

自己同期型パイプラインシステムのマクロフローモデル

三 宮 秀 次[†] 大 森 洋 一[†] 岩 田 誠[†]

自己同期型パイプライン (STP) システムの設計では、自律的に移動するパケットによる、動的な負荷状況を把握するために、パイプラインの充足率と利用効率に着目したマクロフローモデルが有効である。

本稿では、実際的なステージ長にばらつきのある STP をマクロにモデル化し、仮想的なパケット間の距離に着目することで、パケットの転送時間が最大となるステージにより全体の性能が決まることを示す。得られたモデルを用いてシミュレータを実装し、プロセッサと比較したところ、高速かつ簡単にシステム性能を概観できることが判った。また、チップマルチプロセッサ構成におけるプロセッサ間干渉について、性能に対する影響についても考察する。

A Macro-Flow Model for Self-Timed Pipeline Systems

SHUJI SANNOMIYA,[†] YOICHI OMORI[†] and MAKOTO IWATA[†]

A macro-flow model focused on the occupation rate and pipeline efficiency is effective to glimpse the dynamic load balance brought by the autonomous behavior of packets in the Self-Timed Pipeline (STP).

In this paper, we propose a new macro-flow model which can deal with the STP with heterogeneous stages. The model utilizing virtual distances among packets says the longest stage rules the entire performance. A simulator implemented over the new model enables a fast and simple settlement of parameters in harmony with the actual STP system. Also, the performance degradation influenced by interferences in a chip multiprocessor is evaluated.

1. はじめに

集積技術の発展に伴い、System-on-Chip や System-in-Package による実装が増えてきている。これらの回路構成技術の 1 つとして、自己同期型パイプライン (Self-Timed Pipeline, STP) がある。隣接するステージ間のハンドシェイクにより、局所同期のみで、パケット転送を制御する STP は高い並列処理性能と極省消費電力を同時に達成可能であり、さまざまな応用が検討されている^{3) 5)}。

STP は、集積化の妨げとなるクロックスキューや大域的同期にまつわる問題から本質的に解放されており、システム設計における部品化が容易であるといった利点が知られている。しかしながら、STP を用いたシステム設計では、パケットは STP 内を自律的に移動し、その負荷状態により性能が大きく変動する。こうしたシステム性能の事前見積りを目的として、確率的なモデル⁶⁾ やマルコフ連鎖を用いた手法⁷⁾ 等、いくつかの性能評価モデルが提案されているが、いずれも実アプリケーションにおける動的な負荷の偏りに応じた見積りは困難であり、実用的な性能評価にはシミュレーションが適している。

従来の挙動モデルでは、STP の動的なパケットの移動を再現するために、ハンドシェイクを忠実に模擬する必要があり、メモリ使用量や実行時間といったシミュレーション

コストが高い。システム設計のほとんどの段階において、性能見積りに求められるのは、個々のパケットの状態遷移ではなく、スルーット等、システム全体の性能を概観できる能力である。

本稿では、パイプラインの充足率とパケット間の距離に着目した仮想的な平均速度を導入することで、パケットの挙動を抽象化した、ハードウェアのシステム性能見積りが可能なマクロフローモデルを提案する。

2 章では、STP の性能評価モデルの要件を整理する。次いで、3 章では、パイプラインの利用効率と充足率に着目し、挙動を抽象化したマクロフローモデルについて説明する。4 章では、ハードウェア実装における特性を考慮した拡張マクロフローモデルの提案を行う。5 章では、マルチプロセッサ構成における物理リンクの干渉のモデル化について述べる。

2. 性能評価モデルの要件

2.1 STP

図 1 に STP の理論的な構成を示す。STP では、各ステージはデータラッチ、処理回路、および通常 C 素子と呼ばれる転送制御回路により構成される。パケットが存在するステージの C 素子は、後続するステージに対し、パケットの転送を要求する send 信号を出し、後続ステージは、ステージが空の場合、転送を許可する ack 信号を返す。もし、後続ステージが空でない場合、パケットはラッチされたまま、ack 信号を待ち、後続ステージが空になったあと、

