

自己タイミング型パイプラインシステムの性能見積りモデル

三宮 秀次[†] 大森 洋一^{††} 酒居 敬一[†] 岩田 誠[†]

A Performance Estimation Model of Self-Timed Pipelined Systems

Shuji SANNOMIYA[†], Yoichi OMORI^{††}, Keiichi SAKAI[†], and Makoto IWATA[†]

あらまし 自己タイミング型パイプライン (STP) は、大域的なクロック信号による制御なしに、隣接するパイプライン段間のハンドシェイク信号のみで大規模なパイプライン処理システムを構成できる回路構成法として有望である。しかし、パイプライン段間のハンドシェイク信号の伝達を逐一模擬する、従来のシミュレーションでは、迅速な性能見積りが困難であった。本論文では、大規模 STP システムの高速シミュレーションのために、リング型パイプライン内のパケットフローをマクロにとらえる性能見積りモデルを提案している。提案モデルでは、STP 内の総パケット数 (負荷) が平均的なパケットの移動速度に与える影響を代数的に定式化した。本モデルによって、シミュレーションに要する計算量を 50%以上削減できる。また、提案モデルに基づくシミュレーションによる性能見積り結果を、実際の STP チップの実測値と比較した結果、LSI 試作前の事前評価シミュレーションとしては十分な精度で見積りが可能なことも確認した。

キーワード 自己タイミング型パイプライン, 性能見積り, シミュレーション, マクロフローモデル

1. ま え が き

大規模な集積回路チップシステムをすべてパイプライン実現する設計手法は、徹底的な分割統治型ハードウェア構成法の一つである。つまり、パイプライン化による処理能力向上だけでなく、設計時の組合せ論的爆発の問題を回避できる [1]。特に、自己タイミング型パイプライン機構 STP (self-timed pipeline) は、隣接するパイプライン段間でのみデータ転送制御信号を局部的に授受するため、配線や電力消費を極小化できるとともに、データ転送時のみ電力を消費する自律的な省電力機能も有している [2], [3]。これらの STP の特徴を活用して、高速データパス回路、非同期回路プロセッサ、大域非同期局所同期システムなど多数の応用が検討されている [4] ~ [6]。筆者らも既に、STP と極めて親和性の高いデータ駆動型メディアプロセッサ DDMP (data-driven multimedia processor) を開

発し、映像信号処理やネットワークプロセッサへの応用を検討している [2], [7]。

本論文では、この STP を徹底的に活用して更に大規模なシステムを組織的に構成する上で不可欠なリング型 STP の動作モデルを定式化して、これに基づく高速な性能評価法を提案する。これにより、データ転送を逐一綿密に模擬する従来の手法に比べて 3 倍以上高速に性能見積りが行えることを示す。

一般に STP システムでは、信号の遷移やデータ授受などのあらゆるイベントが、クロック同期システムのように同期して離散化されず、すべて因果関係のみによって生起する。このため、回路シミュレーションやシステムレベルの動作シミュレーションには多大な時間を要する [8]。これまでにも、マルコフ連鎖による確率過程解析モデル [9] や確率的なタイミング解析 [10]、あるいは回路レベルの信号タイミングをモデル化した Charie 図 [11] に基づくタイミング解析が研究されているが、いずれのモデルでも具体的な入力データやプログラムを STP システム上で実行する際の性能見積りを想定していない。また、従来の、時間ベトリネット [12] 等に基づき個々のデータ転送を逐一追跡する手法では、データ数に比例して計算コストが掛かるため、性能見積りの高速化が困難であった。

[†] 高知工科大学大学院工学研究科, 香美市
Graduate School of Engineering, Kochi University of Technology, Tosayamada-cho, Kami-shi, 782-8502 Japan

^{††} 九州工科大学大学院システム情報科学研究所, 福岡市
Graduate School of Information Science and Electrical Engineering, Kyushu University, Hakozaki, Higashi-ku, Fukuoka-shi, 812-8581 Japan

これに対して本論文では、リング型 STP 内で転送されるデータパケット（以降、単にパケット）の速度 V [パイプライン段/秒] に着目し、STP 内の総パケット数が $\frac{dV}{dt}$ に与える影響を代数モデルとして定式化する。これにより、パイプライン段間で個々のパケットが転送されるタイミング（因果関係）を逐一綿密に考慮しなくとも、マクロなパケットフローに関する動作モデルのみで性能評価が可能になる。以下、2. では STP の動作特性を示し、3. でリング型 STP のマクロフローモデルを提案し、4. において提案モデルに基づくシミュレーションの精度が十分実用的であることを、実際の STP チップの実測結果を用いて検証する。

2. STP システムの動作特性

本章では、まず STP の動作原理を示す。そして、STP 内で転送されるパケットの速度に着目して、性能特性を議論する。更に、STP 内の総パケット数（すなわち、負荷）がパケットの速度に与える影響を考察し、STP システムの動作特性を議論する。

2.1 STP の動作原理

STP は基本的には図 1 のように構成される。各パイプライン段（以下、単にステージ）は、データラッチ、処理回路、及び一致記憶フリップフロップ（Coincidence flip-flop : C 素子）により構成される。直線状の STP 内では各パケットはステージ間の制御信号（send 信号、ack 信号）の伝達によって自律的に移動する。図 2 に、制御信号の遷移を示す。ここでは、send 信号と ack 信号の伝達には、立上りと立下りを同じ意味の信号とする遷移シグナリング [13] を用いている。一方、Master Reset 信号を 0 にすることによって STP 回路全体がリセットされ、また ToDL 信号が 1 のときに DL がオープンされる。

すなわち、まず、リセット時には、すべての C 素

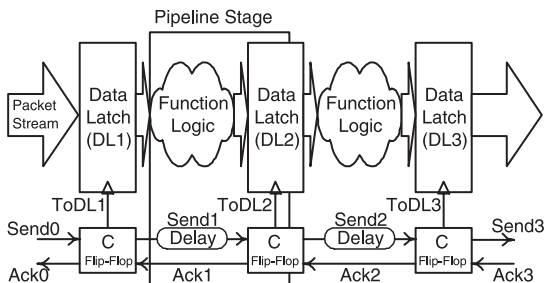


図 1 自己タイミング型パイプライン機構
Fig. 1 Self-timed pipeline (STP).

