A Two-Port SRAM for Real-Time Video Processor Saving 53% of Bitline Power with Majority Logic and Data-Bit Reordering

Hidehiro Fujiwara[1] Koji Nii[1] Junichi Miyakoshi[1] Yuichiro Murachi[1] Yasuhiro Morita[2]
Hiroshi Kawaguchi[3] Masahiko Yoshimoto[3]

[1]Graduate School of Science and Technology, Kobe University
1-1 Rokkodai-Cho, Nada-ku, Kobe 657-8501, Japan

[2]Graduate School of Science and Technology, Kanazawa University
Kakuma-Machi, Kanazawa, Ishikawa 920-1192, Japan

[3]Department of Computer and Systems Engineering, Kobe University
1-1 Rokkodai-Cho, Nada-ku, Kobe 657-8501, Japan

Phone: +81-78-803-6234 fujiwara@cs28.cs.kobe-u.ac.jp

ABSTRACT

We propose a low-power two-port SRAM suitable for real-time video processing. In order to minimize discharge power on a read bitline, a majority-logic decides if input data are inverted in a write cycle, so that "1"s are in the majority. In video data, since more significant bits of adjacent pixel data are fortunately lopsided to either "0" or "1" with higher probability, the data bits in the pixels are reordered in each digit group to exploit the majority logic. The speed and area overheads are 4% and 11% in a 90-nm process technology, respectively. The proposed SRAM achieves 53% power reduction on the bitlines, and saves 43% of a total power when considered as an H.264 reconstructed-image memory.

Categories and Subject Descriptors

B.3.1 **[Semiconductor Memories]**: Static memory (SRAM) design

General Terms

Design

Keywords

Low power SRAM, two-port SRAM, real-time image processing, majority logic, data-bit reordering.

1. INTRODUCTION

As the ITRS predicts, memory area is becoming larger, and will occupy 90% of an SoC in 2013 [1]. In a real-time video SoC, this trend is going on, and an H.264 encoder for a high-definition television requires, at least, a 500-k bit memory as a search-window buffer, which consumes 40% of a total power [2]. As a process technology is scaled down, a large-capacity SRAM will be implemented on a chip as a frame buffer or restructured-image memory, which could consume a larger portion of power. To save an SRAM power in real-time video applications, we propose a low-power two-port SRAM in this paper.

A two-port SRAM is suitable for real-time video processing since it can make one read and one write at the same time in a clock cycle [2-5]. In general, a read port has a single read bitline for area efficiency, and the proposed SRAM also has the same structure in Fig. 1 (a). Two nMOS transistors for a read wordline (RWL) and read bitline (RBL) are added to the conventional single-port 6-T SRAM, which frees a static noise margin (SNM) in a read operation [6]. Therefore, a large β ratio is not required, and two nMOS driver transistors can be minimized.

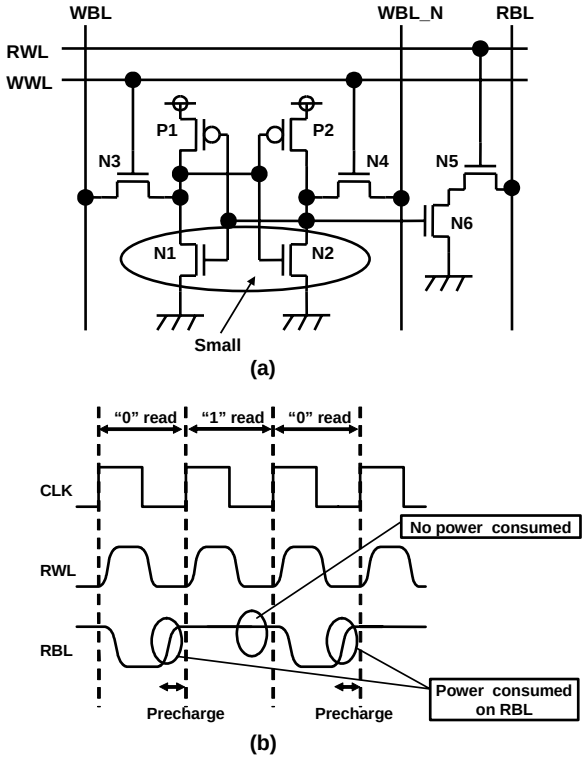


Figure 1. A 8-T two-port SRAM cell.
(a) Circuit, and (b) operation waveforms in read cycles.

ISLPED'06, October 4–6, 2006, Tegernsee, Germany.