[†] 高知工科大学
Kochi University of Technology

ack 信号を受取り、ハンドシェイクを完了する。

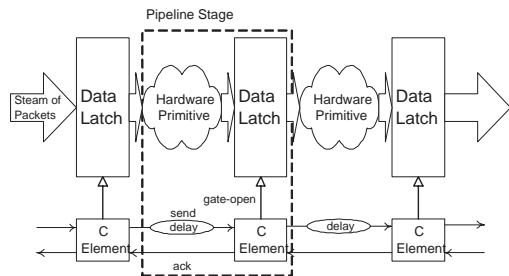


図 1 STP の基本構成

ここで、send 信号の遅延を T_f 、ack 信号の遅延を T_r とすると、パケットの最小出力間隔 [cycle] は、 $(T_f + T_r)$ となり、パケットがステージを占有する時間において、実際にパケットが処理される時間の割合であるパイプライン効率 E は、 $E = \frac{T_f}{T_f + T_r}$ となる。

STP では、通常は全てのパケットが同じ速度で前進するが、図 2 (a) に示すとおり、パイプラインハザードが発生すると、影響を受けるパケットのみが停止し、後続パケットは、ステージが空いている限り前進する。パイプラインハザードが解消され次第、先頭のパケットから前進を開始するが (図 2 (b))、後続のパケットが前進できるようになるためには、 $(T_f + T_r)$ だけ待つ必要がある (図 2 (c))。最終的に、各パケットは、最小でもステージを通過するのに必要な時間 ($T_f + T_r$) だけ隔てられて存在することになる (図 2 (d))。この状態では、パケットは各ステージを T_f で通過する。

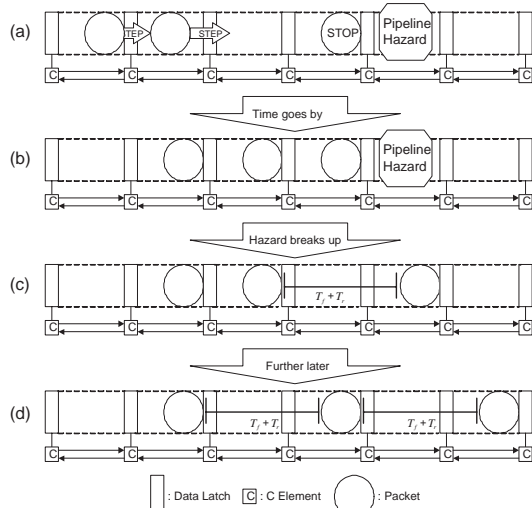


図 2 STP のエラスティック能力

パケットの増加により、パケット間隔が $(T_f + T_r)$ 以下に詰められた場合、パケットの転送には、ack 信号を待つ必要が生じる。これを衝突と呼ぶ。パイプラインのステー

ジ数に対するパケット数の割合であるパイプライン充足率を P_{or} とおくと、 P_{or} が E を超過した場合、パケット間隔の平均が $(T_f + T_r)$ より小さくなり、定常的に衝突が起こる。衝突が発生した場合のパケットの転送時間では、 T_r が支配的になり、パケットは停滞する。

このように、STP では、 $P_{or} > E$ となると極端に性能が低下するので、動的なパケットの挙動把握が不可欠である。

2.2 リング型 STP

自己同期型パイプラインは、データ駆動型メディアプロセッサ DDMP として、商用化されており、消費電力 3W において約 10BOPS (Billion Operations Per Second) を達成し、その有効性が確認されている¹⁾。

STP を利用したデータ駆動処理では、コンテキストスイッチのオーバーヘッドを無視できるため、複数のパイプラインをソフトウェアにより容易に接続可能である。DDMP も含めた STP の実装では、パイプラインを制限なく重畳可能なリング型の STP によりプロセッサエレメント (PE) を構成し、PE がルータを介して均質に接続されたマルチプロセッサ構成が一般的である²⁾。

図 3 に示すとおり、DDMP では、STP の入り口に、パケットを回路的に調停し合流させる Merge Stage とパケットの宛先により選択出力する Branch Stage を配することでリング型 STP を実現している。各 PE は、異なる演算機能を持つことができ、パケットは、処理内容に応じて静的にルーティングされる。従って、動的な負荷分散を考慮した、プログラムの配置が重要となる。

