PAPER Special Section on VLSI Design and CAD Algorithms

7T SRAM Enabling Low-Energy Instantaneous Block Copy and Its Application to Transactional Memory

Shunsuke OKUMURA[†a)], ***Student Member***, **Yuki KAGIYAMA**[†], ***Nonmember***, **Yohei NAKATA**[†], **Shusuke YOSHIMOTO**[†], ***Student Members***, **Hiroshi KAWAGUCHI**[†], ***and*** **Masahiko YOSHIMOTO**[††], ***Members***

SUMMARY This paper proposes 7T SRAM which realizes block-level simultaneous copying feature. The proposed SRAM can be used for data transfer between local memories such as checkpoint data storage and transactional memory. The 1-Mb SRAM is comprised of 32-kb blocks, in which 16-kb data can be copied in 33.3 ns at 1.2 V. The proposed scheme reduces energy consumption in copying by 92.7% compared to the conventional read-modify-write manner. By applying the proposed scheme to transactional memory, the number of write back cycles is possibly reduced by 98.7% compared with the conventional memory system.

key words: *SRAM DMA, transactional memory, checkpoint and recovery, multi-core processor*

1. Introduction

A multi-core processor has been exploited for parallel processing, on which the composition of the direct memory access (DMA) is generally used for data transfer between local memories on the cores and main memory, and/or within local memories [1], [2]. In a high-performance multi-core processor, memory transfer optimization plays an important role in achieving superior total performance. The bottleneck of data transfer bandwidth between memories remains, however, as one issue related to multi-core processors. In particular, the processing time penalty attributable to the latency arising from using the DMA becomes an important difficulty.

In the multi-core processer system, the concern with thread-level parallelism (TLP) programming has been also growing. In the TLP programming, each thread must keep memory coherence exclusively, because the memory is shared by the each core. To avoid the conflict of the memory access, the accessed addresses are locked by each thread, but, there is a possibility of deadlock. Furthermore, the lock policy causes the degradation of the parallelism. To address these problems, transactional memory [3]–[6] that does not leverage the lock algorithm was proposed.

The checkpoint-recovery [7], [8] are another effective usage for the multi-core processor system. In the checkpoint-recovery, a micro architectural state (process state) like registers and a stack area is backuped/resotred by copying at every checkpoint stage.

Manuscript received March 26, 2011.
Manuscript revised June 20, 2011.
[†]The authors are with Kobe University, Kobe-shi, 657-8501 Japan.
[††]The author is with JST, CREST, Kobe-shi, 657-8501 Japan.
a) E-mail: s-oku@cs28.cs.kobe-u.ac.jp
DOI: 10.1587/transfun.E94.A.2693

As described herein, to achieve high-speed data transfer in the multi-core processor system, we have proposed copiable 7T SRAM with a simultaneous data copying feature [9]–[11]. In this paper, we discuss the application of the copiable 7T SRAM.

In the next section, we will mention the proposed copiable 7T SRAM's overview. In Sect. 3, we explain a simultaneous data copying scheme which realizes high-speed and low-energy data copy. In Sect. 4, we discuss the speed of the proposed data copying scheme and tradeoff between peak current and cycle time. In Sect. 5, we show performance improvement in a transactional memory system. Section 6 exhibits measurement results. The final section summarizes this paper.

2. Copiable 7T SRAM Structure

The proposed SRAM can copy a datum between 7T/14T bitcell pair in both directions. This function namely realizes simultaneous block data copying feature without a bus. Figure 1 shows the proposed copy operation. In memory block A and B, different data is stored. Assuming that data of block A is transferred block B, generally data is read from block A, and its data is written to block B by each address with DMA and so on. This copying operation cycle proportionally increases in proportion to the block data size. On the other hand, in the proposed SRAM, a copying cycle does not depend on the block data size. First, data of block A and B are stored in bitcell pair, which are physically adja-

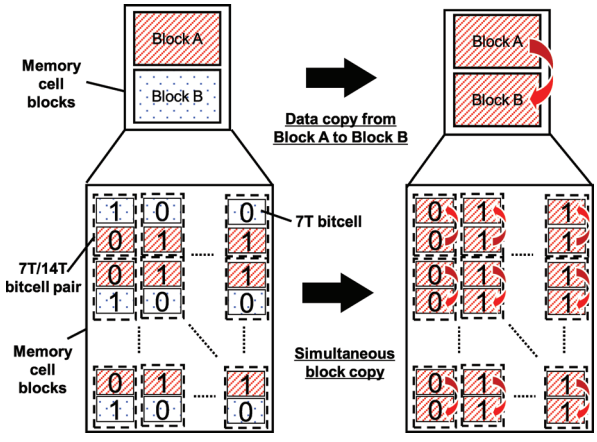

Fig. 1 Block-level simultaneous copying feature.

