

## リアルタイムシステムにおけるRISC性能解析の精度検証

杉村 康

NTT情報通信研究所

筆者は既にRTS(リアルタイムシステム)におけるRISC性能解析手法等を提案した[1]。本報告ではMIPS-R4000相当のRISCを前提として、ドライストーンベンチマーク[2],[3]等をソフトウェアによりエミュレーションを行うことによりアクセス状態データ[1]を得て、キャッシュおよびTLB等の動作のシミュレーションを行うことにより、それらのシミュレーション結果と実測結果との比較を実現する。また、それらの手法により、文献[1]のシミュレーションの精度を検証し、「文献[1]で述べた実験中のRTS」でのTLBミスによる性能低下率等を明らかにする。

### A verification of Analyzing Accuracy of RISC Performance in a Real Time System

Yasushi Sugimura

NTT Information and Communication Systems Labs.

I have already proposed a method of analyzing RISC performance in a real-time system[1]. This paper presents an emulation method and a simulation method etc. in the case of MIPS R4000, and discusses several analysis results in detail by comparing simulation results with actual measurement results in Dhrystone Benchmarks[2,3]etc.. The results clarify the analyzing accuracy of the above method and the rate of performance decrease caused by TLB miss in a Real Time System etc..

#### [1. はじめに]

筆者は、RISC (Reduced Instruction Set Computer) の性能を予測するための性能解析手法等を既に提案した[1]。そこではMIPS-R4000等のように単一バイプラインを有するRISCを前提とし、その場合下記(1)が成り立ち、「CISC(Complex Instruction Set Computer)に比したRISCの処理能力」を決定する主要因は、下記(A)~(C)の三つであることを明らかにした。

(1) バイプラインのストール[4],[5]の発生がない理想的な状況下では、[すなわちキャッシュミスが起こらず、TLB (Translation Look aside Buffer[6])ミスが起こらず、フランチも発生せず、データのロード遅延等も発生しない状況下では] 1秒当りの命令の実行数を表す指標であるMIPS (Million Instructions Per Second)値は、CPUのバイプラインのステージ周波数をMHzで表したものの(以下PSTFと略記)に等しい。

(A) RISC/CISC-DS比(以下RCDSRと略記):CISCの命令をRISCの命令に置き換えた場合に、RISCのDS(ダイナミックステップ数)がCISCのDSの何倍になるかを表す。

(B) PSTF: 上記(1)。

(C) 各種ペナルティ: キャッシュミスやフランチ遅延やロード遅延等により上記の理想的な状態に比べて余分に必要な時間。(以下「この時間をバイプラインのステージクロック数で表したものを」をEAMPとよぶ。)

文献[1]では、RCDSRは実測とほぼ一致することを示したが、キャッシュ等のシミュレーションの精度の検証は今後の課題であった。その検証には、実測結果とシミュレーション結果との比較が必要である。

その検討において実現しなければならない主な項目には以下のものがある。

(D) アクセス状態データ(文献[1]表3)取得手法。

(E) 評価モデル。

(F) TLBミスのオーバヘッド等の反映手法。

これらの内、キャッシュ等のシミュレーションに不可

欠である上記(D)アクセス状態データ取得手法については、次の三つの手法が考えられる。

(a) CPUのデバッグ割込み機能(1命令毎に割込みを起こす機能)等を利用して取得する。[1]、

(b) ハードウェアモニタを使用して取得する。[7]

(c) ロードモジュールを入力としてエミュレーションにより取得する。

本報告では、下記により、上記(c)の手法をとる。

・MIPS-R4000相当のRISC(以下R4000と略記)には上記(a)の機能はない。

・上記(b)の手法は、実用マシンのCPU装置にハードウェアモニタの接続等が今回は許されなかった。

上記(c)の手法は、一般にロードモジュールの形式がマニュアルとして公開されている[8]ので可能である。しかしRISCでは、システムコールの制御の受渡し処理がカーネルで実現されており、且つ、その部分は今回はブラックボックスであるので、その部分のエミュレーションができない。従って、評価に使用するベンチマークは、測定ループ(時間の測定対象となる部分)内にシステムコールの発行がないものでなければならない。

