

平成26年10月2日

村岡研究室

研究テーマ紹介

村岡研究室 – 現在の研究テーマ (1/2)

組込みシステムやシステムLSI (SoC:システムオンチップ) を設計効率を大幅に向上させるために、システム設計の自動化技術や設計手法について研究を進めています。

(1) システムレベル設計技術(システム設計の自動化技術)の研究

アルゴリズムやシステム仕様の記述からハードウェアやソフトウェアの

アーキテクチャを生成し、検証する技術の確立を目指します。

協調シミュレーション

メカ / アナログ / デジタル / ソフトウェア協調シミュレーション(*)の高速化手法

(*) 異種のモデルを結合してハイブリッドモデル(世界初)をつくり、これらを併せてシミュレーションを行い実記作成前にシステムのバグや問題点を検出すること。

ソフトウェアの並列化アルゴリズム

GPUを用いた高速並列演算による高速な論理回路シミュレーションを行う。(SIMD)

マルチコアプロセッサを用いてソフトウェアの並列化(並列処理、パイプライン化)

を行い、高速な論理回路シミュレーションを行う。(MIMD)

アルゴリズムのハードウェア化(FPGA化)

暗号化アルゴリズム(DES, AESなど)のハードウェア化による高速化を行う。

論理シミュレーションアルゴリズムのハードウェア化(シミュレーションエンジンとよばれる)による高速化を行う。

論理合成技術の研究

論理合成の高速アルゴリズムの研究

村岡研究室 – 現在の研究テーマ (2/2)

(2) マイコン教育教材の開発

FPGAマイコン(8bit)の試作(高知大初のマイコン)

インストラクションセットシミュレータ(ISS: Instruction Set Simulator)の試作

(3) データベースおよび医療情報関連

医療データ解析を用いたアラートシステムの研究を行っています。

(4) その他

いままでに、以下の研究を行いました。

クロストーク解析技術

論理シミュレーションを用いてクロストークの解析(LSI内の配線間の信号相互干渉

を行う方法を提案し、クロストーク解析手法として確立を目指します。

分散データベース

医療用分散データベースの構築法について研究を行ってきました。

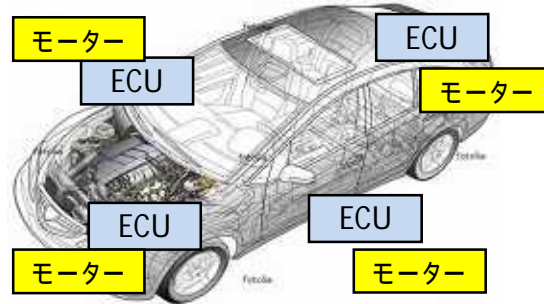
医療データ解析

大規模な医療データベース中の検査データを解析し、投薬とそれによる検査データの変化を抽出する手法の確立を行ってきました。これにより投薬の効果をあきらかにすることができるようになります。

以上

車載用電子制御システム

車載用電子制御部品 (ECU) は車1台当たり数10個 ~ 100個が搭載されておりECU1個あたりの検証スピードを向上する検証環境が求められている



パワートレイン系ECU

エンジン、トランスミッション、他

足回り系ECU

ブレーキシステム、パワーステアリング、他

ボディ系ECU

ボディ統合、ヒューズボックス、エアバッグ、ダッシュボード、ゲートウェイ、電子キー制御 (照合)、シート制御、他

安全系ECU

プリクラッシュセーフティ、パーキングアシスト、ミリ波・レーダーセンサモジュール、他

情報系ECU

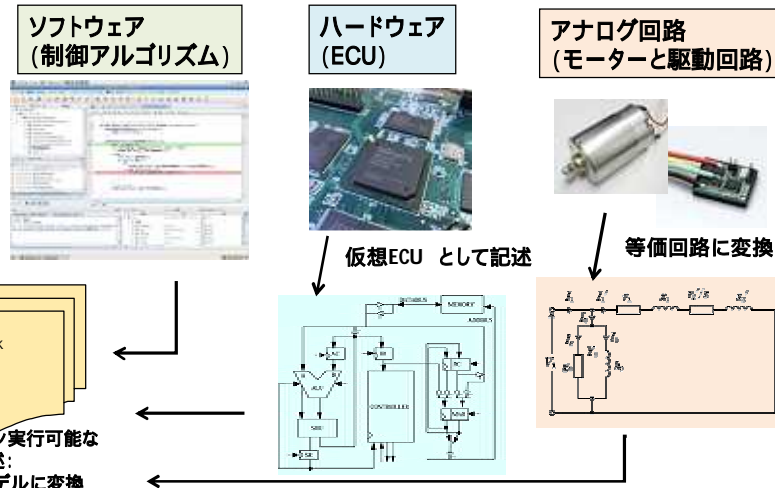
センターナビ (ディスプレイ)、ETC、他

HEV / EV系ECU

パワーコントロール、バッテリーマネジメント、他

高速協調シミュレーション可能な 車載用ハイブリッドモデルの開発

- ソフトウェア、ハードウェア、アナログ回路のハイブリッドモデルを FPGA 上で高速実行可能なプログラミング言語を用いて記述する



GP-GPUを用いた並列論理シミュレーション手法

A Parallel Logic Simulation Method using GP-GPU

～ 論理シミュレーションアルゴリズムの高速化 ～

橋口 拓哉 豊永 昌彦 村岡 道明
高知大学大学院 理学専攻(情報科学分野)

