# A Power-Efficient SRAM Core Architecture with Segmentation-Free and Rectangular Accessibility for Super-Parallel Video Processing

*Yuichiro Murachi, Tetsuya Kamino, Junichi Miyakoshi,*
*Hiroshi Kawaguchi and Masahiko Yoshimoto*
Kobe University, Kobe, Japan
Tel & Fax: 81-078-803-6234, E-mail: t-kamino@cs28.cs.kobe-u.ac.jp

## ABSTRACT

This paper describes a unique SRAM architecture for super-parallel video processing. It features one cycle functional access of a rectangular image data ($n \times m$ pixels) with segmentation-free. To achieve this accessibility, a local word-line select scheme and a merged X-decoder method are newly introduced with elimination of extra X-decoder employed in usage of the conventional divided SRAM macro. The proposed SRAM has been adopted to a search window buffer for H.264 motion estimation processor for HDTV resolution video. As a result, a power and area of the search window buffer are reduced by 49 % and by 48 %, respectively. Furthermore, it is shown that the proposed SRAM is more efficient for super-HDTV resolution video which requires more parallelism.

## 1. INTRODUCTION

As picture resolution is higher in video appliances, computation workloads for a video processing are increased. To realize a real-time operation, highly parallel architecture is expected to be more popular in future video signal processor. Also one cycle access to a rectangular pixel data is required to feed them to a parallel data-path without wait cycles. For example, H.264 [1] interger-pel motion estimation (IME) requires 8 × 8 to 16 × 32 pixels for block matching calculation. In addition, it is necessary that an image cache can provide sub-sampled pixels to handle field image. Consequently, it is desirable that the image cache supplies $n \times m$ pixels in four forms: integer-pel, horizontally sub-sampled, vertically sub-sampled, horizontally and vertically sub-sampled. If above scheme is implemented using the conventional SRAM macro, $2n \times 2m$ SRAM banks are required.

## 2. CONVENTIONAL SRAM MACRO

The conventional 256-SRAM configuration ($n = m = 8$) and its mapping scheme are illustrated in Fig. 1. The conventional 256-SRAM (conv.1) consists of 16 × 16 banks. Pixels on the image are mapped on the SRAM pixel by pixel so that any consecutive 8 pixels can be accessed concurrently in a cycle. Figure 2 shows sub-sampled 8 × 8 pixel access in the conventional 256 SRAM.

Figure 3 shows a pixel mapping scheme in another conventional 16 SRAM configuration (conv.2). Conv.2 consists of 16 banks which can provide 16 pixels in a cycle. Pixels on the image are mapped on the SRAM line by line so that any 16 successive lines can be accessed parallel. Each bank provides horizontal 16 pixels. These banks provide 16 pixels like $x = 0 \sim 15$ or $16 \sim 31$ in a cycle. However, 16 pixels like $x = 10 \sim 25$ can't be fetched in one cycle because of an address boundary. Therefore conv.2 requires two cycles for the rectangular pixels in an arbitrary location.

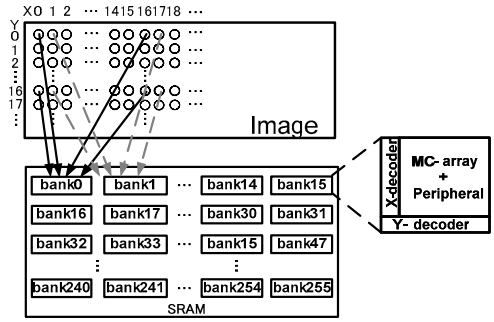



Figure 1. Pixel mapping scheme in the conventional 256 SRAM configuration (conv.1)