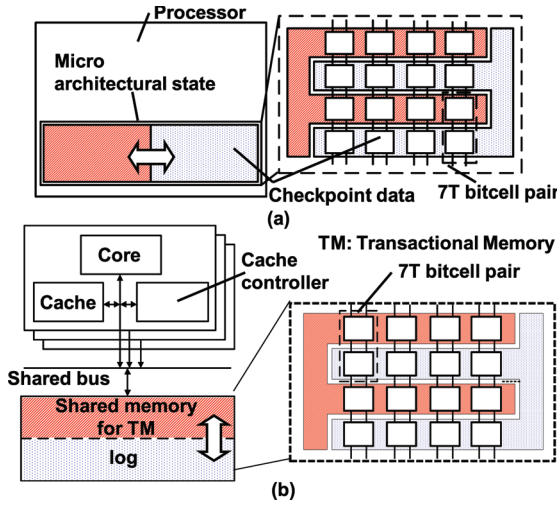

Fig. 2 Memory systems using copiable 7T SRAM: (a) checkpoint-recovery and (b) transactional memory.

cent cells. In the proposed copying operation, data transfer is simultaneously carried out at each bitcell pair, and it does not need read and write operation. Next, we will introduce applications which are effective for usage for the proposed copying SRAM.

To enhance reliability, it is desirable that a processing system has a function of checkpoint-recovery [7], [8]. A processer outputs operation results and copies it as checkpoint data. Then, the data are checked whether it is correct. If the results are incorrect, the processor restores the checkpoint data preserved before and starts recalculation; this is called rollback. Figure 2(a) shows the structure of the backup and recovery using the proposed SRAM. The proposed 7T bitcells can transfer data between bitcell pairs; that is, it can instantly backup the checkpoint data and also can instantly restore them in the rollback phase.

In a multi-core processor, use of transactional memory improves parallelism and enhances performance. A program result that was executed speculatively on each core is copied for backup to a log area [4]. In case where data conflict is happened, the backup data are written back to shared memory form the log area. Copying between the shared memory and log memory is normally done by DMA or software-copying via the shared bus, but the latency arising from the data transfer turns out to much. The proposed structure, which can be exploited as transactional memory, is presented in Fig. 2(b). The 7T bitcell pairs are shared by the shared memory and log memory without a shared bus. The memory copy can be conducted at high speed and low energy.

3. Simultaneous Copying Scheme with 7T Bitcell

Shown in Fig. 3 is layout and schematic of the proposed copiable 7T bitcells that makes such high-speed data copying possible. The 7T bitcells connect their internal nodes to each other using pMOS transistors. They can improve the

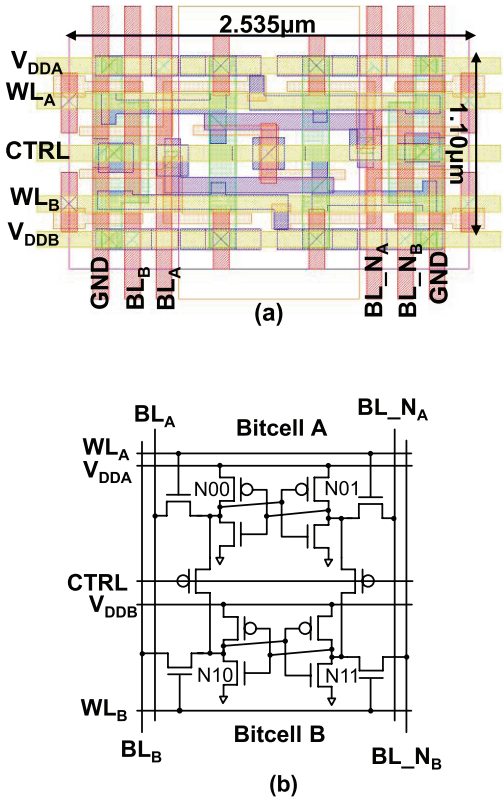

Fig. 3 Copiable 7T bitcell pair: (a) layout and (b) schematic.

reliability when one bit of datum is stored in the two bitcells [9]–[11]. We make use of the memory cell construction, but the supply voltages of the two bitcells are separated (V_{DDA} and V_{DDB}). Consequently, we can realize a data transition between the upper bitcell A and lower bitcell B. In addition, bitcells A and B can be independently accessed because bitlines are separated, which is suitable to the transactional memory. Due to the separated bitline structure, the area overhead becomes 43.7% compared with the conventional 6T bitcell.

