

データ駆動型プロセッサの環状パイプライン構成法

1265106 高橋 龍一 【コンピュータ構成学研究室】

A Study on Circular Pipeline Configuration for Data-Driven Processors

1265106 Ryuichi Takahashi 【Advanced Computer Engineering Lab.】

1 はじめに

近年, IoT(Internet of Things) 技術の普及により, IoT 機器は幅広い分野で利用され, それに伴い, 高性能化と省電力化の需要が高まっている. そこで, 大域的クロックを用いない省電力なセルフタイム型パイプライン STP(Self-Timed Pipeline) 回路により, 非同期で回路が動作し, 多重並列処理が可能なデータ駆動型プロセッサ DDP(Data-Driven Processor) を実現する技術が研究されている.

本研究では, そのデータ駆動型プロセッサの環状パイプライン構成, すなわち, プログラム中の命令実行制御の実現法を工夫することによって, 性能向上ならびに回路規模削減を試みた. 実行制御の各段階で授受するパケット長を可能な限り短くする観点から, PS(Program Storage) 先行型 DDP を考案し, 従来の PS 後続型 DDP と比較・評価を行った.

2 PS 先行型 DDP

DDP 上でのプログラム実行制御には, 二項演算の実行に必要なオペランド対を検出するマッチングメモリ MM(Matching Memory), 実行される演算コード OP と次の演算ノード dest を読みだす PS(Program Storage), 演算を実行する FP(Functional Processor), が基本的構成要素として必要である [1]. その他にも, パケットを複製するための COPY(Copy Unit), 定数演算を行う際の定数を保持しておく CST(Constant Memory) などの追加的な構成要素がある. これらを用いた従来型 DDP

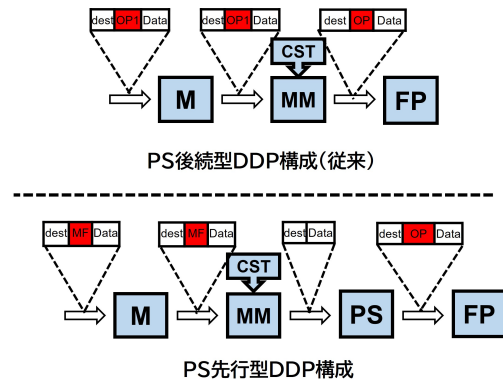


図2 パケット構成の比較

構成を図1(a)に示す.

一方で, これらの構成要素はその接続順序によって, 各構成要素の回路, パケット構成, および, 必要なメモリ量が異なり, 結果として, DDP の実装に必要な回路規模や性能が異なる [2]. 本研究では, これらの組み合わせを網羅的に比較検討を行い, 各パケット長の短縮が高性能化や回路規模削減に寄与することを考慮して, 図1(b)に示す PS 先行型 DDP の構成をパケット形式を含めて検討した.

この構成では, OP を PS と FP 間で局所的に授受するだけで済み, CST に OP を記憶する必要がない. すなわち, 図2に示すように, M から FP 間に, PS 後続型 DDP 構成では FP まで使用することのない OP をパケット入力時から保持しておく必要がある. また, 二項演算の命令コード OP は, 半分に分割されて, OP1 および OP2 として各オペランドパケット内に保持され, マッチング後にビット連結され, 1つの OP を形成しオペランド対パケットとして FP へ転送される. よって, DDP への入力パケットにも, OP1 または OP2 を保持させる必要がある.

これに対し, PS 先行型 DDP 構成では PS で OP を読み出した直後に FP で演算が実行される. しかし, OP をパケット入力時から保持しないためマッチングを行うか否かを判断するフラグである MF(Matching Flag) を新たに追加する必要がある.

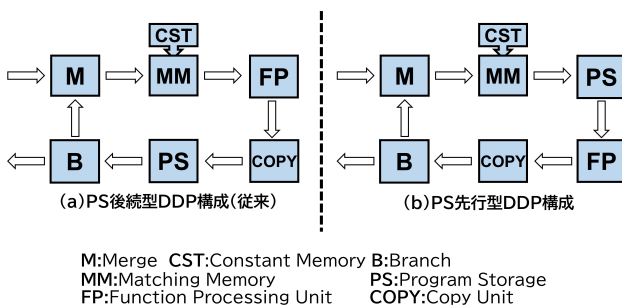


図1 従来の DDP 構成と提案する DDP 構成

3 PS 先行型 DDP の消費電力量

前章に述べたパケット長の短縮により各構成要素の回路規模は低減される．一般に、消費電力は回路規模と強い相関があるが、一方で、DDP における各構成要素の稼働率はプログラムに応じて異なるため、総消費電力量の削減効果を検討する際には、回路規模削減率×稼働率を検討する必要がある．本章では、各構成要素の稼働率を見積もる簡易モデルを検討し、それに基づく総消費電力量の削減効果について考察する．

