A Variation-Aware 0.57-V Set-Associative Cache with Mixed Associativity Using 7T/14T SRAM

Jinwook Jung[1], Yohei Nakata[1], Shunsuke Okumura[1], Hiroshi Kawaguchi[1], and Masahiko Yoshimoto[1, 2]
[1]Graduate School of System Informatics, Kobe University, Japan [2]JST CREST, Japan
jung@cs28.cs.kobe-u.ac.jp

***Abstract*—In this paper, we present a novel cache scheme which efficiently reduces the minimum operating voltage (V_{min}) despite manufacturing-induced defective SRAM cells. The proposed low-voltage scheme exploits the fact that locations of defective SRAM cells are usually non-uniformly scattered. It also leverages the reliable characteristics of 7T/14T SRAM and allows associativites in each index to be different. Our evaluation results show that the proposed cache can reduce V_{min} of 64 KB 8-way set-associative cache by 80 mV within 7.81% capacity and 5.22% area overhead.**

I. INTRODUCTION

As the feature size in transistors continues to shrink along with the advance of process technology, we can integrate more and more devices onto a single chip. The transistor count of current state-of-the-art microprocessor reaches several billion. In this situation, power efficiency has become one of the most important design decisions even for high-performance SoC designs.

Lowering the supply voltage has been one of the most efficient techniques to reduce the power dissipation. However, ongoing technology scaling has led to increase manufacturing-induced variations in different device parameters. These variations create a considerable spread in a transistor threshold voltage, mainly because of random dopant fluctuation (RDF), which has deviation that is inversely proportional to the square of a channel area [1]. Such variations strongly impact functionality and reliability in deep sub-micron process technology [2].

This situation yields severe problems, particularly in SRAM, because minimum-sized transistors are used in its design. Device mismatch among transistors caused by process variations makes memory cell unreliable, which results in cell failures (read failure, write failure, and access time failure) [3]. In addition, the minimum operating voltage (V_{min}) tends to increase according to technology scaling [1]. To make matters worse, SRAMs occupy substantial fraction of the total die area and transistor count in processors [4]. Consequently, a large SRAM block such as on-chip caches determines the V_{min} of the entire processor.

Several studies of low voltage cache technique have been reported [5-6]. However, the low-voltage cache proposed in [5] imposes expensive on-chip fuses to identify the locations of defective cache lines. It also needs additional operating cycles and reduces cache capacity and associativity by half. The architecture proposed in [6] necessitates large additional circuitry and spare SRAM blocks.

In this paper, we develop a low voltage cache scheme for set-associativity cache which is named mixed associativity. The proposed scheme exploits the fact that each index's associativity in an *N*-way set-associative cache can be variable and manufacturing defects in SRAM cells are usually distributed randomly. It allows associativities in each index to be different and leverage the recovery feature of 7T/14T SRAM to achieve efficient low-voltage operations. Our scheme also easily combined with other repairing techniques for reliable low voltage operation such as ECCs.

II. MOTIVATION

Voltage scaling is one of the most effective methods to reduce a processor's power consumption. However, ever-increasing process variations cause SRAM circuits to fail at low voltages. Because caches in microprocessors consist of a great deal of SRAM cells, large caches are typically susceptible to process variations and cannot operate reliably in low voltage region. Therefore these large memory structures typically set V_{min} of the entire processor, which limits voltage scaling.

Intra-die random variations such as RDFs have a critical impact on SRAM cell reliability. As a result, manufacturing-induced defects in the entire cache are scattered. For this reason, it can efficiently reduce V_{min} of the entire cache to disable only defective SRAM cells. However, such fine-grained controls need expensive overheads.

Meanwhile, almost all the memory structures consist of many of subarrays. It is also the same in the case of cache design. Large caches in the modern processors are configured as many SRAM subarrays and peripherals. Figure 1 shows a physical organization example of cache data arrays in the *N*-way set-associative cache [7]. In this example, all the cache ways have four SRAM blocks and common peripheral circuitry. Because caches are already divided into some SRAM subarrays, it is quite easier and more efficient to eliminate the manufacturing-induced defects by some block-level controls.

