設 計 自 動 化 81－3
(1996. 10. 17)

ASIP 向き階層化メモリシステムの評価

佐藤 淳[1], 武内良典[2], 今井正治[2], 吉岡和樹[2], 塩見彰睦[3]

[1] 鶴岡工業高等専門学校 電気工学科
〒997 山形県鶴岡市大字井岡字沢田 104
[2] 大阪大学大学院 基礎工学研究科 情報数理系専攻
〒560 大阪府豊中市待兼山町 1-3
[3] 静岡大学 情報科学部 情報科学科
〒432 静岡県浜松市城北町 3-5-1
E-Mail: peasv@vlsilab.ics.es.osaka-u.ac.jp

あらまし 半導体の集積化技術の発達により, 大容量の RAM および ROM を内蔵する VLSI の実現が可能になった. 近い将来には, メモリシステムのアーキテクチャが現在よりも重要になると考えられる. 特に, 特定用途向き集積化プロセッサ(ASIP)を組込み応用分野に効果的に適用するためには, コスト効率の良いオンチップメモリシステムが必要不可欠となる. 本稿では, はじめに ASIP に適した階層化メモリシステムを提案する. このシステムは高速キャッシュメモリ, 大容量の内部メモリ(DRAM または SRAM), および低速ではあるが非常に容量の大きい外部メモリから構成される効率的な混合型メモリモデルに基づいている. 次に, 内蔵メモリとキャッシュの間の性能とコストのトレードオフに関する評価実験を行った. 実験の結果より, 従来のキャッシュメモリモデルと比較して, 提案したモデルは性能が数十%程度向上することが確認された.

キーワード メモリシステム, キャッシュメモリ, 特定用途向き集積化プロセッサ

Evaluation of a Hierarchical On-Chip Memory System for ASIPs

Jun Sato[1], Yoshinori Takeuchi[2], Masaharu Imai[2], Kazuki Yoshioka[2], Akichika Shiomi[3]

[1] Department of Electrical Engineering, Tsuruoka National College of Technology
104 Ino-oka, Sawada, Tsuruoka, Yamagata 997
[2] Department of Computer Science, Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka 560
[3] Department of Computer Sciences, Faculty of Information, Shizuoka University
3-5-1 Jyouhoku, Hamamatsu, Shizuoka 432
E-Mail: peasv@vlsilab.ics.es.osaka-u.ac.jp

Abstract
The integration scale of VLSI is steadily getting larger every year. As a result, VLSIs which have on-chip RAM and ROM will become much familiar in the near future. The memory architecture in future will play a more important role than present. Furthermore, in order to adopt the Application Specific Integrated Processor (ASIP) to embedded application domain efficiently, cost effective on-chip memory will become essential. In this paper, a new hierarchical memory system for ASIP is proposed. The feature of this system is an efficient mixed memory model which have on-chip fast cache memory, a large amount of on-chip memory (such as DRAM or SRAM), and a slow but huge off-chip memory. The effectiveness of this model and the performance trade-off between on-chip memory and on-chip cache is shown through simulation experiment. According to the experimental results, the performance can be improved several tens percents compared to a conventional cache memory model.

key words Memory System, Cache Memory, Application Specific Integrated Processor

1. はじめに

VLSI の集積度は年々向上しており，2001 年には，デザインルール，クロック周波数，チップに集積可能なトランジスタ数はそれぞれ 0.18 μm，600MHz，64M トランジスタになると予測されている[1]．さらに，プロセス技術の進歩により，同一のチップ上に CPU コアと DRAM や SRAM のようなメモリを集積することが可能となる[2]．したがって，図 1 に示す CPU コア，メモリ，周辺回路からなる特定用途向き集積化プロセッサ(ASIP)は，チップ内のメモリとして DRAM のような，命令およびデータ用の大容量メモリデバイスを実装することが可能となる．大容量のオンチップメモリを使うことにより，スループットの大きい内部データ転送が可能になり，また外部メモリとのアクセスの減少に共なう電力削減が可能になる．