子は後方ステージへ ack 信号を伝達する（図 2(0)）。その後、Master Reset 信号を 1 にしてリセットを終える。

(1) (パケットの転送開始) C 素子が前方ステージへ $send_{i-1}$ 信号を伝達する。同時に、データラッチが前方ステージへパケットを送信する。

(2) (ハンドシェイク) C 素子は $send_{i-1}$ 及び ack_i 信号が到着すると、データラッチ DL_i を開ける。結果、パケットが転送される。

(3) (ack 信号遷移) 同時に C 素子は、 ack_{i-1} 信号を後方ステージに伝達し、後続パケットの転送を許可するとともに、

(4) (send 信号遷移) $send_i$ 信号を前方ステージに伝達し、前方ステージへのパケットの転送を開始する^(注1)。

(5) パケットがある限り(2)~(4)を繰り返す。

厳密には、ハンドシェイクのプロトコルには、前述の遷移シグナリングと、信号レベルが 0 あるいは 1 のときに信号の伝達を意味するレベルシグナリングがある [13]。レベルシグナリングでは、1 回のハンドシェイクの間に、send 信号と ack 信号が、それぞれ 2 回遷移するが、パケットの転送手順は同じである。したがって、本論文では、両プロトコルを包含した一般的な STP システムについて議論する。

以上のようなパイプライン段間の局所的な転送制御により、STP には、(a) 局所的な信号伝達のみによって動作時にだけ電力を消費する省電力特性や、(b) 負

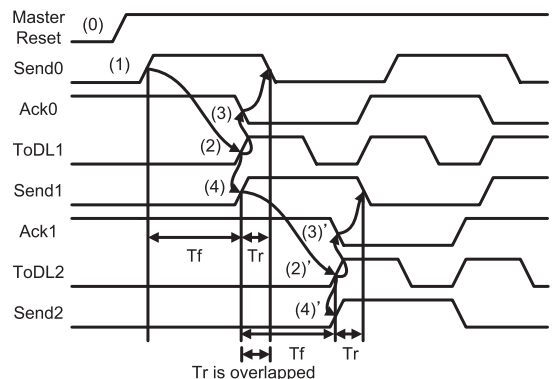


図 2 タイミングチャート
Fig. 2 A timing chart of STP.

(注1): 通常、パケットの値の正当性を保証するために、send 信号と ack 信号の遅延はそれぞれ、処理回路のクリティカルパス及びデータラッチのセットアップホールド時間に一致するように設計する。

荷の変動に対する自律緩衝能力（エラスティック能力）といった、高集積回路実現に適した特性がある [2] .

2.2 STP 中のパケットの速度

前述のハンドシェイクを逐一模擬すれば、STP システム全体の挙動や性能を把握できる一方で、多大なシミュレーション時間が必要になる。これに対して本論文では、STP 内で転送されるパケットの速度に着目して、より簡易に STP システムの挙動を把握する方法を提案する。以下では、まず、本論文における用語の意味を定義する。

[定義 1] (パケットの速度 $V(t)$) $V(t)$ [ステージ/秒] は、あるパケットが単位時間当りに転送されるステージ数である。

$V(t)$ は STP 内で転送されているパケットの疎密状況によって変動する。これは、各パケットがハンドシェイク制御により転送されるためである。この疎密状況を定量的に議論するために、以下では、パケット間の距離 $D(t)$ を定義した後、 $V(t)$ と $D(t)$ の相関を説明する。

[定義 2] (パケット間距離 $D(t)$) $D(t)$ [秒] は、ある時刻 t において、隣接する二つのパケット間の時間的な距離、すなわち後続パケットと先行パケットの間にある回路のクリティカルパス上の信号伝搬時間である。

あるステージに着目すると、パケットが当該ステージを通過する最小時間であるハンドシェイク時間は、物理的な回路構成要素に基づき次のように定義される。

[定義 3] (ハンドシェイク時間 $(T_f + T_r)$) $(T_f + T_r)$ [秒] は、ステージごとの 1 回のハンドシェイクにかかる最小時間である。ここで、 T_f [秒] と T_r [秒] は、それぞれ send 信号と ack 信号の伝搬時間である。

以上の定義に基づき、 $D(t)$ と $V(t)$ の関係を次に説明する。まず、簡単化のため、全ステージの $(T_f + T_r)$ は同一とする。

$D(t) \geq (T_f + T_r)$ となる場合、図 2 に示すように、send 信号の到着に先立ち、ack 信号が到着し、 T_r は無視できる。このため、図 3 に示すように、 $D(t) \geq (T_f + T_r)$ となる範囲内であれば (図 3(a)), 各パケットは $V(t) = \frac{1}{T_f}$ で前進する (図 3(b))。一方、ハンドシェイクが延期され、パケットが一時的に停止する場合、後続パケットは、send 信号の到達後も、ack 信号の到着を待つ (図 3(c))。これを衝突と呼ぶ。この場合、 $D(t) < (T_f + T_r)$ となり、 $V(t) < \frac{1}{T_f}$ となる。

