

[研究助成 (A)]

視覚特性を考慮した新計算原理に基づく高効率コンピュータの創生

A high energy-efficient computer based on approximate computing

2221012



研究代表者

東京大学
大学院情報理工学系研究科

助 教

小 島 拓 也

[研究の目的]

マルチメディア処理や機械学習の需要が増加したことにより、計算機システムにおける消費電力は日々増加している。AI 処理に関して言えば、2030 年の全世界での消費電力が 2018 年と比べおよそ 75 倍となると試算されている[1]。したがって、計算機システムにおける省電力化はますます重要になっている。しかしながら、半導体微細化の鈍化によりこれまでとは違うアプローチによって、消費電力の削減を図る必要がある。

従来の計算機は正確な計算結果を高速に導き出すことを求められ、日々進化を遂げてきた。一方で、マルチメディア処理や機械学習の普及に伴いこの常識が覆りつつある。これらの分野では厳密な計算を必要とせず、ある程度の誤りを許容する。こうした特徴を活用して省エネルギー化を目指す新たな計算原理「Approximate Computing」に注目が集まっている。とりわけ、動画処理の分野では JPEG をはじめ、人間の視覚特性に基づいて品質の劣化を抑えつつ、高い圧縮率を実現する符号化技術が開発されてきた。本研究では、このような特性を考慮して Approximate Computing を行う省エネルギー、省リソースな計算機システムの実現を目指す。

本研究では、Approximate Computing に加え、動画処理や機械学習のワークロードと親和性の高いデータフロー型計算機と呼ばれる方

式の計算機を採用し更なる高効率化を目指す。現在の汎用プロセッサはノイマン型コンピュータと呼ばれ、性能上のボトルネックを抱えている。データフロー型計算機はこのボトルネックを解消する次世代の計算機として期待されている。

[研究の内容、成果]

1. 基本アーキテクチャ

前述の通り、本研究ではデータフロー型の計算機をベースのアーキテクチャとする。この構成を図 1 に示す。加算や乗算などの算術演算や論理演算を行う Arithmetic Logic Unit (ALU) と中間データ用のレジスタからなる Processing Element (PE) が格子状に並んでいる。PE 同士が互いにデータを交換し、処理を進めることで、データメモリへのアクセスを削減する。

このアーキテクチャは、PE で行う演算の種類や PE 間の相互接続をアプリケーション

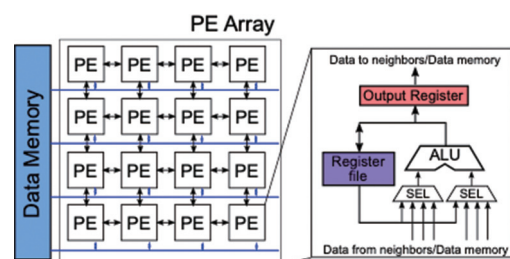


図 1 基本アーキテクチャの構成 (成果 1 より)

に応じて切り替えることができることから、再構成可能ハードウェアの一種であり Coarse-Grained Reconfigurable Architecture (CGRA) と呼ばれる。

先行研究[2]によれば、全体の回路面積の中で6割弱の電力をALUが占めることが示されている。よって、加算や乗算の結果を近似する回路に変更することで、省電力化が可能となる。

2. ハードウェア設計テンプレート

本研究では、PEの実装方法やPEの数、その相互接続を変更し、最適な設計の探索を行う必要がある。そこで、これを効率的に行うために、ハードウェア設計テンプレートを作成した。テンプレートはパラメータ化されたハードウェア設計である。通常ハードウェア設計に用いられるハードウェア記述言語の Verilog-HDL ではパラメータ化に限界がある。そこで、近年注目を集めている Chisel と呼ばれるハードウェア記述言語を採用した。Chisel は Scala 言語のドメイン固有言語として作られており、Scala 言語が持つ関数型言語かつオブジェクト指向型言語という特徴を継承する。また、パラメータ化された設計を容易に行うことができる。

3. 近似加算器および乗算器の調査

計算結果を近似する回路はこれまでに多数提案されている。特に、計算精度と面積、消費電力、および回路遅延などの間にはトレードオフが存在し、用途に応じて適切な方式の近似計算回路方式を選択する必要がある。本研究では、いくつかの方式を検討したのちに、以下の方式の近似加算器と近似乗算器を実際に設計し、予備評価を実施した。

