

## コード配置変更によるハイブリッドローカルメモリの消費エネルギー最小化

松村, 忠幸  
九州大学大学院システム情報科学府

石飛, 百合子  
九州大学大学院システム情報科学府

石原, 亨  
九州大学システムLSI 研究センター

安浦, 寛人  
九州大学大学院システム情報科学研究院

<https://hdl.handle.net/2324/8329>

---

出版情報：情報処理学会研究報告, 2007-SLDM-131. 2007 (105), pp.25-30, 2007-10. 情報処理学会バージョン：

権利関係：ここに掲載した著作物の利用に関する注意 本著作物の著作権は（社）情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。



# コード配置変更によるハイブリッドローカルメモリの消費エネルギー最小化

松村忠幸<sup>1</sup> 石飛百合子<sup>1</sup> 石原亨<sup>2</sup> 安浦寛人<sup>2,3</sup>

<sup>1</sup> 九州大学 大学院 システム情報科学府

<sup>2</sup> 九州大学 システム L S I 研究センター

<sup>3</sup> 九州大学 大学院 システム情報科学研究院

本稿ではハイブリッドメモリの有効利用によるオンチップメモリの消費エネルギー削減手法を提案する。ハイブリッドメモリは次の二つの領域で構成される。1) 低い  $V_{dd}$ , 低い  $V_{th}$  を用いて設計され、動的消費エネルギーの小さい領域, 2) 高い  $V_{dd}$ , 高い  $V_{th}$  で設計され、静的消費エネルギーの小さい領域の二つである。本稿で想定するハイブリッドメモリにおいて重要な点は、異なる二つの領域のアクセス遅延時間が等しくなるように適切に  $V_{dd}$ ,  $V_{th}$  及びセンスアンプのサイズを設定することである。従って提案するハイブリッドメモリを既存のプロセッサに組み込む場合、プロセッサの大幅な修正設計は必要とならない。本稿ではハイブリッドメモリの消費エネルギーを最小にする最適な領域の分割、コード配置の決定方法を示す。評価実験の結果からスクラッチパッドメモリに対して提案手法を適用することで、全ての場合において消費エネルギーの削減が達成され、最大 48% の消費エネルギー削減効果が得られた。

## Energy Minimization Technique for Hybrid On-chip Memory using Code Replacement

Tadayuki Matsumura<sup>1</sup> Yuriko Ishitobi<sup>1</sup> Tohru Ishihara<sup>2</sup> Hiroto Yasuura<sup>2,3</sup>

<sup>1</sup> Graduate School of Inf. Sci. & EE, Kyushu University

<sup>2</sup> Sytem LSI Research Center, Kyushu University

<sup>3</sup> Faculty of Inf. Sci. & EE, Kyushu University

This paper proposes a effective use of hybrid memory architecture which consists of the following two regions; 1) a dynamic-power conscious region which uses low  $V_{dd}$  and  $V_{th}$  and 2) a static-power conscious region which uses high  $V_{dd}$  and  $V_{th}$ . The key of hybrid memory architecture is that the access delays for the two regions are equal to each other, which eases to integrate this memory into processors whithout major modifications of an internal processor architecture. This paper proposes a technique for finding the size and the code allocation for the regions so as to minimize the total power consumption of the memroy. Experimental results demonstarate that the total power consumption of the scarachpad memory can be reduced in every cases, and in the best case, 48% energy reduction is achieved by using our proposal technique.

### 1. はじめに

近年のバッテリー駆動機器の普及とともに、低消費電力化設計は重要な設計指針となった。CMOS 集積回路の消費電力は大きくは動的電力と静的電力の二つに分類できる。現在は動的電力に対してスイッチング電力が支配的であるため、動的電力はスイッチング電力で近似する事ができる。スイッチング電力は下記の式 (1) で表すことができる。

$$P_{dynamic} \simeq a \cdot f \cdot C \cdot V_{dd}^2 \quad (1)$$

ここで  $V_{dd}$  は電源電圧,  $a$  はスイッチング活性率,  $f$  は動作周波数,  $C$  は回路の平均スイッチング容量である。式 (1) から  $V_{dd}$  は動的電力に対して 2 乗の比例関係にあるため、低消費電力化設計において  $V_{dd}$  を下げることが効果的であることがわかる。一方近年の微細

