社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2005－ARC－165（18）
2005／12／1

オンチップ評価機構を搭載した自己タイミング型パイプラインシステムの検討

三　宮　秀　次†　　小笠原　新二†　　岩　田　　誠†

集積化技術の進展によって，近い将来，オンチップ・シミュレータ／エミュレータとしても再構成可能なプロセッサ・コアを多数搭載したチップマルチプロセッサ（CMP）も容易に実現できると考えられる．特に，組込みシステムの分野では，各種の応用領域に特化した組込みチップ・アーキテクチャが要求されるため，ある世代の製品と次世代のプロトタイプが共存できるチップ構成の導入によって，システム設計のコストと期間を大幅に削減できる可能性がある．本稿では，この着想を，自己タイミング型パイプラインによるデータ駆動型マルチプロセッサに適用し，その回路構成法を提案する．

An On-Chip Evaluation Mechanism for Self-Timed Pipelined Systems

Shuji Sannomiya,† Shinji Ogasawara† and Makoto Iwata†

Advance in integrated circuit technology will allow us to develop a chip-multiprocessor equipped with dozens of the processor cores each of which also acts as an on-chip simulator or emulator. Particularly in the area of embedded systems, total design cost and terms could be significantly reduced by integrating both current product and next prototype into a chip-multiprocessor, because application specific architecture is often required, sometimes with dedicated hardware. This paper applies that idea to our self-timed pipelined data-driven multiprocessor chip and proposes a circuit configuration of the processor.

1. はじめに

集積回路システムの大規模化に伴い，システム開発コストの総合的な削減のために，設計やテストを容易化する専用回路を補助的に実装する手法が活用されている．例えば，テスト容易化のための BIST 回路[1] や，回路最適化を簡易化できるオンチップ波形測定回路[2] などがある．本研究では，これらの考え方をアーキテクチャ設計の容易化に応用し，オンチップ・シミュレータ/エミュレータ機構を搭載したチップマルチプロセッサ構成を構想し，これを容易に実現可能な自己タイミング型パイプライン（STP）システムに適用することを試みている．

特に組込みシステムの分野では，個々の応用領域に特化したチップマルチプロセッサ（CMP）が要求される．例えば，異なる命令セットを備えたヘテロジニアスなマルチプロセッサ構成や，特定の処理に特化した専用エンジンと協調可能なマルチプロセッサ構成が求められる．しかしながら，開発期間が半年〜１年と短いため，ハードウェアを事前に試作して，評価する余裕がないのが実情である．これに対して，様々なアーキテクチャ構成を容易に評価可能な機構をチップ上に搭載した CMP を応用分野の技術者に提供すれば，応用分野の視点から必要なアーキテクチャ項目を具体的に評価可能なため，次期チップの設計コストを大幅に削減できる可能性がある．

本論文では，この構想を STP によるデータ駆動型マルチプロセッサ[3] に適用する方法を提案する．STP は，隣接するステージ間でのみデータ転送制御信号を局所的に授受（ハンドシェイク）するため，隣接するステージ間のハンドシェイクのみを保障すれば，システム全体の論理動作を保障できる．したがって，様々なアーキテクチャ構成の評価に必要な付加的な回路を部分的に追加しても，追加したパイプライン段のみのタイミング制約を守れば，他の箇所には一切影響を及ぼさないため，容易にチップ上にシミュレーション／エミュレーション回路を追加できるという特徴を備えている．このような特徴によって，通常時はプロセッサとして機能するが，評価時にはシミュレータ／エミュレータとしても機能できる，多数のプロセッサ・コアを搭載した CMP が構成できる．

以下本稿では，STP によるデータ駆動型マルチプロセッサ・アーキテクチャの機能・性能評価の要件を議論した後，オンチップ・シミュレーション／エミュレーション回路の構成を提案し，回路規模および評価性能の評価結果を紹介する．

2. オンチップ評価の要件

自己タイミング型パイプライン STP によるデータ駆動型マルチプロセッサ DDMP・チップを活用した応用システムの開発では，通常，データフローグラフ（DFG）を用いて応用システムの機能仕様を定義する手法が有効である[4]．これは，データ駆動プロセッサでは，割込み制御やスケジューリング等，本来の処理とは無関係の付加的なコードを必要としないため，DFG を階層的に精錬化すれば，機能仕様から実行コードまでシームレスに詳細設計が