3.1. 近似桁上げ先見加算器

従来の加算器でよく用いられる桁上げ先見加算器では、桁上げ信号の伝搬による加算器の遅延時間を短くするために、桁上げ信号を組み合わせ回路で計算する。 $i+1$ ビット目の桁上げ信号 C_{i+1} は以下の式で計算される。

$$C_{i+1} = G_i + G_{i-1}P_i + \cdots + G_0 \prod_{j=1}^i P_j + C_{in} \prod_{j=0}^i P_j \quad (1)$$

G_i と P_i はキャリー生成項と伝搬項と呼ばれ、それぞれ加数と被加数の i ビット目同士の論理積と排他的論理和である。各桁上がり信号が加数と被加数で計算できるため、高速な加算が可能な一方で、桁数が大きくなるにつれて必要な論理演算数が増加することがわかる。

そこで、近似回路を設計するために、ウィンドウサイズ W を導入して式(1)を2つに分割する[3]。

$$C_{i+1} = \left(\sum_{j=i-W+1}^i G_j \left(\prod_{k=j+1}^{i-1} P_k \right) \right) + \left(\sum_{j=0}^{i-W} G_j \left(\prod_{k=j+1}^{i-1} P_k \right) + C_{in} \prod_{j=0}^i P_j \right) \quad (2)$$

近似計算を行う際は後半部分の計算を省略する。これは、前半部分と比べ、後半部分は下位の桁で計算されるため、省略時に誤差の影響が小さくなることに基づく。また、ウィンドウサイズの選択は精度に影響を与え、小さくするほど省略部分が多くなるため精度が低下する。 W を変更して回路面積を評価した結果を図2に示す。波線は先行研究で提案された近似演算と厳密演算の両方を計算できるようにした場合の回路面積である。一方で、青線にプロットした面積は、各ウィンドウサイズで、近似演算に特化した加算器を設計した場合の面積である。平均してお

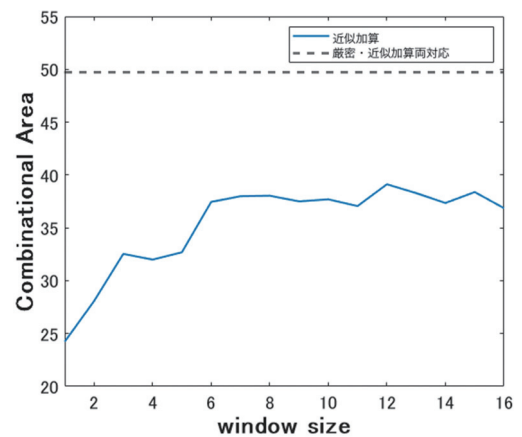


図2 近似加算器の面積削減効果 (成果2より)

よそ 30% の面積削減が得られることがわかった。また、 $W=6$ 程度で削減効果は頭打ちとなることも明らかになった。よって、面積削減効果と精度のトレードオフを考慮して、以降は $W=5$ の近似加算器を用いることとした。

3.2. 4:2 圧縮器による近似乗算器

高速な乗算器の設計では、乗数と被乗数から部分積を並列に計算し、各部分積をキャリー保存加算器で足し合わせる構成が用いられる。

Dadda 乗算器や Wallace 乗算器は、圧縮器と呼ばれる回路を用いて部分積が 2 つになるまでキャリー保存加算によるリダクションが行われる。最後の二数をキャリー伝搬加算によって、最終的な積を計算する。部分和のリダクションは、全加算器や圧縮器と呼ぶ回路をツリー状に接続して計算される。中でも、4:2 圧縮器は、式(3)に基づいて 4 ビットの入力信号 x と桁入力 C_{in} から 2 ビットの出力信号 sum , $carry$ と桁上げ信号 C_{out} を生成する。

$$\begin{aligned} Sum &= x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus C_{in} \\ carry &= (x_1 \oplus x_2 \oplus x_3 \oplus x_4)C_{in} + \overline{(x_1 \oplus x_2 \oplus x_3 \oplus x_4)}x_4 \quad (3) \\ C_{out} &= (x_1 \oplus x_2)x_3 + \overline{(x_1 \oplus x_2)}x_1 \end{aligned}$$

