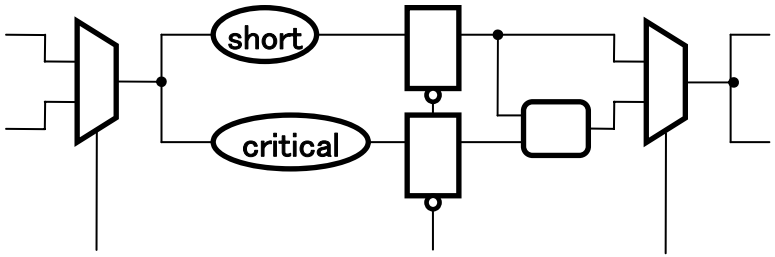
- タイミング故障
  - 回路遅延の動的な変化によって、信号のタイミングに齟齬が生じ、設計者の想定外の動作をする過渡故障
- ばらつきの拡大によって発生確率が増大
  - 動的な検出・回復技術が有効に
- タイミング制約緩和として次の2つを提案
  - 遅延保証フリップフロップ
  - 耐タイミング故障クロッキング方式

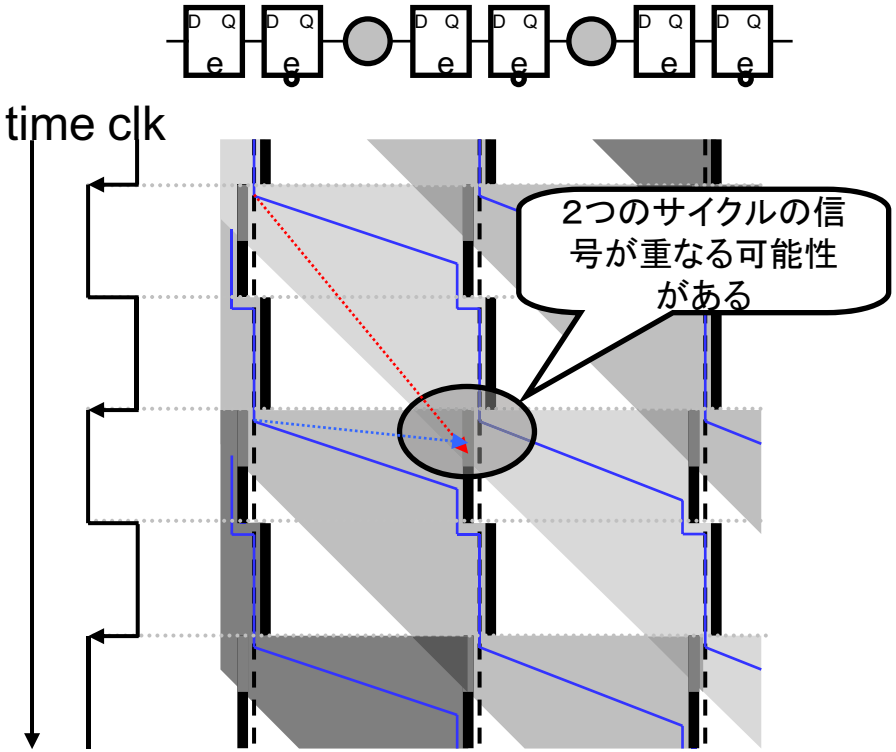
製造ばらつきの増大



S.Borkar, et al., "Parameter Variations and Impact on Circuits and Microarchitecture, " Design Automation Conf. (DAC 2003). Jun. 2003, pp 338-342

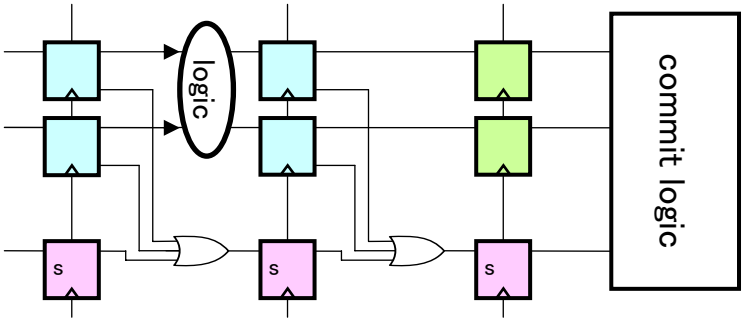
- タイミング故障になるはずの信号をも通過させる
  - 「遅延がクロック・サイクルより長くなっても動作する」
- 基本的なアプローチ
  - 2相ラッチをベース
  - ショート/クリティカル・パスを分離
- H20年度成果
  - 回路構成を決定
- H21年度計画
  - 実装と評価
  - 安浦 G との共同研究を計画



