A Power- and Area-Efficient SRAM Core Architecture for Super-Parallel Video Processing

Junichi Miyakoshi, Yuichiro Murachi, Masaki Hamamoto, Takahiro Iinuma, Tomokazu Ishihara, Hiroshi Kawaguchi, Masahiko Yoshimoto
The Graduate School of Science and Technology
Kobe University
Kobe, Hyogo, Japan
mjun1@cs28.cs.kobe-u.ac.jp

Tetsuro Matsuno
The Graduate School of Science and Technology
Kanazawa University
Kanazawa, Ishikawa, Japan
matsuno@cs28.cs.kobe-u.ac.jp

***Abstract*—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 an area and power by 69% and 59%, respectively, when it is applied to a 128 parallel architecture. The proposed 160-kbit SRAM with 16-read ports (eight-division and 2-read port SRAM) 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.**

I. 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.

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, though the multi-division SRAM [3]-[4] has the segmentation-free capability in the horizontal direction, it can not be accessed in the vertical direction. A block diagram and memory data mapping of the multi-division SRAM in a case of an eight-parallel architecture are illustrated in Fig. 1. 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 or systolic array in parallel. The Y-decoders in the divided SRAMs and barrel shifter achieve the segmentation-free capability in the horizontal direction. However, the multi-division SRAM requires X- and Y-decoders as many as the number of divisions, which results in large area and power.

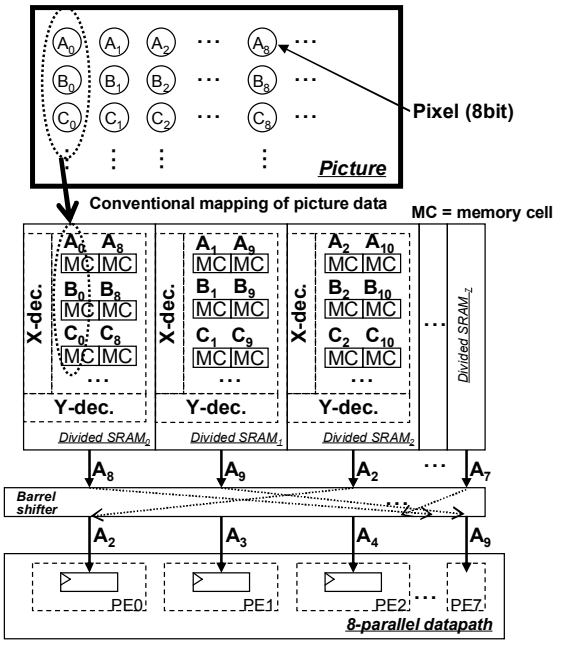

Figure 1. Block diagram of the conventional multi-division SRAM and its memory data mapping.

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.

To solve this issue and achieve the both horizontal and vertical access, we propose a novel SRAM architecture with a spirally-connected local-wordline select line, which is suitable for super-parallel video processing. The proposed SRAM reduces an area and power compared to the conventional multi-division SRAM, which is describes in Section II. Power and area estimations are shown in Section III. Section IV summarizes this paper.

II. PROPOSED SRAM ARCHITECTURE

A. 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. 2, 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.

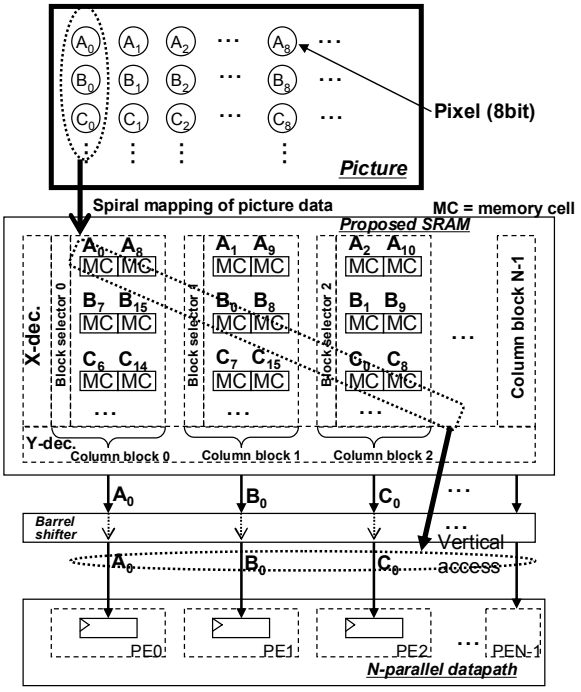

Figure 2. Spiral memory mapping in the proposed SRAM. The number of parallelisms, N, can be easily extended to a larger one.