Figure 1 (b) illustrates simplified operation waveforms in read cycles. Since a precharge scheme is adopted and an RBL is precharged to V_{dd} before the beginning of a clock cycle, charge and discharge power is consumed on the RBL when "0" is read. In contrast, no power is consumed on the RBL when "1" is read. For low-power operation, hence, it is good to write "1"s as much as possible. The possibility that "1" is read is increased, which in turn reduces the RBL power.

For this purpose, we append majority logic to a two-port SRAM. Although majority logic has been used on transmission lines to save an I/O bus power [7], it has not been utilized in a memory bus as far as we know. In the next section the concept of an SRAM with majority logic is introduced.

Other than the majority logic, we also exploit statistical similarity in video data for further power reduction. Adjacent pixels have correlation one another, which implies more significant bits of the adjacent pixel data are lopsided to either "0" or "1" with higher probability. We reorder the data bits of the adjacent data in each digit group to effectively boost the majority-logic function, which is discussed in Section 3 considering H.264 codec.

In Section 4, we describe design and evaluation of a 72-k bit SRAM in a 90-nm process technology, with the proposed features. The final section summarizes this paper.

2. SRAM WITH MAJORITY LOGIC

Figure 2 (a) illustrates the concept of an SRAM with majority logic. In order to maximize the number of "1"s, the majority-logic circuit counts the number of "1"s, and decides if the input data are inverted in a write cycle, so that "1"s are in the majority. The inversion information ("1" means inversion) is stored in an additional flag bit as depicted in Fig. 2 (b). In a read cycle, the process is reversed. Output data is inverted if a flag bit is true, so that the original data can be read.

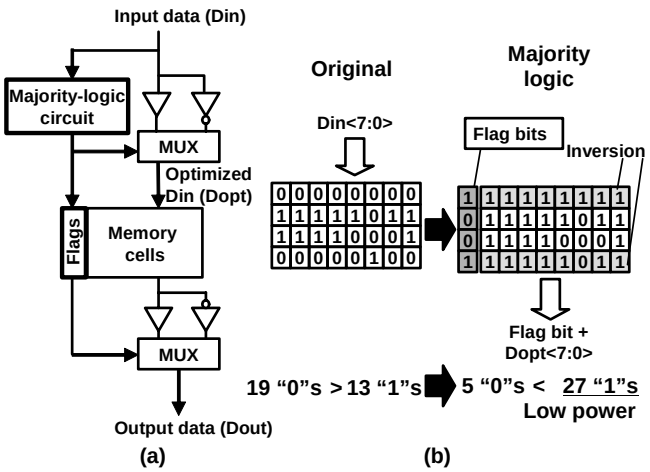


Figure 2. Concept of SRAM with majority logic. (a) Schematic, and (b) flag bit.

The power-reduction factor on the RBL is briefly discussed in Fig. 3. The bit width of data is assumed to be eight here. If the number of "1"s in input data is eight, the data is not inverted and thus, one "0" is stored only in a flag bit and there are no "0"s in the data themselves. This means one charge/discharge is made on RBLs, which is a power overhead. If the number of "1"s are four or less, the input data is inverted by the majority logic to maximize the number of "1"s, and the RBL power is reduces.

When input data have a random pattern, the number of charges/discharges is four out of eight RBLs in the conventional two-port SRAM. However, the majority logic reduces this value to 3.27 although the number of the RBLs is increased to nine. This indicates that the majority logic statistically saves 18% of an RBL power even if the data is random.

Note that it is important to consider which the inversion information in the flag bit should be "0" or "1", because the RBL power even on the flag bit depends on the value as well. If the number of "0"s in whole data is more than that of "1"s, the inversion information should be "1" to maximize the number of "1"s. As previously mentioned, we chose "1" as an inversion flag based on statistical analysis of HDTV test sequences, which is described in detail in the following section.

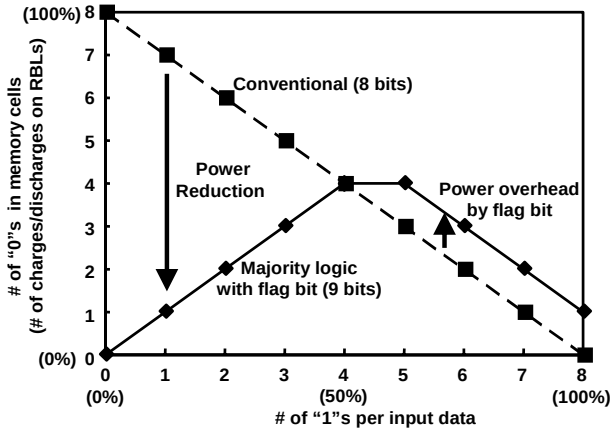


Figure 3. A comparison of RBL powers between the conventional and proposed two-port SRAM with majority logic.