当報告では、上記の「測定ループ内にシステムコールの発行が無」という条件を満たすドライストーンベンチマークを原型として使用することにより、上記(c)を実現する。

上記(E)の評価行為は、次の3点を満たす必要がある。

(d) モデルの妥当性: モデルは、実際のRTS全体のソフトウェアモデルに近いものでなければならない。

(e) モデルの同一性: 実測とシミュレーション等の両者で使用するベンチマークプログラム(以下ベンチマークと略記)は、同一でなければならない。

(f) エミュレーションと実測の正常性: 実測とエミュレーションで、ベンチマークの同一ロジックが正しく動作していることの確認ができればならない。

本報告では、ドライストーンベンチマークを原型とし

て、実際のRTS全体におけるキャッシュのヒット率等を近似した新たなベンチマーク（以下疑似RTSベンチマークと略記）を実現することにより、モデルの妥当性を保証する。また、本報告では、ドライストーンベンチマークに内蔵されている「処理された文字列と数値結果と予想定数との目視比較による正常性確認ロジック」をそのまま流用することによりモデルの同一性と実行ロジックの正常性を保証する。（詳細は文献[2]参照）

上記(F)については、文献[1]では、以下の二つの理由によりTLBミスの影響の評価を省略した。

- ・ R4000では、プログラムをTLBを使用しないエリアに作成が可能である。
- ・ TLBミスの性能への影響は小さいと予想される。

しかしながら評価精度を検証するためには、その影響の評価が不可欠である。ところが一般にRISCのTLB処理はカーネルで実現されており[6],[9]、それらはブラックボックスであり公開されていない。一方、TLBミスの回数は、TLBの構成とアクセス状態が明らかであれば、ソフトウェアによりシミュレーションが可能である。またTLBミスのオーバーヘッドを含んだ全処理時間は実測が可能である。従って、TLBミスの回数のみが異なり、その他の性能低下の要因が非常に少ないようなベンチマークを実現し、そのシミュレーション結果と実測結果を求めれば、下式によりTLBミスの回数とTLB処理のオーバーヘッド時間との関係を求めることができる。

$$T_{i1b-i} = (T_{real-i} - T_{sim-i}) \div N_{i1b-i} \quad (1)$$

但し、 $N_{i1b-i}$  : シミュレーションによるモデルiのTLBミスの回数。

$T_{i1b-i}$  : モデルiの1回当りのTLB処理時間。

$T_{real-i}$  : モデルiのベンチマーク実測走行時間。

$T_{sim-i}$  : モデルiのシミュレーションより求めたベンチマーク走行時間（TLB処理を含んでいない）。

当報告では、上記の目的で、ドライストーンベンチマークを原型とした、簡単な三つのベンチマーク（以下TLBミス測定用ベンチマークと略記）により、あらかじめTLBミスの回数とTLBミスのオーバーヘッド時間との関係を求める。そしてそれによる推定値( $T_{i1b-sr}$  : 単位 $\mu s$ )と「疑似RTSベンチマークのシミュレーションによるTLBミス無し予想走行時間( $T_{sim-sr}$ ; TLB処理時間は含まず)」とを、下記式(2)等により予想走行時間( $TR_{sim-sr}$ )に反映し、それと「疑似RTSベンチマークの実測走行時間」とを比較することにより、文献[1]のキャッシュ等のシミュレーションの精度を検証する。

$$TR_{sim-sr} = T_{sim-sr} + T_{i1b-sr} \quad (2)$$

$$T_{sim-sr} = (DS_{sim-sr} + \Sigma EAMP_{sim-sr}) \div (PSTF * 1000^{*1}) \quad (3)$$

但し \*1:  $\mu s$  への換算値。

$DS_{sim-sr}$  : 疑似RTSベンチマークのDS。

$\Sigma EAMP_{sim-sr}$  : 疑似RTSベンチマークのEAMP合計。

なお当評価手法での評価対象は、オンラインRTSを実現する小型計算機システムを想定し、オンライン中には

必要なデータやプログラムはすべてメモリ上に存在する。

【2. ベナルティの用語の定義と前提条件等】

R4000等のRISCのベナルティの用語を表1に概説する。（詳細は文献[4],[5],[10]参照）以下、各ベナルティは表1の略号で記述する。R4000の主な前提条件の仮定を表2に示す。なお、ページテーブルの構成等についての前提条件は、説明の都合上、3.2で述べる。

