IPSJ Transactions on System LSI Design Methodology Vol. 2 239–249 (Aug. 2009)

*Regular Paper*

# An Optimization Technique for Low-Energy Embedded Memory Systems

Tadayuki Matsumura,[†1] Tohru Ishihara[†2] and Hiroto Yasuura[†3]

On-chip memories generally use higher supply ($V_{DD}$) and higher threshold ($V_{th}$) voltages than those of logic parts to improve the static noise margin and to suppress the static energy consumption. However, the higher $V_{DD}$ increases the dynamic energy consumption. This paper proposes a hybrid memory architecture which consists of the following two regions; (1) a dynamic energy conscious region which uses low $V_{DD}$ and $V_{th}$ and (2) a static energy conscious region which uses high $V_{DD}$ and $V_{th}$. The proposed architecture is applied to a scratchpad memory. This paper also proposes an optimization problem for finding the optimal code allocation and the memory configuration simultaneously, which minimizes the total energy consumption of the memory under constraints of a static noise margin (SNM), a write margin (WM) and a memory access delay. The memory configuration is defined by a memory division ratio, a $\beta$ ratio and a $V_{DD}$. Experimental results demonstrate that the total energy consumption of our original 90 nm SRAM can be reduced by 62.9% at the best case with a 4.56% area overhead without degradations of SNM, WM and access delay.

## 1. Introduction

Low energy design is one of the most important criteria for today's circuit designers. It is particularly important to reduce the energy consumption of on-chip memories because they are one of the most power hungry components of today's microprocessors. For example, ARM920T$^{TM}$ microprocessor dissipates 43% of the power in its cache memories [1),2)]. StrongARM SA-110 processor, which specifically targets low power applications, dissipates about 27% of the power in its instruction cache [3)]. Energy consumption is divided into two components, dynamic and static energy consumption. Since the dynamic energy consumption depends on $V_{DD}$ quadratically, the dynamic energy consumption can be reduced drastically by lowering $V_{DD}$. However, lowering the $V_{DD}$ causes an increase of the delay which degrades the entire synchronous processor performance [4)]. To keep the processor performance, designers have to lower the $V_{th}$ as well. However in deep sub-micron technology, this causes an exponential increase in subthreshold leakage [5)]. Therefore, it is important for designers to consider the dynamic-to-static energy ratio, and to decide the $V_{DD}$ and $V_{th}$ carefully. In general, memory is designed by using high $V_{DD}$ and high $V_{th}$ due to its low activities and a static energy dominant characteristic.

It is common observation that there is reference locality in memory systems [6)], and as a result, there is also deflection in the dynamic energy consumption. We exploit the memory reference locality for reducing the total energy consumption using a hybrid memory architecture. The hybrid memory architecture consists of the following two regions; (1) a dynamic energy conscious region which uses low $V_{DD}$ and low $V_{th}$ and (2) a static energy conscious region which uses high $V_{DD}$ and high $V_{th}$. The total energy consumption can be saved by concentrating memory accesses on the dynamic energy conscious region. In this paper, the proposed technique is applied to the scratchpad memory (SPM). This is because SPM can be more directly controlled by software than cache memories, which makes it possible for compiler to concentrate the memory accesses on the dynamic energy conscious region by optimizing code allocation. The key of our architecture is that the access delays for the two regions are equal to each other, which eases to integrate proposed memory into processors without major modifications of an internal processor architecture.

This paper is an extension of our previous works [7),8)]. In Ref. 7), $V_{DD}$ used for each region is constant and the static noise margin (SNM) and the write margin (WM) are not discussed. Although the SNM issue is considered in a problem formulation presented in Ref. 8), $V_{th}$ variation is not considered in the evaluation of the SNM, and an algorithm for solving the optimization problem is not given. In this paper we consider the SNM and the WM in the optimization problem, and present our algorithm. The rest of the paper is organized as follows. In Section 2, our approach and related work are presented. An optimization problem

†1 Graduate School of Information Science and Electric Engineering, Kyushu University
†2 System LSI Research Center, Kyushu University
†3 Faculty of Information Science and Electric Engineering, Kyushu University

for minimizing the total energy consumption is formally defined in Section 3. Section 4 presents experimental results. The final section concludes the paper.

## 2. Code Allocation for Hybrid Memory Architecture

### 2.1 Related Works

In Ref. 9), non-uniform set-associative (NUSA) cache is proposed. The NUSA cache consists of one fast cache-way and several slow cache-ways. Frequently accessed data are gathered to the fast cache-way and infrequently accessed data are placed to the slow cache-ways. The slow ways use high $V_{th}$ to suppress the leakage power. This technique drastically reduces the leakage power of cache memory. However, since the access latencies for the fast and slow ways are different from each other, the NUSA cache needs a complicated pipeline structure which makes it difficult to integrate this cache into off-the-shelf processor IPs.

In Ref. 10), Biased Partitioning (BP) configuration is proposed. BP divide the on-chip memory into 2 regions so as to reduce the dynamic power consumption. By dividing the memory into biased 2 regions, one region's load capacitance of the bit line gets smaller and by concentrating the access on this small load capacitance region, the dynamic power consumption can be reduced. However, in Ref. 10), the same $V_{DD}$ and $V_{th}$ are assigned to the two divided regions, and static power consumption is not discussed.

In Ref. 11), a technique exploiting a small subprogram memory whose $V_{DD}$ and $V_{th}$ are lower than those of conventional memory is proposed. An optimization flow to find the optimal $V_{DD}$, $V_{th}$ and code allocation for the subprogram to minimize the total power consumption is also proposed. However, this technique needs to insert extra jump instructions at compiling phase. The major disadvantage of this technique is that it does not take memory stability issue into account nevertheless the subprogram memory is assumed to be designed using low $V_{DD}$ and $V_{th}$.

