7T SRAM Enabling Low-Energy Simultaneous Block Copy

Shunsuke Okumura[1], Shusuke Yoshimoto[1], Kosuke Yamaguchi[1], Yohei Nakata[1],
Hiroshi Kawaguchi[1], and Masahiko Yoshimoto[1, 2]
[1]Kobe University, [2]JST, CREST

***Abstract* - 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 1.2V. The proposed scheme reduces energy consumption in copying by 92.7% compared to the conventional read-modify-write manner.**

I. Introduction

Memory latency is an important factor in performance. Particularly, the processing time penalty attributable to the latency arising from DMA (dynamic memory access)[1-2] becomes an important difficulty. To achieve high-speed data transfer, we propose 7T SRAM with a simultaneous data copying feature. The proposed SRAM can copy data between 7T bitcell pairs, and can be used for data transfer between local memories.

"Checkpoint-recovery" [3-4] and "transactional memory" [5-6] are effective usages for the copiable 7T SRAM[7-9]. 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. The transactional memory in a multi-core processor also uses data copying to synchronize each core.

II. Copiable 7T SRAM Structure and its application

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. Fig. 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 direct memory access (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 is not depend on the block data size. First, data of block A and B is stored in bitcell pair, which are physically adjacent 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.

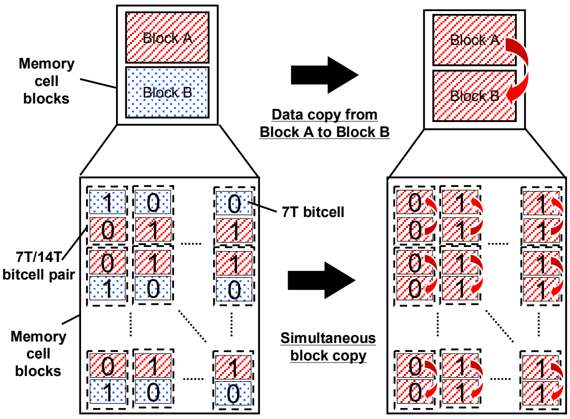

Fig.1 Block-level simultaneous Copying Feature.

To enhance reliability, it is desirable that a processing system has a function of checkpoint-recovery [3-4]. 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. Fig. 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.

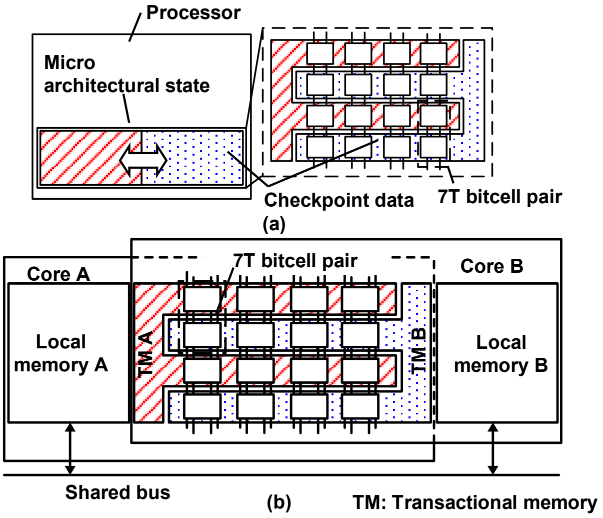

Fig.2 Memory systems using copiable 7T SRAM: (a) checkpoint-recovery and (b) transactional memory.

978-1-4244-5760-1/10/$26.00 ©2010 IEEE

In a multi-core processor, use of transactional memory improves parallelism and enhances performance [5-6]. A program result that was executed speculatively on each core must be broadcast to others through its transactional memory; the transactional memories keep multi cores consistent. Copying between transactional memories is normally done by DMA or software-copying via the shared bus, but the latency arising from such modes of data transfer turns out to have too much time overhead. The proposed structure, which can be exploited as transactional memory, is presented in Fig. 2 (b). The 7T bitcell pairs are shared by two cores and connected without a shared bus. The transactional memories can be copied each other at high speed and low energy.