3. DATA-BIT REORDERING

3.1 Statistical Characteristics of Video Image

In H.264 codec, the YUV format is adopted as pixel data. An example is illustrated in Fig. 4. One pixel is comprised of 8-bit luma (Y signal) and 4-bit chroma (U and V signals). In this study, only luma data are considered. In an image, adjacent pixels have strong correlation one another, and the correlation becomes stronger in a more significant bit. Namely, the most significant bits (MSBs) in contiguous data tend to be lopsided to either "0" or "1" with high probability, while in the least significant bits (LSBs), the values of the bits are random. Thus, correlations in each digit are somewhat different one another.

The distributions of the number of "1"s in different digit groups are represented in Fig. 5 when eight-pixel data (8 × 8 bits) are rearranged in each digit group as shown in Fig. 4. It can be seen that the MSB group tends to have "0", which was pointed out in the previous section. The distribution in the LSB group is normally-distributed (strictly, it is binominal distribution), and the same tendencies can be observed even in the 2nd- and 3rd-digit groups.

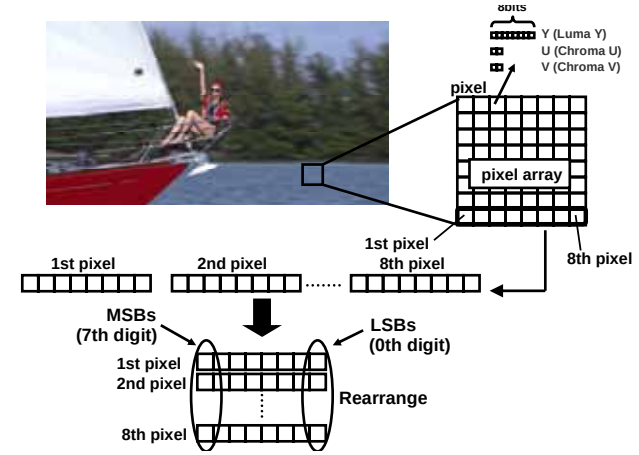


Figure 4. Example of image data.

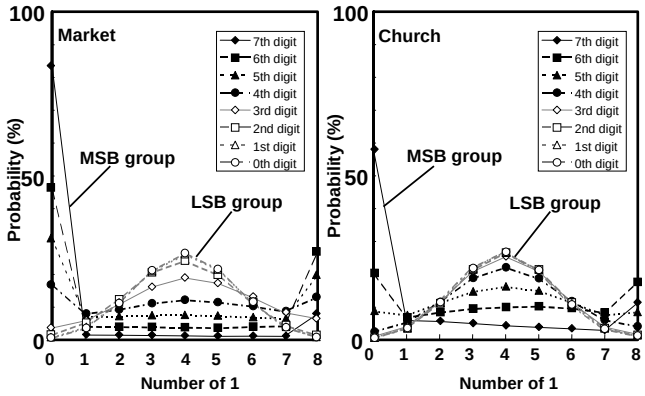


Figure 5. Distributions of the number of "1"s in different digit groups, extracted from the HDTV test sequences, "Market" and "Church".

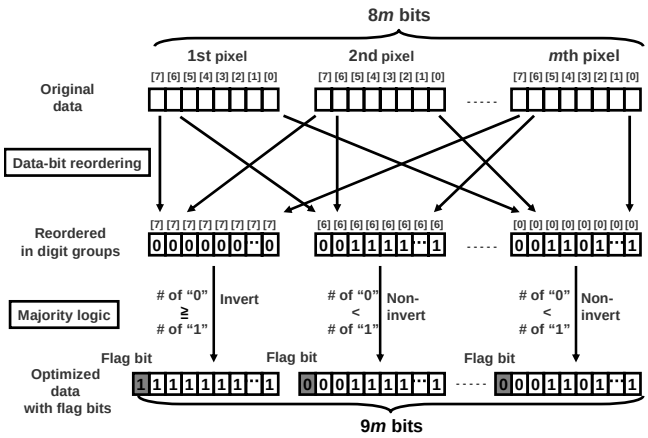


Figure 6. Data-bit reordering and majority logic.