B. Pre-Decoder Architecture

Fig. 3 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 figure includes the pre-decoder (PD), which carries out different operations in the horizontal and vertical accesses. The V/$\overline{H}$ 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/$\overline{H}$=0 means the horizontal access, in which only one global wordline (e.g. GWL_0) is activated according to TABLE I. The other global wordlines are negated since V/$\overline{H}$ =0.

On the other hand in the vertical access, multiple global wordlines should be asserted as mentioned in the previous subsection. TABLE II 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 II). However even in the case that GWL_1 to GWL_8 should be selected (Xaddr[2:0] = 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/$\overline{H}$ 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 II. 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.

TABLE I. PRE-DECODER TRUTH TABLE (HORIZONTAL ACCESS, V/$\overline{H}$=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 II. PRE-DECODER TRUTH TABLE (VERTICAL ACCESS, V/$\overline{H}$=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

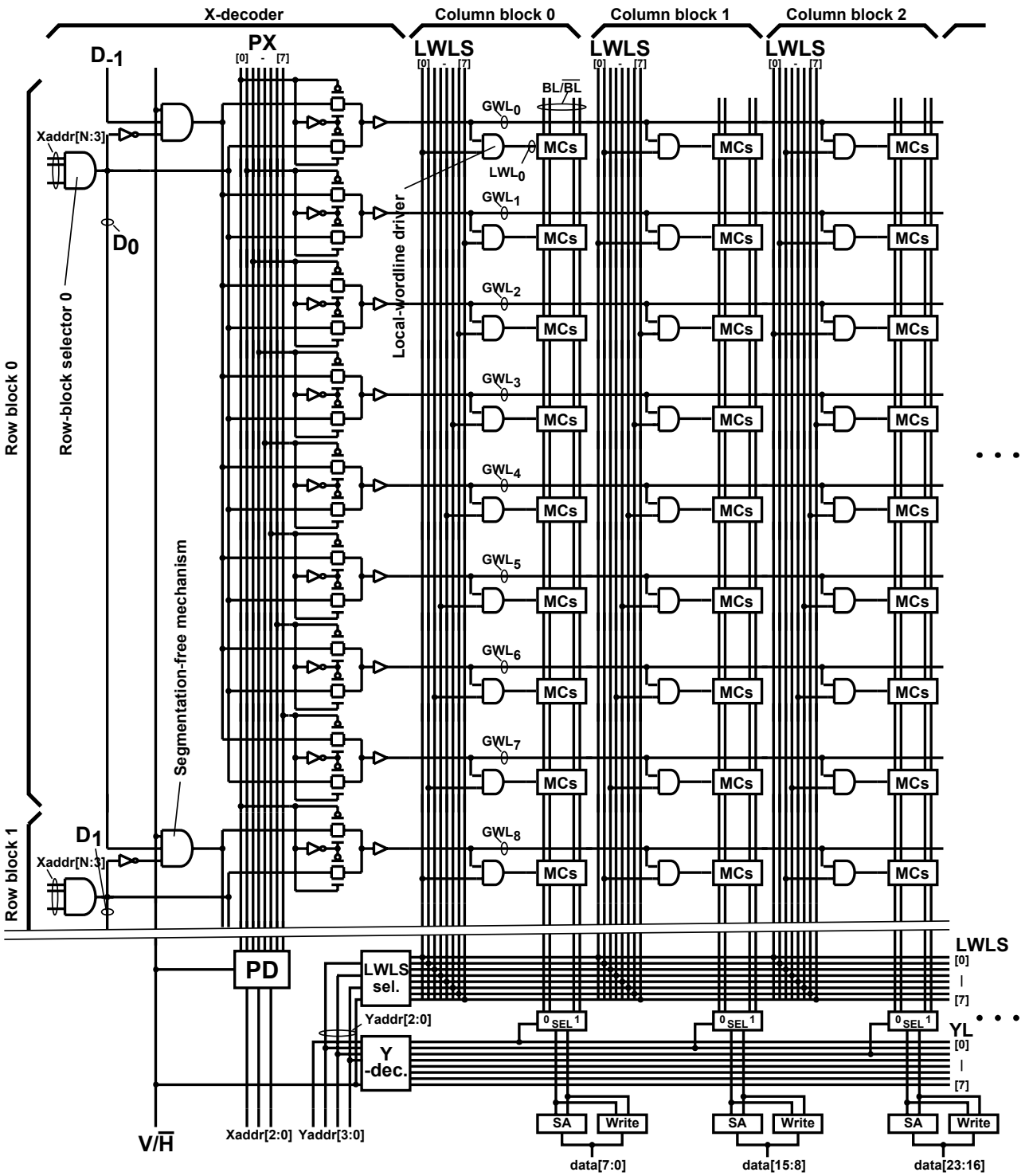

Figure 3. Detailed schematic of the proposed SRAM. SA signifies sense amplifiers.

