計算機アーキテクチャ 124－2
（1997. 6. 27）

TLB-Assisted Cache

鈴木健一[†] 大庭信之[††]
小林広明[†††] 中村維男[†††]

オンチップキャッシュでは，キャッシュのアクセス時間がプロセッサの動作周波数の制約となるため，アクセス時間の短縮は重要である．そこで，本報告では，TLB Assisted Cache (TAC) を提案する．TAC は，一般に TLB のアクセス時間がキャッシュのアクセス時間よりも短いことに着目し，TLB のアクセス結果をキャッシュのヒット/ミス判定にも利用する．これにより，キャッシュタグの比較が完了するよりも前に，キャッシュデータアレイへのアクセスが可能となり，論理的には V-P キャッシュでありながら，V-V キャッシュに匹敵するアクセス時間を実現できる．しかも，論理的には V-P キャッシュとして扱えるため，V-V キャッシュのかかえる様々な欠点を無視することができる．

TLB-Assisted Cache

KEN-ICHI SUZUKI,[†] NOBUYUKI OBA ,[††] HIROAKI KOBAYASHI[†††]
and TADAO NAKAMURA[†††]

This report proposes a new on-chip cache system named "TLB Assisted Cache (TAC)." The TAC determines a cache hit/miss by referring to the TLB and the small assist tag comparisons that are faster than a conventional cache tag comparison. Therefore, it is possible to initiate a cache data array access before a cache tag comparison. Consequently, the TAC achieves an access time as short as a V-V cache. Moreover, the TAC logically acts as a V-P cache so it does not suffer from the V-V cache's shortcomings, such as the synonym problem.

1. はじめに

近年の半導体技術の進歩により，マイクロプロセッサの高速化と高集積化が進んでいる．そして，マイクロプロセッサの動作周波数を向上させるには，オンチップキャッシュのヒット時のアクセス時間を短縮することが非常に重要である．なぜなら，パイプライン化されたマイクロプロセッサでは，オンチップキャッシュが1プロセッササイクルの間にデータを供給することが期待されており，キャッシュヒット時のアクセス時間が動作周波数の足枷となるからである．

本報告では，既存の TLB(Translation Lookaside Buffer) とオンチップキャッシュのアクセス過程が並行して行なわれていること[1)2)] に着目し，TLB へのアクセス結果を基にキャッシュのアクセス時間を短縮する TLB-Assisted Cache (TAC) を提案する．TAC では，一般に TLB のアクセス時間はキャッシュのアクセス時間よりも短いことを利用し，TLB のヒット時には，キャッシュタグ比較の完了を待たずにキャッシュデータアレイへのアクセスを開始する．この結果，TAC では，TLB ヒット/キャッシュヒット時のアクセス時間が V-V (Virtually-indexed Virtually-tagged) キャッシュとほぼ同一の V-P (Virtually-indexed Physically-tagged) キャッシュを実現することができる．

実際には，TAC が V-V キャッシュと同等のアクセス時間を発揮できるのは，TLB とキャッシュの両方がヒットした場合に限られるが，TLB のヒット率は非常に高い (多くの場合 97%以上) ことと，参照の局所性によりキャッシュヒット時には TLB にもヒットしていることが期待されることから，ほとんどのキャッシュデータアレイへのアクセスは，キャッシュタグ比較の完了前に開始できる．しかも，論理的には V-P キャッシュなので，V-V キャッシュに見られるシノニム (エリアス) の問題[3)]は発生しない．**表1** に TAC と従来方式との比較を示す．

最近，キャッシュのアクセス時間短縮のために，Juan ら[4)]は，アソシアティビティの小さいキャッシュにつ

† 宮城工業高等専門学校情報デザイン学科
Department of Design and Computer Applications, Miyagi National College of Technology
†† 日本アイ・ビー・エム株式会社東京基礎研究所
IBM Research, Tokyo Research Laboratory, IBM Japan Ltd.
††† 東北大学大学院情報科学研究科
Department of Computer and Mathematical Sciences, Graduate School of Information Sciences, Tohoku University

