

## ソフトなパイプラインステージによるサイクル時間の短縮

武田浩一、佐藤正俊、大原輝彦

takeda@okilab.oki.co.jp

沖電気工業株式会社

## 1 はじめに

我々は、第五世代コンピュータ・プロジェクトの一環として並列推論マシン PIM/i[1]の研究開発を行っている。PIM/iの要素プロセッサ[2]は、RISC型命令セットを基本とし、タグの支援を行うとともに、パイプライン処理と級数操作の同時実行により、処理の高速化を図っている。同期式のパイプラインでは全てのステージがクロックで一律に同期をとるため、サイクル時間は各ステージの処理時間のうちの最大のもので決定される。このため、他のステージ内に同期待ちの無駄な時間が生じてしまう。

本稿では、PIM/i要素プロセッサの命令パイプラインにおいて、各ステージの処理時間が異なっても、そのバラつきをステージ間で融通し合って吸収し、サイクル時間を短縮する方法について述べる。

## 2 パイプラインのサイクル時間の短縮

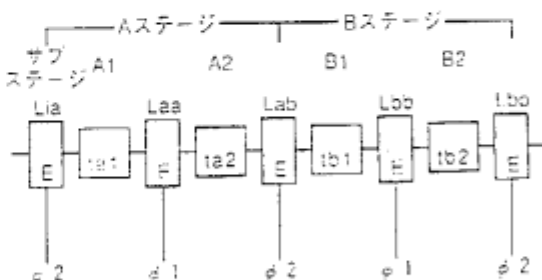
## 2.1 パイプラインの同期待ちオーバーヘッド

パイプライン方式[3]は、一連の実行過程を数つかのステージに分割して、古い工程が完了する前に新しい工程を開始することにより、各ステージをオーバラップして実行させ、高速化を達成する。同期式[4]のパイプラインは、分割された各ステージ間に同相のクロックで動作するフリップフロップが置かれる構造をしている。各ステージはこのクロックにより同期がとられていて、クロックのどちらか一方のエッジで前のステージの処理結果が次のステージに一律に転送される。クロックの周期は、各ステージの処理時間のうちの最大のもので決定される。この場合、各ステージの処理時間が等しいときに最大の性能利得を得ることができる。したがって、各ステージの処理時間はすべて等しくなるように設計することが設計者に期待されている。

しかし、各ステージは機能毎に分割される場合が多いので、各ステージの処理時間を等しくできるとは限らない。このような場合、クロックで同期をとることによる無駄な待ち時間がステージ内に生じてしまう。

## 2.2 ノンオーバーラップ2相クロック方式

同期待ちのオーバーヘッドを無くすためには、そのステージの処理が完了したならば、直ちに次のステージに処理



Lxx: ラッチ  
E: Enable("L"のとき保持"H"のとき貫通)

図1: 2相クロックによるパイプライン

結果を転送することになればよい。これを実現する構成を図1に示す。パイプラインのひとつのステージはふたつのサブステージから構成される。ta1, ta2, tb1, tb2はそれぞれサブステージA1, A2, B1, B2の処理時間である。サブステージの間にはデータ貫通状態を持つラッチ（以下、単にラッチ）が置かれ、ノンオーバーラップ2相クロック[5]で制御する。

図2にタイミングを示す。サブステージ間にラッチを使用しているため、クロックのエッジではなくラッチが貫通状態にある半サイクルの間に、処理結果の転送が行われる。例えば、区間Ca1では、φ1が"H"レベルなのでラッチLaaは貫通状態となり、この間にサブステージA1の処理が完了すれば、処理結果は次のサブステージA2に直ちに渡される。区間Ca2では、φ2が"H"レベルなのでラッチLabは貫通状態となり、この間にサブステージA2の処理が完了すれば、処理結果は次のサブステージB1に直ちに渡される。

ここで、各サブステージが同期待ちをしないための条件は、次のようになる。Tをサイクル時間とし、時刻t0で入力が決定的な値とすると、時刻t1(=t0+ta1)にサブステージA1の処理が終了する。この時刻が区間(nT, (n+1/2)T)内にあればよい。したがって、

$$nT < t_1 < (n + 1/2)T$$

以下、同様に、

$$\begin{aligned} (n + 1/2)T &< t_2 < (n + 1)T \\ (n + 1)T &< t_3 < (n + 3/2)T \\ (n + 3/2)T &< t_4 < (n + 2)T \end{aligned}$$

\*Reducing length of the pipestages by an elastic method. Koichi TAKEDA, Masatoshi SATO, Teruhiko OOHARA  
Oki Electric Industry Co., Ltd.