【3. 評価手法】

3.1 アクセス状態データ生成手法

「ベンチマークの論理アドレスが解決済」のR4000用のロードモジュールを入力して、ロードモジュール内の

表1 R4000のベナルティの用語

No	遅延名	略号	説明
1	ロード	LOADDL	レジスタにロードしたデータを直後の命令で参照できないときの遅延。
2	フランチ	BRANDL	フランチ命令の実行での遅延。
3	部分ワード ライト	PWRDL	3バイト以下のデータのライトの命令での遅延。
4	メモリド	MEMRDL	キャッシュミス時の行データのメモリ→CPU転送時の遅延。(CPU/メモリ間のシステムインタフェース転送遅延を含む。)
5	一次リード	REAI DL	二次キャッシュミス時に余分に行われる一次キャッシュのリードによる遅延。
6	一次ライト	WRIIDL	キャッシュミス時に余分に行われる一次キャッシュのライトによる遅延。
7	二次リード	REA2DL	キャッシュミス時に余分に行われる二次キャッシュのリードによる遅延。
8	二次ライト	WRI2DL	キャッシュミス時に余分に行われる二次キャッシュのライトによる遅延。
9	ライト・リード 競合	CFWRDL	CPU→メモリデータライト・リード連続時にライトが完了するまでの遅延。
10	ライトシステム インタフェース	WSYIDL	CPUからメモリへのライト時のシステムインタフェース転送時間遅延。
11	乗除算	MUDI DL	固定小数点乗・除算の演算遅延。

表2 R4000での主な前提条件

No	項目	前提条件
1	キャッシュ 関連	ライトバック方式、1次キャッシュ/2次キャッシュ有、1次キャッシュは命令/データ独立、2次キャッシュ命令/データ共用、1次キャッシュ/2次キャッシュのバス幅=16バイト、1次命令・データキャッシュ各8KB、2次キャッシュ1MB、1次キャッシュ行長16バイト、2次キャッシュ行長32バイト、1/2次キャッシュのリフレッシュ(置換え)は行単位、メモリは内部にリードバッファとライトバッファを各1個所有、CPU/メモリバッファ間の転送時間はバス幅とデータ長に依存、1次キャッシュ論理アドレスライトマッピング、2次キャッシュ物理アドレスライトマッピング、32ビットアドレス
2	TLB	エントリ数48エントリ、1エントリ2ページマッピング、論理アドレスフルアソシエイティブ、32ビットアドレス
3	その他	ステータス周波数=150MHz、メモリアクセス時間=100ns、ecのオーダ、2次キャッシュのブロックアクセス時間=10nsecのオーダの各一点に固定、ワイトレジスタの値=8。

エントリーポイントからベンチマーク内の命令を一つづつソフトウェアによりエミュレーションし、表3のアクセス状態データを生成する手法を、ここではアクセス状態データ生成手法とよぶ。そのエミュレーションでは、以下を実現する。

(1) 論理空間のエミュレート：以下によりUNIX上でRTSの論理アドレスをエミュレートする。

(A) ロードモジュール内のプログラムとデータをUNIX上の空間にロードする際に、その先頭UNIXアドレスと先頭RISCアドレスと、ロードエリアの大きさの一覧表を作り、命令フェッチやロード命令やストア命令のエミュレート時に、上記一覧表により、RISCアドレスをUNIXアドレス変換してアクセスする。

(2) 物理アドレス：物理アドレスの情報は、ベンチマークのロードモジュール内には存在しないので、表4のNo7～No8のようにパラメータとして与える。

(3) システムコールのシミュレート：測定ループの外の正常性確認等のためのRTS用システムコールのシミュレートは、表4のNo5～6のように、その外部名等をパラメータで指定し、エミュレータがロードモジュール内にある外部名の情報によりシステムコールの言語バインド部のルーチンのRISCアドレスに変換し、ジャンプ(JAL)命令等のエミュレート時にUNIXアドレス上でUNIXシステムコールの実行にすり替える。

### 3. 2 TLBのシミュレート手法

R4000におけるTLBミスは、小型のRTSを前提とした場合、次の二つに分類できる。

(1) UTLBミス：TLBミスが発生した場合「TLBミス専用のベクターテーブルが使用される」TLBミスを、ここではUTLBミスとよぶ。専用ベクターテーブルなので、そのオーバーヘッドは少ない。(文献[5]p.118表6-3 "TLB不一致 EXL=0"に対応)