III. Simultaneous Copying Scheme with 7T Memory Cell

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 reliability when one bit of datum is stored in the two bitcells [7-9]. We make use of the memory cell construction, but the supply voltages of the two bitcells are separated (VDDA and VDDB). 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.

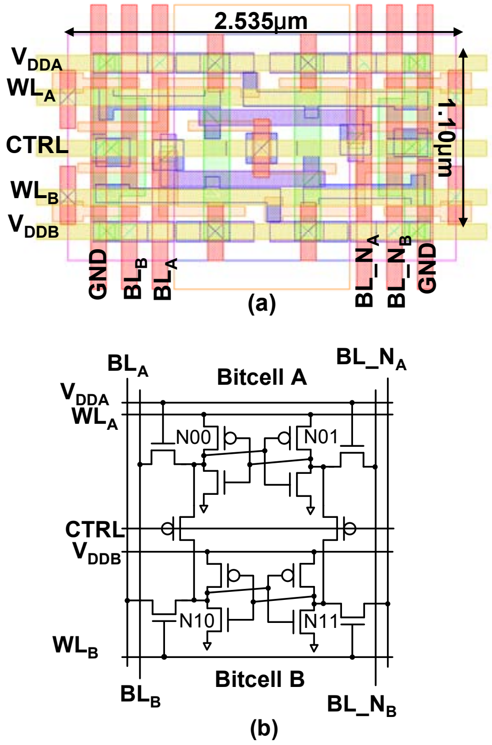

Fig.3 Copiable 7T bitcell pair: (a) layout and (b) schematic.

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

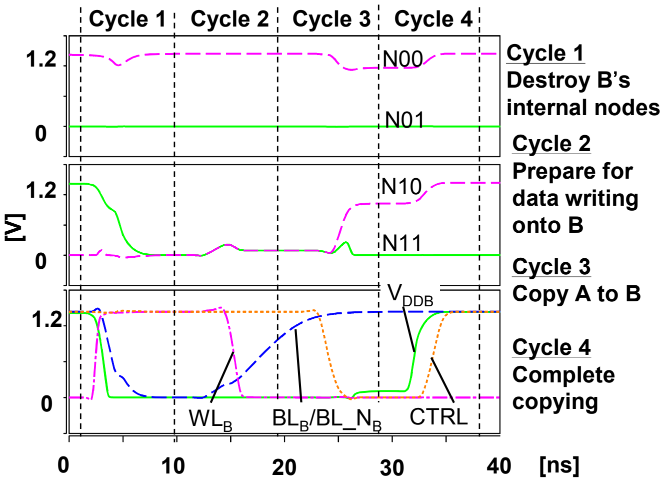

Fig.4 Copy sequence waveform.

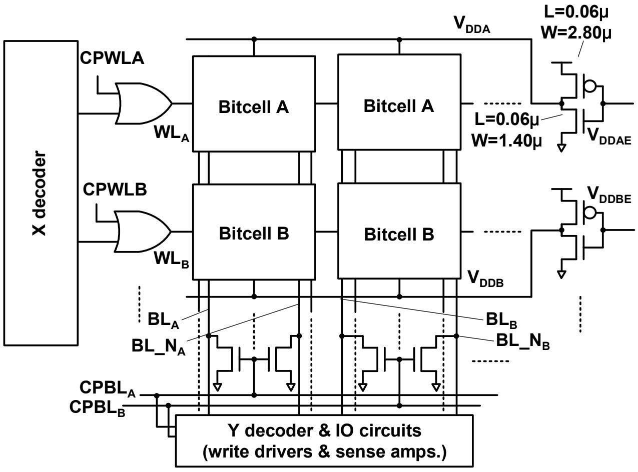

Fig.5 Copiable SRAM Scheme.

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 NOR 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.

IV. Results

The proposed SRAM with the copy function has been implemented in a 65-nm CMOS process. Fig. 6 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). Area overheads of peripheral circuits including VDDA/VDDB and CTRL buffers (Copy buffers) is 0.12%.