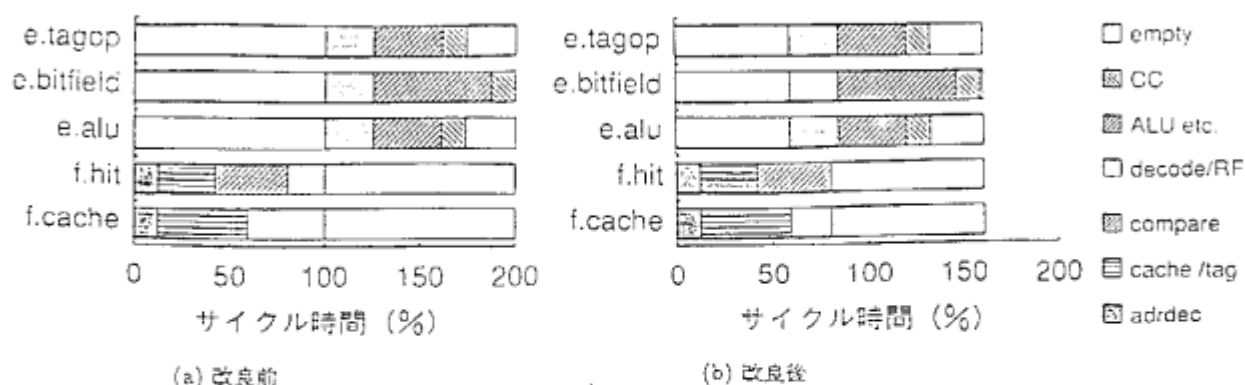


図 3: PIM/i の命令パイプラインの処理時間の割合

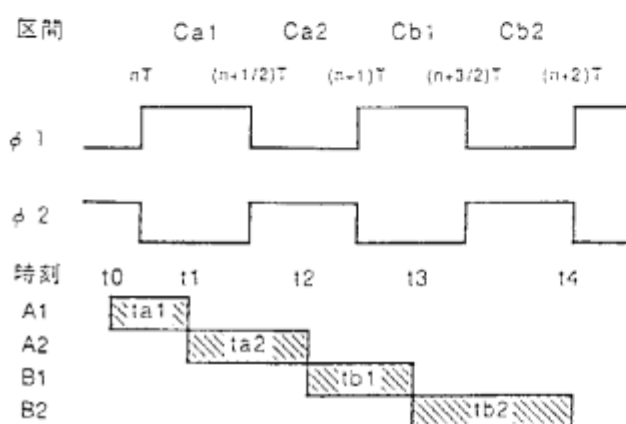


図 2: タイミング

となる。  
ただし、

$$(n - 1/2)T < t_0 < nT$$

$$t_1 = t_0 + ta_1, \quad t_2 = t_1 + ta_2$$

$$t_3 = t_2 - tb_1, \quad t_4 = t_3 + tb_2$$

### 3 PIM/i 要素プロセッサの命令パイプライン

PIM/i 要素プロセッサの命令パイプラインは、F(命令フェッチ)、E(実行)、W(書き込み)の3ステージから構成される。Fステージの処理は、命令アドレス出力とアドレスデコード、命令キャッシュとキャッシュタグのアクセス、アドレス比較である。命令キャッシュはダイレクトマッピング方式である。Eステージの処理は、命令デコードとレジスタ読み出し、演算、条件生成である。

図 3(a) は、Fステージ、Eステージの各処理部に、1 サイクル時間に対する処理時間の割合を示している。最も時間のかかる処理は E ステージにあり、サイクル時間はこの処理時間で決定される。このため、F ステージに同期待ちが生じ得る。

同期待ちを減らすために、E ステージを D、X のサブステージに分割し、前節で述べたようにサブステージ間に

ラッチを置く構成にした。これにより、F ステージの処理結果が D サブステージに直ちに渡される。ここで、ダイレクトマッピングのキャッシュの場合、キャッシュタグのアクセスの他にアドレス比較を行う必要のあるヒット信号よりも、命令のほうが早く供給される。また、命令デコードはキャッシュヒット信号の確定を待たずに開始して良い。このことを考慮して、図 3(b) に処理時間の割合を示す。ただし、サイクル時間は図 3(a) を基準にしている。これからわかるように、サイクル時間は約 80% に短縮できた。

### 4 おわりに

本稿では、パイプラインの各ステージの処理時間が等しくない場合でも、ノンオーバーラップ 2 相クロックを使って各ステージの処理時間のバラツキをステージ間で融通し合うことによって、パイプラインの同期待ちオーバーヘッドを吸収する方法を示した。この方法を PIM/i 要素プロセッサの命令パイプラインの設計に使用し、サイクル時間が短縮できたことを示した。

### 謝辞

日頃、助言をいただく(財)新世代コンピュータ技術開発機構(ICOT)第1研究室、および神電気のPIM担当諸氏に感謝する。

### 参考文献

- [1] 大原他：並列推論マシン PIM/i の概要、情報処理学会第 40 回全国大会、1990
- [2] 武田他：並列推論マシン PIM/i の要素プロセッサのアーキテクチャ、情報処理学会第 40 回全国大会、1990
- [3] H.S.Stone: High-Performance Computer Architecture, Addison-Wesley, 1990
- [4] 高橋義造編：並列処理機構、丸善、1989
- [5] C. Mead and L. Conway: INTRODUCTION TO VLSI SYSTEMS, Addison-Wesley, 1980