(2) PTLBミス：TLBミスの処理中に別のTLBミスが発生した場合(PT(ページテーブル)[6]の参照でミスが発生した場合)のTLBミス(汎用のベクターテーブルが使用される)を、ここではPTLBミスとよぶ。汎用のベクターテーブルは複数の例外で共用されるため、PTLBミスであることの判定等に余分なオーバーヘッドが必要(文献[5]p.118表6-3 "その他"に対応)なお文献[5]の「TLB無効例外(p.123)」や「TLB変更例外(p.124)」は、以下を前提とし、考慮しない。

・OS(オペレーティングシステム)は、タスクディスパッチ時にTLBを無効する必要はない[ASID(アドレス空間ID)[5]さえ正確に反映しておけばTLB不一致例外で処理が可能]ので、後述のようなホットスタートでの評価では、TLB無効例外は発生しない。

・TLBの書込不可ヒットはODP(オンデマンドページング)等のメモリ負荷制御で使用するものであり、RTSでは、通常オンラインサービス中にODP等は動作不要なように設計するので、該ヒットは通常ON。一方TLBの1エントリーでポイントされる2ページのエリアは、8MB

表3 RISC7k7c状態データ の主な内容

No	項目	説明
1	論理アドレス	命令/データフェッチの論理アドレス
2	物理アドレス	命令/データフェッチの物理アドレス
3	フェッチ種別	命令フェッチかデータフェッチかを示す
4	命令モック名	命令モック名(文献[1]の手法では英小文字、当手法では英大文字)
5	リード/ライト種別	データリードフェッチかデータライトフェッチかを示す
6	フェッチ長	データフェッチ長
7	ワシタ情報	ワシタ番号毎の参照更新有無を示す
8	タスクID	R4000のASID(アドレス空間ID)に対応

表4 RISC7k7c状態データ生成手法の入力パラメータ

No	項目名	説明
1	ロードモジュールファイル名	アドレス解決済ロードモジュールのファイル名
2	ベンチマーク起動パラメータ	ベンチマークに与える起動パラメータ
3	トレース開始アドレス	測定ループの先頭命令の論理アドレス
4	トレース終了アドレス	測定ループ終了時の命令の論理アドレス
4'	エミュレート終了アドレス	エミュレートを終了する命令の論理アドレス
5	システムコール名a	トシメ化するシステムコール名
5'	システムコールリターン値	トシメ化するシステムコールのリターン値
6	システムコール名b	UNIXのシステムコールにシミュレーションされるシステムコール名
7	論理アドレスc	物理アドレスに対応する先頭論理アドレス
7'	頁数c	論理アドレスcより連続して昇順に与えられる物理アドレスのページ数
7''	物理アドレスc	頁数cの先頭物理アドレス
8	論理アドレスd	物理アドレスに対応する先頭論理アドレス
8'	頁数d	論理アドレスcより連続して逆順に与えられる物理アドレスのページ数
8''	物理アドレスd	頁数cの先頭物理アドレス

分のPTE(PTエントリ)を格納でき(詳細文献[6]の4.1)、またワイヤードレジスタの値は8であるので、PTEが置かれるエリアをTLBのエントリ0～7の8個で処理するように設計しておけば、64MB以内を使用する限り、オーバーヘッドが大きいPTLBミスは殆ど発生しない。さらに後述のようにホットスタートを前提する場合、タスクの切替えが存在する場合にのみPTLBミスが発生する。

なお64MBは前提である小型システムにおいては十分な大きさである。

### 3. 3 疑似RTSベンチマーク等の作成手法

#### 3. 3. 1 TLBミス測定用ベンチマークの作成手法

ドライストーンベンチマークのキャッシュヒット率はほぼ100%である。[1] TLBミス測定用ベンチマークは、キャッシュのヒット率をなるべく低下させず、TLBのヒット率を低下させた「ドライストーンベンチマークの変形」により実現する。それらは参照するデータ等を散在させた三つのベンチマークからなる。具体的には、表5のNo1のようなコーディングを同表No2のように変更する。すると、該コーディングの次にある変数へのアクセスは、配列Aとは異なるTLBエントリを使用することとなる。このような変更の数を減らすことにより

表5 TLB のヒット率を低下させるための手法の例

No	項目	コーディングの内容
1	変更前	long int A[31];
2	変更後	long int A[2048];