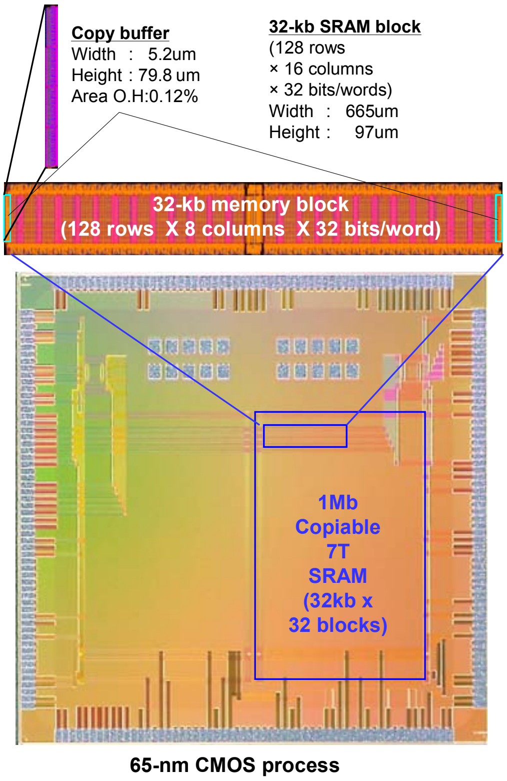

Fig.6 Die photograph and layout.

A measured Shmoo plot of the copy function is shown in Fig. 7. We verified a 0.475-V copy operation at an operating frequency of 1 MHz. Additionally, we confirmed, at 120MHz, 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).

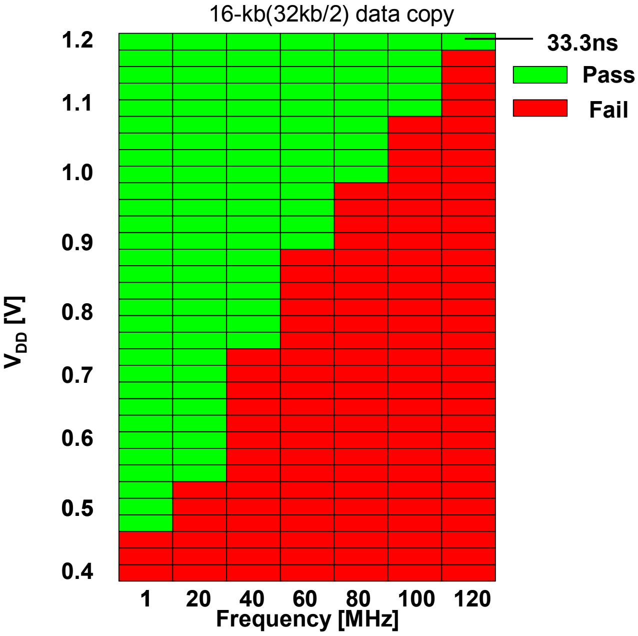

Fig.7 Measured copy function Shmoo plot.

The measured energy consumption and cycle time is depicted in Fig. 8 (a) and (b). 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. In addition, the clock cycle is reduced dramatically; in conventional functions, it takes 64 × 8 × 2 clock cycles (64 × 8 reads and 64 × 8 writes), although it can be reduced to only four clock cycle in our scheme. In other words, the proposed SRAM achieves high-speed copying scheme.

V. Conclusion

We designed a 7T SRAM which realizes block-level simultaneous copying feature in 65-nm process technology. The copying function of proposed SRAM is suitable for data transfer between local memories such as checkpoint data storage and transactional memory. The 1-Mb SRAM comprises 32 kb × 32 blocks. In the copiable SRAM, 16-kb data can be copied in 33.3 ns at 1.2V, 120 MHz. The proposed scheme reduces energy consumption and cycle time in copying by 92.7% and 99.6% respectively, compared to the conventional read-modify-write manner.

