

TRON仕様32ビットマイクロプロセッサG_{MICRO}/100

2U-5

(1) マイクロプログラムによる高機能命令の実現とその評価

岩田俊一 清水徹 松尾雅仁 吉田豊彦 日向純一 富沢治

(三菱電機株式会社 L S I 研究所)

1. はじめに

TRON仕様に基づく32ビットマイクロプロセッサG_{MICRO}/100は、命令の実行制御を水平型マイクロプログラムで行っており、基本命令の大部分を1マイクロ命令で実行できる。また、マイクロプログラムによってデータ演算部における処理をパイプライン化することにより高機能命令、特にビットマップ処理命令の高速実行を可能にした。本稿では、特にマイクロプログラムによるパイプライン制御について詳しく述べ、シミュレーションによる評価結果もあわせて報告する。

2. マイクロプログラム開発の狙い

G_{MICRO}/100は図1で示すような5段のパイプライン制御を行っており、マイクロROMはその中の4段目のパイプラインステージにある。命令コードは命令フェッチ部でフェッチされた後、命令デコード部1、2でデコードされる。デコードされた結果、マイクロプログラムの実行開始番地とその命令に必要なパラメータがマイクロROM部へ出力される。マイクロROMから読みだされたマイクロプログラムはパラメータと共にマイクロ命令デコーダによってデコードされ、データ演算部を制御する信号を出力する。データ演算部はマイクロROM部から出力された制御信号によって、命令で指定された処理を行う。

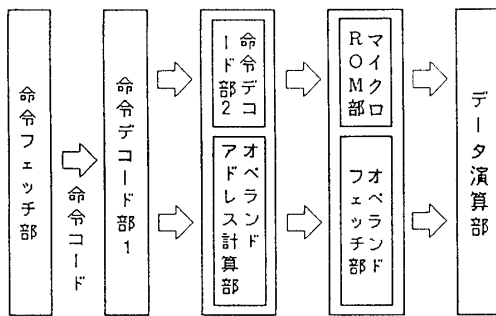


図1 G_{MICRO}/100のパイプライン処理

G_{MICRO}/100が備える命令を大きく分けると、MOVやCMPなどの基本命令とビットマップ処理命令などの高機能命令とがある。プロセッサの高性能化のためにはどちらの命令も高速な実行が必要であるが、基本命令と高機能命令とではマイクロプログラムの果たす役割が違うため高速化のためのアプローチが異なってくる。

基本命令の処理は単純なため、データ演算部の演算器は1回の動作で済む場合がほとんどである。このような場合、マイクロプログラムは1つの命令コードが来る度にデータ演算部の制御信号を1回出力するとい

う処理を行うので、命令デコードの一部として働くことになる。基本命令は使用頻度が多く、これらの命令の実行速度がプロセッサ全体の性能を大きく左右するため、できるだけ高速にする必要がある。そこで、基本命令を1マイクロ命令で実行させるために、マイクロプログラム1ワードでデータ演算部すべての制御を行えるような水平型マイクロプログラムとした。G_{MICRO}/100のマイクロプログラムの1ワードは45のフィールドで構成されており、1マイクロ命令が2クロックサイクルで実行される。

G_{MICRO}/100は多くの高機能命令を備えているが、特にワークステーションに必要な不可欠な高速なウィンドウ操作を可能にするためにビットマップ処理命令が強化されている。高機能命令の場合には、基本命令を組み合わせて実現される機能をマイクロプログラムで行っているのであり、いわばチップ上にコーディングされたサブルーチンとしての役割を持っている。高機能命令には同じ処理を何度も繰り返し行う箇所が含まれていることが多い。このような繰り返し処理に対して前回の繰り返し処理の一部と次回の繰り返し処理の一部とをオーバーラップさせるようプログラムすることにより、繰り返し処理の実質的な処理時間の短縮を図っている。つまり、マイクロプログラムによってデータ演算部の処理をパイプライン制御しているわけである。以下、ビットマップ処理命令のひとつであるBV MAP命令を例にして説明する。

3. マイクロプログラムによるパイプライン制御

BV MAP命令はビットマップの転送や重ね合わせを行うための命令で任意長のビットフィールド間で論理演算を行うことができる。つまり、ソース側のビットフィールドとデスティネーション側のビットフィールドとの間で各種論理演算を行い、その結果をデスティネーション側のビットフィールドに格納する。ビットフィールドは、ベースアドレス、ビットオフセット及びビットフィールドの長さによって表され、ビット単位で任意の位置、任意の長さを指定できる(図2)。

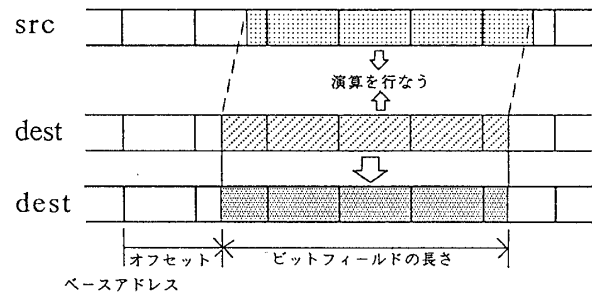


図2 BV MAP 命令

Implementation of a 32-bit microprocessor G_{MICRO}/100 based on TRON specification (1)Implementation and evaluation of microprogram

Shunichi IWATA, Toru SHIMIZU, Masahito MATSUO, Toyohiko YOSHIDA, Junichi HINATA, Osamu TOMISAWA
MITSUBISHI ELECTRIC CORPORATION