### 2.2 Hybrid Memory Architecture

Since there is a reference locality in memory accesses, a large percentage of dynamic energy is consumed in a small number of frequently accessed addresses [6]. To exploit this reference locality, our hybrid memory architecture employs the following two regions; a dynamic-energy-conscious region (we refer to this region as DE) and a static-energy-conscious region (we refer to this region as SE). The SE region is the same as a conventional memory which uses higher $V_{DD}$ and higher $V_{th}$ than those of DE region. The DE region is designed with low $V_{DD}$ and low $V_{th}$ to decrease dynamic energy consumption without increasing the access delay. The key of our hybrid memory architecture is that the $V_{th}$ of the DE region is lowered to compensate an increase of access delay nevertheless it causes an increase of the leakage energy consumption. Because of this, there is no performance degradation. Moreover it makes it easy to embed our memory into the processor since there is no difference between our hybrid memory and conventional memory with regard to a memory access delay. This feature reduces the design cost since it helps the reuse of IPs.

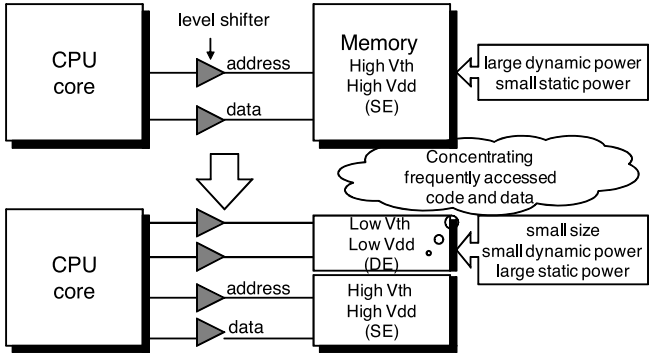


**Fig. 1** Target System and Proposed System.

### 2.3 Target System

We target a processor system which consists of a CPU core and an on-chip SRAM. Since the on-chip SRAM may use higher $V_{DD}$ than that of a logic part in the future technologies such as 65 nm, 45 nm or beyond [12], level shifter circuits are required between the CPU core and the SRAM to shift up the signal voltage level. Our technique replaces the on-chip memory with our hybrid memory. We assume that the supply voltage of CPU core is lower or equal to that of a DE region of our hybrid memory. Therefore, no extra delay is involved by level conversion in our hybrid memory. **Figure 1** shows a simplified image of conventional system and our proposed system.

### 2.4 Our Approach

In this paper we apply the idea of hybrid memory architecture to scratchpad memory (SPM) for reducing the dynamic energy consumption. SPM is a small and high speed on-chip memory which typically consists of an SRAM. At first, we find a memory configuration for a given application domain. The memory configuration is defined by a DE-to-SE region ratio, a supply voltage of the DE region and an SRAM cell size of the DE region. The application domain represents a set of target applications. The optimal memory configuration can be found by solving an optimization problem described in Section 3, which minimizes the total memory energy consumption under constraints of an SNM, a WM and an access delay. Simultaneously, we find the optimal code allocation to DE and SE regions for each application program. In past, there are several code allocation techniques proposed for SPM to improve performance and to reduce energy consumption [13),14)]. These techniques suppose that the code allocation is done at the compilation phase. In this paper, we find functions and data objects (these are referred as memory objects) which should be allocated into the two regions of the hybrid memory for minimizing the total energy consumption of the memory. The data objects include global variables and constants. For finding the optimal code allocation, we need to measure the number of accesses to each memory object for a given application program. We use an instruction set simulator for obtaining this information. The dynamic and static energy consumptions of memory modules can be obtained through SPICE simulation. For such given base data, we find the optimal code allocation to minimize the total energy consumption of the memory by solving an optimization problem described in Section 3. **Figure 2** indicates our proposed optimization flow.

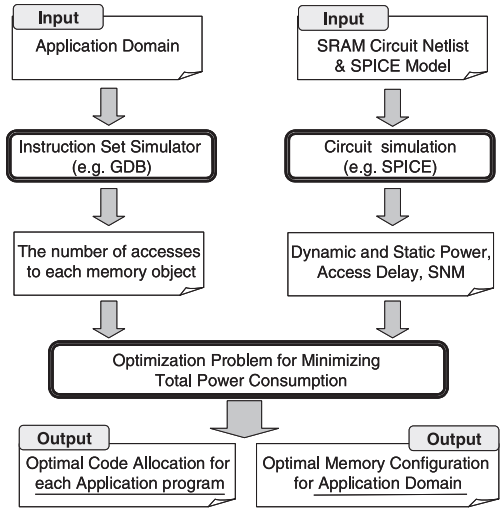


**Fig. 2** Optimization flow.

## 3. Optimization Problem for Minimizing Energy Consumption

In this section, the optimization problem for minimizing the total energy consumption under constraints of a memory access delay, a static noise margin and a write margin is defined. By solving this optimization problem, the optimal memory configuration which includes a DE-to-SE region ratio, a supply voltage ($V_{DD}$) and SRAM cell size ($\beta ratio$) of the DE region for a given application domain are found, and the optimal code allocation for each application program is also found.

### 3.1 Energy and Delay Models

The access delay, the dynamic and static energy consumptions for each memory region are obtained by a circuit simulation. These parameters depend on $V_{DD}$, $\beta$ ratio and memory division ratio. These dependencies are stored in a look-up table to calculate the access delay and the dynamic and static energy consumption in our optimization problem. Since the dynamic energy consumption per memory access and the access delay depend on the memory division ratio (i.e., memory size), the energy consumption and the access delay to each memory region are formulated as functions of the number of SRAM cells connected to each bit line. **Figure 3** shows the access delay and the dynamic energy consumption per memory access for the different numbers of SRAM cells connected to each bit line. Figure 3 indicates that the access delay and the dynamic energy consumption can be accurately approximated as a linear function of the number of SRAM cells connected to each bit line. These approximation coefficients are also stored in the look-up table for each $V_{DD}$ and $\beta$ ratio.