表 1 TAC と従来の方式との比較
Table 1 Conventional caches and the TAC

	V-V	V-P	TAC
Access time(TLB hit)	fast	slow	fast
Access time(TLB miss)	fast	slow	slow
Synonym	yes	no	no

いて，バンク (ウェイ) を選択するのに必要な情報を格納する小さなタグを用意することで，キャッシュタグ比較の完了前にデータのマルチプレクシングを開始する差分ビットキャッシュを提案した．これにより，アクセス時間を 30%程度短縮できるとしている．

2. 従来のアドレス変換機構とキャッシュ

仮想記憶機構を採用しているプロセッサでは，主記憶参照の際に，仮想アドレスから実アドレスへのアドレス変換が必要である[2]．アドレス変換は，主記憶上に置かれるアドレス変換表を参照することにより行なわれるが，これでは主記憶参照のたびに追加の主記憶参照を必要とするので，処理のボトルネックとなる．

そこで，アドレス変換を高速に行なうために，多くのアーキテクチャでは TLB を設けている．これはアドレス変換表専用のキャッシュメモリであり，アドレス変換表を参照して行なわれたアドレス変換の結果をキャッシュしておくことにより，同じ変換を次回に行なうときには，主記憶を参照することなくアドレス変換を完了することができる[1][2]．

キャッシュメモリへのアクセスは次のようにして行なわれる．

(1) アドレスの中位のビットを使って，一つのキャッシュセットを選択する (インデキシング)．
(2) 選択されたキャッシュセットについて，アドレスの上位ビットをそれぞれのアドレスタグと比較する (タグ比較)．

それぞれの操作を仮想アドレスで行なうか，実アドレスで行なうかによって，キャッシュメモリは 4 つに分類される．すなわち，V-V，V-P，P-V，P-P の 4 種類である[5]．ただし，P-V キャッシュはほとんど例が見られない[5]ことから，本報告では触れないことにする．

P-P (Physically-indexed Physically-tagged) キャッシュは，キャッシュアクセスを全て実アドレスだけで行なう．P-P キャッシュでは，アドレス変換を完了してからでなければキャッシュアクセスを開始できないため，参照に要する時間は，アドレス変換に要する時間とキャッシュのアクセス時間の和となる．

一方，V-V キャッシュでは，キャッシュアクセスが全て仮想アドレスで行われる．このため，アドレス変換を開始すると同時にキャッシュアクセスを開始でき，キャッシュヒット時にはアドレス変換のオーバヘッドを隠すことができる．しかしながら，V-V キャッシュには次のような問題点がある．

- 複数の仮想アドレスが 1 つの実アドレスにマップされた場合に，それを検出できないため，別々のキャッシュラインに格納してしまい，一貫性を保つことが困難となる (シノニムの問題[3])．
- プロセッサ以外のユニットが主記憶を書き変えた場合，その書き変えをキャッシュに反映させるために，何らかの機構が必要となる．これには，例えば，実アドレスから仮想アドレスへの逆変換機構[6]や，プロセッサ以外のユニットが主記憶に書き込みを行なった場合にキャッシュを全て無効化する機構などが考えられる．

これらの欠点から，V-V キャッシュは，ヒット時のアクセス時間を大きく短縮できるという長所を持ちながら，広く使用されるには至っていない．

V-P キャッシュは，インデキシングだけを仮想アドレスで行ない，タグ比較を実アドレスで行なう．**図 1** は，V-P キャッシュの構成を TLB とともに示している．この構成では，アドレス変換が，仮想ページ番号から実ページ番号への変換として表されている．この変換を行なうのが TLB である．TLB によるアドレス変換と並行に，仮想アドレスの下位ビットを使用してキャッシュのインデキシングが行われる．

V-P キャッシュでは，キャッシュのインデキシングとアドレス変換を並列に行なうことにより，V-V キャッシュには及ばないものの，P-P キャッシュよりも短いアクセス時間を実現できる．また，V-V キャッシュの欠点である，外部ユニットによる主記憶書き変え時の問題が発生しない点が，V-V キャッシュよりも有利である[5]．一方，V-P キャッシュにおけるシノニム問題は，キャッシュのインデキシングに用いるビットがアドレス変換の前後で変わるか，変わらないか，によって以下のように場合分けされる．

