PAPER Special Section on Novel Device Architectures and System Integration Technologies

A Power- and Area-Efficient SRAM Core Architecture with Segmentation-Free and Horizontal/Vertical Accessibility for Super-Parallel Video Processing

Junichi MIYAKOSHI[†a)], **Yuichiro MURACHI**[†], ***Student Members***, **Tomokazu ISHIHARA**[†], **Hiroshi KAWAGUCHI**[†], ***Nonmembers***, ***and*** **Masahiko YOSHIMOTO**[†], ***Member***

SUMMARY For super-parallel video processing, we proposed a power- and area-efficient SRAM core architecture with a segmentation-free access, which means accessibility to arbitrary consecutive pixels, and horizontal/vertical access. To achieve these flexible accesses, a spirally-connected local-wordline select signal and multi-selection scheme in wordlines are proposed, so that extra X-decoders in the conventional multi-division SRAM can be eliminated. Consequently, the proposed SRAM reduces a power and area by 57–60% and 60%, respectively, when it is applied to a 128 parallel architecture. The proposed 160-kbit SRAM with 16-read ports (2-read port SRAM with eight-parallel architecture) is implemented to a search window buffer for an H.264 motion estimation processor core which dissipates 800 μW for QCIF 15-fps in a 130-nm technology.

key words: *SRAM, low power, parallel processing, image signal processing, H.264, MPEG*

1. Introduction

As image-processing algorithms, there are MPEG2, MPEG4, JPEG2000, H.264 [1], and so on, which require complex calculation and heavy workload to achieve high coding efficiency. In the calculation, both horizontal and vertical accesses to successive pixels in an image cache take place. For example, in the H.264 standard, horizontal and vertical six-tap filters are used to create pixels with a half-pel accuracy. In a motion estimation of H.264, successive eight pixels pointed by an arbitrary search vector need to be horizontally/vertically read from a cache SRAM, to calculate an optimum motion vector. Usually, eight-time accesses are necessary to read vertically successive eight pixels since an address has to be changed eight times. Even in a horizontal access, it takes two accesses or more, unless horizontal eight pixels are aligned in a data segmentation. This implies that horizontal/vertical access with segmentation-free capability, which means a coinstantaneous access to arbitrary consecutive pixels, is desirable in the cache SRAM.

To meet the above requirements and achieve the both horizontal and vertical access, we propose a novel SRAM architecture with a spirally-connected local-wordline select line and a multi-selection scheme in wordlines, which is suitable for super-parallel video processing. The proposed SRAM reduces an area and power compared to the conventional SRAM which is described in Sect. 2. The proposed SRAM architecture is explained in Sect. 3. Power and area estimations are shown in Sect. 4. Section 5 summarizes this paper.

Manuscript received April 10, 2006.
Manuscript revised June 19, 2006.

[†]The authors are with the Graduate School of Science and Technology, Kobe University, Kobe-shi, 657-8501 Japan.
a) E-mail: mjun1@cs28.cs.kobe-u.ac.jp

DOI: 10.1093/ietele/e89–c.11.1629

2. Conventional Techniques

This section describes in detail pixel data accesses using the conventional SRAM architectures.

Figure 1 shows a block diagram of a general SRAM and its memory data mapping for the eight-parallel datapath. In order to perform the segmentation-free access with the general SRAM, the multiple accesses are often required because the general SRAM can not read out/write in suc-

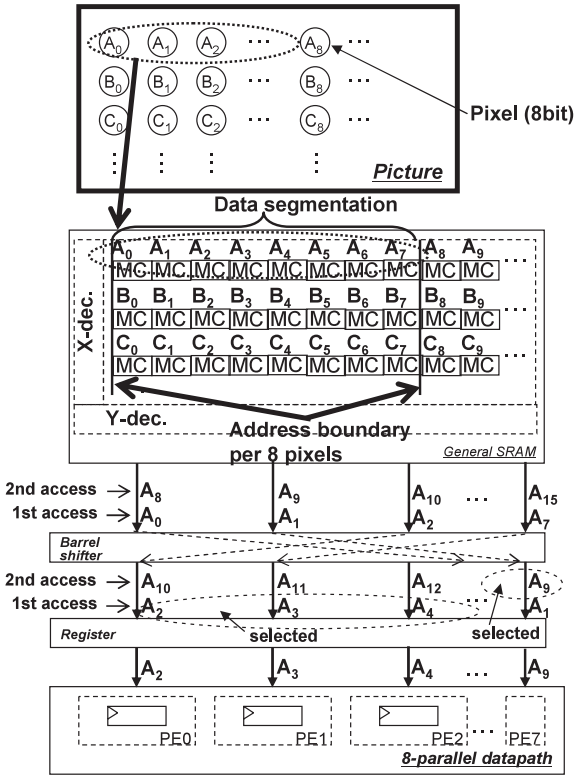


Fig. 1 Block diagram of general SRAM and its memory data mapping (accessing A_2 to A_9 pixels in a horizontal direction).

cessive pixels over an address boundary between data segmentation at a time. In the figure, when the arbitrary eight pixels, A_2 to A_9, are required, A_0 to A_7 pixels are read out at the first access, and then A_8 to A_{15} pixels are accessed at the second access. By choosing the required A_2 to A_9 pixels from these A_0 to A_{15}, we can get the arbitrary eight pixels. For the vertical access, the general SRAM requires eight read-out cycles to obtain pixel data in a vertical direction. The general SRAM requires many cycles and dissipate much power to access arbitrary consecutive pixels.