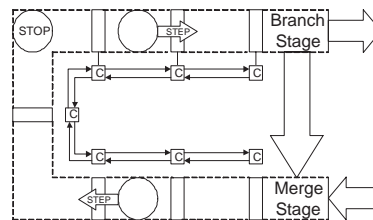


図 3 リング型 STP

以下、本稿では、リング型 STP を対象としたモデル化について検討する。

3. マクロフローモデル

3.1 参照モデル

理論的なシステム性能の計測は、図 3 の構成におけるハンドシェイクを逐一シミュレートすることで実現できる。これをナイーブモデルと呼びシステム設計時の参照性能を与える。

しかしながら、ナイーブモデルでは、各ステージ毎に、パケットの有無、C 素子の状態を管理する必要があり、メモリ空間を圧迫する。加えて、 P_{or} が E に満たない状況では、パケットの移動は予測可能であり、シミュレーションに無駄が生じる。これらを踏まえ、パケットの挙動のみを

シミュレーションするようにモデルの軽量化を試みた。

3.2 理想的なモデル

最初に、ステージ長が均一である場合を検討した。

STP では、パケットが $(T_f + T_r)$ の間隔で配されていた場合、パケットの衝突は、直ちに後続パケットに伝播し、システム全体の性能が低下する。一方、十分にパケットが隔てられている場合、先行パケットの衝突は間隔が $(T_f + T_r)$ になるまで吸収され、STP はあたかもバッファを持つかのように振舞う。

これらの事実に着目し、実行時変数 P_{or} とハードウェアに依存する E により、パケットの停滞を分類し、パケットの平均速度 V_p [stage/cycle] を仮定することで、状態管理の軽減を図った。図 4 に示すとおり、全てのパケットに適用する V_p を一元管理することにより、ナイーブモデルより、大幅なメモリ使用量と計算時間の削減を達成した。

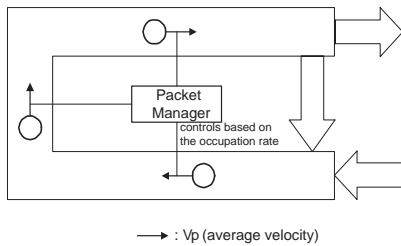


図 4 マクロフローモデル

(a) 通常負荷 ($P_{or} \leq E$) 時

STP のエラスティック能力により、パケットの衝突は巨視的に無視できて、 V_p は T_f に依存し、

$$V_p = \frac{1}{T_f} = (\text{一定}) \quad (1)$$

となる。

(b) 過負荷 ($P_{or} > E$) 時

パケットの衝突は P_{or} が E を超過した割合に比例して起こる。衝突による一回の待ち時間は衝突の回数に比例するので、過負荷時の平均速度を導出すると、負荷の変動がない場合、 P_{or} に関する 2 次式で近似される。

得られたマクロフローモデルを評価するために、マクロフローモデルとナイーブモデルを用いたシミュレータを Java を用いてそれぞれ実装し、シングルプロセッサを対象に比較したところ、ほぼ一致する挙動が観測された。さらに、マルチプロセッサを対象とした実アプリケーションのシミュレーションを行ったところ、ナイーブモデルを用いた場合より約 5 倍の高速化を達成できた⁴⁾。

4. マクロフローモデルの拡張

マクロフローモデルによる性能見積りの適用範囲をさらに拡大させるため、ハードウェアにより実装された STP との比較を行った。その結果が図 5 である。図 5 は、 P_{or} を変化した場合のスループットを最大性能について正規

化したものである。

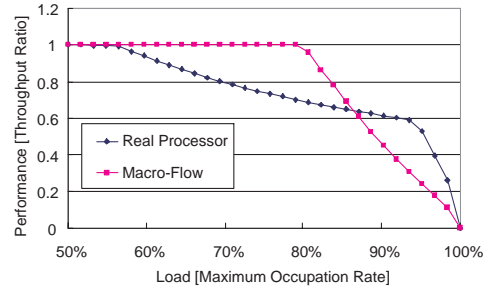


図 5 マクロフローモデルと実機との比較

このように理論的なモデルとの誤差が生じた理由として、配置配線の最適化により、アーキテクチャレベルでは均一に設計したパイプラインステージであっても、不均質となる点が考えられる。この遅延量の調整は極めて困難であり、ハードウェアにおける実効性能の予見には、ステージ長のばらつきを考慮した評価が必要である。実際に、ステージ長の一部を変化させてナイーブモデルを用いたシミュレーションを行ったところ、任意のパターンで図 6 のような概形を示した。