(1) インデキシングビットがアドレス変換の前後で変わらない場合: この場合にはシノニム問題は発生しない．
(2) インデキシングビットがアドレス変換の前後で変わる場合: シノニム問題は，V-V キャッシュよりも軽減されるが発生する．

(2) の手法では，シノニムを扱うための特別なハードウェアを必要とするため[5]あまり用いられず，多くのマイクロプロセッサでは，(1) の方式の V-P キャッシュを採用している[7][8][9]．(1) の条件を満たすためには，キャッシュのインデキシングに用いるビットをページオフセットの範囲に収めるようにするか，ページ番号の下位ビット (図 1 の網かけした部分) がアドレス変換後も変わらないようにページマッピングを工夫する必要がある (Colored Cache[10])．

TAC は，(1) のタイプの V-P キャッシュのアクセス時間を V-V キャッシュと同程度まで高速化することを目的としている．次節では，TAC の詳細を述べる．

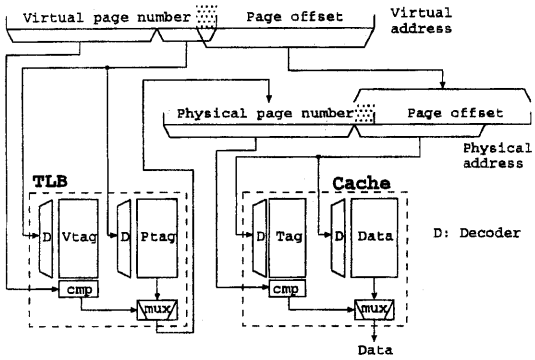

図 1 通常の TLB と V-P キャッシュの概念図
Fig. 1 A conventional TLB and a V-P cache

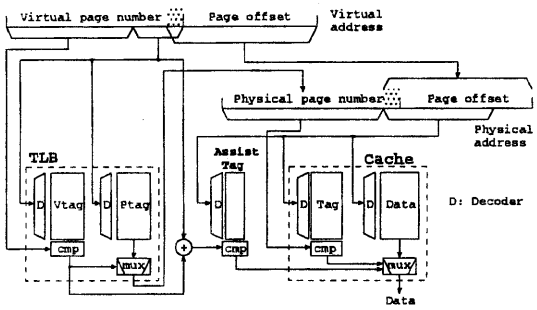

図 2 TAC の概念図
Fig. 2 TLB-Assisted Cache (TAC)

3. TLB-Assisted Cache (TAC)

3.1 概　　要

キャッシュのタグ比較には，アドレスの上位ビットが必要である．これは，V-P キャッシュでは，実ページ番号の上位ビットである．ところが，図 1 からも分かるように，これらのビットは TLB の実アドレスタグ (Ptag) に格納されている．よって，TLB の実アドレスタグとキャッシュタグの関係を表すテーブルを用意することで，キャッシュタグによるヒット/ミス判定の前に，キャッシュヒットを検出できる可能性がある．

図 2 は TLB-Assisted Cache (TAC) の概念図である．従来の V-P キャッシュの構成に加えて，Assist タグが用意されているのが特徴である．Assist タグは，小容量のタグメモリであり，キャッシュタグと TLB の実アドレスタグとの関係を保持する．

Assist タグは，キャッシュライン数と等しい数のエントリから成り，各エントリは，それぞれがキャッシュラインの一つ一つに対応している．Assist タグの各エントリは，対応するキャッシュラインのデータが所属する実ページ番号を格納しているページの TLB 上での位置 (TLB エントリ番号) を保持する．例えば，図 3 では，実ページ番号 P に所属するデータがキャッシュされており，このページのアドレス変換情報は TLB の第 n 番目のエントリに格納されているので，このキャッシュラインの Assist タグには n が格納される．