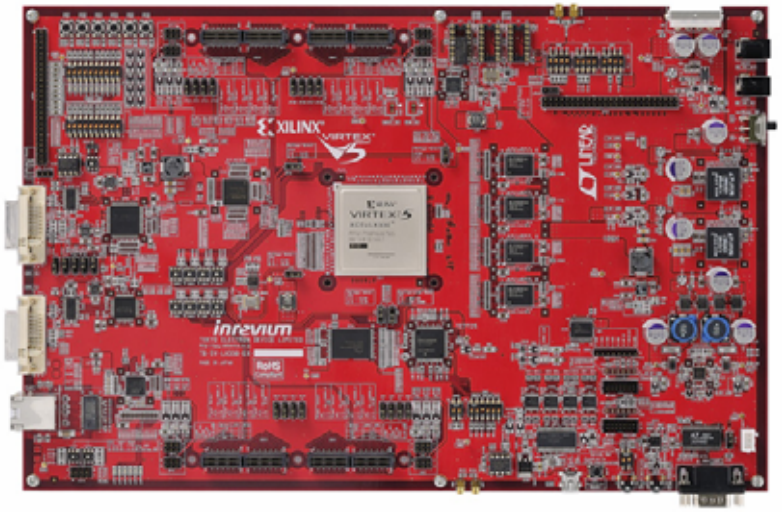


故障耐性を持つスーパスカラプロセッサ

- プロセッサのいかなる箇所に故障が発生しても，正しく回復できるプロセッサを実現
  - パイプラインに沿ったフォルトの伝播
  - アーキテクチャ・ステートを保持しつつプロセッサをリセット
- H20年度成果
  - より大規模で詳細な評価を行うための評価ボード Ver. 2 の製作
- H21年度計画
  - 評価ボード Ver. 2 を用いた詳細な評価
  - 安浦 G との共同研究を計画中



