

## 非同期式プロセッサ TITAC-3 の命令供給機構

2H-5

藤木崇宏

Byung-Soo Choi

中村 宏

南谷 崇

東京大学 先端科学技術研究センター

### 1 はじめに

素子の高速化によりゲート遅延が小さくなる一方、回路面積の増大によって配線遅延は絶対的・相対的に増加している。そのため、チップ全体を単一のクロックで同期させる同期式回路では、素子の高速性を十分活かすことができない。

この問題を解決する方法の1つとしてシステムを非同期式で構成する方法がある[1]。現在実現されている非同期式汎用プロセッサには、複数命令の同時発行を行っているものがない。これに対し、我々の関与している非同期式プロセッサ TITAC-3 は、複数命令の同時発行を行うものである[2]。本論文では、TITAC-3 の命令供給機構を提案する。

### 2 命令供給方式

スーバスカラ・マシンにとって、分岐による命令流の分断は大きな性能低下を引き起こす。性能向上のために通常分岐予測を行うが、分岐の方向と分岐先をより速く、より正確に予測することが重要である。従来の分岐先バッファ(Branch Target Buffer(BTB))[3]は、同時にフェッチしようとしている複数命令内に分岐命令がある場合、分岐先の命令は次のサイクルにフェッチしていた。しかしこの方法では、分岐命令以降の命令は同時にフェッチできないため、分岐によってフェッチ幅が制限されてしまう。しかし分岐命令でフェッチ幅を区切るのではなく、分岐先の命令も同時にフェッチすれば、より高いフェッチ幅を得ることができると考えられる。

そこで、このような分岐を越えて命令をフェッチする命令供給方式について考える。

複数の分岐を越えた命令列を同時に発行する方式として、*Collapsing Buffer*を用いた方法がある[4]。この方法では、複数の分岐を越えた命令列をフェッチするために、BTBを複数回参照して、複数の分岐予測を同時に行い、複数の Basic Block をフェッチ、連結して1つの命令列を作っている。この方法で問題となるのは、複数の分岐を越えた1つの命令列を作るのに大きなコストと時間がかかってしまうことである。その問題を解決する方法として、1度作った命令列をキャッシュに保持して再利用する *Trace Cache*[5]がある。

しかし TITAC-3 の命令実行部[2]は

- 4命令同時発行

- 同時に発行できる分岐命令数は1

であるので、*Collapsing Buffer*を用いた方法[4]のように複雑な構成にする必要はない。1度にフェッチする命令列は、分岐命令までの命令列と分岐先の命令列の2つの命令列である。分岐先の命令列が分岐命令を含んでいれば、その分岐命令はフェッチしない。

分岐命令までの命令列と分岐先の命令列を同時にフェッチするため、従来の BTB[3]では、分岐先の分岐情報も調べるために BTB を2度参照する必要があった。しかし1度の参照で分岐先の分岐情報がわかるように従来の BTB[3]に分岐先の分岐情報を付加すれば、2度参照する必要はなくなる。

そこで、TITAC-3 の命令供給機構として、以下の方式を提案する。

- 発行しようとする4命令中に分岐命令が含まれるとき、分岐先の命令も同時に発行(但し、2個目の分岐命令は発行しない)

- BTBは分岐命令とその分岐先の情報だけでなく、分岐先にある命令の分岐情報も保持
- より高い精度の分岐予測をするために BTB と分岐予測機構を分割
- 分岐予測と命令フェッチを並列に行うために、命令供給機構を2つのステージに分割

### 3 命令供給機構の構成

非同期式プロセッサ TITAC-3 の命令供給機構は、Branch Target Buffer、分岐予測機構、2-Port の命令キャッシュから構成される(図1)。