G_{MICRO}/100は32ビットの外部バスを持ち、1度に4バイトのデータをアクセスできる。BVMAP命令では4バイト単位で演算及び転送を行うため、ビットフィールドの両端を除くと4バイトのデータの演算と転送という処理の繰り返しとなる。この繰り返し処理を高速に行うことがBVMAP命令の高速化につながる。

G_{MICRO}/100においては、BVMAP命令の1回の繰り返し処理は図3で示すような5ステップとなる。各ステップを1マイクロ命令で実行すれば5マイクロ命令を要する。

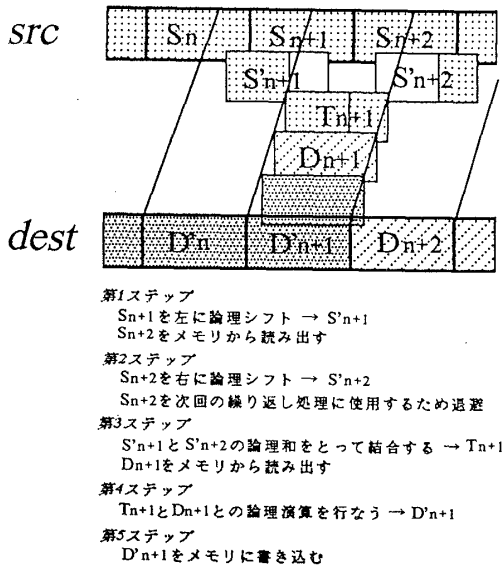


図3 BVMAP命令の処理方法

ところで、1回の繰り返し処理にはソース側の読みだし、デスティネーション側の読みだし及び書き込み、の計3回のメモリアクセスを行っている。1回の繰り返し処理を3マイクロ命令で実行すれば、外部バスを最大限に利用できる。これを実現するために、図3で示す第1ステップと第4ステップ、第2ステップと第5ステップをそれぞれ並列に処理し、繰り返し処理を3マイクロ命令で実行できるようにした(図4)。このように、データ演算部での繰り返し処理をマイクロプログラムでパイプライン制御することにより、外部バスを最大限に使い、実質的な処理時間を短縮できた。

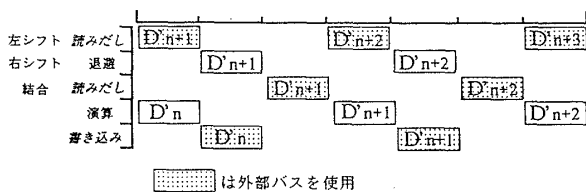


図4 マイクロプログラムによるパイプライン処理

繰り返し処理は、BVMAP命令だけでなく多くの高機能命令に含まれている。そのような命令に対してもG_{MICRO}/100では上で述べたようなマイクロプログラムによるパイプライン制御を行い、繰り返し処理の実行ステップ数を減らした。このようにして、繰り返し処理の含まれる高機能命令の高速化を図った。

4. 評価

ビットマップ処理命令の実行速度をG_{MICRO}/100の機能をクロックレベルまで記述したシミュレータを用いて評価した。

ビットフィールド間で論理積をとりながら転送するという処理を、BVMAP命令を用いた場合と、AND、MOV、分岐命令を組み合わせた場合の両方について比較した。さらにシミュレーションは、ビットフィールドの長さ及びメモリウエイトのクロック数を変化させながら行った。結果を図5に示す。縦軸が実行クロック数、横軸がビットフィールドの長さをバイト数で表したものである。

メモリウエイトがない場合、BVMAP命令は、AND、MOV、分岐命令を組み合わせた場合の38%のクロック数で実行できる。また、メモリウエイトが2クロックの場合でも60%のクロック数で実行できることが確認できた。

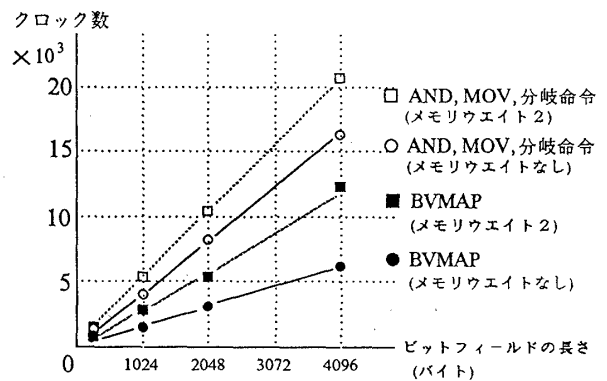


図5 シミュレーション結果

5. 結論

TRON仕様マイクロプロセッサG_{MICRO}/100は、命令実行制御に水平型マイクロプログラムを用いて基本命令の大部分を1マイクロ命令で実行できるようにした。また、高速ウィンドウ処理に必要なビットマップ処理命令などの高機能命令に対しては、マイクロプログラムによってデータ演算部をパイプライン制御することで高速化を図った。シミュレーションによる評価では、BVMAP命令が基本命令を組合せた場合の38~60%の実行時間で処理できることが確かめられた。

謝辞

本研究に関して有意義な討議と助言を頂いた東京大学坂村助教授をはじめ、G_{MICRO}/100の開発に携わっている方々に感謝致します。

(参考文献)

- [1]K.Sakamura; "Architecture of VLSI CPU in the TRON Project", 2nd TRON Project Symposium, TRON Association, Mar. 1987.
- [2]榎本龍弥; "TRON仕様に基づいた32ビットマイクロプロセッサ", 第2回TRONプロジェクトシンポジウム, TRON評議会, Mar. 1987.
- [3]T.Shimizu et.al.; "A 32-bit Microprocessor Based on the TRON Architecture Design of the G_{MICRO}/100", COMPCON Spring 1988.