C. 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. 3 are connected to local-wordline drivers to control the local-wordline activation. TABLE III 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.

TABLE III. LWLS SIGNALS TRUTH TABLE (VERTICAL ACCESS, $V/\overline{H}=1$)

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

D. 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. 4 (a). The spiral memory mapping is already explained in Section II.A. 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. 3. TABLE IV 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. 4 (b), where B_7 to I_7 are read out (or written in). TABLE V is its truth table. Unlike the 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.

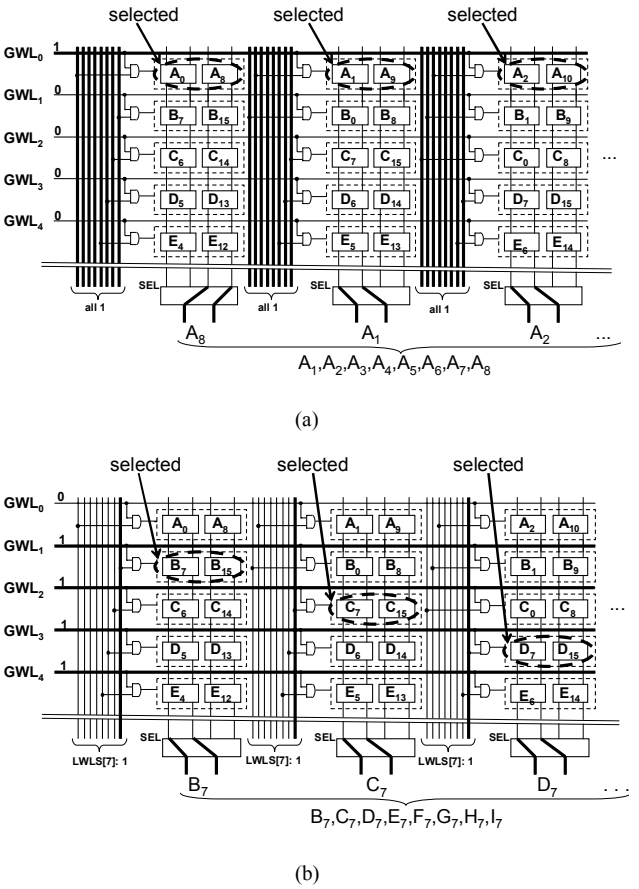

Figure 4. Segmentation-free accesses in (a) horizontal direction and (b) vertical direction.

TABLE IV. Y-DECODER TRUTH TABLE (HORIZONTAL ACCESS, $V/\overline{H}=0$)

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 V. Y-DECODER TRUTH TABLE (VERTICAL ACCESS, V/$\overline{H}$=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

III. ESTIMATION OF POWER AND AREA REDUCTION AND VLSI IMPLEMENTATION

The power and area reductions in the proposed SRAM are shown in Figs. 5 and 6, respectively. The horizontal axes in the figures represent the number of divisions in SRAMs. The performance comparisons are estimated on 130-nm CMOS technology with 1.0-V supply votage and 100-MHz operating frequency.