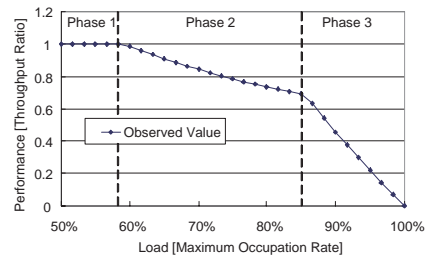


図 6 不均質なステージ長を持つ STP

続いて、Phase 1、Phase 2、および Phase 3 のそれぞれについて、 P_{or} と V_p の関係を導出する。

4.1 Phase 1

全ステージの T_f の総和、全ステージの T_r の総和、最長ステージ長、およびパイプライン段数を、それぞれ、 $\sum T_f$ 、 $\sum T_r$ 、 $\max(T_f + T_r)$ 、 pl とする。

P_{or} が十分に低い場合、最長ステージを通過したパケットの間隔は、 $\max(T_f + T_r)$ 以上を保ち続ける。全てのパケットが $\max(T_f + T_r)$ 以上の間隔で配されている場合、各ステージの通過時間において、 T_r は隠蔽され、1つのパケットが 1 周する時間は、 $\sum T_f$ となる。このことから、 V_p を最大に保つためには、パケット数を P_{total} とした場合、 P_{total} は、

$$\sum T_f \geq \max(T_f + T_r) \times P_{total} \quad (2)$$
を満たす必要がある。図 7 に、パケット数が式 2 を満足するときのパケット間隔を図示する。

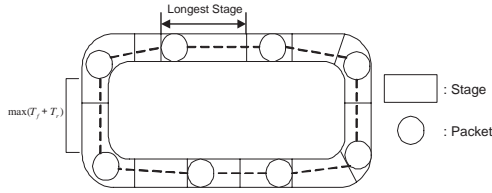


図 7 パケット間隔の模式図 (Phase 1)

Phase 1 では、 V_p は $\sum T_f$ に依存し、

$$V_p = \frac{pl}{\sum T_f} = (\text{一定}) \quad (3)$$

となる。

4.2 Phase 2

P_{total} の増加により式 2 が満たされなくなった場合、最長ステージを通過したパケット群の間隔は $\max(T_f + T_r)$ を維持し続けるが、最長ステージへ到達したパケットは衝突し、影響は最長ステージに先行する連続したステージに波及する。衝突による待ち時間は、1 パケット増加するごとに $\max(T_f + T_r)$ である。

最長ステージを S_0 とし、先行ステージを順に S_1, S_2, \dots, S_{pl} とすると、 S_i における緩衝能力は、 $\max(T_f + T_r) - (T_{fi} + T_{ri})$ である。ただし、 $(T_{fi} + T_{ri})$ は S_i での T_f と T_r の和である。

式 2 を超過するパケット数を P_{over} とおくと、衝突の影響が波及するステージ数 n は、

$$P_{over} \times \max(T_f + T_r) \leq \sum_{i=1}^n \{ \max(T_f + T_r) - (T_{fi} + T_{ri}) \} \quad (4)$$

を満たす最小の n である。図 8 に、衝突が起きたときの模式図を示す。

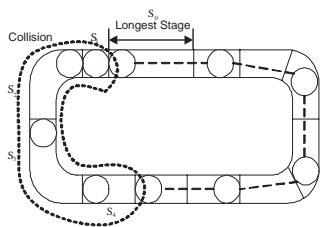


図 8 パケット間隔の模式図 (Phase 2)

この場合、遅延時間は、 $P_{over} \times \max(T_f + T_r)$ となり、 P_{over} に比例する。

ここで、Phase 2 における P_{over} の許容量は、

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

となる。左辺は定義より、

$$P_{over} = P_{total} - \frac{\sum T_f}{\max(T_f + T_r)} \quad (6)$$

であり、右辺は、衝突が 1 周する $n = pl$ の時最大で、

$$\frac{\sum_{i=1}^{pl} \{ \max(T_f + T_r) - (T_{fi} + T_{ri}) \}}{\max(T_f + T_r)} = \frac{\sum_{i=1}^{pl} \max(T_f + T_r)}{\max(T_f + T_r)} - \frac{\sum_{i=1}^{pl} T_{fi} + \sum_{i=1}^{pl} T_{ri}}{\max(T_f + T_r)} \quad (7)$$