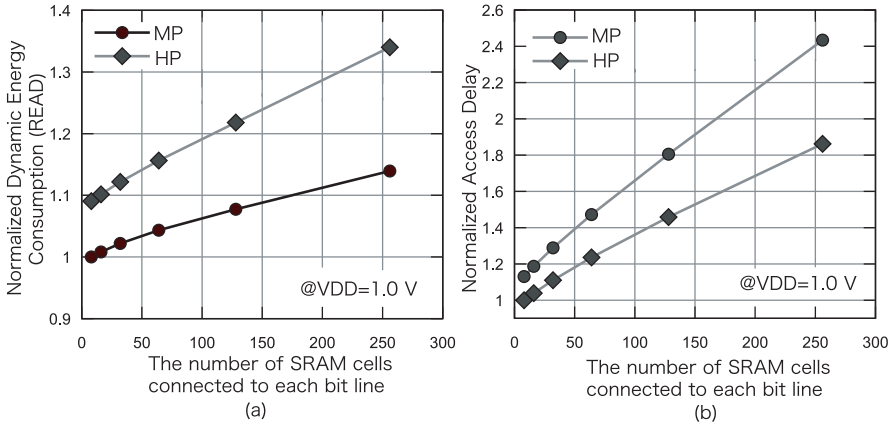


**Fig. 3** (a) Dynamic Energy Consumption and (b) Delay vs. the number of SRAM cells connected to each bit line.

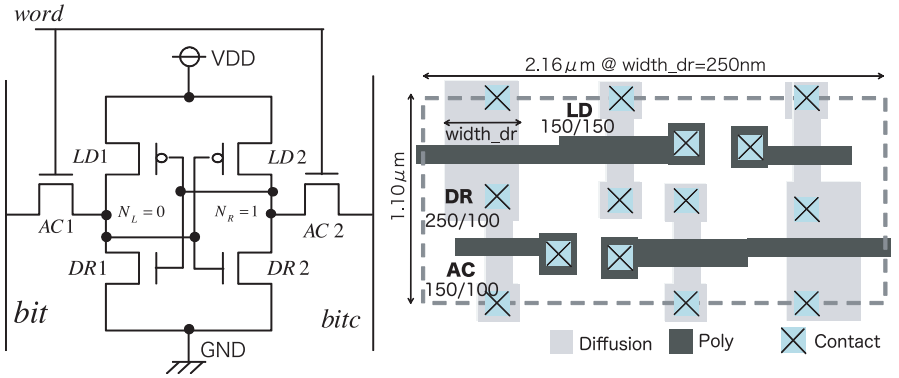


**Fig. 4** 6T SRAM cell schematic and its layout.

### 3.2 SRAM Stability Model

The SRAM cell stability is one of the most important criteria for SRAM circuit design since it affects the SRAM circuit yield. The static noise margin (SNM) is one of the most widely used criteria for representing a SRAM cell stability [15),16)]. The SNM is defined as the minimum DC noise voltage necessary to flip the state of a cell [15)]. In this paper we regard the SNM of a read operation as a criterion of the SRAM stability since the SRAM cell stability degrades mostly in read operation. Threshold voltage ($V_{th}$) variation is one of the major reasons of the SNM degradation. More specifically, local $V_{th}$ variation degrades the SNM drastically since it appears randomly in 6 transistors. This breaks an electrical symmetry of an SRAM cell.

Monte Carlo simulations were performed to calculate the SNM considering $V_{th}$ variation by using circuit simulation varying the $V_{th}$ of all the 6 transistors in the SRAM cell in **Fig. 4**. The normal distribution is assumed for the $V_{th}$ variation. Since a standard deviation of $V_{th}$ is proportional to the inverse square root of a transistor channel area ($1/\sqrt{LW}$) [17)], the standard deviation of the $V_{th}$ for all of the transistor are different according to their transistor channel area. 10,000 simulations were run in each case. The $\sigma_{V_{th}}$ is assumed to be 20 mV in the case that channel length and channel width are 100 nm and 150 nm respectively. In this paper, a SPICE model of a commercial 90 nm CMOS process technology is used for the circuit simulation consistently. The target process technology model used in this paper does not include a low standby power (LSTP) model. We use middle performance (MP) and high performance (HP) models in this paper. In this model, 2 process options; HP and MP are provided. The HP model library is a performance oriented model, and its $V_{th}$ and $T_{ox}$ (gate oxide thickness) are chosen for increasing performance of the circuit. On the other hand, the parameters of the MP model library are chosen for low power design. In proposed hybrid memory architecture, DE region and SE region are designed using HP model and MP model, respectively. In this simulation, the chip temperature is set to 125°C to consider the worst case temperature. The SNM considering $V_{th}$ variation are calculated using obtained mean $\mu_{SNM}$ and standard deviation $\sigma_{SNM}$ of the SNM as follows.

$$SNM = \mu_{SNM} - N \cdot \sigma_{SNM} \quad (1)$$

$N$ is decided according to the memory size. $N$ is assumed to be 5 in this paper. It means that more than 92% yield is guaranteed for the 32 KB memory which is the largest memory size used in our experiment.

**Figure 5** shows the relations among the SNM, $V_{th}$ and $V_{DD}$. The SNM decreases along with a reduction of $V_{DD}$ and $V_{th}$. In proposed hybrid memory

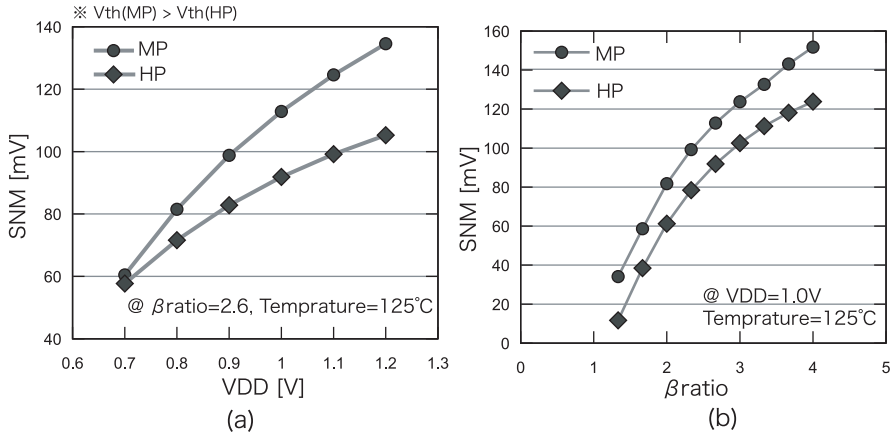


**Fig. 5** (a) SNM vs. $V_{DD}$ and $V_{th}$ (b) SNM vs. $\beta$ ratio.

architecture, low $V_{DD}$ and low $V_{th}$ are assigned to the DE region, which degrades the SNM. Figure 5 demonstrates that the SNM of the DE region is much less than that of the SE region.