化の進んだ CMOS 集積回路の遅延時間は下記の式で与えられる<sup>1)</sup>

$$delay \propto \frac{V_{dd}}{(V_{dd} - V_{th})^\alpha} \quad (2)$$

ここで  $V_{th}$  は閾値電圧,  $\alpha$  はおよそ 1.3 となるプロセステクノロジーに依存する定数である。式 (2) から  $V_{dd}$  を下げることは遅延時間の増加を引き起こし、プロセッサの性能低下を引き起こすことが分かる。そのため低消費電力化を達成し、且つ回路の性能を維持するためには同時に  $V_{th}$  を下げなければならない。しかし近年の微細化が進んだ CMOS 集積回路において、 $V_{th}$  を下げるとことはリーク電力の指数関数的増加を引き起こす。近年のプロセスにおいては総リーク電流に対してサブスレッショルドリーク電力とゲートリーク電流が支配的である<sup>2)</sup>。サブスレッショルドリーク電流は

温度に指数関数で依存する．動作時はチップの温度が高くなるため，動作時においては総リーク電流に対してサブスレッショルドリーク電流が支配的であると考えられる．サブスレッショルドリーク電流は下記の式で表わされる．

$$I_{Subthreshold} \propto \exp \frac{-V_{th}}{\theta \cdot V_T} \quad (3)$$

ここで  $V_T$  は温度電圧， $\theta$  はサブスレッショルドファクタであり，およそ 1.4 から 1.65 の間の値をとる．

以上のことから低消費電力化設計では動的電力と静的電力の比率を考慮し，慎重に  $V_{dd}$ ， $V_{th}$  の値を決定し，動的電力と静的電力両方をバランスよく削減することが必要であることがわかる．一般的にメモリ回路はロジック部に比べ活性化率が低く静的電力の比率が大きいため，メモリ回路は性能の低下を引き起こすこと無く，低消費電力化を達成するため高い  $V_{dd}$ ，高い  $V_{th}$  を用いて設計される．一般的にメモリへの参照は局所性をもつため，メモリ回路において動的電力消費は偏りをもつと考えられる．つまり少数のアドレスが頻繁に参照され，動的電力の大部分はその少数のアドレス領域で消費されると考えられる．本稿ではこの参照の局所性を利用した低消費電力化手法としてハイブリッドメモリアーキテクチャの有効な利用を考える．ハイブリッドメモリアーキテクチャは二つのメモリ領域から構成される．一つの領域は動的電力消費の小さい領域であり，低い  $V_{dd}$ ，低い  $V_{th}$  を用いて設計される．もう一方の領域は静的電力消費の小さい領域であり，高い  $V_{dd}$ ，高い  $V_{th}$  を用いて設計される．メモリへの参照を動的消費電力の小さい領域に集中させることで総消費電力を削減する．本稿で想定するハイブリッドメモリアーキテクチャにおける重要な点は，二つの異なるメモリ領域へのアクセス遅延時間は等しくなるように設計される点である．従って提案するハイブリッドメモリを既存のプロセッサに組みこむ場合，プロセッサ側に大幅な修正設計を加えることなく組みこむことが可能である．すでに同様の低消費電力化設計手法が提案されている<sup>3)</sup>．しかし文献<sup>3)</sup>で提案される手法では 2 種類の異なるアクセス遅延時間が存在するため，本研究の提案する手法に比べ，プロセッサの大幅な修正設計が必要になる．

## 2. 提案手法と関連研究

### 2.1 ハイブリッドメモリ

メモリ回路において動的電力と静的電力の最適な比は温度，メモリ容量，アクセス頻度，クロック周波数，使用するプロセスなど様々な要因に依存する．一般的にメモリへの参照は局所性を持つため，動的電力と静的電力の比率はメモリ回路全体で一様ではない．ハイブリッドメモリではメモリ回路を消費電力特性の異なる 2 つの領域にメモリを分割する．1 つは動的電力削減を目的とした領域 (以降 DP 領域と呼ぶ)，もう一方