The 7T bitcell pairs are controlled by a CTRL signal as well as by the separated supply voltages: when CTRL is "low", the bitcells are connected; when it is "high", they are separated. Simulated waveforms in a data copying sequence are presented in Fig. 4. Therein, a datum in bitcell A is copied to bitcell B (upper to lower). Four clock cycles are necessary for copying:

1. First, WLB is activated and BLB and VDDB are discharged. At this time, the internal nodes of bitcell B are destroyed.
2. Next, WLB is negated, and BLB is charged to the supply voltage. The preparation for data writing onto bitcell B in the subsequent clock cycle is done.
3. Then, CTRL is asserted and connects the upper bitcell A's internal nodes to the lower bitcell B's to write the bitcell A's datum to bitcell B (copying).
4. Finally, copying is completed by disabling CTRL and recharging VDDB. Data copying can be done in both

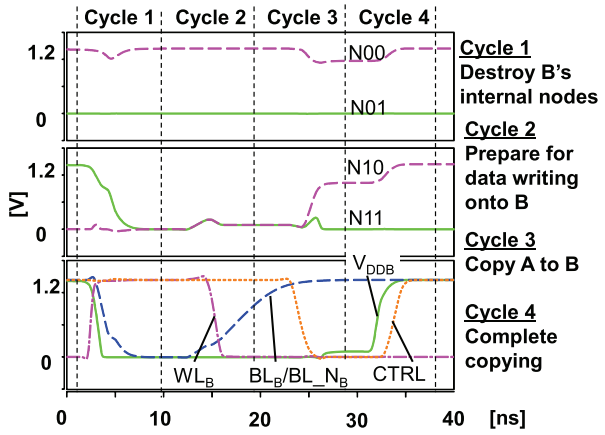

Fig. 4 Copy sequence waveform.

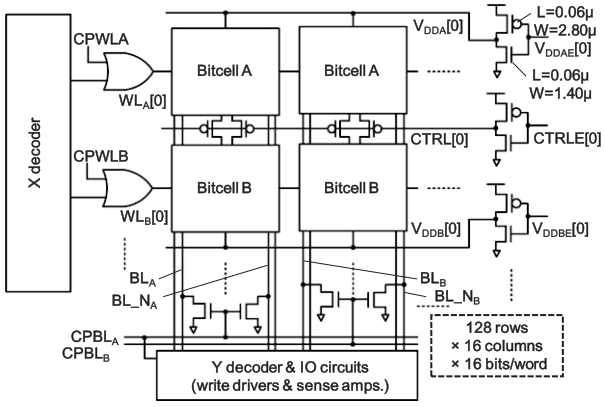

Fig. 5 Copiable SRAM scheme.

directions; copying from bitcell B to bitcell A is also possible by carrying out the opposite operation.

The peripheral circuits and bitcell pairs in the copiable SRAM are presented in Fig. 5. All bitcell pairs in a memory block can be copied simultaneously. In our design, the memory block size is 32 kb, and VDDA and VDDB for it are discharged and recharged by large buffers. Wirings of VDDA and VDDB are connected along with the row direction. Usually, VDDA and VDDB are at a supply voltage, but while copying, they are controlled on a block-by-block basis according to the above sequence. During copying, WLA and WLB are controlled as well, with OR gates by CPWLA and CPWLB signals aside from the output from the X decoder. When the CPBLB signal is "high", both bitlines (BLB, BL_NB) are discharged. The data size that can be copied simultaneously is scalable by using this structure. The time penalty for copying is independent of the size even if we increase the memory block capacitance.

4. Tradeoff between Peak Current and Cycle Number

In this section, the cycle time and peak current of the proposed copy scheme are described. Table 1 summarizes the configuration of the SRAM circuits in the following simulation. We use a 65-nm CMOS process technology, and assume a 32-kb SRAM block that has 128 rows × 8 columns × 32 bits/word. By applying the proposed copying scheme, 16-kb data are transferred at once.

Table 1 Configuration of copiable 7T SRAM block.

Process technology	65-nm CMOS process
SRAM block capacity	32 kb
Row	128
Column	8
Word	32 bit
Simultaneous copy unit	16 kb

Table 2 Copying cycle time in each sequence.

	Cycle 1	Cycle 2	Cycle 3	Cycle 4
Time [ns]	1.06	0.15	1.17	0.87

4.1 Cycle Time Simulation of the Proposed Simultaneous Copy Scheme

We evaluate copying time in the proposed copying scheme. It is comprised of four cycles, whose cycle time is calculated by HSPICE simulation. As shown in Fig. 5, this simulation is carried out, considering buffers for wordline, bitline, V_{DDA}/V_{DDB}, and CTRL signal. We define each cycle time as follows:

1. (Cycle 1) A period from a time at which a CPWLB rises to VDD to a time at which the both of the internal nodes of bitcell B are "low".
2. (Cycle 2) A period from a time at which CPWLB falls to GND to a time at which WL_B is discharged.
3. (Cycle 3) A period form a time at which CTRLE falls to GND to a time at which a datum in a bitcell B is written to a bitcell A. In other words, a time that the voltage difference of N10 and N11 is expanded to VDD/2 is the end of Cycle 2.
4. (Cycle 4) A period form a time at which V_{DDBE} falls to GND to a time at which the internal nodes of the bitcell A and B are charged to VDD and their data become stable.