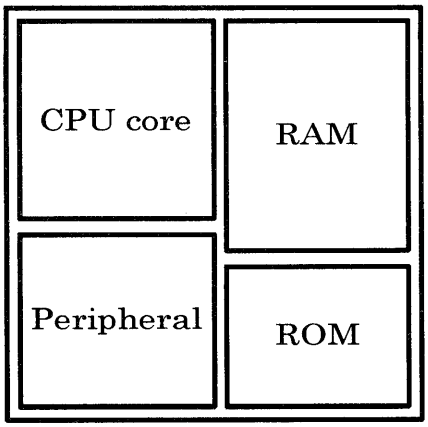


図 1 - ASIP の概念図

一般に，CPU コアのクロック周波数はメモリのアクセスサイクル周波数よりも非常に高いため，CPU コアと内部メモリおよび外部メモリの間に高速なキャッシュメモリが必要となる．さらに，マルチメディア関連の組込みアプリケーションは非常に大きいメモリ空間を必要とするために，付加的な外部メモリが必要不可欠である．したがって，膨大なデータを扱うことのできる ASIP のための効率的なメモリシステムのアーキテクチャモデルが必要とされる．

本稿では，はじめに ASIP に適した階層化メモリシステムを提案する．次に，提案したメモリシステムモデルの性能およびコストの定式化を行い，キャッシュメモリとオンチップメモリの間のトレードオフについて考察する．最後に，サンプルプログラムを用いて提案したメモリシステムモデルの有効性を評価する．

2. メモリシステムモデル

一般に，データや命令のワーキングセットが全てメモリアクセスが高速な内部メモリに格納された場合はシステムの性能は高くなる．しかし，命令やデータのワーキングセットの大きさは対象となるアプリケーションの特徴に大きく依存する．また，アプリケーションによっては内部メモリだけでなく，付加的な外部メモリも必要となる場合がある．大容量の外部メモリを DRAM(または SRAM)で実現すると，I/O バス幅の制約や外部メモリのアクセス速度の限界等から，高速なメモリアクセスを行うことが困難になる．したがって，内部メモリと外部メモリの特性を考慮した効率的なメモリシステムのモデルが必要である．

提案したメモリシステムモデルの概念的なブロック図を図 2 に示す．このモデルは以下の要素から構成される：

- キャッシュメモリ
- 内部メモリ
- 外部メモリ
- TLB(Translation Look-aside Buffer)

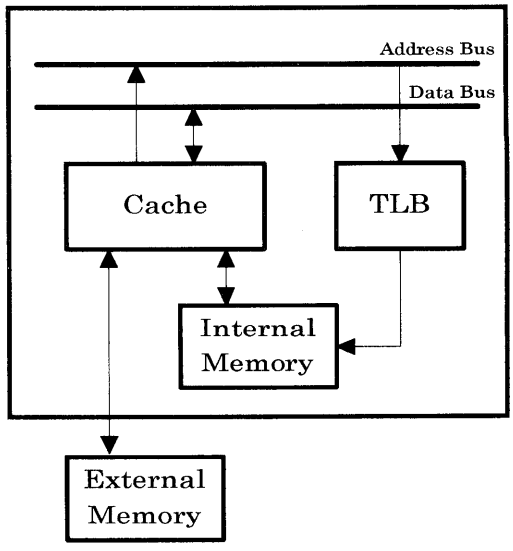


図 2 - メモリシステムモデルの構成

次に，提案したメモリシステムモデルの特徴の概要を以下に示す．

i. アクセス制御機構を有する大容量の内部メモリ
ii. CPU コア等の内部機能ユニットと DRAM メモリ等のアクセスの遅いメモリの間のバッファとしてのキャッシュメモリ
iii. 仮想記憶管理ユニットの TLB と同様な機能を有する内部メモリ管理機能