表6 TLB ミスオーバヘッド測定用ベンチマークの特性

No	項目↓	ワル名→	TLB-A	TLB-B	TLB-C
1	1次命令キャッシュヒット率[%]		98.5	98.5	98.7
2	1次データキャッシュヒット率[%]		89.3	88.0	88.4
3	1次平均キャッシュヒット率[%]		96.3	95.9	96.2
4	2次キャッシュヒット率 [%]		100.0	100.0	100.0
5	1命令当りTLB ミス回数(NOP命令含まず)	①	0.007211	0.017668	0.028491
			②	③	
6	TLB ミスの回数相対比		1.00	2.45	3.95

表7 実システムのRTS と疑似RTS ベンチマークの特性

No	項目↓	ワル名→	(A) 実RTS	(B) 疑似RTS	(C) 相違
1	1次命令キャッシュヒット率[%]		85.2	85.1	-0.1
2	1次データキャッシュヒット率[%]		91.8	92.9	1.2
3	1次平均キャッシュヒット率[%]		87.8	87.2	-0.7
4	2次キャッシュヒット率 [%]		96.8	96.6	-0.2

TLB のヒット率の低下を変化させる。シミュレーションにより求めた「作成したTLB ミス測定用ベンチマークの特性」の概要を表6 に示す。該表より、キャッシュのヒット率等は殆ど同じだが、TLB の一命令当りのミス回数、それぞれおよそ2.5 倍と4.0 倍になった三つのTLB ミス測定用ベンチマークを実現できたことが判る。

### 3. 3. 2 疑似RTS ベンチマークの作成手法

ドライストーンベンチマークは、文献[3] で述べられているように、システムプログラムの特性を良く考えて作成されたベンチマークであるが、RTS のキャッシュのヒット率を反映していない[1]。実システム上で稼働中である文献[1] のRTS のトレースデータを、前記2. の前提条件の構成で、文献[1] のキャッシュシミュレータにより求めた諸特性を表7 の(A) に示す。それらを疑似RTS ベンチマークは、以下の(1)、(2) により疑似する。

(1) 各ルーチン間にタミールルーチンを挿入し、実行されるルーチンの相対位置を調整することにより、1次命令キャッシュのヒット率を調整する。

(2) 新たなルーチンを追加し、測定ループ内で交互に実行させ、そのデータ参照位置を調整することにより、1次データキャッシュのヒット率と2次キャッシュのヒット率を調整する。

これらにより表7 の(B)、(C) に示すように、実RTS の1次キャッシュと2次キャッシュの両特性に近い疑似ベンチマークを実現した。

## 【4. 評価結果】

### 4. 1 評価における共通事項

実測環境等の都合により、以下の前提をおく。

(1) スタート状態：複数回の連続したベンチマークの測定データの内、2 回目以降のデータのみを使用することにより、ホットスタート[1] での評価のみを行う。

表8 ドライストーンベンチマークの実測結果とシミュレーション結果

No	分類	測定回数	千回ループ 平均実行時間	備考
1	実測値	19	10,971,666 [ns]	
2	〃	19	15,125,024 [ns]	データ削除
3	〃	19	10,972,106 [ns]	
4	〃	19	10,969,997 [ns]	
5	〃	19	10,964,969 [ns]	
6	〃	19	10,973,866 [ns]	
7	実測平均値		10,970,525 [ns]	
8	シミュレーション結果		10,960,120 [ns]	
9	誤差:(No8-No7)÷No7 ×100 =		-0.095[%]	

(2) 使用タイマー：当評価でのOSのタイマーの分解能は100ns のオーダである。

(3) 実測時の定期走行タイマー処理等の除去：当報告での実測環境は、FDDI (光ループ) に接続されている複数のR4000 の内の1 台を間借りして行なわさるを得なかったため、ベンチマークとは関係のない割込みが、頻度は低いが不定期に発生し得る。また定期走行タイマー処理等の停止のためのOSの変更も許されなかったため、実測環境のOSの「割込み処理の最初と最後等でタイムスタンプを蓄積する機能」等を使用して、該割込み処理時間等を測定結果よりすべて差し引く。

また、4. 4の疑似RTS ベンチマークの評価においては、ベンチマークの測定ループを終了した直後にコアダンプを収集させ、表4のNo7 ~No8の入力パラメータを自動生成するプログラムの開発により、ベンチマークの各走行時の物理メモリの割当状況を正確に反映した。