For the horizontal/vertical access, the SRAM-based FIFO memory [2] is introduced, however, it does not have the segmentation-free capability. Therefore, two accesses or more is sometimes needed if data are laid over data segmentations. On the other hand, the multi-division SRAM [3], [4] has the segmentation-free capability in the horizontal direction but in the vertical access. A block diagram and memory data mapping of the multi-division SRAM in a case of an eight-parallel architecture are illustrated in Fig. 2. The number of divisions of the SRAM should be equal to the number of parallelisms in a processor. Successive eight pixels on a row are mapped to different SRAMs, but they are accessed at the same time. Then, the eight pixels read out are processed with processor elements in parallel. The Y-decoders in the divided SRAMs and barrel shifter achieve the segmentation-free capability in the horizontal direction. On a vertical access for the multi-division SRAM, the activated SRAM block is only one although it takes the eight access cycles to read eight pixel data. However, the multi-division SRAM requires X- and Y-decoders as many as the number of divisions, which results in large area and power.

On a high-definition television (HDTV; 1920 × 1080 pixels), huge computing power is necessary to encode/decode the high-resolution picture, where the number of parallelisms will become 16, 32, 64, or more [5]. To the super-parallel architecture, however, the conventional multi-division SRAM can not be applied because the number of divisions turns out large.

3. Proposed SRAM Architecture

3.1 Spiral Memory Mapping

To achieve the horizontal/vertical access in the proposed SRAM, picture data are spirally mapped in memory cells as shown in Fig. 3, by which pixels along the vertical direction can be read out or written in at a time. Thus in the case of the vertical access, multiple wordlines should be activated by the X-decoder, while in the case of the horizontal access, only one wordline is asserted. A pre-decoder in the X-decoder handles this procedure.

3.2 Pre-Decoder and Segmentation-Free Mechanism

Figure 4 shows a detailed schematic of the proposed SRAM when an eight-parallel architecture is implemented. Note that the number of parallelisms is easily extended to a larger one for super-parallel architecture. The X-decoder in the

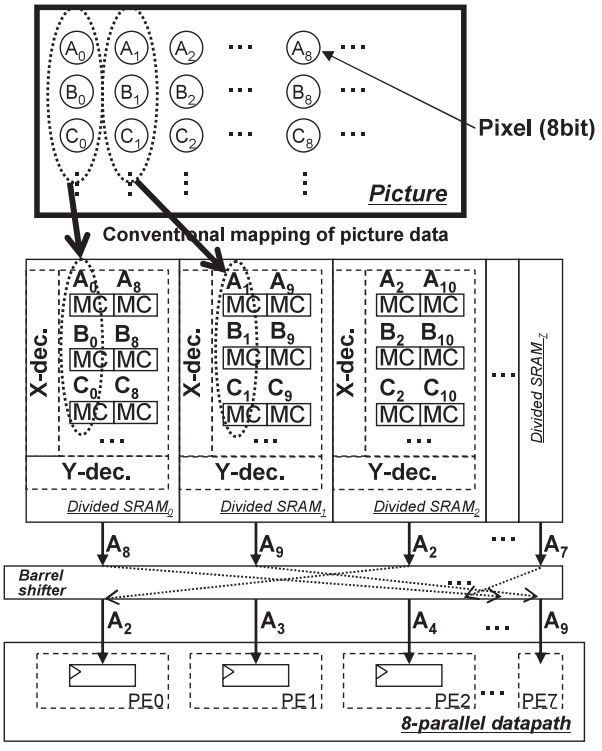


Fig. 2 Block diagram of the conventional multi-division SRAM and its memory data mapping (accessing A_2 to A_9 pixels in a horizontal direction).

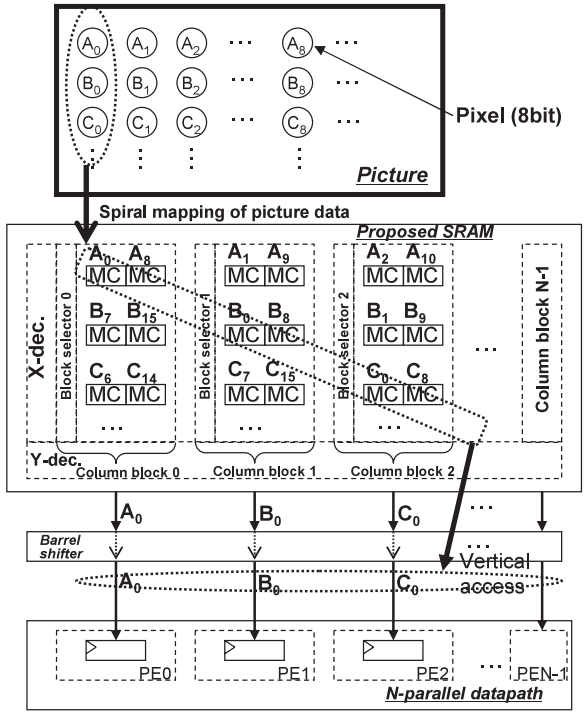


Fig. 3 Spiral memory mapping in the proposed SRAM. The number of parallelisms, N, can be easily extended to a larger one (accessing A_0 to I_0 pixels in a vertical direction).