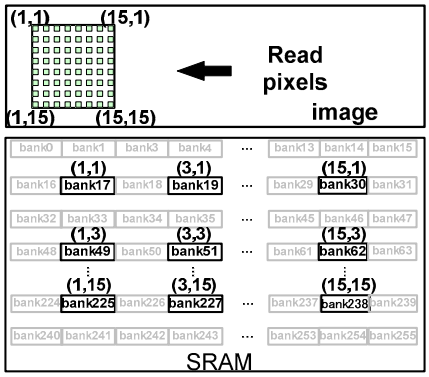



Figure 2. Sub-sampling rectangular access in the conv.1

Conv.1 consists of 256 SRAM banks because 16 × 16 pixels accessibility is required for sub-sampling access, which results in an area and power overhead of X-decoders. Conv.2 consists of only 16 banks but requires two cycles to read rectangular pixels over segmentation.

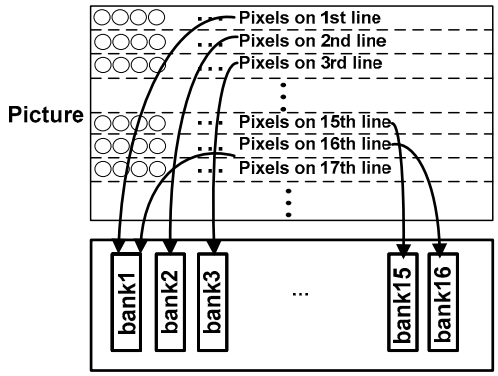



Figure 3. Pixel mapping scheme in the conventional 16 SRAM configuration (conv.2)

978-1-4244-1617-2/08/$25.00 ©2008 IEEE.

## 3. PROPOSED SRAM OVERVIEW

A block diagram of the proposed SRAM ($n$ = $m$ = 8) is illustrated in Fig. 4. The figure also shows a mapping manner of pixels in an image. The SRAM consists of $m$ (= 8) banks: each has left and right blocks. 16 SRAM blocks can be accessed in parallel. Pixels on the image are mapped on the SRAM line by line so that any consecutive 8 lines can be accessed parallel in a cycle. Each block also has a segmentation-free accessibility by which consecutive $m$ (= 8) pixels (integer-pel) or horizontal sub-sampled $m$ (= 8) pixels in an arbitrary location can be accessed.

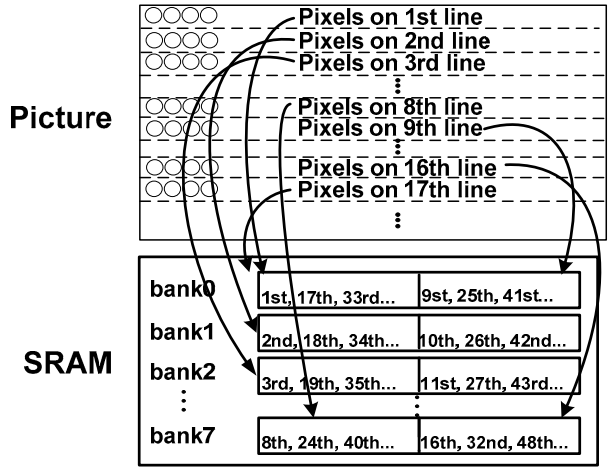



Figure 4. Pixel mapping scheme in proposed SRAM configuration

A block diagram of an SRAM bank is shown in Fig. 5. A X-decoders of the left block and the right block are merged to reduce the area and power. The merged X-decoder asserts GWL which is shared by two blocks. However, two blocks can be accessed independently with an aid of AND circuits which are switched by block control signals. This method reduces the number of X-decoders by half.