研究背景

近年、システムの大規模化や半導体微細化技術の進歩により、設計の規模・複雑性が増大

設計の検証(機能・論理シミュレーション)には膨大な時間がかかる



研究目的

本研究では

- 並列論理シミュレータを作成
- GPUの性能を最大限引き出せるようアルゴリズムを最適化

GPUを用いた並列論理シミュレーションアルゴリズムを開発
アルゴリズムの更なる高速化

GPUの構成

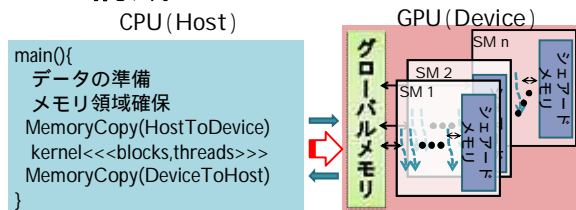
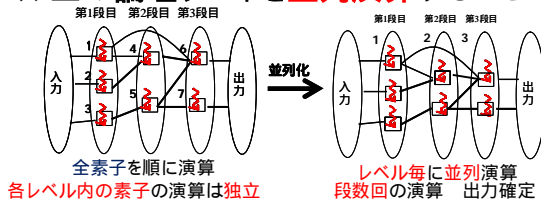


図1. GPUの構成

論理シミュレーション手法

- 本研究では並列化に向くレベルソート法を採用
- レベル上の論理ゲートを並列演算することで高速化



回路分割手法

目的: 回路を分割することで演算量を削減

- ファンアウトコーン: 論理回路の各外部出力端子からコーンを切り出す コーン間に通信発生しない
- コーン数(外部出力端子数) > GPUのSM数
SM数と等しくなるようにコーンをグループ化

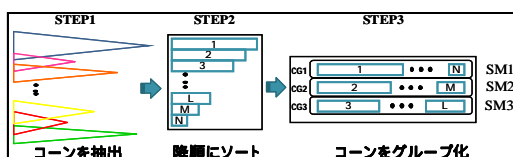


図2. コーングループの生成手順

評価

GPUを用いた並列論理シミュレータと市販高速シミュレータModelSimとの性能比較

GPU : Geforce GTX480

PC環境 : Intel Core i7-950 3.07GHz

テストベクタ長: 100,000テストパターン

・ シミュレータ

- SEQSim: 内部開発の逐次シミュレータ(レベルソート法)
- GPUSim: 提案並列シミュレータ(レベルソート法)
- ModelSim SE 6.2e: 市販高速シミュレータ(イベントドリブン法)

表1. 評価回路

評価回路	インスタンス数	論理段数
cpu x 1	2148	56
cpu x 20	42599	56
cpu x 40	85179	56

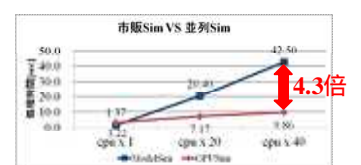
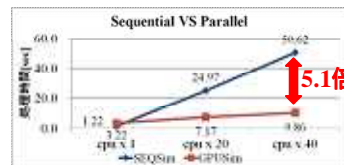


図3. 並列化による性能比較

図4. 市販シミュレータとの性能比較

評価のまとめ

- 評価回路の論理段上の論理ゲート数が増加すると、回路の並列性が大きくなり高速化
- レベルソート法を逐次的に処理するSEQSimと並列に処理するGPUSimを比較すると5.1倍の高速化

- 提案する並列シミュレータ(GPUSim-M)は市販の高速シミュレータと比較して4.3倍の高速化

今後の課題

- ◆ 大規模かつ実用的な回路での評価

32bit processorなど

- ◆ 最新のGPU*を用いた評価

*Geforce TITAN(演算コア数: 2688)

- ◆ 更なる高速化

メモリアクセスの最適化、ゲートの複合化などで市販simの1.0倍以上の高速化を目指す

並列化アルゴリズムによる論理シミュレーションの高速化手法の研究

Research of Acceleration Method for Logic Simulation based on Parallel Algorithm

竹内 勇矢 豊永昌彦 村岡 道明

高知大学大学院 理学専攻(情報科学分野)

