A study on low-energy memory architecture for image processors

Mori, Haruki

(Degree)
博士（工学）

(Date of Degree)
2019-03-25

(Date of Publication)
2021-03-25

(Resource Type)
doctoral thesis

(Report Number)
甲第7517号

(URL)
https://hdl.handle.net/20.500.14094/D1007517

Doctoral Dissertation

A study on low-energy memory architecture for image processors

画像処理プロセッサ向け低エネルギメモリアーキテクチャに関する研究

平成 31 年 1 月

神戸大学大学院システム情報学研究科

Haruki Mori
森 陽紀

Abstract

This dissertation reports low-energy and low-cost memory architecture for real-time and energy efficient image recognition application.

Chapter 1 shows research background of this dissertation and fundamental characteristics of the multi-port static random access memory (SRAM) as an application specific image memory for the real-time motion detection processor. Then, the fundamental features of distributed deep learning and its memory system for image recognition is also described in this chapter.

In Chapter 2, intrinsic features and issues in the multi-port SRAMs as image memory and in the memory architecture of deep learning processor is described. Where, an increased SRAM operating active energy in read/write cycles and the unnecessary energy consumption on bitline (BL) in SRAM array, and an exponentially increased memory capacity and bandwidth in distributed deep learning processor, are explained. This dissertation presents a low-energy and low-cost SRAM circuit designs, and the high scalable and energy efficient deep learning algorithm/hardware development overlooking whole data-flow and memory architecture optimization.

Chapter 3 describes 1-write/2-read eight-transistor (8T) three-port SRAM design; a novel 1-write/2-read three-port SRAM based on the 8T bitcell, and the combination with majority-logic are presented.

1) This study presents a low-energy and low-voltage 64-kb 8T three-port image memory using 28-nm FD-SOI process technology. Our proposed SRAM accommodates eight-transistor bit cells comprising one-write/two-read ports and a majority logic circuit to save active energy. The test chip operates at a supply voltage of 0.46 V and access time of 140 ns. The minimum energy point is a supply voltage of 0.54 V and an access time of 55 ns (=18.2 MHz), at which 484 fJ/cycle in a write operation and 650 fJ/cycle in a read operation are achieved assisted by the majority logic circuit. These factors are 69 % and 47 % smaller than those in a conventional 6T SRAM using the 28-nm FD-SOI process technology. Furthermore, the operating energy consumed on the proposed SRAM is saved by 290 μW, which signifies 24 % of energy reduction in total over the conventional H.264 motion estimation

image processor.

Chapter 4 presents two-types of BL swing reduction techniques for low-energy 8T dual-port SRAM; 1) a selective sourceline drive (SSD) scheme with the consecutive memory access and 2) an MSB-based inversion logic for low-energy 8T dual-port SRAM in image processing.

1) This study presents a low-energy 64-kb 8-transistor (8T) one-read/one-write dual-port image memory with a 28-nm fully depleted SOI (FD-SOI) process technology. Our proposed SRAM adopts a selective sourceline drive (SSD) scheme and a consecutive data write technique for improving active energy efficiency at low voltage. The novel SSD scheme controls sourceline voltage and eliminates leakage energy at unselected columns in read operations. We fabricated a 64-kb 8T dual-port SRAM in the 28-nm FD-SOI process technology. The 8T SRAM cell size is 0.291 × 1.457 μm^2. The test chip exhibits 0.48-V operation at access time of 135 ns. The energy minimum point is at a supply voltage of 0.56 V and an access time of 35 ns, where 265.0 fJ/cycle in write operations and 389.6 fJ/cycle in read operations are achieved. These factors are, respectively, 30% and 26% smaller than those of the 8T dual-port SRAM with the conventional scheme.
2) This study presents low-energy 8T dual-port SRAM with a novel MSB-based (most-significant-bit-based) inversion logic for an image processor such a deep-learning processor. Our proposed SRAM is suitable for real-time and low-power image processing, in which data have statistical correlation and data bit reordering are exploited. The proposed MSB-based inversion logic eliminates an additional flag bit in a majority logic; the MSB digit in an input datum judges whether or not to invert the datum. Thus, the area overhead of 16.1 % for the 8-bit conventional majority logic is dramatically saved. The area overhead of the proposed SRAM is merely 1.8 % for the MSB-based inversion logic. We verified that, with the proposed technique, 14.7 % of total energy can be saved in a 28-nm 64-kb FD-SOI SRAM when a set of images is read out. Furthermore, the saving factor is extended to 17.3 % when image processing in the VGG-F convolutional neural network (CNN) is considered, where 312.4 fJ/cycle in the read operation is achieved.

In Chapter 5, memory bandwidth and capacity reduction techniques with the model parallelism for high scalable distributed deep learning are described. Where, a layer-block-wise pipeline stochastic gradient decent (SGD) algorithm and its hardware architecture are proposed for distributed deep learning.

1) This study presents a pipelined stochastic gradient descent (SGD) algorithm and its hardware architecture with a memory distributed structure. In the proposed architecture, a pipeline stage takes charge of multiple layers: a "layer block". The layer-block-wise pipeline has much less weight parameters for network training than conventional multithreading because weight memory is distributed to workers assigned to pipeline stages. The memory capacity of 1.95 GB for the four-stage proposed pipeline is approximately half of the 3.79 GB for multithreading when a batch size is 32 in VGG-F network model. Unlike multithreaded data parallelism, no parameter server for weight update or shared I/O data bus is necessary. Therefore, the memory bandwidth is drastically reduced. The proposed four-stage pipeline only needs memory bandwidths of 36.3 MB and 17.0 MB per batch, respectively, for forward propagation and backpropagation processes, whereas four-thread multithreading requires a bandwidth of 1.21 GB overall for send and receive processes to unify its weight parameters. At the parallelization degree of four, the proposed pipeline still maintaining training convergence by a factor of 1.76, compared with the conventional multithreaded architecture although the memory capacity and the memory bandwidth are decreased.

In the final chapter 6, we summarize this dissertation. This thesis proposes the low-energy and low-cost memory architecture to realize high-speed and high-scalable image processing overlooking whole memory architecture. The work contributes to achieve an energy-efficient SRAM design for advanced technology and development of high-speed and energy-efficient image processing flame work with higher scalability.

Keywords: 8T SRAM, 28-nm SRAM, Consecutive Access, FD-SOI, Image Memory, Low Power, Multi-Port SRAM, Majority Logic, MSB-Based Inversion Logic, Deep Neural Network, Model Parallelism, Pipelined Backpropagation, Distributed Memory, Memory Capacity Reduction, Memory Bandwidth Reduction.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Background of Research Area

1.1.1 The role of SRAM architecture in image processing

The work for this dissertation is twofold. First, I implemented low-power and low-voltage embedded static random access memory (SRAM) design aiming high energy efficiency on real-time image recognition processor. I also developed a high-speed and an energy-efficient distributed deep-learning (DL) algorithm and its hardware accelerator overlooking whole data flow and memory architecture optimization.

Low-energy image recognition is demanded for internet of things (IoT) devices in various fields such as safety driving systems, machine vision and augmented reality (AR) systems with fine resolution. Image resolution enhancement requires large memory capacity and large chip area. It also entails higher energy consumption because of the increased amounts of image data that must be processed. The memory capacity and area cost is relatively increased than the cost of logic part and other peripheral circuit according to the technology scaling. In fact, the power consumption in memory (global memory, caches, and register files) dissipates more than 40 % of the energy of the image processor (e.g. GTX 580, Nvidia Corp.) [1]. For IoT devices handling image information, more energy-efficient memory technology is anticipated.

SRAM is the most common type of the embedded memories for the modern SoCs. SRAM has better compatibility to logic circuits and faster random access performance than the other memories. Processors leverages SRAM as the first level cache memory, a scratch pad memory and a main memory on a video coding system and image recognition system. As process technology is scaled down, SRAM occupies over 50% of the total area and 65% of the total power in 2022 [2], as shown in Fig. 1.1. Therefore, SRAM must play an integral role in the power, performance, and an area of the modern SoCs.

Input data for image processing are stored temporarily in SRAM. In an image processor, many processing cores access SRAM for multi-thread processing. Fig. 1.2

portrays the memory system in an image processing unit. The SRAM array stores data such as image maps, feature maps, and various parameters for its processing on the many processing elements (PE). Demand for multi-port SRAMs has increased to accommodate high-speed, low-energy image processing. The multi-port SRAM is suitable for parallel operation. It improves the total chip performance and/or memory bandwidth by enabling multiple simultaneous operations in the same bank [3]. Parallel processing is a key technology for real time image applications that require embedded memories with multiple access ports [4–6]. However, the energy reduction of SRAM part is remaining important challenges for future IoT devices. Actually, as the transistor scaling, larger SRAM capacity than ever will be implemented in an SoCs. Fig. 1.3 shows the energy consumption of SRAM in whole image processor. 43% of energy is consumed in SRAM part manufactured by 65 nm process [7], while the energy consumption of SRAM part is increased to 60% in 28 nm process technology. To date, multiport SRAMs that support simultaneous write and read operations with low-energy operation have been proposed for use as image processors [8–10].

