シミュレータの高速化に関する一考察

4K-9

川北　英幸　　平尾　繁晴
株式会社　東芝　システムソフトウェア技術研究所

1．はじめに

東芝では、TRONCHIP仕様に基づく３２ビットマイクロプロセッサＴＸ１およびＴＸ１応用開発支援システムを開発中である。[1][2]
この開発支援システムで開発したソフトウェアをデバッグ及びテストするためには、ホストシステムであるＥＷＳ上でターゲットシステムのＴＸ１、メモリ、Ｉ／Ｏの動作をソフトウェア的に模擬するシミュレータが必要である。
３２ビットマイクロプロセッサＴＸ１では、データ、プログラムのサイズが１Ｍバイト以上といった大きなソフトウェアを動作させることが予想される。この様に大きなプログラムをデバッグおよびテストする場合、命令の高速実行が極めて重要である。ところで、ＴＸ１は複雑な命令セットを持っており、デコードの高速化に対しては特に考察が必要である。
本発表では、シミュレータの命令実行の高速化に関する一考察について報告する。

2．　命令実行のシミュレーション

シミュレータの命令実行は、メモリ上のプログラムの命令を１命令づつ順番に処理し、ＴＸ１で行なっているパイプライン処理のシミュレーションは行なわない。
[3] そして、１命令に対して以下に示す(1) から(3) までの処理を実際に次のような順で行なっている。
(1)　メモリにおかれているコードをフェッチし、以下に示すのアドレス計算、実行処理に必要なデータ（１命令につき１０バイト）に変換する。
(2)　(1) の処理したデータをもとに実行する命令がメモリオペランドを持っている場合には、そのアドレスを計算する。
(3)　(1) で処理したデータ、(2) で得たアドレスをもとにその命令の処理を行なう。
また、このシミュレータでは、命令シミュレーションに関して、(1) の処理を行なう命令デコーダ部、(2) の処理を行なうアドレス計算部、(3) の処理を行なう実行処理部の３つのブロックに分けた。

2．1　命令デコーダ

命令デコーダ部は、ＴＸ１の命令フォーマットを解釈し、オペランドアドレス計算、実行処理が可能な形式に変換する部分である。ＴＸ１の命令セットは１６ビットを基本命令長としており、次のような特徴を持っている。[1]

(1)　同一の命令に対して複数のフォーマットを持っている。例えばｍｏｖ命令のフォーマットは７種類ある。
(2)　また、命令のフォーマットを識別するオペコード部は、不連続に置かれており先頭の１６ビットだけで命令コードを識別できない（オペランドを示すコードがオペコードの間にある）こともある。

今回、以下の方針による命令のデコードを検討した。

(1)　命令群を分けるビットフィールドを見て分岐し、デコードに必要な情報を得る。
(2)　命令ビットパターンの先頭からのビットフィールドをキーとしてテーブルを引き、デコードに必要な情報を得る。

(1) の方法で、容易に作成及びコーデイングを行なうことが可能であるが、いかなる命令でもデコード結果を得るためには、少なくとも数回条件分岐を行なう必要がある。
(2) の方法では、命令実行時にテーブルを１回引くだけでデコードを完了することが出来るが、テーブルのサイズはビットフィールドの幅の２のべき乗に比例して増加する。

この命令セットでは、デコード処理時間の短縮化が重要であると判断し、テーブルを引く方法を用いることにした。ところが、先頭からの１６ビットでデコードを行なうとすると命令の全フォーマット１６４種中の７１種デコードを完了することが出来るが、デコードテーブルは６４０ＫＢ必要となる。そこで、ＴＸ１の全命令フォーマット中でデコードが完了する命令の種類を調べた。

ビット幅	フォーマット	サイズ(ＫＢ)
8	２２	２．５
１０	２０	１０
１２	６	４０
１６	２３	６４０
∞	９３	∞
合計	１６４	－

表１　　ビット幅とデコード完了命令数

この結果より、先頭から８ビットをインデックスとしてデコードした場合は、全命令フォーマットの８分の１しかデコードが完了しない、インデックスを２ビット増して１０ビットをインデックスとした場合、約２倍の命令のデコードが完了するが、さらにインデックスを２ビット増やしても６命令しか増えないので、先頭から１０ビットをインデックスとしてテーブルを引きデコードするのが最適であると判断した。

A Study of High Speed Decode of Simulator
Kawakita Hideyuki　Hirao Shigeharu
TOSHIBA Corporation　Systems & Software Engineering Lab.