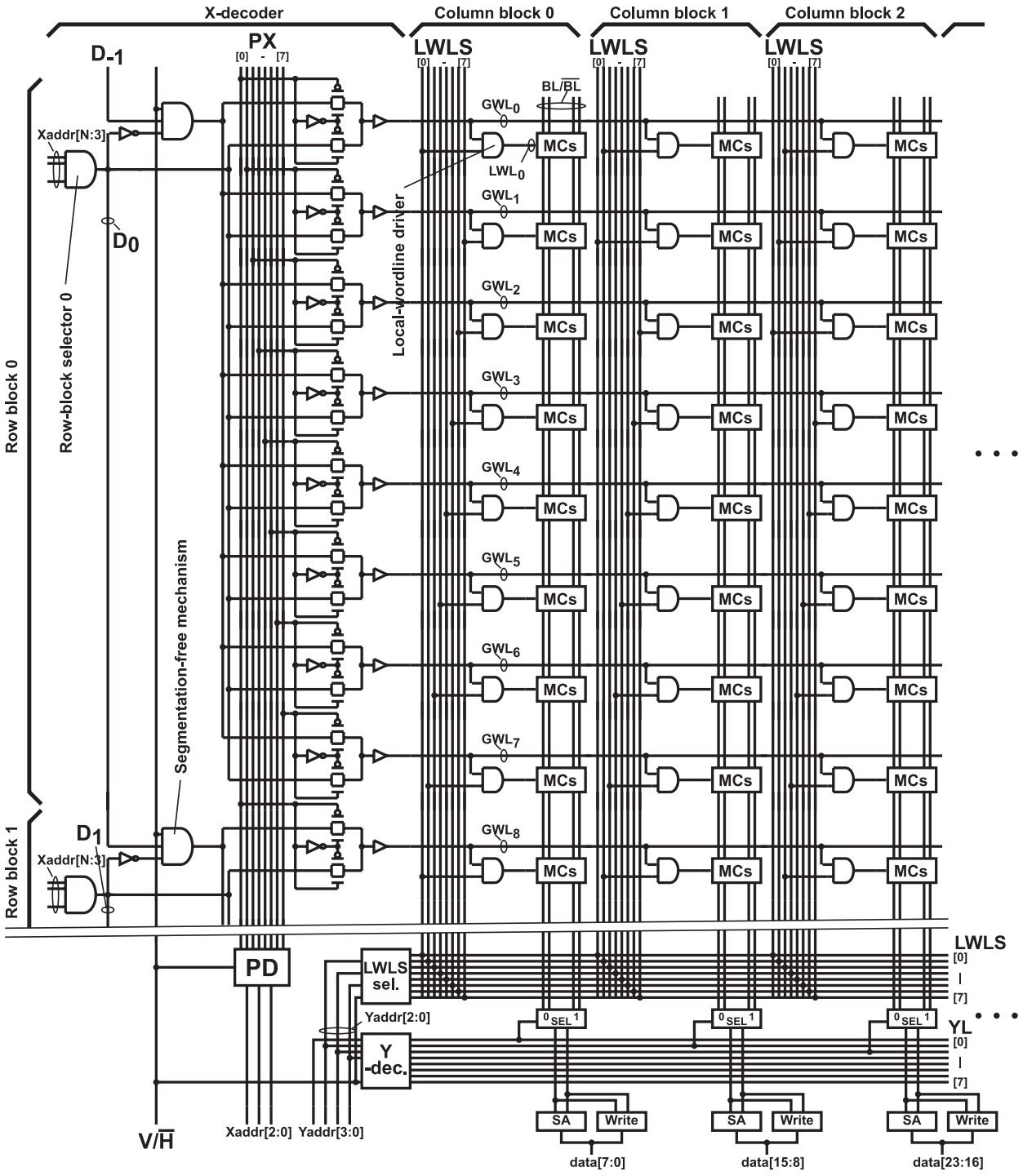


Fig. 4 Detailed schematic of the proposed SRAM. SA signifies sense amplifiers.

figure includes the pre-decoder (PD), which carries out different operations in the horizontal and vertical accesses. The V/H signal signifies the access direction. In the figure, MCs store two-pixel data (16 bits). The number of pixels stored in each MC is appropriately determined, depending on the picture width and the number of parallelisms.

V/H=0 means the horizontal access, in which only one global wordline (e.g. GWL_0) is activated according to Table 1. The other global wordlines are negated since V/H=0.

On the other hand in the vertical access, multiple global wordlines should be asserted as mentioned in the previous subsection. Table 2 is a truth table in the vertical access. For instance, if eight global wordlines from GWL_0 to GWL_7 are activated, the operation is simply understood (see the row of Xaddr[2:0] = 000 in Table 2). However even in the case that GWL_1 to GWL_8 should be selected (Xaddr[2:0] =

Table 1 Pre-decoder truth table (horizontal access, V/H=0).