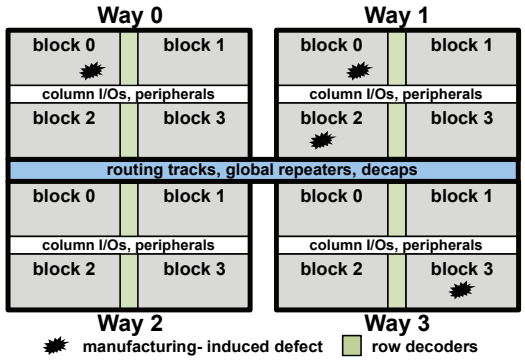


Figure 1. A physical organization of cache data arrays in a 4-way set-associative cache with manufacturing-induced defects.

978-1-4673-0821-2/12/$31.00 ©2012 IEEE

III. THE PROPOSED MIXED ASSOCIATIVITY SCHEME

In this section, we describe the proposed mixed associativity scheme. The proposed mixed associativity scheme exploits the characteristics of defective SRAM cell locations in memory structures and the block structure of the large caches. As mentioned above, manufacturing-induced defects in memory structures are distributed randomly, and caches are organized in several SRAM subarrays. Therefore V_{min} of the caches can be efficiently reduced by eliminating the defective SRAM cells on block-level control. In the proposed mixed associativity scheme, only SRAM subarrays, which include defective SRAM cells, are recovered. We will describe the fact that caches can work normally even after some SRAM subarrays have defects in this section.

The proposed scheme adopts the recovery feature of 7T/14T SRAM cell to efficiently lower the V_{min}. In addition, it is also possible to utilize half of cache lines in the defective SRAM subarrays comprising of the 7T/14T SRAM cells.

The proposed mixed associativity scheme addresses the cache data array. In general, the tag array has much smaller capacity and area than the data array. Therefore, in order to lower V_{min}s of the tag array, it is permissible to use larger SRAM cells, because it makes little impact on total cache area. In this paper, we assume that the tag array is implemented with larger 6T SRAMs which operate reliably at low voltages.

A. 7T/14T SRAM cell

We have proposed 7-Transistor/14-Transistor (7T/14T) SRAM [8]. Figure 2 shows a schematic of 7T/14T SRAM: two additional pMOS transistors (M20 and M21) are added between internal nodes (N00 and N10, N01 and N11) in a pair of the conventional 6-Transistor (6T) SRAM cells. This structure achieves the dependable mode which features margin enhancements by combining two memory cells especially in low-voltage regions.

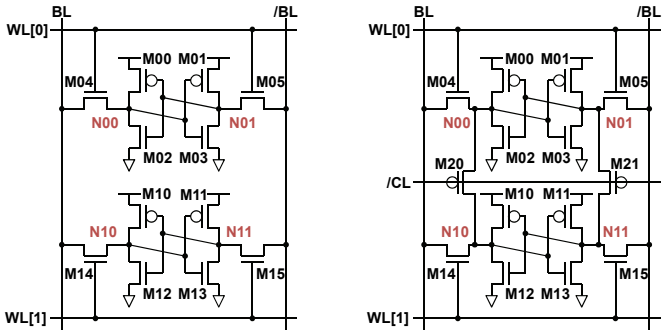


Figure 2. Schematics of (a) Conventional 6T SRAM and (b) 7T/14T SRAM

The 7T/14T SRAM has two modes as follows.

- Normal mode (7T): Additional transistors are turned off (CL="H"); 7T/14T SRAM acts as two 6T cells.
- Dependable mode (14T): Additional transistors are turned on (CL="L"); the internal nodes are shared by the memory cell pair. During write operations, both WL0 and WL1 are driven. On the other hand, either WL0 or WL1 is asserted during read operations.