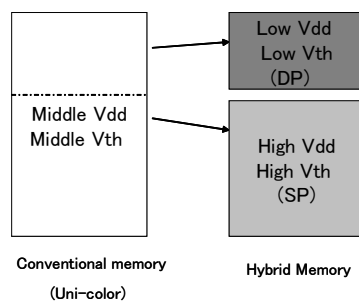


図 1 従来のメモリとハイブリッドメモリ

表 1 想定するキャッシュの仕様

Processor		M32R-II	
Clock Frequency		400MHz	
Temperature		75	
L1 (Instruction)	The number of sets	128	
	The number of ways	2	
	Line size	16 byte	
	Total size	32 KB	
L1 (Data)	The number of sets	128	
	The number of ways	2	
	Line size	16byte	
	Total size	32 KB	
L2 (Unified)	The number of sets	128	
	The number of ways	4	
	Line size	16 byte	
	Total size	64 KB	

は静的電力削減を目的とした領域 (以降 SP 領域と呼ぶ) である．DP 領域は動的電力を削減するために低い  $V_{dd}$ ，低い  $V_{th}$  を用いて設計する．逆に SP 領域は静的電力を削減する目的で高い  $V_{dd}$ ，高い  $V_{th}$  を用いて設計する．頻繁に参照されるデータを DP 領域に配置し，参照頻度の低いデータを SP 領域に配置することで総消費電力を削減することができる．図 1 はハイブリッドメモリの例を示している．本稿ではハイブリッドメモリをスクラッチパッドメモリ (SPM) に適用する．

### 2.2 多階層メモリへの適用例

本節では低消費電力化設計において，動的電力と静的電力の比を考え  $V_{dd}$ ， $V_{th}$  を決定することが有効であることを示す．ここで Level-1 (L1) と Level-2 (L2) の二つの階層からなるオンチップメモリを搭載したプロセッサを考える．一般的に L1 キャッシュと L2 キャッシュのアクセス頻度は大きく異なる．図 2，図 3，図 4 に M32R-II プロセッサにおける L1，L2 キャッシュそれぞれのメモリへのアクセス回数，および消費エネルギーの見積り結果を示す．本実験のメモリの想定条件は表 1 に示すとおりである．それぞれのキャッシュへのアクセス回数はキャッシュシミュレータを用いて計測した．また消費エネルギーは SYNOPSIS の商用回路シミュレータ HSPICE とアクセス回数の結果から計算した．エネルギーの見積りに使用した DP 領域，SP 領域それぞれにおけるパラメータの値は表 2 に示すとおりである．本実験では DP1，SP1 の値を

用いる．表中の P.O. , Delay , leak/cell , leak/SA ,  $SW_{bit}$  , および  $SW_{word}$  はそれぞれプロセスオプション , 読み込み遅延時間 , SRAM セルあたりのリーク電流 , センスアンプあたりのリーク電流 , ビットラインの充放電時に消費されるエネルギー , ワードラインの充放電時に消費されるエネルギーを示す．表 2 の値は HSPICE を用いて求めた．SPICE シミュレーションに用いたトランジスタモデルは商用の 90nm プロセステクノロジーを用いた．実験に用いたライブラリでは HP , MP の二つのプロセスオプションが提供されている．HP モデルライブラリはパフォーマンス重視のモデルであり ,  $V_{th}$  ,  $T_{ox}$  (ゲート酸化膜厚) はパフォーマンスがあがるように設定されている．逆に MP モデルは消費電力重視のモデルであり ,  $V_{th}$  ,  $T_{ox}$  は消費電力が下がるように設定されている．各々のメモリへのアクセス回数はアプリケーションプログラムに依存するが , すべての場合において , L2 キャッシュへのアクセス回数は L1 キャッシュへのアクセス回数よりも非常に少ない (図 2 参照) ．そのため L1 キャッシュと L2 キャッシュでは , 動的電力と静的電力の比が大きく異なる．図 3 および図 4 は DP 設計 , SP 設計で各メモリを設計した場合の , 100 万命令実行までに消費されるエネルギーの見積もり結果である．L1 命令キャッシュはすべての場合において動的電力削減を目的とした DP 条件での設計を用いることでエネルギー消費を抑えることができ , SP 設計の時に比べ DP 設計では最大 48% のエネルギーが削減される．このとき DP 設計と , SP 設計ではメモリアクセス遅延時間はほぼ等しいため (表 2 参照) , パフォーマンスの低下は無いということが重要である．逆に L2 キャッシュでは静的電力削減を目的とした SP 設計を用いることで消費電力を抑えことができ , DP 設計に比べ SP 設計を用いた場合最大 38% の電力が削減される．DP 設計 , SP 設計のどちらを用いるのが良いのかの違いは , アクセス頻度の違いに起因する動的電力と静的電力の比から生じている．本実験から動的電力と静的電力の比を考慮し , 慎重に  $V_{th}$  と  $V_{dd}$  を決定することが重要である事がわかる ．