フォルトの伝播

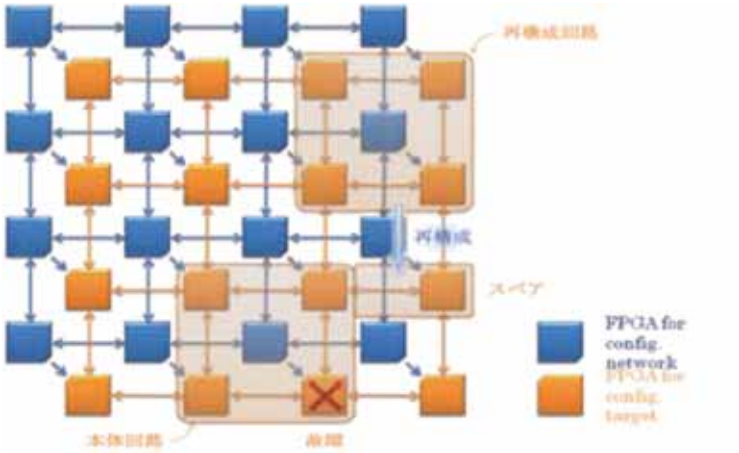


耐故障スーパスカラプロセッサを実装した大容量FPGAボード Ver. 2

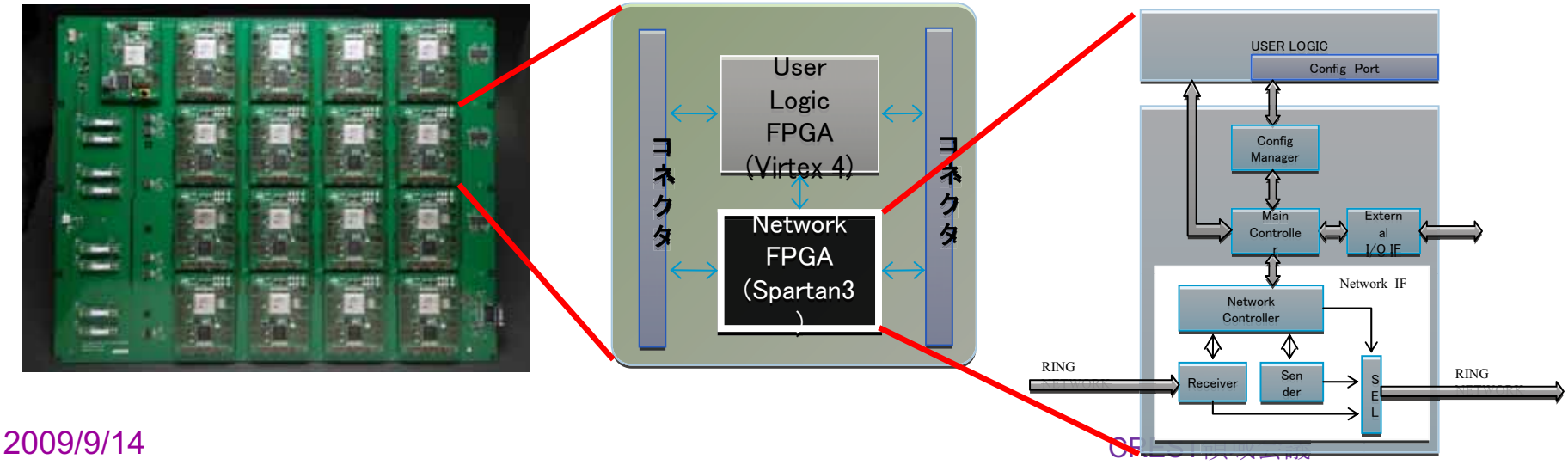
耐永久故障 FPGA アーキテクチャ

高い一時・永久故障耐性をもつFPGAベースのVLSIアーキテクチャ

- 提案方式： 再構成を行う制御回路をも、本体回路と同様に再構成対象として実現
  - 再構成回路自身も本体回路と同様のTMRで構成し、故障時には自分自身を再構成することでTMR状態を回復
  - 単一障害点を最小化
- H20年度 成果
  - テストベッドに提案手法を実装・初期評価
- H21年度 計画
  - アーキテクチャの最適化
  - エラー耐性の評価



耐故障FPGA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のためのSmartCoreシステム

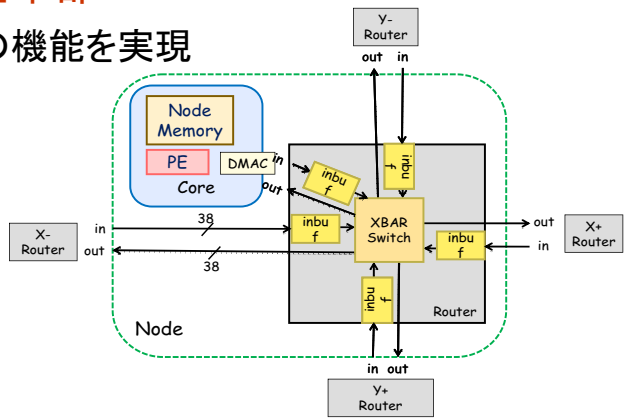
高機能ルータを核として送受信パケットのレベルで冗長実行を実現するシステムの開発

■ SmartCoreシステム => 高機能ルータアーキテクチャ基本部

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

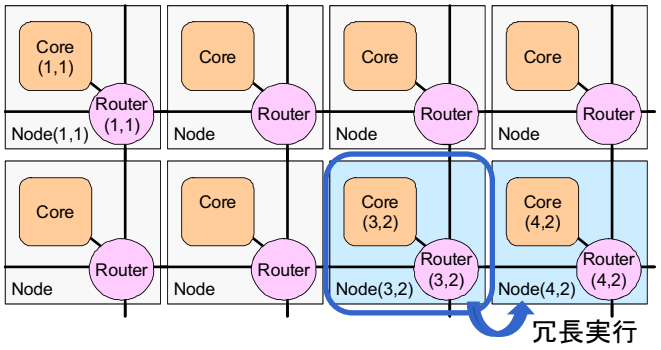
多数のコアと高機能ルータによってメニーコアプロセッサの
ディペンダビリティ向上と速度向上を目指す