In this paper, to compensate for the SNM degradation, an SRAM cell having larger $\beta$ ratio ($=\beta_{DR}/\beta_{AC}$) is used in the DE region. $\beta_{DR}$ and $\beta_{AC}$ represent the transconductance factors of the transistor DR and AC respectively (see Fig. 4). $\beta$ is given by $W/L$. Figure 5 shows the relation between the SNM and $\beta$ ratio. Figure 5 indicates that, the SNM degradation of DE region can be compensated by using large $\beta$ ratio SRAM cell. In this paper, only the width of transistor DR is changed for tuning the $\beta$ ratio and other parameters are unchanged. The SNM increases by using a large $\beta$ ratio though it causes an undesirable area overhead. However, DE region is expected to be small due to a memory reference locality and as a result, an entire area overhead is also very small. The SNM values are stored in a look-up table as a function of $V_{DD}$ and $\beta$ ratio.

### 3.3 Write Margin

In addition to the SNM (read margin) constraint, a write margin (WM) constraint should be also considered [18)–20)]. The WM also decreases along with a reduction of $V_{DD}$. Generally, there is a trade-off between read noise margin (SNM) and write margin (WM) in a conventional 6T SRAM. Therefore a technique such as Ref. 21) which uses different operating voltages for read and write operations is effective to compensate for the degradation of WM. Our technique can be easily extended to use different voltages for read and write operations, since our technique originally uses multiple supply voltages in the SRAM module.

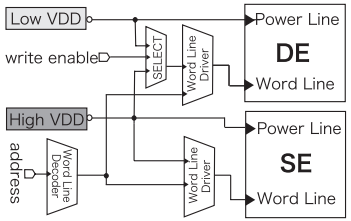


**Fig. 6** Write Margin Compensation System Block Diagram.

In our proposed hybrid memory architecture, the word line driver voltage of the DE region is different according to the operation (i.e., read or write). **Figure 6** shows simplified block diagram of this system. A write enable signal controls word line driver voltage. For the read operation, the lower voltage which is equal to the supply voltage of the DE region are used to reduce the dynamic energy consumption. However, the lower voltage is not used in the write operation to satisfy the WM constraint. The higher voltage which is equal to the supply voltage of the SE region is used in the write operation. Therefore, the dynamic energy consumption of the word line capacitance is not reduced in the write operation. Monte Carlo simulation is performed to calculate the WM. The definition of the WM proposed in Ref. 20) are used. In this simulation, the chip temperature is set to $-40$°C to consider the worst case temperature. The write margin is also calculated considering $V_{th}$ variation as like SNM calculation. The WM values are stored in a look-up table as a function of $V_{DD}$ and $\beta$ ratio.

### 3.4 Notation

This section shows the notations which are used in the next section. $A$, $MS$ $VDD_{SE}$, and $\beta R_{SE}$ are given parameters.

- $A$: The number of given application programs.
- $N_i$: The number of memory objects in the $i^{th}$ application program.
- $MS$, $s$: The size of the total SPM size and the DE region in byte, respectively.
- $T_i$: The total program execution time of the $i^{th}$ application program.

- $FS_{i,j}$: The size of the $j^{th}$ memory object in the $i^{th}$ application program.
- $ACCR(W)_{i,j}$: The number of Read (or Write) accesses to the $j^{th}$ memory object in the $i^{th}$ application program.
- $VDD_{DE,SE}$: The supply voltages of the DE and SE regions, respectively.
- $\beta R_{DE,SE}$: The beta ratios of the DE and SE regions, respectively.
- $EDR(W)_{DE,SE}(VDD_{DE,SE}, \beta R_{DE,SE}, s)$: The dynamic energy consumptions per Read (or Write) access to DE and SE regions in SPM, respectively.
- $PS_{DE,SE}(VDD_{DE,SE}, \beta R_{DE,SE})$: The static power consumptions per byte in DE and SE regions in SPM, respectively.
- $D_{DE,SE}(VDD_{DE,SE}, \beta R_{DE,SE}, s)$: The access delays to DE and SE regions in SPM, respectively
- $SNM_{DE,SE}(VDD_{DE,SE}, \beta R_{DE,SE}, \sigma_{vth})$: The static noise margins of DE and SE regions, respectively.
- $WM_{DE,SE}(VDD_{DE,SE}, \beta R_{DE,SE}, \sigma_{vth})$: The write margins of DE and SE regions, respectively.
- $a_{i,j}$: 0-1 integer variable to be determined. If the $j^{th}$ function or data object in the $i^{th}$ application program is allocated on the DE, $a_{i,j}$=1. Otherwise $a_{i,j}$=0

### 3.5 Problem Description

At first, we find memory objects which should be allocated to SPM from the all memory objects in a given application program. In this paper, memory objects allocated to SPM are found by solving a knapsack problem which maximizes the number of accesses to SPM. After finding functions and data objects allocated to the entire SPM, our optimization flow starts. We find optimal $VDD_{DE}$, $s$ (i.e., DE region size), $\beta R_{DE}$ and optimal code allocation to DE and SE regions for minimizing the energy consumption under constraints of a memory access delay, a static noise margin and a write margin degradation. The objective function and constraints are given by Eqs. (2)–(7). $T_{Delay}$ (target delay), $T_{SNM}$ (target SNM) and $T_{WM}$ (target WM) represent the access delay, the SNM and the WM of the original memory which is designed using MP model and high $V_{DD}$ for the entire memory. As described in previous section, $EDR(W)$ (dynamic energy consumption), and $D$ (access delay) of each memory region are functions of $V_{DD}$, $\beta R$ and $s$. $PS$ (static power consumption) is also a function of $V_{DD}$ and $\beta R$. The values for $EDR(W)$, $PS$, $D$, $SNM$ and $WM$ are calculated by SPICE simulation as described in previous section, and these are stored in a look-up table so that we can use them in our optimization problem. The optimal $VDD_{DE}$, $\beta R_{DE}$ and $s$ are found for a given application domain which consists of several application programs, and the optimal code allocations are found for each application program.