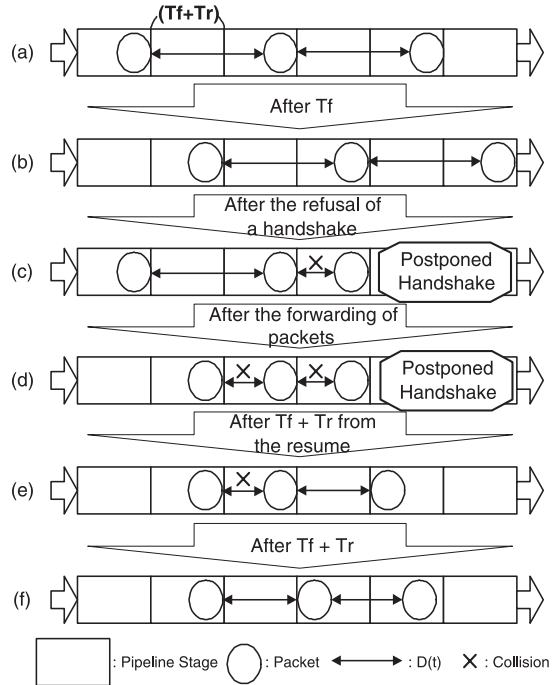


図 3 STP のパケット転送
Fig.3 Packet transfer scheme of STP.

STP では、このような一時的な衝突は、自律的に緩衝される。まず、先行パケットが衝突した場合でも、後続パケットは可能な限り前進できる (図 3(c)~(d))。また、STP では、ハンドシェイクが完了すると同時にパケットは前進を再開する。したがって、1 ステージの空きがあれば処理が継続される。このように、STP には、パケット密度の増加に伴う衝突により、たとえ一部のパケット群で一時的に $D(t) < (T_f + T_r)$ となっても、 $D(t) \geq (T_f + T_r)$ を維持する、エラスティック能力がある (図 3(e)~(f))。したがって、一時的な衝突が $V(t)$ に与える影響も緩和される。

2.3 マクロな動作特性

実際には、各ステージの $(T_f + T_r)$ は異なる。例えば、複数の STP との合流点における調停により、ハンドシェイクが延期される場合、これは直前のステージへの T_r が伸びたとみなせる。また、設計ツールや製造環境に依存する回路実現工程において、設計段階で定めた平均遅延を厳密に保つのは困難であるため、 $(T_f + T_r)$ にばらつきが生じる。これらのことから、実用的な STP の模擬には、ステージごとに T_f または T_r の異なる STP をモデル化する必要がある。

このような STP では、 $(T_f + T_r)$ の最大値を T_{max} とおくと、パケットは自律的に $D(t) \geq T_{max}$ を維持しようとする。パケット数の増加により $D(t) < T_{max}$ となると、常に衝突が発生し、待ち時間を後続パケット群で緩衝しようとする。後続する個々のパケットは $D(t) - T_{max}$ だけ時間的な余裕をもっており、この余裕で待ち時間を吸収する。このため、STP へ流入するパケットに対する衝突の影響は隣接パケット間の $D(t)$ の総和、つまり、一連の後続パケット群の両端のパケット間の $D(t)$ で決まる。よって、衝突の影響は、個々のパケットの分布によらず、後続パケット数によって決まる。したがって、 $V(t)$ はパケット数を基準に算出でき、パケット群の疎密状況を追跡せずとも、STP の挙動が模擬できる。

3. マクロフローモデル

本章では、STP システムの基本構成である、リング型 STP の、マクロフローモデルを提案する。本モデルは、不連続に変化するパケットの速度 $V(t)$ を、ハンドシェイク時間 $(T_f + T_r)$ とパケット間距離 $D(t)$ の関係から、代数的に表現する。

3.1 リング型 STP

直線状の STP を環状に結合したリング型 STP は、プログラムの実行を行う環状データパス [2] や、演算アルゴリズム中の繰返し構造を STP に写像した ALU 回路 [14] 等への応用が可能のため、大規模 STP システムを組織的に構成する上で不可欠な基本構成の一つである。

リング型 STP は、図 4 に示すように、直線状の STP の出口を入口に接続して、環状のパイプライン内でパケットを転送する構成である。一般的には、二つのパケット流を調停し合流させるマージステージ (M)、及び一つのパケット流を選択し分流するブラン

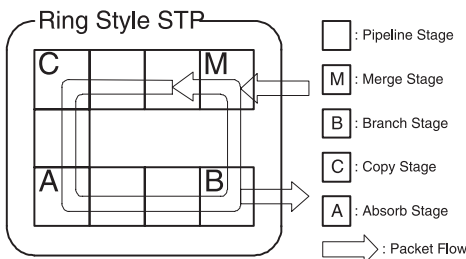


図 4 リング型 STP のモデル
Fig. 4 Model of ring style STP.

チステージ (B) を配し、更にパケットの消去/複製を行うステージ (A/C) を内蔵する。この構成では、パイプライン容量が許す限りパイプライン処理を時間的に重畳できるため、高いパイプライン並列処理性能が期待できる。

このようなリング型 STP (以降、単に STP) で構成するシステム的设计では、定常的に高いパイプライン利用率を維持するように、ハードウェア構成、すなわち M・B・A・C の配置やステージ数等と、応用プログラムを設計すれば、STP の処理能力をできる限り活用できる。こうした設計に基づくシステムでは、システム内に存在するパケットの総数 (以降、 P_{total}) の時間的な変動は緩やかであるため、あるパケットが STP 内を 1 周する間は、巨視的に、 P_{total} は単調に増減するとみなせる。

