

データ駆動型プロセッサのコンポジットコア化の検討

1255117 古田 雄大 【コンピュータ構成学研究室】

A Study on Composite Core Architecture of Data-Driven Processor

1255117 Furuta, Yudai 【Advanced Computer Engineering Lab.】

1 はじめに

クラウドの負荷分散を担うエッジ・デバイスに搭載されるコアには高性能かつ省電力性が求められている。それを実現する技術として、1つのCPUに異種の複数コアを持つヘテロジニアスマルチコア HMC (heterogeneous multicore) や単一のコア内に異種な複数の実行系を持つ強連結ヘテロジニアスコア TCHC (tightly-coupled heterogeneous core) が提案されている。それぞれの代表例として ARM big.LITTLE, Composite Core (CC) がある [1, 2].

これらの技術はノイマン型プロセッサを対象としている一方で、自己タイミング型パイプライン機構 STP (Self-Timed Pipeline) で動作するデータ駆動型プロセッサ DDP (Data-Driven Processor) は、その構造から割り込み処理を行わず、多様なセンサ等から到着する異なる複数のストリームデータに対する多重処理が可能である高性能性と、クロック信号を用いず、データ入力トリガとなり処理が実行されるため、必要な時に必要な回路のみ動作する省電力性も兼ね備えていることから、エッジ・デバイス用アーキテクチャとして有望である。

異種のコアや実行系を DDP にも導入すれば、それらの利点を兼ね備えたアーキテクチャが実現でき、エッジ・デバイスの更なる高性能化と省電力化が見込まれる。本稿では、DDP の CC 化を提案し、性能および消費電力の観点からの評価結果を報告する。

2 DDP の CC 化

HMC では各コアでそれぞれ独立したキャッシュや分岐予測器を持つため、コア間でのコンテキストスイッチング時のオーバーヘッドが非常に大きくなる問題点がある。一方、TCHC である CC は単一のコア内に、in-order で省電力動作する Little μ Engine と、out-of-order で高性能に動作する Big μ Engine という実行系が備えられている。そのため、HMC と比べて細粒度で実行系の切替制御が可能である。この特性を従来 DDP に取り入れると、以下の特性を持つアーキテクチャとなることが予想される。

- 【CC 特性】タスクに応じて使用する実行系を切り替えられるため、高性能かつ省電力動作が実現できる

- 【CC 特性】実行系間のコンテキストスイッチング時のオーバーヘッドを相対的に低減できる
- 【DDP 特性】両実行系ともに多重処理 (out-of-order 実行) が可能である
- 【STP 特性】クロック信号を必要とせず、データの入力がトリガとなって処理を実行する

DDP の CC 化を行うにあたり、CC の Little μ Engine, Big μ Engine に相当する実行系の導入が必要である。従来 DDP の命令セットは、汎用的な処理 (算術演算, 論理演算, 条件分岐等) の集合であったため、より高機能な命令セットの導入により、Big μ Engine を実現する方法が望ましいと考えた。本研究では、近年注目されている深層ニューラルネットワーク等で多用されるベクトル・行列計算を含む高機能命令セットを定義し、従来命令セット, 提案高機能命令セットを実行制御できるパイプライン機構群をそれぞれ Little μ Engine, Big μ Engine として導入した。

これらの各 μ Engine の実装法として、図 1(a) に示す従来 DDP の演算機構 FP (Functional Processor) とメモリアクセス機構 MA (Memory Access) を、図 1(b) に示す Little μ Engine と Big μ Engine を並列に配置した並列パイプライン構成に置換する。

各 μ Engine での実行を切り替えるために、既存の分流機構および合流機構を EB (Execution Branch Unit) と EM (Execution Merge Unit) として追加している。そして、sFP (Scalar FP) とする既存の FP と既存の

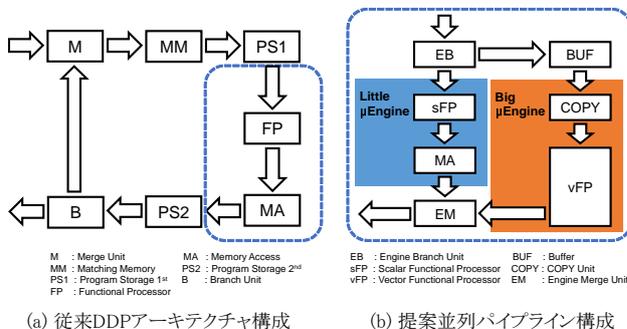


図 1 Little μ Engine, Big μ Engine の実装法

MA を併せて Little μ Engine とし、新規に導入したパケット複製機構 COPY (Copy Unit) とベクトル・行列演算機構 vFP (Vector FP) を併せて Big μ Engine とする。Big μ Engine の回路構成を図2に示す。

