

グリッチ削減のためのパイプライン構造の最適化

小島 拓也[†] 安藤 尚輝[†] 奥原 颯[†] 天野 英晴[†]

[†] 慶應義塾大学大学院 理工学研究科 223-8522 神奈川県横浜市港北区日吉 3-14-1

E-mail: †{tkojima,ando,hayate,hunga}@am.ics.keio.ac.jp

あらまし CGRA(Coarse Grained Reconfigurable Array) ではエネルギー効率を高めるために複数の PE(Processing Element) を統合して 1 クロックサイクルで複数の命令が実行できるものが提案されている。こうした機能はグリッチの伝搬を引き起こし、電力の増加の一因となっている。そこで我々はグリッチ伝搬を考慮したダイナミック電力モデルとそれを利用した CGRA の最適化手法を提案する。提案モデルでは PE アレイのスイッチング回数を近似的に計算し、それを基に消費エネルギーを見積もることを目標とする。本研究ではこのモデルと最適化手法を VPCMA(Variable Pipeline Cool Mega Array) と呼ばれる低電力 CGRA の実チップへ適用し評価を行った。実チップ測定の結果を利用した提案モデルは、ポストレイアウトシミュレーションと比べ 10000 倍以上速く、より小さい誤差で消費エネルギーを見積もることができた。また、多くのベンチマークにおいて本手法で最適化したパイプライン構造は固定したパイプライン構造と比べて、消費エネルギーを小さくすることができた。

キーワード CGRA グリッチ 電力最適化 可変パイプライン

Takuya KOJIMA[†], Naoki ANDO[†], Hayate OKUHARA[†], and Hideharu AMANO[†]

[†] Graduate School of Science and Technology, Keio University Hiyoshi 3-14-1, Kohoku-ku, Yokohama, Kanagawa, 223-8522 Japan

E-mail: †{tkojima,ando,hayate,hunga}@am.ics.keio.ac.jp

1. はじめに

来たるべき IoT デバイスやウェアラブルコンピュータは極めて小さい消費電力で、高い処理能力を要求される。CGRA(Coarse-Grained Reconfigurable Arrays) は高いエネルギー効率を持ち、そうした要求を満たすアーキテクチャとして期待されている。多くの CGRA は簡単な処理を行う PE(Processing Element) をアレイ状に持ってあり、メモリが分散して配置されている。PE とメモリは相互に接続され、PE で実行する演算や接続を切り替えることでアプリケーションを効率的に実行することができる。

VPCMA (Variable Pipeline Cool Mega Array) [1] は CMA (Cool Mega Array) [2] を基に提案された低電力 CGRA である。他の CGRA と同様に、大きな PE アレイと小さなマイクロコントローラ、バンク化されたデータメモリを持つ。動的に再構成可能な CGRA とは異なり、各 PE は動的には再構成されず、レジスタファイルを持たない。PE アレイはパイプライン化されていて、対象とするアプリケーションや要求される性能に応じてパイプライン構造を変化させることが可能である。パイプラインレジスタをバイパスする場合、前のパイプライン

ステージと後のパイプラインステージは統合され、1 クロックサイクルで実行される。この統合されたステージ内の PE はレジスタを使用せずに互いに接続される。このようなステージの統合はクロック周期を低下させてしまうものの、各ステージの遅延時間を調整するのに有効であり、低い性能が要求される場合電力を削減することが可能になる。[1] ではパイプライン構造を変化させることによるトレードオフが議論されている。また、[3] [4] では汎用的な CGRA に対して動的再構成を利用した同様のアプローチを取っている。

こうした PE の統合ではグリッチの伝搬の影響を十分に検討する必要がある。グリッチは本来計算には必要のないスイッチングであり、組み合わせ回路への各入力が異なる遅延時間で入力されることによって発生する。複数の PE にグリッチが伝搬するとそれによって電力が増加し、時には PE 統合による電力削減効果を上回ってしまう。先行研究ではグリッチの影響を指摘しつつも、それを定量的に扱っていなかった [1]。そこで、我々はグリッチの影響を考慮した電力モデルを提案する。また VPCMA のパイプライン構造を最適化する手法を提案する。本モデルと最適化の結果は実チップ測定により確認する。