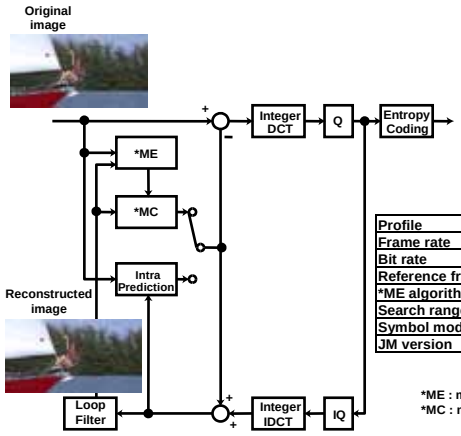
As discussed in the previous section, the power reduction on the RBLs is theoretically expected thanks to the majority logic even if input data is normally-distributed. Besides, further power reduction is promising because the image data is lopsided to "0"s in more significant digit groups as indicated in Fig. 5. We exploit these characteristics to reduce the RBL power.

The rearrange of digits is called data-bit reordering in this paper. Again, we explain data-bit reordering illustrated in Fig. 6. In a write cycle, data comprised of m pixels ($8m$ bits) are reordered in each digit group. The optimum value of m is discussed in the next subsection. If the number of "0"s in a digit group is equal to or larger than that of "1", that is in other words, if the number of "0" is equal to or larger than $m/2$, bits in the digit group are inverted. Alternatively, if the number of "0"s is smaller than that of "1"s, they are not inverted. The majority logic and data-bit reordering maximize the number of "1"s in image data, and optimize the RBL power.

In a read cycle, the optimized data are either inverted or non-inverted according to a flag bit in a digit group. If a flag bit is "1", the data are inverted, and then the reordered bits are put back to the original pixel data.

3.2 Optimum Value of *m*

In order to obtain the optimum value of m, statistic analysis is carried out with the original image and reconstructed image, extracted from ten HDTV test sequences, "Bronze with Credit" (Bronze), "Building along the Canal" (Canal), "Church" (Church), "Intersections" (Inters), "Japanese Room" (Jpnroom), "European Market" (Market), "Yachting" (Sail), "Street Car" (Stcar), "Whale Show" (Whale), and "Yacht Harbor" (Yacht). The original image is encoded, and then a reconstructed image is generated in a local decoding loop and utilized for motion estimation. The encoding configuration is shown in Fig. 7.



Profile	Main profile
Frame rate	30fps
Bit rate	7.5Mbps
Reference frames	2
*ME algorithm	UMHexagon search
Search range	±128 x ±128
Symbol mode	CABAC
JM version	9.8

Figure 7. H.264 encoding process and simulation conditions.

Although pixel data are segmented every m pixels, data-bit reordering does not cause addressing problem since consecutive pixels are simultaneously processed in a video memory.

Figure 8 illustrates the normalized RBL powers in comparison with the conventional two-port SRAM. Figure 8 (a) shows a case where the original image is utilized as input data and only the majority logic is applied to a set of pixels. The number of pixels to which the majority logic is applied, is varied (one, two, and four pixels). The figure demonstrates that 20% of the RBL power is saved on average only with the majority logic and without data-bit reordering, even though the flag bit is appended.

As illustrated in Fig. 8 (b), the saving factor is further extended to 45% with both the majority logic and data-bit reordering, which indicates that the statistical characteristic of image data is well exploited. Moreover, the maximum power save is achieved when the proposed two-port SRAM is utilized as a reconstructed image that has stronger correlation than the original image, as shown in Fig. 8 (c). 53% reduction of the RBL power is possible. The proposed SRAM is suitable for real-time video codec such as

MPEG2, MPEG4, and H.264 which require a large-capacity reconstructed-image memory for motion estimation.