*Minimize*:

$$\sum_{i=1}^{A}\{T_i \cdot PS_{DE} \cdot s + PS_{SE} \cdot (MS - s)\} + \sum_{i=1}^{A}\sum_{j=1}^{N_i}(EDR_{SE} \cdot ACCR_{i,j} + EDW_{SE} \cdot ACCW_{i,j}) \cdot (1 - a_{i,j}) + \sum_{i=1}^{A}\sum_{j=1}^{N_i}(EDR_{DE} \cdot ACCR_{i,j} + EDW_{DE} \cdot ACCW_{i,j}) \cdot a_{i,j} \quad (2)$$

*For each* $k = 1 \cdots A$

$$\sum_{j=1}^{N_k} FS_{k,j} \cdot a_{k,j} \leq s \quad (3)$$

$$\sum_{j=1}^{N_k} FS_{k,j} \cdot (1 - a_{k,j)} \leq (MS - s) \quad (4)$$

$$D_{DE}(VDD_{DE}, \beta R_{DE}, s) \leq T_{Delay} \quad (5)$$

$$T_{SNM} \leq SNM_{DE}(VDD_{DE}, \beta R_{DE}, \sigma_{vth}) \quad (6)$$

$$T_{WM} \leq SNM_{DE}(VDD_{DE}, \beta R_{DE}, \sigma_{vth}) \quad (7)$$

### 3.6 Algorithm

**Procedure MinimizeEnergy**
**Input**: $A$, $N_i$ $MS$, $T_i$, $FS_i$, $ACCR(W)_i$, $VDD_{SE}$, $\beta R_{SE}$
**Output**: $Opt_{vdd}$, $Opt_{\beta R}$, $Opt_s$, $Opt_{a_{i,j}}$
  $e_{min} = +\infty$;
  calculate $T_{delay}$, $T_{SNM}$ and $T_{WM}$ for given $VDD_{SE}$, $\beta R_{SE}$;
  $s = 0$;
  **while** $(s \leq MS)$

```
  VDD_DE = VDD_min;
  while (VDD_DE ≤ VDD_max)
    βR_DE = βR_min;
    while (βR_DE ≤ βR_max)
      calculate D_DE, SNM_DE and WM_DE;
      if (D_DE ≤ T_delay ∩ T_SNM ≤ SNM_DE ∩ T_WM ≤ WM_DE);
        exit from while(VDD_DE ≤ VDD_max);
      end if
      increment βR_DE;
    end while
    increment VDD_DE;
  end while
  calculate EDR(W)_DE,SE, P_DE,SE;
  for each application program
    solve the 0-1 ILP given by (2)-(4);
    accumulate the energy values in e_tmp;
  end for
  if (e_tmp ≤ e_min)
    e_min = e_tmp;
    Opt_βR=βR_DE;   Opt_vdd=VDD_DE;
    Opt_s=s;   Opt_a_i,j=a_i,j;
  end if
 increment s;
 end while
end Procedure
```

This section shows an algorithm which solves the problem defined by Eqs. (2)–(7). The inputs of our algorithm are $A$, $N_i$, $MS$, $T_i$, $FS_i$, $ACCR(W)_i$ $VDD_{SE}$, and $\beta R_{SE}$. Then, $D_{SE}$, $SNM_{SE}$ and $WM_{SE}$ are calculated by using given $VDD_{SE}$, $\beta R_{SE}$ and look-up tables described in previous section. These values are set as the $T_{delay}$ (target delay), $T_{SNM}$ (target SNM) and $T_{WM}$ (target WM). The key of our algorithm is that the problem is solved for a fixed $s$ (DE region size) iteratively. This strategy is motivated by the fact that $V_{DD}$ has a stronger impact on the access delay and energy consumption than $\beta ratio$. Based on this fact, when the memory division ratio is fixed, the optimal $VDD_{DE}$ and the optimal $\beta R_{DE}$ which satisfy the constraints Eqs. (5)–(7) can be found easily since these variables can be determined independently from the memory objects allocation. When the $s$, $VDD_{DE}$ and $\beta R_{DE}$ are fixed, dynamic energy consumption parameters ($EDR(W)_{DE,SE}$) and static power consumption parameters ($P_{DE,SE}$) could be calculated using look-up table. Then, the optimization problem could be regarded as a simple 0-1 integer linear programing (ILP) for each application program. ILOG CPLEX optimization engine [22] is used to solve the ILP. This procedure is repeated as the $s$ is incremented by word line size while $s$ is lower than $MS$ iteratively.