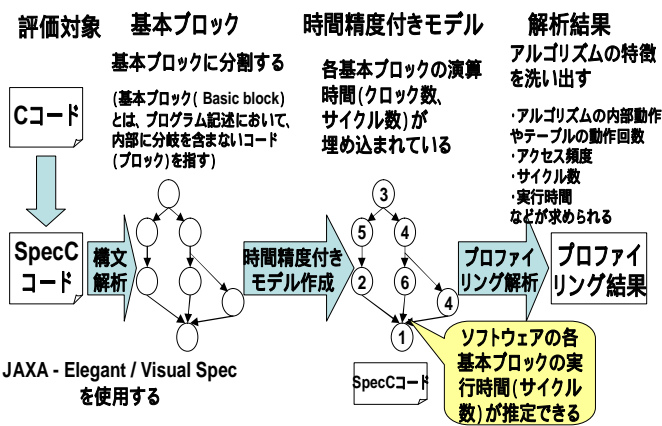
研究背景

- システムや半導体技術の向上により回路の大規模化が進行 シミュレーション時間が膨大に
- シミュレーションの高速化が必要 **並列化**

研究目的

本研究では、実行時間の見積もり手法を用いて、論理シミュレーションアルゴリズムに適応し、**マルチコア**を前提とした効率の良い並列化アルゴリズムを提案し、高速化を目指す。

ソフトウェアの実行時間の見積もり手法



論理シミュレーション手法

本研究では、並列化に向く**レベルソート法**(伝搬遅延時間を考えず論理機能を検証する)を採用。

並列論理シミュレーション手法

ボトルネックを分割し、マルチコアを用いて並列処理することにより**実行時間を短縮**

- ・ボトルネック 論理演算部(回路の演算処理)
- ・回路分割 **ファンアウトコーン**を採用(図1)
- ・マルチコアへの割付方法 インスタンス数が均等になるようコーングループ(コア数分)を生成(図2)

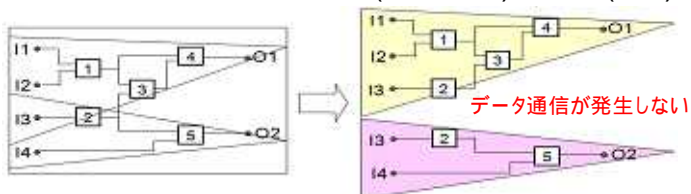


図1. ファンアウトコーン

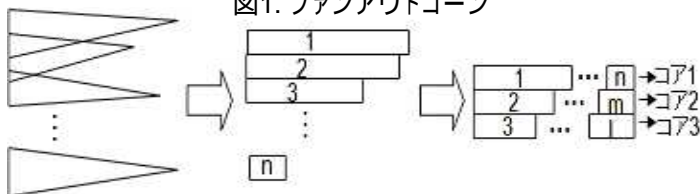


図2. コーングループ

評価結果

本手法を用いた並列論理シミュレーションと市販シミュレーションModelSimとの性能比較

•並列環境

- シミュレータ:Elegant/Visual Spec(ver4.1.6)
- プロセッサ:ARM946E-S(200MHz)

*性能比較の結果は商用環境との周波数比で換算した値となる

•商用環境

- ModelSim SE 6.2e(イベント・ドリブン法)
- PC環境: Intel Core i7-950 3.07GHz

•テストパターン長: 10,000サイクル

表1. 回路情報

評価回路	論理ゲート数	FF数	論理段数
cpu x1	2,111	173	56
cpu x2	4,222	346	56
cpu x4	8,444	692	56
cpu x8	16,888	1,384	56

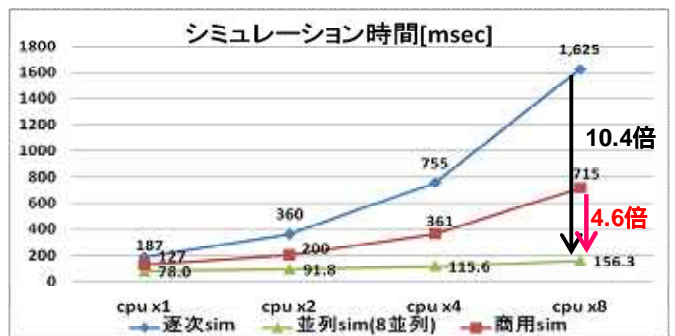


図3. 性能比較

評価結果まとめ

•逐次sim vs 並列sim(8並列)

逐次的にシミュレーションを行う逐次simと論理演算部を8並列で処理する並列simと比較して順序回路cpu x8において**10.4倍**の高速化

•並列sim(8並列) vs 商用sim

並列simは市販シミュレータと比較して順序回路cpu x8において**4.6倍**の高速化

•評価回路とソフトウェアの**並列度が高い**ほど高速化率が増加

今後の課題

•高速化

- コア数の増加(推測: 32並列で10倍以上)
- 並列化のオーバーヘッドを削除(2倍程度)

•大規模回路対応

- 並列化やゲートの複合化を評価

•AND・ORプレーンと比較

FPGAを用いた論理シミュレーション手法

A Logic Simulation Method using FPGA

～ 論理シミュレーションエンジンの提案 ～

松本夏樹

村岡道明

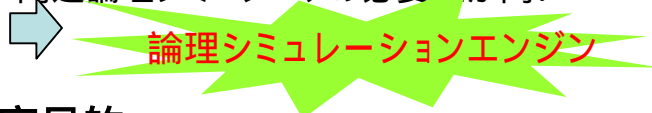
高知大学大学院

理学専攻(情報科学分野)