† 高知工科大学
Kochi University of Technology

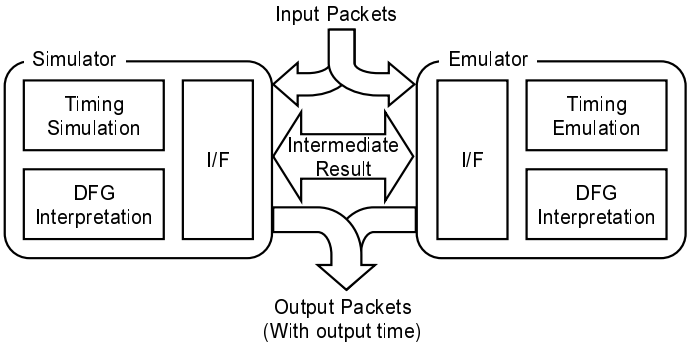

図 1 オンチップ・シミュレーション／エミュレーション構想

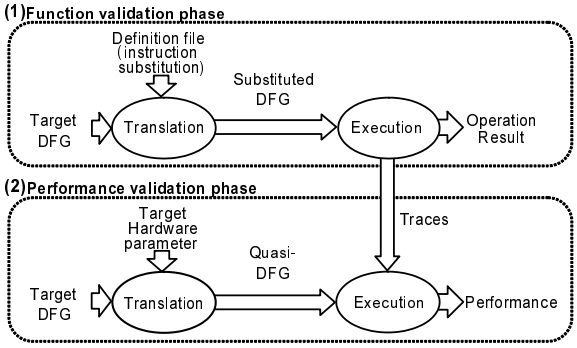

図 2 バイナリ変換によるシステム検証

可能になるためである．この段階的設計の過程で，応用システムの性能要求を満たせるように，ソフトウェアで実装する機能とハードウェアで実装する機能を切り分ける必要がある．もし，要求性能が満足できなければ，複合命令セットや専用ハードウェアエンジンなどを導入した新規アーキテクチャの設計も必要になる．そのために，新規アーキテクチャ上で実行される応用プログラムの定量評価を，なるべく高速で精密に行える評価ツールの開発が望まれている．

筆者らはこれまでに，STP システム上でのパケットの振る舞いを簡易に模擬できる，マクロフローモデルを提案し，シミュレーション時間の短縮を図っている[5]．また，このシミュレーションと連携可能な，トレース駆動型エミュレーション機構を提案し，性能評価時間をより一層短縮する工夫をしている[6]．シミュレーションは詳細に実行時の情報が得られるが低速である．実際，マクロフローモデルにより，ワークステーション上でのシミュレーションを 2~5 倍まで高速化したが，システムの実行時間に比して遥かに低速である．一方，エミュレーションでは，ほぼシステムの実行時間内での評価が可能であるが，パイプライン段数などのハードウェア構成に既存のものを利用する点で，柔軟性に乏しい．

これらの欠点が設計作業を妨げないよう，高速かつ柔軟なシステム評価を可能とする，オンチップ・シミュレータ／エミュレータ機構を提案する．本機構は，図 1 に示すように，新規のハードウェア構成を伴う DFG のサブグラフを，マクロフローモデルに基づきシミュレーションする．同時に，既存のハードウェア構成で模擬できるサブグラフをエミュレーションする．さらに，シミュレーションとエミュレーション結果を統合するインタフェース機構により，システム全体をチップ上で高速に評価する．

提案機構に基づくシステムの検証では，図 2 に示すように，命令の論理的な動作と実行タイミングを独立して評価する．

機能検証

データ駆動原理では，データの到着順序に依らずに，演算結果を保障するため，DDMP システムの機能は，性能とは独立に評価できる．これを利用して，対象の DFG を，機能的に等価な既存の命令セットで置換することで，直接的に実行することができる．そのため，システム機能は，ほぼ実行時間内で評価できる．よって，オフラインのシミュレーションに比べて，遥かに高速に検証が可能である．

性能検証

上記の機能検証の過程で，性能を左右する，分岐先等の実行時情報（トレース）をあらかじめ取得しておき，これを反映するための擬似命令を用いて，対象の DFG 中の新規命令を置換し，擬似 DFG を得る．次に，トレースを基に命令の実行タイミングを再現しながら，擬似 DFG を実行することによって，性能を計測し，評価する．ただし，単純な擬似命令の置換によってだけでは，正確な実行タイミングが再現できない場合には，タイミングのシミュレーションが必要となる．

