5P−3

PNプロセッサの機能分割とトークン通信に関する評価

有田隆也*　伊藤広明**　髙木浩光*　曽和将容*
(*名古屋工業大学工学部　**日本IBM)

1. はじめに

細粒度の並列度を効率的に抽出するプロセッサアーキテクチャとして、我々は PN プロセッサ[1]を提案し、開発を進めている。PN プロセッサには、1)命令の種類に応じたプロセッサの機能分割、2) シンプルで高速な命令実行順序制御方式の採用、などの特徴がある。プロトタイプモデルでは、プロセッサの処理を、1) 転送、2) 演算、3) フロー制御（分岐）、の3つに分割して、スカラ処理を細粒度並列実行化している。本稿では、シミュレーションに基づき、上記の特徴について評価する。

2 PN プロセッサの概要

PN プロセッサの基本構成を図1に示す。TU,AU,FCU は、データ転送命令、演算命令、フロー制御命令をそれぞれ実行する処理ユニットであり、それぞれを、転送ユニット、演算ユニット、フロー制御ユニットと呼ぶ。各処理ユニットにはそれぞれプログラムメモリ(TM.AM,FCM)が接続されており、独立に命令をフェッチする。プログラムはコンパイラにより、フロー解析した後、命令の種類によって、各メモリに分けられて格納される。データメモリは転送ユニットにのみ接続されている。各処理ユニットには、マルチポートのレジスタファイル（RF）が接続されている。

各処理ユニット間の命令実行順序制御は、単なる計数カウンタであるトークンカウンタ(TC)を用いた方式で実現される。トークンカウンタは、命令実行の進み具合の差を、処理ユニット間にまたがる命令先行関係の数によってカウントするものである。命令に付加されたタグによりトークン（命令実行の許可権を意味し、物理的には単なる電気パルス）は信号線 CTPを通じて伝達され、トークンカウンタをインクリメントする。このような機構は、シンプルであり、しかも、命令間に存在する先行関係のみを保証し、本質的に不要な待ちを生じさせないことが示されている[2]。

3. シミュレーションの内容と前提

PN プロセッサでは、3命令流の並列実行を実現するが、非均質型の分散を行っているため、機能分割のバランスの良さがその効率化にとって重要である。各命令の動的な実行頻度にその命令実行時間をかけた値を、各処理ユニットごとに加算したものを各処理ユニットの「負荷」とし、バランスの程度を調べる。

もうひとつの評価は、トークン通信の実行時間に関する影響である。パラメータとしての「実質通信時間」TD を次のように定義する（図2）。

$$TD = Tc - (Tr + Tz)$$

ここで、 Tc は、通信そのものに要する時間を表している。Tsは、その命令の実行終了時刻よりどれだけ早くトークンを送れるかということを表しており、逆に Trは、トークンを受け取るまでに、その命令が、どれだけ命令の実行（フェッチも含む）を進めることができる

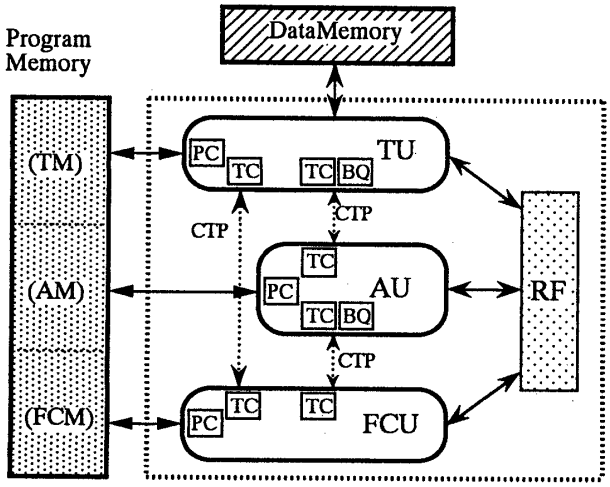

図1 ＰＮプロセッサプロトタイプの基本構成

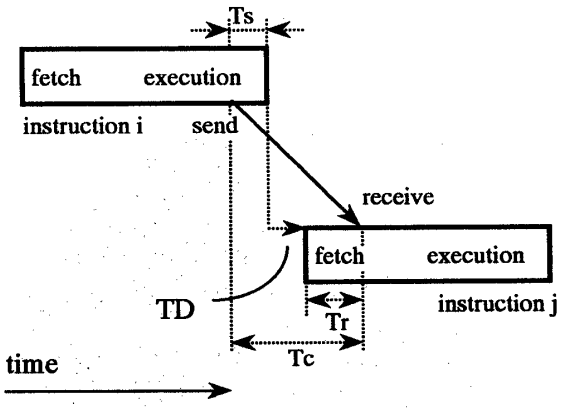

図2 トークン送受信のタイミング

Evaluation of Function-partitioning and Token-passing in the PN Processor

Takaya ARITA*, Hiroaki ITO**, Hiromitsu TAKAGI*, Masahiro SOWA*
*Nagoya Institute of Technology, **IBM Research

かを表している。したがって、トークンの送受信のタイミングをずらして Tr,Ts の値を大きくすることは、「実質通信時間」を減らすという最適化に相当する。

UNIX 上に記述された PN計算機の機能レベルシミュレータにより、上記の２点について評価する。各命令の実行時間は，従来の平均的なプロセッサの命令実行時間を参考にして設定した．ＰＮプロセッサは、それぞれの処理ユニット内でパイプライン化が可能であるが，本評価では各ユニット内でのパイプライン化は想定していない。評価に使用したプログラムは、バブルソート、クイックソート、ナイト＆ルク（パズル）、フィボナッチ数列計算、行列積計算の６個である．