マルチコアプロセッサを構成するノード(Node)の詳細

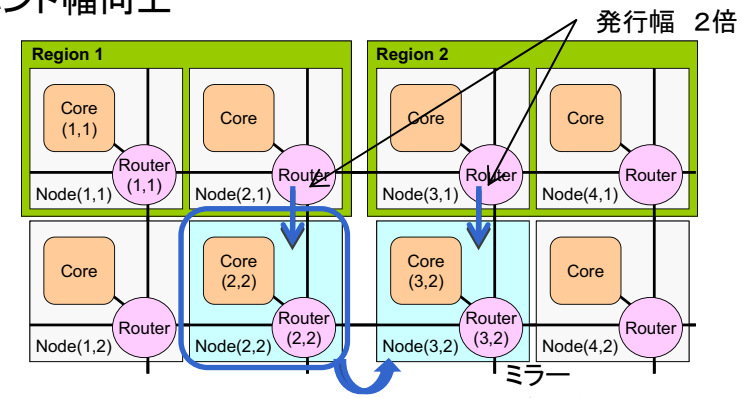
SmartCoreシステム搭載マルチコアプロセッサの例

ディペンダビリティ向上



Node(3,2)のディペンダビリティ向上のため、同等の機能を持つNode(4,2)を冗長実行のために割り当てる。

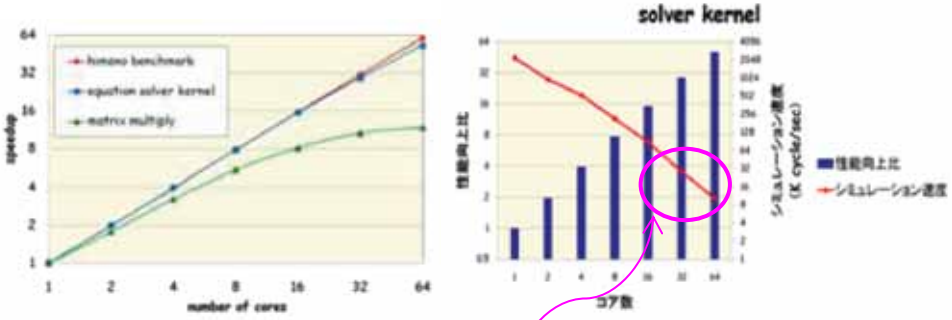
バンド幅向上



Region1、2からNode(2,2)へのアクセスバンド幅向上のため、Node(3,2)にミラーノードとして同様の動作をさせる。

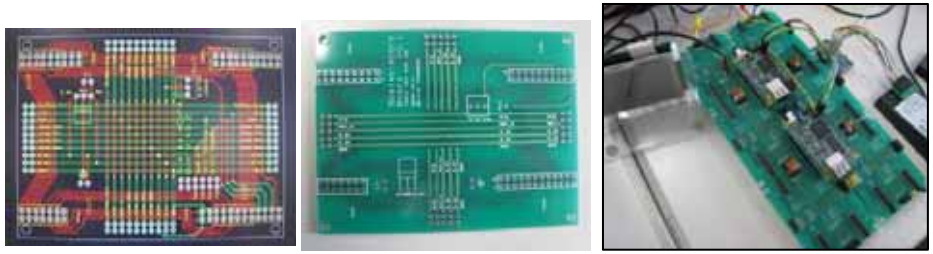
超ディペンダビリティ支援高機能ルータグループの研究計画

- メニーコアシミュレータおよびチップ試作
  - 評価のためのソフトウェアシミュレータ（C++）を開発
  - 評価のためのFPGAシステムの整備
  - 32ビットRISCプロセッサコア，シンプルな4個のルータを個別のチップとしてVDECで試作
- 平成21年度計画
  - 低オーバヘッドかつ柔軟にディペンダビリティのレベルを変更できる高機能ルータアーキテクチャの開発
  - マルチコアシミュレータの改良および高機能ルータアーキテクチャの初期評価
  - 評価のためのFPGAシステムの整備
  - ディペンダビリティを支援する高機能ルータの方式を組み込んだルータチップの試作
  - マルチコア開発支援ソフトウェア
- 外部連携
  - 具体的アプリケーションへの適用を関連企業と連携し、推進
- ディペンダビリティ評価方針
  - 文献調査
  - 評価方針、尺度の策定

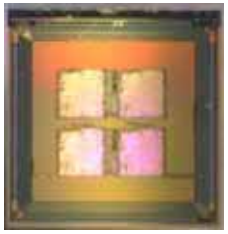


ソフトウェアシミュレータSimMcを用いたマルチコアプロセッサの検討

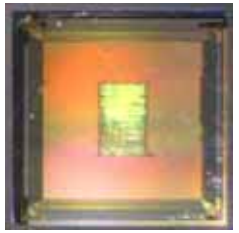
行列計算のような通信が多発する計算においては、
プロセッサ数の増加に伴いシミュレーション速度が低下する
=> FPGAシステムによる実現が望ましい