以降では 2. 節で VPCMA の概要を説明する。3. 節では電力

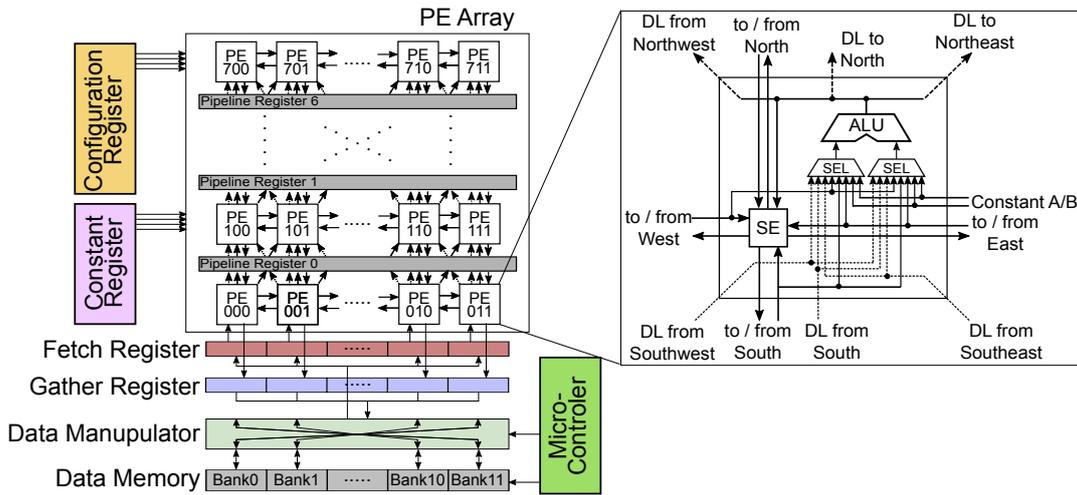


図 1: VPCMA のアーキテクチャ

モデルと最適化手法を説明する。4. 節では VPCMA の実装環境と使用を示す。5. 節では本モデルと本手法の評価を行う。6. 節で結論を述べる。

2. VPCMA の概要

VPCMA は Straight Forward CGRA(SF-CGRA) に分類される。SF-CGRA はパイプライン化された PE アレイとメモリモジュール、それらの間のデータ転送を行うネットワークで構成される。効率的にデータを転送するために、PE アレイの入力部、出力部に置換網が置かれる。メモリモジュールから読み出されたデータはパイプライン化された PE アレイへと入力され、計算結果がメモリモジュールへ書き戻される。各 PE の演算や相互接続はコンフィギュレーションデータにより制御され、これらはしばしば動的に切り替えられる。Piperench [5] や Kilo-core [6]、S5 engine [7] など SF-CGRA に分類される。一部の SF-CGRA は VLIW (Very Long Instruction Word) 型のアーキテクチャとして扱われることもある。

VPCMA は単純な SF-CGRA であり、計算に必要な電力以外の電力を削減することを目標としたアーキテクチャである。各 PE はレジスタファイルを持たず組み合わせ回路のみで構成されているため、クロックツリーの分配が不要である。

図 1 は VPCMA の PE アレイの構成を示している。PE アレイはパイプラインレジスタを含む 12 列 × 8 行の PE で構成されている。PE アレイの他にマイクロコントローラ、バンク化されたメモリを持つ。入力のフェッチレジスタに一度データが書き込まれると、PE アレイでの演算が開始し、計算結果が出力のギャザラレジスタへと格納される。図 1 には PE の構成も示している。

一つの ALU(Arithmetic Logic Unit)、入力のセクタ、SE(Switching Element) が各 PE に配置されている。相互接続はアイランドスタイルの SE を利用したものと、SE を利用しないダイレクトリンクがある。ダイレクトリンクは図 1 に示すように ALU の出力を北方向、北西方向、北東方向へ転送する。計算結果は南方向へのリンクを通して、ギャザラレジスタへと