4. シミュレーション結果

4.1 負荷バランス

６つのプログラムでの平均の「負荷」は、転送ユニット、演算ユニット、フロー制御ユニットでそれぞれ、49.9,54.6,28.3 であった（一命令当たりの平均実行時間はそれらの合計値 133である）。さらに細かい分布を含めて図３に円グラフを示す。Compare, Control は、それぞれ、比較関連命令、分岐関連命令をまとめたものである。転送ユニットと演算ユニットはほぼ負荷が等しく、バランスがとれていることがわかる。一方、フロー制御ユニットの負荷は、他処理ユニットの 55 % 程度である。もし、フロー制御ユニットの負荷を上げて飽和に近付けた場合には、1) フロー制御ユニットの先行処理（条件判断、飛び先アドレス転送）による他処理ユニットの「ゼロタイム分岐」（分岐処理の大幅軽減）の無効化、2) 無条件分岐時における転送ユニット、演算ユニットの本質的でない待ちの発生、を招く可能性が高い[3]。したがって、この程度の稼働率が妥当であると考えられる。

4.2 トークン通信

「実質通信時間」の実行時間に対する影響を図４に定性的に示す。

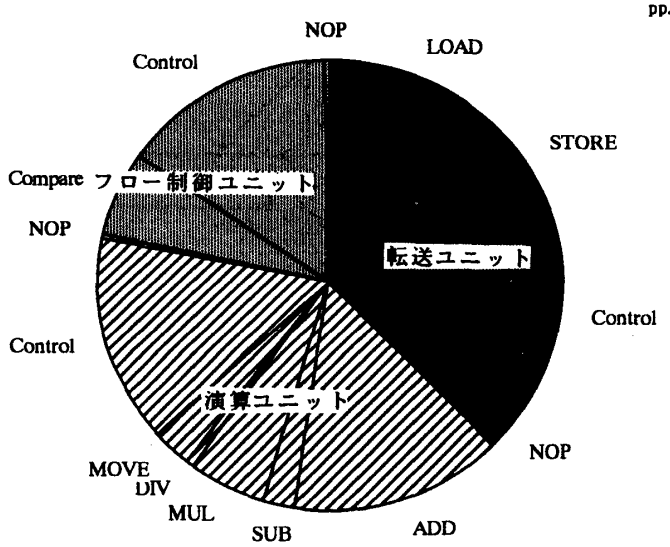

図3 各処理ユニットの負荷バランス

６つのプログラムすべてが、このような直線で構成されており、傾きが変化する点が存在する。これは、「実質通信時間」が増加していくと、待ちを伴う通信の回数に比例して、線形に実行時間が増加するが、ある時点で待ちを伴う通信の回数が増加し、傾きが変化するためであると考えられる。

通信自体は導線を通して電気パルスを送りカウンタをインクリメントする程度の簡単な機構で実現されるので、通信そのものに要する時間（Tc）は、極めて小さいと考えることができる。この値をたとえば、40 とし、Ts=20, Tr=50 とすれば、TD= -30 となる。この場合には、最適化をまったく行わず、かつ、通信そのものの遅延をゼロとした場合より、性能は約１割程度向上するということが、図４からわかる。

5. まとめ

PN プロセッサのもつ特徴について、シミュレーション結果に基づいて評価した。PNプロトタイプにおける各処理ユニット間の負荷バランスは、PNプロセッサのフロー制御ユニットの先行処理の特徴を考慮すると、妥当であると考えられる。通信時間の実行時間に対する影響に関しては、その最適化による効果も含めて、主に定性的に示した。詳細なPNプロセッサの設計や PN プロセッサで採用された命令実行順序制御方式の一般化などを進めている。

参考文献

1) 曽和将容，有田隆也，河村忠明，高木浩光，”機能分割型プロセッサによる複数命令流実行方式”，電子情報通信学会論文誌 D-I，Vol.J73-D-I, No.3、pp.280-285 (1990)．

2) 高木浩光，有田隆也，河村忠明，曽和将容，”問題が持つ先行関係のみを保証する高速な静的実行順序制御機構の構成法”，並列処理シンポジウム JSPP'90論文集，pp.57-64 (1990)．

3) 有田隆也，高木浩光，河村忠明，曽和将容、”PN プロセッサにおけるフロー制御方式について”、情報処理学会第 39 回全国大会論文集、pp.1896-1897 (1989)．

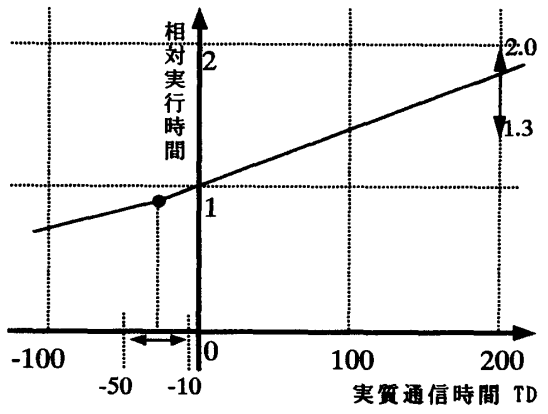

図４実質通信時間の実行時間に対する影響