Low-voltage operation is one of the primary challenges for active-power improvement, but the minimum operating voltage (V_{min}) of SRAM part is remaining still higher than the V_{min} of logic part in SoCs. Fully depleted silicon on insulator (FD-SOI) process technology is one of the promising ways to provide high-speed and low-voltage SRAM [11, 12]. Figure 1.4 depicts the simplified device structure in the FD-SOI process technology. A 28-nm FD-SOI process technology is a fine process, that has a fully depleted transistor with an ultra-thin silicon body and a thin buried oxide (BOX) layer, giving them excellent electrostatic control in near-threshold voltage region. Therefore, it brings stable features at low-voltage operation. The BOX layer reduces the leakage current by controlling the electrical flow in a transistor from a source node to a drain node. Moreover, the BOX layer suppresses the parasitic capacitance between the source node and the drain node. These features of the 28-nm FD-SOI process technology realize the production of ultra-low-power SRAM design [13–17].

Energy efficiency of SoCs is improved in the near-threshold region because dynamic energy and leakage energies are well balanced [18]. The combination of a low threshold voltage and low supply voltage is beneficial for high-activation logic circuitry, whereas a high threshold voltage and a high supply voltage are suitable for memory operations.

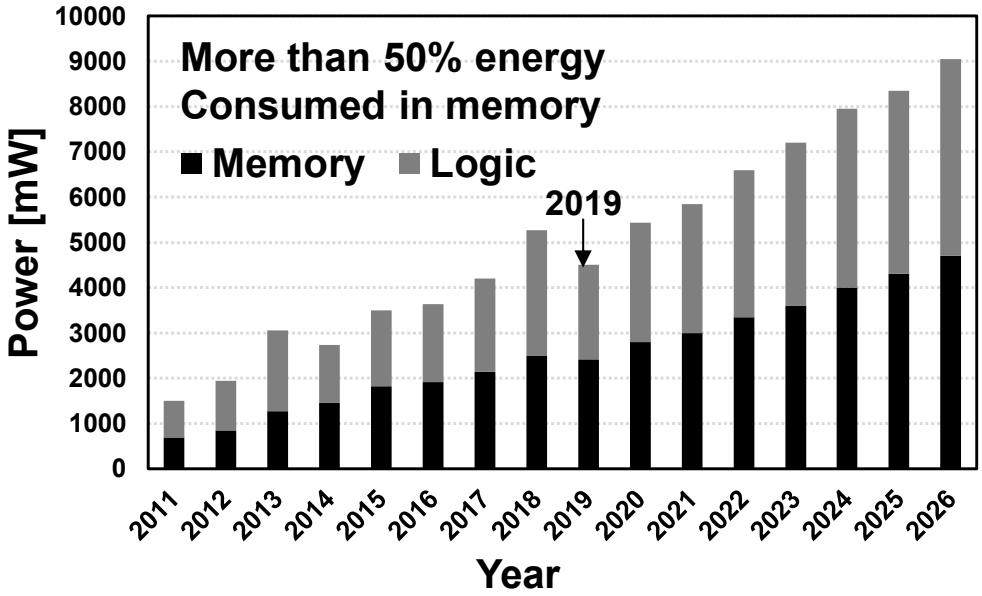


Fig. 1.1 Trend of power consumption in SoCs; the memory consumes over 50% of whole energy in image processor.

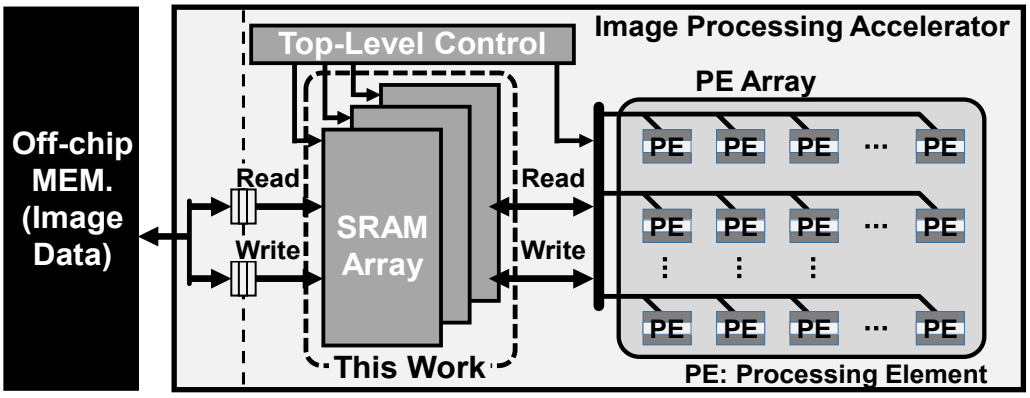


Fig. 1.2 The memory system in image processor.

For memory, the activation is low because only a selected wordline (WL) and certain bitlines are activated. In such cases, the high threshold voltage suppresses the leakage current and total energy. Process technologies such as fin field-effect transistor (Fin-FET) and FD-SOI have a smaller *S*-factor. Moderate threshold voltage and moderate supply voltage achieve the best scenario, especially for memory [19–22]. However, the SRAM also tends to be affected by process variation (local variation and global variation) in the low voltage region. The low-power and proper SRAM circuit design is more important than ever. Thus, consequently, SRAM has imperative issues of energy dissipation, minimum operating voltage, and variation effect in the deep submicron technology.

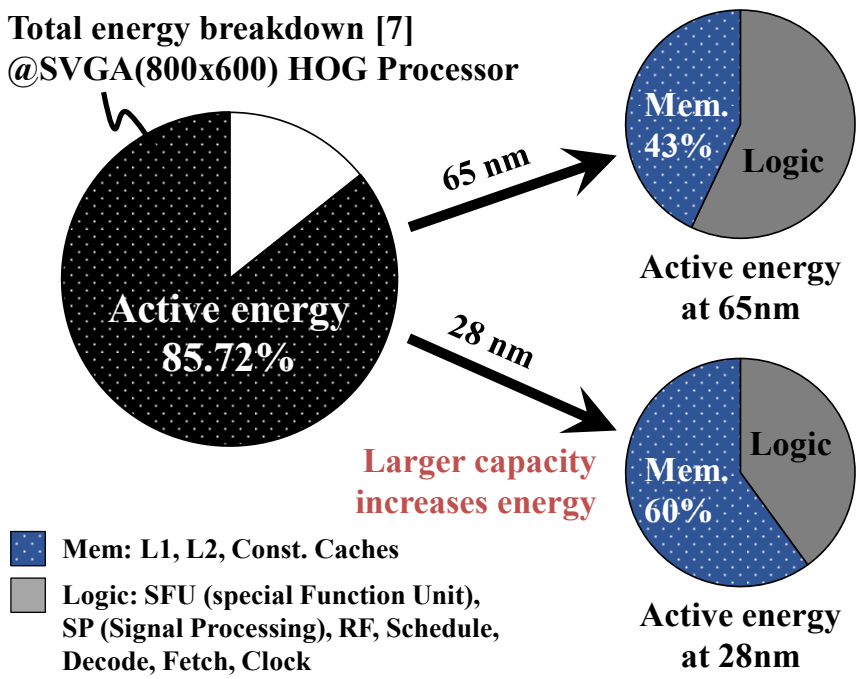


Fig. 1.3 The energy breakdown in the HOG processor; the memory has impact its energy performance.

- **FD-SOI has been adopted in the 28nm node [11].**
  - Improving transistor behavior, especially at low supply voltage.
- **28nm FD-SOI enables an ultra-low power SRAM design [14].**
  - Thicker gate dielectric reduced gate leakage.

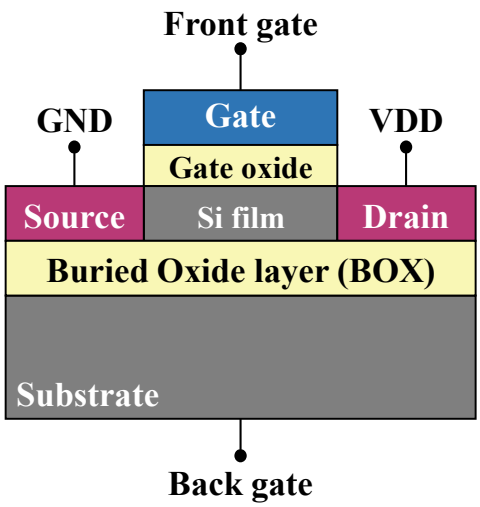


Fig. 1.4 The device structure of FD-SOI process technology; the BOX layer and ultra-thin silicon body provides better electrostatic behavior.