このように対象命令をバイナリ変換する手法では，サイクルタイムの正確さが保障できないという問題が指摘されている[7]．この要因としては，メモリ階層の構成が性能を支配する点が大きい．DDMP では，DFG によりプログラムが与えられるため，データのプリフェッチが容易に行える．このため，こうしたデータ配置の問題を容易に回避できる．一方，STP は負荷の変動に対する緩衝能力を備えており，性能評価においては，この特性を正確に反映できるよう，STP のハンドシェイク・タイミングを正確に模擬する必要がある．また，トレースの評価時間が，ボトルネックとならないよう，回路を構成する必要がある．

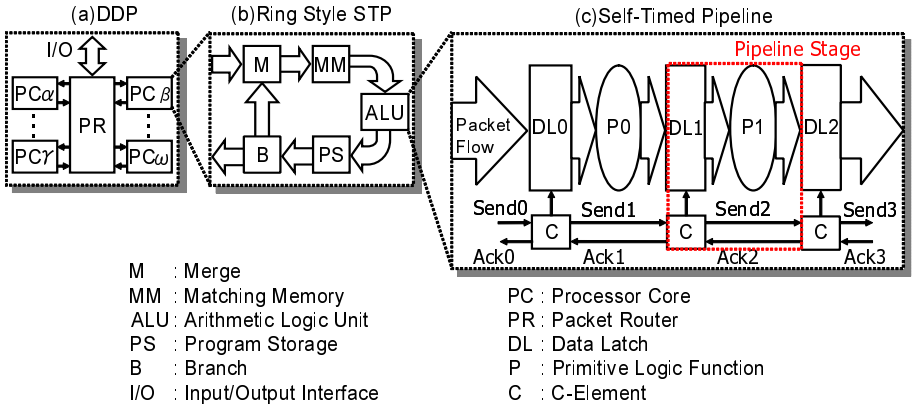

図 3 DDMP コアの基本構成

3. オンチップ・シミュレータ／エミュレータ機構

DFG の実行では，命令の論理的な解釈と，正確なハンドシェイクを再現する必要がある．このため，命令の解釈・実行を STP に実装した DDMP コアを用いた，直接的な DFG の実行が有効である．本章では，まず，DDMP コアの構成を概説し，エミュレーションに必要なトレース評価機構を示す．次いで，オンチップ・シミュレーション／エミュレーション回路の構成を示す．

3.1 DDMP コアの回路構成

図 3 に示すように，DDMP コアでは，データを分流／合流するモジュール（B/M）を配して，リング型の STP を形成している．このリング型 STP を用いて，待ち合せ（MM），演算（ALU），及びプログラム・フェッチ（PS）といった，データ駆動原理に基づき命令を解釈・実行するモジュールをパイプライン分割する．このような構成により，空間的な並列性を時間的な（パイプライン）並列性に重畳することができる．すなわち，各種粒度の並列性を自然に引き出すことで，高いパイプライン処理効率を達成できる．また，命令の実行時間は，データがリング型 STP を 1 周する時間（周回時間）で規定できる．この周回時間は，STP 中のデータ（パケット）数（P_{total}）を基準に変化する[5]．

以上より，(a) DFG を解釈・実行して，P_{total} を変化させるトレースを取得して，(b) P_{total} に基づいて，周回時間を再現することで，性能が計測できる．オンチップ・シミュレーション／エミュレーションでは，既存ハードウェア構成に割り当てられる，擬似 DFG のサブグラフ（命令シーケンス）を，エミュレーション機構を持つ DDMP コアで実行する．一方で，マクロフローモデルに基づき，周回時間を再現するシミュレーション機構を持つ DDMP コアで，新規ハードウェア構成に割り当てられる命令シーケンスを実行する．

3.2 オンチップ・エミュレーション回路

DDMP では，パケットは DFG に沿って STP 内をフローするため，動的な P_{total} は分岐命令により，大きく変動し得る．また，オフチップ SDRAM へのアクセスを伴うような複合命令では，ハンドシェイク時間を延長して，周回時間を僅かに延長する必要がある．これらのことから，トレースとして，分岐方向と命令コードが必要となる．これらのトレースを評価するため，図 4 に示すように，可変遅延（VD）機構と分岐履歴メモリ（BHM）機構を DDMP コアに追加する．