### 4. 2 ドライストーンベンチマークの実測結果とシミュレーション結果との相違

表8 にドライストーンベンチマークの全実測結果とシミュレーション結果を示す。該表のNo1 ~No6 より、No2 の実測結果のみが、他の実測結果と異なることが判る。この相違は、No2 においてのみ「キャッシュが割込み処理等により大きな擾乱を受けた」ためと考えられるので、No2 の測定結果だけを削除して求めた平均値(該表No7)を、実測結果値として採用する。それとシミュレーション結果(該表No8)との相違は0.095[%](該表No9)しかなく、シミュレーションと実測結果は非常に良く一致することが判る。

上記のキャッシュ擾乱要因には以下が考えられる。

- ・光ループ接続された他の計算機からの割込み処理
- ・定期走行タイマー割込み処理
- ・タイムスタンプ情報の格納処理

以下、上記の擾乱をまとめて「割込み等による擾乱」と略記する。なお、文献[1] と上記結果の間で、シミュレータに以下の修正を行った。

(1) 新文献[5] のp. 32の表2-11のDIV, MULT命令等のベンチマーク量を反映(増加)。

(2) NOP 命令検出時の割込みカウンタの誤減算を修正  
上記(2) は、文献[1] の4. 2. 2(4)のツールの使用時に

表9 ドライストーンベンチマークの特性例の修正版

No	項目	新	旧	
1	一次命令キャッシュ ヒット率	100.0 [%]	100.0 [%]	
2	一次データキャッシュ ヒット率	99.4 [%]	99.4 [%]	
3	二次キャッシュ 全ヒット率	100.0 [%]	100.0 [%]	
4	RISC平均命令実行時間	0.0184μ秒	0.0172μ秒	
5	キャッシュ	ブランチ	39.6 [%]	47.6 [%]
6	等を除く	ロード	40.9 [%]	29.0 [%]
7	く	乗算・除算	12.0 [%]	9.1 [%]
8	イ	部分ワードライト	6.2	7.3 [%]
9	ドライストーン 値[回/秒]	64,892.9	71,806.4	
10	CMIPS (RCDSCR=1.69時)	32.2	34.4	
11	文献[15]表1と上記No9から求めたCMIPS 値	34.7	38.4	
12	No10の誤差 (対No11)	-7.2[%]	-10.4[%]	
13	ドライストーンの予想RCDSCR	1.57	1.51	

表10 TLB ミス測定用ベンチマークの実測結果の全一覽

No.	ベンチマーク名	測定日	測定回数①	千回ループの平均実測走行時間[ns]②	対全平均値誤差[%]③
1	TLB-A	A-1	19	36,818,696	-0.05
2		A-2	17	36,836,058	-0.01
3		A-3	19	38,257,401	3.85
4		B-1	19	36,871,225	0.09
5		C-1	9	36,840,708	0.01
6		D-1	19	40,492,660	9.92
7		E-1	19	36,827,203	-0.03
8		E-2	19	42,471,302	15.29
		全平均値④		36,838,778 (No3, 6, 8除く)	
9	TLB-B	A-1	19	42,007,888	0.43
10		A-2	19	41,790,306	-0.09
11		A-3	19	41,865,925	0.09
12		C-1	19	41,625,306	-0.48
13		D-1	19	41,847,475	0.05
		全平均値⑤		41,827,330	
14	TLB-C	A-1	19	53,047,917	0.04
15		B-1	19	52,967,490	-0.11
16		C-1	19	53,054,944	0.05
17		C-2	19	53,123,222	0.18
18		D-1	19	53,086,083	0.11
19		D-2	19	52,883,595	-0.27
		全平均値⑥		53,027,209	

のみに通るロジックの修正であり、上記(1)は、他のワル量に比べるとわずかであるので、文献[1]の評価結果の大勢には、影響を与えないが、ペナルティの絶対量が少ない「文献[1] 4.2.2(4)の評価結果」は、影響を受けるので、その修正版を表9に示す。

#### 4.3 TLBミスのオーバヘッドの評価結果

表6に示した三つのTLB ミスオーバヘッド測定用ベンチマーク(TLB-A, TLB-B, TLB-C)の実測結果のすべての一覽を表10に示す。それらのデータの内、No3とNo6とNo8(表中の網掛部分)だけが表中③の欄に示すように誤

表11 TLB ミスのオーバヘッドの実測値一覽

No	ベンチマーク名	オーバヘッド相対値	誤差[%]
1	TLB-A	1.015	1.5
2	TLB-B	0.977	-2.3
3	TLB-C	1.010	1.0
	平均値	1.000	0.0