前章に述べたように、直線状の STP では、 $D(t)$ と $(T_f + T_r)$ の大小関係によって、衝突が生じない状態 (I) と生じる状態 (II) がある。一方、リング型 STP では、衝突が生じる状態には、更に、その影響が STP のエラスティック性により解消される状態 (II-1) と、解消されきれずにその影響が 1 周回する状態 (II-2) がある。これらの STP の状態は、 P_{total} に応じて遷移する。本論文では、それぞれを状態 I、II-1、及び II-2 と呼ぶ。次節以降では、 $D(t)$ と $(T_f + T_r)$ を用いて、各状態における $V(t)$ を求めるとともに、各状態における P_{total} の上・下限を明らかにする。以降、STP のステージ数、ステージ i における $(T_f + T_r)$ 、全ステージの T_f の合計、及び全ステージの T_r の合計を、それぞれ、 pl 、 $(T_{fi} + T_{ri})$ 、 $\sum T_f$ 、及び $\sum T_r$ とおく。

3.2 状態 I

全ステージにおいて $D(t) \geq (T_{fi} + T_{ri})$ 、すなわち衝突がない状態を状態 I と呼ぶ。この状態では、STP のエラスティック能力により、 $(T_{fi} + T_{ri}) = T_{max}(T_f + T_r)$ の最大値) であるステージ i (以降、最大ステージ) を通過したパケット群では $D(t)$ は T_{max} 以上に保たれる。このとき、各ステージでのパケットの転送時間において T_r は無視でき、パケットが 1 周する時間は $\sum T_f$ である。したがって、 $V(t)$ は定数、

$$\frac{pl}{\sum T_f} \tag{1}$$

となる。

全ステージのハンドシェイク時間から $D(t) =$

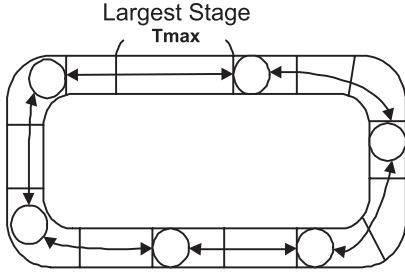


図5 パケットの模式図 (状態 I)
Fig.5 Temporal distance of packets (State I).

$\frac{\sum T_f + \sum T_r}{P_{total}}$ であるが、状態 I では衝突が発生せず、 $\sum T_r = 0$ である。このため、図 5 に示すように、 P_{total} は、 $D(t) \geq T_{max}$ となるよう、

$$\sum T_f \geq T_{max} \times P_{total} \quad (2)$$

を満たす必要がある。すなわち、状態 I における P_{total} の上・下限は、

$$0 \leq P_{total} \leq \frac{\sum T_f}{T_{max}} \quad (3)$$

である。

3.3 状態 II-1

P_{total} の増加により、式 (3) が満たされないうとき、最大ステージの通過に際し、最初のパケット衝突が発生する。このとき、STP のエラスティック能力により、最大ステージを通過するパケットは、 T_{max} 以上に隔てられるため、パケットは最大ステージの直前で、最大 T_{max} の間待つことになる。つまり、パケットが周回する時間は、式 (3) の右辺を超過するパケット一つにつき、 T_{max} 増加する。この状態を状態 II-1 と呼ぶ。したがって、状態 II-1 における $V(t)$ は、

$$\frac{pl}{\sum T_f + T_{max} \times P_{over}} \quad (4)$$

となる。ここで、 P_{over} は、式 (3) の上限を超過したパケット数である。

状態 II-1 では、図 6 に示すように、衝突は、最大ステージを起点に後方ステージに伝搬する。この範囲は、各ステージがもつ時間的な余裕で決まる。最大ステージを S_1 、後方のステージを順に S_2, S_3, \dots, S_{pl} とおくと、 S_i がもつ余裕時間は、 $T_{max} - (T_{fi} + T_{ri})$ となる。よって、衝突に影響されるステージ数 n は、

$$P_{over} \times T_{max} \leq \sum_{i=1}^n \{T_{max} - (T_{fi} + T_{ri})\} \quad (5)$$

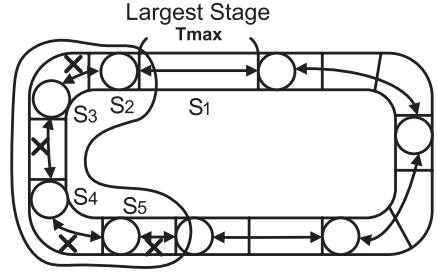


図6 パケットの模式図 (状態 II-1)
Fig.6 Temporal distance of packets (State II-1).

を満たす最小数となる。

状態 II-1 は、式 (5) より、 P_{over} が

$$P_{over} \leq \frac{\sum_{i=1}^n \{T_{max} - (T_{fi} + T_{ri})\}}{T_{max}} \quad (6)$$

を満たす状態である。左辺は、式 (3) より、

$$P_{over} = P_{total} - \left\lfloor \frac{\sum T_f}{T_{max}} \right\rfloor \quad (7)$$

となる。右辺は衝突が 1 周する $n = pl$ のとき最大で、

$$\begin{aligned} & \frac{\sum_{i=1}^{pl} \{T_{max} - (T_{fi} + T_{ri})\}}{T_{max}} \\ &= \frac{\sum_{i=1}^{pl} T_{max}}{T_{max}} - \frac{\sum_{i=1}^{pl} T_{fi} + \sum_{i=1}^{pl} T_{ri}}{T_{max}} \\ &= pl - \frac{\sum T_f + \sum T_r}{T_{max}} \end{aligned} \quad (8)$$

となる。式 (6) に式 (7) と式 (8) を代入すると、

$$\begin{aligned} P_{total} - \left\lfloor \frac{\sum T_f}{T_{max}} \right\rfloor &\leq pl - \frac{\sum T_f + \sum T_r}{T_{max}} \\ \left\lfloor \frac{\sum T_r}{T_{max}} \right\rfloor &\leq pl - P_{total} \end{aligned} \quad (9)$$

となる。すなわち、状態 II-1 における P_{total} の上限は、

$$P_{total} \leq pl - \left\lfloor \frac{\sum T_r}{T_{max}} \right\rfloor \quad (10)$$

