

VLIW プロセッサにおける命令圧縮方式

5K-5

田中 和彦, 鍛田 真(*), 野尻 徹, 西岡 清和

(株) 日立製作所 システム開発研究所

(株) 日立製作所 デバイス開発センタ(*)

1 はじめに

メディアプロセッサ MAPCA は、4 命令同時実行の VLIW (Very Long Instruction Word) 型プロセッサである [1] [2]。VLIW アーキテクチャでは、同時に実行する命令を連結し、長命令を構成する。処理に依存関係があるために命令を割り付けられない演算器には、NOP (No Operation) を供給する。MAPCA では、NOP の除去により長命令を圧縮した形式をとることで、キャッシュの使用効率を向上した。本報告では、この命令圧縮方式について紹介する。

2 命令ブロック形式による命令圧縮

命令圧縮方式をとる MAPCA は、NOP を除去した長命令と NOP の復元に使用する属性情報から構成される 256 バイトの命令ブロックを単位とする命令形式を使用する (図 1)。

通常、4 命令同時実行の VLIW 型プロセッサでは、命令アドレスと演算器の対応がとられ、それらの関係は次式で表される。

$$\text{命令アドレス} = w(4n+i) \rightarrow \text{演算器 } i \text{ で実行}$$

$$(n=0, 1, 2, \dots, i=0\sim 3, w \text{ は命令幅}(=4))$$

ところが、本圧縮方式では、長命令から NOP を除去するため上記の対応関係が崩れる。そこで、次のフィールドから構成される 4 ビットの属性情報を各命令に付加し、命令と演算器の対応関係を保持する (図 2)。

- (1). 演算器指定フィールド：対応する命令を実行する演算器番号 i を指定する。
- (2). 終了フラグ：長命令の境界を表す。対応する命令が圧縮後の長命令中の最終命令となる場合は 1、それ以外の場合は 0 を指定する。

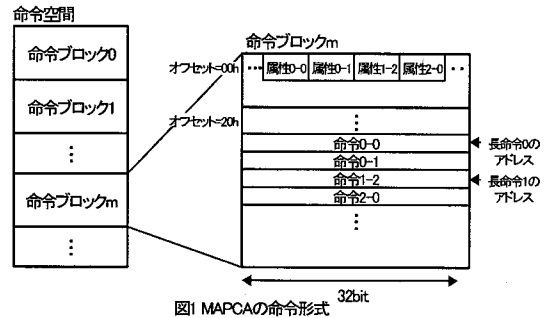


図1 MAPCAの命令形式

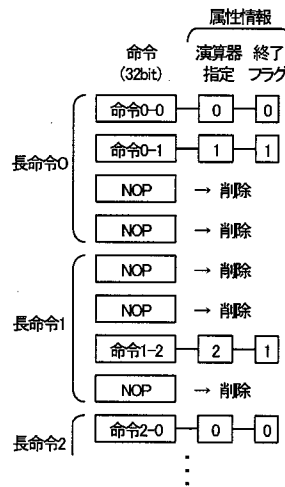


図2 属性情報の付加

この命令形式では、先頭 32 バイトの領域に、1 命令ブロックに含まれる命令に対応した属性情報をまとめて格納し、残りの領域に命令を格納する。各命令とそれに対応する属性情報のブロック内オフセットは次式で表される。

$$\text{命令} : n \times 4h + 20h$$

$$\text{属性情報} : \text{int}(n/2) \text{ の上位 } 4\text{bit} \quad (n: \text{偶数})$$

$$\text{〃} \quad \text{の下位 } 4\text{bit} \quad (n: \text{奇数})$$

$$(n=0\sim 55, \text{int}(x) \text{ は } x \text{ を超えない最大の整数})$$

プログラムカウンタは各長命令の先頭アドレスを指す。通常は、現在の長命令のサイズを加算すること

Instruction compression mechanism for VLIW processor
Kazuhiko Tanaka, Makoto Kuwata(*), Tohru Nojiri, Kiyokazu Nishioka
Systems Development Laboratory, Hitachi, Ltd.
Device Development Center, Hitachi, Ltd. (*)