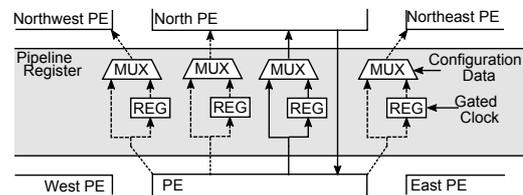


図 2: パイプラインレジスタの実装

転送される。

パイプラインレジスタは各 PE 行の間に配置されているため、PE アレイには 7 本のパイプラインレジスタがある。各パイプラインレジスタは選択的に使用することができ、様々なパイプライン構造をとることができる。図 2 はパイプラインレジスタの構成を示している。2つの動作モード(ラッチまたはバイパス)を実現するために、マルチプレクサを使用して ALU の出力またはレジスタの出力どちらか一方のみを次のステージへ転送する。動作モードはコンフィギュレーションデータによって指定することができる。バイパスモードの場合はレジスタでの電力消費を抑えるためにクロックゲーティングを施す。北方向から南方向へのパスは計算結果を転送するためだけに使用されレジスタを経由しない。

マイクロコントローラが実行する命令によって、PE アレイへの入力データはデータメモリからフェッチレジスタへと転送され、計算結果も同様にギャザラレジスタからデータメモリへ転送される。ここでは、フェッチを前者、ギャザラを後者のデータ転送を表すこととする。ギャザラ命令が発行された後、その実行はパイプラインレイテンシの分だけ数クロック遅らされる。

データマニピュレータと呼ぶ置換網はバンクメモリとフェッチレジスタ、ギャザラレジスタとの間で柔軟なデータ転送を可能にする。フェッチの場合、メモリバンクから読み出されたデータはフェッチレジスタの任意の位置へ転送することができる。フェッチレジスタとバンクの対応付けはマッピングテーブルによって指定され、マイクロコントローラで実行される命令によってどのマッピングテーブルを適用するかを指定する。このような機構によって、エネルギーを多く消費する動的再構成

を行わずに様々なアプリケーションを実行することが可能となる。

3. グリッチを考慮した電力モデル

3.1 グリッチの伝搬

グリッチは計算には不要な信号変化であり、そのスイッチングが原因である程度の電力を消費してしまう。グリッチはゲートへの入力信号が遅延することにより発生する。同様に、異なる遅延時間で PE ヘデータが入力されるとグリッチが発生する。パイプラインレジスタがバイパスされると、発生したグリッチは次のステージへと伝搬し、消費電力が増加する。一方で、パイプラインレジスタがラッチモードの場合グリッチの伝搬を抑制することができるが、パイプラインレジスタへのクロック分配による電力が必要となる。

3.2 電力モデル

3.1 で述べたトレードオフを考慮して、電力を削減できるパイプライン構造を見つけるためにグリッチを考慮したダイナミック電力のモデルが必要である。

CMOS VLSI におけるダイナミック電力 P_{dyn} は一般に以下のように計算される。

$$P_{dyn} = \alpha C f V_{dd}^2 \quad (1)$$

ここで、 α はスイッチング率、 C は負荷容量、 f はクロック周期、 V_{dd} は電源電圧である。

式 (1) を基に、 V_{dd} を固定して考えると、組み合わせ回路部のダイナミック電力 ($P_{dyn,comb}$) は以下のように単純化することができる。

$$P_{dyn,comb} = E_{sw} S_{total} f, \quad (2)$$

ここで、 E_{sw} はスイッチング 1 回あたりの消費エネルギー、 S_{total} はグリッチを含む PE アレイの総スイッチング数を表す。 E_{sw} は PE の構造や使用するプロセスに依存する定数である。この値は対象チップのシミュレーションにより取得可能である。また、 $P_{dyn,comb}$ は組み合わせ回路におけるダイナミック電力のみを表し、レジスタやクロックツリーによって発生する電力は 3.3 節で検討する。

本研究では S_{total} の近似的な値を求めることに焦点を当てる。そこで、モデル化を行うために 4. 節で詳細を述べる VPCMA の設計を基に様々なパイプライン構造でシミュレーションを実施した。

図 3 はパイプライン構造を変化させたときの組み合わせ回路で消費するダイナミックエネルギーの例である。シミュレーションしたアプリケーションは 4. 節で詳しく説明する gray である。8 ステージパイプラインは各 PE がパイプラインレジスタによって分割されているためグリッチの伝搬が最小化されている。図 3 のピンクの部分グリッチが最小の時の消費エネルギーである。青の部分がそれとの差、つまり、グリッチ伝搬による消費エネルギーを表す。これらは組み合わせ回路部分のみの消費エネルギーであり、クロックツリーやレジスタでの消費エネルギーは含まれていない。また、図 3 では log スケール