1.1.2 Background of memory architecture in Deep-Learning image recognition processor

The perceptron, a primitive artificial neural network, has a single layer comprising input synapses and output neurons as nonlinear activations [23]. The multilayer perceptron is an extension of the single-layer perceptron with hidden layers, in which training is conducted through backpropagation [24]. A convolutional neural network (CNN) imitates part of the human visual cortex in the cerebrum. It is an extension of a multilayer perceptron.

Recently, a deeper network having more than three layers is generally called as a "deep neural network (DNN)" or "deep learning". The DNN has exhibited its potential for image recognition ability. Its accuracy is improving year by year. At the ImageNet Large Scale Visual Recognition Competition (ILSVRC), AlexNet with five CNN layers and three fully connected layers made an overwhelming achievement over conventional feature-based image recognition schemes in 2012 [25]. Its top-five error rate was 15.3%, which was more than 10 % better than the second-best entry based on the handmade features. Since 2012, the error rate in image recognition has been improved by DNNs. At ILSVRC 2015, ResNet, comprising with 151 CNN layers and one fully connected layer, remarkably won the competition by a top-five error rate of only 3.57 % [26],

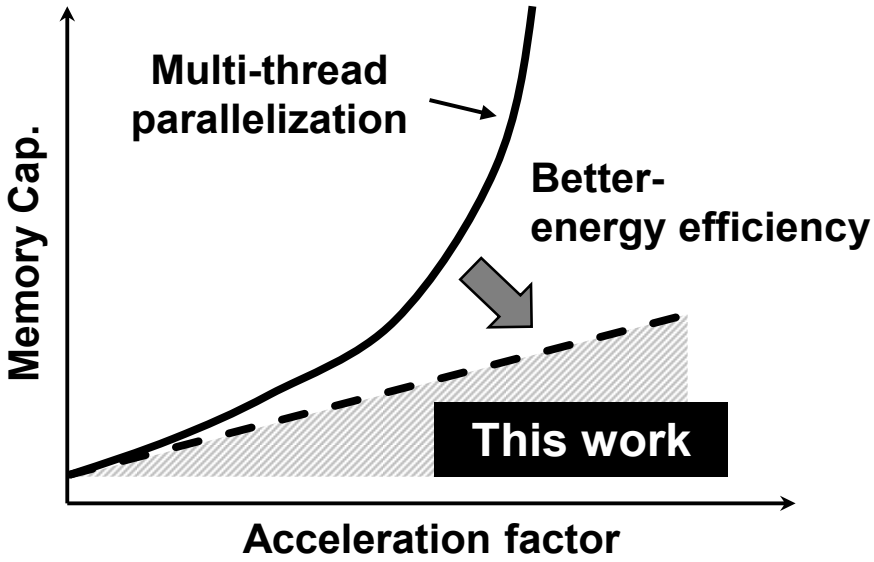


Fig. 1.5 The relationship between the total memory size and the acceleration factor when using highly parallelized deep learning; this work targeted high scalable and energy efficient deep learning.

which is better than the 5.1% figure representing human ability [27]. Today, DNNs are applied mainly to image recognition applications, but DNNs themselves has general-purpose characteristics and abilities; DNNs are now attracting attention not only in for engineering, but also for use in medicine, pharmacy, and biology applications [28]. The development of deep learning technology is expected to contribute to the improvement of wide range industrial fields.

Figure 1.5 shows a trend of memory capacity versus acceleration factor in distributed deep learning. In the conventional multi-thread parallelism, the memory capacity increases lineally. However, because of the memory bandwidth of communication between the parameter server and many workers, the acceleration factor saturates in the middle of parallelization degree. In this case, it is not faster even with a lot of graphics processors. The conventional parallelism has scalability constraints in its memory structure. The communication delay on the data bus and the huge memory capacity with duplicated network, which degrades scalability and increase energy consumption. When considering the future networks that have numerous layers must be trained by numerous users in various industrial fields, the existing computing resource is not sufficient to satisfy the deserved accuracy, although we use the highly paralleled GPGPUs. In order to improve such issues, more high-scalable algorisms and low-cost hardware architecture is anticipated. Therefore, our target is low-cost and high-scalable deep learning flame work.

1.2 Objectives of This Study

This dissertation focuses on application specific design of memory architecture for low-power, real-time and high-scalable image processors.

The first objective of this study is active energy reduction in read and write operation of multi-port SRAMs. The multi-port SRAM is suitable for parallel operation. In particular, an image processor requires larger multi-port SRAM capacity. Therefore, its energy consumption is drastically increased by higher resolution. Consequently, multi-port SRAMs with lower active or standby energies have become more important than ever. All the SRAMs in this dissertation employ eight-transistors (8T) multi-port SRAM bitcells (the 8T three-port SRAM, and two-types of 8T dual-port SRAM) with improved active or leakage energy.

The second objective is to decrease leakage energy consumption for further energy reduction. In submicron process technologies, the leakage and unnecessary current are more critical than those by larger technology nodes. This study presents selective sourceline (SL) drive circuit technics to eliminate unnecessary current in unselected read bitlines (RBLs) and to effectively improve the energy efficiency in read operation. Another dual-port design proposes the most significant bit (MSB) based inversion logic. This circuit technique reduces the number of RBL swing for low-energy operation especially in the image recognition deep learning tasks.

The third objective is to decrease the memory bandwidth and the memory capacity in the distributed deep learning. The highly parallelized deep learning for image recognition is suffering from the scalability deterioration by increased data bus communication between a lot of parallel workers. The large amount of memory bandwidth increases communication delays in data bus and therefore computational time is also increased. Furthermore, it entails higher energy consumption in memory with increased number of memory access. The objective of this study is to develop a high throughput deep learning system which accelerates the training period. The co-design of algorism and hardware architecture enables development of optimized memory architecture and parallel data flow. For this purpose, we propose a novel layer-block-wise-pipeline algorism and its architecture to reduce memory bandwidth and memory capacity with segmented data bus architecture and distributed memory architecture. In this dissertation, we discuss multiple versions of parallelization models and compare them.

1.3 Overview of This Dissertation

Figure 1.6 illustrates an overview of this dissertation. Firstly, I explain the background, objectives, and overview of this study in chapter 1. Secondly, the issues of memory architecture in image processing technique are presented in chapter 2. In this dissertation, the novel techniques are explained to address the issues which are denoted in chapters 3, 4, and 5.

Chapter 3 presents one-write/two-read (1W/2R) 8T three-port SRAM design and implementation. Here, a novel 1W/2R three-port SRAM with 8T bitcell is proposed. The combination of the proposed SRAM and a majority logic circuit exhibits

low-energy performance. The proposed 8T three-port SRAM accommodates eight-transistor bit cells comprising one-write/two-read ports and a majority logic circuit to save active energy. We fabricated a 64-kb 8T three-port SRAM using 28-nm FD-SOI process technology and compared it with conventional ten-transistor (10T) three-port SRAM in ME264 (with H.264 codec) motion estimation image processor.

In Chapter 4, two types of BL swing and leakage reduction technique for low-energy 8T dual-port SRAM. Firstly, a low-energy 8T dual-port image memory with sourceline drive technique is presented. The proposed 8T dual-port SRAM with selective sourceline drive (SSD) scheme improves active energy efficiency at the low- voltage. We implemented a 64-kb 8T dual-port SRAM in the 28-nm FD-SOI process technology. Secondly, an 8T dual-port SRAM with a novel most significant bit (MSB) based inversion logic is presented to save the active energy on RBLs. Our proposed SRAMs are advantageous for real-time and low-power image processing, in which data have statistical correlation. Furthermore, the proposed MSB based inversion logic eliminates an additional flag bit, therefore, our SRAMs have smaller area overhead than the conventional scheme.

Chapter 5 introduces memory bandwidth and capacity reduction techniques for high-scalable parallelism in distributed deep learning. The layer-block-wise pipeline algorithm and its hardware architecture are presented to speedup the stochastic gradient descent (SGD) algorism with low cost. In the proposed architecture, a pipeline stage takes charge of multiple layers: a “layer block”. The layer-block-wise pipeline has much less weight parameters for network training than conventional multithreading because weight memory is distributed to workers assigned to pipeline stages. Unlike multithreaded data parallelism, no parameter server for weight update or shared I/O data bus is necessary. Therefore, the memory bandwidth and internal memory capacity are drastically reduced in the deep learning processors.

Finally, the conclusion of this dissertation is summarized in Chapter 6. This thesis consistently presents the energy efficient and low-cost memory architecture for high-speed, low-energy and high-scalable image processors overlooking whole memory architecture in future image applications.