Xaddr [2:0]	PX							
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
000	1	0	0	0	0	0	0	0
001	0	1	0	0	0	0	0	0
010	0	0	1	0	0	0	0	0
011	0	0	0	1	0	0	0	0
100	0	0	0	0	1	0	0	0
101	0	0	0	0	0	1	0	0
110	0	0	0	0	0	0	1	0
111	0	0	0	0	0	0	0	1

Table 2 Pre-decoder truth table (vertical access, V/H=1).

Xaddr [2:0]	PX							
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
000	1	1	1	1	1	1	1	1
001	0	1	1	1	1	1	1	1
010	0	0	1	1	1	1	1	1
011	0	0	0	1	1	1	1	1
100	0	0	0	0	1	1	1	1
101	0	0	0	0	0	1	1	1
110	0	0	0	0	0	0	1	1
111	0	0	0	0	0	0	0	1

Table 3 LWLS signals truth table (vertical access, V/H=1).

Yaddr. [2:0]	LWLS							
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
000	1	0	0	0	0	0	0	0
001	0	1	0	0	0	0	0	0
010	0	0	1	0	0	0	0	0
011	0	0	0	1	0	0	0	0
100	0	0	0	0	1	0	0	0
101	0	0	0	0	0	1	0	0
110	0	0	0	0	0	0	1	0
111	0	0	0	0	0	0	0	1

001), the operation is still guaranteed. This is because, in a segmentation-free mechanism in the figure, the row-block select signal, D_0, is true, D_1 is false, and V/H is true. Either output of the segmentation-free mechanism or output (D_1) of the row-block selector are selected to assert multiple GWLs according to the PX signals in Table 2. Then, the output of the segmentation-free mechanism is selected as GWL_8 since PX[0] = 0. Thus, the number of activated global wordlines is always fixed in the vertical access, which demonstrates the segmentation-free capability in the arbitrary vertical direction.

Also, the PD circuit is comprised of the decoder for horizontal access, the decoder for vertical access, and the selector of outputs of these decoders.

3.3 Local-Wordline Control

Since multiple global wordlines are activated in the vertical access, only one pixel should be selected in a global wordline. Otherwise, a multi-selection problem in a bitline would happen.

The local-wordline select signals (LWLS[7:0]) in Fig. 4 are spirally connected to local-wordline drivers to control the local-wordline activation. Table 3 shows a truth table for the LWLS signals in the vertical operation. The LWLS signals are controlled with Yaddr[2:0]. In this manner, we can obtain arbitrary vertical pixels in security.

On the other hand in the horizontal access, all LWLS signals become true to obtain horizontally successive pixels. This does not cause the multi-selection problem since only one global wordline is asserted in the horizontal access. The LWLS selector is comprised as with the PD.

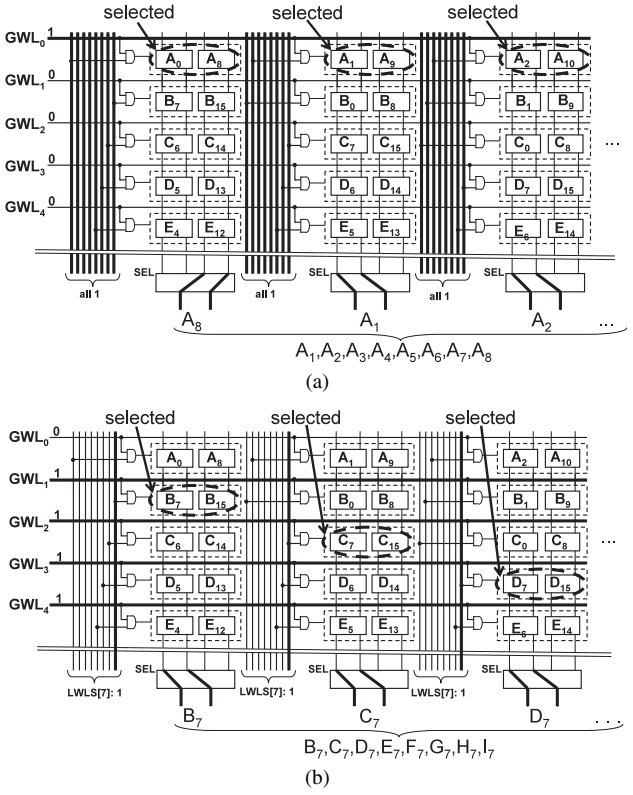


Fig. 5 Segmentation-free accesses in (a) horizontal direction and (b) vertical direction.

3.4 Y-Decoder

We discussed the spiral memory mapping, pre-decoder architecture, and local-wordline control scheme in this section, however the segmentation-free capability can not be achieved only with them.