COPY は、パケットが vFP に転送される前にパケットを複数回コピーし、それらパケットで一つの命令を実行するためのパケット複製機構である。従来 DDP の構成的な問題やその演算の特徴等により、一度のパケット通過で処理を終えることが難しい命令 (例えば、行列積) を考慮する。パケット複製機能付き C 素子 CC (C Copy) は、パケットが保持する命令コードをもとに生成された CPY フラグを見てパケットを複製するかどうかを判断する。複製する場合は、CC から CP 信号を通常の出力分に加え複製回数分出力させる。その後、各パケットに識別情報を付加して vFP に転送する。

vFP は内部にベクトル・レジスタ、ALU ARRAY を有することを大きな特徴とした演算機構である。パケットが保持するポインタ情報がベクトル・レジスタにセットされ、その情報をもとにオペランドを読み出し、ALU ARRAY に渡す。そのうち、2つ目のオペランドは行列命令の種類によってそれを転置したものや Scalar にもなる。ALU ARRAY は内部に多段に連結した ALU を備え、行列の要素一つ一つに対して同時に処理できる。処理結果はベクトル・レジスタに書き戻す。パケット消去機能付き読込・書込信号生成 C 素子 CWRE (C Write after Read & Erase) は、ベクトル・レジスタからのオペランド読み出しのための初めの CP 信号の出力に加え、処理結果を書き戻すための CP 信号も出力できる。複製したパケットを消去する機能も有する。

また、Big μ Engine での処理時間が長いとパケットが STP 内に溢れ、その間 Little μ Engine での処理も不可能となるため、一時的にパケットを溜めるためのバッファ機構 BUF (Buffer) を Big μ Engine 前に配置する。

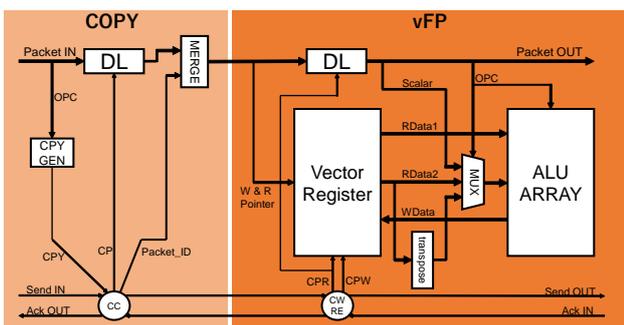


図2 Big μ Engine 回路構成

3 FPGA 回路設計と評価

従来 DDP と提案 DDP を対象に FPGA 回路を設計・実装し、それぞれで行列命令 (今回はスカラー倍、行列積、行列和) の処理時間、Little μ Engine、Big μ Engine に使われる回路資源 (LUT 数、Register 数、BRAM サ

イズ) を測定した。その様子をそれぞれ表1、表2に示す。ただし、今回、行列サイズは 8×8 としている。

表1 従来 DDP, 提案 DDP による各行列命令の処理時間

	従来 DDP	提案 DDP
スカラー倍	179.02 [μ s]	0.848 [μ s]
行列積	1696.37 [μ s]	1.13 [μ s]
行列和	185.15 [μ s]	0.848 [μ s]

表2 Little μ Engine, Big μ Engine に使われる回路資源

	Little μ Engine	Big μ Engine
LUT [個]	420	6257
Register [個]	97	129
BRAM [KB]	2.25	144
Total ¹	519.25	6530

提案 DDP による各行列命令の処理時間は従来 DDP のそれと比べ、表1より、スカラー倍は 211.15 倍、行列積は 1502.41 倍、行列和は 218.37 倍の速度向上を達成した。

両 μ Engine の消費電力 P は Little μ Engine、Big μ Engine の稼働率をそれぞれ α , $1 - \alpha$, 消費電力をそれぞれ P_L , P_B で表すと、(1) 式で求められる。

$$P = \alpha \times P_L + (1 - \alpha) \times P_B \quad (0 \leq \alpha \leq 1) \quad (1)$$

消費電力と回路規模はおおよそ比例すると仮定すると、Little μ Engine、Big μ Engine の消費電力 P_L と P_B は、それぞれ回路資源 R_L と R_B で代替できるため、表2より、 $P_L = R_L = 519.25$, $P_B = R_B = 6530$ 。この時、 α を 80% と仮定すると、 P は $0.8 \times 519.25 + 0.2 \times 6530 = 2136.80$ となる。これは α を 0% つまり、Big μ Engine のみを動作させた時 ($P = P_B = 6530$) と比べ、67.28% 電力を削減できることを示唆している。

4 まとめ

異種のコアや実行系を DDP にも導入すれば、それらの利点を兼ね備えたアーキテクチャが実現でき、エッジ・デバイスの更なる高性能化と省電力化が見込まれたため、本稿では、DDP の CC 化を提案し、性能および消費電力の観点からの評価結果を報告した。評価時、回路設計・実装に重きを置いたため C 素子の遅延回路の最適化を行っていない。今後、真の性能・消費電力評価のために、回路の最適化が残されている。

参考文献

- [1] ARM Ltd., “big.LITTLE Technology: The Future of Mobile Making very high performance available in a mobile envelope without sacrificing energy efficiency,” WHITE PAPER, 2013.
- [2] A. Lukefahr, et al., “Composite Cores: Pushing Heterogeneity Into a Core,” Proc. MICRO, pp. 317—328, 2012.

¹BRAM 1Byte \doteq LUT 1 個 \doteq Register 1 個 と仮定して換算