メモリシステムモデルの各々の構成要素に対する構成パラメータを以下に示す：

i. キャッシュメモリ
   - ブロックサイズ (Block_Size)
   - ブロック数 (Number_of_Block)
   - 連想段数 (Associativity)
ii. 内部メモリ
   - ビット幅 (Bit_Width)
   - ワード数 (Number_of_Word)
iii. 外部メモリ
   - ビット幅 (Bit_Width)
   - ワード数 (Number_of_Word)
iv. TLB
   - エントリ数 (Number_of_Entry)

3. メモリシステムモデルの予備的考察

ハードウェアコストの見積もりを以下のように定式化する．

$$\begin{aligned}\text{Hardware\_Cost} &= \text{Cache\_Cost} + \text{Internal\_Mem\_Cost} \\ &\quad + \text{TLB\_Cost} + \text{External\_Mem\_Cost} \quad (1) \\ &= h_1 \cdot (\text{Block\_Size} \times \text{Number\_of\_Block}) \\ &\quad + h_2 \cdot \text{Associativity} \\ &\quad + r_1 \cdot (\text{Bit\_Width} \times \text{Number\_of\_Word}) \\ &\quad + r_2 \cdot (\log(\text{Number\_of\_Word})) \\ &\quad + t \cdot \text{Number\_of\_Entry} \\ &\quad + e \cdot (\text{Bit\_Width} \times \text{Number\_of\_Word}) \quad (2)\end{aligned}$$

ここで，係数 h1，h2，r1，r2，t，e は単位当りの面積係数である．

次に性能の見積もりは以下のように定式化される．

$$C = C_1 + C_2 + C_3 \quad (3)$$

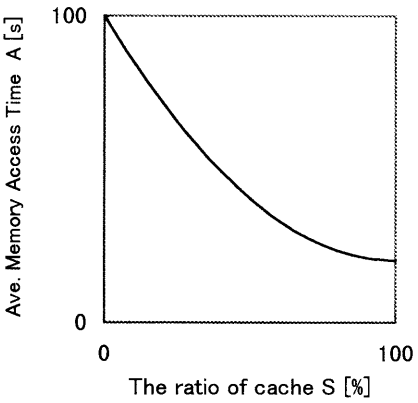
$$T = k_1 \cdot C_1 + k_2 \cdot C_2 + k_3 \cdot C_3 \quad (4)$$

C はメモリの全アクセス数，T はメモリの全アクセス時間である．また，C_1，C_2，C_3 はキャッシュメモリ，内部メモリ，外部メモリに対するアクセス数である．係数 k_1，k_2，k_3 は各々キャッシュメモリ，内部メモリ，外部メモリのアクセスに要するコストを表す．ここで，係数 k_2 はキャッシュメモリのミスヒットによるペナルティ，係数 k_3 は内部メモリのミスヒットによるペナルティを含む．

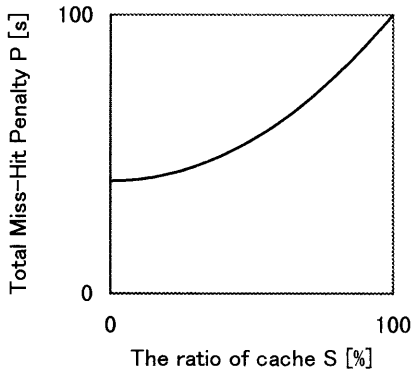
今回の実験では，以下のような仮定を行った：

i. プログラムメモリはチップ内部に実装されており，全ての命令を含むものとする．プログラムメモリが ROM で実装されている場合は，命令キャッシュが必要になる可能性がある．しかし，命令キャッシュに対するヒット率は十分高く，命令キャッシュのミスヒットによる影響を無視できるものと考える．例えば，組込み用途のアプリケーションでは，命令サイズはそれほど大きくなく，全ての命令をキャッシュメモリにロードすることが可能であると考えられるからである．
ii. キャッシュメモリと内部メモリにおけるデータの置き換えは LRU(Least Recently Used) アルゴリズムを使用する．本実験において，LRU アルゴリズムは理想的なものとする．
iii. 本実験においては，連想記憶，メモリデコーダー，TLB，外部メモリなどのハードウェアコストは考慮していない．キャッシュや内部メモリアレイのハードウェアコストと比較すると，他の構成要素のハードウェアのコストは相対的に小さいと考えられるからである．