In the normal mode, a one-bit data is stored in one memory cell, which is more area-efficient. In the dependable read mode, only one wordline is asserted to gain a large β ratio (a ratio of two driver transistors' total size to one access transistor size). An SRAM cell with no static noise margin [9] is recovered by the other SRAM cell through the two connecting pMOS transistors. In the dependable write mode, a data is written into a pair of memory cells by asserting both wordlines, which averages and mitigates the write margin degradation. In addition, the dependable mode has better soft-error tolerance because its internal node has more capacitance [10].

The dependable mode and normal mode can be switched according to the operating voltage, power limit, and required voltage margin in the current operating condition. If 7T/14T SRAM has sufficient operation margins, then its recovery feature can be inactivated by negating two additional pMOS transistors. In the dependable mode, a one-bit data is stored in two memory cells, although the quality of the information differs from that of the normal mode. We designate this concept as 'quality of a bit (QoB)', in which the operating voltage, power, and bit error rate (BER) are controlled as attributes of one-bit information [8].

B. Set-Associative Cache with Mixed Associativity

Figure 3 illustrates a 4-way set-associative cache including manufacturing-induced defective 6T SRAM cells, whose cache ways consist of four SRAM sub arrays. In the proposed mixed associativity scheme, SRAM blocks with defective SRAM cells are disabled (as marked red in Figure 3). In this situation, cache lines in the disabled SRAM blocks cannot be utilized, but the entire cache still has other cachlines to allocate.

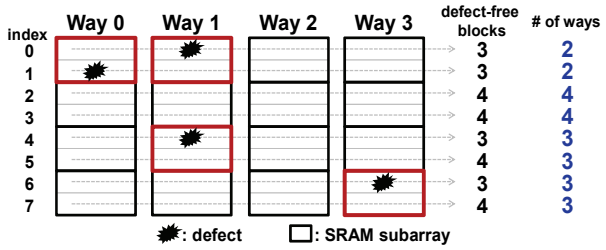


Figure 3. A logical organization example of 4-way set-associative cache with the proposed mixed associativity scheme using 6T SRAM

For example, in Figure 3, although cache lines in Way 0 and 1 cannot be allocated for index 0 and 1, we can still allocate data to cache lines in Way 2 and 3. Therefore, the entire caches can still work normally even if some SRAM subarrays are disabled. In this case, the associativity at each indexes are different from each other. In Figure 3, the cache operates as 2-way set-associative cache at index 0 and 1, 4-way cache at index 2 and 3, 3-way cache at index 4 to 7. Thus, the associativity in the proposed mixed associativity varies depending upon locations of manufacturing-induced defects.

However, if all cache ways on a specific index include defective SRAM cells, the entire cache cannot work properly because there are no cache lines to allocate at that index. Figure 4 depicts such situation. As all cache ways (Way 0 to 3) on index n and n+1 have defective SRAM cells, we cannot use any cache lines because SRAM blocks with defective cells are disabled in the proposed scheme.

978-1-4673-0821-2/12/$31.00 ©2012 IEEE

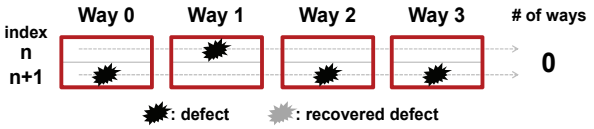


Figure 4. Mixed-associativity scheme with conventional 6T SRAM

C. Mixed Associativity with 7T/14T SRAM

In order to solve this problem, we use the 7T/14T SRAM cell in the proposed mixed associativity scheme. Figure 5 shows how to use 7T/14T SRAM in the proposed scheme; adjacent two cache lines (index *n* and *n+1*) in one cache way compose 7T/14T SRAM pair. Leveraging the recovery feature of 7T/14T SRAM, the dependable mode will be applied instead of merely disabling defective SRAM blocks. In this situation, the capacity of each SRAM blocks is halved. However, defective SRAM cells are recovered thereby non-defective cache lines can be obtained. The cache with mixed associativity therefore can operate normally by exploiting this half of recovered cache lines as shown in Figure 5. Odd-numbered cache ways allocate recovered cache lines to odd indexes, and even-numbered cache ways allocate these cache lines to even indexes. In other words, even indexes in odd-numbered ways and odd indexes in even-numbered ways are disabled in the dependable mode.