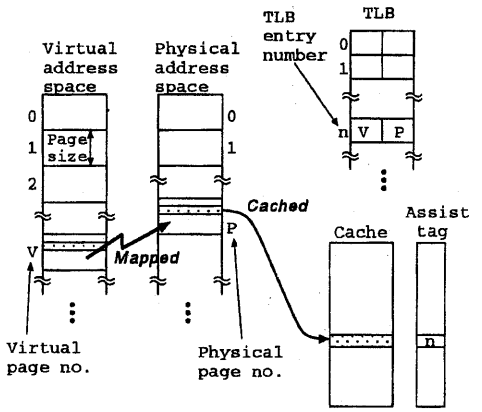

図 3 Assist タグと TLB，キャッシュの関係
Fig. 3 Relationship of Assist tag, TLB, and cache

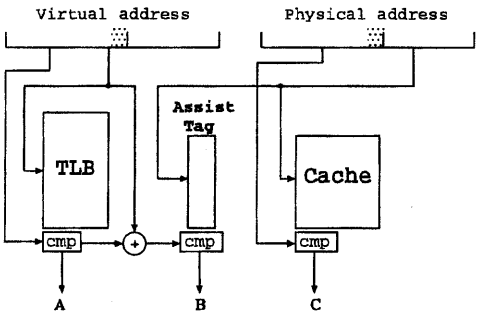

図 4 TAC のキャッシュヒット/ミス判定機構
Fig. 4 Cache hit/miss check mechanism of TAC

次に，TAC のヒット/ミス判定機構を図 4 に沿って述べる．

まず，TLB は，通常と同じように機能する．ヒット/ミス判定結果を A とする．

また，キャッシュも通常の V-P キャッシュと同様に機能する．すなわち，仮想アドレスの下位ビットを使ってインデシキングを行ない，選択されたラインについて，実アドレスによるキャッシュタグ比較の結果から，ヒット/ミス判定を行なう．このヒット/ミス判定結果を C とする．

一方 Assist タグでも，キャッシュのヒット/ミス判定を行なう．この判定は，次のようにして行なわれる．

(1) キャッシュと同一の方法で，インデキシングを行なう．

表 2 各タグの比較結果とキャッシュのヒット/ミスの関係
Table 2 Cache hit/miss determined by tags

A (TLB)	B (Assist)	C (Cache)	Cache status
Hit	Hit	—	Hit
Hit	Miss	—	Miss
Hit	Invalid	Hit	Hit
Hit	Invalid	Miss	Miss
Miss	—	Hit	Hit
Miss	—	Miss	Miss

表 3 本報告で用いる記号
Table 3 Notation in this report

P	ページサイズ
E_{tlb}	TLB エントリ数
A_{tlb}	TLB アソシアティビティ
Col_{tlb}	TLB コラムサイズ ($\frac{E_{tlb} \cdot P}{A_{tlb}}$)
C	キャッシュサイズ
A_{cache}	キャッシュアソシアティビティ
Col_{cache}	キャッシュコラムサイズ ($\frac{C}{A_{cache}}$)

(2) ヒットした TLB 番号と選択された Assist タグの内容を比較する.

比較が一致した場合には Assist タグヒット, 一致しなかった場合は Assist タグミスとなる. その結果を B とする.

各タグの判定結果は, 表 2 に従ってキャッシュのヒット/ミス判定に利用される.

まず, TLB がヒット (A=Hit) の場合には, Assist タグの判定が意味を持つ. この場合, Assist タグが無効 (Invalid) でないかぎり, Assist タグのヒット/ミス (B) により, キャッシュのヒット/ミスを判定できる. ただし, Assist タグが無効 (Invalid) だった場合には, 通常の V-P キャッシュと同様に, キャッシュタグによるヒット/ミス判定 (C) を待たなければならない.

一方, TLB がミス (A=Miss) の場合, Assist タグの判定結果は意味を持たないので, 通常の V-P キャッシュと同様に, キャッシュタグによるヒット/ミス判定 (C) を待たなければならない. しかしながら, 一般に TLB のヒット率は非常に高いので, ほとんどの場合に, Assist タグを使ってキャッシュのヒット/ミスを完了することができる.

3.2 Assist タグの内容

表 3 に本報告で用いる記号を示す.