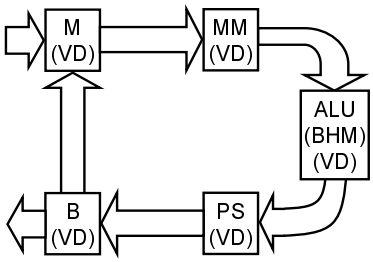

図 4 エミュレーション回路の構成

VD 機構

ハンドシェイク時間は，データ転送制御信号を任意に遅延させることで調節できる．VD は，選択信号により排他的に選択できる複数の遅延素子（D）を持っており，擬似命令の命令コードにより，遅延を選択することで，複数のハンドシェイク時間を提供する．このように動的なハンドシェイクの調節を保障するには，データ転送要求（send）信号の到着に先立って，選択信号は安定している必要がある．この制約を満たすため，図 5 に示すように，VD 機構は，2 ステージにパイプライン分割する．すなわち，まず，第 1 ステージで，VD_Dec. は命令コード（OPC）を解釈し，擬似命令に応じた遅延選択信号（Select Signal）を生成する．その後，後続するステージで，VD の遅延が選択信号により有効となる．このようなパイプライン実装により，VD_Dec. の処理時間は，データパス中の機能ロジック（P0）に隠蔽できるため，ハンドシェイク時間を動的に調節することによるレイテンシの増加はない．

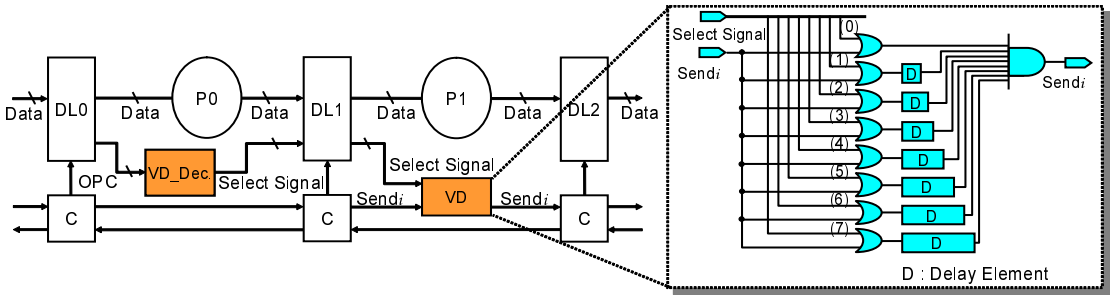

図 5 VD 機構のパイプライン実装

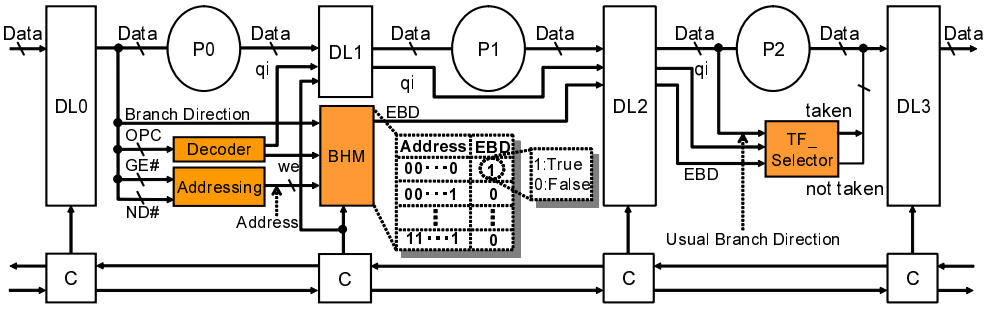

図 6 BHM 機構のパイプライン実装

BHM 機構

BHM は，分岐方向を格納するメモリである．格納される分岐方向は，taken を'1' とし，not taken を'0' として，1bit に符号化することで，必要となるメモリ容量を抑える．BHM は，書き込み（wb）と読み出し（rb）を行う擬似命令に対して動作する．すなわち，まず，性能評価に先立って，符号化された分岐方向（EBD）は，wb 命令により BHM へ格納される．性能評価では，rb 命令を，BHM 回路通過時に EBD に従って分岐させることにより，rb 命令は，恰も対象 DFG の分岐命令のように振舞う．