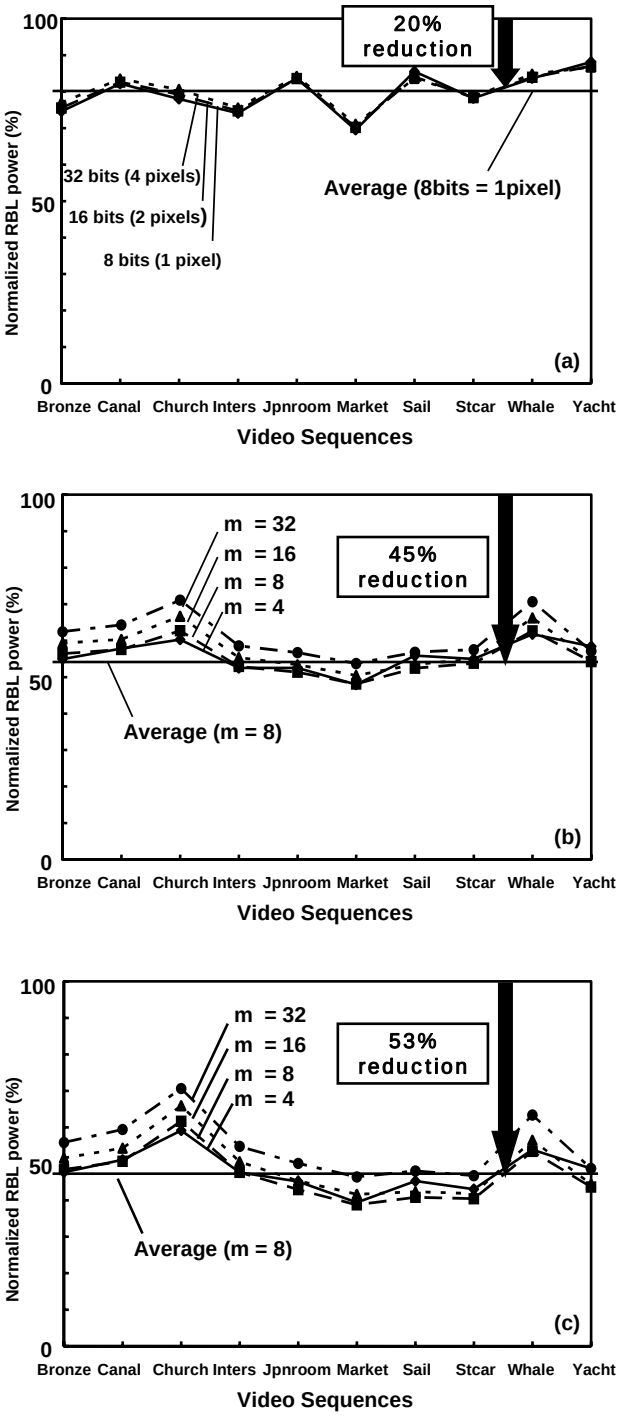


Figure 8. Normalized RBL powers in the (a) original image only with majority logic, (b) original image with both majority logic and data-bit reordering, and (c) reconstructed image with both majority logic and data-bit reordering. 100% signifies a case of the conventional two-port SRAM.

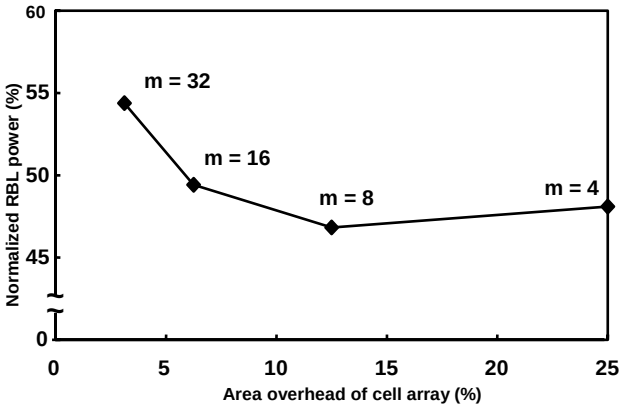


Figure 9. Normalized RBL power vs. area overhead in the proposed SRAM used as a reconstructed-image memory.

In the both cases of the original image and reconstructed image, *m*=4 or *m*=8 is optimum in terms of power reduction. However, if *m* is small, area overhead becomes large. Figure 9 shows the relationship between power and area overhead when *m* is changed. As *m* is increased, the area overhead becomes decreased, but the RBL power is raised. This is because correlation in a digit group becomes weaker as *m* is larger. The area overhead is 12.5% when *m*=8. *m*=4 is not a design choice due to its large area overhead.

4. DESIGN IN 90-nm TECHNOLOGY

4.1 Memory Cell

Figure 10 is a layout of the proposed two-port SRAM cell in a 90-nm process technology. The cell area is $3.15 \times 0.76\ \mu m^2$. The schematic have been already shown in Fig. 1 (a), and the transistor sizes are shown in Fig. 10. Figure 11 is a block diagram, where a capacity of memory cells is 72-k bits. *m*=8 is chosen in this design, and thus 64-k bits are for data themselves, and the other 8-k are flag bits. A hierarchical RBL structure is applied to void a speed overhead of a single-bitline scheme [6]. The write bitlines (WBLs and WBL_Ns) do not have precharge transistors since they are dedicated for data write-in.