$$= pl - \frac{\sum T_f + \sum T_r}{\max(T_f + T_r)} \quad (8)$$

となる。よって、式 5 は、

$$P_{total} - \frac{\sum T_f}{\max(T_f + T_r)} \leq pl - \frac{\sum T_f + \sum T_r}{\max(T_f + T_r)} \quad (9)$$

$$\frac{\sum T_r}{\max(T_f + T_r)} \leq pl - P_{total} \quad (10)$$

とおける。ここで、 $pl - P_{total}$ は、STP 中のバブルの数を表しており、衝突の影響が 1 周しないためには、バブルの数を B_{total} とした場合、

$$\sum T_r < \max(T_f + T_r) \times B_{total} \quad (11)$$

を満たす必要があることがわかった。

Phase 2 における V_p は、 P_{over} に比例するため、

$$V_p = V_c \times C_p \times P_{over} \quad (12)$$

と求められる。 V_c は、Phase 1 における V_p である。ここで、 C_p は、最長ステージ長に依存する定数である。

4.3 Phase 3

パケット数の増加により、式 2 および 11 とともに満足できなくなった場合は、図 9 に示すとおり、衝突の堆積が 1 周してしまい、新たな衝突にまで伝播する。

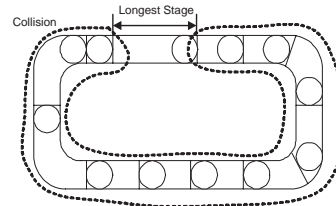


図 9 パケット間隔の模式図 (Phase 3)

この状態では、パケットの衝突は直ちに全ての後続パケットに伝播し、衝突回数はパケット数に比例し、衝突でこの待ち時間は衝突回数に比例するため、式 11 を超過するパケット数を P_{over2} とおくと、パケット速度は、 P_{over2} に関する 2 次式で近似される。

ステージ長が均一な STP は、 $\sum T_f + \sum T_r = \sum T_{fi} + \sum T_{ri}$ であるため、 $P_{over2} - P_{over} = 0$ となり、過負荷状態 $P_{or} > E$ となった場合、直ちに、式 11 を満たさなくなる特別な場合である。

4.4 ハードウェアによる検証

マクロフローモデルを用いて、DDMP ハードウェアを評価し、モデルの妥当性を検証した。ハードウェアのモデル化においては、特別な場合を除いて、各ステージでの T_f 、 T_r 、ステージ数、パイプライン負荷の時間的変動などを正確に計測できないという問題がある。このため、モデルの評価として、定常的な負荷を与える簡単なプログラムを

DDMP の 2 種類の PE である PE と PE で実行し、そこで得られたパラメータによってモデルをスケーリングし、実行中に負荷が変動する実用的なプログラムに対しても適用可能であることを示す。

ハードウェアで計測可能なマクロフローモデルのパラメータは、ステージ数 pl 、Phase 1 と Phase 2 の境界となるパケット数 P_{1-2} 、Phase 2 と Phase 3 の境界となるパケット数 P_{2-3} である。

pl に関しては、リング構造のパイプラインにおいて、全ステージが充足した際にデッドロックが発生することを用いて、パケットのコピー操作によりパイプラインを充足させ、個々の PE のステージ段数を推定した。また、 P_{1-2} および P_{2-3} は、定期的に負荷を与えるプログラムにより性能値をグラフにプロットし、推定した。

シミュレーションに必要なパラメータは、シミュレーションサイクル $(T_f + T_r)$ と T_f である。 $\sum T_f$ は、式 2 を用いて、 P_{1-2} から推定可能である。また、 $\sum T_r$ は、式 11 を用いて、 P_{2-3} から推定可能である。式 2 において、 $\max(T_f + T_r)$ は、式 5 における P_{over} との関係から、推定可能である。PE の場合、 pl は 62 段と推定できて、図 10 から変曲点のパケット数を読み取ることで、 $P_{1-2} = 35$ 、 $P_{2-3} = pl - 5 = 57$ を得る。さらに、式 5 の両辺を pl で割ることで、 $\frac{P_{over}}{pl} = \frac{\max(T_f + T_r) - (T_f + T_r)}{\max(T_f + T_r)}$ となり、 $\frac{22}{62} = \frac{\max(T_f + T_r) - (T_f + T_r)}{\max(T_f + T_r)}$ より、 $(T_f + T_r)$ の平均 40 を得る。式 2 より、 $\sum T_f = P_{1-2} \times \max(T_f + T_r)$ 、 $T_f = \sum T_f / pl = P_{1-2} = 35$ を得る。最後に、 $V_c = \frac{1}{35/40} = \frac{40}{35}$ を得る。