である。

3.4 状態 II-2

更なる P_{total} の増加により、式 (10) が満たされないうとき、図 7 に示すように、各ステージに衝突を緩衝する余裕がないので、衝突の伝搬は 1 周回し、最大ステージでの新たな衝突にまで波及する。すなわち、いったん発生した衝突は STP を周回し続け、すべて

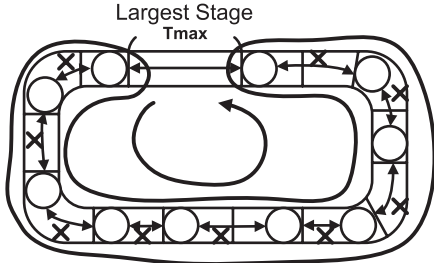


図7 パケットの模式図(状態 II-2)
Fig. 7 Temporal distance of packets (State II-2).

の packets がステージを前進するたびに衝突する。この状態を状態 II-2 と呼ぶ。

状態 II-2 では、全ステージで衝突が発生しており、パケットはパイラインバブルとすれ違わない限り前進できない。STP 中のバブルの総数を $B_{total} (= pl - P_{total})$ とすると、パケットは、バブルの一団と 1 回すれ違うことで、 B_{total} [ステージ] だけ前進できるから、パケットが STP を 1 周するには、 $\frac{pl}{B_{total}}$ 回だけバブルの一団とすれ違うことになる。また、1 回すれ違うために、バブルの一団は、 $pl - B_{total}$ [ステージ] だけ前進する必要がある。これらのことから、パケットが STP を 1 周する時間は、バブルの一団が STP を $\frac{pl}{B_{total}} \times (pl - B_{total}) \div pl = \frac{pl - B_{total}}{B_{total}}$ 回だけ周回する時間で規定できる。ここで、バブルが STP を 1 周する時間は、 $\sum T_r$ である。

以上より、状態 II-2 における $V(t)$ は、

$$pl \div \frac{(pl - B_{total}) \times \sum T_r}{B_{total}} = \frac{pl}{\sum T_r} \times \frac{pl - P_{total}}{P_{total}} \quad (11)$$

となり、 P_{total} に関して定義できる。

状態 II-2 における P_{total} は、 pl 未満であるため、上限は、

$$P_{total} < pl \quad (12)$$

である。

以上より、リング型 STP の性能が、 P_{total} に関する三つの状態において、 $V(t)$ を用いて統一的に扱えるマクロフローモデルが定義できる。

3.5 実システムのシミュレーション

マクロフローモデルでは、単調に増減する P_{total} に関して $V(t)$ を定めた。一方、実システムでは、M・B・A・C の動作により、 P_{total} が増減する。したがっ

て、シミュレーションでは、 P_{total} が変化した時点で、STP 中のパケットの $V(t)$ を再設定する。すなわち、 P_{total} に変化がある時刻にイベントを発生させ、イベント内で、 P_{total} に応じた $V(t)$ に基づきパケットを移動させて、 P_{total} が変化する時刻に新たにイベントを発生させる、イベント駆動シミュレーションを採用する。これにより、動的に負荷の変動する実システムの振舞いを、パケットの転送タイミングを逐一追跡する従来の手法に比べて、高速に模擬できる。

また、実システムにおいて、合流 M では、M ステージへの send 信号の到着が遅い方のパケットの転送が、最大で 1 パケット分の転送時間だけ遅れる。更に、複製 C では、ラッチしたパケットを前方ステージに連続して 2 回転送することで、1 パケット分の転送時間の遅延のみで複製が完了する。これらの遅延の影響は、後続パケットを遅延させ、M/C ステージから離れるほど小さくなる。これに対して、M/C をもつ構成のシミュレーションにおいては、常に P_{total} を $P_{total} + 1$ とみなして、1 パケット分の転送時間だけ全パケットを遅延させる。これにより、付加的な処理なく、M/C の影響を最大に見た安全側の見積りが期待できる。

3.6 パラメータ設定法

マクロフローモデルに基づくシミュレーションでは、 pl と各状態における $V(t)$ 及び P_{total} の上限値が必要である。ここで、状態 I における P_{total} の上限値及び状態 II-1 における P_{total} の上限値を、それぞれ、 P_{IU} 及び P_{II-1U} と呼ぶ。これらのパラメータは、システムの設計時に見積もられた、 pl と、 $(T_{fi} + T_{ri})$ から算出する。つまり、システムの機能をパイライン分割する際に見積もられる論理ゲートや配線の遅延に基づき、 T_{fi} と T_{ri} を決定し、パラメータ値を求める。

マクロフローモデルは、 P_{IU} 、 P_{II-1U} 、及び $V(t)$ は、 pl 、 $\sum T_f$ 、 $\sum T_r$ 、及び T_{max} から導出できることを示している。すなわち、 $(T_{fi} + T_{ri})$ の組合せに対して、すべての組合せではなく、 pl 、 $\sum T_f$ 、 $\sum T_r$ 、及び T_{max} が変化する場合のみパラメータを設定して、シミュレーションを行えば十分である。よって、各ステージごとにランダムなばらつきが想定される場合でも、効率的に性能見積りが行える。

P_{IU} と P_{II-1U} は、それぞれ式 (3) と式 (10) より、

$$P_{IU} = \left\lfloor \frac{\sum T_f}{T_{max}} \right\rfloor, P_{II-1U} = pl - \left\lfloor \frac{\sum T_r}{T_{max}} \right\rfloor \quad (13)$$

であり、設計値により定まる。また、 $V(t)$ は、式 (1)、

式 (4), 及び式 (11) より, 設計値により定まる.