この圧縮器の近似手法として以下の式(4) による回路が提案されている[4]。

$$\begin{aligned} sum &= \overline{\overline{x_1 \oplus x_2 \cdot x_3 \oplus x_4}} \\ carry &= \overline{\overline{x_1 x_2 \cdot x_3 x_4}} \quad (4) \end{aligned}$$

本研究でもこの方式の近似圧縮器を用いた Dadda 乗算器を近似乗算器として用いることとした。

4. 近似計算用データフローアーキテクチャの設計

3 節で述べた近似演算器と従来の厳密計算を行う加算器および乗算器を適切に使い分けることで、面積効率の良い PE の設計を行う。PE 中の ALU がサポートする演算種のパターンとその回路面積を評価した結果を表 1 にまとめる。近似演算と厳密演算の両方を選択的に利用可能

表 1 演算器面積の比較 (成果(2)より)

加算器	乗算器	組合せ回路面積 (μm^2)
両対応		666.4
近似演算	近似演算	310.7
近似演算	厳密演算	314.9
厳密演算	近似演算	480.6
厳密演算	厳密演算	484.7

な演算器は自由度が高い一方で大きな面積を消費することがわかる。

加算と乗算の両方を近似演算のみにした場合、当然もっとも面積が小さいが、全ての演算器をこれに置き換えてしまうと動画像処理を行う際に所望の計算精度を維持できない恐れがある。

そこで、本研究では格子状に多数配置される PE の冗長性に注目し、異なる演算精度の PE を分散させる。コンパイラはユーザーが求める計算精度を満たせるように演算器を適切に選択しながらデータフローのマッピングを行う。これによって、回路面積を削減しつつ、可変精度の計算を実現できる。

5. 計算精度を考慮したデータフローマッピングコンパイラ

データフローマッピングのアルゴリズムを実装するにあたり、我々が開発を進めている GenMap という遺伝的アルゴリズムを用いた最適化ツールを用いた。GenMap は任意の目的関数を追加し、多目的最適化が可能である。本研究では、計算誤差に関する目的関数を追加し、ユーザーからの許容誤差に収まるようにマッピングを最適化できるよう改良を行った。

誤差のモデリングに関しては、先行研究で用いられている誤差分散と誤差感度という指標[2] を本研究でも採用した。これはデータフロー中の演算を近似演算した場合、最終的な出力結果にもたらされる誤差をモデル化したものである。これに基づき、出力結果に与える誤差の影響が少ない計算が優先的に近似演算器で処理されるようになる。

【結 果】

提案手法の有効性を検証するために、回路面積と消費電力の評価を行った。さまざまな構成の評価を実施したが、本報告ではその中でも最も面積・電力削減効果の大きかった図3の構成について述べる。

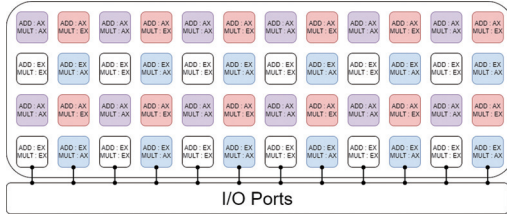


図3 評価した演算器の構成

この構成では4種の異なる演算器が利用されている。紫色の演算器は加算と乗算が共に近似演算回路で行われる。赤色のものは乗算が厳密演算回路、加算は近似演算回路で行われる。青色のものはその逆であり、白色のものは加算と乗算が共に厳密演算回路で行われる。先行研究は全ての演算器を表1の近似演算と厳密演算の両方に対応した演算器のみを用いた構成しているため、それと比較すると組み合わせ回路面積が40%削減されることがわかった。一方で、実際にデータフローをマッピングした際に生じる制約から、演算器アレイ上での占有範囲増加が懸念される。なぜなら、図3の構成において厳密演算が多く必要とされるケースでは、間に存在する紫色の演算器などは利用できず無駄になってしまうことがあるためである。図4は5

