

汎用プロセッサの性能試験方法の検討

2C-1

名古屋 彰

NTT電気通信研究所

1. はじめに

プロセッサの性能試験の方法として、典型的なプログラム走行によるベンチマーク試験と、性能を決定づける要素をプログラムによって個々に測定し、その積み上げによって平均命令実行時間を求める方法がある。特に後者の方法は、各要素が性能に与える影響を明確にすることから、ハードウェア、ファームウェアの設計評価のための有効な手段となる。しかし、大型プロセッサのように高速化のための多数の機構を有している場合、後者の測定を正確に行うためには、他の機構の影響をすべて排除した上で、特定要素の測定を行う必要があり、ハードウェア構造に対応して異なる方法を用いなければならない。本稿では、最近の汎用大型プロセッサのハードウェア構造に注目し、後者の性能試験を実施するための各要素の測定方法を提案する。

2. 性能を決定する要素

プロセッサの高速化を図るための手段として、従来より、命令のパイプライン制御、キャッシュメモリ、アドレス変換バッファ(ATB)の内蔵等の手段が用いられている。最近の汎用大型プロセッサでは、さらに以下の方式に発展している。

- (1) キャッシュメモリの2階層化(以下では、命令プロセッサに近い方からLM(ローカルメモリ)、BM(バッファメモリ)と呼ぶ)
- (2) パイプラインのオーバラップ処理を向上させるため、LMおよびATBの命令用(I-LM)/オペランド用(O-LM)への分離
- (3) 分岐によるパイプラインの乱れを減らすため、分岐予測テーブルによる分岐先アドレスの予測

一般に、平均命令実行時間(g)は以下のように表すことができる。

$$g = g_0 + \Delta T_1 + \Delta T_2 + \Delta T_3$$

ここで、 g_0 は、命令、オペランドが各LMにすべて存在し、かつATBがヒットし、パイプラインの乱れが無い場合の平均実行時間であり、想定する処理種別(リアルタイム系、科学技術計算系等)に基づく命令MIXによって、平均を求めたものである。 ΔT_1 は所要の命令、オペランドがLMに存在しない場合の遅れ、 ΔT_2 はATBミスヒットによる遅れ、 ΔT_3 はパイプラインの乱れによる遅れをそれぞれ1命令当たりの時間で表したものである。

前述のような高速化方式を採用したプロセッサの例では、 ΔT_{1-3} を表1~3で表すことができる。この中で表2に示すパラメータは、想定する処理種別とプロセッサのハードウェア構成に依存するものであり、命令トレースデータを用いたシミュレーション等によって値を定める。表3に示す項目が、各事象が発生した場合の実際の遅れ時間である。

以下では、 g_0 の環境下での各命令の実行時間、および表3に示す各項目の測定方法を示す。

表1 平均命令実行時間における ΔT_1 、 ΔT_2 、 ΔT_3 の算出式

項目	算出式	説明
ΔT_1	$\Delta T_{11} + \Delta T_{10} + (a_{10} + a_{01}) \cdot (r_{bms} + r_{bmcs})$	LMヒットによる遅れ
	$\Delta T_{11} \quad (n_{10} - a_{11}) \cdot r_{bms} + a_{11} \cdot r_{msl} + a_{10} \cdot r_{msl} + a_{11} \cdot r_{msl}$	1-LMへのアクセスによる遅れ
	$\Delta T_{10} \quad (a_{00} - a_{01}) \cdot r_{bms} + a_{01} \cdot r_{msl} + a_{00} \cdot r_{msl} + a_{01} \cdot r_{msl}$	0-LMへのアクセスによる遅れ
	r_{bms} これらは、1命令当たりの1/2回数、BMからの1/2回数、BM/PT	ATB状態によるBM/PT待ち
	r_{bmcs} 1/2回(4等分)待ちの遅延に基づいて計算する。(ここでは、計算式を省略する。)	1-LM/PT/PTによるBM/PT待ち
	r_{msl} $(1-m) \cdot r_{msl0} + (1-m) \cdot r_{msl1}$	MEMからのアクセスによる遅れ
	r_{msl0} $(1-m) \cdot r_{msl0} + (1-m) \cdot r_{msl1}$	
ΔT_2	$b \cdot (E_0 \cdot r_{tc0} + E_1 \cdot r_{tc1} + E_2 \cdot r_{tc2})$	ATBヒットによる遅れ
ΔT_3	$E \cdot C_1 \cdot \Delta I + \Delta I_{branch}$	パイプラインの乱れによる遅れ
	$\Delta I_{branch} \quad E \cdot w_j \cdot (a_j \cdot K_{1j} + \beta_j \cdot K_{2j})$	分岐予測失敗による遅れ