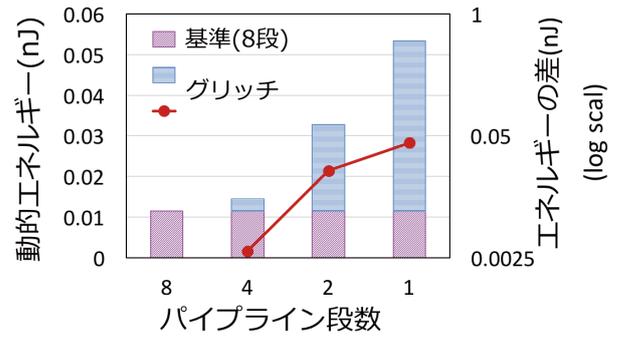


図 3: グリッチによるエネルギー消費

のエネルギー差も示している。この例から、統合するパイプラインステージの数に対して線形的な増加よりも大きな増加である一方で指数関数的な増加ではないことが確認できる。この結果から、統合された PE 間でグリッチは伝搬し、伝搬されたスイッチング回数は累積していくため、統合する PE の数が増えるにつれてグリッチは指数関数に近い割合で増加していくと想定することができる。しかし、グリッチ伝搬は部分的に妨げられるため単純な指数関数とはならない。また、PE はいくつかの PE から複数の入力を受けるが、伝搬されるグリッチの大きさは、最もスイッチング回数が大きい PE からの入力によって定まると想定する。これらを考慮して、スイッチング回数は以下の式のように表すことができる。

$$S_{total} = \sum_{i=0}^n \sum_{j=0}^m S_{PE}(i, j) \quad (3)$$

$$S_{PE} = S_{single}(op) + \beta \gamma^{length} \max_{dir} S_{prev}(dir) \quad (4)$$

$$dir = \{\text{SOUTH, WEST, EAST, DL.South, DL.Southwest, DL.Southeast}\} \quad (5)$$

ここで、

- S_{total} : PE アレイ全体のスイッチング回数
- $S_{PE}(i, j)$: i 行 j 列目 PE のスイッチング回数
 i と j はそれぞれ $[0, n]$ 、 $[0, m]$ の範囲の整数 ($m \times n$ の PE アレイの場合)
- S_{single} : 演算 op が PE にマッピングされたときのグリッチを含まないスイッチング回数
- S_{prev} : 前段に接続された PE のスイッチング回数
- β, γ : 伝搬係数
- $length$: 直前のアクティブなパイプラインレジスタからの距離
- dir : 前段の PE が接続されている入力方向を表す。

式 (4) の最後の項がグリッチの伝搬を表している。ある PE と接続された前段の PE の間にあるパイプラインレジスタがラッチモードとなっている場合、ここでグリッチの伝搬が止められるため S_{prev} を 0 とする。また、PE に演算がマップされていない場合はその PE のスイッチング回数 S_{PE} を 0 と計算することにする。

図 4 はこのモデル式によって S_{total} が計算される例である。図中の四角は PE を表しており、マッピングされた演算とその演

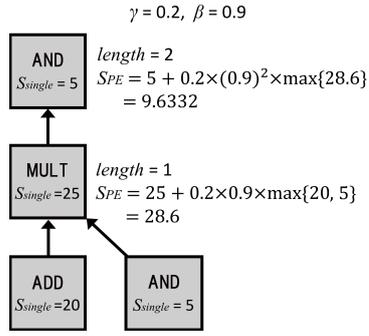


図 4: 本モデルによるグリッチ伝搬の計算例

算における S_{single} が示されている。また、伝搬係数を $\beta=0.9$ 、 $\gamma=0.2$ と想定している。

1 段目のパイプラインステージにある PE はグリッチの伝搬はないので $S_{PE} = S_{single}$ となる。2 段目の PE には 2 つの異なる PE からの入力があるが、ADD がマップされた PE の方がスイッチング回数が多いためこれを用いてスイッチング回数を計算する。3 段目についても同様に計算が行われ S_{total} は $20 + 5 + 28.6 + 9.6332 = 63.2332$ と計算される。