4. 評価

自己タイミング型パイプライン機構 STP で実装された実プロセッサ (DDMP) チップを対象に, 前章で導出したマクロフローモデルに基づくシミュレーションを行い, シミュレーションに要した時間及び精度を, パケットの転送タイミングを逐一追跡する従来の手法と比較し評価した.

4.1 評価環境

マクロフローモデルに基づくシミュレータ, 及びパケットの転送タイミングを逐一追跡する手法^(注2)に基づくシミュレータ (従来シミュレータ) は, Java を用いて実装した. これらのシミュレータは, 応用プログラム, 入力パケット, 及び pl 等のプロセッサ要素単位の情報を入力として, タイムスタンプを付与した処理結果パケットを出力する. そしてイベントごとに更新されるパケットの位置に基づいて, プログラムに従った演算を適用する構成となっている. 提案モデルによる計算コスト削減の効果を示すため, モデルに依存する部分以外は同一の実装とした.

また, LSI の製造工程における, $(T_{fi} + T_{ri})$ のばらつきも反映するため, 実際の DDMP チップから, 3.6 に述べたパラメータを採取した. DDMP は, 複数のリング型 STP プロセッサをパケットルータで接続した CMP (chip multi-processor) 構成をとっている [2]. 図 8 に機能ブロックを示す. 各プロセッサ要素 (PE) は, データ駆動型処理方式を採用しており, M・B・A・C に加え, パケットの待合せ, 発火, 演算, 及び次命令フェッチの各機能を STP 内の各処理回路に分割して実装されている. また, 10 個の PE があり, PE 間は, M と B で構成する多段パケットルータで結合される. PE はステージ数と命令セットにより三つ

に分類され, ここでは, それぞれ $PE : x$, $PE : y$, 及び $PE : z$ とする. 各 PE は, 内部メモリアクセス命令等の共通の命令以外に, $PE : x$ は外部メモリアクセス, $PE : y$ はタグ処理及び算術演算, 更に $PE : z$ は論理演算のための命令セットをもつ.

採取した各 PE のパラメータを表 1 に示す. 表中, pl はステージ数, また, P_{IU} 及び P_{II-1U} は, それぞれ, 状態 I 及び状態 II-1 における P_{total} の上限値である. また, $V(t)$ は, 最大値を $1 \left(\sum \frac{pl}{T_f} = 1 \right)$ として, 式 (13) をもとに, 正規化した値を算出した. これらは, システム設計時に見積ったパラメータと同等のものである.

これらの環境に基づき, 次節以降で, 提案モデルに基づくシミュレータ及び従来シミュレータの見積り値と, 実プロセッサの実測値を議論する.

4.2 負荷変動がない場合

まず, 処理負荷, すなわち, P_{total} が一定になるデータ駆動プログラムを用いて比較評価を行った. 実プロセッサの実測値 (Real) を図 9 に示す. グラフの横軸は STP の負荷すなわち $\frac{P_{total}}{pl}$ であり, 縦軸は $V(t)$ である. $PE : y$ と $PE : z$ の結果はほぼ一致しており, グラフ上の差異はないため, 図には $PE : x$ と $PE : y$ の結果のみを掲載する. 結果から, 段数や, 各状態におけるパケット数の上限値が異なる構成であっても, マクロフローモデルに基づく見積り (Macro-Sim.) は実測値とほぼ一致している.

ここで, 実プロセッサの実測値において, 状態 I であってもパケットの速度は微かに変動し, 数%の揺らぎがある. これは, 設計段階で定めた $(T_{fi} + T_{ri})$ がばらついたことによる, パケット群の偏りが主因であると考えられる. このようなばらつきは, LSI チップの製造ばらつきやチップ温度・電圧変動等に起因しており, 現行の LSI 製造・実装技術では不可避である. ところが, ばらつきによる偏りの性能見積りへの反映は, 半導体素子レベルの模擬を必要とし, 設計中には困難である. したがって, 性能見積りでは, この揺

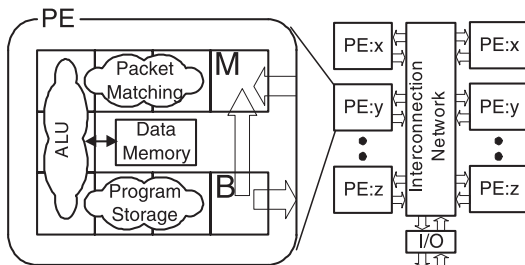


図 8 DDMP: リング型 STP システム
Fig. 8 DDMP: A ring-style STP system.

表 1 DDMP のパラメータ
Table 1 Parameters for DDMP.

	$PE : x$	$PE : y$	$PE : z$
pl	63	43	43
P_{IU}	34	31	30
P_{II-1U}	58	39	39

(注2): 図 2 に示した信号遷移を離散イベントとして逐一評価する手法.

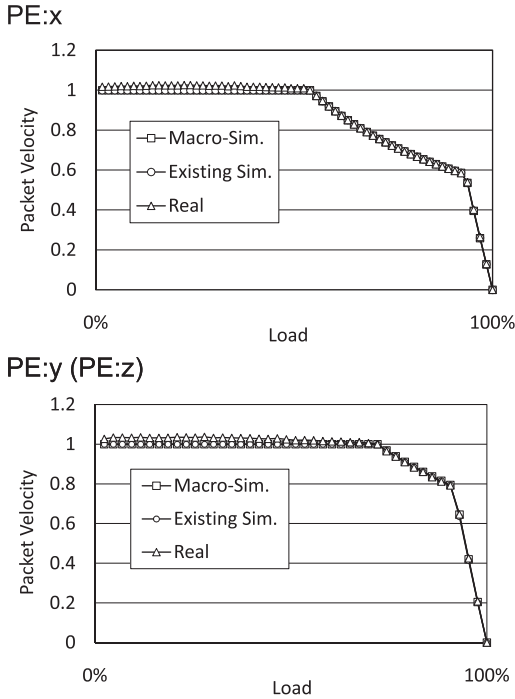


図 9 負荷一定の場合の比較
Fig.9 Performance estimation for steady load.