Compared with the conventional SRAM, the proposed scheme reduces the power by 55%, 57%, and 59%, when the numbers of divisions are 32, 64, and 128, respectively. The saving factor is increasing as the number of the divisions becomes larger. This is because extra X-decoders in the conventional multi-division SRAM can be eliminated in the proposed scheme, thanks to the spirally-connected LWLS signals. Fig. 7 illustrates the power breakdowns in the conventional and proposed SRAM when the number of divisions is eight, which demonstrates that the power consumed by the X-decoders is dramatically decreased even in the eight-division case.

The area is also reduced by 47%, 60% and 69% at the same conditions. As the number of divisions is increased, the LWLS lines occupy larger area. This area overhead is, however, much smaller than the extra X-decoders in the conventional SRAM. The area breakdowns in Fig. 8 show the situation. Although the area overhead of the X-decoder is reduced by 31%, the LWLS overhead is just 5%.

Consequently, the proposed scheme is more suitable for super-parallel environment, which exhibits applicability to HDTV image signal processing. 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 proposed SRAM was implemented to a search buffer of an H.264 motion estimation processor core on which the power is 800 μW for QCIF 15-fps in the 130-nm CMOS process technology. Fig. 9 shows chip layouts of the processor core and proposed SRAM. The capacity is 160-k bits, and the number of divisions is eight. The SRAM has two-read ports and one-write ports to achieve 16-read port by the eight-division for a 16-way SIMD datapath. In this case, the comparisons of the area and power were already discussed in Figs. 7 and 8, respectively. The power and area are reduced by 42% and 19%, comparing with the conventional one. The cycle time is 9.0 ns, and the access time is 7.2ns. In this case, the speed overhead of the proposed SRAM is 1.3 ns.

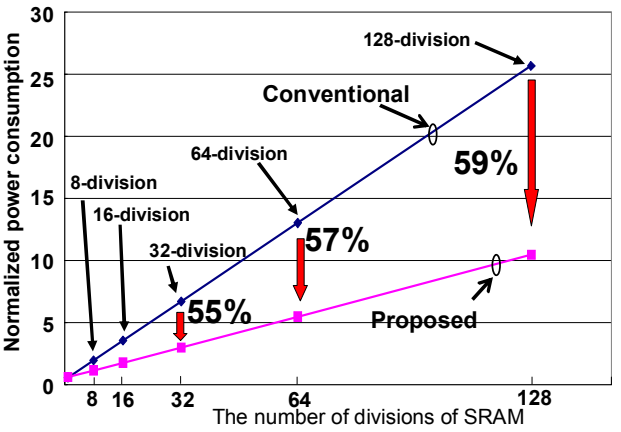

Figure 5. Power comparison between the convetional and proposed SRAMs

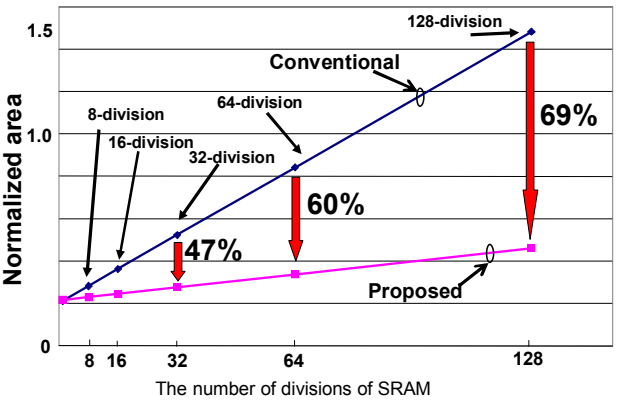

Figure 6. Area comparison between the conventional and proposed SRAMs

IV. SUMMARY

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 42% and 19%, respectively, compared with the conventional design. Furthermore, the proposed SRAM potentially save up to 59% and 69% in the power and area, respectively, when it is implemented to a 128 parallel architecture.