分岐命令の実行は，通常，命令コードの解釈，条件式の評価，及び分岐方向の選択といった，3 ステージにパイプライン分割される．図 6 に示すように，この構成では，条件式の評価を，BHM のアクセスに切り替えることで，wb と rb を実現できる．すなわち，まず，命令コードの解釈において，命令コードが wb であった場合，書き込み許可信号（we）をアサートする．一方，命令コードが rb であった場合，qi フラグを立てることにより，擬似命令の実行を後続ステージに伝える．DDMP コアでは，分岐命令の実行は，パケットのタグ情報である世代番号（GE#）と宛先番号（ND#) により検知できる．これらの情報に基づき，Addressing は，BHM の実効アドレスを算出する．第 2 ステージには，BHM が SRAM として実装されており，EBD を出力する．第 3 ステージでは，qi フラグが立っていれば，すなわち rb 命令の実行であれば，EBD を復号した結果をパケットに分岐方向として付帯させる．このような構成では，BHM に対するアクセス時間をデータパス上の機能ロジック（P1）に隠蔽することができるため，分岐方向を評価することによるレイテンシの増加はない．

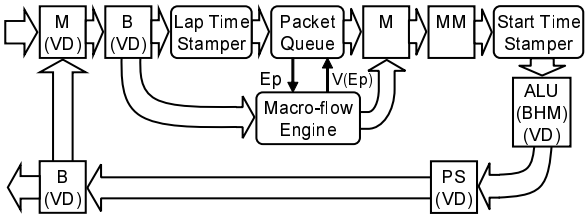

図 7 シミュレーション回路の構成

これらの実装により，BHM と VD によるトレースの評価時間は隠蔽できる．DDMP コアは，これらのエミュレーション機構を搭載することで，擬似 DFG を直接的に実行し，正確なタイミングで結果パケットを出力できる．

3.3 オンチップ・シミュレーション回路

シミュレーションでは，周回時間の算出を簡略化できる，マクロフローモデルを用いる．マクロフローモデルは，リング型 STP の周回時間とパイプライン利用率（E_p）の関係を，パケットの速度（$V(E_p)$）で代数的に定量化している．すなわち，$V(E_p)$ は E_p に応じて 3 段階に変化する．

$$\text{Phase1}\ (\sum T_f \leq max(T_f + T_r) \times P_{total})$$

$$V(E_p) = \frac{pl}{\sum T_f}$$

$$\text{Phase2}\quad (\sum T_r < max(T_f + T_r) \times B_{total})$$

$$V(E_p) = \frac{pl}{\sum T_f + max(T_f + T_r) \times P_{over}}$$

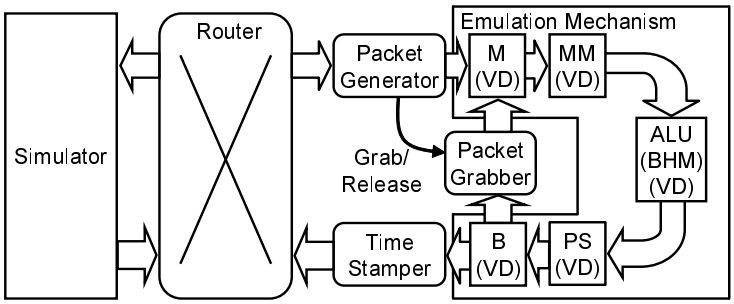

図 8 停止・再開回路の構成

$$\text{Phase3 } V(E_p) = -\frac{1}{2plC_1}(P_{total})^2 + C_2 \cdot P_{total} + C_3$$

ここで，B_{total} と P_{over} は，それぞれ，STP 中の総バブル数と Phase1 を超過するパケット数を意味する．また，$(T_f + T_r)$ は，ハードウェア構成で定まる，ステージ毎のハンドシェイク時間を意味する．さらに，C_1，C_2，及び C_3 は定数であり，$(T_f + T_r)$ により定まる．E_p は，ステージ数を pl とすると，$\frac{P_{total}}{pl}$ と規定されるが，通常，pl は，ハードウェア構成に対して固定（定数）である．つまり，新規ハードウェア構成における周回時間は，$V(P_{total})$ で規定できる．

マクロフローモデルに基づくシミュレーションでは，P_{total} の変化を追跡するために，擬似 DFG の実行・解釈が必要となり，オーバヘッドとなり得る．また，$V(P_{total})$ に基づく周回時間の動的な変更には，VD を利用できる．これらのことから，図 7 に示すように，エミュレーション機構を搭載する DDMP コアを拡張する．