3.1 パケット流量と稼働率

パイプライン内を周回するパケットの総流量はプログラム中の演算ノード間のデータ依存関係により増減する．二項演算の実行時に MM の前後でパケット数が1減少する．また、パケット複製時は COPY の前後で1増加することになる．入力パケットと環状パイプラインを周回するパケットの合流部 M(Merge) の後では、合流したパケットの総数分が DDP での演算対象となる．一方、環状パイプラインを周回して再度演算対象となるか、あるいは、外部へ出力されるパケットかを判断するパケット分支部 B (Branch) では、出力したパケット以外が周回するパケットとなる．

本研究では、 n 入力 m 出力であり、マッチングが必要な二項演算（要マッチング）、定数を対象とする二項演算（マッチング不要）、および、演算結果の複製を含む任意の非巡回データ依存関係を有するのデータ駆動プログラムを対象として、環状パイプライン内を流れるパケット数のモデル化を検討した．その結果、マッチング後のパケット数は、 $n + C + U - m$ となった．ただし、 n, m, U, C はそれぞれ入力パケット数、出力パケット数、定数演算数、パケット複製回数である

それに基づく、COPY から出力されるパケット量 N_{COPY} は、直前の構成要素から受け取ったパケットのうち複製回数分のパケットが増加する．そのため、受け取るパケット数から C だけ増加したパケット量であるため、

$$N_{COPY} = n + 2C + U - m \quad (1)$$

となる．一方、 M から出力されるパケット量 N_M は、入力されたパケット数 n と周回してきたパケット数の和であるため、

$$N_M = 2(n + C - m) + U \quad (2)$$

となる．式 (1)(2) が示すように、構成要素間で授受されるパケット量には差が生じる．そのため構成要素間で授受されるパケット量が多いと、プログラム実行時により多く稼働するといえる．この稼働率の差が、ひいては消費電力の差に繋がる．以上のことから、各構成要素の回路規模と授受するパケット量を明確にすることで、総消費電力量を算出することが可能となる．

4 評価

本研究では、回路規模を評価するために、データ幅 16bit, MM 容量 64 エントリ, PS 容量 128 命令として、提案 DDP 構成と従来構成とを Xilinx 社の FPGAZynq7000 上に回路実装した．さらに、それらの結果に基づく総消費電力量について、前章に述べた簡易モデルを用いて比較検討した．また、従来の DDP では MM がレジスタで実装されている．しかし、そのことが原因で MMCAM のリソースの使用量が大きくなってしまいう可能性がある [3]．そこで、本研究で提案する DDP と従来の DDP に対して MMCAM を BRAM で実装した場合とレジスタで実装した場合の計 4 つに演算後のデータをメモリに読み書きする構成要素である MA(Memory Access) を追加し、回路を設計する．回路規模の評価を表 1 に示す．

	PS 先行型 DDP		PS 後続型 DDP	
	RAM	REG	RAM	REG
LUT	4164 個	1906 個	5474 個	2102 個
FF	475 個	1680 個	590 個	1681 個
BRAM	8.5 個	2.5 個	6.5 個	2.0 個
	272Kb	90Kb	236Kb	64Kb

表 1 回路規模

各構成要素間の授受するパケット量から算出できる消費電力削減量をそれぞれの簡易モデルに実数値を代入した結果を基に PS 先行型 DDP 構成と PS 後続型 DDP 構成の MM をレジスタで実装した回路で評価する． $n=100, C=30, U=50$ とすると PS 先行型 DDP 構成は 9%消費電力量を削減できる．同様に C, U が最大値となる場合を算出すると、それぞれ 8%, 9%削減できた．

5 まとめ

本研究で DDP のパイプライン構成に着目し、構成要素の接続順序を変更することでより省電力な DDP を検討した．本研究で提案する PS 先行型 DDP 構成は、RAM 型で LUT が 31%増加、FF が 24%減少された．REG 型では LUT が 10%減少、FF が 0.06%減少する結果となった．本結果から、構成要素の接続順序を変更することにより回路規模が縮小し、ひいては消費電力量が削減され、DDP の省電力化が実現できた．

参考文献

- [1] H. Terada, et al., “DDMP’s: Self-Timed Super-Pipelined Data-Driven Processors,” *Proc. of the IEEE*, Vol. 87, No. 2, pp. 282–296, Feb. 1999.
- [2] 高橋龍一, “データ駆動型プロセッサの環状パイプライン構成の比較検討,” KUT 修士学位論文, 2022.
- [3] 松阪拓海, “データ駆動型プロセッサの FPGA 向け電力効率改善手法の検討,” KUT 修士学位論文, 2023.