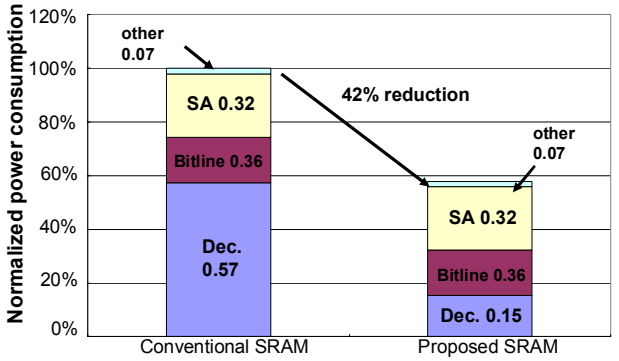

Figure 7. Power breakdowns in the conventional and proposed SRAMs (eight-division case)

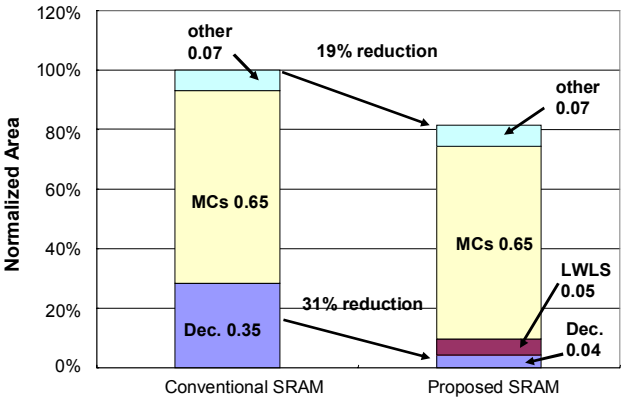

Figure 8. Area breakdowns in the conventional and proposed SRAMs (eight-division case)

ACKNOWLEDGMENT

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

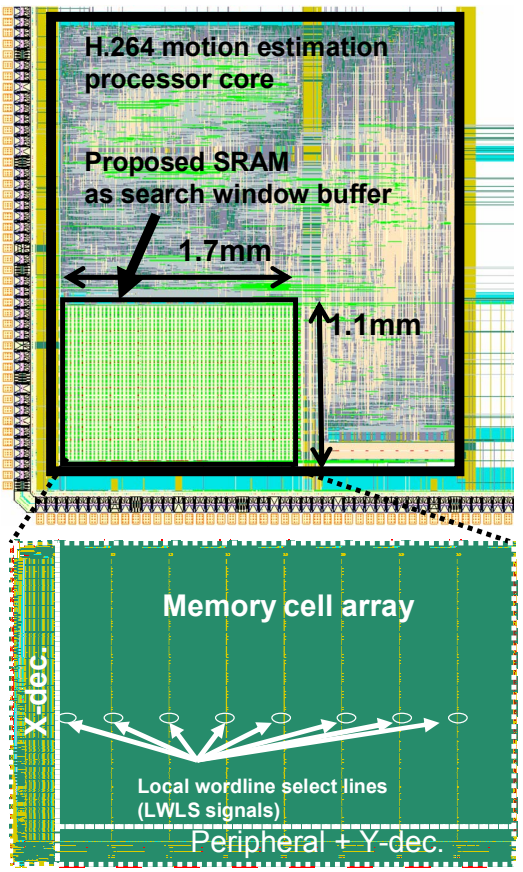

Figure 9. Layout of the H.264 motion estimation processor and the proposed SRAM layout

REFERENCE

[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] Nobutaro Shibata, Muyumi Watanabe, and Yasuyuki Yanabe, "A Current-Sensed High-Speed and Low-Power First-In-First-Out Memory Using a Wordline/Bitline-Swapped Dual-Port SRAM Cell", IEEE Journal of Solid-state Circuits, VOL. 37, NO. 6, JUNE 2002.

[3] Masayuki Miyama, Junichi Miyakoshi, Yuki Kuroda, Kousuke Imamura, Hideo Hashimoto, Masahiko Yoshimoto, "A sub-mW MPEG-4 Motion Estimation processor core for mobile video application" IEEE Journal of Solid Circuits, vol. 39, pp. 1562 – 1570, September 2004.

[4] Yuichiro Murachi, Koji Hamano, Tetsuro Matsuno, Junichi Miyakoshi, Masayuki Miyama, and Masahiko 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, December 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, L-G. Chen, "A 1.3TOPS H.264/AVC Single-Chip Encoder for HDTV Applications", IEEE International Solid-state Circuit Conference, pp128-129, Junuary 2005