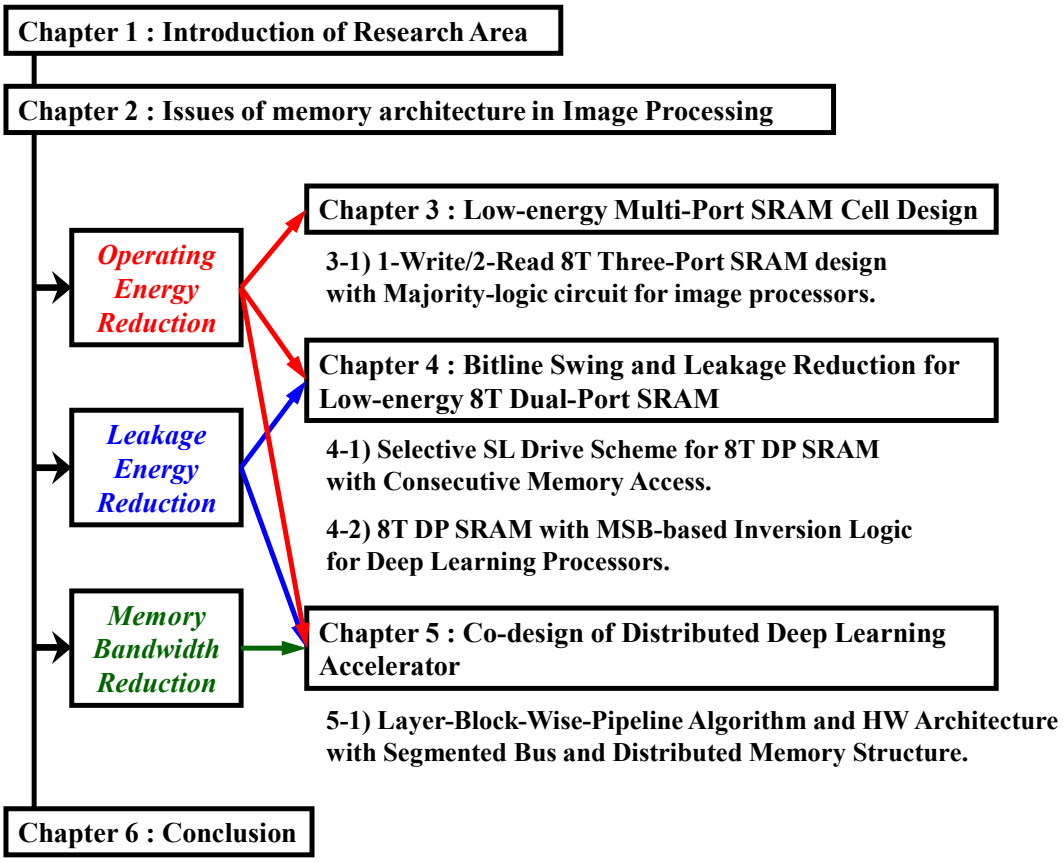


Fig. 1.6 Overview of this dissertation.

Chapter 2 Issue of Memory Architecture in Image Processors

As described in the previous chapter, the memory architecture plays significant role to optimize memory flow in the image processing. It influences energy, area, and the whole performance in the image processors. In this chapter, the specific features and inherent issues of memory architecture in image processing are summarized. To optimize the energy performance, the increased active or leakage energy in multi-port SRAM are discussed. Also, exponentially increased memory bandwidth and capacity caused by the conventional distributed neural network training and those impact in the computational cost must be explained.

Firstly, we describe the conventional SRAMs as image memory. Here, the features of 6T, 8T, and 10T SRAM bitcells and its design issues are explained. Secondly, we focus to the advanced features of dual-port SRAM and its inherent issues. The unnecessary current and leakage current on the read-port are primary issues in the dual-port SRAMs. RBL swing in the dual-port SRAM also increases active energy. In the deep learning processors, the issues of increased memory bandwidth and capacity must be addressed for future deep learning tasks. The enormous bus communication and drastically increased internal memory capacity degrade scalability of parallelism in deep learning. This chapter notes the primary issues and important challenges to be tackled for future energy-efficient image processing.

2.1 Features and issues in the SRAM architecture

2.1.1 Fundamental features of multi-port SRAM

Figure 2.1 (a)-(b) illustrates variation of commonly composed single-ported and multi-ported 6T, 8T, and 10T SRAM bitcells. Fig. 2.1(a) shows the 6-transistors (6T) SRAM bitcell. The 6T SRAM bitcell consists of two pMOS pull-up (load) transistors (M1, M3), two nMOS pull-down (driver) transistors (M2, M4), and two nMOS pass-gate (access) transistors (M5, M6). This type of SRAM bitcell is most generally used in the high performance SoCs, because of symmetrical structure. It can achieve

high density integration and quick generation by SRAM compilers. Most of conventional embedded SRAMs are based on the 6T single-port SRAM. The cross coupled inverter circuit holds a bit data as a storage node. The pair of the BLs enables either single write or read (1WR) operation during certain access time. This BL pair is precharged to VDD every cycle before operation and standby mode. The WL is activated for upcoming write or read operation. The BL pair supports quick differential sensing which is effective for high-speed and energy-efficient operation (=small signal sensing). In the 6T SRAM, wordlines (WLs) and bitline-pairs (BLs and BLBs) are vertically and horizontally assigned in the SRAM array.

While the SRAM architectural idea, such as SRAM matrix duplication, can be used to support more than 1WRs in single operation, larger number of transistors are required to realize multi-port functionality, results of certain layout area overhead. Generally composed examples such multi-ported function, include the 8T 2WR dual-port SRAM cell shown in Fig. 2.1(b). It has two pairs of nMOS pass-gate transistors to support independent read and write operations. However, since BL pairs are commonly used for read and write cycle, well known half select disturb problem is remaining. The 8T 1W1R dual-port SRAM consists of 6T bitcell and a dedicated read-port to enable simultaneous write and read access, as depicted in Fig. 2.1(c). The dedicated read port is comprised of two nMOS transistors (M7, M8), this type of SRAM bitcell is commonly used for disturb-free read operation. This nMOS decoupled read port is called as single-ended read port. In this structure, since the SL of read port is generally connected to the GND, the RBL voltage has to fully charged and discharged in the read operation (=large signal sensing). Therefore, read energy on RBL charge/discharge in the single-ended read port is repeatedly consumed. Those features of 8T 1W1R dual-port SRAM with the single-ended read-port significantly increases active energy in the read operation. The extended version of 1W1R dual-port SRAM with dedicated read port realized with the 10-transistor (10T) bitcell is proposed as 10T 1W2R three-port SRAM. It has two single-ended read ports to realize one write and two read (1W2R) operations simultaneously. Thus, the multi-port SRAM can provide SRAM operation in parallel in the same bank, which is preferable for highly paralleled application such as matrix-matrix operations included in image processing. However, its bitcell structure leads larger area and higher energy consumption in the SRAM array.

Figure 2.2 shows the block diagram of SRAM bitcell matrix and peripheral control circuitry. The target cell in the matrix can be accessed by selecting the WL and BL pair, which aligned to horizontal and vertical directions. The row/column enable signals are generated by the X and Y address decoders, which respectively select a WL and a BL pair in the matrix. The sense amplifiers are enabled in the read cycle, which read the voltage difference between the BL and the BLB. Finally, the SRAM outputs the read out data (Dout) signals. In the write cycle, the BL pairs are precharged to VDD voltage before the data inputs. After that, BL and BLB is biased to opposite voltage level, such as (BL: VDD, BLB: GND) or (BL: GND, BLB: VDD) by the write drivers according to the input data. The selected WL is enabled by the WL drivers. The WL pulse activates pass-gate transistors in the bitcell, which is a trigger for the write operation to the target SRAM cell.

The multi-thread architecture is commonly used for the image processing applications to realize real-time processing, which can be adopted simply with two or more independent processing units. The execution of numerous matrix-matrix operations in parallel is effective way to accelerate the high-quality image or video processing. Figure 2.3 describes the block diagram of memory system in multi-core and multi-thread processor. In this example, the SRAM array is divided into two banks; the SRAM Bank-1 for the input image data and Bank-2 for the feature maps. To execute the filter operation in processing cores, the input data and feature map data are required simultaneously. Thus, the many processing cores access the SRAM simultaneously for multi-thread processing. In such case we explained here, demands for multi-port SRAM have been increased to realize high-speed and low-power image processor. The multi-port SRAM is reportedly suitable for plural core accesses. It significantly improves energy and total performance. To date, a multiport SRAM that supports simultaneous write and read operations is proposed for use as the image processor [5–6]. In those image processor, the number of read operations is drastically increased than write operations. To reduce the energy consumption and optimize whole throughputs, the energy reduction and read circuit improvement techniques are anticipated for such a multi-core image processor.