ここで伝搬係数 (β と γ) はマッピングするアプリケーションによらず一定の値であり、シミュレーションや実チップ測定を基にこれらを決定する。これによりアプリケーションとそのマッピングが決定すれば、PE アレイのスイッチング回数の推定が可能となる。

3.3 最適なパイプライン構造の探索

VPCMA は 7 本のパイプラインレジスタによって $2^7=128$ パターンのパイプライン構造をとり得る。静的電力はパイプライン構造によらず一定である。そこで、対象のアプリケーションとマッピングが固定されたとすると、以下の問題を解くことで最適なパイプライン構造を探索することができる。

$$\min(P_{dyn,comb}) + P_{dyn,reg} \times N_{preg} \quad (6)$$

制約条件

$$D_l = \sum D_{PE}(op) \quad (7)$$

$$D_l \leq D_{req}, \quad \forall \text{ datapath } l \quad (8)$$

ここで、 $P_{dyn,reg}$ はパイプラインレジスタ 1 本あたりの動的電力を示し、 N_{preg} はアクティブなパイプラインレジスタの本数である。 D_{PE} は PE に演算 op がマップされたときの遅延時間を示す。 D_l は l 番目のデータパスの合計遅延時間を表し、許容される最大遅延時間 D_{req} 以下となる必要がある。 D_{req} の逆数は要求動作周波数 f を表す。

4. 評価用 VPCMA

4.1 VPCMA のチップ実装

VPCMA の実チップ実装の仕様を表 1 に示す。また、図 5 が VPCMA のチップ写真である。図中の黄色の線で囲まれた部分は PE アレイの 1 行であり、8 行あることが確認できる。赤線で囲まれた部分はチップ間誘導結合 TCI(Through Chip Interface) というチップ間無線通信を行う部分であるが、本文中では TCI の使用は検討しない。

表 1: VPCMA のチップ仕様

設計	Verilog HDL
シミュレータ	Cadence NC-Verilog
プロセス	Renesas SOTB 65 nm / LPT-8
論理合成	Synopsys Design Compiler 2016.03-SP4
配置配線	Synopsys IC Compiler 2016.03-SP4
チップサイズ	6mm × 3mm

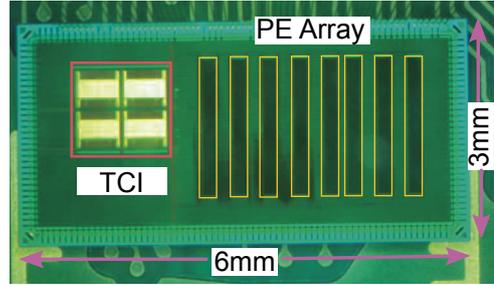


図 5: VPCMA のチップ写真

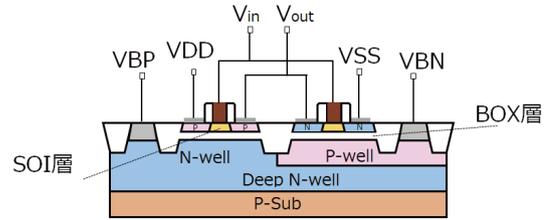


図 6: SOTB のトランジスタ構造

実チップは 65nmSOTB プロセスを利用している。SOTB は FD-SOI の一種であり、図 6 に示すように薄い BOX 層の上にトランジスタが形成された構造をしている。このトランジスタ構造によって比較的高い周波数での動作を低電力で実現することが可能である [8]。SOTB ではボディバイアス制御を用いることで遅延時間とリーク電力を制御することができるが本研究では動的電力の削減を検討しているため、今回ボディバイアス制御は検討しない。