差が異常に大きいので、該データは、割込み等による大きな擾乱を受けていると見なして、全平均値(表中の④)の算出から除外した。その他のデータは全平均値(表中の④、⑤、⑥)に比して、0.5[%]未満の相違しかなく、各測定値の精度は高いと考えられるので、表中の④、⑤、⑥の全平均値を、それぞれのベンチマークの実測値として採用する。それらと同じベンチマークをシミュレーションし、式(1)によりTLB ミス1回当たりの処理時間( $T_{11b-1}$ )を求めた結果の一覽を表11に示す。値は平均値に対する相対値である。該表よりTLB ミスの頻度を大きく変えても $T_{11b-1}$ は数%の違ひしか無いことが判る。従って後述の4.4の評価での $T_{11b-1}$ は、それらの平均値を用いる。なお $T_{11b-1}$ の絶対値はステージクロック換算で、文献[9]3.2の値の6割程度である。

#### 4.4 疑似RTS ベンチマークの予想走行時間と実測結果との相違の評価

表7に示した特性を持つ疑似RTS ベンチマークの実測走行時間と、「同一のベンチマークのシミュレーション結果( $T_{sim-1}$ )と上記4.2より求めた $T_{11b-1}$ 」により式(2)で求めた予想走行時間( $TR_{sim-1}$ )の全一覽をそれぞれ、表12の①と②に示す。同表の測定日の先頭一文字が同一のものは一回のベンチマークの走行の中で「連続して測定された」データである。ベンチマークの複数回の連続測定で取得出来たデータの個数が連続測定当り1(例; 同表No1)~5(例; 同表No17~21)と大きく変動している原因は、割込み処理等による擾乱にある。論理アドレス割当状況は同一であるので、TLB ミスの回数は同一であり、同表②のシミュレーション結果より、物理メモリの割当状況もベンチマーク走行時間に影響を与えていることが判る。同表③より、各実測値とシミュレーション結果との相違には、およそ7.7~35.4[%]の変動が見られる。しかし上記4.2と4.3による実測結果より、実測結果が異常にシミュレーション結果と異なる場合は、すべて実測結果がシミュレーション値より異常に大きくなっているため、最も割込み処理等による擾乱が小さいと考えられる「相違の最低値7.73[%](表12No7の③)」が最もシミュレーションの状態に近いと考えてよい。また表12の④に示すように、「相違の最低値7.73[%]」の状態でも、なお割込みによる擾乱を含んでいると考えられるので、2次キャッシュのミスを含んだ疑似RTSモデルでのシミュレーションの実質的誤差は、7.7%をかなり下回ると考えられる。

#### 4.5 RTS処理能力に対するTLBミスの影響の例

文献[1]で述べた実際のRTSのトレースデータから得られたアクセス状態データを、上記2章で述べた前提の

元で、当手法によりシミュレーションした結果のTLBミスによるオーバヘッドを表13に示す。該表中、No4のPTLBミスのオーバヘッドについては一回当りの正確な値

表12 疑似RIS ベンチマークの実測結果等の全一覧

No	測定日	①千回ループ 実測走行 時間[ns]	②(μ) 状況 反映シミュレ ーション結果[ns]	③誤差 [%]	④ 97マ 以外の割 込み回数
1	A-1	315,329,437	284,579,441	10.81	2
2	B-1	309,508,484	280,812,976	10.22	2
3	B-2	329,806,011		17.45	5
4	B-3	329,739,262		17.42	2
5	C-1	329,298,633	287,217,041	14.65	2
6	C-2	329,375,113		14.68	2
7	C-3	309,432,844		7.73	2
8	E-1	297,162,564	273,291,641	8.73	2
9	E-2	317,077,255		16.02	2
10	E-3	317,293,633		16.10	5
11	E-4	297,064,324		8.70	1
12	F-1	363,991,667	276,491,176	31.65	2
13	F-2	374,433,950		35.42	10
14	G-1	339,294,473	290,527,741	16.79	2
15	G-2	319,277,702		9.90	2
16	G-3	329,344,073		13.36	2
17	H-1	317,310,638	273,404,911	16.06	5
18	H-2	327,215,787		19.68	7
19	H-3	327,378,478		19.74	5
20	H-4	306,952,442		12.27	2
21	H-5	316,928,384		15.92	1
22	I-1	308,763,178	280,571,176	10.05	5
23	I-2	328,384,880		17.04	5
24	I-3	318,559,920		13.54	2
25	I-4	308,522,291		9.96	2
26	I-5	308,642,218		10.00	5
27	J-1	323,156,306	273,404,911	18.20	2
28	J-2	312,967,826		14.47	1
29	K-1	362,672,094	272,451,711	33.11	6
30	K-2	362,515,352		33.06	5
31	K-3	362,313,883		32.98	5
32	L-1	332,606,911	276,004,911	20.51	4
33	L-2	322,441,733		16.82	3
34	M-1	329,611,202	291,393,811	13.12	2
35	M-2	339,398,082		16.47	2
36	M-3	329,674,489		13.14	5
37	N-1	314,304,075	276,144,711	13.82	2
38	N-2	324,452,482		17.49	4
39	O-1	334,498,860	286,817,376	16.62	2
40	O-2	334,475,049		16.62	2
41	P-1	297,387,924	275,551,376	7.92	5
42	P-2	327,063,837		18.69	2
43	P-3	306,993,117		11.41	2
44	P-4	317,189,582		15.11	7

