

(1990. 6. 22)

V80 マイクロプロセッサにおけるキャッシュと分岐予測

山畑 均、 小林 升

211 川崎市中原区下沼部 日本電気 マイクロコンピュータ事業部

V80は独自アーキテクチャによる32ビットVシリーズMPUの最上位に位置する。0.8 μ m CMOSプロセスにより約98万トランジスタを集積し33MHzにて動作する。ビット(フィールド/列)・データや浮動小数点数の操作を命令として持つ他、OSの記述を支援するための命令や仮想記憶管理機能の内蔵により、各種のアプリケーションやOSの記述を容易にしている。高速化のためにパイプラインの改良の他、命令/データ独立のキャッシュと分岐予測機構を持つ。本論文では、新たな評価指標を導入することで、キャッシュと分岐予測との性能に対する貢献度を定量的に比較評価することを試みた。

"An analysis of cache and branch prediction of the V80 microprocessor"

Hitoshi YAMAHATA, Noboru KOBAYASHI

Microcomputer Division, NEC Corporation
Shimonumabe, Nakahara-ku, Kawasaki, 211 JAPAN

This report will describe a single chip 32-bit CMOS VLSI microprocessor V80. It has been implemented by using a double metal-layer CMOS process technology with 0.8 μ m design rule to integrate 980,000 transistors. It has bit(s) manipulating functions, floating-point operations, system management instructions and virtual memory management function for supporting application and system programs. It integrates separate cache memories for instruction and data and the branch prediction unit (BPU) to avoid frequent pipeline purges. This paper will focus on the performance improvement by the cache and the BPU by using a new metrics.

1 はじめに

近年、マイクロプロセッサ(MPU)の性能向上には著しいものがある。特に32ビットMPUでは、パイプライン処理構造やキャッシュの搭載など、大型計算機同様の高速化手法を取入れている。このような背景のもと、我々は、32ビットMPUの独自アーキテクチャを定め、V60、V70、V80という一連のMPUを開発してきた。

V80は11の内部ユニットからなり、1命令を7つのパイプライン・ステージで処理する。64エントリーのトランスレーション・ルックアサイド・バッファ(TLB)を内蔵し、仮想記憶管理をオンチップでサポートする他、IEEE-754準拠の32/64ビット浮動小数点演算命令を持つ。0.8μmルール、2層メタルCMOS技術を用い、14.49×15.47mmのチップ上に約98万トランジスタを集積して33MHzで動作する。V80が採用した高速化方式のうち代表的なものは以下3つである。

- 各段の負荷を均等化した7段パイプライン
- 命令/データ独立な1K+1Kバイトのキャッシュ
- 64エントリーの分岐予測テーブル(BPT)システム・サポートとしては、マルチプロセッサ構成に効果的なバス・モニタリング機能や、高信頼性システム実現のためのFRM(Functional Redundancy Monitor)機能、パリティ・チェック機能を持つ。

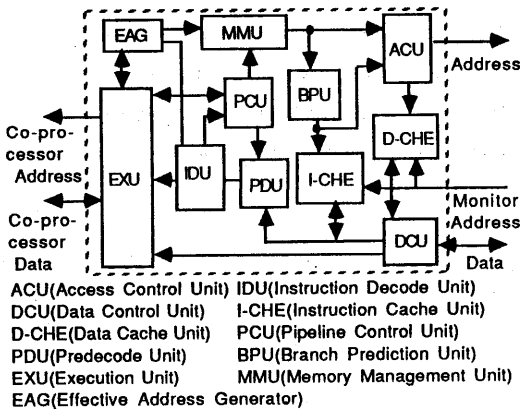


図1 V80の内部ブロック構成

Item	V80
General registers	32-bit * 32
Instructions	119-type
Instruction format	2-operand symmetry
Virtual memory Management	4G-byte Virtual Space 4G-byte Real Space Paging(4K-byte/Page) 4-level Protection
Translation Lookaside Buffer(TLB)	64-entry 2-way set associative hardware replace
Floating-point operation	32/64-bit based on IEEE-754
Pipeline structure	11-unit, 7-stage, execute max 6 instructions
Cache memory	Instruction/Data separate (each 1K-byte, write through)
Branch Prediction	64-entry 2-way set associative
High reliability function	Functional Redundancy Monitor Parity check
No. of transistors	980,000
Process	0.8um double Al CMOS
Chip size	14.49*15.47mm
Package	280-pin PGA
Clock freq.	25/33MHz
Performance	16.5MIPS (peak @33MHz)
Power dissipation	4.75W (@33MHz)