## 4. HORIZONTAL SEGMENTATION-FREE ACCESS

The segmentation-free accessibility and horizontal sub-sampling is achieved by specific decoding architecture and pixel mapping. Figure 6 shows the schematic of a left block and mapping manner of pixels on a line. Pixels on a line are mapped on the block at intervals of $n$ (= 8) pixels. Logical values of local word lines (LWLs) are decided by AND operation between a global word line from X-decoder and LWL select lines (LWLSLs) from Y-decoder. When one GWL was asserted, the next GWL were also asserted by the modified X-decoder. A block diagram of the modified X-decoder is illustrated in Fig. 7. An insertion of OR circuits enables X-decoder to activate two adjacent GWLs. LWLs are selected by two GWLs and switching 2 × $n$ (2 × 8) LWLSLs. Read circuit selects two pixels from four pixels in LWL. Consequently, any consecutive or sub-sampled horizontal 8 pixels can be read out. Figures 8 and 9 illustrate a segmentation-free 8-pixel access and horizontally sub-sampled 8-pixels access which can be realized by combination of GWLs and LWLSLs. In these figure, black lines mean signals in active. 2$n$ (= 16) X-decoders with the conventional SRAM scheme are reduced to one X-decoder in the proposed SRAM block.

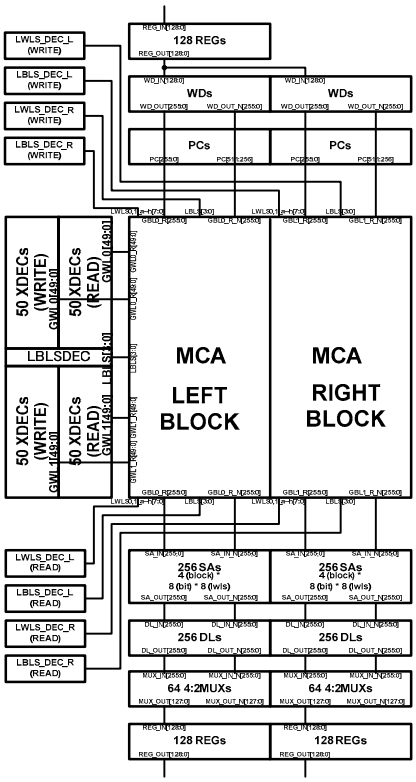



Figure 5. Block diagram of the proposed SRAM bank

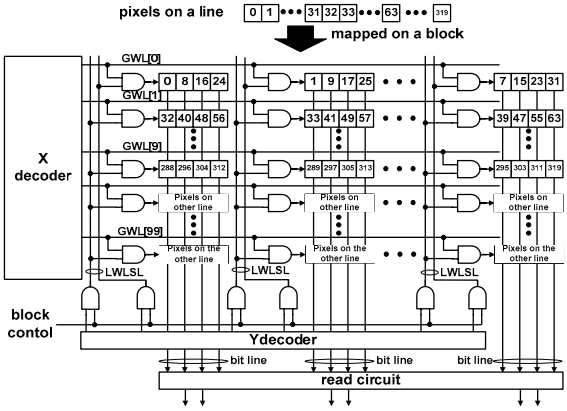



Figure 6. Pixel mapping to the SRAM block

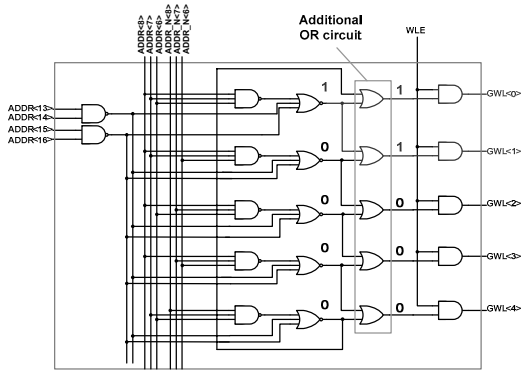



Figure 7. Block diagram of the modified X-decoder

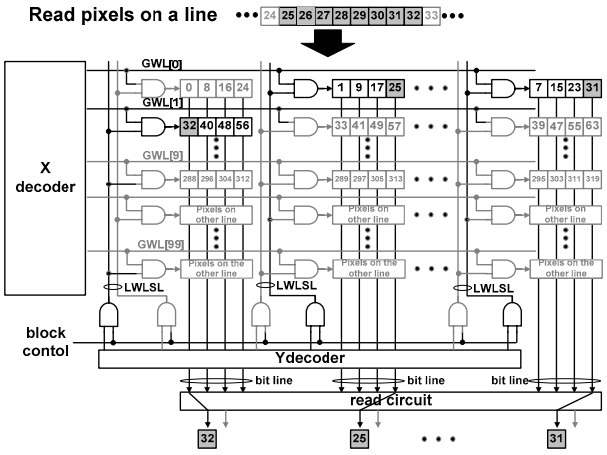



Figure 8. Horizontal segmentation-free access

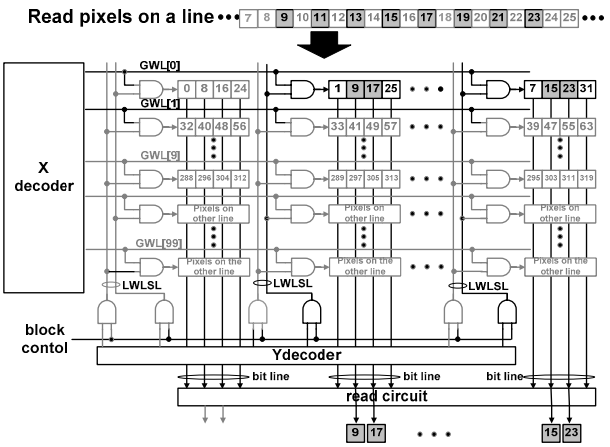



Figure 9. Horizontal segmentation-free sub-sampling access