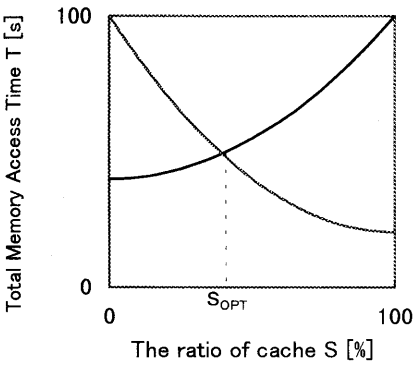
次に，提案したメモリモデルのふるまいについて考察を行う．以下の議論において，メモリ全体のハードウェア量を定数 K に固定する．一般に，キャッシュメモリの容量が大きければ，図 3-(a)に示すようキャッシュメモリのヒット率は高くなる．ここで，S はメモリ全体のハードウェア量におけるキャッシュメモリの割合を示し，A は平均メモリアクセス時間を示す．しかし，キャッシュメモリの占める割合が増加すると内部メモリのヒット率が下がるので，図 3-(b)に示すようにキャッシュのミスヒットによるペナルティ時間が増加する．ここで，P は全ペナルティ時間を示す．したがって，上述したメモリアクセスの影響を考慮することにより，図 3-(c)に示すように最適なキャッシュメモリの割合を決定することが可能である．図中の S_{OPT} は最適なキャッシュメモリの割合を示し，T は全メモリアクセス時間を示す．



(a) キャッシュメモリの効果



(b) キャッシュのミスヒットの効果



(C) キャッシュと内蔵メモリのトレードオフ

図3 - メモリモデルのふるまい

4. 実験

本節では，与えられたハードウェアコストの制約の下で，キャッシュと内部メモリの割合を変化させた場合の性能のトレードオフを，前述した単純化したメモリモデルを用いて実験する．

本実験において，キャッシュと内部メモリのヒット率を計算するために，以下のデータを計測する．

i. キャッシュメモリの全アクセス回数
ii. 内部メモリの全アクセス回数
iii. 外部メモリの全アクセス回数

実験対象のプロセッサとしてDLX[3]を使用した．データ転送のトレースデータを得るために，DLXシミュレータとDLXのクロスCコンパイラ[4]を用いた．得られたトレースデータをdineroキャッシュシミュレータ[5]により解析した．本実験で使用したサンプルプログラムはPVRG-JPEG[6]，PVRG-MPEG CODEC[7]，PVRG-P64 CODEC(ITU-T H.261)[8]である．本実験で使用したサンプルプログラムでは，いずれもDCT変換アルゴリズムを用いた画像の圧縮・展開を行う処理を含んでいる．

本実験において，いくつかのメモリモデルのパラメータを以下のように設定する．

i. キャッシュのブロックサイズと内部メモリのビット幅は共に256ビットとする．
ii. 外部メモリのバス幅は64ビットとする．
iii. キャッシュの連想方式はダイレクトマップ方式とする．
iv. キャッシュの書き換え方式はライトバック方式とする．
v. ミスヒットに対するペナルティの係数 k_1，k_2，k_3 は1，5，40とする
vi. 内部メモリの面積の係数 r とキャッシュの面積の係数 h の比率は1：10とする

図4にJPEG（入力データ 128×128 ピクセル），図5にMPEG（入力データ 4 フレーム），図6にP64（入力データ 2 フレーム）の実験結果を示す．ここで，S は与えられたハードウェアの制約におけるキャッシュの割合，T はメモリアクセスサイクル数を表わす．ハードウェアの制約はキャッシュのハードウェアコストで表現する．例えば，ハードウェア制約 16K は，16K バイトのキャッシュに相当するハードウェアコストを制約条件として使用することを示す．