研究背景

大規模論理回路のFPGA論理エミュレーションはデバッグ効率がよくない。また、ソフトシミュレータは膨大なシミュレーション時間が必要

大規模論理回路におけるデバッグ性のよい高速論理シミュレータの必要性が高い



研究目的

本研究では

- 論理シミュレーションのハードウェアアルゴリズム化
- FPGAへの実装の検討
- 論理ゲートの並列演算

ハードウェア化による高速化 (論理シミュレーションエンジン)

アルゴリズムの更なる高速化

論理シミュレーションの高速化を目指す手法

- 本研究では並列化に向くレベルソート法を採用

シミュレーションエンジンの機能構成

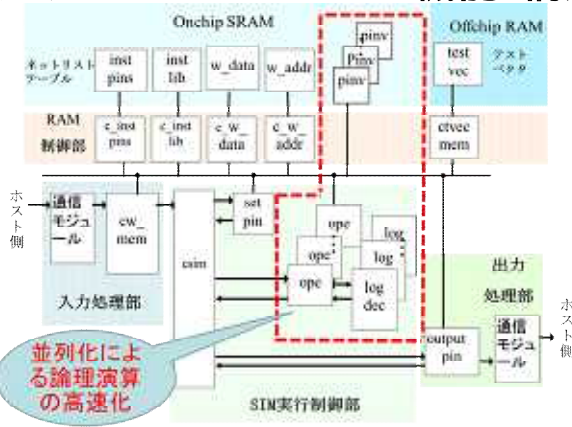


図1. 論理シミュレーションエンジンのブロック図

- 入力処理部 … ネットリストテーブル等の受信
- SIM実行制御部 … 論理ゲートの論理演算
- 出力処理部 … 出力端子の値の送信
- Onchip SRAM … ネットリストテーブルの格納

評価結果

FPGAを用いた論理シミュレーションエンジンと市販シミュレータModelSimとの性能比較

FPGAの周波数は50MHz

テストベクタ長は10,000テストパターン

ModelSim SE 6.2e(イベント・ドリブン法)

PC環境 : Intel Core i7-950 3.07GHz

表1. 評価回路

評価回路	論理ゲート数	FF数	論理段数
cpu x 1	2,111	173	56
cpu x 4	8,444	692	56
cpu x 16	33,776	2,768	56

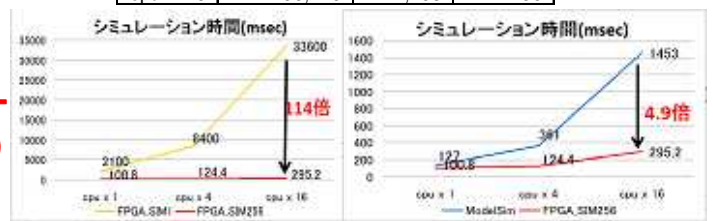


図2. 並列化による性能比較

図3. 市販シミュレータとの性能比較

評価結果のまとめ

- 256並列化シミュレーションエンジン(図2)(FPGA_SIM256)は並列化なしシミュレーションエンジン(FPGA_SIM1)と比較して順序回路 cpu x 16において**114倍**の高速化

- 256並列化シミュレーションエンジン(図3)(FPGA_SIM256)は市販シミュレータと比較して順序回路cpu x 16において**4.9倍**の高速性

- 評価回路の論理段上の論理ゲート数が増加すると、並列性が大きくなり高速化

◆大規模論理回路対応

BRAMの容量の大きなFPGAへの実装 ネットリストテーブルをoffchipRAMへ格納

◆高速化

パイプライン化 約**2~3倍**の高速化
並列化の増加 **並列化数倍**の高速化

化

論理ゲートの複合化 約**2~3倍**の高速化
1桁以上の高速化が見込まれる

1シクによる10倍以上の高速化

FPGA簡易マイコンの提案

~ A Proposal of FPGA Microcontroller ~

山中秀知 村岡道明
高知大学大学院 理学専攻(情報科学分野)

研究背景

8ビットマイコンを搭載した8ビットマイコンボードがいくつか販売されているが、内部動作を確認できるものはなく、初学者が8ビットマイコンを学習するための教材として用いるのは難しい。

デバッグをサポートするための内部動作を出力する8ビットマイコンが必要。

研究目的

- FPGAマイコンを開発し内部レジスタの値を出力 **▶ デバッグのサポートが可能**
- 実行方式に一括実行とステップ実行を導入 **▶ 各ステップごとの内部動作を確認**
- ホストPCとの通信を実装 **▶ 命令の作成・送受信を視覚的に確認**
- GUI上でのデータの送受信

FPGAマイコンの構成・機能仕様

本マイコンは、ホストPCとFPGAマイコンで構成され、FPGAマイコンは図1の10個のモジュールで構成される。また、FPGAマイコンを2台接続し相互に通信できる。