Assist タグは, キャッシュに格納されているデータと TLB エントリとを結び付ける機能を持っている. Assist タグが保持する情報は, キャッシュと TLB のパラメタ (サイズとアソシアティビティ) によって異なる. 図 5 は, ページサイズ 4KB の場合について, 64 エントリ 4-way の TLB と, 32KB, 2-way のキャッシュの組み合わせを表わしている. 図中で 1 個の箱は, ページサイズ (4KB) 相当の記憶単位を表わしている. すなわち, TLB については 1 エントリが箱 1 個, キャッシュについては 4KB 分の記憶容量が箱 1 個である. このパラメタでは, 1 つのキャッシュラインと結び付く可能性のある TLB エントリは 16 個である. 例えば, 図 5 で網かけをした部分にあるキャッシュライン (4KB 分) は, 同じ網かけをした 16 個の TLB エントリのいずれかとだけ結び付く.

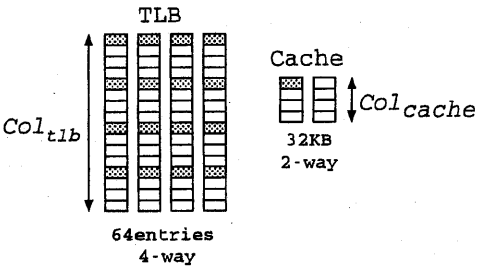

図 5 キャッシュ上での位置と TLB 上での位置の関係
Fig. 5 A position on the cache and a position on the TLB

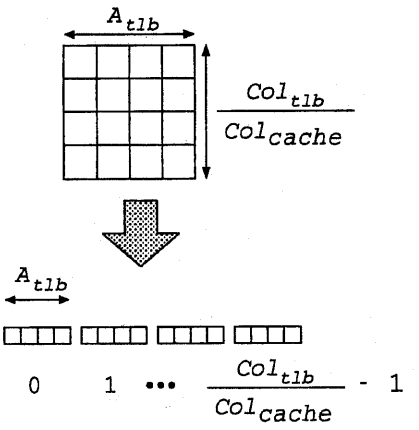

図 6 Assist タグのビット
Fig. 6 An assist tag entry

したがって, Assist タグの各エントリには, 図 6 に示すように 16 ビットが用意される. これらのビットのうち, セットされているビットが, 結びつきのある TLB エントリを指し示す. 複数の仮想ページが一つの実ページにマップされている場合には, これらのビットが複数セットされることも起こり得る. なお, これらのビットは実装上は図 6 の下に示すとおり, 通常のタグと同様, 一列に並べられる.

一般に, Assist タグの各エントリには,

- $Col_{tlb} \geq Col_{cache}$ のとき:
  $A_{tlb} \times \frac{Col_{tlb}}{Col_{cache}}$ [bits]
- $Col_{tlb} < Col_{cache}$ のとき:
  A_{tlb} [bits]

が必要となる.

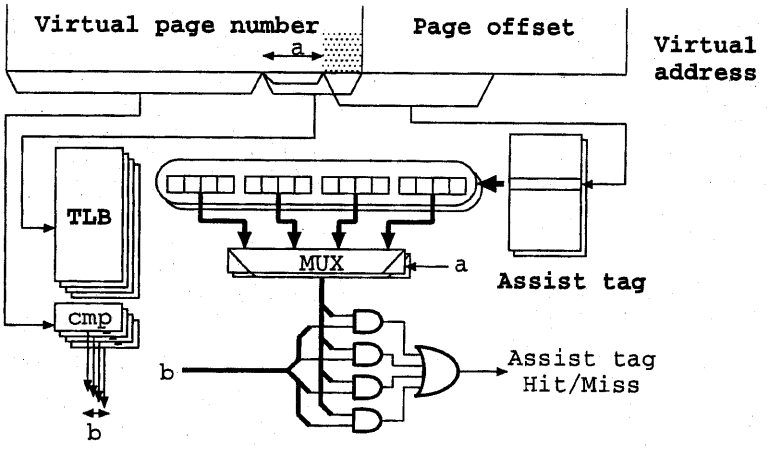

図 7 Assist タグの比較機構
Fig. 7 Assist tag compare mechanism

Assist タグの各ビットは，図 7 にも示すように，

(1) キャッシュのインデキシングと同じビットによりインデキシングを受ける．
(2) キャッシュのインデキシングに使われず，かつ，TLB のインデキシングに使用されたビット (図中の a) により，A_{tlb} ビットが選択される．
(3) ビット毎に，TLB の比較結果との AND を取られる．
(4) その結果の OR を取る．