オンチップ・シミュレーション機構では，擬似 DFG を直接的に評価する．すなわち，まず，評価に先立って，$V(P_{total})$ を Macro-flow Engine に格納する．また，トレース情報と擬似 DFG を，それぞれ，BHM と PS に格納する．入力されたパケットは，Packet Queue で一時的にキューイングされる．Packet Queue は，キューイングされたパケットのタイムスタンプ値から，P_{total} を算出する．Macro-flow Engine は，算出された P_{total} に基づき，各 VD を適切に選択することで，パケットの周回時間を調節する．その後，Packet Queue から取り出されたパケットは，Start Time Stamper で周回開始時刻を与えられた後，STP を周回して，Lap Time Stamper でタイムスタンプ値に周回時間が積算される．このような構成により，1 命令の模擬が，ほぼ 1 命令の実行時間内で行えるため，シミュレーション時間は大幅に短縮できる．

3.4 インタフェース回路

オンチップ・シミュレーション機構は，擬似命令の高速な模擬を可能とする．しかし，エミュレーション機構は擬似 DFG 全体を直接的に実行するという点で，シミュレーションが性能評価のボトルネックとなり得る．これに対して，一部の DFG に対して，評価に必要なパケットに対するシミュレーションが終了した時点で，直ちにエミュレーションを開始することで，シミュレーション時間の一部を隠蔽できる．このような部分エミュレーションを実現するため，図 8 に示すように，エミュレーションの停止・再開機構を設ける．

停止・再開機構により，一定数量のパケットをバッファ後，直ちにエミュレーションは着手される．すなわち，まず，シミュレータや他のエミュレータから入力されるパケットは，Packet Generator に一時的にバッファされる．一定量のパケットをバッファ後，Packet Generator は，パケットが到着した間隔で，パケットをエミュレータへ放出する．その際，バッファされているパケット数が一定値を下回ると，エミュレーションの停止を意味する Grab 信号をアサートする．これを受け，Packet Grabber は，STP 中の全パケットを格納することで，擬似 DFG の実行を停止する．その後，Packet Generator が再び一定量のパケットをバッファ後，パケットの放出を開始し，同時にエミュレーションの再開を意味する Release 信号をアサートすることで，Packet Grabber は格納されたパケットを開放して，擬似 DFG の実行が再開される．このような構成により，エミュレーションとシミュレーションは可能な限り並行して実行できる．

3.5 事前評価

回路規模

オンチップ・エミュレーション回路の実行可能性を検証するため，FPGA を用いた実装を行った．FPGA チップには，オンチップ・エミュレーション回路を収容できる論理要素数を持った，Altera 社 APEX20KE EP20K1500 を用いた．また，論理合成と配置配線には，それぞれ，Mentor-Graphics 社 LeonardoSpectrum と Altera 社 QuartusII を用いた．結果，DDMP コアの機能やスループットには影響を与えずにトレースが評価できることを確認した．

配置配線後，データシートに基づき，論理要素数を標準ゲート数へ換算した結果を表 1 に示す．実装では，BHM のサイズは，RAM 領域の制限により，4k[bit] としたが，実効アドレスは GE# と ND# より直接求められるため，BHM の容量増加は僅かな回路規模の増加に収まる．一方，VD は，ALU の一部（10 ステージ）にのみ実装した．結果よ

表 1 回路規模の比較

	DDMP Core	DDMP Core+VD+BHM
Logic Elements	44477	46196
Gates ∗	1286950	1336690
Ratio	1	1.0386

∗gates/logic element ≈ 29

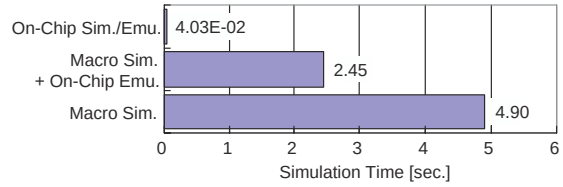

図 9 シミュレーション時間の比較

り，エミュレーション回路の追加による回路規模の増加は，約 4%程度であった．現行の DDMP コアの総ステージ数は，約 60 であるため，全ステージに VD を配する場合であっても，回路規模は約 2 割程度の増加に収まっており，許容の範囲内と考えられる．