We summarizes each cycle time in Table 2. The conditions are the SF corner, a temperature of −40°C and a supply voltage of 1.2 V, which corresponds to the worst case. The cycle time in Cycle 3 takes 1.17 ns and it restricts the copying sequence cycle. Namely, the result shows that the proposed copying scheme can execute 16-kb data copy in four cycles at 854.7 MHz.

Next, we will describe the merit of the copy cycle reduction. By changing the number of activated wordlines in data copying, the proposed copy scheme can also change the amount of copying data. As shown in Fig. 6, in a large amount of data copying like the 16 kb, the dramatic cycle reduction is achievable; in the conventional function, it takes 64 × 8 × 2 clock cycles (128/2 × 8 reads and 128/2 × 8

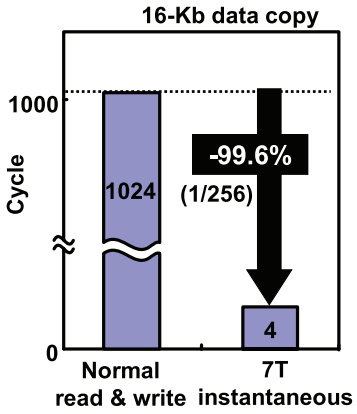

Fig. 6 Copy cycle reduction.

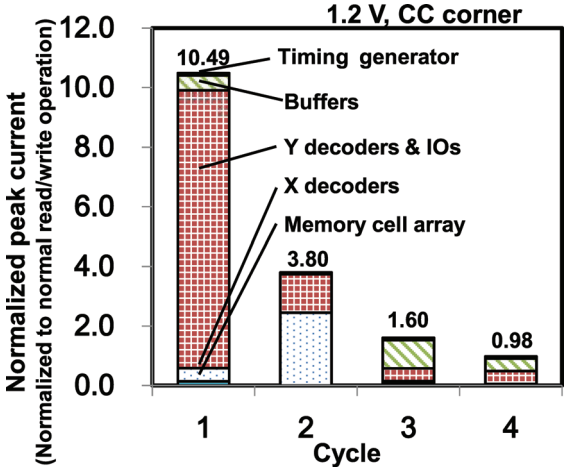

Fig. 7 Normalized peak current in the proposed copying sequence. Current in "buffers" is for VDDA/VDDB and CTRL signal. The nMOS drivers for bitline control are included in "Y decoders & IOs." The OR gates for wordline control are included "X decoders."

writes), although it can be reduced to only four cycles in our proposed scheme. In other words, the proposed SRAM achieves high-speed copying scheme.

4.2 Peak Current Simulation

In the proposed data copying scheme, the peak current should be estimated in real design, because a large amount of current flows through copying blocks and it might gives an impact on another circuits. Figure 7 shows normalized peak power in each cycle; the peak current in Cycle 1 is the largest. In particular, the Y decoders and IOs consume the large current because the bitline drivers for copying discharge all bitlines and all internal nodes in bitcells. Compared to the current in the normal read/write operation, Cycle 1 in the proposed copying scheme draws a 10-times lager current. We should consider suppressing the peak current in accordance with a current limit in chip design.

In Cycle 1, the peak current can be reduced by dividing the bitcell rows; however the number of cycles is increased. For example, when the rows in the bitcell array are split into

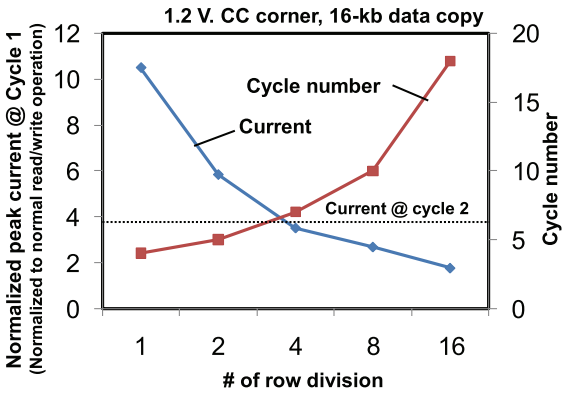

Fig. 8 Tradeoff between peak current and cycle number.

two, the bitline voltage is discharged by 64 rows and Cycle 1 is executed twice. In this case, the copying cycle number becomes five. Figure 8 shows the tradeoff between the peak current and cycle number. When the bitcell rows are divided into 16, the cycle number is increased to 19; however, the peak current can be suppressed to 1.61. When the row division number is 4 or more, the peak current in Cycle 2 turns out to be dominant in the proposed copying scheme. In such a case, as well as Cycle 1, the peak currents in Cycles 2 and 3 can be reduced by dividing columns or rows. This is a design choice; designers have to consider the tradeoffs between peak current and cycle number, depending on design demands.