### 2.3 関連研究

文献 3) では non-uniform set-associative (NUSA) cache が提案されている．NUSA キャッシュの基本アイデアはハイブリッドメモリと同様である．NUSA キャッシュは低い  $V_{th}$  を用いて設計される高速なウェイと , 高い  $V_{th}$  を用いて設計される低速なウェイから構成される．頻繁にアクセスされるデータを高速なウェイに集めることで , 性能の低下を抑えつつリーク電力を削減する．しかし NUSA では頻繁にアクセスされるデータを高速なウェイに集中させるために , 高速なウェイと低速なウェイ間でデータを授受する機構を新たに付け加えなければならない．このことは設計を複

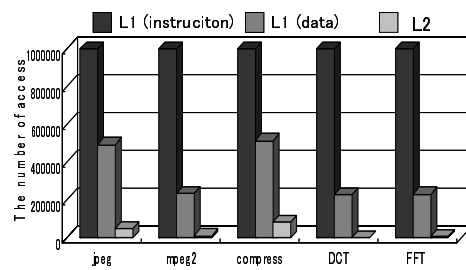


図 2 The number of accesses to each cache

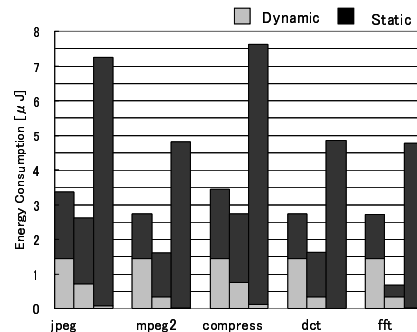


図 3 Energy breakdown of DP-design

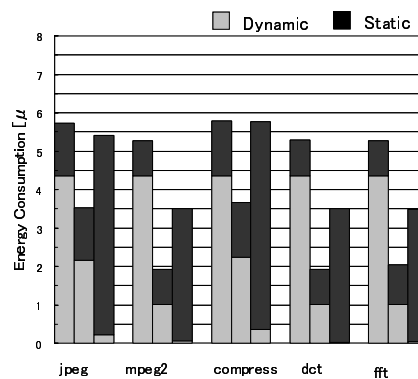


図 4 Energy breakdown of SP-design

雑にするだけでなく , ウェイ間のデータ送受信に伴う性能低下が問題となる．それに加え高速なウェイと低速なウェイの異なる 2 つのアクセス遅延が存在するため , パイプライン機構が複雑になり , 既存のプロセッサにそのまま組み込むことが難しい ．

### 2.4 提案手法

本研究ではハイブリッドメモリアーキテクチャをスクラッチパッドメモリ (SPM) に適用する．SPM はキャッシュメモリと同様に容量は小さいが , 高速なアクセスが可能なオンチップメモリである．キャッシュメモリ , SPM とともに SRAM セルから構成される．しかし SPM はキャッシュメモリに比べ消費エネルギーが小さい．その理由は SPM ではアドレスがソフトウェアプログラムにより静的に割り当てられているため , キャッシュメモリで必要となるタグ比較を行う必要が無いためである．そのため現在 SPM は性能を上げる