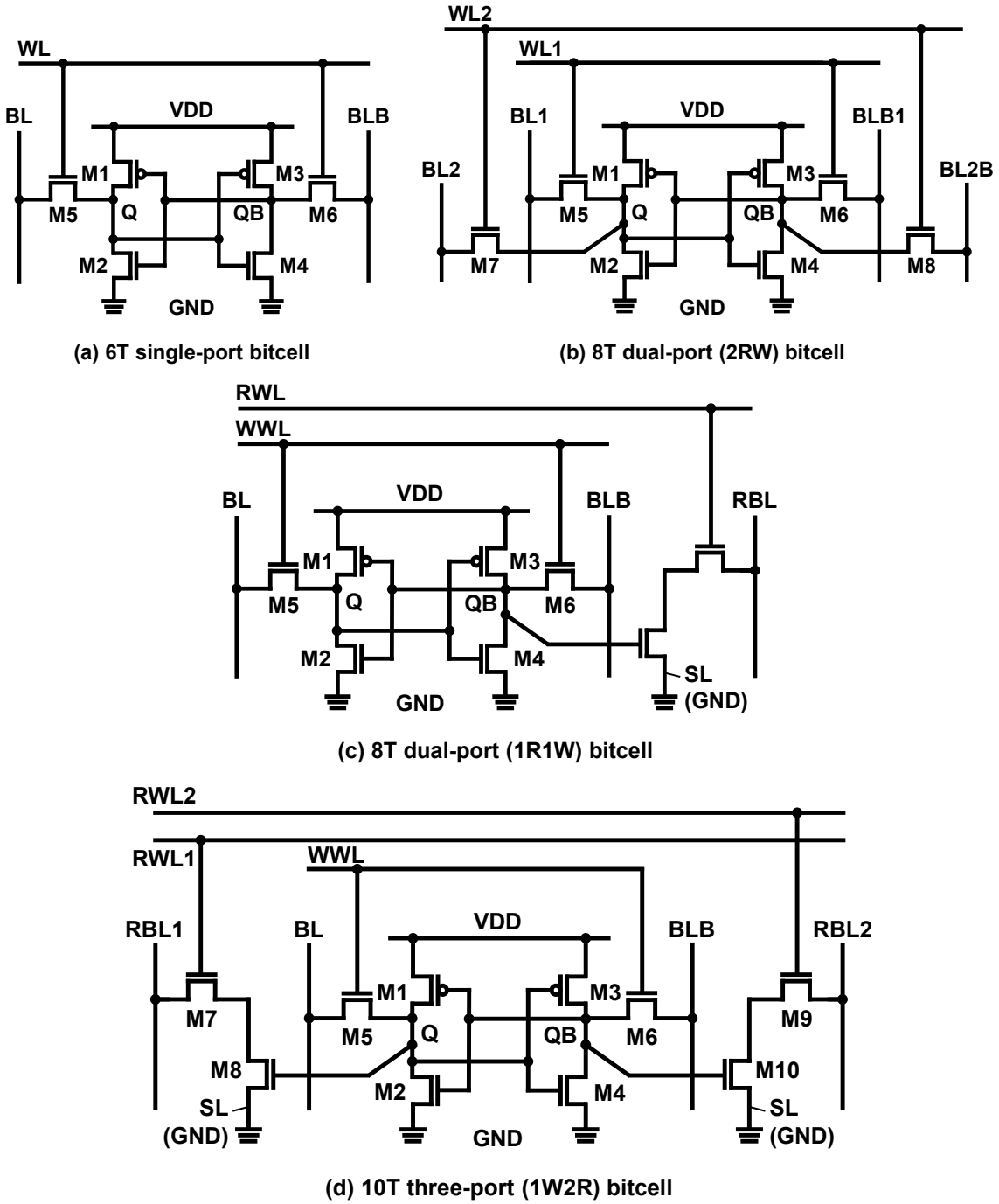


Fig. 2.1 Variation of commonly composed single-ported and multi-ported 6T, 8T, and10T SRAM bitcells.

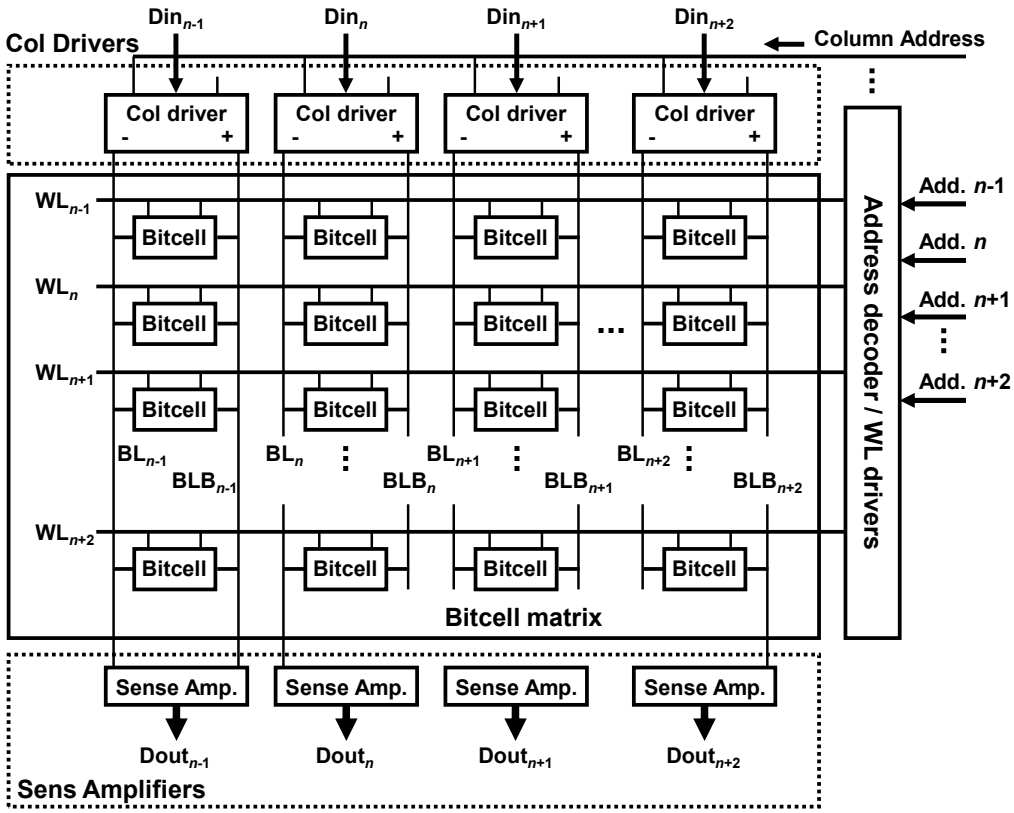


Fig. 2.2 Block diagram of SRAM array.

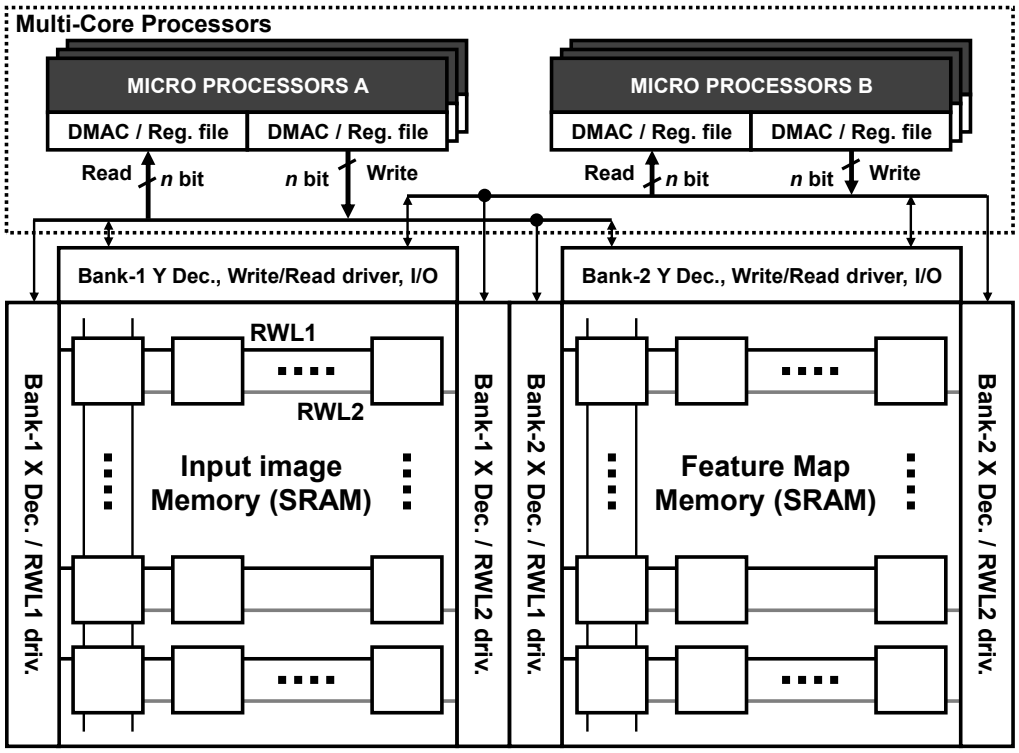


Fig. 2.3 Block diagram of memory system in the multi-core processor.