らぎに対する、設計余裕を見込む必要がある。結果のグラフでは、 $P_{total} = P_{IU}$ におけるパケットの速度を1として正規化している。この結果、 $PE : x$ と $PE : y(PE : z)$ では、それぞれ約3%と約4%の設計余裕が必要と分かった。

以上のことから、マクロフローモデルによる見積りと従来シミュレータによる見積り (Existing Sim.) の誤差は、この設計余裕内に収まっていれば許容できる。実際、見積りの誤差は、いずれのPEでもすべて設計余裕内に収まっており、マクロフローモデルは、従来シミュレータと同等の十分な精度をもつといえる。

一方、マクロフローモデルに基づくイベント駆動シミュレーションは、従来シミュレータと比較して、計算コストを大幅に削減できる。パケットの転送タイミングを逐一追跡するには、1パケットの移動(ハンドシェイク)につき、send信号の遷移とack信号の遷移の2イベントが発生する。このため、総イベント数はパケット数 P_{total} に比例する。一方、マクロフローモデルでは、全パケットの移動につき、唯一 $V(t)$ を求めるだけであり、イベント数はパケット数に依存しない。そのため、性能見積りに要する時間を50%以上

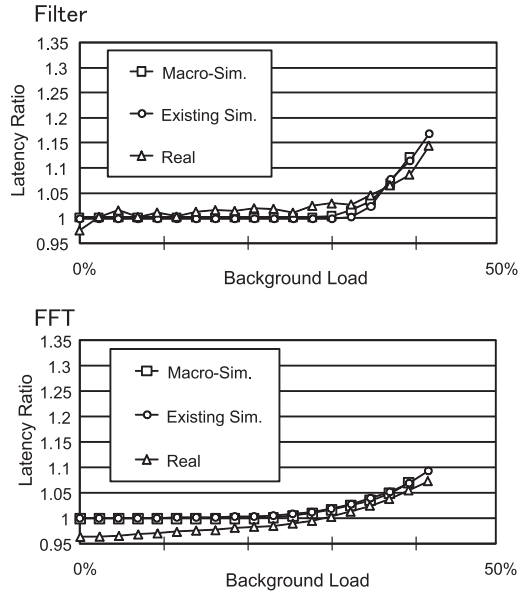


図 10 応用プログラムの性能見積り
Fig.10 Performance estimation for applications.

短縮でき、理想的には $\frac{1}{P_{total}}$ まで短縮できる。

4.3 負荷変動がある場合

実際の応用プログラムでは、実行時に動的に処理負荷、すなわち、 P_{total} が変動する。DDMPが採用するデータ駆動処理方式では、パケットが処理コンテキストを内包するため、明示的なコンテキストスイッチが不要である。このため、各プログラムの負荷の変動を抑えれば、複数のプログラムを並列実行でき、高スループット化が期待できる。この場合、応用システムの開発においては、各プログラムが要求される入力データレートで実行可能かどうかを第一の性能評価指標になる。本評価では、応用プログラムの入力データレートを固定して実行し、これと同時に並行に、定常的な負荷 (Background Load) を発生するプログラムを実行した。追加する負荷の量を変えることによって、状態I、II-1、及びII-2で応用プログラムが動作する状態を設定でき、三つの状態を扱えるマクロフローモデルの有効性が判断できる。

また、応用プログラムとしては、 P_{total} が変動する場合における提案モデルの見積り精度を評価するために、 P_{total} の時間的な増減が比較的大きいラプシアン近傍フィルタ (Filter) と、比較的小さい高速フーリエ変換 (FFT) を選択した。図10に、結果を示す。図中横軸は負荷であり、縦軸は最良値を1とした処理

表 2 シミュレーション時間の比較
Table 2 Comparison of simulation time.

	Simulation Time [s]			
	Filter		FFT	
Background Load	0%	30%	0%	30%
Macro-Simulation	19.40	38.04	24.38	62.80
Existing Simulation	62.12	121.33	104.58	205.80
Speed-Up Ratio	3.20	3.19	4.29	3.28

レイテンシである。両プログラムとも、マクロフローモデルに基づく見積りは、従来シミュレータの見積りとの誤差が約 3% の設計余裕内に収まっており、従来シミュレータと同等の精度の見積りが可能である。また、実測値と比較して、オーバフロー点を安全側に見積もっていることから、最終的な応用プログラムの動作も保証できる。マクロフローモデルに基づく見積りと実測値との誤差は、約 $\pm 4\%$ であった。

両アプリケーションのシミュレーションに要した時間を表 2 に示す。マクロフローモデルは、3 倍以上のシミュレーションの高速化を達成している。

本評価に用いた応用プログラムは、時間的な負荷の増減を抑えるようにプログラムを最適化済である。これは、最適化によって、定常的に高いパイプライン利用率を維持できて高性能化できるためである。実用的な応用プログラムは通常、このような最適化を施されるため、 P_{total} の時間的な増減は Filter プログラムと比較して同程度以下であるといえる。

以上より、実システムであっても、 P_{total} の時間的な変動が極端ではない、すなわち Filter プログラムと同程度以下であれば、マクロフローモデルは、LSI 試作前の事前評価に十分な精度をもつ高速なシミュレーションを実現可能である。