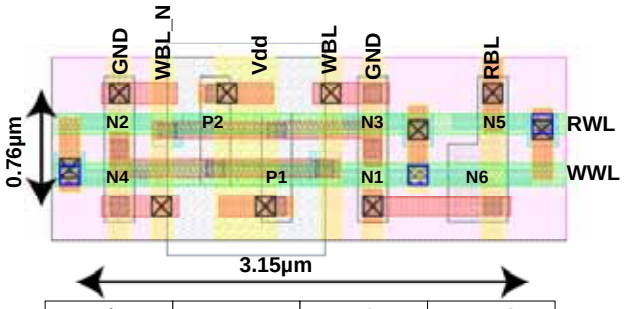
4.2 Write/Read Circuitry with Majority Logic

In a write circuit, there should be a majority-logic circuit. However, it would require 29 cell gates for eight-bit majority logic if designed as a digital circuit [7, 8], which might result in a large area overhead. Figure 12 is the proposed write circuit with majority logic. The majority-logic circuit is achieved with a precharge logic, and thus the majority logic is evaluated by pull-down networks in flip-flops. A sense amplifier amplifies a voltage difference between majority signals, JL and JL_N.

Figures 13 (a) and (b) show the operation waveforms of the majority logic in both cases that "1"s and "0"s are in the majority. When the number of "0"s is eight, eight pull-downs and one dummy rapidly sink a JL node, while the falling time of the JL node takes slower when the number of "0" is four. In this case that the numbers of "1" and "0" are same, the voltage difference between JL and JL_N would be theoretically zero, which might cause a meta stable in the sense amplifier. However in reality, there is some voltage difference between them thanks to the dummy circuit. J in the dummy circuit always sinks the JL node so as not to make the differential voltages same, even if the number of "1"s is as same as that of "0"s. However, as *m* is

increased, the voltage difference could be closer. From this point of view, it can be said that *m*=8 is a good design choice.

Figure 14 is a read circuit resuming the original data, which inverts data bits depending on a flag bit. The conditional inversion is implemented with EX-ORs.



Transistors	N1 - N5	N6	P1, P2
Length / Width	0.1um / 0.2um	0.1um / 0.4um	0.13um / 0.2um

Figure 10. Memory cell layout

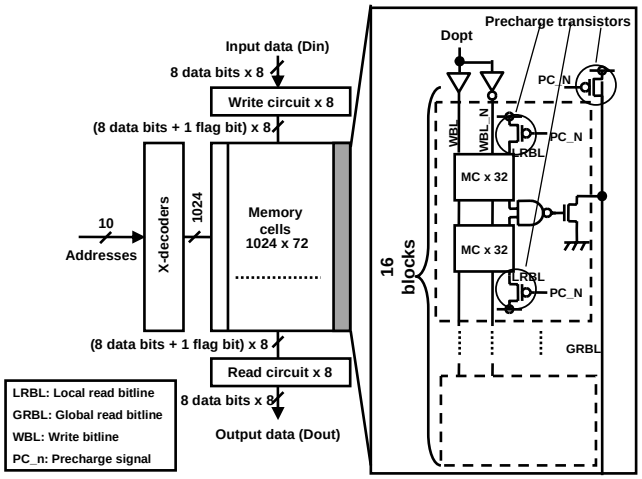


Figure 11. Block diagram of the proposed 72-k bit SRAM.

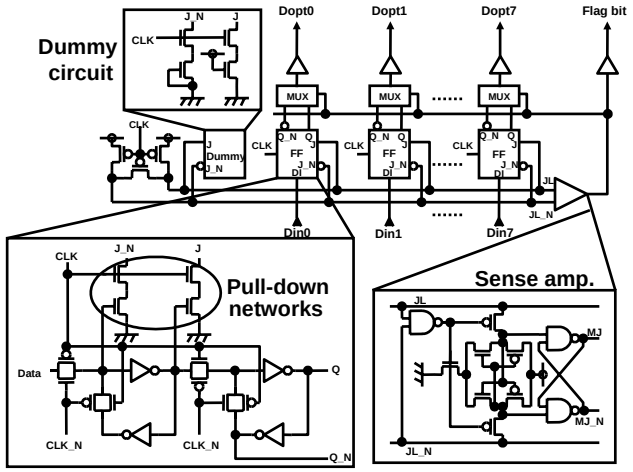
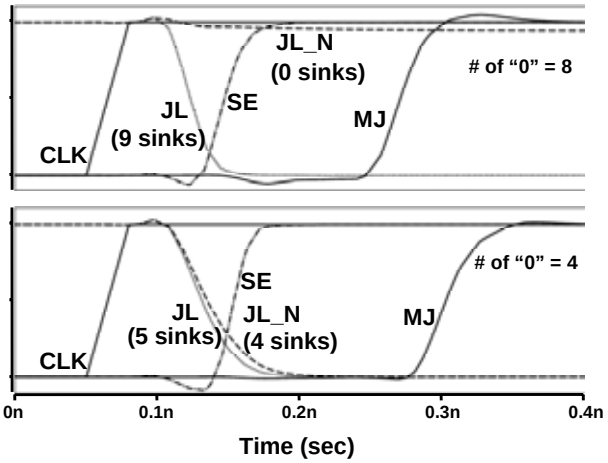


Figure 12. Write circuit with majority logic.