図7から図11に提案したメモリモデルと従来の

キャッシュメモリモデルの比較結果を示す．ここで，従来のキャッシュメモリモデルとは，内部メモリを持たずキャッシュしか有していないメモリモデルである．以下に改善率(Speedup)の計算式を示す．式中の T(X)はキャッシュの割合が X%の場合のメモリアクセスサイクル数である

$$\text{Speedup} = (T(100) - T(X)) / T(100) \qquad (5)$$

図 7 は JPEG，図 8 は MPEG の比較結果であり，使用した入力データは図 4 および図 5 の実験と同様である．図 9 から図 11 は入力データが 1 フレームから 3 フレームの場合の P64 の比較結果である．

5. 考察

はじめに，使用したサンプルプログラムの実験結果について考察を行う．

(1) **JPEG**:

JPEG のデコードおよびエンコードは同様な手順で行われる．したがって，必要とされる探索テーブル等のデータ空間も同様であり，実験結果もほぼ同様な傾向である．

図 4 および図 7 より，(1) キャッシュメモリの大きさが 8K バイトと 16K バイトの場合のアクセスサイクル数の違いが大きいこと，(2) 制約条件が 4K および 8K の場合の実験結果から，内部メモリの大きさが数十 K バイトの場合の改善率が最も大きいことから，データのワーキングセット全体の大きさが数十 K バイト程度であり，特にアクセス頻度の高いワーキングセットの大きさは数 K バイト程度であると思われる．

したがって，制約条件が 4K および 8K の場合でも内部メモリの割合を大きくすることにより，32K バイトのキャッシュを持つ場合（制約条件 32K において内部メモリの割合が 0%）と同等の性能を得られると考えられる．

(2) **MPEG**:

MPEG のエンコードの演算量はデコードの演算量と比較して非常に大きい．また，画像の探索テーブル等に必要なデータ空間の大きさも JPEG よりも大きい．

図 5 より，制約条件 8K と制約条件 16K の差が非常に大きいことがわかる．したがって，アクセス頻度の高いワーキングセットの大きさは数十 K バイト程度になると思われる．

図 8 より，制約条件 8K におけるエンコードの改善率が高いことがわかる．エンコードのデータアクセス数はデコードの場合の十数倍であることから，データキャッシュおよび内部メモリのヒット率が高くなったことが原因だと考えられる．

(3) **P64**:

P64 は MPEG と同様な方式で画像の圧縮・展開を行うが，演算量は MPEG と比較すると少ない．また，探索テーブル等に必要なデータ空間も MPEG とは異なる．

図 6 より，制約条件 8K の結果はデコードとエンコードにおいて使用されるデータ量の違いが大きく影響していると考えられる．図 9 から図 11 の結果から，デコードの場合は制約条件 256K 以上の場合，エンコードの場合は制約条件 1M 以上の場合は内部メモリを有することによる性能改善が見られない．したがって，エンコードのワーキングセットの大きさはデコードの数倍であると考えられる．

次に，図 9 から図 11 の結果より，入力されるフレーム数が大きくなるにしたがって，デコードおよびエンコードの性能改善率が徐々に低下している．これは，入力される画像フレーム間の違いが小さくなることが影響していると考えられる．

評価実験の結果から，提案したメモリモデルは従来のキャッシュだけを有するメモリモデルよりメモリアクセス時間を改善することができることが知られた．また，サンプルプログラムのワーキングセットの大きさに影響されるが，与えられた制約条件のもとで最適なメモリの割合を得ることができる．最適な内部メモリを実装することにより，より少ないメモリ量で高いメモリアクセス性能を得ることができる．

6. おわりに

高速キャッシュメモリ，大容量の内蔵メモリ，および非常に容量の大きい外部メモリからなる ASIP のための新しい階層化メモリモデルを提案した．そして，このモデルの有効性を評価した．我々の例では，従来のキャッシュメモリモデルと比較して数十%性能が向上した．

より正確なモデルの定式化，および最適なメモリ構成の決定手法等は今後の課題である．

謝辞

貴重な資料の提供およびご助言をいただいた株式会社 SRA の引地信之氏，株式会社日立製作所の堀田正生氏，松下電器産業株式会社の村岡道明氏，また討論に参加してしていただい大阪大学 VLSI システムデザイン研究室の諸兄に深謝致します．なお，本研究の一部は文部省科学研究費補助金 奨励研究 (A) 08780316 の研究助成による．

参考文献

[1] "Feature Size and Integrated Circuit Capability, 1995 - 2001," National Technology Roadmap for Semiconductors, 1994.