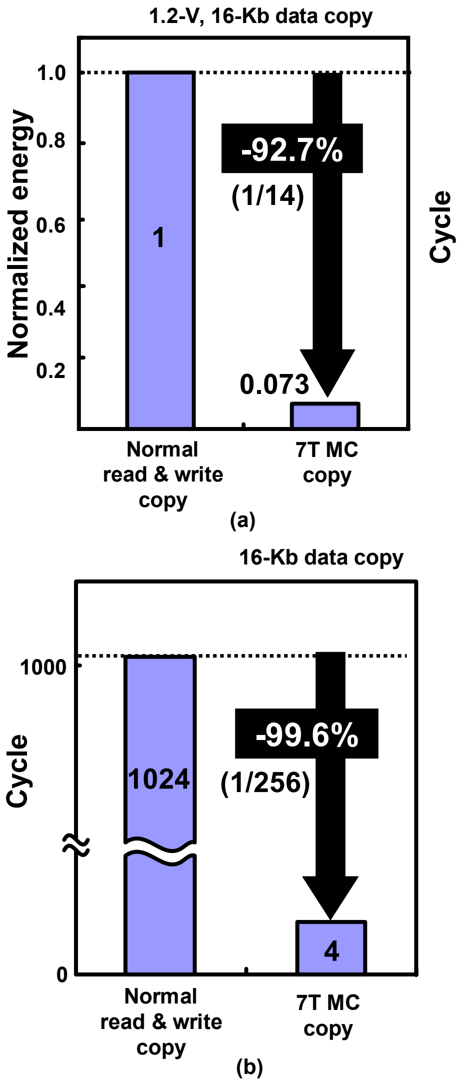

Fig.8 Measured (a) copy power consumption and (b) copy cycle.

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] Michael Kistler, Michael Perrone, Fabrizio Petrini, “Cell Multiprocessor Communication Network: Built for Speed”, Micro, IEEE, pp10-23, May-June, 2006 .

[2] Yong Dou, Lin Deng, Jinhui Xu, Yi 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, pp170-179, December, 2008.

[3] Timothy J. Slegel, Robert M. Averill III, Mark A. Check, Bruce C. Giamei, Barry W. Krumm, Christopher A. Kryugowski, Wen H. Li, John S. Liptay, John D. MacDougall, Thomas J. McPherson, Jennifer A. Navarro, Eric M. Schwarz, Kevin Shum, and Charles F. Webb, “IBM’s S/390 G5 MICROPROCESSOR DESIGN” Micro, IEEE, Volume 19, Issue 2, pp. 12 – 23, March-April 1999

[4] Kun-Lung Wu, W. Kent Fuchs, Janak H. Patel, “Error Recovery in Shared Memory Multiprocessors Using Private Caches”, IEEE Transactions on Parallel ad DistributedSystems pp231-240, April, 1990.

[5] Lance Hammond, Vicky Wong, Mike Chen, Brian D. Carlstorm, John D. Dabvis, Ben Hertzberg, Manohar K. Prabhu, Honggo Wijaya, Christos Kozyrakis, and Kunle Olukotun., “Transactional Memory Coherence and Consistency”, Proc. of 31st International Symposium on Computer Architecture 2004, pp. 102–113, June, 2004.

[6] Jayaram Bobba, Neelam Goyal, Mark \d. Hill, Michael M. Swift and David A. Wood, “TokenTM: Efficient Execution of Large Transactions with Hardware Transactional Memory”, Proceedings of the 35th International Symposium on Computer Architecture, pp.127-138, June, 2008.

[7] Hidehiro Fujiwara, Shunsuke Okumura, Yusuke Iguchi, Hiroki Noguchi, Yasuhiro Morita, Hiroshi Kawaguchi and Masahiko Yoshimoto, ” Quality of a Bit (QoB) A New Concept in Dependable SRAM”, International Symposium on Quality Electronic Design, pp. 98–102, March, 2008.

[8] Hidehiro Fujiwara, Shunsuke Okumura, Yusuke Iguchi, Hiroki Noguchi, Hiroshi Kawaguchi, and Masahiko Yoshimoto, “A 7T/14T Dependable SRAM and Its Array Structure to Avoid Half Selection” VLSI Design 2009, pp.295-300, January, 2009

[9] Hidehiro Fujiwara, Shunsuke Okumura, Yusuke Iguchi, Hiroki Noguchi, Hiroshi Kawaguchi, and Masahiko Yoshimoto, “A Dependable SRAM with 7T/14T Memory Cells”, IEICE Transactions Electronics, April, 2009.