A horizontal-access case is illustrated in Fig. 5(a). The spiral memory mapping is already explained in Sect. 3.1. If we want to access eight pixels from A_0 to A_7, it is easy to handle just by asserting GWL_0. However, the issue is, for instance, a case that we want to access A_1 to A_8, which are laid over data segmentation. A_8 in the figure should be read out (or written in) from the right-hand pixel by a bitline selector (SEL), while the other pixels are the left-hand pixels. This means that the SELs carry out different operations on a pixel-by-pixel basis, which is achieved by the Y-decoder in Fig. 4. Table 4 is the truth table for the YL signals from the Y-decoder. If a value is "0," a left-hand pixel is accessed. A vertical-access case is shown in Fig. 5(b), where B_7 to I_7 are read out (or written in). Table 5 is its truth table. Unlike the

Table 4 Y-decoder truth table (horizontal access, V/H=0).

Yaddr [3:0].	YL							
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
0000	0	0	0	0	0	0	0	0
0001	1	0	0	0	0	0	0	0
0010	1	1	0	0	0	0	0	0
0011	1	1	1	0	0	0	0	0
0100	1	1	1	1	0	0	0	0
0101	1	1	1	1	1	0	0	0
0110	1	1	1	1	1	1	0	0
0111	1	1	1	1	1	1	1	0
1000	1	1	1	1	1	1	1	1
1001	0	1	1	1	1	1	1	1
1010	0	0	1	1	1	1	1	1
1011	0	0	0	1	1	1	1	1
1100	0	0	0	0	1	1	1	1
1101	0	0	0	0	0	1	1	1
1110	0	0	0	0	0	0	1	1
1111	0	0	0	0	0	0	0	1

Table 5 Y-decoder truth table (vertical access, V/H=1).

Yaddr [3:0].	YL							
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
0000	0	0	0	0	0	0	0	0
0001	0	0	0	0	0	0	0	0
0010	0	0	0	0	0	0	0	0
0011	0	0	0	0	0	0	0	0
0100	0	0	0	0	0	0	0	0
0101	0	0	0	0	0	0	0	0
0110	0	0	0	0	0	0	0	0
0111	0	0	0	0	0	0	0	0
1000	1	1	1	1	1	1	1	1
1001	1	1	1	1	1	1	1	1
1010	1	1	1	1	1	1	1	1
1011	1	1	1	1	1	1	1	1
1100	1	1	1	1	1	1	1	1
1101	1	1	1	1	1	1	1	1
1110	1	1	1	1	1	1	1	1
1111	1	1	1	1	1	1	1	1

horizontal-access case, values are one-sided to either "0" or "1" since vertical pixels in a same coordinate (e.g. B_7, C_7, ..., I_7) all are put in either left-hand or right-hand side.

4. Estimation of Power and Area Reduction and VLSI Implementation

The energy reductions in the proposed SRAM with the numbers of parallelisms are demonstrated in Fig. 6 and Fig. 7. The conventional 1 and 2 have been already explained in Fig. 1 and Fig. 2 each. Here we supposed that the number of parallelisms in these figures is eight. Figure 6 shows the energy comparison in the horizontal access, and also Fig. 7 shows that in the vertical access. These energies are calculated from the following equation.

$$Energy = (access\ cycle) \times (P_{access}) \quad (1)$$

Here, the access cycle indicates required cycles to read/write data from/in the SRAMs. P_{access} represents power consumption of SRAMs per a horizontal access or a vertical access. The access cycles of each operation are summarized in Table 6 where the number of parallelism is m. Also, the memory capacity assumed in Figs. 6–10 is

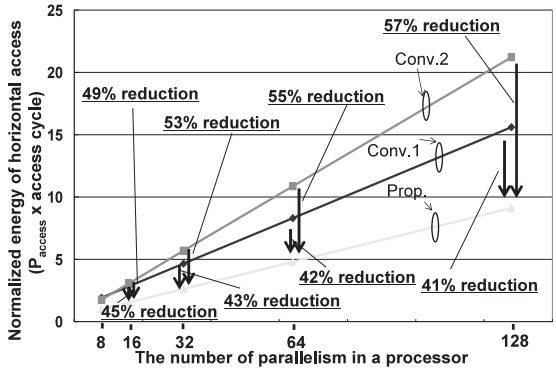


Fig. 6 Energy comparison of horizontal access between the conventional and proposed SRAM where a memory size is 160-kbits.

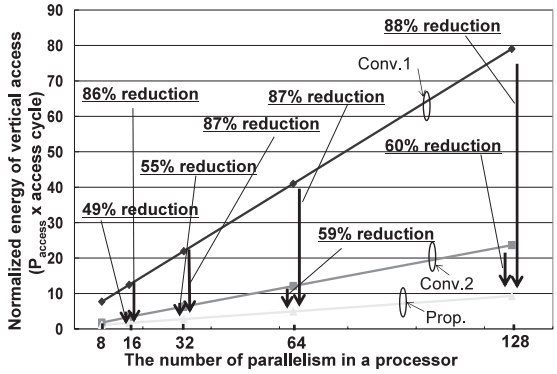


Fig. 7 Energy comparison of vertical access between the conventional and proposed SRAM where a memory size is 160-kbits.

Table 6 Access cycles of each operation with m-way parallelism.