4.2 本モデルのパラメータ取得

4.2.1 アーキテクチャ依存のパラメータ

提案モデルにおいて、スイッチング 1 回あたりのエネルギー E_{sw} と伝搬係数 β 、 γ は対象システムのハードウェア構造やプロセス技術に依存する。シミュレーションによってこれらのパラメータを決定することもできるが、本研究では実チップ評価の結果を用いてパラメータを決定した。128 種類のパイプライン構造における消費エネルギーを測定し、最小二乗法を用いてパラメータを決定した。以下にその結果を示す。

$$E_{sw} = 0.1117, \beta = 1.325 \text{ and } \gamma = 0.053.$$

全てのパイプライン構造の測定をするのは時間のかかる作業であるものの、一度パラメータを取得すればどのようなアプリケーション、要求性能においても同じパラメータを用いることができる。

4.2.2 $S_{single}(op)$

$S_{single}(op)$ は前述のようにマップされる演算によって異なる

表 2: 各演算における PE の平均スイッチング回数

op	ADD	SUB	MULT	SL	SR	SRA	AND	OR	NOT
switching	17.1693	20.0153	31.4623	6.79133	4.97267	7.318	5.217	16.9223	11.1277

表 3: シミュレーションしたアプリケーション

アプリケーション	内容	使用する行数
gray	24 bit (RGB) gray scale	8 rows
sepia	8 bit sepia filter	6 rows
sf	24 bit (RGB) sepia filter	8 rows
af	24 bit (RGB) alpha blender	7 rows
dct	8-point DCT	8 rows

る値となるため、各演算ごとにスイッチング回数を見積る必要がある。表 2 に PE で各演算を実行した場合の平均スイッチング回数を示す。各演算に関して、NC-Verilog を用いた RTL シミュレーションをランダムに生成した入力パターンで 100000 回平均をとっている。また、スイッチング回数は PE の出力のスイッチング回数を記録したものである。

シフト演算 (SL, SR and SRA) と比較して、論理演算 (AND, OR and NOT) は大きなスイッチング回数である。算術演算 (ADD, SUB and MULT) のスイッチング回数は最も多く、シフト演算の約 3 倍である。

5. 評価

本モデルの利点は (1) 時間のかかるポストレイアウトシミュレーションなしに電力を見積ることができること、(2) 要求性能と対象アプリケーションを決めれば自動的にパイプライン構造を決定できることである。

表 3 に示す 5 つのアプリケーションを VPCMA 上で実行した。グリッチの影響を明らかにするために、低い動作周波数 (5MHz) ($D_{req} = 200$ ns) と低い電源電圧 (0.55V) を用いた。まず初めに、実チップ測定の結果と本モデルを用いた結果、ポストレイアウトシミュレーションによる結果を比較する。アプリケーションは *gray* であり、ここではパイプライン段数が 3 段の場合を例に示す。3 段の場合パイプライン構造は $\frac{71}{2^{(7-2)}} = 21$ パターン存在する。

実チップにおける組み合わせ回路部分での消費エネルギーを図 7(a) に示す。実チップ測定の結果、消費エネルギーは明らかにパイプライン構造に依存して変化している。図 7(b) に提案モデルによって計算した結果を示す。これらの値は実チップ測定の結果と比較した場合平均で 3.327% の誤差を含んでいるが実チップの結果を概ね見積もることができている。一方で図 7(c) に示すポストレイアウトシミュレーションの結果は最大で 71.1% の誤差がある。この誤差は SOTB プロセスのモデルの精度によるところが大きく、他の一般的なプロセスを用いればこの誤差は小さくなる。ところが、一般にポストレイアウトシミュレーションによる電力の見積もりはある程度の誤差を含む。

全アプリケーションに関して、全パイプラインパターンでの平均誤差を表 4 に示す。

5.1 本モデルによる動的電力

af と *sf*、*dct* においては誤差は十分小さいが、*sepia* と *gray*

表 4: 本モデルの平均誤差

	gray	sepia	af	sf	dct
error (%)	24.78	34.99	4.72	5.11	7.08

は比較的大きな誤差である。この 2 つのアプリケーションは算術演算が占める割合が大きく、画素データの一部は 0 を持つことが頻繁にある。本モデルではランダムな入力値で平均スイッチング回数を見積もったため、実際のアプリケーションプログラムよりも大きな電力を見積もっている。つまり、ランダムな入力値を用いてスイッチング回数を見積もった場合、本モデルは電力の見積もりに関してやや保守的になってしまう。