現在，オンチップ・シミュレーション回路とインタフェース回路を設計しており，それらの実行可能性と回路規模の検証が残されている．

評価性能

提案したオンチップ評価機構と既存の評価手法で，シミュレーション時間を比較した．結果を，図 9 に示す．対象システムは，ラプラシアン 4 近傍フィルタとした．マクロフローモデルに基づくシミュレータ（マクロシミュレータ）は Java を用いて実装し，ワークステーション（Pentium4 3GHz）上で実行した．オンチップ・エミュレータは，マクロシミュレータで模擬した．1 画像データに対する，フィルタシステムの実行命令数は，約 0.4M であった．これに対して，マクロシミュレータのシミュレーション時間（Macro Sim.）は，約 4.9[*sec.*] であった．また，エミュレーション時間に関しては，約 670[us] であった．これらのことから，演算が均等に分散すると仮定して，マクロシミュレータとオンチップ・エミュレータを併用した場合のシミュレーション時間を見積った（Macro Sim. + On-Chip Emu.）．また，前述の通り，オンチップ・シミュレータでは，1 命令の模擬に掛かる時間は，パケットが STP を 1 周する時間にほぼ等しいが，これは現行の DDMP コアでは約 200[ns] である．これに基づき，演算が均等に分散すると仮定して，オンチップ・シミュレータ／エミュレータによる，シミュレーション時間を見積った（On-Chip Sim./Emu.）．結果，オンチップ・シミュレータ／エミュレータは，既存の評価手法に比べて，60 倍以上の高速なシミュレーションが可能であると見積られた．

4. おわりに

本稿では，アーキテクチャ設計の容易化を目指し，マクロフローモデルとトレース駆動型エミュレーションに基づくオンチップ・シミュレータ／エミュレータ機構を搭載するデータ駆動型マルチプロセッサ構成と，その回路構成法を提案した．回路構成法の内，オンチップ・エミュレータ機構に関しては，パイプライン実装を提案し，FPGA を用いて，約 2 割程度の回路の投資で実現できることを確認した．一方，提案した構成により 60 倍以上のシミュレーションの高速化が期待でき，大幅な設計作業の効率化が期待できる．

提案したオンチップ・シミュレータ機構は，エミュレータ機構を内包しているため，アーキテクチャ構成に応じて，動的に切り替えて使用可能である．もし，これらの回路コストの投資が，開発コストの削減で回収できれば，すべてのプロセッサ・コアに，オンチップ・シミュレータ機構を搭載して，シミュレーションの柔軟性を最大化することも可能である．今後，提案機構を搭載したプロトタイプ・チップの LSI 試作を通して、回路コストの評価ならびに性能評価精度の実測を行うと共に、開発コストの削減効果についても定量的に把握することが課題である．

参 考 文 献

1) O. A. Petlin, S. B. Furber, “Built-In Self-Testing of Micropipelines,” in Proc. of Third International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC ’97), Eindhoven, Netherlands, pp.22–29, Apr. 1997.
2) M. Takamiya, M. Mizuno, and K. Nakamura, “An On-Chip 100GHz-Sampling Rate 8-channel Sampling Oscilloscope with Embedded Sampling Clock Generator,” IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, USA, pp.182–183, Feb. 2002.
3) H. Terada, S. Miyata, and M. Iwata, “DDMP’s: self-timed super-pipelined data-driven multimedia processors,” Proc. IEEE, Vol.87, No.2, pp.282–296, Feb. 1999.
4) D. Morikawa, M. Iwata, and H. Terada, “Super-pipelined implementation of IP packet classification,” Journal of Intelligent Automation and Soft Computing, Vol.10, No.2, pp.175-184, Aug. 2004.
5) S. Sannomiya, Y. Omori, and M. Iwata, “A Macroscopic Behavior Model for Self-Timed Pipeline Systems,” Seventeenth Workshop on Parallel and Distributed Simulation (PADS 2003), San Diego, USA, pp.133-140, Jun. 2003.
6) S. Ogasawara, S. Sannomiya, Y. Oomori and M. Iwata, “Implementation of An On-Chip Trace-Driven Emulation for Self-Timed Data-Driven Processors,” International Conference on Next Era Information Networking (NEINE ’05), Shanghai, China, pp.548-553, Sep. 2005.
7) 中田尚，大野和彦，中島浩, “高性能マイクロプロセッサの高速シミュレーション,” 先進的計算基盤システムシンポジウム（SACSIS 2003）論文集, pp.89-96, May. 2003.