図 10 に、得られたパラメータによるマクロフローモデルによるシミュレーションと、PE、PE において、スループットを測定した結果を示す。これにより、最大約 8% の誤差で、システム性能の概観が可能となった。

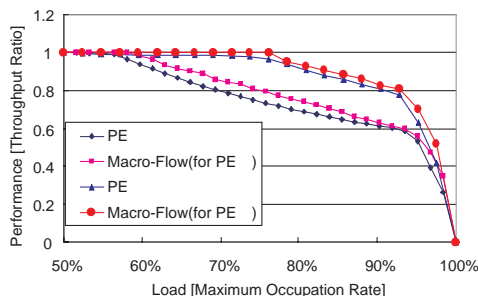


図 10 実機との比較

これまでに得られた個々のパケットに関するモデルから、パイプライン中のパケット数を掛けて、システム全体のスループットを求めると、図 11 のようになる。すなわち、アプリケーションの開発では、システム全体のスループット

性能を最高に保つため、定期的なパイプラインの充足率は Phase 2 になるよう調整すべきであることがわかる。

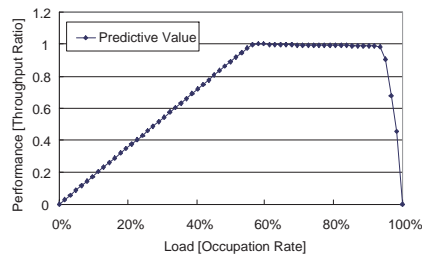


図 11 充足率とシステムスループット

性能見積りの最も困難な Phase 2 と 3 を遷移するようなアプリケーションにおける性能低下の状況を再現するため、充足率が約 80% から 93% まで周期的に変動するプログラムにおいて、マクロフローモデルによるシミュレーションと PE、PE のそれぞれとで、スループットを測定した結果を表 1 に示す。表では、定期的に充足率を約 80% に保った場合を 1 とし、負荷を最大約 93% に変動させた場合のスループットの比を示す。

表 1 負荷変動によるスループットの測定

	DDMP	Macro-Flow	ratio
Throughput Ratio (PE)	0.913	0.884	-3.2%
Throughput Ratio (PE)	0.914	0.920	+0.7%

この結果から、システム性能は概観できるが、STP の出口と最長ステージの位置により、動的な負荷変動パターンが異なることに起因する誤差が残っていると考えられる。

5. マルチプロセッサモデル

STP では、局所同期により、タイミング問題が緩和されており、極めて容易にマルチプロセッサを構成できる。このため、マクロフローモデルでは、リング型 STP を単位にモデル化しており、マルチプロセッサシステムの評価においては、物理リンクの干渉のモデル化が重要である。簡単な STP の接続方法として、Merge Stage と Branch Stage の組み合わせによる方法がある。しかし、この手法では、1 つの STP における充足率が他の STP の充足率に影響し、システム全体の性能が低下し得る。

5.1 STP のネットワーク

DDMP は複数の PE をもつチップマルチプロセッサとなっており、そのネットワーク構成を図 12 に示す。本構成では、1 段目の合流ステージを共有する 1 組の PE では、ルータ部において、2 段目以降のステージが全て共有されることになり、相互のパケット充足率が互いに影響する。

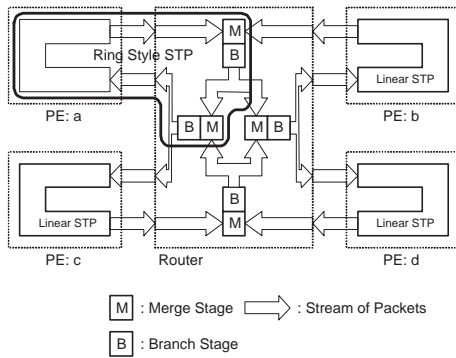


図 12 STP の基本構成

図 13 に、1 段目の合流ステージを共有する PE 1 と PE 2、および PE 1 と PE 2 の組み合わせに対して、パケット充足率を変化させた場合の出力性能を示す。計測された値は、各組み合わせの左側の PE のターンアラウンドタイムを、単体で計測した場合との比として図示している。図中のパケット充足率 (30%、80%、93%) は、それぞれ、Phase1 から 3 における値である。