## 4. Experiments and Results

### 4.1 Simulation Setup

This section shows the evaluation results of our proposed technique and demonstrates its effectiveness for energy reduction. The processor used in this experiment is SH3-DSP which is a 32-bit RISC processor developed by Renesas. Two clock frequencies, 200 MHz and 400 MHz, of the processor are assumed to examine for the different types of dynamic to static energy ratio situations. The temperature of the chip is assumed to be 75 degrees centigrade for estimating the *active leakage current* of the memory instead of the *stand-by leakage current*. Application domain is composed of three benchmark programs (i.e., JPEG, MPEG2, compress). Three different sizes of SPM are experimented. The energy consumption, the static noise margin and the write margin are calculated by SPICE simulations for different $VDD_{DE}$ values ranging from 0.7 V to 1.2 V by 0.1 V and for different $\beta ratio_{DE}$ ranging from 1.0 to 4.0 by 0.33. Input parameters $VDD_{SE}$, and $\beta ratio_{SE}$ are assumed to be 1.2 V and 1.66 respectively. $ACCR(W)_i$ and $T_i$ are calculated from an instruction trace obtained by an instruction set simulator of SH3-DSP processor. The length of the trace is 1 million instruction long. The relation between the $\beta$ ratio and SRAM cell area overhead is calculated based on an SRAM layout presented in Fig. 4. A design rule for logic circuits is used for designing the SRAM cell. Only the transistor width of DR (width_dr in Fig. 4) is changed for tuning the $\beta$ ratio. An Intel Xeon quad

**Table 1** The Experimental result (200 MHz).

| MS | application | s/MS | $VDD_{SE}$ | $VDD_{DE}$ | $\beta ratio_{DE}$ | $E_{org}$ [$\mu$J] | $E_{hyb}$ [$\mu$J] | Reduction | A.O. | Opt. Time [sec] |
|---|---|---|---|---|---|---|---|---|---|---|
| 8 KB | JPEG | 0.197 | 1.2 | 0.7 | 3.33 | 5.72 | 3.86 | 32.5% | 4.56% | 11.8 |
| | MPEG2 | | | | | 4.91 | 3.83 | 22.1% | | |
| | compress | | | | | 7.91 | 3.84 | 51.4% | | |
| 16 KB | JPEG | 0.136 | 1.2 | 0.7 | 3.33 | 9.23 | 6.60 | 28.5% | 3.14% | 25.2 |
| | MPEG2 | | | | | 7.69 | 5.80 | 24.7% | | |
| | compress | | | | | 13.04 | 6.65 | 49.0% | | |
| 32 KB | JPEG | 0.071 | 1.2 | 0.7 | 3.33 | 16.24 | 11.92 | 26.6% | 1.64% | 52.2 |
| | MPEG2 | | | | | 13.25 | 10.31 | 22.1% | | |
| | compress | | | | | 23.89 | 12.56 | 47.5% | | |

**Table 2** The Experimental result (400 MHz).

| MS | application | s/MS | $VDD_{SE}$ | $VDD_{DE}$ | $\beta ratio_{DE}$ | $E_{org}$ [$\mu$J] | $E_{hyb}$ [$\mu$J] | Reduction | A.O. | Opt. Time [sec] |
|---|---|---|---|---|---|---|---|---|---|---|
| 8 KB | JPEG | 0.197 | 1.2 | 0.7 | 3.33 | 4.40 | 2.40 | 45.4% | 4.56% | 11.9 |
| | MPEG2 | | | | | 3.86 | 2.66 | 31.1% | | |
| | compress | | | | | 6.67 | 2.48 | 62.9% | | |
| 16 KB | JPEG | 0.201 | 1.2 | 0.7 | 3.33 | 6.59 | 3.79 | 42.5% | 4.65% | 25.0 |
| | MPEG2 | | | | | 5.59 | 3.16 | 43.5% | | |
| | compress | | | | | 10.57 | 4.19 | 60.4% | | |
| 32 KB | JPEG | 0.071 | 1.2 | 0.7 | 3.33 | 10.96 | 6.45 | 41.2% | 1.64% | 52.0 |
| | MPEG2 | | | | | 9.03 | 5.94 | 34.2% | | |
| | compress | | | | | 19.0 | 7.42 | 60.8% | | |

CPU computer running Linux at 3 GHz with 16 GB memory is used to find the optimal solution of the optimization problem defined in the previous section.

### 4.2 Results

This section shows the experimental results. **Tables 1**, **2** and **Fig. 7** show the evaluation results. In the result tables, "Opt. Time" represents the computation time to find the optimal solution using our algorithm. When the memory size is larger, the number of memory object allocated to SPM increases, and our algorithm needs more iterations. This is the reason why the computation time is larger when the total memory size is larger. However, the computation time is less than 1 minute in this experiments. In Fig. 7, left side bars of each application program show the energy consumptions of the conventional memory which is designed using a single $V_{DD}$ and a single $V_{th}$. This correspond to "$E_{org}$" in Table 1 and Table 2. Right bars show the energy consumptions of the hybrid memory where the memory objects are optimally allocated to the two regions by our technique. This correspond to "$E_{hyb}$" in Table 1 and Table 2. In Table 1 and Table 2, $s/MS$, $VDD_{DE}$ and $\beta ratio_{DE}$ indicate the ratio of DE region size to total memory size, optimal supply voltage and optimal $\beta$ ratio of DE region, respectively. **Figure 8** shows the energy breakdown for different ratios of DE region size to total size when the total memory size is 8 KB. Every points satisfy the optimization constraints Eqs. (3)–(7) for corresponding fixed DE to SE ratio. From Fig. 8, it can be seen that the optimal memory division ratio, that is DE to SE ratio, depends on the dynamic to static energy ratio. The proposed technique decreases the dynamic energy consumption while increases the static energy consumption. From the experimental results, the total energy consumption reduces in every case since the reduction of the dynamic energy consumption is larger than the increase of the static energy consumption. The proposed technique is