表1 V80の諸元

この論文ではキャッシュとBPTとによる性能向上について焦点を当て、定量的な評価を試みている。新たな尺度としてPIA(ピア、Performance Improvement per Area)を導入して、キャッシュとBPTの面積性能効率を評価した。PIAは、特定の付加機能について、単位面積当りの性能向上を表す指数である。即ちPIAは、その付加機能による性能向上率を、その機能によるチップの面積増加率で割った数値で示される。

結果としては、キャッシュのPIAは1.05で、BPTのPIAは1.03であった。この結果によれば、V80でのインプリメントの場合におけるキャッシュと分岐予測の面積当り性能効率はほぼ同じで、キャッシュの方が若干良いことを示している。

図1にV80の内部ブロック構成図を、表1に諸元を示す。

2 V80のアーキテクチャ

V80アーキテクチャの特徴を列挙すると、

- 32本の汎用レジスタ
- 21種類のアドレッシング・モード
- ビット/ビット・フィールド/ビット列、浮動小

数点数を含む豊富なデータ・タイプ・サポート

○デマンドページングにより管理される4 Gバイトのニアアドレス空間

○各種データ・タイプに対する操作や関数呼出/戻り、OS処理をサポートする豊富な命令セット

○ハードウェアにより細かく分類され、逐次化される割込み/例外処理

○アドレス比較によるソフトウェア・デバッグ・トラップ

などである。

32本の汎用レジスタは、レジスタ上での倍精度浮動小数点数演算の他、コンパイラによる大域的最適化の実現に十分な余地を提供する。ビット(列)データ処理は、制御/画像処理のアプリケーションにおける構造体データの一部のフィールド処理や、フィールド・メモリに対するBitBLT処理に適する。タスク切替えや、仮想記憶のテーブル参照/更新、アクセス権のチェックなど、OSに典型的な処理をマイクロプログラム化された高機能命令として持つ。OSカーネル部分の記述が容易となり、処理も高速で行なえ、OSのリアルタイム性を向上させる。

割込み、仮想記憶に伴うページフォルト、浮動小数点命令の実行で発生する演算例外などの各種例外の同時発生に対してもハードウェアで逐次化が行なえられる。即ち、例外スタック上に優先順位の低い例外から、例外情報(例外種別コード、プログラム・ステータス語、PCなど)を積み、最後にスタックに積んだ(最も優先順位の高い)例外ハンドラに処理を移す。優先順位の高い例外ハンドラが終了するとハンドラからの戻り先は次の優先度の例外ハンドラになっている。例外ハンドラを記述する際には、個々の例外ごとに独立して記述できる。また、ページフォルト処理中の例外スタックのページフォルトのような場合でも、自動的に割込みスタックを用いてページフォルト例外ハンドラ起動される。OS設計者はリエントラントにさえ気を付ければ、複雑なOSも容易に記述できる。

3. バイプライン

パイプライン構成により同時に複数の命令を処理することは、今日では、ほとんどのMPUにおいて行なわれている。パイプラインがスムーズに流れるようにすることが最も性能向上に重要な点である。キャッシュも分岐予測機構もバレル・シフタのような高度な演算器も、すべてパイプラインの流れをできるだけスムーズにするために他ならない。キャッシュや分岐予測機構の他にもV80ではパイプライン・フローの改善のための対策を行なっている。

RISC型プロセッサに比べてV80は複雑な命令コード体系を持つため、加算や転送などの基本的命令で比べると、命令デコードの処理負担が大きい。V80ではプリデコードと命令デコードとの二つのパイプライン・ステージを設けた。プリデコード部分で命令とアドレッシング・フィールドの整列/切出しを行なうため、命令デコード部分ではRISC型のように固定長情報を扱えばよくなり、デコード部分の負担を軽減している。