5. Transactional Memory Application

In this section, we introduce the application using the proposed copying structure. The proposed copiable 7T SRAM can be applied to the log-based transactional memory (LogTM) system [7]. The LogTM is a hardware-based transactional memory (HTM) system, in which each core speculatively executes multiple threads to improve its parallelism. Each thread, however, may cause a conflict when it writes data to shared memory. To keep their consistency, when the threads access the shared memory, the write/read version is recorded in a location and they validate that the address is not changed by other threads. If a transactional thread detected a conflict, it cancels its process and invalidates the results (called "abort"). If there was no conflict until the thread finishes, the write version written in the locations is cleared and fixed (called "commit").

Figures 9 and 10 show an example of flow in the LogTM. When the thread stores data to the shared memory, the old data is backed up in the log region (Figs. 9(a) and 10(a)). If the thread committed, the log region is cleared (Figs. 10(b)). On the other hand, when the thread aborts, the log region are written back to the shared memory (Figs. 9(b) and 10(c)). At this time, the thread must execute a large number of write backs, which degrades the performance.

To address this problem, we apply the proposed copying scheme to the transactional memory (the shared memory and log region). As shown Fig. 10, the proposed structure

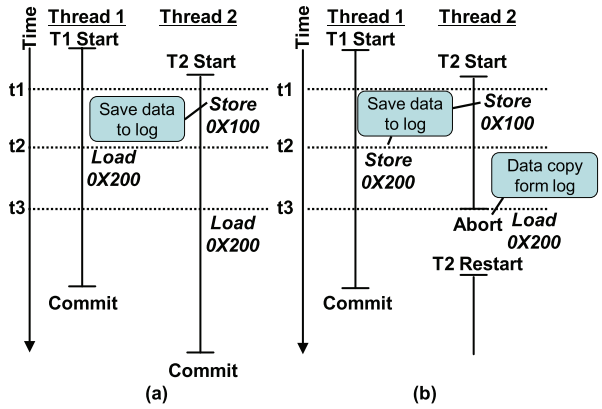

Fig. 9 Thread flows: (a) commit transaction and (b) abort transaction.

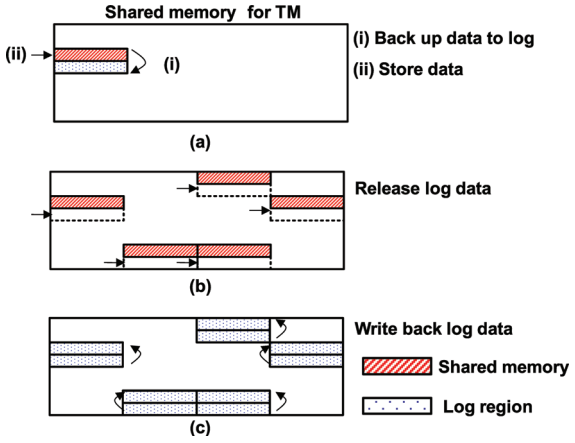

Fig. 10 Memory protocol images in transactions: (a) store, (b) commit and (c) abort.

Table 3 Benchmarks for transaction estimations.

	intruder	kmeans	yada
# of transaction	24193947	483865	4757773
# of aborts	765820	134333	2242386
# of commits	23529293	349534	2520099
# of write backs	351160	16	12439987

enables the simultaneous data copy and reduces the write back cost.

We adopted STAMP (Stanford transactional applications for multi-processing) [12] for estimating the transactional memory with the proposed copying scheme. The STAMP has benchmarks for a transactional memory system and HTM system like the LogTM. In this paper, we describe the results of "intruder", "kmeans", and "yada" that have different characteristics in the transactional process; Table 3 summarizes the results of the all transactions process. The table shows the total number of write backs, as well. Even in a single benchmark, the number of stored data in the log region also differs among threads. If the abort thread does not execute any store instructions, the write backs not need to be conducted. In an abort thread that causes a large amount