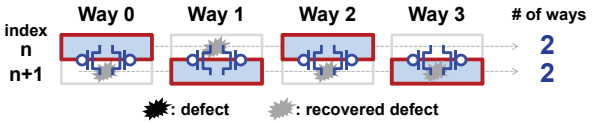


Figure 5. Mixed associativity scheme with 7T/14T SRAM

Figure 6 shows the entire logical organization of the proposed mixed associativity scheme using 7T/14T SRAM. Compared to the case of Figure 3, half of defective blocks are repaired and the average associativity is increased from 3 to 3.5.

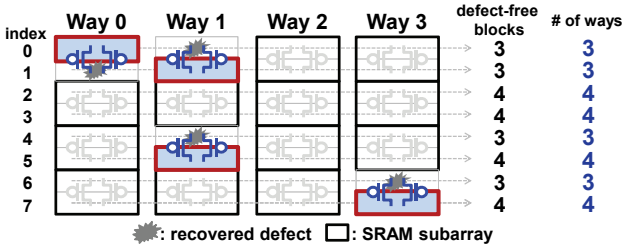


Figure 6. A logical organization example of 4-way set-associative cache with the proposed mixed associativity scheme using 7T/14T SRAM

D. Cache Operations (Cache Hit/Miss)

In the set-associative cache with the mixed associativity, there are both non-defective cache lines and disabled cache lines. Obviously, disabled cache lines cannot be used for allocating data. Thus, these cache lines have to be cut out in cache operations. It can be easily implemented by making a small change in the cache replacement algorithm. We use the Least Recently Used (LRU) algorithm which is most common as a cache replacement algorithm..

Figure 7 shows a modified LRU algorithm for the proposed scheme. In this algorithm, disabled cache lines' LRU tree bits are fixed to 0s, which means the most recently used (MRU) cache line. In the LRU algorithm, the LRU cache line will be replaced when a cache miss occurs (Figure 7 (a)). Thus, fixing its LRU tree bits to 0s, a disabled cache line is never used for cache replacement and allocating new data.

Figure 7 (b) describes the situation of a cache hit to Way 1, while Way 2 is disabled because it operates in the dependable mode. In the proposed scheme, no cache hit to disabled cache lines occurs, because of their *valid bit* fixed to 0 as described later in this section. Therefore there is no need to consider cache hits to disabled cache lines. It is only needed to modify LRU bits of a disabled cache line to 0 after a cache hit occurs.

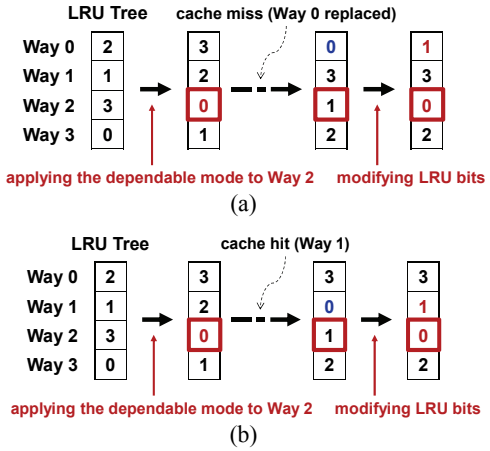


Figure 7. Cache operation examples with the modified LRU algorithm : (a) cache miss, (b) cache hit

This modified LRU algorithm has no cycle penalty because the LRU tree modification can be executed with other operations. It is needed to modify the LRU tree after updating the LRU tree. However, it can be carried out simultaneously with other operations. Therefore, there is no need to take additional cycles to modify LRU bits for disabled cache lines.