表2 平均命令実行時間算出のためのパラメータ

パラメータ	説明
a_{10}	1命令当たりの命令要因によるアクセス回数(BM→LM)回数
a_{00}	1命令当たりのオペランド要因によるアクセス回数(BM→LM)回数
a_{11}	1命令当たりの命令要因によるアクセス回数(MEM→LM)回数
a_{01}	1命令当たりのオペランド要因によるアクセス回数(MEM→LM)回数
m	リフレッシュされるBMのアクセスが書き換えられている割合
b	1命令当たりのATB変換情報がない割合
E_0	ST、PTともLMにある割合
E_1	STがLMにあり、PTがBMにある割合
E_2	STがLMにあり、PTがMEMにある割合
C_1	パイプラインの乱れとなる各事象(レジスタコッパリ等)の1命令当たりの発生頻度
w_j	分岐予測失敗の要因となる分岐命令の発生頻度
a_j	対象分岐命令における分岐方向の予測失敗の発生頻度
β_j	対象分岐命令における分岐先アドレス予測失敗の発生頻度

表3 ハードウェア要因による遅れ

遅れ要素	説明
r_{bms}	BMへの命令リードアクセスによる時間増分
r_{bmcs}	BMへのオペランドリードアクセスによる時間増分
r_{msl0}	MEMへの命令リードアクセスによる時間増分(注1)
r_{msl1}	MEMへのオペランドリードアクセスによる時間増分(注2)
r_{msl0}	MEMへの命令リードアクセスによる時間増分(注1)
r_{msl1}	MEMへのオペランドリードアクセスによる時間増分(注2)
r_{msl}	アクセス時の転送遅れによる時間増(命令)
r_{msl}	アクセス時の転送遅れによる時間増(オペランド)
r_{tc0}	ATBミスヒット時の時間増(ST、PTともLM上)
r_{tc1}	ATBミスヒット時の時間増(STはLM上、PTはBM上)
r_{tc2}	ATBミスヒット時の時間増(STはLM上、PTはMEM上)
ΔI	パイプラインの乱れとなる各事象(レジスタコッパリ等)の発生による時間増
K_{1j}	分岐方向の予測失敗による時間増
K_{2j}	分岐先アドレス予測失敗による時間増

(注1) リフレッシュされるBM上のアクセスは書き換えされていない
(注2) リフレッシュされるBM上のアクセスは書き換えされている

3. 測定方法

(1) g_0 の測定

測定命令を数1000回以上繰り返し実行し、 μsec 単位のタイマ(例えばインタバルタイマ)で測定することにより、1命令当たりの実行時間をnsec以下の精度で求めることができる。ただし、測定中に、LM、ATBのミスヒットが生ずることがないように、測定プログラム、およびオペランドは、限定された測定エリア内(例えば連続する2ページ内)のみで実行される等の考慮が必要である。

繰り返し実行する部分には、対象とする命令の他に、対象命令の前後でパイプライン制御のインタロックが生じないように埋め込むダミー命令、対象命令のオペランドを用意する命令、繰り返しを制御する命令等を含める必要がある。この場合対象命令だけの実行時間の抽出は、①対象命令を含む命令シーケンス(P_H)の実行時間と、②同一条件で、対象命令だけを含まない命令シーケンス、または対象命令のみを他命令(同一命令長で実行時間が既知)に置き換えた命令シー

ケンス (P_{HL}) による実行時間とを測定し、①と②の差を求めることで可能である。

測定開始時において、I-LM、O-LM、ATBおよび、分岐予測テーブル等の条件を整えるためには、タイマをスタートさせる直前で、繰り返し実行部分を1度ダミー実行させる。なお、各測定項目において、ダミー実行開始から測定終了までの期間は、完全に測定エリア内だけで動作するように (SVC命令等も含めて) アドレスを管理する必要がある。しかし、命令シーケンス発生時や、測定の後処理の期間はこの制限は不要である。