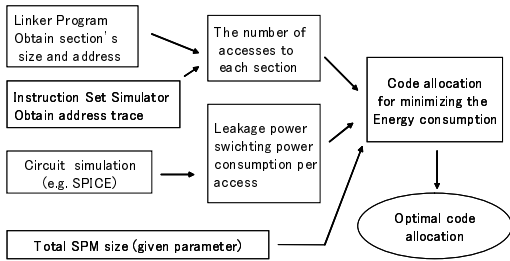


図 5 最適化フロー

ためだけでなく、低消費電力化を目的としても広く使われている。典型的なコード配置手法では性能改善、消費電力削減を目的として、メインメモリに配置する代わりに SPM に配置すべき基本ブロックや関数を決定する。コード配置は一般的にコンパイラ処理の段階において行われる<sup>(8)(9)</sup>。本稿ではハイブリッドメモリアーキテクチャを適用した SPM への、総消費電力を最小にするメモリオブジェクトの配置を決定する問題を考える。配置換えの対象となるメモリオブジェクトは関数とグローバル変数、グローバル定数とする。総消費電力を最小とするコード配置を決定するには、与えられたアプリケーションプログラムにおける各関数、およびデータオブジェクト（以降メモリオブジェクトと呼ぶ）へのアクセス回数を知る必要がある。本研究では各関数、データオブジェクトへのアクセス回数を調べるために命令セットシミュレータを用いた。またコンパイルログから各関数、データオブジェクトのサイズを得た。メモリ回路の動的電力、静的電力は SPICE シミュレーションにより求めた。これらの基本データをもとに、総消費電力を最小にする最適なコード配置を求める。最適なコード配置を求めるフローを図 5 に示す。

### 3. 問題定義

#### 3.1 表記の定義

- $A$ : 対象アプリケーションプログラムの数。
- $FS_{i,j}$ :  $i$  番目のプログラムに含まれる、 $j$  番目のメモリオブジェクトのサイズ。単位はバイト。
- $X_{i,j}$ :  $i$  番目のプログラムに含まれる、 $j$  番目のメモリオブジェクトのアクセス回数。
- $NA_i$ :  $i$  番目のアプリケーションプログラムに含まれるメモリオブジェクトの数の総和。
- $N_i$ :  $i$  番目のアプリケーションプログラムに含まれるメモリオブジェクトの内 SPM に配置されるメモリオブジェクトの数の総和。
- $ED_{DP}, ED_{SP}$ : DP, SP 領域それぞれへのアクセス時に消費される動的エネルギー
- $ES_{DP}, ES_{SP}$ : DP, SP 領域それぞれの 1 バイトあたりの静的消費エネルギー。
- $MS$ : SPM の総容量。単位はバイト。
- $s$ : DP 領域の容量。単位はバイト。

- $T$ : 総プログラム実行時間。
- $a_{i,j}$ : 0-1 の整数変数。 $i$  番目のアプリケーションプログラムにおいて  $j$  番目のメモリオブジェクトが DP 領域に配置される時 1, それ以外の時 0。
- $b_{i,j}$ : 0-1 の整数変数。 $i$  番目のアプリケーションプログラムにおいて  $j$  番目のメモリオブジェクトが SPM に割り当てられる時 1, それ以外の時 0。

$ED_{DP}, ED_{SP}, ES_{DP}, ES_{SP}$  の値は SPICE シミュレーションから得た。 $MS$  の値は与えられるとする。本研究では  $ED_{DP}, ED_{SP}$  は  $MS$  や  $s$  には依存しないと仮定した。この仮定は現実的ではない場合がある。より現実的なモデルを用いた実験を行うことは今後の課題である。

#### 3.2 SPM へ配置するメモリオブジェクトの決定

対象とするアプリケーションプログラムのすべてのメモリオブジェクトの中から、SPM 領域に配置するメモリオブジェクトを決定する。既に SPM を有効的に利用するための多くの手法が提案されている<sup>(8)(9)</sup>。本稿では SPM に配置するメモリオブジェクトは SPM への参照回数が最大となるように決定する。すなわち以下の最適化問題を解くことにより決定する。