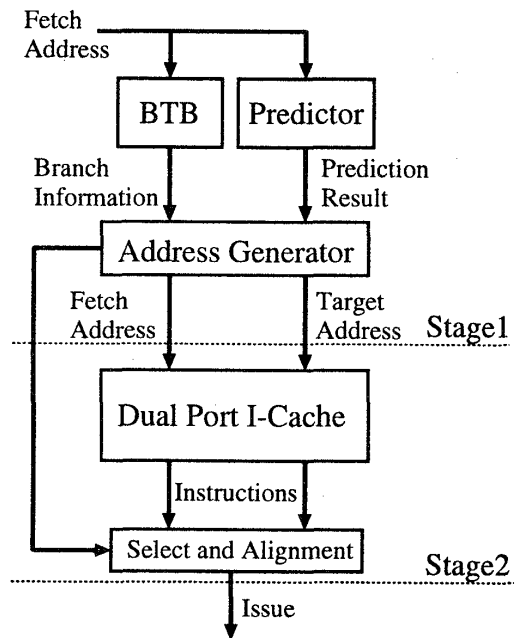


図1: 命令供給の流れ

第1ステージ 発行しようとする命令内に分岐命令があるか BTB で調べ、分岐命令があれば、分岐先アドレスなど (Branch Information(3.1)) を出力する。またそれと並行して分岐予測(条件分岐のみ)を行う。その結果から Address Generator が命令キャッシュをアクセスするためのアドレスを生成し、フェッチすべき命令数を決める。

第2ステージ 第1ステージで生成されたフェッチアドレスによって、命令キャッシュから命令列をフェッチし、有効な命令のみ取り出し、並び替えて (Select and Alignment) 発行命令列を生成する。

#### 3.1 Branch Target Buffer(BTB)

BTB[3]は

- これから発行しようとする命令内に分岐命令があるか調べる。
- 分岐命令が含まれるならば分岐先のアドレスを出力する。

を行うための情報を保持するバッファである。

本稿では、1回の参照で分岐先の命令内の分岐情報を知ることができる BTB を提案している。そのためには、BTB

のエンタリに分岐先の命令から2個目の分岐命令までの距離(命令数)という情報を加えれば、BTBを1回参照するだけで分岐先の発行可能な命令数を知ることができる。

BTBは以下のエンタリ(図2)を持つ。

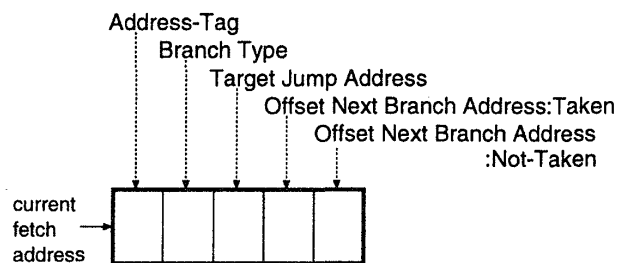


図2: Branch Target Buffer

**Address Tag** 分岐命令の Tag(Tagが一致すればその命令は分岐命令)

**Branch Type** 分岐命令の種類(無条件分岐、条件分岐、間接分岐)

**Target Jump Address** 分岐先アドレス

**Offset of Next Branch Address:Taken** 現分岐命令が成立(Taken)と予測されたときの分岐先の命令から次の分岐命令までの距離

**Offset of Next Branch Address:Not Taken** 現分岐命令が不成立(Not Taken)と予測されたときの分岐命令から次の分岐命令までの距離

図1の Branch Information は、上の Branch Type, Target Jump Address, Offset of Next Branch Address に相当する。BTBを参照することによって得られる上の情報から分岐命令があるかないか、分岐命令があれば、分岐先アドレス、フェッチすべき命令数がわかり、分岐予測の結果と合わせることによって、Address Generator(図1)は、命令キャッシュからフェッチすべき命令列とその必要な命令数を決定することができる。そして次サイクルにフェッチするアドレスを決めることができる。