[2] Toru Shimizu et al: "A Multimedia 32b RISC Microprocessor with 16Mb DRAM," IEEE International Solid-State Circuits Conference, Digest of Tech. Paper, pp. 216 - 217, 1996.

[3] John L. Hennessy and David A. Patterson: Computer Architecture A Quantitative Approach, Morgan Kaufmann Publishers, 1990.

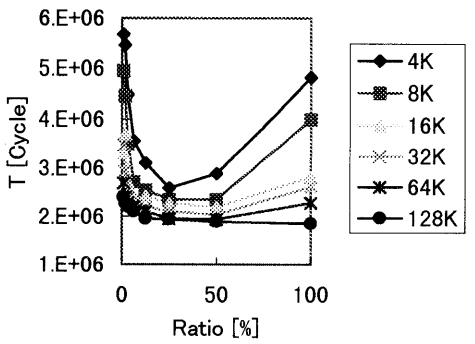
[4] http://www-mount.ee.umn.edu/mcerg/software.html

[5] Mark D. Hill et al: "Experimental Evaluation of On-Chip Microprocessor Cache Memories," Proc. of 11th International Symposium on Computer Architecture, pp. 158 - 166, 1984.

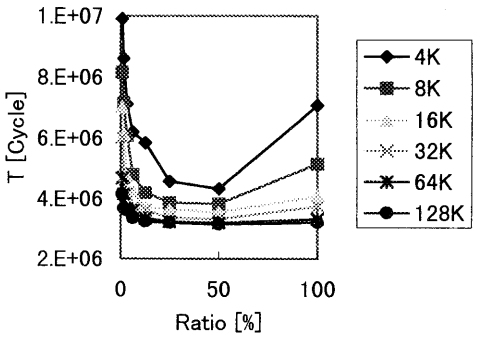
[6] ftp://havefun.stanford.edu/pub/jpeg/JPEGv1.2.1.tar.Z