V80の命令実行部分はマイクロプログラム制御方式であるため、命令デコードが完了しても、実行開始までにマイクロROMのアクセス時間を必要とする。パイプラインが連続的に流れている間は、このアクセス時間はパイプライン的に無視できる。しかし、フラグ・ハザードなどの発生によりパイプラインがシーケンシャルになると、一つの命令がパイプラインに入ってから出るまでのスループットが重要になる。V80では、基本的な命令についてマイクロROMを使用せずに、直接に命令実行を開始することで、パイプラインの流れを改善している。

V80では、仮想アドレスからのアドレス変換を高速化するためのTLBを、V70での16エントリから64エントリに増加させると共に、TLBミス時のテーブル参照をV70でのマイクロプログラム制御から、完全ハードウェア制御に変えて命令実行と平行に高速(6~11クロック)で処理している。

4. キャッシュ

キャッシュの構成を決定するに当たっては、複数のアプリケーション・プログラムから採取した実行時

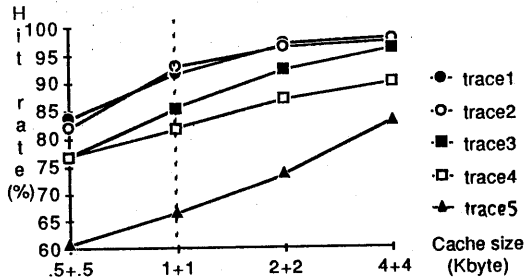


図2 キャッシュ・サイズとヒット率

トレース・データを用い、モデルを変えながらシミュレーションを行なった。図2には代表的なシミュレーション結果からキャッシュのサイズとヒット率の関係を示す。

V80は命令キャッシュ(I-CHE)とデータ・キャッシュ(D-CHE)とを独立して持つことで、異なるパイプライン・ステージからの命令/データの同時アクセスを可能としている。命令/データの各キャッシュはそれぞれ1Kバイトの容量があり、16バイトのブロックからなる2ウェイ・セットアソシアティブ方式である。データ・キャッシュはライト・アロケート機能を持ち、ライトスルー方式をとる。さらに、専用の端子を用いたバス・モニタリング機能により、マルチプロセサ・システム構成時のキャッシュのコヒーレンスを容易に維持できる。また、アプリケーションの種類やシステム構成によっては、フレームバッファなどのように主記憶上の特定領域をキャッシング禁止にする必要があるため、V80ではページテーブル上にキャッシング禁止指定ビットを設けている。

命令キャッシュとデータ・キャッシュは、合わせて約156,000トランジスタからなり、総チップ面積の4.4%を占める。両方のキャッシュを使用することで、キャッシュ未使用時に比べてドライストン・ベンチマークの実行が10%高速化される。従ってPIAは、 $1.10 / 1.044 = 1.05$ になる。試みに、命令/データそれぞれに2Kバイトづつのキャッシュを持つモデルをシミュレーションしたところ、キャッシュ未使用時に比べて15%の性能向上が得られた。キャッシュの容量を倍にす

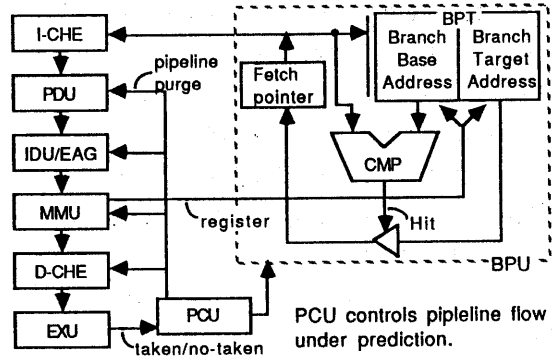


図3 分岐予測機構のブロック図

ることで、おおよそ倍の面積が必要になるとすると、この場合のPIAは、 $1.15 / 1.088 = 1.06$ になる。

5 分岐予測

図3に構造を示した分岐予測機構は、分岐命令によって起こされるパイプラインのバジを避けるために採用された。32ビットVシリーズ・アーキテクチャでは、一般的なアプリケーションにおける分岐命令の出現頻度は、3~5命令に1回である。それゆえ、分岐命令の処理を円滑に行なうことは高性能MPUを実現する上で重要な問題である。