２．２　デコーダ部の処理

デコーダ部では、さきに述べた事前評価をもとに、以下に示す方針でデコードを行なっている。

(1)　まず、先頭からの１０ビットをインデックスとしてデコードする。１０ビットでデコード終了したものは、アドレス計算部、実行部へ制御を渡す。
(2)　(1) の処理でデコードできないものは、さらに２ビットを見る。この２ビットを見てデコードを終了したものは、アドレス計算部、実行部へ制御を渡す。
(3)　(2)の処理でデコードできないものは、残った４ビットを見る。この４ビットを見てデコードを終了したものは、アドレス計算部、実行部へ制御を渡す。
(4)　(3) の処理で第２項目があることがわかったものは、第２項目のデータもみてデコードを終了させ、アドレス計算部、実行部へ制御を渡す。

３．　デコードテーブルの評価

今回作成したシミュレータのテーブルのサイズの正当性を評価するため、シミュレータで様々なプログラムを動かしてみて、実行した命令のデコードを完了するために必要なビットフィールドの幅を調べてみた。
今回用いたプログラムは次のようなものである。
(1)　ＱＵＩＣＫソートのＣ言語で記述されたプログラム[4] のアルゴリズムをもとにアセンブルしたもの。今回は１０個の数字をソートした。
(2)　(1) と同じプログラムで、ＴＸ１Ｃコンパイラで作成したもの。
(3)　DHRYSTONE ベンチマークのＣ言語で記述されたプログラムをＴＸ１Ｃコンパイラで作成したもの。

このようなプログラムを実際に動かして実行した命令を調べ、その命令をデコードするために必要な先頭からのビットフィールドの幅を調べたものを表２に示す。

ビット幅	QUICK SORT	QUICK SORT TX1CC(*)	DHRYSTONE TX1CC(*)
8	２１９	２０３	２９６
１０	３７８	１３６	５４９
１２	0	0	0
１６	1	1	2
１７ﾋﾞｯﾄ以上	８１	１６１	１８６
総実行命令数	６７９	５０１	１０３３

(*) TX1CC : ＴＸ１Ｃコンパイラ
表２　実行した命令数とデコードに必要な情報のビット幅

この結果から、(1) のプログラムでは、９０%近くの命令が、(2) のプログムでは、７０%近くの命令が、(3) のプログラムでは、８０%以上の命令がこのデコードテーブルを引くだけでデコードを完了する。この結果から、ほとんどの命令がこのテーブルを引くだけでデコードが完了し、高速にデコードを行なうために有効であることがわかる。

一方、テーブルのサイズを８ビットに縮小した場合、(1) のプログラムでは、３２%の命令、(2) のプログラムでは、４０%の命令、(3)のプログラムでは、２９%の命令しかデコードを完了することが出来ない。このことから、このテーブルでサーチするビットフィールドの幅を８ビットに縮小するとあまり効果が期待できない。一方、見るフィールドの幅を１０ビットより大きくしても効果が期待できないこともわかった。
従って、今回評価に用いたプログラムでは、サーチする際に１０ビットをインデックスとしてテーブルを引く方法が最も効果的であることがわかる。

４．　結論

今回、シミュレータの命令実行のデコーダ部で、オペコードの集中する部分のビットパターンをキーとしてテーブルをサーチする方法を用い、この部分の幅の最適値を求めた。シミュレータを作成する以前に命令のフォーマット分類で評価して、最適であると判断したビットパターンの幅が、実際にプログラムを動かして評価してみて正当であることがわかった。また、このテーブルをサーチするだけでデコードを完了できる命令は、フォーマット別に分類を行なったときよりも実際にプログラムを動かしてみるとはるかに多くなっていることがわかる。
同時に、実際に作成されるプログラムでは、オペコードが命令コードの先頭に集中するデコードの容易な命令フォーマットが多く用いられることがわかった。従って、マイクロプロセッサの持っている全命令フォーマットがデコード可能であるような大きなテーブルは必要ない。

５．　おわりに

今回は、命令デコーダ部の高速化に重点を置いて開発を行なった。今後は、命令実行部の中のアドレス計算部、実行処理部の高速化およびデバッガ部のマンマシンインターフェースの強化に焦点を置き改良を行なっていく。

参考文献

[1]　Ken Sakamura,
"TRON VLSI CPU:Concepts and Architecture"
TRON Project 1987 Springer Verlag, pp.199-238
[2]　Keiji Namimoto, Tai Sato and Akira Kanuma,
"TX Series Based on TRONCHIP Architecture"
TRON Project 1987 Springer Verlag, pp.291-308
[3]　Misao Miyata, Hidechika Kishigami,
Kosei Okamoto, Shigeo Kamiya,
"The TX1 32-Bit Microprocessor:
Performance Analysis, and Debugging Support"
IEEE MICRO Vol.8 No.2 pp.37-46.
[4]　Byte Editorial Staff, "High-Tech Horsepower" ,
BYTE,July,1987,pp101-108.