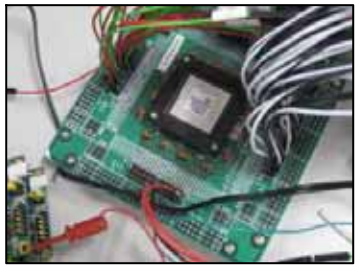
FPGAシステムの整備、ボード設計および小規模システムのテスト



試作した4個の
シンプルなルータ



試作した32ビット
RISCプロセッサ



試作した32ビット RISC
プロセッサの動作検証

チップ試作による実現可能性検証のための整備

③企業連携の具体的な内容・構想

- *国内企業：CADソースコード提供*
  - 国内5社、特に2社（NDA）
- *国内企業：　回路・コアアーキテクチャ*
  - 東大TLOを介してLSIベンダ（NDA）と交渉
  - 知財・IP戦略が中心に
  - バークレイにベンチャー企業：　情報収集中
- *国内企業：　マルチコアアーキテクチャ*
  - メニーコア：　組込VLSIへの展開
- *海外大学：　NDA下によるCADソースコード提供*
  - Indian Institute of Science
  - University of Florida
  - New York City College
- *海外機関：　CADオブジェクトコード提供*
  - NASA
  - Bremen University（ドイツ）
- *海外大学：　ディペンダブルアーキテクチャ研究協力：　RAMPグループ*
  - Carnegie Mellon University
  - MIT
  - University of Texas, Austin
- *学会・協会*
  - 電子情報通信学会CPSY・DC：　CREST共同企画による研究会(2008/4)
  - 情報処理学会（50周年全国大会企画など）
  - 学術会議：　セキュリティ・ディペンダビリティ分科会（提言）との協調

今後

国内SoCベンダ
- **CADツール提供**
- **アーキテクチャ共同開発**

マイクロプロセッサベンダ　（I社等）
- **知財提供**

欧米大学
- **ツール提供による共同研究**
- **アーキテクチャ共同開発**

プレゼンテーション項目④⑤⑥

④ 評価指標

- 形式検証
  - 検証可能なゲート数： 最新VLSI規模
  - 検証時間： 一桁以上の向上
- 回路・アーキテクチャ
  - FTの指標： 動作確率、 MTBF
  - 緩和される遅延ばらつきの大きさ [ps]
  - ディペンダビリティをあげたことによる負の効果の最小化： 0%−数%以下に
    - 性能(Spec, MediaBench)
    - 面積・コスト
    - 電力

    「オーバヘッド**%以下で達成できるディペンダビリティ」という問題設定もある

⑤ 組込OSとの関係についての考え方

- 回路・アーキテクチャによるディペンダビリティは、BIOSレベルでケア
- 組込OSに対しては、透過的に提供

⑥領域運営への希望事項

- 企業との「お見合い」の場の提供をお願いいたしたく
  - 特にSmartCoreグループについて
- 博士課程学生への援助の件、感謝申し上げます。

まとめ

VLSIシステムの信頼性を飛躍的に高める技術の研究

＝　形式的検証とアーキテクチャの最適な協調

- 形式的検証手法
  - C/C++言語ベースVLSI高位設計に対する等価性検証ツール
  - 算術演算回路のための新規技術
  - 対話・自動ドキュメント化のための要素技術
- 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
  - 上位設計から下位設計まで対応できる設計デバッグ支援技術
  - フィールドプログラマブル性を部分的に導入可能な合成手法 ＝ テスト段階でデザインミスをとる
- 回路技術
  - タイミング制約緩和回路
- アーキテクチャ技術
  - 故障検出・回復機構の提案・実現
  - 制御部を含めたFPGA仮想化
  - 耐故障高機能ルータ
  - 超ディペンダブルプロセッサ
- 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
- 各設計階層間のディペンダビリティ役割分担を最適化

今後の展開・課題

- 産業界・海外サイトとの連携を具体化しながら研究開発を進める
- システムとしての実装・最適化・完成度の向上
- チーム間統合技術（新アーキテクチャの検証など）の具体化・実装

形式的検証手法

- 企業からの実用的例題に対する評価
- 各提案手法のツール上でのエンハンス
- 算術回路設計支援ツール（検証だけでなく、合成・最適化）の提案

HW・SW協調実行による検証高速化

- 試作チップのテスト・実証実験
- ディペンダビリティを強化したチップの設計・試作

回路技術

- タイミング制約緩和回路の評価

アーキテクチャ技術

- 故障検出・回復機構の改良・評価
- 制御部を含めたFPGA仮想化： 実装・評価
- 耐故障高機能ルータ： 詳細提案・実装・評価
- 超ディペンダブルプロセッサ： 設計・評価

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の検証手法の一般化
- 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

各設計階層間のディペンダビリティ役割分担

- 実例による最適化実験