## 5. Arbitrary Rectangular Access

The proposed SRAM provides the rectangular accessibility at any pixel position. All of $n \times m$ pixels in 4 forms that are $n \times m$ rectangle (integer-pel), $n \times m$ rectangle (horizontally sub-sampled), $n \times m$ rectangle (vertically sub-sampled) and $n \times m$ rectangle (horizontally and vertically sub-sampled) are fetched in a single cycle from the SRAM. Figures 10 and 11 shows two kinds of access manners in case of 8×8 pixels. Figure .10 is for rectangular $n \times m$ (integer-pel) access and Fig .11 is for rectangular $n \times m$ (horizontally and vertically sub-sampled) access. Hence the vertical consecutive $n$-line access, the vertical segmentation-free access and the vertical sub-sampling access are supported by combination of $2m$ SRAM blocks, while horizontal access schemes are realized in each SRAM block. While conventional SRAM macros require 2n × 2m X-decoders for any rectangular sub-sampling access, the proposed SRAM achieves this functional accessibility using only m X-decoders.

## 6. Design Example of Video Processor Using the Proposed SRAM

The proposed SRAM embedded as a search window buffer into H.264 main profile at level 4.1 interger-pel motion estimation (IME) processor core which performs motion vector detection for HDTV-30fps video with sub-100mW power in 90-nm CMOS process technology at 1.0-V supply voltage and 100-MHz operating frequency [3]. A plot of the proposed SRAM bank is shown in Fig. 12. The area of proposed SRAM bank is 2.0 × 0.15 mm$^2$. The search window buffer consists of eight SRAM banks and the total area is 2.0 × 1.2 mm$^2$. 8 × 8 pixels are accessed in a cycle and the total memory capacity is 320 × 160 Byte. The power of the proposed SRAM is estimated to be 20.3mW using nanosim simulator. Figure 13 shows clip layout of the processor core and proposed SRAM.