5.2 計算時間

本モデルを用いて各パイプライン構造の消費電力を得るために C 言語で記述した簡単なプログラムを用意した。これを用いることで 128 パターンの電力を約 0.05 秒で計算することができた。

一方で、グリッチ伝搬を含む動的電力を得るためにポストレイアウトシミュレーションを利用する場合 SAIF データが必要となる。また、電力分析ツールも不可欠である。ハイエンドサーバで Cadence NC-Verilog と Synopsys PrimeTime を使用してこれらを行なった場合、一つのパイプライン構造について電力を得るのに約 6 分を所用した。つまり、すべてのパイプライン構造を見積もるのは 10 時間以上かかることになり現実的な手段ではない。

よって、短時間に様々なパイプライン構造の電力を得ることのできる本モデルは有用である。

5.3 適したパイプライン構造の探索

提案した探索手法の正確性を確認するために、前述の 5 つのアプリケーションに対していくつかの要求性能のもとで最適化を行った。比較として固定的なパイプライン構造 (1,2,4,8 段パイプライン) の電力も測定した。

例として *gray* を様々な動作周波数で実行した場合の消費電力を図 8 に示す。ここで“提案手法”とは本モデルを用いて探索したパイプライン構造を適用した場合の電力である。1 段パイプラインおよび 2 段パイプラインは $D_{req} = 50$ ns を満たすことができないことを示している。同様に 4 段パイプラインは $D_{req} = 33.3$ ns を満たすことでできていない。この結果から提案手法は要求性能を満たしつつ、適切なパイプライン構造を決定することができている。

各アプリケーションとパイプライン構造に関して、様々な要求性能で平均した消費エネルギーを表 5 にまとめる。*dct* の場合、固定 4 段パイプラインと比べて若干劣る結果となった。

この原因は以下の 2 つである。1 つ目は、提案手法は確実な動作を保証するために、 D_{PE} に最悪遅延時間を使用していることである。ところが、固定パイプラインの場合は実チップの実験によって動作するかを確認した。提案手法では 4 段のパイプ

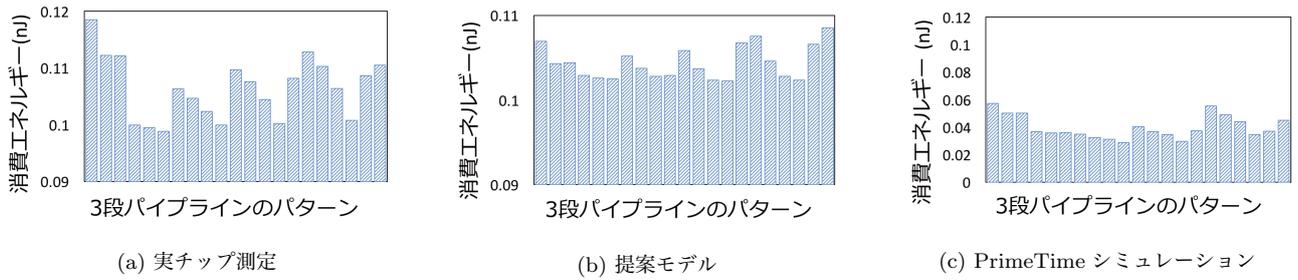


図 7: 動的消費エネルギーの結果

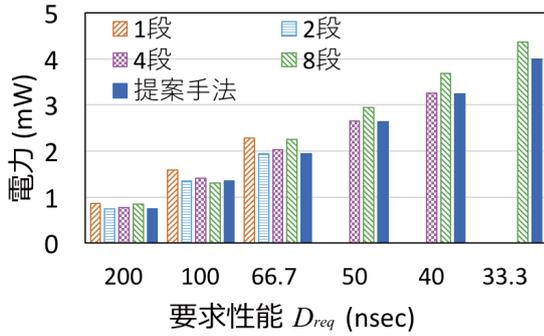


図 8: Power of pipeline structures (gray)

表 5: 各パイプライン構造でのエネルギー比較 (pJ)

	1-stage	2-stage	4-stage	8-stage	Proposal
gray	160.86	137.20	138.97	148.64	134.39
sf	153.89	121.42	122.32	134.18	117.34
dct	212.85	171.71	146.05	153.97	150.06
	1-stage	2-stage	4-stage	7-stage	Proposal
af	138.07	127.90	130.70	139.90	126.45
	1-stage	2-stage	3-stage	6-stage	Proposal
sepia	107.12	100.21	99.21	109.60	98.66