(2) ΔT_{1-3} の測定

g_0 における各命令の実行時間の測定方法は、そのまま表3の各項目の測定においても適用できる。この場合、対象とする遅れ事象を発生させる手法が論点となる。

(a) ΔT_1

τ_{bms} 、 τ_{mms} の測定の際には、対象とする命令、オペランドがそれぞれ I-LM、O-LM に存在する場合、LM には無く BM に存在する場合、LM、BM のいずれにも存在しない場合のそれぞれの環境を、連続して繰り返し測定する中で作り出す必要がある。そのためには、I-LM、O-LM の分離、LM と BM の構成 (ブロックサイズ、行、カラム) の差異等を利用して、表4のような手法を用いることができる。

なお、ブロックの追い出しのためには、測定エリア外の、多数のページをアクセスすることになるが、この場合、測定エリア内のプログラム、ST、PT との間で影響がないよう

表4 τ_{bms} 、 τ_{mms} 測定時の環境設定方法

ハードウェア構造		設定環境	
		LMに無くBMに存在	LM、BMに存在せず
(最小構成機種) BMカラム数	(最大構成機種) LMカラム数	LMから追い出す際にアクセスする同一カラムがBMでは異なるカラムとなるようにアドレスを指定し、BMに残す	LM、BMともそれぞれ同一カラムとなるようなアドレスを連続してアクセスすることにより両者から追い出す
上記の条件を満足しない	I-LMとO-LMに分離している	右欄の方法で該当LM、BMから追い出した後、該当LMとは反対のアクセス (命令用ならばアドレスアクセス) でBMへロードする	LM、BMともそれぞれ同一カラムとなるようなアドレスを連続してアクセスすることにより両者から追い出す
	LMは分離していない	LM、BMへロードした後LMをオーガナイズする命令を使用する (プログラムエリア、各種テーブル等も追い出されてしまうので注意が必要)	LM、BMともそれぞれ同一カラムとなるようなアドレスを連続してアクセスすることにより両者から追い出す

表5 ブロック転送の転送ずれによる遅延測定法

測定項目	PM (対象とする遅れ発生)	PM (対象とする遅れ発生せず)
τ_{mcs1}		
τ_{mcs0}		

にアドレスを考慮しなければならない。

また、同一シリーズで性能の異なる機種を実現する場合、LM、BMの容量を増減する手段がよく用いられる。どの機種でも同一の測定プログラムで試験を行うことができるような配慮も必要である。(例えば、ブロックの追い出し部分は、最大容量機種に対応していれば、下位機種でもそのまま利用できる)

τ_{mcs} については、表5のような方法で測定できる。

(b) ΔT_2

τ_{10} の各項目の測定のためには、ST、PTが、LMにある場合、LMには無くBMに存在する場合、LM、BMのいずれにも存在しない場合の環境を作り出し、ATBヒットの場合との差を求める。

(c) ΔT_3

Δ_1 の測定は、 g_0 と同様の手法で測定可能である。

Δ_{branch} では、分岐テーブルによる分岐方向予測の失敗、分岐先アドレス予測の失敗による遅れを、表6の手法で測定できる。なお、測定中の分岐テーブルエントリのクリアは、他エントリの登録により現エントリを追い出す手法で実現できるが、P_HとP_{HL}とで測定対象以外の条件が異なることのないように、細心の注意が必要である。

表6 分岐予測失敗による遅れ時間測定法 (対象とする分岐命令をYとする)

測定項目	PM (対象とする遅れ発生)	PM (対象とする遅れ発生せず)
K1 NOGO予測であるがGOとなる場合		
K1 GO予測であるがNOGOとなる場合		
K2 分岐先アドレス予測失敗の場合		

4. おわりに

上記に示した手法に基づく試験プログラムは、実際に大型汎用プロセッサの性能試験に使用しており、予想通りの精度での測定が実現している。

今後の検討課題としては、性能算出方法も含めて、以下の項目がある。

- ①命令先取りバッファの効果の明確化
- ②直前、直後の命令間以外のパイプライン・コンフリクトの影響の考慮
- ③プログラムからは不可視のメモリアクセス (ハードウェア領域に対するアクセス) による影響の明確化
- ④測定開始時の各種条件の統一の厳密化 (特に分岐予測テーブル)

最後に、日頃御指導頂いている小柳津処理装置研究室長、塩川主幹研究員、および関係諸氏に感謝します。