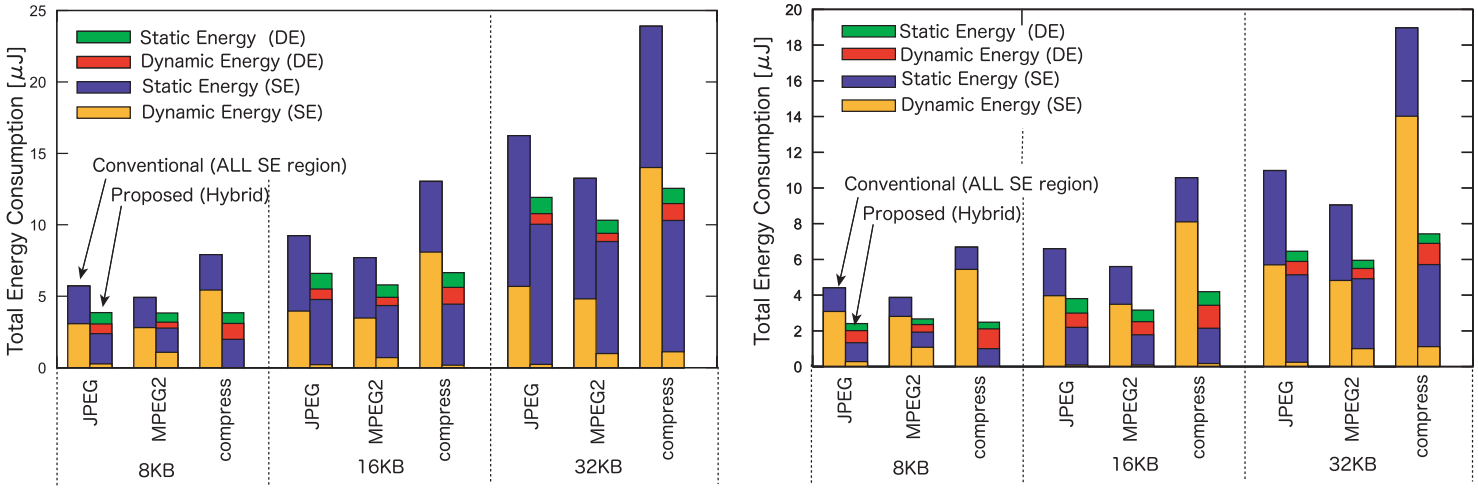


**Fig. 7** Experimental Result. Left Fig. is 200 MHz and Right Fig. is 400 MHz case.

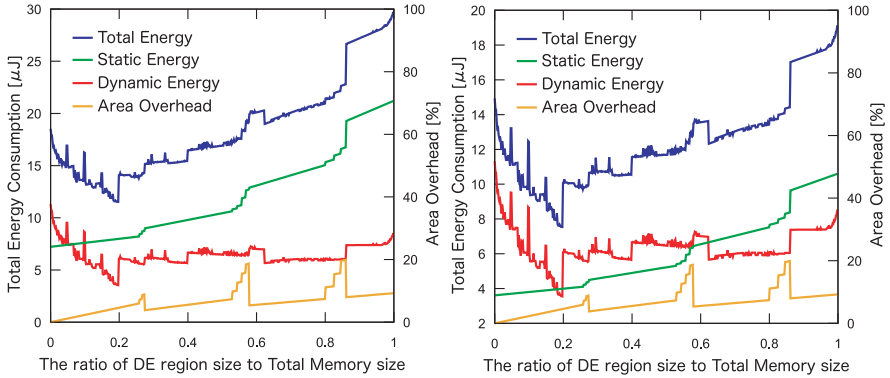


**Fig. 8** Energy breakdown for different ratios of DE region size to total size (8 KB). Left Fig. is 200 MHz and Right Fig. is 400 MHz case.

more effective at the higher frequency situation since the dynamic energy consumption ratio to the total energy consumption is larger. The major reason why the total energy consumption can be drastically reduced is that the optimal DE region size is very small due to a memory reference locality. Therefore, even if the low $V_{DD}$ is assigned, the delay constraint can be satisfied (see Fig. 3) and it suppresses the increase of the static energy consumption. Although the large energy saving can be obtained by applying our proposed technique, as described in Section 2, the DE region requires a large $\beta$ ratio cell to satisfy the SNM constraint and it enlarges the memory array area. However, the DE region of the optimal memory configuration is much smaller than the SE region. More specifically, the size of the DE region in the optimal 32 KB SPM configuration is only 7.1% of the total SPM size. Therefore, area overhead of the entire memory array area is tolerable. The experimental results show that our proposed technique exploits the reference locality effectively and it is very effective method for reducing the energy consumption of on-chip memory. The most important point is that our technique does not involve any performance, SNM and WM degradations.

## 5. Conclusion

Hybrid memory architecture and code allocation problem for the hybrid memory are proposed for minimizing the on-chip memory energy consumption under constraints of an SNM, a WM and an access delay. The proposed technique is

applied to SPM, and its effectiveness is demonstrated by simulations. The results show that our proposed technique can save the total energy consumption by 62.9% at the best case compared to the conventional memory with 4.56% memory array area overhead.

Although the SRAM cell is designed based on a logic circuits design rule in this paper, the SRAM cell is generally designed based on an SRAM specific design rule to enhance the memory density. It is one of the our future work to evaluate the proposed techniques using more practical design rule. We believe that the proposed concepts could be applicable to such a high density SRAM in principle. In this paper, the SNM degradation is compensated by enlarging the $\beta$ ratio. An 8T-SRAM cell also solves the SNM degradation problem [23]. Considering the optimal memory configuration including such an 8T-SRAM is also future work. In this paper, only memory array area overhead is discussed and any other peripheral circuits overhead is not discussed. Evaluating area and energy overheads for the peripheral circuits is also our future work.

## Acknowledgement

This work is supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys Corporation. This work is also supported by CREST ULP program of JST and JSPS Grant-in-Aid for Young Scientists (B) (20700049).