は不明である。当RTSにおいては、性能最優先に設計されており、PTLBミスは汎用のベクターテーブルがポイントするプログラムの先頭で判定されることを前提として、「PTLBミスによるオーバヘッドはJTLBの2倍である」と仮定して、表13のように計上した。該表No5より、TLBミスのオーバヘッドは、該オーバヘッドが無い場合の全CPU時間のおよそ5.4%である。

表13 RTSのTLBミスのオーバヘッドの評価結果

No	項目	結果
1	シミュレーション	TLBミス回数 501 [回]
2	結果	PTLBミス回数 18 [回]
3	全CPU時間に	TLBミスオーバヘッド 4.99 [%]
4	対する割合	PTLBミスオーバヘッド 0.36 [%]
5		合計 5.35 [%]

### 【5. むすび】

上記で示したように、当手法の実現により下記を明らかにすることができ、当初の目的を十分に果たした。

- ・ 文献[1]で述べたキャッシュのシミュレート時の評価精度は、かなり高い。(ドライストーンベンチマークでの誤差は0.1%未満、RTS疑似ベンチマークでの誤差は7.7%よりかなり下回る。)

- ・ TLBミスの1回当りのオーバヘッドは、その頻度にあまり関係なく一定に近く、文献[9]で使用されている値のおおよそ6割程度である。

- ・ 実験中の文献[1]のRTSでのTLBミスによるオーバヘッドは、およそ5.4%である。

### 文 献

- [1]Y. Sugimura, "Analysis of RISC Performance in Real Time System," Systems and Computers in Japan, vol. 27, no. 8, pp. 1-18, Jul. 1996. [2]R. P. Weicker, "Dhrystone Benchmark: Rationale for Version 2 and Measurement Rules," SIGPlan Notices, vol. 23, no. 8, pp. 49-62, Aug. 1988. [3]R. P. Weicker, "Dhrystone: A Synthetic Systems Programming Benchmark," Commun. ACM, vol. 27, no. 10, pp. 1013-1030, Oct. 1984. [4]MIPS Computer Systems Inc., "MIPS R4000 Microprocessor User's Manual," pp. 1-593, 1991. [5]NEC日本電気株式会社, "VR4400TM 64ビットマイクロプロセッサアーキテクチャ編(第4版)," pp. 1-420, 1995. [6]R. Uhlig, D. Nagle, T. Stanley, T. Mudge, S. Sechrest, R. Brown, "Design Tradeoffs for Software-Managed TLBs," ACM Trans. Comput. Syst., vol. 12, no. 3, pp. 175-205, 1994. [7]T. Horikawa, "TOPAZ: Hardware-Tracer Based Computer Performance Measurement and Evaluation System," NEC Res. & Develop., vol. 33, no. 4, pp. 638-647, Oct. 1992. [8]Integrated Device Technology, "R3000 FAMILY ASSEMBLY PROGRAMMER'S GUIDE," IDT, Aug. 1988. [9]J. B. Chen, A. Borg, N. P. Jouppi, "A Simulation Based Study of TLB Performance," The 19th Annual Symposium on Comput. Architecture, IEEE, pp. 114-123, 1992. [10]NEC日本電気株式会社, "VR4400TM 64ビットマイクロプロセッサハードウェア編," pp. 1-134, 1993.