ラインは性能を満たすことができず最適解として排除し、より保守的なパイプライン構造を選択していた。2つ目は、動的電力モデルの誤差が関係している。表 4 で示したように、*dct* の場合、平均で 7.08% の誤差を含んでいる。一見すると他のアプリケーションと比べると小さい誤差であるが、パイプライン段数が小さい時の誤差が大きく、それによって要求性能が低い場合、つまり、小さい段数のパイプライン構造が適している時により良いパイプライン構造を見逃してしまっていた。

6. 結論

我々は CGRA のためのグリッチ伝搬を考慮した動的電力モデルとそれを利用した最適化手法を提案した。本モデルの目的は PE アレイのスイッチング回数を近似的に計算し、エネルギーを見積もることである。ポストレイアウトシミュレーションを利用して見積もった消費エネルギーと比べて、本モデルは 10000 倍以上高速で、なおかつより小さい誤差で消費エネルギーを見積もることができた。本モデルを用いて最適化されたパイプライン構造を利用すると、ほとんどのケースでより小さな消費エネルギーを達成した。

本研究では、PE アレイへのアプリケーションのマッピングは固定して検討していた。しかし、エネルギー効率はグリッチ

伝搬を考慮してマッピングも変化させた方が向上する。本モデルを適用したマッピングツールの開発が今後の課題である。また、本研究の目的は動的電力の最適化であったが、ボディバイアス制御を行い静的電力の削減についても検討すべきである。

謝辞

本研究は、JSPS 科研費 (S) ビルディングブロック型計算システムに関する研究の助成を受けたものである。また、東京大学大規模集積システム設計教育研究センターを通し、シノプシス株式会社の協力で行われたものです。関係者の皆様に感謝致します。

文献

- [1] N. Ando, K. Masuyama, H. Okuhara and H. Amano: “Variable pipeline structure for coarse grained reconfigurable array cma”, 2016 INTERNATIONAL CONFERENCE ON FIELD-PROGRAMMABLE TECHNOLOGY, pp. 231–238 (2016).
- [2] N. Ozaki, Y. Yoshihiro, Y. Saito, D. Ikebuchi, M. Kimura, H. Amano, H. Nakamura, K. Usami, M. Namiki and M. Kondo: “Cool mega-array: A highly energy efficient reconfigurable accelerator”, Field-Programmable Technology (FPT), 2011 International Conference on IEEE, pp. 1–8 (2011).
- [3] Y. Park, H. Park and S. Mahlke: “Cgra express: accelerating execution using dynamic operation fusion”, Proceedings of the 2009 international conference on Compilers, architecture, and synthesis for embedded systems ACM, pp. 271–280 (2009).
- [4] G. Ansaloni, P. Bonzini and L. Pozzi: “Egra: A coarse grained reconfigurable architectural template”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, **19**, 6, pp. 1062–1074 (2011).
- [5] H. Schmit, D. Whelihan, A. Tsai, M. Moe, B. Levine and R. R. Taylor: “Piperench: A virtualized programmable datapath in 0.18 micron technology”, Custom Integrated Circuits Conference, 2002. Proceedings of the IEEE 2002IEEE, pp. 63–66 (2002).
- [6] B. Levine: “Kilocore: Scalable, High Performance and Power Efficient Coarse Grained Reconfigurable Fabrics”, Proc. of International Symposium on Advanced Reconfigurable Systems, pp. 129–158 (2005).
- [7] J. M. Arnold: “S5: the architecture and development flow of a software configurable processor”, Proceedings. 2005 IEEE International Conference on Field-Programmable Technology, 2005.IEEE, pp. 121–128 (2005).
- [8] Y. Morita, R. Tsuchiya, T. Ishigaki, N. Sugii, T. Iwamatsu, T. Ipposhi, H. Oda, Y. Inoue, K. Torii and S. Kimura: “Smallest Vth variability achieved by intrinsic silicon on thin BOX (SOTB) CMOS with single metal gate”, 2008 Symposium on VLSI Technology, pp. 166–167 (2008).