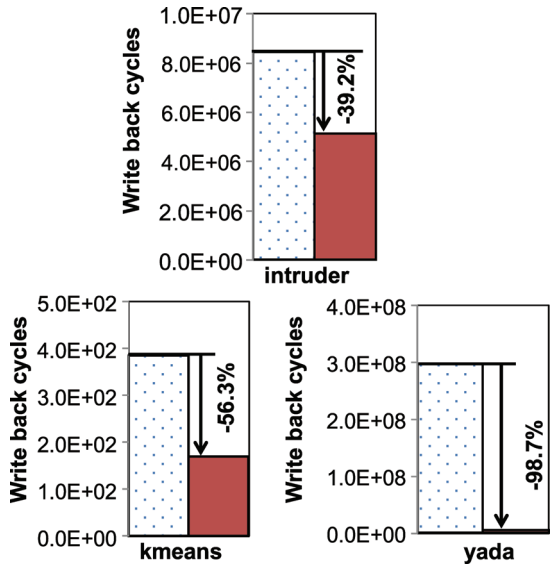

Fig. 11 Savings in write back cycles.

of write backs, the proposed scheme is expected to reduce the cost of the write backs. This is because, in the conventional system, it takes certain cycles that are proportion to the log data size in the DMA manner. On the other hand, in the proposed scheme, the number of copying cycles does not depend on the log data size, and thus can reduce the number of write backs.

Figure 11 shows the savings of the write backs in the LogTM system with the proposed scheme. In the "yada" program that has the highest percentage of the abort threads, the number of write backs is reduced by 98.7% compared with the conventional memory system. As well, in the "intruder" and "kmeans" programs, the write back costs are reduced by 39.2% and 56.3%, respectively.

As shown above, the proposed scheme can reduce the data copying cycles in a program which executes a large amount of data copy, such as transactional memory or check point and recovery system.

6. Measurement Result

The proposed SRAM with the copy function has been implemented in a 65-nm CMOS process. Figure 12 shows a layout of a 32-kb SRAM block and a photograph of a 1-Mb 7T SRAM test chip and the layout of a 32-kb memory block. The 1-Mb SRAM comprises 32 kb × 32 blocks (one memory block is 128 rows × 8 columns × 32 bit/words). The Area overhead of peripheral circuits including VDDA/VDDB and CTRL buffers (Copy buffers) is 0.12%.

A measured Shmoo plot of the copy function is shown in Fig. 13. We verified a 0.475-V copy operation at an operating frequency of 1 MHz. Additionally, we confirmed, at 120 MHz, the copy functions at 1.2 V, in which case data copying of 16 kb (a half of 32 kb) takes only 33.3 ns (= four clock cycles of 120 MHz).

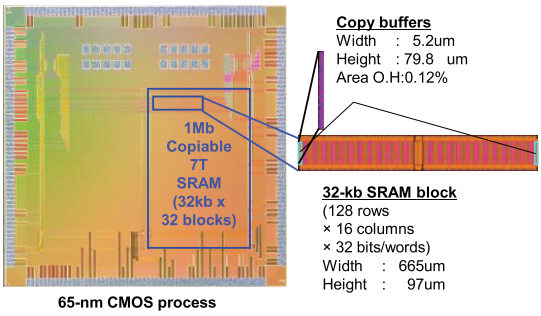

Fig. 12 Die photograph and layout.

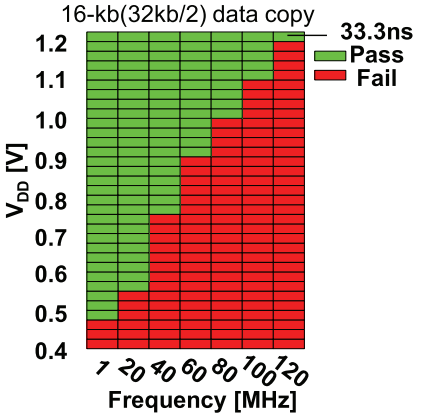

Fig. 13 Measured copy function Shmoo plot.

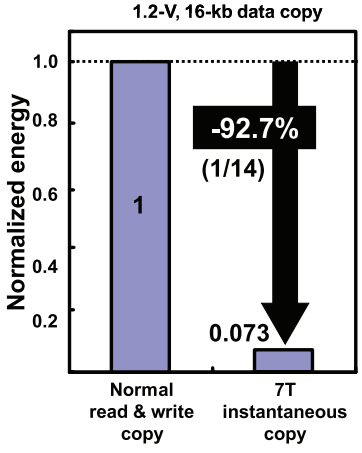

Fig. 14 Measured copy energy consumption.

The measured energy consumption is depicted in Fig. 14. Using the proposed copying scheme, copying all data is possible merely by charging and discharging a half of all wordlines, all bitlines, and all CTRL signals at once. In contrast, in a normal read-modify-write copy, it is impossible to read and write all the data at once. Reading and writing data in different columns incurs multiple charging and discharging of the wordlines and bitlines. Therefore, the proposed scheme makes it possible to reduce energy consumption in copying by 92.7% (1/14) compared to conventional read-modify-write functions.