によりプログラムカウンタを更新する。ただし、属性情報を格納した領域は読み飛ばす。分岐時には、分岐先アドレスを次のプログラムカウンタ値とし、これを用いて命令キャッシュから命令と属性情報を読み出す。

MAPCA の命令転送経路を図 3 に示す。主記憶から命令バッファまでは、長命令を圧縮した形式で格納する。

命令読み込みバッファには、1 命令ブロック分の命令とそれに対応する属性情報を格納する。命令読み込みバッファから命令キャッシュへの転送は、4 組の命令と属性情報から構成されるキャッシュライン単位で行う (図 4)。

命令バッファは、2 つのエントリから構成され、キャッシュラインにまたがる長命令 (例: 図 4 の命令 2-1~2-3) を読み出す。

命令分配器は、命令バッファより、現在のプログラムカウンタが指す命令から終了フラグが 1 の命令までを取り出し、次に実行する長命令として演算器指定フィールドに従い各演算器へ供給する。実行する命令が存在しない演算器には、NOP を供給する。

命令バッファの一方のエントリに含まれる全ての命令の供給が完了した時点で、命令キャッシュからそのエントリへ次のキャッシュラインを転送する。この転送で命令キャッシュミスが発生した場合には、命令読み込みバッファから命令キャッシュ、命令キャッシュから命令バッファへの転送を順次行う。命令読み込みバッファにも該当する命令が無い場合は、主記憶から命令読み込みバッファへの転送を行った後、命令キャッシュ、命令バッファへの転送を順次行う。

3 評価

本方式では、属性情報の付加によりオーバーヘッドが発生する。このため、長命令列中の NOP の割合が 1/8 未満である場合は、本方式を使用することで逆に命令キャッシュの利用効率が低下する。そこで、実際のアプリケーション (MPEG2 デコーダ) で命令圧縮後のプログラム容量を算出したところ、元のプログラムの約 57% となり本方式の有効性が確認された。

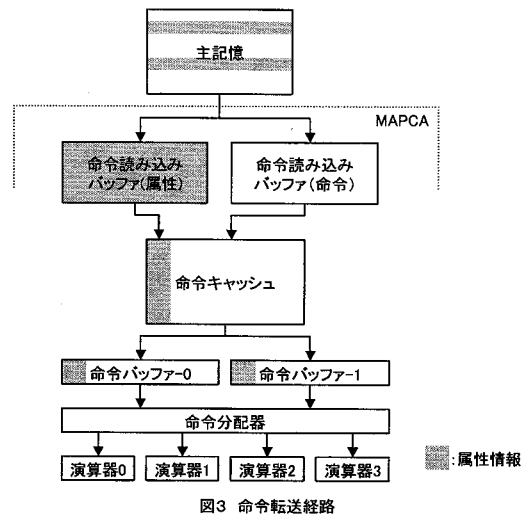


図3 命令転送経路

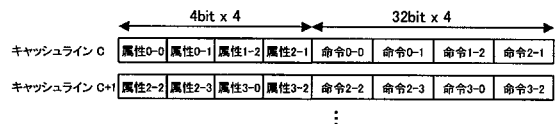


図4 命令キャッシュ中の格納形式

MAPCA では、命令圧縮のために命令キャッシュの約 7% の面積に相当する回路を付加している。一方、命令圧縮を行わずに同じキャッシュヒット率を達成するためには、1/0.57 倍の容量を持つ命令キャッシュが必要であり、約 56% の面積増となる。両者を比較すると、本方式により命令キャッシュの面積は約 69% (=1.07/1.56) に削減されるという結論が得られる。

4 おわりに

VLIW 型のプロセッサで、命令キャッシュの使用効率を向上するために、ブロックを単位とした命令形式を用いる命令圧縮方式を提案した。

本方式を MAPCA に適用し、命令キャッシュのヒット率を低下させることなく命令キャッシュの面積を約 69% に削減した。

参考文献

- [1]. Chris Basoglu et al., "The MAP1000A VLIW Media-processor", IEEE Micro March-April 2000
- [2]. Peter N. Glaskowsky, "MAP-CA Ready for Prime Time", Microprocessor report, March 2000