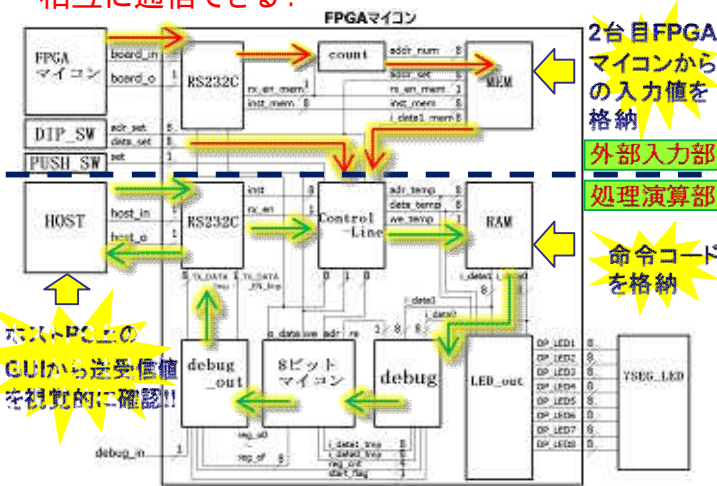


図1. FPGAマイコンのブロック図

- 8ビットマイコン：命令を実行する
- RS232C：送受信データの変換を行う
- control-line：RAMに転送するデータを選択する
- Debug：内部レジスタの値をホストPCへ転送中、8ビットマイコンを一時停止させる。
- RAM/MEM：データを格納するメモリ。
- debug_out：内部レジスタの値をホストPCに出力。
- LED_out：実行中の命令を7SEGLEDに出力。

FPGAマイコン実行例

FPGAマイコンの構成で示したブロック図を実際に接続した例を以下に示す。



図2. ホストPC - FPGAボード間の接続

ホストPCとFPGAボードを図2のように接続！！
FPGAマイコンとホストPCの通信が可能に！！

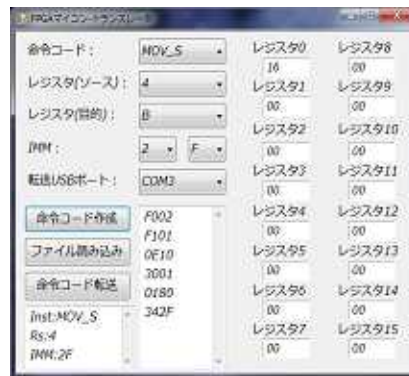


図3. 命令送受信のGUI

命令コードを作成し転送
↓
FPGAマイコンが演算
↓
レジスタの値を表示！！

評価結果とまとめ

- 最高動作周波数
本FPGAマイコン 40MHz
- メモリの容量
最高で約5760の命令格納可能!!
- ホストとFPGA間の通信
5760命令を約1秒で転送可能!!

LSI版とほぼ同等の性能!!

学習するには充分の容量!!

実時間中に転送可能!!

今後の課題

- ◆本FPGAマイコンにおける命令セットシミュレータ(ISS)の作成。
- ◆ISSとFPGAマイコンとの連携
 - 1.ISSで命令のシミュレーション
 - 2.FPGAマイコンで命令を実装
 - 3.ISSとFPGAマイコンの実行結果の比較

部分回路の簡易論理合成手法

Compact Logic Optimization Method for Partial Logic Circuits

蘆苺 将大 村岡 道明

高知大学大学院 理学専攻(情報科学分野)

研究背景

LSIの微細化やシステムの大規模化に伴い、回路の遅延時間や面積の大きさが問題となっている。



部分回路のタイミングかつ面積の最適化を高速におこなう必要がある。

内容

部分回路の簡易論理合成手法を提案し高速な最適化を目指す。

対象とする部分回路を取り出し、その入力と出力から真理値表を使って、準最適な回路を生成する。

簡易論理合成手法

基本アルゴリズム

Step0: 回路の入出力から真理値表を作成(read)

Step1: 真理値表から出力1を取り出す

Step2: ハミング距離1を総当たり探す

Step2.1: ハミング距離1の場合、マージし新しい行を生成する(merge)

Step3: ハミング距離1が見つからなくなるまで、新しく生成された行同士でStep2を繰り返す

Step4: 積和標準形をカバーしているか確認する(entry)

Step5: 多出力の場合、同じゲートを使っている部分を共有化する

Step6: 論理式を出力する(make)

高速化アルゴリズム

Step0: 回路の入出力から真理値表を作成

Step1: 真理値表から出力1を取り出す

Step2: 出力が1になるものと0になるものの数を比較し、多い方を対象とする

Step3: 入力の値の合計を計算する

Step4: 合計の差が1になるものを総当たり探す

Step4.1: ハミング距離1の場合、マージし新しい行を生成する

Step5: ハミング距離1が見つからなくなるまで、新しく生成された行同士でStep2を繰り返す

Step6: 積和標準形をカバーしているか確認する(entry)