	Horizontal access cylces	Vertical access cycles
Conv.1	2 cycles	m cycles
Conv.2	1 cycle	m cycles
Proposed	1 cycle	1 cycle

160 kbit. That is required as a search window buffer in an H.264 motion estimation processor (ME) which detects motion vectors for CIF 30-fps video under the condition of ± 32 pixels × ± 16 pixels search area and three reference frames. In order to enhance a computing performance of the ME processor, the number of parallelisms in the ME tends to increases, so that this analysis makes sense to show the effectiveness of the proposal architecture. Also, in these figures, the energy and area are normalized by that of the proposed SRAM with eight-parallelism.

Compared with the Conv.-1 SRAM, the proposed scheme reduces the power on a horizontal access by 43%, 42%, and 41%, when the numbers of divisions are 32, 64, and 128, respectively. Also, compared with the Conv.-2 SRAM, the power reductions by the proposed SRAM are 53%, 55%, and 57% at the same conditions, as shown in Fig. 6. On a vertical access, the saving powers of the proposed SRAM are 87%, 87%, and 88% comparing to the Conv.-1 SRAM, 55%, 59%, and 60% compared with the Conv.-2 SRAM shown in Fig. 7. In this case, the saving

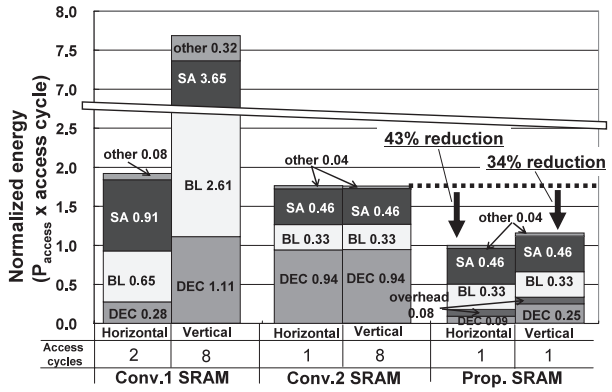


Fig. 8 Power breakdowns in the conventional and proposed SRAMs (eight-parallelism case).

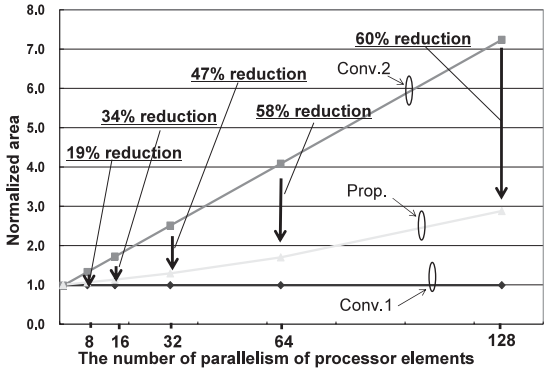


Fig. 9 Area comparison between the conventional and proposed SRAMs where a memory size is 160-kbits.

factor is increasing as the number of parallelisms becomes larger. This is because extra power of X-decoders in the Conv.-1 SRAM and the Conv.-2 SRAM can be eliminated in the proposed scheme, thanks to the spirally-connected LWLS signals. Figure 8 illustrates the power breakdowns in the conventional and proposed SRAMs on a horizontal and vertical access when the number of parallelism is eight, which demonstrates that the power consumed by the X-decoders is dramatically decreased even in the eight-parallel case. The power overhead by the LWLS, LWL driver, and segmentation-free mechanism is just 8%, shown in Fig. 8.

The area is also reduced by 47%, 58% and 60% at the same conditions compared with the Conv.-2 SRAM, as shown in Fig. 9. The area of the Conv.-1 SRAM is constant for the difference number of parallelisms because the Conv.-1 SRAM feeds data to a super-parallel datapath by increasing the access cycle to a super-parallel architecture. As the number of parallelism is increased, the LWLS lines occupy larger area in the proposed configuration, so that the area increases along the second-order curve, as shown in Fig. 9. This area overhead is, however, much smaller than the extra X-decoders in the conventional SRAM. The area breakdowns in Fig. 10 show that the area overhead of the X-decoder is reduced by 31%, the LWLS overhead is just 6%.

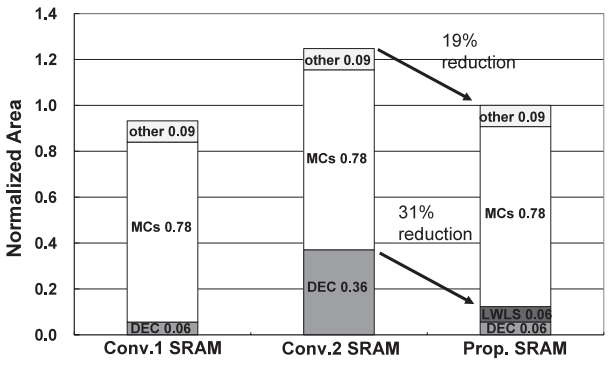


Fig. 10 Area breakdowns in the conventional and proposed SRAMs (eight-parallelism case).

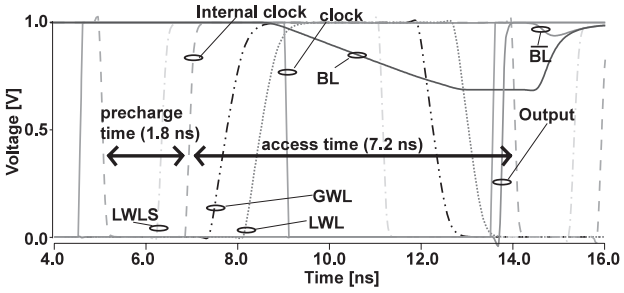


Fig. 11 Simulation waveform of the proposed SRAM.