For each  $k = 1 \dots A$

$$\text{Maximize: } \sum_{j=1}^{N_k} X_{k,j} \cdot b_{k,j} \quad (4)$$

$$\text{Subject to: } \sum_i^{NA_k} FS_{k,j} \cdot b_{k,j} \leq MS \quad (5)$$

目的関数、制約条件は (4), (5) でそれぞれ与えられる。これは典型的なナップサック問題である。

#### 3.3 ハイブリッドメモリにおける消費エネルギーを最小化するコード配置問題の定式化

SPM に配置するメモリオブジェクトの決定後、SPM のエネルギー消費を最小とする DP 領域の容量および、メモリオブジェクトの配置を決定する。目的関数、および制約条件は (6), (7) によりそれぞれ与えられる。以下の問題を解くことによりエネルギー消費を最小とする  $s$  を得る。

$$\begin{aligned} \text{Minimize: } & \sum_{i=1}^A \left\{ \sum_{j=1}^{N_i} ED_{DP} \cdot X_{i,j} \cdot a_{i,j} \right. \\ & \left. + \sum_{j=1}^{N_i} ED_{SP} \cdot X_{i,j} \cdot (1 - a_{i,j}) \right\} \\ & + A \cdot \{ ES_{DP} \cdot s + ES_{SP} \cdot (MS - s) \} \cdot T \quad (6) \end{aligned}$$

For each  $k = 1 \dots (A-1)$

$$\sum_{j=1}^{N_k} FS_{k,j} \cdot a_{k,j} \leq s \quad (7)$$

表 2 DP, SP 領域の想定とアクセス遅延時間, 電力消費

	Vdd [V]	P.O.	Delay [nsec]	leakage/cell [nW]	leakage/SA [nW]	$SW_{bit}$ [J]	$SW_{word}$ [J]
DP1	0.75	HP	0.197	9.27	98.9	4.43e-14	3.38e-14
SP1	1.2	MP	0.198	6.71	60.7	1.32e-13	1.40e-13
DP2	0.85	HP	0.151	10.5	122	6.07e-14	5.77e-14
SP2	1.0	MP	0.157	4.14	167	1.05e-13	8.13e-13

最後に決定された  $s$  に対して, 以下の問題をそれぞれのアプリケーションプログラムに対して解くことにより, 消費エネルギーを最小とするコード配置を決定する.

$$\begin{aligned}
 \text{Minimize: } & \sum_{j=1}^{N_i} ED_{DP} \cdot X_{i,j} \cdot a_{i,j} \\
 & + \sum_{j=1}^{N_i} ED_{SP} \cdot X_{i,j} \cdot (1 - a_{i,j}) \\
 & + (ES_{DP} \cdot s + ES_{SP} \cdot (MS - s)) \cdot T(8) \\
 \text{Subject to: } & \sum_{j=1}^{N_i} FS_{i,j} \cdot a_{i,j} \leq s \\
 & (MS - s) \leq \sum_{i=1}^{N_i} FS_{i,j} \cdot (1 - a_{i,j}) \quad (9)
 \end{aligned}$$

#### 4. 評価実験

本節ではハイブリッドメモリを SPM に適用した場合の, 消費エネルギーの見積もり結果を示し, 低消費電力化設計にハイブリッドメモリおよびコード再配置が有効であるということを示す.

##### 4.1 実験環境

本研究では実験対象として SH3-DSP プロセッサを使用した. クロック周波数は 400MHz とした. 3 種類のベンチマークプログラム, 3 種類の容量の SPM に対して実験を行った. DP, SP 領域の想定は 2 種類の想定に対して行った. 想定条件は表 2 を参照. DP, SP それぞれのセンスアンプは構造は同じであるが, それぞれのアクセスレイテンシを等しくするためにトランジスタサイズが異なる. チップ温度は待機時リーク電流ではなく, 動作時リーク電流を見積もることを目的とし 75 °C を想定した.

##### 4.2 実験結果

表 3, 4, 図 6, 7 に実験結果を示す. ハイブリッドメモリアーキテクチャと従来のメモリの消費エネルギーを比較するため, メモリ回路全体を DP, SP で設計した場合 (それぞれを DP 設計, SP 設計) の時の消費エネルギーの見積もり結果も示す. 表中の 'Reduction.SP' (または DP) は SP 設計 (または DP 設計) に比べハイブリッドメモリを用いることにより達成される消費エネルギー削減率を表す. 本実験ではアプリケーションドメインは 3 つのプログラムから構成される (jpeg, mpeg2, compress). 従って DP 領域の容量