### 3.2 分岐予測機構(Predictor)

分岐予測機構は以下のような方式である。

- 条件分岐の分岐方向(成立・不成立)を予測する。
- 分岐予測は1サイクルに1分岐命令まで行う。
- 予測方法は、2ビット飽和型アップダウンカウンタ方式[3]と予測する分岐命令アドレスとグローバル分岐履歴をXORしたもので、2ビットカウンタを参照し、予測結果を出力する gshare[6]とを組み合わせたハイブリッド方式[6][7]を用いる。

### 3.3 命令キャッシュ

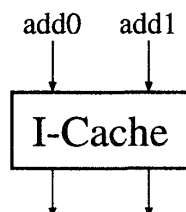


図3: 2 Port 命令キャッシュ

- ダイレクトマップ
- 2 Read Port

の命令キャッシュ(図3)を考える。図1の Address Generator は、ラインフェッチに必要なラインを指定する先頭アドレス(add0,add1)を生成する。add0はカレントフェッチアドレスであり、add1はフェッチする命令列内に分岐命令があればその分岐先アドレス、あるいはフェッチする命令列がライン境界を越えて2ラインにまたがる場合は2ライン目を指定するアドレスである。

Dual Port Cache を用いることによって、分岐命令までの命令列と分岐先の命令列を同時にフェッチすることができ、またフェッチする命令列が1ラインを越えても連続する命令をフェッチすることができる。

4命令内に分岐命令がなく、ライン境界を越えないときは1ラインのみフェッチすれば良い。4命令内に分岐命令が存在し、分岐命令までの命令列と分岐先の命令列がそれぞれライン境界を越えるときは最大4ラインフェッチしなければならない。しかしその頻度はそれほど多くないと考えられるので、設計の容易さ、コストの面からも2Portで十分であると考えている。

### 3.4 Select, Align of Instructions

最後にキャッシュからフェッチしたラインの中から有効な命令を取り出し、正しい順序に並び替えて4命令を発行する。

## 4 まとめ

非同期式プロセッサ TITAC-3 の命令供給機構の構成方法を提案した。

より高い命令フェッチ幅を得るための方式として、分岐でフェッチ幅を区切るのではなく、分岐先の命令も同時にフェッチする方式を考えた。それを実現するための、分岐先にある命令の分岐情報を付加したBTBの構成方法を示した。また、分岐をまたいだ2つの命令列を同時にフェッチできる2Portの命令キャッシュ、より精度の高い予測をするためのハイブリッド方式を用いた分岐予測機構の構成方法を示した。

本研究の一部は科学研究費補助金(基盤研究(B)09480049)、及び(株)半導体理工学研究センターとの共同研究によるものである。

## 参考文献

- [1] 南谷崇, “非同期式プロセッサ — 超高速VLSIシステムを目指して —,” 情報処理, Vol.34, No.1, pp.72-80, Jan. 1993.
- [2] 小沢基一, 中村宏, 南谷崇, “非同期式プロセッサ TITAC-3 の命令実行機構,” 情報処理学会 第58回全国大会 1999.
- [3] J. K. F. Lee, and A. J. Smith, “Branch Prediction Strategies and Branch Target Buffer Design,” *IEEE Transactions on Computer*, Volume 17, Number 1, Jan. 1984, pp.6-22.
- [4] T. Conte, K. N. Menezes, P. M. Mills, and B. A. Patel, “Optimization of Instruction Fetch Mechanisms for High Issue Rate,” *Proceedings of the International Symposium on Computer Architecture*, June 1995.
- [5] E. Rotenberg, S. Bennett, and J. E. Smith, “Trace Cache: A Low Latency Approach to High Bandwidth Instruction Fetching,” *Proceedings of the 29th annual International Symposium on Microarchitecture*, 1996.
- [6] Scott Mcfarling, “Combining Branch Predictors,” *WRL Technical Note TN-36, Digital Equipment Corporation*, June 1993.
- [7] Po-Yung Chang, Eric Hao and Yale N. Patt, “Alternative Implementations of Hybrid Branch Predictors,” *Proceedings of 28th annual International Symposium on Microarchitecture*, Nov. 1995.