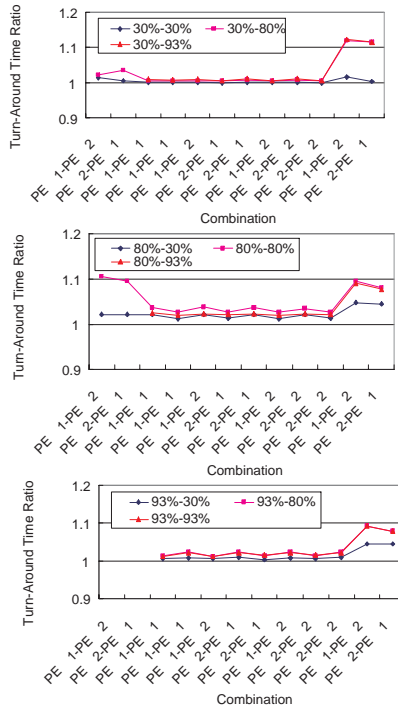


図 13 網干渉の影響

5.2 考察

全ての組み合わせにおいて、単体で出力性能を観測する場合に比べ、性能低下が確認された。1 段目の合流ステー

注：比がプロットされていない組み合わせでは、デッドロックが生じている。これは、共有されたステージの充足が原因と考えられる。

ジを共有しているペアでは、予測どおり、強い相互干渉が観測されたが、他の組み合わせにおいては、干渉の度合いは、一定の値に収まっている。また、負荷に応じて影響は増大するが、同じく一定の値に収まっている。これは、網における衝突を抑制するよう、パケットの流量の変動を吸収できるキューバッファが⁸⁾、随所に内装されていることに起因している。

6. おわりに

本研究では、不均質なステージ長を持つ STP の振る舞いを解析し、パイプライン充足率とパケット間の距離の関係から、パケットの仮想的な平均速度を決定できるよう従来のマクロフローモデルを拡張した。拡張モデルを用いたシミュレータを実装し、実機と比較したところ、性能の概観が可能であることが判った。

本研究を進めるにあたり、DDMP に関して有益なご議論を頂いた村松様をはじめ、Sharp 株式会社の皆様にご感謝致します。

参考文献

- 1) H. Terada, S. Miyata, and M. Iwata, "DDMP's: Self-timed Super-pipelined Data-driven Multimedia Processors," Proceedings of the IEEE, 87(2), pp. 282-296, 1999.
- 2) I. E. Sutherland, "Micropipelines," Communication of ACM, 32(6), pp. 720-738, 1989.
- 3) D. Morikawa, M. Iwata, H. Hayashi, and H. Terada, "Superpipelined IP-Address Lookups on a Data-driven Network Processor," in Proc. of Int. Conf. on Parallel and Distributed Comp. and Sys., pp.431-436, 2001.
- 4) S. Sannomiya, Y. Omori, and M. Iwata, "A Macroscopic Behavior Model for Self-Timed Pipeline Systems," in Proc. of Parallel and Distributed Simulation Conf., pp. 133-140, 2003.
- 5) K. Y. Yun, P. A. Beerel, and J. Arceo, "High-performance Asynchronous Pipeline Circuits," in Proc. of the 1996 Int. Symp. on Advanced Research in Asynchronous Circuits and Systems, pp. 17-28, 1996.
- 6) S. Chakraborty and R. Angrish, "Probabilistic Timing Analysis of Asynchronous Systems with Moments of Delays," in Proc. of Int. Symp. on Advanced Research in Asynchronous Circuits and Systems, pp.99-108, 2002.
- 7) A. Xie and P. A. Beerel, "Accelerating Markovian Analysis of Asynchronous Systems using String-based State Compression," in Proc. of Int. Symp. on Advanced Research in Asynchronous Circuits and Systems, pp. 247-260, 1998.
- 8) 上方輝彦, 岩田誠, 滝根哲哉, 寺田浩詔, "分散キューバッファを持つデータ駆動型プロセッサ Qv-x の性能評価," 電気学会論文誌 C, Vol.116-C, No.11, pp.1295-1300, Nov., 1996.