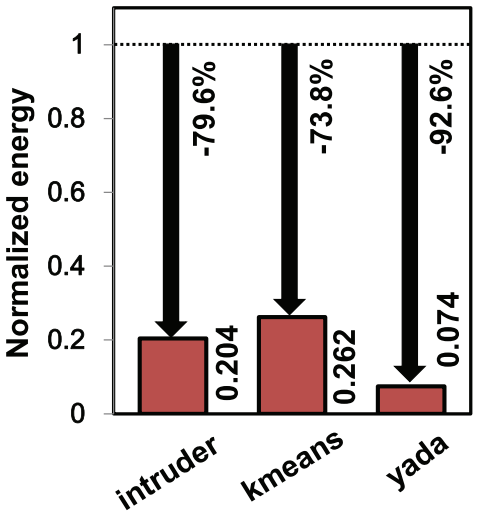

Fig. 15 Energy reductions in transactional memory.

In addition, we evaluated the energy reduction when applying our designed copiable SRAM to the transactional memory. Figure 15 shows the energy consumptions and energy reduction ratios in write back operation. The results are respectively normalized by the energies in the normal read-modify-write copies. The data sizes in the write back operation were calculated by the STAMP. The energy consumptions were estimated on the basis of the measured results (Fig. 14). The energy consumption in the proposed scheme is decreased with the number of write backs and its data size; effective energy reduction is possible in a large number of write backs and a large data size. In the benchmarks, "intruder", "kmeans", and "yada", the energy consumptions are reduced by 79.6%, 73.3%, and 92.6%, respectively.

7. Conclusion

We designed a 7T SRAM which realizes block-level simultaneous copying feature in 65-nm process technology. The 1-Mb SRAM comprises 32 kb × 32 blocks. In the copiable SRAM, 16-kb data can be copied in 33.3 ns at 1.2 V, 120 MHz. The copying function of proposed SRAM is suitable for data transfer between local memories such as checkpoint data storage and transactional memory. We confirmed that the proposed copying scheme possibly reduces the number of write backs and energy consumption by 98.7% and 92.6% respectively, when applied as a transitional memory. The proposed scheme reduces and a cycle time in copying by 99.6%, compared to the conventional read-modify-write manner.

Acknowledgments

The VLSI chip in this study was fabricated in the chip fabrication program of VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with STRAC, e-Shuttle, Inc., and Fujitsu Ltd.

References

[1] M. Kistler, M. Perrone, and F. Petrini, "Cell multiprocessor communication network: Built for speed," IEEE Micro, vol.26, no.3, pp.10–23, May-June 2006.
[2] Y. Dou, L. Deng, J. Xu, and Y. Zheng, "DMA performance analysis ad multi-core memory optimization for swim benchmark on the cell processor," International Symposium on Parallel and Distributed Processing with Applications, pp.170–179, Dec. 2008.
[3] L. Hammond, V. Wong, M. Chen, B.D. Carlstorm, J.D. Dabvis, B. Hertzberg, M.K. Prabhu, H. Wijaya, C. Kozyrakis, and K. Olukotun, "Transactional memory coherence and consistency," Proc. 31st International Symposium on Computer Architecture 2004, pp.102–113, June 2004.
[4] J. Bobba, N. Goyal, M. Hill, M.M. Swift, and D.A. Wood, "TokenTM: Efficient execution of large transactions with hardware transactional memory," Proc. 35th International Symposium on Computer Architecture, pp.127–138, June 2008.
[5] K.E. Moore, J. Bobba, M.J. Moravan, M.D. Hill, and D.A. Wood, "LogTM: Log-based transactional memory," 12th International Conference on High-Performance Computer Architecture, Feb. 2006.
[6] D. Dice, O. Shalev, and N. Shavit, "Transactional locking II," DISC'06: Proc. 12th International Conference on Architectural Support for Programming Languages and Operating Systems, Oct. 2006.
[7] T.J. Slegel, R.M. Averill, III, M.A. Check, B.C. Giamei, B.W. Krumm, C.A. Kryugowski, W.H. Li, J.S. Liptay, J.D. MacDougall, T.J. McPherson, J.A. Navarro, E.M. Schwarz, K. Shum, and C.F. Webb, "IBM's S/390 G5 MICROPROCESSOR DESIGN," IEEE Micro, vol.19, no.2, pp.12–23, March-April 1999.
[8] K.-L. Wu, W.K. Fuchs, and J.H. Patel, "Error recovery in shared memory multiprocessors using private caches," IEEE Trans. Parallel Distrib. Syst., vol.1, no.2, pp.231–240, April 1990.
[9] H. Fujiwara, S. Okumura, Y. Iguchi, H. Noguchi, Y. Morita, H. Kawaguchi, and M. Yoshimoto, "Quality of a Bit (QoB) a new concept in dependable SRAM," International Symposium on Quality Electronic Design, pp.98–102, March 2008.
[10] 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," VLSI Design 2009, pp.295–300, Jan. 2009.
[11] H. Fujiwara, S. Okumura, Y. Iguchi, H. Noguchi, H. Kawaguchi, and M. Yoshimoto, "A dependable SRAM with 7T/14T memory cells," IEICE Trans. Electron., vol.E92-C, no.4, pp.423–432, April 2009.
[12] C.C. Minh, J.W. Chung, C. Koazyrakis, and K. Olukotum, "STAMP: Stanford transactional applications for multi-processing," IISWC'08, Proc. IEEE International Symposium on Workload Characterization, pp.35–46, Sept. 2008.