Consequently, the proposed scheme is more suitable for super-parallel environment.

The delay overhead which is caused by insertion of the segmentation-free mechanism is 1.3 ns in a 130-nm CMOS technology whose supply voltage is 1.0 V. The waveform of the proposed SRAM by SPICE simulation is shown in Fig. 11. The cycle time is 9.0 ns, and the access time is 7.2 ns.

The proposed SRAM has been implemented to a search window buffer of an H.264 motion estimation processor core which consumes 800 μW for QCIF 15-fps in the 130-nm CMOS process technology. Figure 12 shows chip layouts of the processor core and proposed SRAM. The capacity is 160-k bits, and the number of parallelisms is eight. The SRAM has two-read ports and one-write ports to achieve 16-read port by the eight-parallel SRAM for a 16-way SIMD datapath. In this case, the comparisons of the power and area were already discussed in Fig. 8 and Fig. 10, respectively. The power and area are reduced by 34% to 43% and 19%, comparing to the Conv.-2.

Furthermore, the power and the area estimation of the proposed SRAM implemented to an H.264 motion estimation processor for higher resolution videos are shown in Fig. 13 and Fig. 14 respectively. The power comparisons in Fig. 13 are calculated on the condition which the horizontal access and the vertical access occur with same probability. A higher resolution causes increasing pixel data to be processed, so that the workload of the image processor is getting larger. So the super-parallel architecture should be

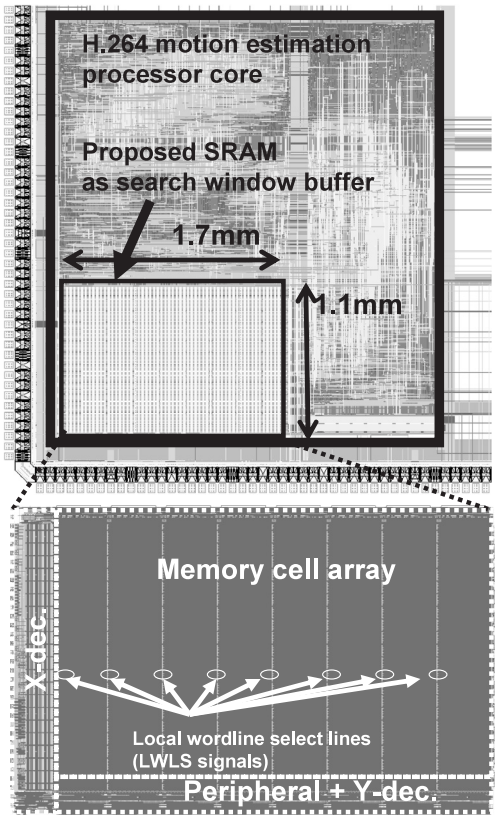


Fig. 12 Layout of the H.264 motion estimation processor and the proposed SRAM layout.

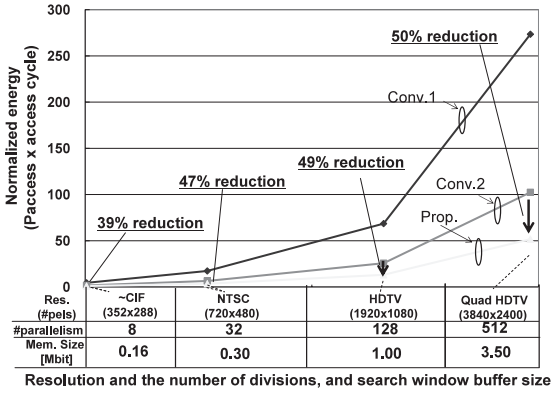


Fig. 13 Energy comparison between the conventional and proposed SRAM for resolutions, the number of divisions, and memory buffer sizes.

taken in the processor for the realistic implementation by moderate clock frequency.

At the eight-parallelism for CIF-resolution-video (352×288), the power reduction is 39%, and the area reduction is 19% compared with the Conv.-2 SRAM. At 512-parallels for processing quad-HDTV (3840×2400 pixels), the power and area are lowered by 50% and 19%. Although the area of Conv.-1 SRAM is smallest, that power consumption drastically increases due to many access cycles required for super-parallel architecture, as shown in Fig. 13 and Fig. 14. It is seen that the proposed techniques are more

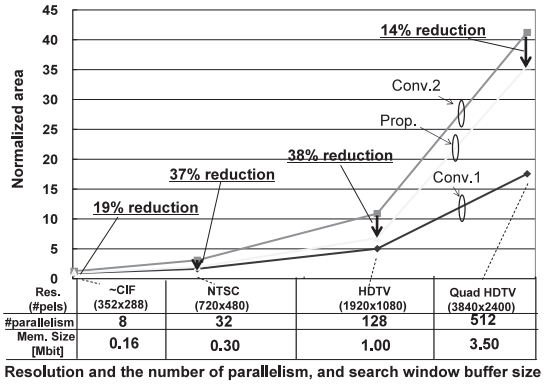


Fig. 14 Energy comparison between the conventional and proposed SRAM for resolutions, the number of divisions, and memory buffer sizes.

effective for higher resolution video.