2.1.2 The issues in the multi-port SRAM

As described previously, the multi-port SRAM is commonly used for multi-core image processors. In this part, we focus the 1W1R dual-port SRAM, which enables simultaneous write and read operation without half select problems. A thing in multi-port SRAM to be considered is its increased area cost. The multi-port SRAM generally require larger area than the conventional 6T SRAM bitcell. Actually, as I shown in Fig. 2.1(d), the 10T three-port SRAM structure is also used as the 10T dual-port SRAM, when the RWL1 and the RWL2 is unified as an RWL. Then, the RBLs can be used as RBL pair. However, its bitcell size is increased significantly. In the high resolution image processing, the larger SRAM capacity is preferable. Thus, the SRAM bitcell area has large impact for entire area cost in SoCs. Therefore, in these dual-port SRAM, the dedicated read-port is mainly adopted as a single-ended structure. The single-ended read port with asymmetric bit-cell can be achieved lower area cost than the symmetrical one, thanks of the lower number of composed transistors.

In the single-ended read port structure with two or more nMOS transistors, the source line of read port is conventionally connected to the ground (GND) voltage. Therefore, the RBL must be fully amplified to read out the stored data when the storage node QB holds “1”. It is noteworthy that the single-ended read port has a strong data dependency on its energy consumption. Those feature of conventional single-ended read port, which possibly increases power consumption on the RBLs than the differential sensing scheme. This fact is remaining as an important issue in the dual-port SRAM.

The RBL timing problem is another factor. Because the RBL has to be fully discharged in the read operation, longer access time is necessary for the single-ended structure than the differential one. When considering the operation in sub-threshold region, since the *Vth* variation become worse, and setup/hold margin design should be more severe. Therefore, its operation frequency is lowered than the differential sensing scheme, because the single ended structure needs longer access time. Nevertheless, these differential sensing devices and symmetrical SRAM bitcells entail a greater area cost. To address these issues explained here, various important earlier works have proposed for the dual-port SRAM architectures. (I will explain the details in the chapter 4.)

As described in an earlier report, an 8T 1W1R dual-port SRAM is typically used for

leveraging disturb-free access because of the dedicated read port [29]. The 8T dual-port SRAMs with lower active/standby powers have become more important than ever. The 8T 1W/1R dual-port SRAM structure can eliminate the well-known read disturb problem by preventing charge sharing with internal storage nodes when a read WL is activated. A read port of the 8T dual-port SRAM employs a SL as a footer line, which is shared in the same row address to perform low-energy operations. Figure 2.4 shows the circuit schematic of the SL structure, and the current flow model of unnecessary read current at the unselected column in conventional 8T SRAM cells with single ended read ports. In the read operation, the RBLs are precharged to "1" and the RWL selects a target row address. At the selected column, the RBL voltage is discharged to "0", therefore, the sense amplifier (SA) can amplify the RBL voltage, in the case of Fig. 2.4(a). On the other hand, in the half selected column where RWL is shared with selected column, the RBL current is consumed unnecessarily although the column is not selected for the read out. This structure increases energy consumption on the RBLs, and degrades active energy efficiency. While the circuit techniques, the divided WL structure, can be adopted to the RWLs to reduce energy consumption, it requires additional peripheral circuitry and significantly degrades area performance. As our earlier work [30] proposes the selective sourceline control (SSLC) structure with 8T 1W1R dual-port SRAM, which is developed to reduce leakage current through unselected RBLs. However, some read bitlines are, still discharged slightly in unselected columns because the floating SLs of the conventional scheme [30] degrades energy efficiency.

In this dissertation, our proposed works address the multi-port functionality, area efficiency, variation effects in sub-threshold region, and energy efficiency from the point of circuit design to improve these issues in the 8T dual-port SRAM. The details are discussed in the Chapter 4.

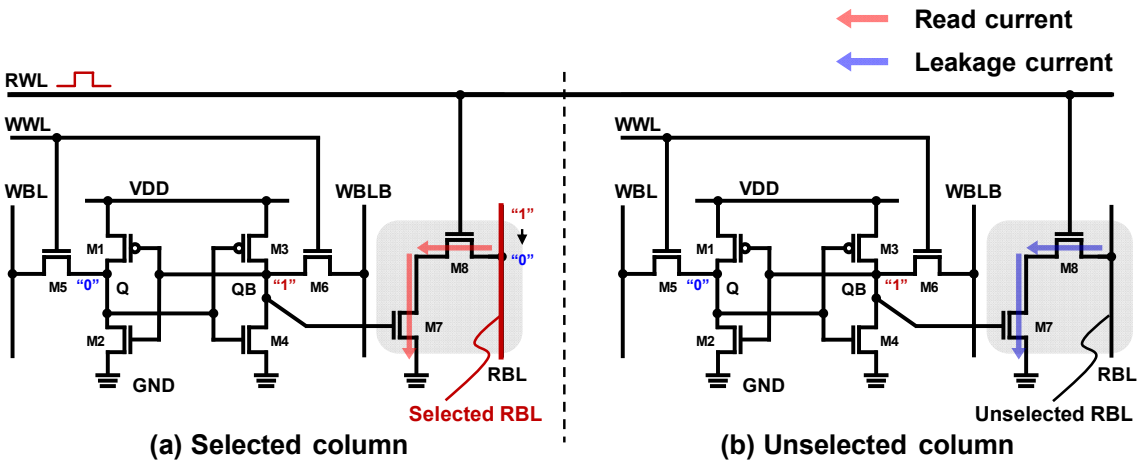


Fig. 2.4 Circuit schematic of sourceline (SL) structure, and current flow model of unnecessary read current at the unselected column in conventional 8T SRAM cells with single-ended read ports.

2.2 Fundamental Features and Issues of Deep Neural Network

2.2.1 Fundamental features of deep neural network

The neural network imitates cerebral nerve system in the animal or human brain. A hierarchical neural network has plural units in each layer and the units are connected between adjacent layers. The units of each layer are called "perceptron" and weight parameters $\{w_{1,1}, w_{1,2}, \ldots, w_{(m,n-1)}, w_{(m,n)}\}$ are assigned to the connection (synapse) of the perceptron. The calculation of neural network is performed by the multiplication of input data and synapse. When each data $\{x_1, x_2, \ldots, x_m\}$ is transferred to the input layer, the u_j is calculated by the multiplication with the weight parameters and the addition of the bias parameter b_j. Then, the u_j is transferred to the next layer, by the Equation (1).

$$u_j = \sum_{i=1}^{m} w_{i,j} x_i + b_j \quad (j = 1,2,\ldots,n) \tag{1}$$

The u_j must be multiplied by the activation function $f(u_j)$ to compute an output of target layer. The output of activation function is expressed as Eq. (2). For the activation function, a nonlinear function is generally used in neural network such as monotonically increase nonlinear functions.

$$z_j = f(u_j) \quad (j = 1,2,\ldots,n) \tag{2}$$

While the activation functions include step function, softsign, sigmoid functions, the normalized linear function as rectified linear units (ReLU) represented by equation (3), is typically used these days. In the ReLU function, when the partial derivative is defined as in Eq. (4), both the forward propagation and the back propagation should have much less computational cost than the other functions listed above.

$$f(u) = \begin{cases} u & (u > 0) \\ 0 & (u \leq 0) \end{cases} \tag{3}$$

$$\frac{\partial f}{\partial u} = \begin{cases} 1 & (u > 0) \\ 0 & (u \leq 0) \end{cases} \tag{4}$$

The forward propagation is calculated by the above procedure, and the output of each unit in the final layer is defined as y_j. Once the output of the final layer of the neural network is obtained, an error function (loss function) $E_n(w)$ is calculated. The error function $E_n(w)$ is an indicator to measure the accuracy of the output data generated by the present weight parameter. The regression problems commonly use a square error function. The number of neurons in the output layer must be matched with the number of categories of training data. Here, we assume the number of neurons in the output layer: y and the training data (categories): d is k, then, the output of loss function $E_n(w)$ is expressed as Eq. (5).