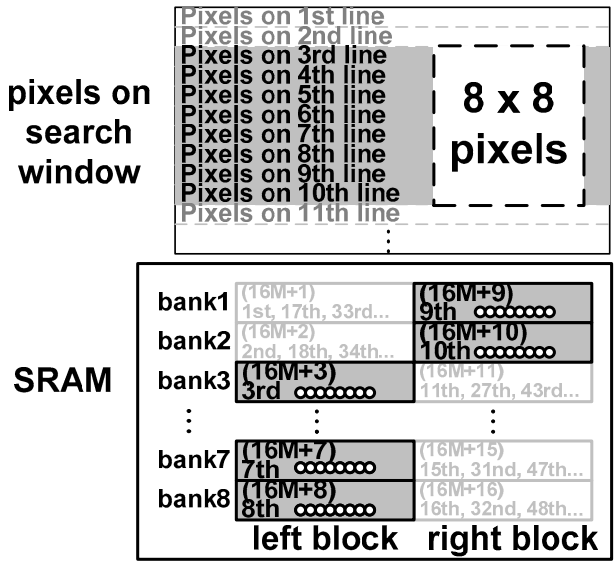



Figure 10. Access operation of 8 × 8 pixels (integer-pel)

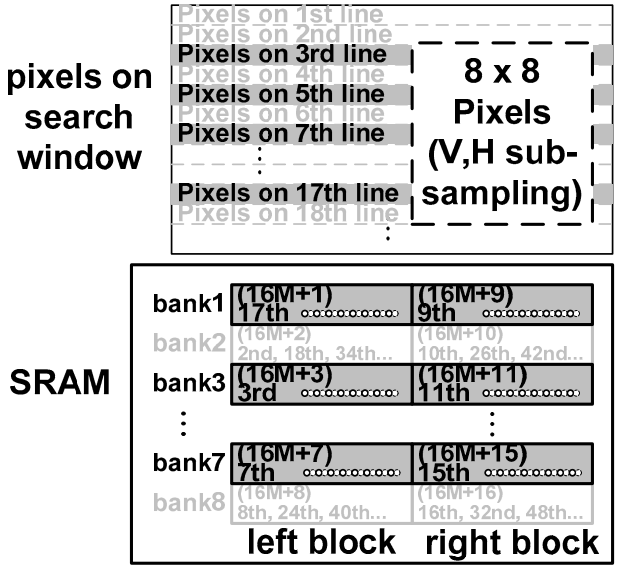



Figure 11. Access operation of sub-sampled 8 × 8 pixels (horizontally and vertically sub-sampled)