Step7: 多出力の場合、同じゲートを使っている部分を共有化する

Step8: 論理式を出力する(make)

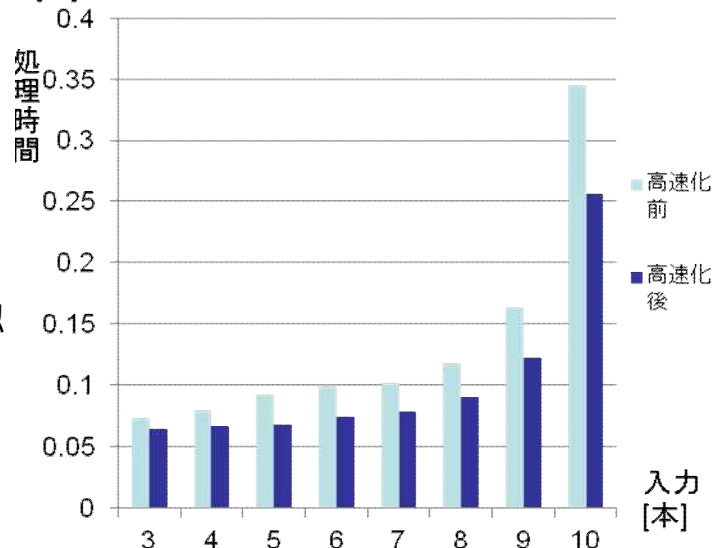
Step9: 出力0になるものを対象にしている場合、論理式全体にNOTをつける

評価結果

表1. ランダム回路(高速化)

	3	4	5	6	7	8	9	10
read	0.0203	0.0211	0.0227	0.0223	0.0238	0.0268	0.0293	0.0335
merge	0.0031	0.003	0.0029	0.0031	0.0037	0.0088	0.0299	0.139
entry	0.0046	0.0046	0.0048	0.0046	0.0049	0.0048	0.0069	0.0128
make	0.0366	0.0373	0.03785	0.0449	0.0461	0.0499	0.0562	0.071
sum	0.0646	0.066	0.0684	0.0749	0.0785	0.0803	0.1223	0.2563

[秒]



今後の課題

大規模論理回路対応

ネットリスト入力対応(回路分割、順序回路)

高速化、並列化

- マルチコア化

- GPUへの実装

省電力化

- スイッチング回数を削減する最適化方法



高速化

FPGAを用いた暗号化アルゴリズムのハードウェア化

High Speed Encryption Unit based on FPGA for Mobile Terminal

～ 高速暗号化ユニットの検討 ～

浪越隆生

村岡道明

高知大学 理学部(応用科学課程情報科学コース)

研究背景

データ転送のセキュリティ向上のためには暗号化技術が必要であるが大容量のデータを暗号化するには時間がかかる

暗号化アルゴリズムの高速化が望ましい

高速暗号化ユニット

研究目的

本研究では

- 暗号化アルゴリズムAESをハードウェア化による高速化(暗号化ユニット)使用
- ハードウェア化
- FPGAへの実装
- 並列処理による高速化

暗号化アルゴリズムAES

- 共通鍵暗号化方式
- 128bitのデータブロックを使用
- 4つの暗号化方式と1つの拡張鍵生成方式使用
- 繰り返し処理することで暗号文を生成

暗号化ユニットの構成

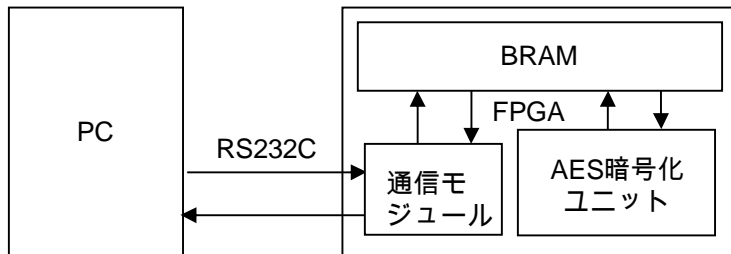


図1. 暗号化システムユニットの構成図

- PC …… データを送受信するホスト
- RS232C …… 通信インターフェース
- FPGA …… 暗号化システム
 - AES暗号化ユニット …… 暗号処理部
 - BRAM …… データの一時保存
 - 通信モジュール …… ホスト間のデータ変換