Shunsuke Okumura received his B.E. and M.E. degrees in Computer and Systems Engineering in 2008 and 2010, respectively from Kobe University, Hyogo, Japan, where he is currently working in the doctoral course. His current research is high-performance, low-power SRAM designs, dependable SRAM designs, and error correcting codes implementation. He is a student member of IPSJ and IEEE.

Yuki Kagiyama was born on August 6, 1987. He received his B.E. degree in Computer anc Systems Engineering in 2010 from Kobe University, Hyogo, Japan, where he is currently working in the M.E. course. His current research is dependable SRAM designs.

Yohei Nakata received the B.E. and M.E. degrees in computer and systems engineering from Kobe University, Hyogo, Japan in 2008 and 2010, respectively, where he is currently pursuing the Ph.D. degree in engineering. His current research interests include multi-core processor architecture, low-power processor and dependable processor designs. He is a student member of the IEEE.

Shusuke Yoshimoto received a B.E. degree in Computer and Systems Engineering from Kobe University, Hyogo, Japan, in 2009. He is currently working in the master course at the same university. His current research is soft-error tolerant and low-power SRAM designs. He is a student member of IEEE.

Hiroshi Kawaguchi received his B.E. and M.E. degrees in Electronic Engineering from Chiba University, Chiba, Japan, in 1991 and 1993, respectively, and earned a Ph.D. degree in Engineering from The University of Tokyo, Tokyo, Japan, in 2006. He joined Konami Corporation, Kobe, Japan, in 1993, where he developed arcade entertainment systems. He moved to the Institute of Industrial Science, The University of Tokyo, as a Technical Associate in 1996, and was appointed as a Research Associate in 2003. In 2005, he moved to Kobe University, Kobe, Japan. Since 2007, he has been an Associate Professor with the Department of Information Science at that university. He is also a Collaborative Researcher with the Institute of Industrial Science, The University of Tokyo. His current research interests include low-voltage SRAM, RF circuits, and ubiquitous sensor networks. Dr. Kawaguchi was a recipient of the IEEE ISSCC 2004 Takuo Sugano Outstanding Paper Award and the IEEE Kansai Section 2006 Gold Award. He has served as a Design and Implementation of Signal Processing Systems (DISPS) Technical Committee Member for IEEE Signal Processing Society, as a Program Committee Member for IEEE Custom Integrated Circuits Conference (CICC) and IEEE Symposium on Low-Power and High-Speed Chips (COOL Chips), and as a Guest Associate Editor of IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences and IPSJ Transactions on System LSI Design Methodology (TSLDM). He is a member of the IEEE, ACM and IPSJ.

Masahiko Yoshimoto earned his B.S. degree in Electronic Engineering from Nagoya Institute of Technology, Nagoya, Japan, in 1975, and the M.S. degree in Electronic Engineering from Nagoya University, Nagoya, Japan, in 1977. He earned a Ph.D. degree in Electrical Engineering from Nagoya University, Nagoya, Japan in 1998. He joined the LSI Laboratory, Mitsubishi Electric Corporation, Itami, Japan, in April 1977. During 1978–1983 he was engaged in the design of NMOS and CMOS static RAM including a 64K full CMOS RAM with the world's first divided-wordline structure. From 1984, he was involved in research and development of multimedia ULSI systems for digital broadcasting and digital communication systems based on MPEG2 and MPEG4 Codec LSI core technology. Since 2000, he has been a Professor of the Department of Electrical and Electronic Systems Engineering at Kanazawa University, Japan. Since 2004, he has been a Professor of the Department of Computer and Systems Engineering at Kobe University, Japan. His current activities are focused on research and development of multimedia and ubiquitous media VLSI systems including an ultra-low-power image compression processor and a low-power wireless interface circuit. He holds 70 registered patents. He served on the Program Committee of the IEEE International Solid State Circuit Conference from 1991 to 1993. Additionally, he served as a Guest Editor for special issues on Low-Power System LSI, IP, and Related Technologies of IEICE Transactions in 2004. He received R&D100 awards in 1990 and 1996 from R&D Magazine for development of the DISP and development of a real-time MPEG2 video encoder chipset, respectively.