## References

1) Segars, S.: Low Power Design Techniques for Microprocessors, *IEEE International Solid-State Circuits Conference, Tutorial Note* (2001).
2) ARM Ltd.: ARM Processor Core Overview. http://www.arm.com/products/CPUs
3) Montanaro, J., et al.: A 160 MHz, 32 b 0.5 W CMOS RISC Microprocessor, *IEEE Journal of Solid-State Circuits*, Vol.31, No.11, pp.1703–1714 (1996).
4) Sakurai, T. and Newton, A.R.: Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas, *IEEE Journal of Solid-State Circuits*, Vol.25, No.2, pp.584–594 (1990).
5) Chang, H. and Sapatnekar, S.S.: Full-chip analysis of leakage power under process variations, including spatial correlations, *Proc. 42nd Annual Conference on Design Automation*, pp.523–528 (2005).
6) Hennessy, J.L. and Patterson, D.A.: *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann Publishers Inc. (2003).
7) Matsumura, T., Ishitobi, Y., Ishihara, T. and Yasuura, H.: A Hybrid Memory Architecture for Low Power Embedded System Design, *Proc. 14th Workshop on Synthesis and System Integration of Mixed Information Technologies*, pp.56–62 (2007).
8) Matsumura, T., Ishihara, T. and Yasuura, H.: Simultaneous optimization of memory configuration and code allocation for low power embedded systems, *Proc. 18th ACM Great Lakes symposium on VLSI*, pp.403–406 (2008).
9) Sakanaka, A., Fujii, S. and Sato, T.: A leakage-energy-reduction technique for highly-associative caches in embedded systems, *Proc. Workshop on Memory Performance*, pp.50–54 (2003).
10) Kawabe, N. and Usami, K.: Low-power technique for on-chip memory using biased partitioning and access concentration, *Proc. IEEE Custom Integrated Circuits Conference*, pp.275–278 (2000).
11) Ishihara, T. and Asada, K.: A system level memory power optimization technique using multiple supply and threshold voltages, *Proc. Conference on Asia South Pacific Design Automation*, pp.456–461 (2001).
12) Morifuji, E., Yoshida, T., Tsuno, H., Kikuchi, Y., Matsuda, S., Yamada, S., Noguchi, T. and Kakumu, M.: New guideline of Vdd and Vth scaling for 65 nm technology and beyond, *Symposium on VLSI Technology, Digest of Technical Papers*, pp.164–165 (2004).
13) Steinke, S., Wehmeyer, L., sik Lee, B. and Marwedel, P.: Assigning Program and Data Objects to Scratchpad for Energy Reduction, *Proc. Conference on Design, Automation and Test in Europe*, pp.409–415 (2002).
14) Banakar, R., Steinke, S., Lee, B.-S., Balakrishnan, M. and Marwedel, P.: Scratchpad memory: design alternative for cache on-chip memory in embedded systems, *Proc. 10th International Symposium on Hardware/software codesign*, pp.73–78 (2002).
15) Seevinck, E., List, F. and Lohstroh, J.: Static-Noise margin analysis of MOS SRAM cells, *IEEE Journal of Solid-State Circuits*, Vol.22, No.5, pp.748–754 (1987).
16) Tachibana, F. and Hiramoto, T.: Re-Examination of Impact of Intrinsic Dopant Fluctuations on Static RAM (SRAM) Static Noise Margin, *Japanese Journal of Applied Physics*, Vol.44, No.4B, pp.2147–2151 (2005).
17) Pelgrom, M.J.M.: Matching properties of MOS transistors, *Nuclear Instruments and Methods in Physics Research A*, Vol.305, pp.624–626 (1991).
18) Tsukamoto Y., et al.: Worst-case analysis to obtain stable read/write DC margin of high density 6T-SRAM-array with local Vth variability, *Proc. 2005 IEEE/ACM International Conference on Computer-Aided Design*, pp.398–405 (2005).
19) Yamaoka, M., et al.: 90-nm process-variation adaptive embedded SRAM modules with power-line-floating write technique, *IEEE Journal of Solid-State Circuits*, Vol.41, No.3, pp.705–711 (2006).
20) Takeda, K., Ikeda, H., Hagihara, Y., Nomura, M. and Kobatake, H.: Redefinition of

Write Margin for Next-Generation SRAM and Write-Margin Monitoring Circuit, *IEEE International Solid-State Circuits Conference, Digest of Technical Papers*, pp.2602–2611 (2006).
21) Morita, Y., et al.: A 0.3-V Operating, Vth-Variation-Tolerant SRAM under DVS Environment for Memory-Rich SoC in 90-nm Technology Era and Beyond, *IEICE Transaction Fundamental Electron Communication Computer Science*, Vol.E89-A, No.12, pp.3634–3641 (2006).
22) ILOG Inc.: CPLEX 9.1 Reference Manual 2005.
23) Chang, L., et al.: Stable SRAM cell design for the 32 nm node and beyond, *Proc. Symposium on VLSI Technology, Digest of Technical Papers*, pp.128–129 (2005).

(Received November 17, 2008)
(Revised February 20, 2009)
(Accepted April 13, 2009)
(Released August 14, 2009)

(Recommended by Associate Editor: *Takashi Sato*)

**Tadayuki Matsumura** received his B.S. and M.S. degrees in computer science from Kyushu University in 2007 and 2009, respectively. He currentry works for Hitachi Ltd., Kokubunji, Tokyo, Japan. His current research interests include embedded processor and low power SoC design.

**Tohru Ishihara** received his B.S., M.S., and Ph.D. degrees in computer science from Kyushu University in 1995, 1997 and 2000 respectively. From 1997 to 2000, he was a Research Fellow of the Japan Society for the Promotion of Science. For the next three years he worked as a Research Associate in VLSI Design and Education Center, the University of Tokyo. From 2003 to 2005, he worked at Fujitsu Laboratories of America as a research staff of an advanced CAD technology group. In 2005, he joined System LSI Research Center, Kyushu University as an Associate Professor. His research interests include low power SoC design and hardware/software co-design. He is a member of IPSJ, IEEE and ACM.

**Hiroto Yasuura** received the B.E., M.E., and Ph.D. degrees in computer science from Kyoto University. He was an associate professor in Kyoto University and moved to Kyushu University in 1991. He was a professor of Department of Computer Science and Communication Engineering, Graduate School of Information Science and Electrical Engineering of Kyushu University. He is currently a Trustee/Vice President of Kyushu University. He was involved in research of design methodology for VLSI system, CAD and hardware algorithm. He was a recipient of the Achievement Award from IEICE and Awards for Persons of Merit in Industry-Academia-Government Collaboration/Ministry of Education, Culture, Sports, Science and Technology Award in 2001 and 2007 respectively. He served as General Chair of ICCAD and ASP-DAC, and a Vice President of IEEE CAS Society. He also served as Director of IPSJ/IEICE. He is a Fellow of IPSJ.