AES暗号化ユニットの構成図

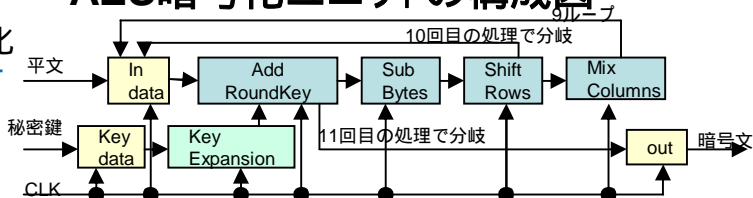


図2. AESのハードアルゴリズムのブロック図

- 暗号化モジュール
- 鍵生成モジュール
- 入出力モジュール

並列化処理の例

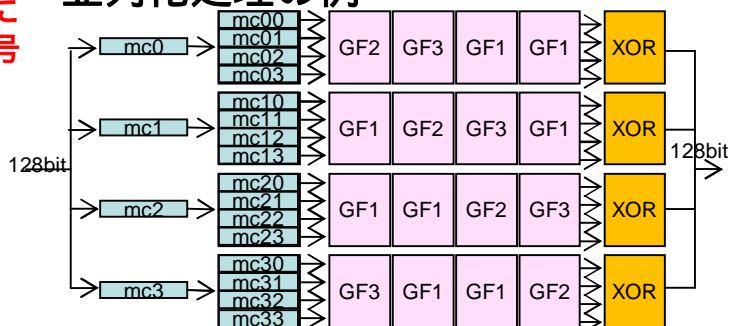


図3. MixColumns処理の並列化

- 4つの暗号化方式のうち1つMixColumnsを並列処理化

GF演算を16並列化

評価

2.4MBのデータを用いて計測

- ソフトウェアアルゴリズム …… SpecCを用いてARM9(周波数200MHz)上での実行時間をシミュレーション **11.2sec**
- ハードウェアアルゴリズム …… ModelSimを用いて周波数100MHzで動作するAES暗号化ユニットをタイミングシミュレーション **0.19sec**

結論

ソフトウェアアルゴリズムとハードウェアアルゴリズムの処理速度を比較するとハードウェアアルゴリズムのほうが約59倍高速である見通しを得た。

今後の課題

- ◆パイプライン化による高速化
- ◆FPGAへの実装と性能評価

医療データ解析を用いたアラートシステムの提案

Proposal of Alert System using Medical Data Analysis

古野 智大 村岡 道明
高知大学 理学部 情報科学コース

片岡 浩巳 奥原 義保
高知大学 医学部附属医学情報センター

研究背景

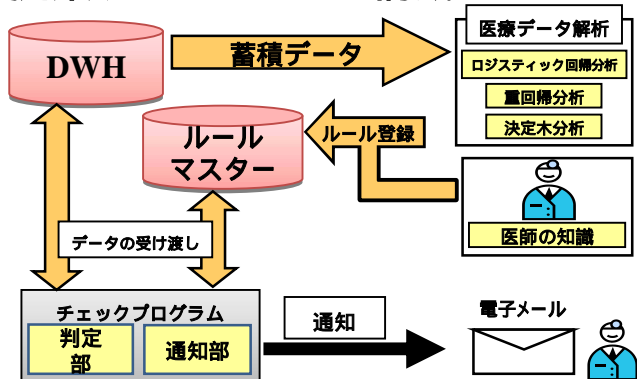
病院での情報技術の活用(電子カルテ、検査情報データベース)が進んでいる。
しかし、最終判断を行うのは人間。

→ ミスを避けきれない
(データ見落とし、薬品投与ミス)

研究目的

上記のミス等の軽減を図るための、データベースに蓄積された検査データを用いたアラートシステムの提案。

提案するシステムの構成



DWH (データウェアハウス): 時系列で整理された大量のデータ、またはその管理システムのこと。

図1. 提案するシステムの構成図

チェックプログラム

DWHのデータ、ルールマスターに登録されたルールを使用して予測を行い、判定結果に基づいて通知を行う。

医療データ解析

今回使用する3つの解析手法

重回帰分析・・・一つの目的変数の、複数の説明変数による予測。次の回帰式で計算される。

$$Y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$

: 定数, n: 回帰係数

Y: 目的変数 (予測したい検査値)

xn: 説明変数 (各検査項目の検査値)

ロジスティック回帰分析・・・事象の発生確率の予測。目的変数は0~1の値をとり、1に近いほど発生確率が高い。次の回帰式で計算される。

$$Y = \frac{1}{1 + e^{-(\alpha + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n)}}$$

: 定数, n: 回帰係数

Y: 目的変数 (疾患の罹患確率)

xn: 説明変数 (各検査項目の検査値)