5. む す び

本論文では、自己タイミング型パイプライン機構 STP をパケットの速度でモデル化する、マクロフローモデルを提案した。提案モデルでは、パイプライン内で転送されるパケットの時間的な間隔に着目し、STP 中のパケット数に応じたパケットの速度の変化を代数的に定量化することで、3 段階に不連続に変化する実システムの実効性能の見積りを可能とした。実際に、提案モデルに基づくシミュレータを実装して、現行の LSI 製造技術に不可欠な設計余裕内でシミュレーションが可能であることを示した。また、性能見積りでは、パケットの速度のみで、性能変化を統一的に扱うこと

で、STP のパケット転送の網羅的な模擬が不要になり、3 倍以上のシミュレーション時間の短縮が可能になった。

今後は、提案モデルをもとに、プロセッサ間相互通信網等を含む STP チップ全体を対象にしたモデルに関して検討する予定である。

謝辞 本研究は、科学技術振興機構「JST」の戦略的創造研究推進事業「CREST」における研究領域「情報システムの超低消費電力化を目指した技術革新と統合化技術」の研究課題「超低消費電力化データ駆動ネットワークシステム」による。

文 献

- [1] 寺田浩詔, “超高速・低電力・高性能な新システム・アーキテクチャの考え方—刷り込みからの脱却,” 信学技報, ICD 2003-189, Dec. 2003.
- [2] 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.
- [3] I.E. Sutherland, “Micropipelines,” Commun. ACM, vol.32, no.6, pp.720–738, June 1989.
- [4] A. Lines, “Nexus: An asynchronous crossbar interconnect for synchronous system-on-chip designs,” Proc. 11th Annual Hot Interconnects Conf., pp.2–10, Stanford, U.S.A., Aug. 2003.
- [5] A. Takamura, M. Kuwako, M. Imai, T. Fujii, M. Ozawa, I. Fukasaku, Y. Ueno, and T. Nanya, “TITAC-2: A 32-bit asynchronous microprocessor based on scalable-delay-insensitive model,” Proc. 1997 Intl. Conf. on Computer Design, pp.288–294, Austin, U.S.A., Oct. 1997.
- [6] T. Villiger, H. Kaslin, F.K. Gurkaynak, S. Oetiker, and W. Fichtner, “Self-timed ring for globally-asynchronous locally-synchronous systems,” Proc. 9th Intl. Symp. on Asynchronous Circuits and Systems, pp.141–150, Vancouver, Canada, May 2003.
- [7] D. Morikawa, M. Iwata, and H. Terada, “Super-pipelined implementation of IP packet classification,” J. Intelligent Automation and Soft Computing, vol.10, no.2, pp.175–184, Aug. 2004.
- [8] 鹿毛哲郎, “VLSI 回路シミュレーション,” 信学誌, vol.83, no.11, pp.838–842, Nov. 2000.
- [9] A. Xie and P.A. Beerel, “Accelerating markovian analysis of asynchronous systems using string-based state compression,” Proc. 4th Intl. Symp. on Advanced Research in Asynchronous Circuits and Systems, pp.247–260, San Diego, U.S.A., March 1998.
- [10] S. Chakraborty and R. Angrish, “Probabilistic timing analysis of asynchronous systems with moments of delays,” Proc. 8th Intl. Symp. on Advanced Research in Asynchronous Circuits and Systems, pp.99–108, Manchester, U.K., April 2002.

- [11] J.C. Ebergen, S. Fairbanks, and I.E. Sutherland, "Predicting performance of micropipelines using charlie diagrams," Proc. 4th Intl. Symp. on Advanced Research in Asynchronous Circuits and Systems, pp.238-246, March 1998.
- [12] W.M. Zuberek, "Event-driven simulation of timed petri net models," Proc. 33rd Annual Simulation Symp., pp.91-98, Washington, U.S.A., April 2000.
- [13] C.J. Myers, Asynchronous Circuit Design, Jhon Wiley & Sons, 2001.
- [14] T.E. Williams and M.A. Horowitz, "A zero-overhead self-timed 160-ns 54-b CMOS divider," IEEE J. Solid-State Circuits, vol.26, no.11, pp.1651-1661, Nov. 1991.

(平成 20 年 8 月 25 日受付, 21 年 1 月 15 日再受付)



岩田 誠 (正員)

1986 阪大・工・電子卒. 1991 同大大学院博士課程単位取得後退学. 同年大阪大学工学部助手, 1997 高知工科大学情報システム工学科助教授, 2002 同教授, 2006 同学科長, 現在に至る. 2002 より, 東北大・電気通信研究所 IT21 センター客員助教授, 2006 同客員教授を兼務. 2008 カリフォルニア大学アーバイン校客員研究員. 博士(工学). データ駆動パラダイムを核とした, ソフトウェア環境及び ULSI 向きアーキテクチャの研究に従事. 情報処理学会, IEEE 各会員.



三宮 秀次 (正員)

2002 高知工科大・工・情報システム卒. 2004 同大大学院修士課程了. 2006 高知工科大学助手. 2007 同大学院博士課程単位取得後退学, 現在に至る. 自己タイミング型パイプラインシステムとその開発手法に関する研究に従事. 情報処理学会, IEEE

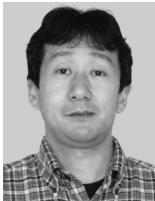
各会員.



大森 洋一 (正員)

1993 京大・工・情報卒. 1998 奈良先端科学技術大学院大学博士課程了. 同年高知工科大学工学部助手. 2004 九州大学助手. 2007 同助教, 現在に至る. 博士(工学). 高次システム設計及び組み込みシステムソフトウェアに関する研究に従事. 情報処理学

会, IEEE 各会員.



酒居 敬一 (正員)

1993 広島大・工卒. 1995 同大大学院工学研究科博士課程前期了. 同年広島大学工学部助手. 2003 高知工科大学工学部講師, 現在に至る. 博士(工学). 計算機アーキテクチャ, コード最適化に興味をもつ.