In the network training, to optimize the weight parameters which assigned at each synapses, an optimization function $E_n(w)$ is commonly used. This optimization procedure is repeatedly executed. It is known that a stochastic gradient descent algorism (SGD) is effective method to find the optimum update value of weight coefficients and to decrease the error function's value. In the SGD algorism, the gradient of the error function with respective weight coefficient is calculated.

$$E_n(\boldsymbol{w}) = \frac{1}{2}\sum_{k=1}^{K}(y_k - d_k)^2 \tag{5}$$

$$\nabla E(\boldsymbol{w}) = \frac{\partial E(\boldsymbol{w})}{\partial \boldsymbol{w}} \tag{6}$$

Hereinafter, we define the parameter meanings; l: total layer number, i: the layer number and $j = i+1$, W: weights, dW: deltas, and ε: learning rate. Then, the weight coefficient from the layer i to the layer j is expressed as Eq. 7.

$$w_{i,j}^{l} = w_{i,j}^{l} - \epsilon \Delta w_{i,j}^{l} \tag{7}$$

Here, the learning rate (LR): ε is an important parameter which control the update amount of weight coefficient. The LR is one of the hyper parameter, which value is determined by the users. In partial derivative calculation Eq. (7), the computational cost

increases sharply from the output layer to the input layer. In order to perform this calculation efficiently, the weight update amount is calculated by the back propagating gradient from the output layer to the input layer. The details of the transformed error propagation calculation are expressed below equations. At the first step, the gradient is defined as Eq. (8). Then, the deltas in the output layer is shown as Eq. (9).

$$\delta_j^{(l)} \equiv \frac{\partial E_n}{\partial u_j^{(l)}} \tag{8}$$

$$\delta_j^{(L)} = \left(y_j - d_j\right) * \ f'\left(u_j^{(L)}\right) \tag{9}$$

The gradients of the intermediate layer are calculated by below equation, Eq. (10). After that, the update amount of the weight coefficients at each layer can be obtained as shown in Eq. (11).

$$\delta_j^{(l)} = \sum_k \delta_k^{(l+1)} w_{kj}^{(l+1)} f'\left(u_j^{(l)}\right) \tag{10}$$

$$\Delta w_{i,j}^l = \frac{\partial E}{\partial w_{i,j}^l} = \delta_j^{(l)} z_i^{(l-1)} \tag{11}$$

This results means that the deltas of the (l+1)-th layer is transferred to the upstream layer, then the delta in the layer l is obtained from the weights of the (l + 1)-th layer and the input of the layer l at the forward propagation. Therefore, the update amount of each layer's weight coefficients can be calculated. Thanks to the characteristics of the propagation rule; the deltas are obtained sequentially from the output layer to the input layer to calculate update amount of the weight coefficient; the backpropagation procedure can be parallelized easily.

In the convolutional neural network (CNN), it consists of plural network layers, which uses not a fully connected layer as described so far. In the convolutional layer, calculation is performed by a filter operation like an image processing instead of the simple product-sum operation in the fully connected layer. Here, we consider the H×W size image data. The input of the convolutional layer defined as $x_{i,j}$. The N×N filter as weight coefficient is defined as $w_{s,t}$, and the parameter b is bias. Then, the output of the

$$u_{p,q} = \sum_{s=0}^{N-1} \sum_{t=0}^{N-1} w_{s,t} x_{(p+s),(q+t)} + b \tag{12}$$

convolutional layer $u_{p,q}$ is expressed as Eq. (12).

The output of the convolutional layer is usually input to the pooling layer. In the pooling layer, a compression of the pixel data to decrease the vertical and horizontal pixel sizes is executed. The convolutional neural network (CNN) is composed of many convolutional layers. Figure 2.5 shows a concept of CNN network. The network contains convolutional layers, normalization layers, activation layers, pooling layers, fully connected layers, and so on. Actually, CNNs have been scaled up with numerous synapses and neurons in deeper layers.

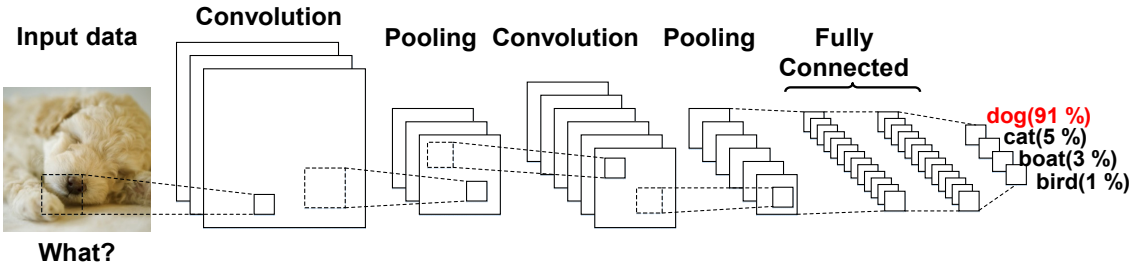


Fig. 2.5 The concept of convolutional neural network (CNN).

2.2.1 The issue of memory architecture in the deep neural network

As CNNs have generality with a deeper and larger-scale network, their error rates of cognition continue to improve. Accordingly, computational times become much longer, particularly those for training purpose. Furthermore, the recognition accuracy is improving steadily according to the network size [31], as shown in Fig. 2.6.

Recently, to accelerate the CNN computations, the mini-batch processing is popularly used. Normally, the network training requires the numerous training samples. In the ImageNet dataset that is provided from ILSVRC 2012, it contains 1.28 million images for network training purpose. Therefore, when we consider the network training with a large dataset, enormous computational cost and training time is required because of the increased number of parameter updates for each sample. In this situation, the mini-batch training has become popular as an effective way to execute the network training. The

mini-batch training updates the weight parameters with multiple samples in the single iteration. Because the variation of gradients is compensated by the multiple samples, the training convergence will be better. However, in cases of memory capacity, the input data memory or activations of each convolutional layers are lineally increase as the increased number of sample images which must be hold. In the larger batch size, it no longer fits in the internal memory on a single worker.

For further acceleration, the data parallelism is known as one of the promising way to speed up the deep learning. The data parallelism divides the dimensions of the data. The divided data is distributed to the parallel workers. The worker trains same network but with a different data samples. The mini-batch training is categorized as data parallelism, which can be applied to the multi-worker training. In those data parallelism, to increase the parallelization degree, many workers duplicate same network model and hold it in the internal memory. The unification process of deltas collects and take average the deltas that is generated by the back propagation, it must be transferred from each worker to the parameter server. The weight unification process updates weight parameter by using the unified delta. The many parallel workers need large amount of transfer bandwidth, this is why the communication delay will be longer and longer. Figure 2.7 shows the relationship between the acceleration factor and the number of workers. The data parallelism with multiple workers are effective to accelerate the deep learning tasks. However, the total memory capacity in the internal memory is drastically increased. Furthermore, the communication delay on the data bus is significantly increased at the highly distributed parallelization. The acceleration factor saturates even if over 50 GPUs are paralleled [32]. In the conventional parallelization model, the memory bandwidth and memory capacity limits a scalability of deep learning acceleration.

In this work, we targeted the development of low-cost and high-scalable deep learning processor with lower internal-memory capacity and lower memory bandwidth on the data transfer bus. Our algorism and architecture expands a scalability with lower memory cost and higher energy-efficiency than the conventional data parallelism.

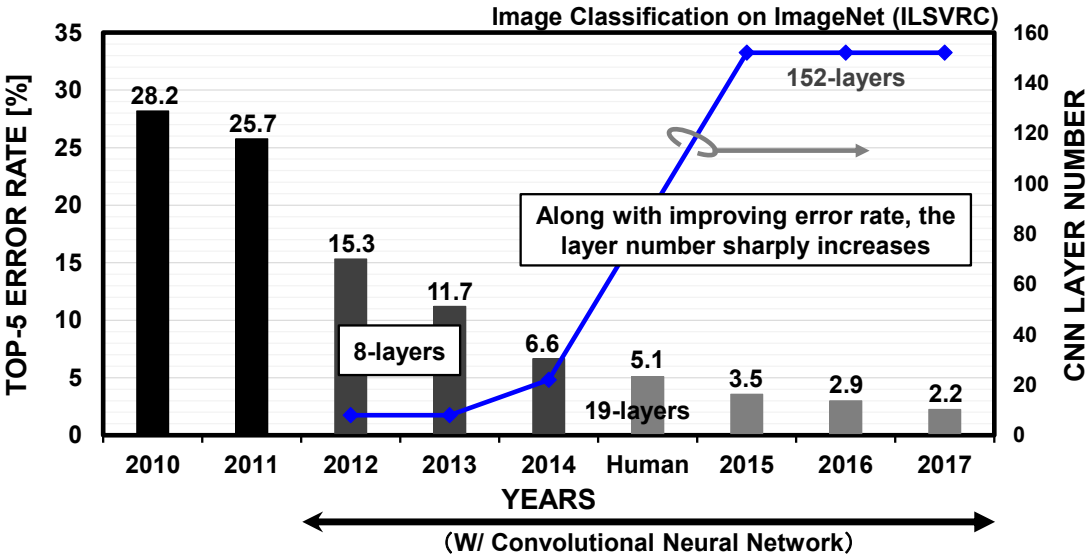


Fig. 2.6 The correlation between the accuracy and the deepnesss of network model [31].

1. **Accelerate training time is the key for future DNN application.**
2. **External data communication between GPUs and servers are required for weight unification.**
3. **Speed up factor is saturated with large-number of GPUs, that is the scalability limitation in distributed deep learning.**

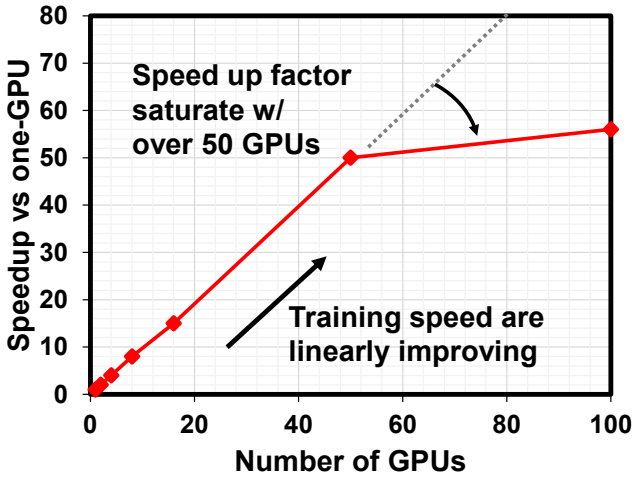


Fig. 2.7 The relationship between the acceleration factor and the .number of workers. The acceleration factor saturates over 50 GPUs situation [32].