決定木分析・・・木構造を用いた、IF-THENルールによる分類

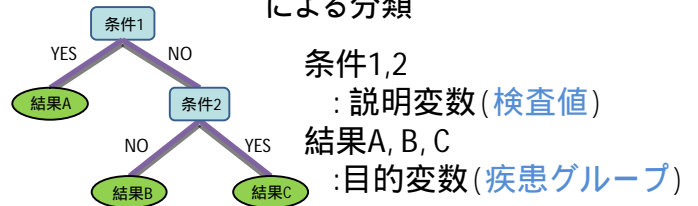


図2. 決定木の例

ルールマスター

疾患の予測計算に必要な、各種データを登録する。機能の拡張に伴い複雑化する可能性や様々なフォーマットへの対応を考慮し、XMLで作成する。

疾患	検査項目	係数	基準値
1		A	S ₁
	X ₁	B ₁	
	X ₂	B ₂	
2		A	S ₂
	X ₁	B ₁	
	X ₂	B ₂	

$$A$$

$$X_1 \times B_1 = B_1 X_1$$

$$X_2 \times B_2 = B_2 X_2$$

$$A + B_1 X_1 + B_2 X_2 + \dots$$

$$Y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$

$$Y \text{ 比較 } S_1$$

図3. 重回帰分析のルールマスターの構成と、それを使用した計算のイメージ

通知

基準と判定

計算された疾患リスクが、登録された基準値を超えた場合に通知を行う

通知方法

医師の持つ携帯端末に対する電子メールの送信

通知内容

個人情報が含まれることを考慮し、疾患リスク検出結果の表示、システムへのアクセス要求のみとする

評価方法

- ・DWHを使用した予測結果の信頼性検証
- ・プログラムの動作や、通知内容の妥当性などの医師による評価

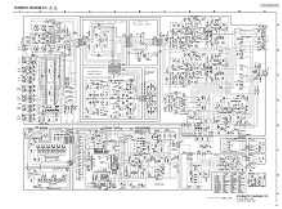
今後の課題

- ・プロトタイプ completion
- ・ルールマスターに登録するルールの導出
- ・実際に動かして、予測の信頼性検証やルール等の調整を行う
- ・電子カルテとの連携など、より効果的な通知方法の検討

GP-GPUを用いた並列論理シミュレーションの性能評価

Evaluation of parallel logic simulation performance using GP-GPU

青野寛之 橋口拓哉 村岡道明
高知大学 理学部(情報科学コース)

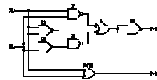


研究背景

近年システム(論理回路)の大規模化により
検証(シミュレーション)時間が増加



高速化が必要!



先行研究

(GP-GPUを用いた論理シミュレーションの高速化)

GPUを使って市販シミュレータの
約8倍高速化が達成されている

(Graphics Processing Unit, 画像処理装置)

多数の演算コアがGPUの特徴

例) 先行研究で使用 GTX480 : 演算コア480個

この多数の演算コアを画像処理以外に利用する

GP-GPU(General Purpose, 汎用)

研究目的

各種のGPUを使用し論理シミュレーションの性能評価を行う

• GPUの種類と論理シミュレーションの実行時間の

関係を調べる

• GPUを用いた論理シミュレーションの高速化に

評価環境

SM ... 演算コアの集合単位

コア/SM ... 1SM内の演算コア数

以下の表よりGTX780がコア数, クロック周波数ともに最も高性能といえる

表1. 使用したGPU

型番	コア/SM	SM数	全体コア数	クロック周波数 (MHz)	メモリ量(GB)
GTX 780	192	12	2,304	900	3
GTX 480	32	15	480	700	1.5
Quadro600	48	2	96	640	1
GT 540M	32	2	64	672	1

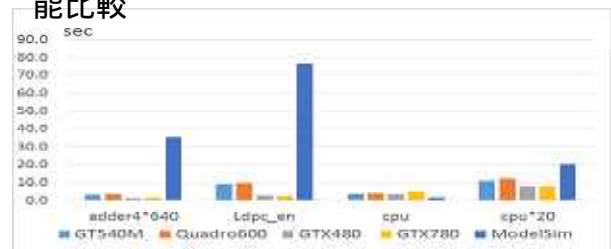
評価結果

テストベクタ長は100,000テストパターン
ModelSim SE 6.2e(イベント・ドリブン法)
PC環境 : Intel Core i7-950 3.07GHz

表2. 評価回路

評価回路	adder4x640	Ldpc_en	cpu x 1	cpu x 20	cpu x 40
インスタンス数(ゲート数)	16009	78806	2148	42599	85179
論理段数	10	12	56	56	56

図1. GP-GPUを用いた論理シミュレー
シヨタと市販シミュレータModelSimとの性
能比較



評価結果のまとめ

• Adder4 x 640において全GPUで市販シミュ
レータと比較して**10倍以上**の高速性(図1)

• Ldpc_enにおいて**GT540M, Quadro600**で
約7倍, **GTX480, 780**で市販シミュレータと比
較して**約25倍の高速性**(図1)

• 評価回路の論理段上の論理ゲート数が増
加すると、並列性が大きくなり高速化

今後の課題

◆ 大規模論理回路対応

数十万規模回路(今回最大80,000ゲ
ト)

並列化の増加

◆ 複数GPU使用検証

高速メモリ容量の増加

大規模回路への対応

◆ GTX780の高速化

現在GTX480とほぼ同処理時間

主な就職先

平成26年(2014)
(内定)

三菱自動車、デンソーテクノ
三菱インフォメーションシステム
NTTビジネスソリューションズ
佐賀電算センター

平成25年(2013)

(全員が修士進学)

平成24年(2012)

湯山製作所

平成23年(2011)

三菱電機

NTTデータ四国、

四国情報管理センター

平成22年(2010)

富士通

大日本印刷

日立システム

平成21年(2009)

NECシステムテクノロジー

高知電子計算センター

平成20年(2008)

三菱電機メカトロニクスソフトウェア

ケイレックス・テクノロジー