Disabled cache lines can be easily identified by using the cache status bits. Table II shows the truth table of cache status bits. In the typical write-back caches, the case of {*valid bit, dirty bit*} = {0, 1} cannot be reachable. Therefore, we can exploit this status to identifying disabled cache lines. In other words, assigning *valid bit* and *dirty bit* of disabled cache lines to {0, 1}, these disabled cache lines can be easily identified.

TABLE I. CACHE STATUS BITS

Valid bit	Dirty bit	Status
0	0	Empty
0	1	Invalid status
1	0	Clean
1	1	Dirty

E. Impacts on Processor Performance

As described above, the proposed scheme necessitates no additional cycles. It has an influence on performance mainly by reduced the entire cache associaitivity and capacity. However, in the large cache of recent microprocessors, the impacts on cache miss rates caused by decreased associativity and capacity are fairly small [11]. Therefore, reduced associativity and capacity of the proposed cache lead to relatively small performance overhead.

978-1-4673-0821-2/12/$31.00 ©2012 IEEE

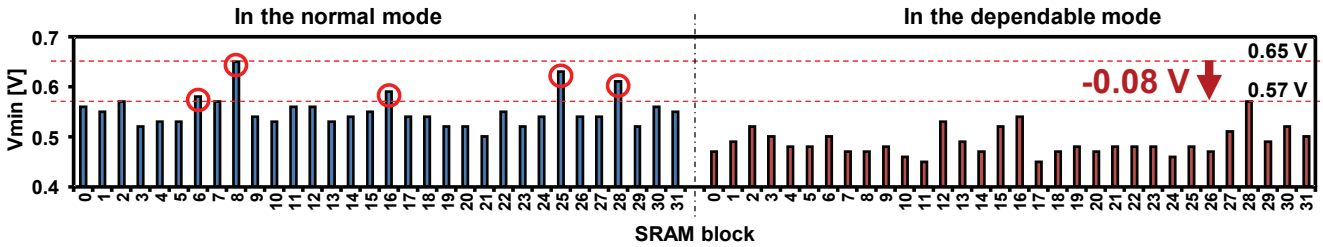


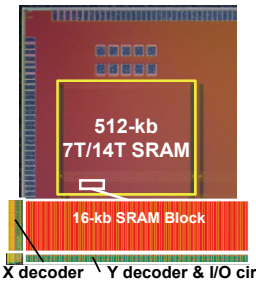
Figure 8. Measured V_{min}s of each 16-Kb SRAM blocks in a 512-Kb SRAM macro according to each operating mode

IV. Evaluation

In this section, we describe evaluation results of the proposed mixed associativity scheme.

A. Measurement Results

In order to evaluate the proposed mixed associativity, we manufactured 512-kb 7T/14T SRAM macro in a 65-nm CMOS technology, which consists of 32 16-kb SRAM blocks. Figure 9 shows a layout of 16-kb SRAM block and a die photograph of 512-kb 7T/14T SRAM macro.



Technology	65-nm CMOS 12 metal layers
Area	2.04 mm² (1260 um × 1620 um)
SRAM organization	16-kb block × 32 × 2 (= 512 Kb)
Block configuration	64 rows × 8 columns × 16 bits/word

Figure 9. 512-kb 7T/14T SRAM die photograph and 16-kb block layout

Figure 8 summarizes measurement results. The 512-kb 7T/14T SRAM can operate 0.57 V in the dependable mode. Thus, we can obtain 80 mV lower V_{min} by applying the dependable mode to SRAM blocks whose V_{min} is higher than 0.57 V (only five blocks marked by red circles in Figure 8). In this case, the entire capacity reduced to 472-kb, which means the proposed cache sacrifices cache capacity by only 7.81%.

B. Area Overhead

The proposed mixed associaitivity scheme uses 7T/14T SRAM cells in its data array and larger 6T SRAM cells in its tag array. The 7T/14T SRAM cell's area is 11% greater than that of the conventional 6T memory cell. We assume that the tag array is implemented with 1.3 times larger 6T SRAM cells. In 65-nm CMOS process, these larger 6T SRAM cells can operate reliably under 0.5 V. Based on the design parameters, we estimated the area overhead of the proposed cache in a 64-KB 8-way set-associative cache with CACTI [12]. In 65-nm process technology, the proposed cache's area overhead is 5.22%. Table II summarizes the area estimation results.