[7] ftp://havefun.stanford.edu/pub/mpeg/MPEGv1.2.1.tar.Z

[8] ftp://havefun.stanford.edu/pub/p64/P64v1.2.2.tar.Z

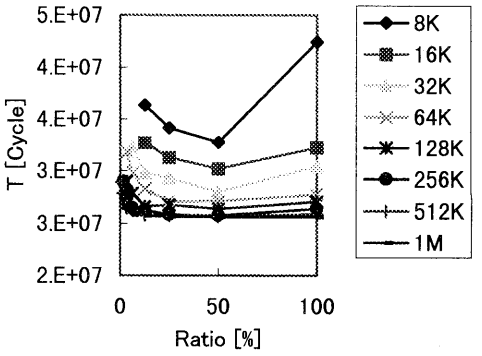


(a) decode

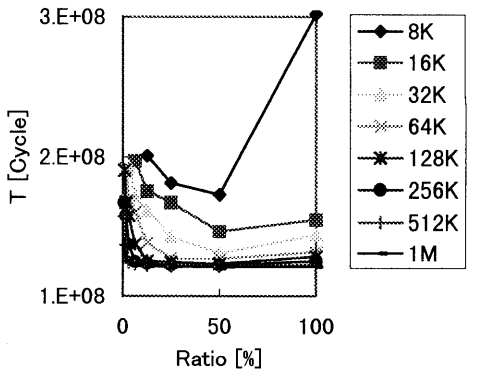


(b) encode

図 4 - JPEG (128×128 pixel)のアクセス時間

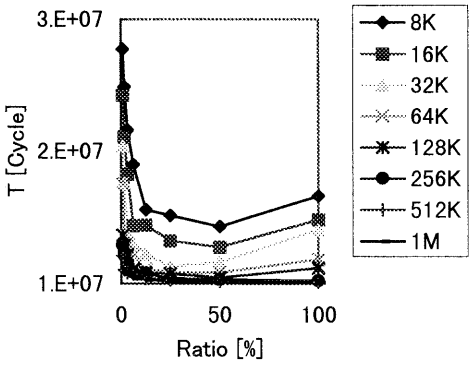


(a) decode

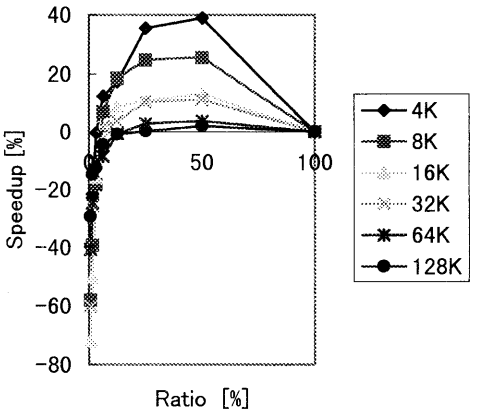


(b) encode

図 5 - MPEG (4 frame)のアクセス時間

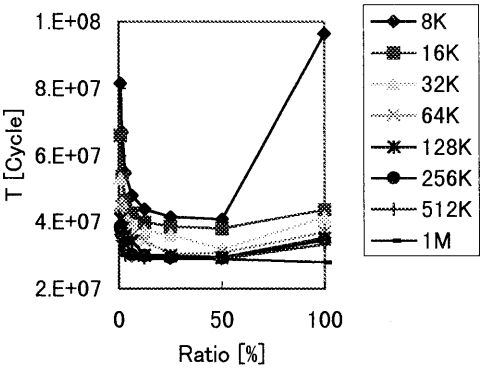


(a) decode



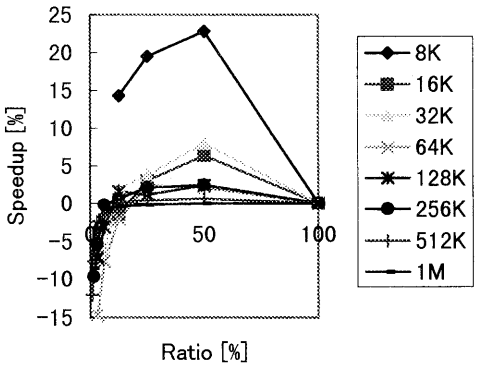
(b) encode

図 7 - JPEG(128×128 pixel)のアクセス時間改善率

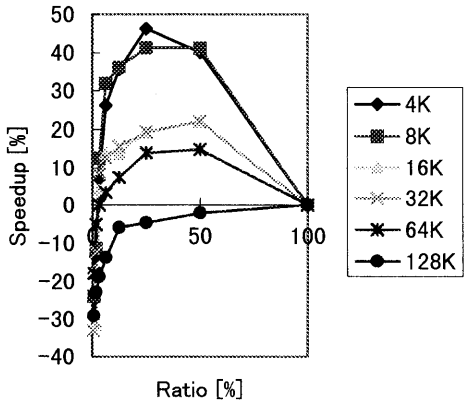