5. Conclusion

We proposed a power and area-efficient SRAM core architecture enabling horizontal/vertical accesses, with segmentation-free capability. This is suitable for super-parallel video processing. To achieve the horizontal/vertical access function, a spirally-connected local wordline select signals and multi-selection scheme in global wordlines are applied, so that extra X-decoders are eliminated in the proposed scheme. The features were implemented to a H.264 motion estimation processor core in a 130-nm CMOS process technology, which demonstrated that the power and area were reduced by 34–43% and 19%, respectively, compared with the conventional design. Furthermore, the proposed SRAM potentially save up to 57–60% and 60% in the power and area, respectively, when it is implemented to a 128-parallels architecture. In addition the estimation results for higher resolution video imply that the proposed techniques are promising for future parallel video processing.

Acknowledgments

This work is supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Cadence Design Systems, Mentor Graphics and Synopsys, Inc.

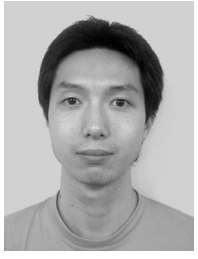
References

[1] ITU-T Rec. H.264|ISO/IEC 14496-10 AVC, “Draft ITU-T recommendation and final draft international standard of joint video specification,” 2003.
[2] N. Shibata, M. Watanabe, and Y. Yanabe, “A current-sensed high-speed and low-power first-in-first-out memory using a wordline/bitline-swapped dual-port SRAM cell,” IEEE J. Solid-State Circuits, vol.37, no.6, pp.735–750, June 2002.
[3] M. Miyama, J. Miyakoshi, Y. Kuroda, K. Imamura, H. Hashimoto, and M. Yoshimoto, “A sub-mW MPEG-4 motion estimation processor core for mobile video application,” IEEE J. Solid-State Circuits, vol.39, pp.1562–1570, Sept. 2004.

[4] Y. Murachi, K. Hamano, T. Matsuno, J. Miyakoshi, M. Miyama, and M. Yoshimoto, "A 95 mW MPEG2 MP@HL motion estimation processor core or portable high-resolution video application," IEICE Trans. Fundamentals, vol.E88-A, no.12, pp.3492–3499, Dec. 2005.
[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 International Solid-state Circuit Conference, pp.128–129, Jan. 2005.

Junichi Miyakoshi received the B.E. degree in electrical and information engineering in 2002 and the M.E. degree in electronics and computer science from Kanazawa University, Ishikawa, Japan, in 2004. He is currently a Ph.D. candidate at Kobe University, Kobe, Japan. His current research interests include high-performance and low-power multimedia VLSI designs. He is a student member of IEEE.

Yuichiro Murachi was born on November 1, 1980. He received a B.S. degree from Kanazawa University in 2003. He received an M.E. degree from Kanazawa University, Ishikawa, Japan, in 2005. He is currently enrolled in the Doctoral course in Kobe University. His research interests are VLSI systems and implementation of multimedia communication systems.



Tomokazu Ishihara was born on December 28, 1983. He received a B.E. degree from Kobe University, Hyogo, Japan in 2006. He is currently on the master course at Kobe University. His research interests include low-power VLSI algorithms and architectures, and multimedia signal processing.

Hiroshi Kawaguchi received the B.E. and M.E. degrees in electronic engineering from Chiba University, Chiba, Japan, in 1991 and 1993, respectively. He received the 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 to be a Research Associate in 2003. Since 2005, he has been a Research Associate in the Department of Computer and Systems Engineering, Kobe University, Kobe, Japan. He is also a Collaborative Researcher in the Institute of Industrial Science, the University of Tokyo. He is a recipient of the IEEE ISSCC 2004 Takuo Sugano Award for Outstanding Far-East Paper. He has served as a program committee member for IEEE Symposium on Low-Power and High-Speed Chips (COOL Chips). He is a guest associate editor of IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences. His current research interests include low-power VLSI design, wireless sensor network, and robot vision. Dr. Kawaguchi is a member of the IEEE and ACM.

Masahiko Yoshimoto received the 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 received a Ph.D. degree in Electrical Engineering from Nagoya University, Nagoya, Japan in 1998. He joined the LSI Laboratory, Mitsubishi Electric Corp., Itami, Japan, in April 1977. From 1978 to 1983 he was engaged in the design of NMOS and CMOS static RAM including a 64 K full CMOS RAM with the world's first divided-word-line 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 Dept. of Electrical and Electronic Systems Engineering at Kanazawa University, Japan. Since 2004, he has been a Professor of the Dept. of Computer and Systems Engineering at Kobe University, Japan. His current activity is 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. In addition, he has served as a Guest Editor for special issues on Low-Power System LSI, IP, and Related Technologies of IEICE Transactions in 2004. He received the R&D100 awards from R&D Magazine for development of the DISP and development of a realtime MPEG2 video encoder chipset in 1990 and 1996, respectively.