という手順で利用される．このうち，(2) までの操作は，TLB アクセスと並行して行なうことができるので，TLB の比較終了後に行なわなければならない操作は，(3) と (4) だけである．

なお，Assist タグを実装するのに必要なハードウェア量は，キャッシュと TLB のパラメタにもよるが，キャッシュ1 ラインあたり最大で 32 ビット程度であると予想される．これは，キャッシュアドレスタグとほぼ同程度，32 バイトラインのキャッシュデータアレイの 8 分の 1 程度の量に相当する．今後，ワード長が大きくなるにつれ，キャッシュのラインサイズは増加すると考えられるが，その場合，Assist タグのハードウェアコストがキャッシュシステム全体に占める割合は，縮小することになる．

3.3 アクセス時間の評価

メモリ参照に要する時間は，V-P キャッシュを含むシステムの場合，次の式で与えられる．

$$T_{tlb} \cdot P_{tlb} + T_{addr} \cdot (1 - P_{tlb}) + T_{cache} \cdot P_{cache} + T_{mem} \cdot (1 - P_{cache})$$

ただし，P_{tlb}，P_{cache}は，それぞれ TLB とキャッシュのヒット率，T_{tlb}，T_{cache} は，それぞれ TLB とキャッシュのアクセス時間，T_{addr}と T_{mem}は，それぞれ主記憶上のアドレス変換表とデータへのアクセス時間を表している．多くの場合，P_{tlb}は 97%以上，P_{cache}は 90%以上であることから，T_{tlb}や T_{cache}を短縮することは重要である．特に，オンチップキャッシュの場合には，キャッシュアクセスを 1 プロセッササイクル以内に完了することが必要なため，T_{cache}がプロセッサの動作周波数に大きな影響を与える．

このような観点から，TAC は，TLB，Assist タグの両方がヒットした場合に T_{cache}を短縮することを目標としている．以下では，TLB，Assist タグ，キャッシュ全てがヒットした場合の T_{cache}について検討する．

まず，通常の V-P キャッシュのアクセス時間は，図 1 より，

$$\begin{aligned} T_{VP} &= T_{tlb} + t_{c\_cmp} + t_{c\_mux} \\ &= t_{tlb\_dec} + t_{tlb\_tag} + t_{tlb\_cmp} + t_{tlb\_mux} + t_{c\_cmp} + t_{c\_mux} \end{aligned} \quad (1)$$

となる．一方，V-V キャッシュのアクセス時間は，

$$T_{VV} = t_{c\_dec} + t_{c\_tag} + t_{c\_cmp} + t_{c\_mux} \quad (2)$$

である．

それに対して，TAC のアクセス時間は，図 2 より，

$$\begin{aligned} T_{TAC} &= \max(T_{tlb\_vtag}, T_{assist}) + t_{ass\_cmp} + t_{c\_mux} \\ &= \max((t_{tlb\_dec} + t_{tlb\_tag} + t_{tlb\_cmp}), (t_{ass\_dec} + t_{ass\_tag})) + t_{ass\_cmp} + t_{c\_mux} \end{aligned}$$

となる．デコーダとタグアクセスに要する時間は，ビット数に影響されるので，多くの場合，

$$(t_{tlb\_dec}+t_{tlb\_tag}+t_{tlb\_cmp}) > (t_{ass\_dec}+t_{ass\_tag})$$

である．よって，

$$T_{TAC} = t_{tlb\_dec} + t_{tlb\_tag} + t_{tlb\_cmp} + t_{ass\_cmp} + t_{c\_mux} \quad (3)$$

と書くことができる．

Wilton らは，Alpha プロセッサのオンチップキャッシュ(8KB，0.8μm テクノロジー) について，詳細な理論モデルからアクセス時間を推測する手法を提案し

表 4 アクセス時間算出の条件
Table 4 Condition to estimate access time

Memory space	32bit
Page size	4KB
Number of TLB entries	32
TLB associativity	2
Cache size	32KB
Cache associativity	2
Cache line size	32byte

