6X-7

PNプロセッサにおける フロー制御方式について

有田　隆也　高木　浩光　河村　忠明　曽和　将容
(名古屋工業大学)

1. はじめに

機械語命令レベルからレジスタトランスファレベル程度の細粒度における並列実行方式である PN アーキテクチャ[1]では、プロセッサ内の複数の処理ユニットにそれぞれ命令流を独立に与えることによって並列実行を行う．各処理ユニットが独立に動作し、必要に応じて高速通信を行うので、VLIW方式[2]のようにコンパイラで完全にスケジューリングを行い実行時には同期を行わない方式より、きめの細かな並列度を抽出する可能性があり[3]、またタイミングの不安定なメモリなどのハードウェアを組み込むのも容易である．

処理ユニット数を３としたプロトタイプモデルは、従来の逐次的な機械語命令を、①データ転送命令②演算命令③フロー制御命令、の３つに分類し、制御フロー計算モデルに基づいた通信を行いながら、それぞれを専用の処理ユニットで実行する．フロー制御を行う処理ユニットに独自の命令流を与えたことにより、分岐判断の先行実行が可能となり、分岐処理に要する時間を大幅に減ずることも特長のひとつである[4]．本稿では、PN アーキテクチャの分岐命令の処理方式について比較検討する．

2. PN プロセッサの概要

PN プロセッサのプロトタイプモデルの構成を図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、転送ユニット、演算ユニット、フロー制御ユニットと呼ぶ．各処理ユニット間には、命令実行の許可を表すトークンを伝達する信号線 CTPが張られている．各処理ユニットにはプログラムメモリ（TM,AM,FCM）が接続されており、命令を独自にフェッチする．プログラムメモリには、制御フローグラフを構成する命令群が３種類の処理ユニットに対応して分割して格納されている．データメモリは、転送ユニット TU に接続されている．汎用レジスタファイル(RF)は、各処理ユニットが共有しており、複数の処理ユニットは異なるレジスタに対して同時にアクセス可能である．

各処理ユニット間で命令の実行許可を表すトークンを送受しながら実行は進む．トークンはパルスとして送り、しかも、送受信相手は確定しているので、個々のトークンの識別は必要ない．したがって、トークンの格納場所はキュー構成でなく、単にトークンの到着数を数えるカウンタとなっており、高速通信がシンプルなハードウェアで実現されている．処理ユニット間通信の有無は、機械語命令に付加するタグによって表す．命令の前に付くタグを前置タグと呼び、命令の後に付くタグを後置タグと呼ぶ．たとえば、

```
LOAD.a      R0,R1
```

は、LOAD 命令実行後に、処理ユニット a にトークンを送ってから、次の命令の実行に移ることを意味しており、

```
t.ADD       R1,R2,R3
```

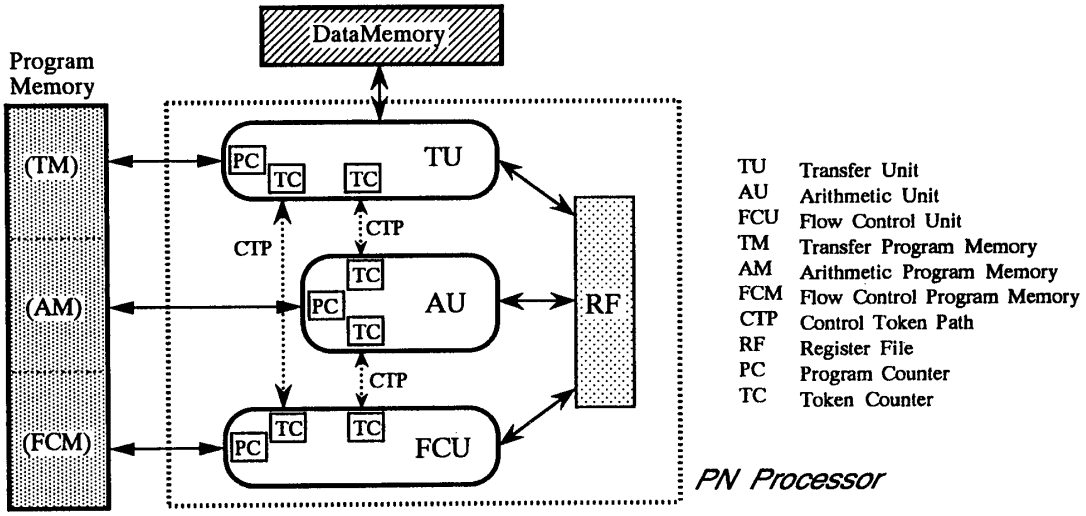

図１　PN プロセッサ（プロトタイプモデル）の構成

Flow Control Method of the PN Processor
Takaya ARITA, Hiromitsu TAKAGI,
Tadaaki KAWAMURA, Masahiro SOWA
Nagoya Institute of Technology