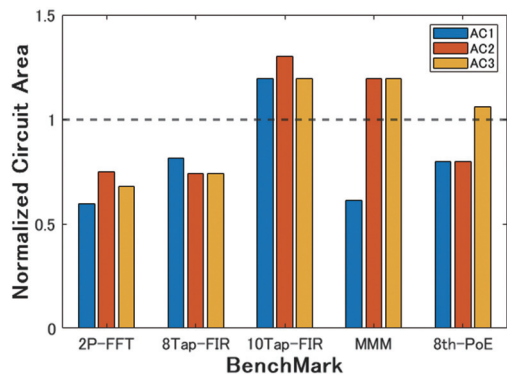


図4 使用部分に関する回路面積の比較

つのベンチマークアプリケーションを実行した場合に占有される演算器の面積を比較したグラフである。AC1, 2, 3はそれぞれ許与される誤差レベルを示し、AC1が最も誤差を許容するケースである。いくつかのベンチマークでは占有面積が先行研究と比べ増加しているが、未使用領域も含めれば前述の通り40%ほど面積が削減されていることに変わりはない。

次に、2GHzの動作周波数における消費電力の比較を図5に示す。本研究では面積の削減に焦点を当てたため、電力削減効果は限定的であるものの、20%程度の削減があるケースが確認された。

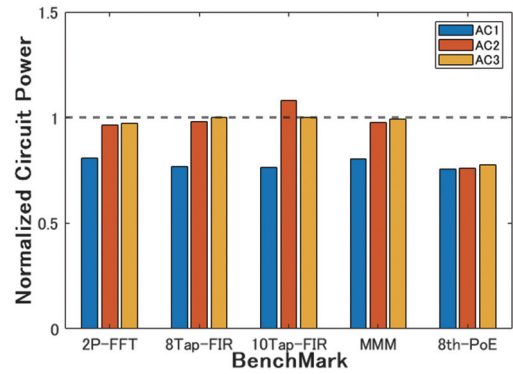


図5 消費電力の比較

【ま と め】

本研究では、動画像処理において演算精度と消費電力や回路面積のトレードオフを最適化する Approximate Computing と非ノイマン型のデータフロー型計算器を組み合わせることで、より高効率な計算機の実現を図った。従来の設計では演算器の柔軟性を維持するために、余分な回路が多く含まれていたため、回路の面積が大きいという問題があった。そこで、本研究ではこれらの面積オーバーヘッドを削減し、それに合わせてアプリケーションマッピングを最適化するコンパイラを開発することで、これまで通りのアプリケーションを動作させたまま、回路面積の削減や消費電力の削減を達成した。

現時点で評価に用いたアプリケーションベン

チマークは小規模なものに限られている。これらを実用的な動画像処理に拡張し、引き続き有効性の検証を行うのが今後の課題である。

[謝 辞]

本研究実施は、東京大学 沓名海斗氏による提案手法の実装や評価、データ収集などの多大な協力により成し遂げることができたものであり、深く感謝申し上げます。

[参考文献]

- [1] 国立研究開発法人科学技術振興機構 低炭素社会戦略センター, “情報化社会の進展がエネルギー消費に与える影響 (Vol.2)”, 2020 年 2 月.
- [2] O. Akbari, M. Kamal, A. Afzali-Kusha, M. Pedram, and M. Shafique, “X- CGRA: An energy-efficient approximate coarse-grained reconfigurable architecture,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, pp. 2558-2571, 2020.
- [3] Omid Akbari, Mehdi Kamal, Ali Afzali-Kusha,

and Massoud Pedram. RAP-CLA: A reconfigurable approximate carry look-ahead adder. Vol. 65, pp. 1089-1093, 2018.

- [4] Omid Akbari, Mehdi Kamal, Ali Afzali-Kusha, and Massoud Pedram. Dual-quality 4:2 compressors for utilizing in dynamic accuracy configurable multipliers. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 25, No. 4, pp. 1352-1361, 2017.

[成果の発表, 論文など]

- (1) 小島拓也, 齋藤 真, 中村 宏, “多様な CGRA を実現する Diplomacy を活用した設計手法の検討”, デザインガイア 2022 —— VLSI 設計の新しい大地 ——, 信学技報, vol. 122, no. 283, VLD2022-22, pp. 19-24, 2022 年 11 月.
- (2) 沓名海斗, 小島拓也, 高瀬英希, 中村 宏, “近似演算器を用いた CGRA とアプリケーションマッピングの協調設計”, 信学技報, vol. 122, no. 402, VLD2022-88, pp. 91-96, 2023 年 3 月