$s$  はこれら 3 つのプログラムからなるアプリケーションドメインに対して最適な値に決定される. 結果からハイブリッドメモリおよびコード再配置を適用することで, すべての場合において消費エネルギーの削減が達成されている. SP 設計に比べ最大 49% の消費エネルギー削減が達成される (compress 8KB). このときの消費エネルギー削減は動的電力による消費されるエネルギーの削減によるものである. 一方 SP 設計と比較する場合, すべての場合において静的電力による消費エネルギーは増加している. この理由は SP 設計は静的電力に対して最適な設計であるからである. しかし結果的には静的エネルギー消費の増加を補う動的エネルギー消費削減が得られるため, 総消費エネルギーの削減が達成されている. 重要なことは, この時性能の低下は無いという点である. 表 4 からわかるとおり DP1, SP1 を想定した場合, 提案手法を用いることにより SP 設計と比較して大幅な消費エネルギーの削減が達成されている. しかし DP1, SP1 を想定した場合, DP 設計と比較して提案手法を用いることによる消費エネルギーの削減は小さい. 特に SPM のサイズが 8KB のとき DP 設計に対して提案手法を用いることの優位性が小さく, 消費エネルギーの削減は 10% 以下である. 8KB では総消費電力に対し動的電力が支配的であるためである. 結果的に動的消費エネルギーに対して最適な構成である DP 設計に対して, 消費エネルギー削減の余地が小さい. またもう 1 つの理由は SP1 の使用する  $V_{dd}$  と DP1 の使用する  $V_{dd}$  の差が大きいということである. 二つの領域へのアクセス遅延時間を等しくするために, SP 領域には高い  $V_{dd}$  を与えなければならない. しかしこのことは SP 領域の静的消費エネルギーを増大させ, SP 領域を用いることで得られる静的消費エネルギーの削減効果を小さくしている. 以上のことから最適な  $V_{dd}$  を考えるということも重要な課題である.

#### 5. おわりに

オンチップメモリの消費エネルギーを削減する手法として, ハイブリッドメモリの有効的利用のためコード配置問題を提案した. また提案手法を SPM へ適用した場合, 提案手法が低消費エネルギー化へ有効であるということを示シミュレーション実験により示した. 本研究では提案手法を SPM へ適用したが, 提案手法は他の種類のメモリに対しても適用可能である. 今後の課題は本研究をキャッシュメモリへ適用することである.

表 3 DP1, SP1 を想定したときの結果

MS	s / MS	application	E_hyb [ $\mu$ J]	E_sp [ $\mu$ J]	Reduction (SP)	E_dp [ $\mu$ J]	Reduction (DP)
8KB	0.347656	jpeg	3.67	6.38	42.5%	3.97	7.6%
		mpeg2	3.20	5.81	44.8%	3.44	7.0%
		compress	3.85	7.56	49.1%	4.23	9.0%
16KB	0.199219	jpeg	5.30	8.07	34.3%	6.22	14.8%
		mpeg2	4.51	7.10	36.5%	5.24	13.9%
		compress	5.39	9.09	40.7%	6.32	14.8%
32KB	0.123047	jpeg	8.53	11.29	24.4%	10.71	20.3%
		mpeg2	7.09	9.80	27.7%	8.82	19.6%
		compress	8.31	12.05	31.0%	10.54	21.1%

表 4 DP2, SP2 を想定したときの実験結果

MS	s / MS	application	E_hyb [ $\mu$ J]	E_sp [ $\mu$ J]	Reduction (SP)	E_dp [ $\mu$ J]	Reduction (DP)
8KB	0.347656	jpeg	4.19	5.21	19.5%	4.99	16.0%
		mpeg2	3.71	4.80	22.6%	4.36	14.8%
		compress	4.58	6.09	24.9%	5.38	15.0%
16KB	0.130859	jpeg	5.31	6.43	17.4%	7.52	29.4%
		mpeg2	4.87	5.76	15.5%	6.38	23.6%
		compress	5.59	7.22	22.5%	7.75	27.8%
32KB	0.064453	jpeg	7.71	8.84	12.8%	12.57	38.7%
		mpeg2	6.79	7.68	11.6%	10.41	34.8%
		compress	7.83	9.46	17.2%	12.48	37.2%