は、処理ユニット t から制御トークンが送られてきているのを確認した後、ADD 命令を実行することを意味している．トークンが未到着ならば、到着するまで待つ．送信側は、受信側が到着していなくても待つ必要がない．

3． フロー制御方式

フロー制御ユニットは条件分岐に関する処理を行うユニットである．本稿では、フロー制御ユニットが、①条件分岐のみ制御するか、無条件分岐も制御するか、ということと、②分岐先アドレスを他の処理ユニットに転送するか、分岐先アドレスを転送しない（各処理ユニットがフェッチする）か、ということを問題とする．検討する方式の分類を表１に示す．同表では、フロー制御ユニットが他の処理ユニットに転送する情報を条件分岐と無条件分岐に分けて記している．"None"は、制御しないということを表している．

A 方式では、条件分岐のときだけフロー制御ユニットが比較命令を実行した結果のみを他の処理ユニットのキューに送る．他の処理ユニットは、通常の条件分岐命令により、キューからのデータに従い分岐を行う．フロー制御ユニットが比較命令を実行した時点ですぐに結果を送ることができる．フロー制御ユニットと他の処理ユニットとの間にバスを張る必要がない．

B 方式では、フロー制御ユニットが条件分岐命令を実行したときに比較結果とともに条件成立時の各処理ユニットの飛び先アドレスも送る．フロー制御ユニットと他の処理ユニットとの間にバスをひく必要があるが、条件分岐命令をなくして前命令にブランチタグ[4]を埋め込むことにより、分岐にかかる時間をゼロに近づけられる．

C 方式では、フロー制御ユニットが無条件分岐のときも分岐先アドレスを他の処理ユニットに送信する．フロー制御ユニットが先行することが多い場合、他の処理ユニットは、分岐命令のときだけでなく無条件分岐命令のときも分岐時間をゼロに近づけられる．しかし、制御フロー処理ユニットの処理が重い場合、無条件分岐でも他の処理ユニットに待ちが発生する．

分類としては上記３方式に帰着するが、理論的にはつねに上記３方式より遅くならない融合方式 D がある．条件分岐の場合は B,C 方式と同じである．無条件分岐のときは、①ブランチタグ付きの命令の実行終了までに真偽およびアドレスが到着した場合、分岐先アドレスへ制御を移す（B 方式と同様）．②届いていない場合、ブランチタグ付きの命令の次に書かれた（①の場合はスキップされる）無条件分岐命令の実行に移り、それによって分岐し（A 方式と同様）、後で届いたアドレスなどは無効化する．さらに、③両者(①②)の間、つまり無条件分岐命令実行中に到着した場合、即座に切り替え、飛び先命令のフェッチと実行に移るという方式である．

表１ フロー制御ユニットの転送内容による分類

Method	Conditional Branch	Unconditional Branch
A	True/False	None
B	True/False, Address	None
C	True/False, Address	Address
D	True/False, Address	Address/None

4． シミュレーション

機能レベルのシミュレータによってサンプルプログラムを実行した．パイプライン化せず、命令実行時間が均一であり、トークン信号の伝達時間はゼロとしている．トークン到着の有無の判断は、命令フェッチ、デコード、あるいは実行ステージの一部を終えるまで遅らせることができる．この判断実行時点を１命令実行時間の 50% 経過時としている．

D 方式はかなり制御が複雑になることが予想されるので、プロトタイプモデルとして、A,B,C の各方式について検討した．各方式に準ずる PN プロセッサシミュレータで、Sum（加算のループ）、Bubble（バブルソート）、Matrix（行列の乗算）の３つを実行した結果を実行ステップ数で示す（表２）．同表より、A 方式よりも、B,C 方式のほうがやや優れており、B,C 方式にはあまり差がないことがわかる．ただし、フロー制御ユニットの負荷が比較的軽い Matrix では、無条件分岐の先行実行の効果により、やや C 方式のほうがよい結果を示している．

表２ シミュレーション結果（実行ステップ数）

Program	Method A	Method B	Method C
Sum	67	54	54
Bubble	394	386	394
Matrix	499	449	408

5． 終わりに

PN プロセッサのプロトタイプモデルにおけるフロー制御方式について比較検討を行った．シミュレーションでは、無条件分岐のときもフロー制御ユニットがアドレスを転送する B,C 方式が優れているという結果が示された．

参考文献

1) 曽和将容、"ＰＮコンピュータの提案"、情報処理学会「コンピュータアーキテクチャ」シンポジウム予稿集、pp.109-114(1988).

2) J.A.Fisher,"Very Long Instruction Word Architecture and the ELI-512",Proc. of 10th International Symposium on Computer Architecture, pp.140-150 (1983).

3) 高木浩光、曽和将容、有田隆也、"ＰＮコンピュータに向いた命令セットアーキテクチャ"、電気情報通信学会研究報告、Vol.89,55,pp.17-23 (1989)．

4) 河村忠明、曽和将容、"ＰＮコンピュータにおける分岐処理"、電気情報通信学会研究報告、Vol.88,155,pp.67-72(1988).