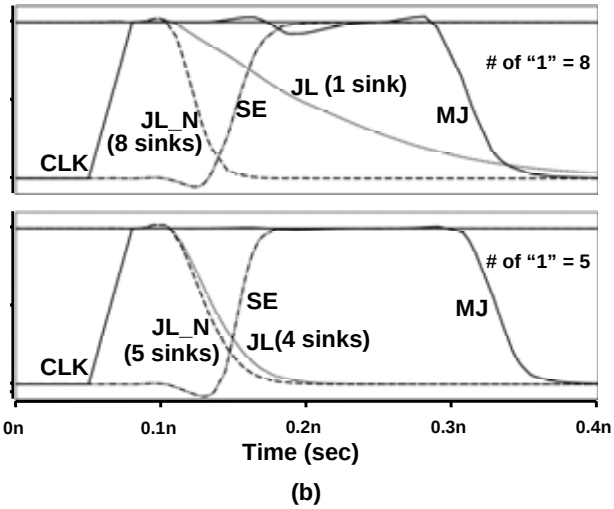


Figure 13. Operation Waveforms of majority logic when (a) "0"s and (b) "1"s are in the majority.

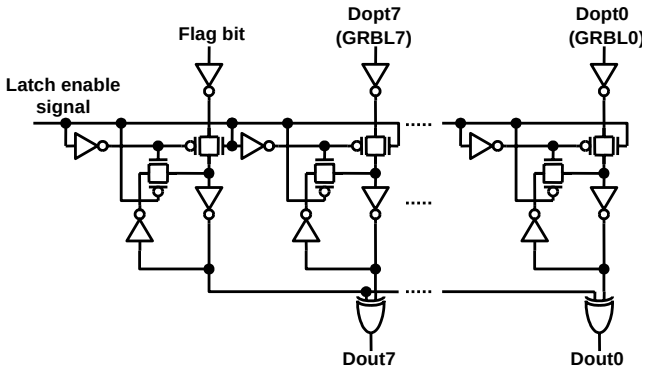


Figure 14. Read circuit resuming the original data.

Although some additional circuits are implemented to the write and read circuits for the majority logic, the speed overhead in a read cycle is 4% in a simulation. Alternatively, there is no speed overhead in a write cycle since it is canceled out by a wordline delay. The area overhead of the majority logic itself is less than 1%, and overall area overhead including the flag bits becomes 11%. The power caused by the majority logic is only 21 μW, which is negligible compared with a bitline power.

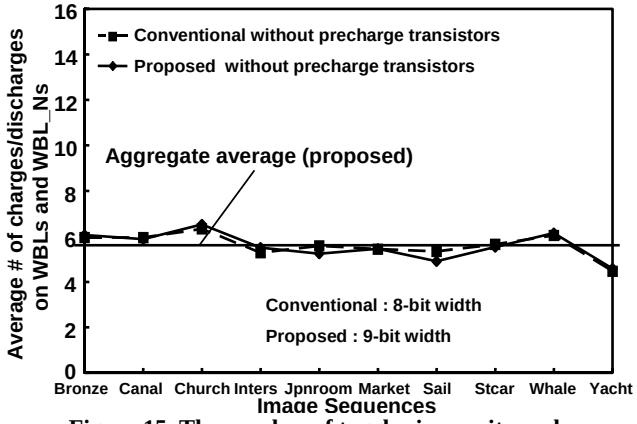


Figure 15. The number of toggles in a write cycle.

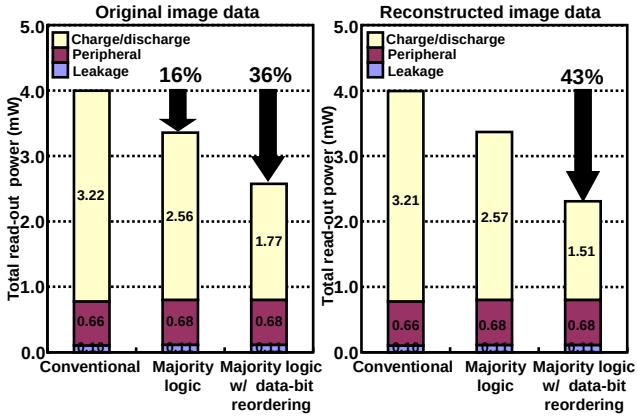


Figure 16. Readout power reduction (T=25°C).