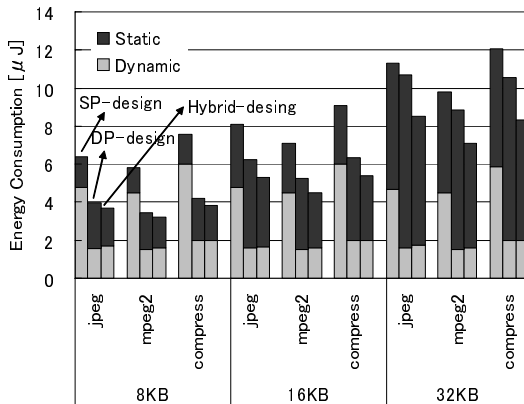


図 6 DP1, SP1 を想定したときの実験結果

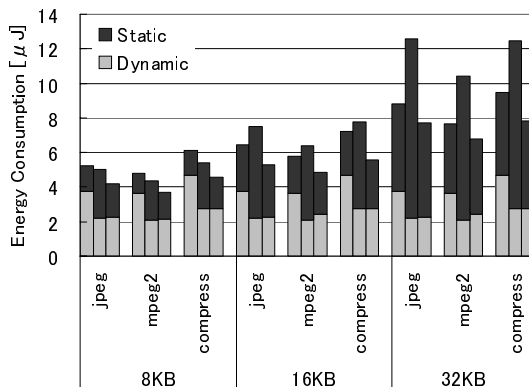


図 7 DP2, SP2 を想定したときの実験結果

## 謝 辞

本研究は東京大学大規模集積システム設計教育研究センターを通して、株式会社ルネサステクノロジ、株式会社半導体理工学研究センター、松下電器産業株式会社、NEC エレクトロニクス株式会社、株式会社東芝、シノプシス株式会社、ケイデンス株式会社、メンター株

式会社の協力で行われたものである。本研究の一部は科学技術振興機構 (JST) の戦略的創造研究推進事業 (CREST) によるものである。

## 参 考 文 献

- 1) T. Sakurai and A. Newton, "Alpha-Power Law MOS-FET Model and Its Applications to CMOS Inverter Delay and other Formulas", IEEE JSSCC Vol. 25, No.2. April 1990.
- 2) H. Chang and Sachin S. Sapatnekar "Full-Chip Analysis of Leakage Power Under Process Variations, Including Spatial Correlations", in Proc. of DAC, pp.523-528, June, 2005.
- 3) A. Sakanaka, S. Fujii and T. Sato, "A Leakage-Energy-Reduction Technique for Highly-Associative Cache in Embedded Systems" ACM SIGARCH Computer Architecture News Vol. 32, No. 3, June 2004.
- 4) R. Gonzalez, Benjamin M. Gordon, and Mark A. Horowitz "Supply and Threshold Voltage Scaling for Low Power CMOS" IEEE JOURNAL OF SOLID-STATE CIRCUITS, Vol. 32, No. 8, august 1997
- 5) T. Ishihara and K. Asada, "A System Level Power Optimization Technique Using Multiple Supply and Threshold Voltages", in Proc. of ASP-DAC, pp.456-461, 2001
- 6) E. Morifuji, T. Yoshida, M. Kanda, S. Matsuda, S. Yamada and F. Matsuoka, "Supply and Threshold-Voltage Trends for Scaled Logic and SRAM MOS-FETs" IEEE TRANSACTIONS ON ELECTRON DEVICES, Vol. 53, No. 6, June 2006
- 7) John L. Hennessy and David A. Patterson, "COMPUTER ARCHITECTURE A Quantitative Approach First Edition" MORGAN KAUFMAN PUBLISHERS, 1990
- 8) S. Stenke, L. Whmeryer, B. Lee and P. Marwedl, "Assigning Program and Data Objects to Scratched for Energy Reduction" in Proc of Design, Automation and Test in Europe Conference and Exhibition, pp.409-415, 2002.
- 9) R. Banakar, S. Steinke, B. Lee, M. Balakrishnan and P. Marwedel, "Scratchpad Memory: A Design Alternative for Cache On-chip memory in Embedded Systems" in Proc. of Design space exploration and architectural design of HW/SW system, pp73-78, 2002.