表 5 アクセス時間の概算値
Table 5 Access time estimations

	V-P	V-V	TAC
TLB decode	1.5	—	1.5
TLB tag	1.0	—	1.0
TLB cmp	2.5	—	2.5
TLB mux	2.0	—	—
Assist cmp	—	—	1.2
Cache decode	—	2.5	—
Cache tag	—	1.6	—
Cache cmp	2.0	2.0	—
Cache mux	2.0	2.0	2.0
Total	11.0	8.1	8.2

ている．そこで，上記の式 (1)〜(3) に Wilton らのモデルを適用して，アクセス時間を概算する．ここでは，キャッシュや TLB のパラメタを表 4 に従って定めた．計算結果を表 5 に示す．

この結果から，TAC は V-V キャッシュに匹敵するアクセス時間を実現できることが分かる．特に，TLB のマルチプレクサを迂回することによる効果が大きい．

しかも TAC は，論理的には V-P キャッシュとして扱うことができるため，V-V キャッシュの欠点であるシノニムの問題や，実アドレスから仮想アドレスへの逆変換の問題は発生せず，従来の V-P キャッシュに代わる高速オンチップキャッシュとして有望であると考えられる．

4. おわりに

最近のマイクロプロセッサの多くは，オンチップキャッシュのアクセス時間によって，その動作周波数を抑えられている．したがって，オンチップキャッシュのアクセス時間の短縮は，計算機の高速化に大きく貢献する．本報告では，TLB とオンチップキャッシュへのアクセスが並行して行なわれることに注目し，TLB-Assisted Cache (TAC) を提案した．

TAC は，従来の V-P Colored キャッシュに Assist タグと呼ばれる小容量のタグメモリとキャッシュのヒット/ミス判定用の小さなハードウェアを追加した構造をしている．Assist タグにはキャッシュの各ラインと TLB エントリの実アドレス上での関係が格納され，この情報を基にキャッシュのヒット/ミスを判定することができる．Assist タグは，小容量であることと，TLB アクセスと並行にアクセスできることから，高速にキャッシュのヒット/ミスを判定できる．また，TAC は，論理的には V-P Colored キャッシュとして扱えるので，V-V キャッシュのかかえる種々の問題を無視することができる．アクセス時間の評価の結果，TLB，Assist タグ，キャッシュタグが全てヒットした場合の TAC のアクセス時間は，V-V キャッシュに匹敵することが明らかとなった．

TLB，キャッシュともにヒット率がかなり高いことを考えると，TAC は V-P キャッシュに代わる高速オンチップキャッシュとして活用できると考えられる．

参考文献

1) A. J. Smith, "Cache Memories," Computing Surveys, Vol. 14, No. 3, pp. 473-530, 1982.
2) D. A. Patterson and J. L. Hennessy, ed., "Computer Architecture A Quantitative Approach," Morgan Kaufmann Publishers, California, 1990.
3) C. Chao, M. Mackey, and B. Sears, "Mach on a virtually addressed cache architecture," USENIX Workshop Proceedings. Mach, pp. 31-51, 1990.
4) T. Juan, T. Lang, and J. J. Navarro, "The Difference-bit Cache," 23th International Symposium on Computer Architecture, pp. 114-120, 1996.
5) C. E. Wu, Y. Hsu, and Y.-H. Liu, "A Quantitative Evaluation of Cache Types for High-Performance Computer Systems," IEEE Transactions on Computers, Vol. 42, No. 10, pp. 1154-1162, 1993.
6) J. R. Goodman, "Coherency for Multiprocessor Virtual Address Caches," Proc. of the 2nd International Conference on Architectural Support for Programming Languages and Operating System (ASPLOS), 22(10), pp. 72-81, 1987.
7) i486 Microprocessor Hardware Reference Manual, Intel, 1990.
8) MC88110 RISC Microprocessor User's Manual, MOTOROLA Inc., 1991.
9) PowerPC 601 RISC Microprocessor User's Manual, MOTOROLA Inc., 1993.
10) B. K. Bray, W. L. Lunch, and M. J. Flynn, "Page Allocation to Reduce Access Time of Physical Caches," Technical Report, Stanford University, Computer Systems Laboratory, Number STAN//CSL-TR-90-454, 1990.