V80では分岐した分岐命令の分岐先をテーブルに登録して、再度、同一の分岐命令が出現した場合の分岐先を予測している。このことは、「分岐しなかった分岐命令の分岐先はテーブルに登録しない」ということによって分岐しない方向についても予測を行なっていることになる。すなわち、限られたテーブルの容量を生かして実質的に分岐/非分岐の双方向の分岐予測を実現している。

分岐予測は以下の手順で行なわれる。

- 1) 初めて出現した分岐命令を命令デコード・ユニット(IDU)が検出して、その分岐命令自身のアドレスと分岐先のアドレスと分岐命令の命令長などを分岐予測ユニット(BPU)内部の分岐予測テーブル(BPT)に登録する。
- 2) BPUは、命令フェッチのアドレスとBPT内に登録されている分岐命令アドレスとを毎フ

フェッチごとに連想比較する。

3) BPT内に登録されている分岐命令アドレスが命令フェッチのアドレスと一致すると、BPUは分岐命令自身の命令フェッチの完了を待って、命令フェッチ先をBPT内に登録されていた分岐先に切替えると共に、予測した分岐先に切替えたことをIDUに通知する。

4) 以後は予測した分岐先に関する処理(命令デコード、オペランド・アドレス計算、オペランド・リードなど)を、予測命令ストリームであることを示すフラグと共に先行処理する。

分岐命令が実行されて初めて、分岐予測の成否が確定する。分岐すると予測した分岐先へ確かに分岐した場合(予測成功)には、パイプライン上の遅れなく分岐命令を処理する。分岐しなかった場合(予測失敗)には、予測された命令ストリームに付随する先行処理は予測命令ストリームを示すフラグを基にバージされ、IDUにて、分岐しなかった場合の命令ストリームを再フェッチする。表2には、いくつかの分岐命令について、予測時と未予測時の所用クロック数を示す。BPTは約40,000トランジスタからなり、総チップ面積の2.0%を占める。分岐予測は、主に、ループ構造をなす分岐を多数含むプログラムにおいて効果を発揮する。分岐予測を使用することで、未使用時に比べてドライストン・ベンチマークの実行が5%高速化される。従ってPIAは、 $1.05/1.020=1.03$ になる。

6 結論

表3に示すように、V80においては、PIAを尺度とした場合にキャッシュと分岐予測の面積性能効率はほぼ同じで、キャッシュの方が分岐予測より

Instruction	Prediction	
	HIT	MISS HIT
JMP always	2	--
Branch on condition	2	7
Decrement and branch on condition	2	7
Test and Branch	2	7

表2 分岐所用クロック数

	performance improvement	die size increase	PIA
BPT	5%	2.0%	1.03
1K+1K cache	10%	4.4%	1.05
2K+2K cache (simulation model)	15%	8.8%	1.06

表3 PIA表

も若干効果的であるという結論が得られた。ドライストーンのような簡易なベンチマークの他、幾つかのアプリケーション・プログラムにおいても同様の傾向の結果を得ている。

キャッシュや分岐予測の方式によってPIAの値は当然変化するものであるし、プロセッサのパイプライン構成やアーキテクチャによっても、同一のキャッシュ/分岐予測が性能に与える影響は異なるはずである。さらには、特異な構造を持つプログラムにおいてはPIAの値が大きく変化するであろう。例えば、分岐は多く発生するが、IF文のような前方参照が多く、ループの少ないプログラムの場合は分岐予測の効果が低くなる。しかしながら、PIAのような定量的尺度を導入して、基本的なデザインを評価することは、他のMPUでの機能評価においても重要であると考えられる。今後は評価尺度の改良により、キャッシュと分岐予測の他の高速化機構についても、それぞれの実現方式を越えて比較評価できる方法を探っていきたい。

7 謝辞

本発表の機会を与えて頂きました可児事業部長、藤高事業部長代理に感謝致します。

【参考文献】

[1] 金子 ほか：キャッシュと分岐予測機構の内蔵などでパイプラインの乱れを抑えて性能を上げた32ビット・マイクロプロセッサV80、日経エレクトロニクス、p.141-152. June, 28, 1989

[2] 鈴木 ほか：7段パイプライン構造を採用した32b、33MHz、マイクロプロセッサ、電子情報通信学会研究報告、ICD-89-12. April, 21, 1989