4.3 Power Estimation

Since there are no precharge transistors on the write bitlines, it does not consume any power as far as same data are successively written. From this point of view, the statistical characteristic of image data helps to save the write-bitline power. Figure 15 shows the average number of charges/discharges in a write cycle. Even though the bit width is increased to nine by a flag bit, the reduction factor in the proposed SRAM is as same as that of the conventional one thanks to the majority logic. Consequently, the proposed SRAM has no power overhead on write bitlines.

In Fig. 16, the total read-out powers are compared between the conventional and proposed SRAMs. In a video memory, power reduction in a read operation is technically important since read-out is made more frequently than write-in. In the conventional 64-k bit SRAM, a total read-out power is estimated at 4.0 mW, at a supply voltage of 1.0 V and frequency of 300MHz. The RBL power occupies 80% of the power in the conventional SRAM, and thus the proposed SRAM reduces this part. Only with the majority logic, 16% of the total read-out power can be saved in the proposed 72-k bit SRAM when the original H.264 image is considered. In addition, 36% reduction can be achieved with both the majority logic and data-bit reordering. As a reconstructed-image memory, the proposed SRAM further saves 43% of the total read-out power.

5. SUMMARY

We proposed a two-port SRAM with majority logic and data-bit reordering. This SRAM is suitable for real-time image processing in which data have statistical similarity. Only with the majority logic, it was verified that 20% of read-bitline power can be saved in a 72-k bit SRAM in a 90-nm process technology. Moreover, the saving factor is extended to 45% thanks to data-bit reordering. The proposed SRAM achieves 53% power reduction on the bitlines when considered as an H.264 reconstructed image, and saves 43% of the total read-out power. The speed and area overheads are 4% and 11%, respectively.

6. REFERENCES

[1] International Technology Roadmap for Semiconductors 2005, http://www.itrs.net/Common/2005ITRS/Home2005.htm.

[2] J. Miyakoshi, Y. Murachi, K. Hamano, T. Matsuno, M. Miyama, and M. Yoshimoto, "A Low-Power Systolic Array Architecture for Block-Matching Motion Estimation," IEICE Trans. Electronics, Vol.E88-C, No.4, pp.559-569, Apr. 2005.

[3] Y. Murachi, K. Hamano, T. Matsuno, J. Miyakoshi, M. Miyama, and M. Yoshimoto, "A 95 mW MPEG2 MP@HL Motion Estimation Processor Core for Portable High-Resolution Video Application," IEICE Trans. Fundamentals, Vol.E88-A, No.12, pp.3492-3499, Dec. 2005.

[4] S. Ishiwata, T. Yamakage, Y. Tsuboi, T. Shimazawa, T. Kitazawa, S. Michinaka, K. Yahagi, A. Oue, T. Kodama, N. Matsumoto, T. Kamei, M. Saito, T. Miyamori, G. Ootomo, and M. Matsui, "A Single-Chip MPEG-2 Codec Based on Customizable Media Embedded Processor," IEEE J. Solid-State Circuits, Vol.38, No.3, pp.530-540, Mar. 2003.

[5] Y-W. Huang, T-C. Chen, C-H.Tsai, C-Y. Chen, T-W. Chen, C-S. Chen, C-F. Shen, S-Y. Ma, T-C. Wang, B-Y. Hsieh, H-C. Fang, and L-G. Chen, "A 1.3TOPS H.264/AVC Single-Chip Encoder for HDTV Applications," IEEE Int. Solid-State Circuits Conf., pp.128-129, Jan. 2005.

[6] K. Takeda, Y. Hagihara, Y. Aimoto, M. Nomura, Y. Nakazawa, T. Ishii, and H. Kobatake, "A Read-Static-Noise-Margin-Free SRAM Cell for Low-Vdd and High-Speed Applications," IEEE J. Solid-State Circuits, Vol.41, No.1, pp.113-121, Jan. 2006.

[7] M. R. Stan, and W. P. Burleson, "Bus-Invert Coding for Low Power I/O," IEEE Trans. VLSI Systems, Vol.3, No.1, pp.49-58, Mar. 1995.

[8] Y. Shin, and K. Choi, "Narrow Bus Encoding for Low Power Systems," Asia and South Pacific Design Automation Conf., pp.217-220, Jan. 2000.