TABLE II. Area estiation results in 65-nm CMOS process

Scheme	Tag array (mm²)	Data array (mm²)	Total (mm²)	Norm. area
6T SRAM (Vmin=0.65 V)	0.0611	3.2422	3.3033	1
Proposed (Vmin=0.57 V)	0.0793	3.4060	3.4852	1.0522

V. Conclusion

In this paper, we propose the mixed associativity scheme using 7T/14T SRAM, which can reduce the minimum operating voltage of the entire cache efficiently. The proposed mixed associativity scheme allows associativities in each index to be various. It exploits the characteristics of manufacturing-induced defects in memory structures and the recovery feature of 7T/14T SRAM. The proposed scheme has no additional cycle penalty. According to our measurement results, the proposed scheme can reduce the minimum operating voltage by 80 mV. Area estimation results show that the area overhead of the proposed cache scheme is 5.22% in 64-KB cache 8-way set-associative cache.

References

[1] K. Itoh, "Adaptive Circuits for the 0.5-V Nanoscale CMOS Era," *IEEE International Solid-State Circuits Conference*, pp. 14-20, 2009.

[2] S. Borkar, T. Karnik, and V. De, "Design and realibility challenges in nanometer technologies," *ACM/IEEE Design Automation Conference*, pp. 75, 2004.

[3] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Modeling of failure probability and statistical design of SRAM array for yield enhancement in nanoscaled CMOS," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 12, pp. 1859-1880, 2005.

[4] E. J. Marinissen, B. Prince, and D. K.-S. Y. Zorian, "Challenges in embedded memory design and test," *Design, Automation and Test in Europe Conference and Exhibition*, pp. 722–727, 2005.

[5] C. Wilkerson, H. Gao, A. R. Alameldeen, Z. Chishti, M. Khellah, and S. -L. Lu, "Trading Off Cache Capacity for Reliability to Enable Low Voltage Operation," *IEEE International Symposium Computer Architecture*, pp 203-214, 2008

[6] J. Kim, N. Hardavellas, K. Mai, B. Falasafi, and J. C. Hoe, "Multi-Bit Error Tolerant Caches Using Two-Dimensional Error Coding," *ACM/IEEE International Symposium MicroArchitecture*, pp. 197-209, 2007

[7] J. L. Shin, B. Petrick, M. Singh, and A. S. Leon, "Design and Implementation of an Embedded 512-KB Level-2 Cache Subsystem," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 9, pp. 1815-1820, 2005.

[8] H. Fujiwara, S. Okumura, Y. Iguchi, H. Noguchi, H. kawaguchi and M. Yoshimoto, "A 7T/14T Dependable SRAM and Its Array Structure to Avoid Half Selection," *IEEE International Conference on VLSI Design*, pp. 295-300, 2009.

[9] E. Seevinick, F. J. List, and J. Lohstroh, "Static-Noise Margin Analysis of MOS SRAM Cells," *IEEE Journal of Solid-State Circuits*, vol. 22, no. 5, pp. 748-754, 1987.

[10] S. Yoshimoto, T. Amashita, S. Okumura, K. Yamaguchi, H. Kawaguchi, and M. Yoshimoto, "Bit Error and Soft Error Hardenable 7T/14T SRAM with 150-nm FD-SOI Process," *IEEE International Reliability Phisics Symposium(IRPS)*, pp. 876-881, 2011.

[11] J. Hennessy and D. Patterson, "Computer Architecture: a Quantitative Approach," Morgan Kaufmann Publishers, Inc., 5th edition, 2012

[12] N. Muralimanohar, R. Balasubramonian, and N. P. Jouppi, "CACTI 6.0," Technical Report HPL-2009-85, Hewlett Packard Labs, 2009

978-1-4673-0821-2/12/$31.00 ©2012 IEEE