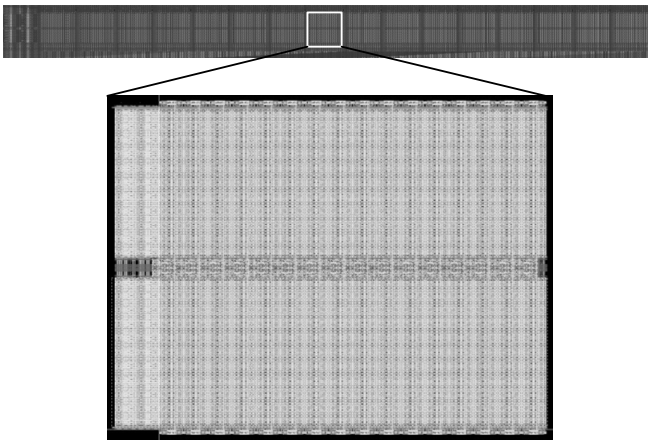


Figure 12. Layout of the proposed SRAM bank

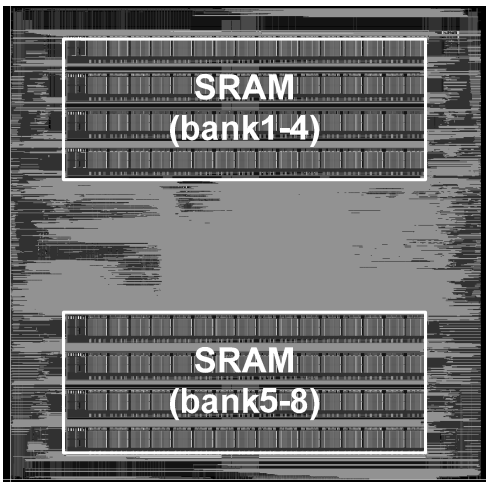



Figure 13. Layout of H.264 main profile at level 4.1 IME processor core

## 7. Power and Area Comparison

A performances of the proposed SRAM is evaluated and compared with 2 types of conventional methods assuming that these SRAM provide rectangular accessibility in 4 forms – 8 × 8 (integer-pel), 8 × 8 (horizontally sub-sampled), 8 × 8 (vertically sub-sampled), 8 × 8 (horizontally and vertically sub-sampled) and memory capacity is 320 × 160 Byte. Conv.1 consists of 256 SRAMs for divided 256 banks and conv.2 consists of 16 SRAMs for divided 16 banks at the expense of twice of cycle count. The performance comparisons are done on 90-nm CMOS technology with 1.0-V supply voltage and 100-MHz operating frequency. Figure 14 shows normalized power per access and area. The proposed SRAM reduces the power by 49 % and the area by 48 % compared with conv.1. Conv.2 suffers from high power consumption because 2-cycle accesses with x2 frequency are needed. The power of proposed SRAM is less than 50 % compared with conv.2. Furthermore, the power in case that the proposed one is employed for the search window buffer for H.264 motion estimation (ME) processor is estimated and summarized in Fig. 15. Since the number of parallelism in ME data-path is increased as resolution is higher, the proposed one is getting more effective for future high resolution video. The proposed SRAM reduces power and area by 55 % and by 53 % compared with conv.1 at 4096 parallelism for super-HDTV resolution (7680 × 4320).

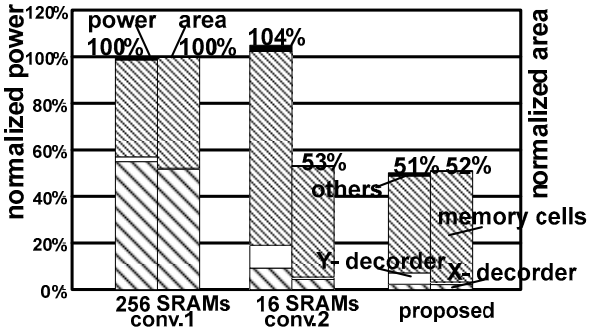
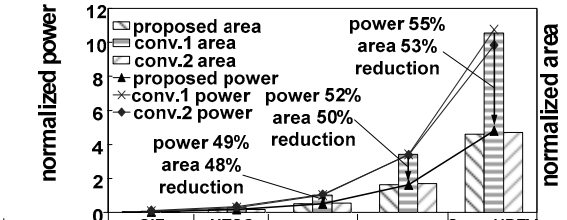



Figure 14. Area and power comparison




| Resolution (pels) | CIF (352x288) | NTSC (720x480) | HDTV (1920x1080) | QuadHDTV (3840x2160) | SuperHDTV (7680x4320) |
|---|---|---|---|---|---|
| #parallelism | 16 | 64 | 256 | 1024 | 4096 |
| Search window (pels) | 64x64 | 128x128 | 320x160 | 512x320 | 896x512 |

Figure 15. Power and area estimation for super-parallel video processing

## 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] J. Miyakoshi, Y. Murachi, T. Ishihara, H. Kawaguchi, and M.Yoshimoto, "A Power- and Area-Efficient SRAM Core Architecture with Segmentation-Free and Horizontal/Vertical Accessibility for Super-Parallel Video Processing," *IEICE Trans. Electron.*, vol.E89-C,no.11, pp.1629-1636, November 2006.

[3] Y. Murachi, K. Mizuno, J. Miyakoshi, M. Hamamoto, T. Iinuma, T. Ishihara, F. Yin, J. Lee, T. Kamino, H. Kawaguchi, and M. Yoshimoto "A Sub 100 mW H.264 MP@L4.1 Integer-Pel Motion Estimation Processor VLSI for MBAFF Encoding", *2008 IEEE International Symposium on Circuits and Systems*, May 2008 (to be held).