(b) encode

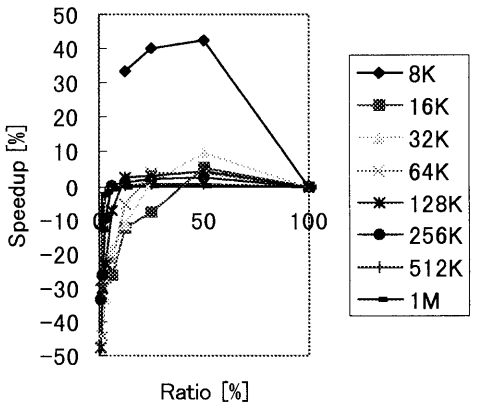
図 6 - P64 (2 frame)のアクセス時間



(a) decode

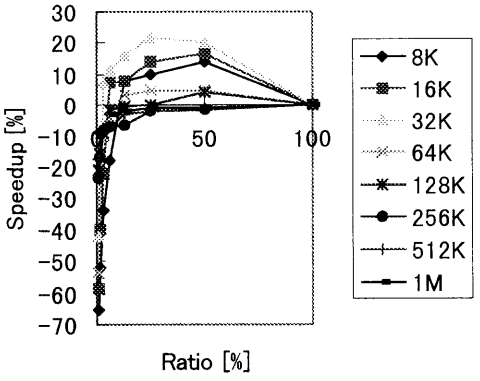


(a) decode

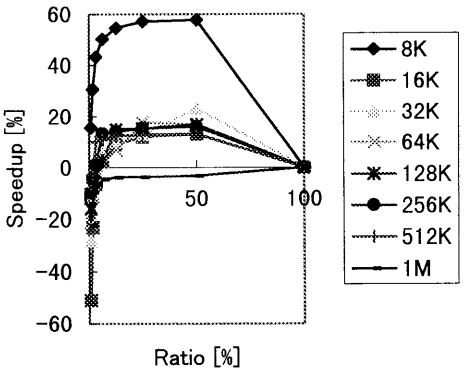


(b) encode

図 8 - MPEG(2 frame)のアクセス時間改善率

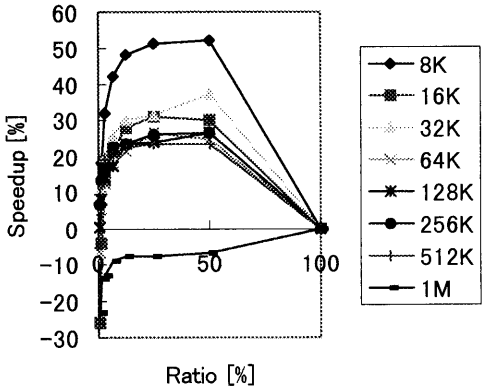


(a) decode



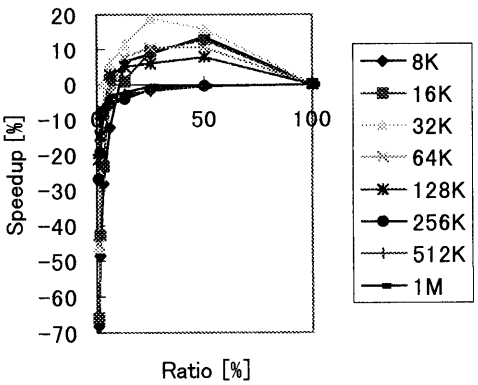
(b) encode

図 10 - P64(2 frame)のアクセス時間改善率

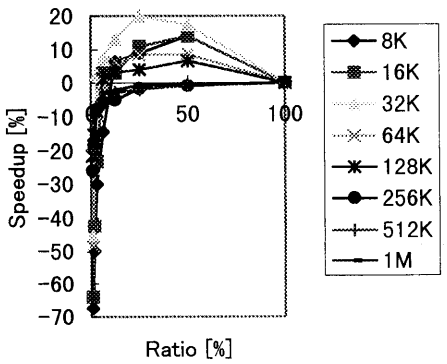


(b) encode

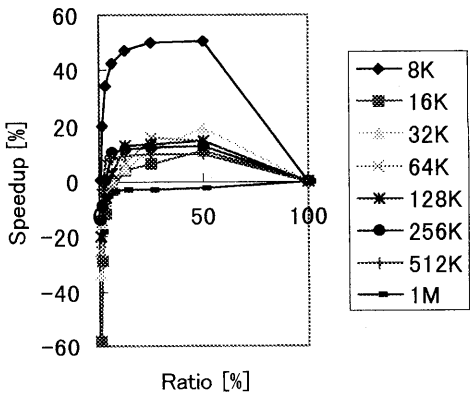
図 9 - P64(1 frame)のアクセス時間改善率



(a) decode



(a) decode



(b) encode

図 11 - P64(3 frame)のアクセス時間改善率