2.3 Summary

This section summarizes the issues of multi-port SRAMs as an image memory, and the memory system in the distributed image recognition processors.

For the multi-port SRAM, these issues are explained

- Increased active energy on the access ports, and its area cost
- Energy efficiency degradation by the unnecessary current flow and leakage current.

For the deep learning processor, below issues and challenges are explained.

- How to accelerate the computational time to train network models.
- Increased amount of internal memory capacity
- Increased memory bandwidth with parallelized workers

These issues have to be considered on whole memory architecture level to achieve low-power and low-energy image recognition processor. The cooperative design with device technology in SRAM, memory efficient algorithm design, and hardware development have to be integrated. In this dissertation, the novel techniques are presented in Chapters 3 to 5 to address the issues.

Chapter 3 **Low-energy Multi-Port SRAM Cell Design**

All the static random access memory (SRAM) in this chapter were implemented in the 28-nm FD-SOI process technology supported by ST-Microelectronics Co. Ltd. This chapter describes low-power 2-read/1-write 8T three port SRAM design. We studied following contents:

1) The design of novel 2-read/1-write three-port 8T SRAM bitcell in 28-nm FD-SOI with small area overhead.
2) The combination with the proposed SRAM and majority logic circuit for further energy reduction.

Finally, the energy efficiency and the performance improvement in the actual H.264 motion estimation image processor is explained when the proposed SRAM techniques are adopted.

3.1.1 The multi-port SRAM design in image processor

When considering the mechanism of image or video sequence, image processor must perform the matrix-matrix computation for filtering operation. Normally, the high-speed SRAM as an internal memory is indispensable for effective data loading, because the external data access conspicuously lead longer access delay in memory. Actually, high-performance SRAM is used as a frame buffer and a reconstructed image memory in a real-time video or image processing in the H.264 image processor, shown in Fig. 3.1. In particular, parallel computation by multi-thread processing of matrix-matrix operations as seen in the image processing is beneficial for its real-time operation.

From this reason, input data for image processing are stored temporarily in SRAM. In an image processor, many processing cores access the SRAM for multi-thread processing, as presented in Fig. 1. Demands for multi-port SRAM have been increased to accommodate high-speed and low-power image processing. The multi-port SRAM is suitable for parallel operation. It improves the total chip performance. To date, a multiport SRAM that supports simultaneous write and read operations is proposed for use as the image processor [5], [6]. The three-port SRAM is reportedly suitable for use as an image processor [7], [33]. When comparing features of two images, simultaneous

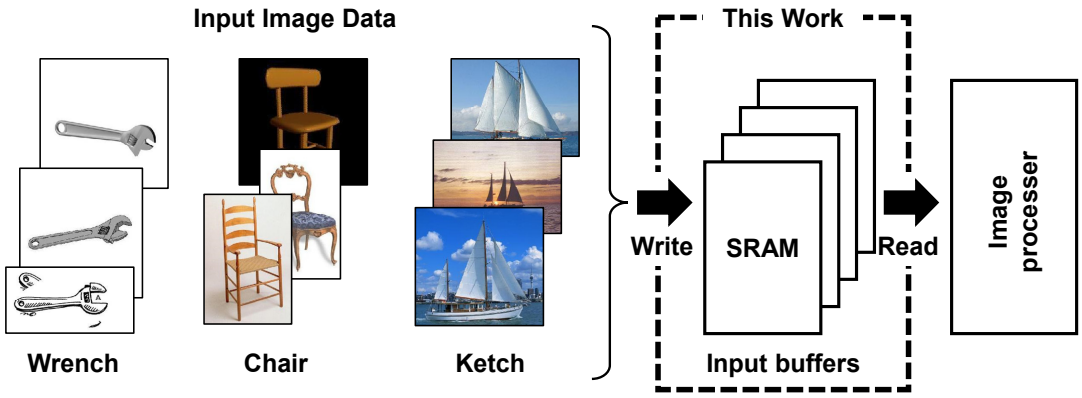


Fig. 3.1 Memory system in image processing. This work targeted the development of low-power and low-cost image memory.

read operations are requested to SRAM cells. Furthermore, realizing real-time processing requires a write operation for the next comparison at the same time as the read operation. Therefore, two read operations and one write operation must be performed simultaneously, which require multiport SRAMs that have two-read/one-write access ports for the image processor.

In the multi-port SRAM, the larger area overhead is critical due to the increasing number of transistors in the SRAM cell. In the image processor, the larger number of SRAM capacity is required to process high-resolution image and video sequence. The resolution improvement brings higher energy consumption in SRAM.

Conventionally, the bit-cell layout in the three-port SRAM needs a larger bit-cell area than that of an 8T dual-port SRAM cell due to the larger number of transistors which must be accommodated in the SRAM cell [33]. In particular, an image processor requires larger multiport memory capacity, which gives a serious impact on its cost. In this paper, we exhibit an 8T three-port SRAM smaller than the conventional three-port bitcell. Its area is as small as the conventional 8T dual-port SRAM.

We propose two types of circuit techniques in the multi-port SRAM to improve area efficiency and to achieve low-voltage and low-power operation. We designed a 28-nm FD-SOI 8T three-port SRAM for a low-power image processor and compared it to a 28-nm FD-SOI 6T SRAM in the conventional form.

3.1.2 1-Write/2-Read 8T three-port SRAM cell design

Multi-port SRAM with plural read ports improves functionality to handle the simultaneous accesses. Since the write and read ports are controlled independently, write and multiple read operations can be performed simultaneously on different cells in the same bank. In this way, the multi-port SRAM is suitable for parallel operation such as image processing. The conventional 1-write/2-read three-port SRAM needs larger bitcell area than the 8T dual-port SRAM due to the increasing number of transistors in the SRAM cell [33]. In particular, an image processor requires a larger multi-port SRAM capacity, which gives a serious impact on its cost.

A circuit schematic of the proposed 8T three-port SRAM is presented in Fig. 3.2. It has a pair of write bitlines and two single-ended read bitlines (one-write/two-read bitcell structure). The proposed SRAM has two pull-up pMOSs (load-pMOS), two pull-down nMOSs (drive-nMOS), and four transfer nMOSs (access-nMOS). In this circuit, M7 and M8 transistors are the two single-ended read ports. Source nodes of M7 and M8 transistors are connected to node QB. The drain nodes are connected to read bitlines (RBL_A, RBL_B). The gate nodes of M7 and M8 are connected to the read wordlines (RWL_A, RWL_B). Both read ports (read ports A and B) are consolidated on one side

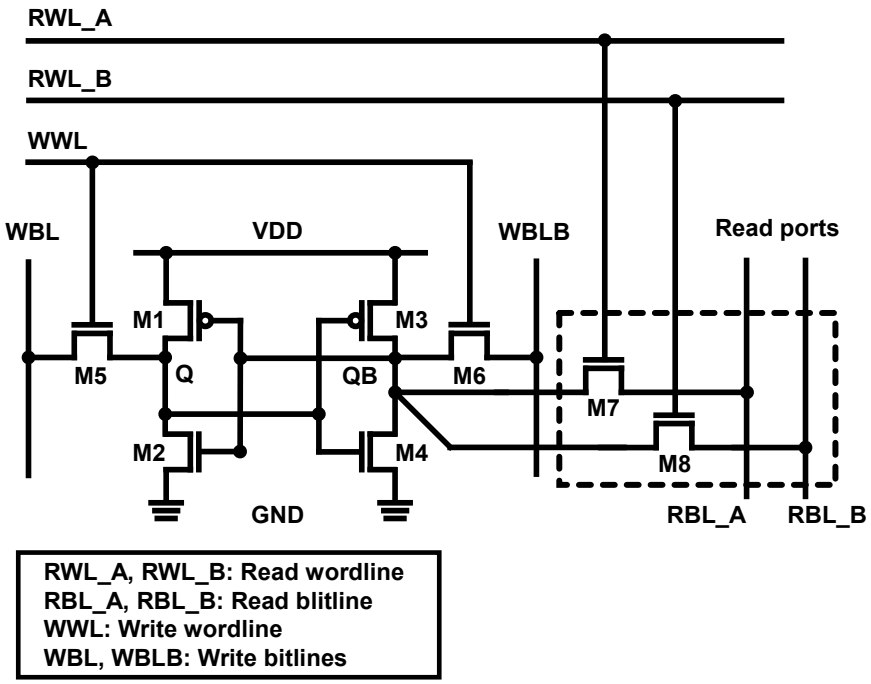


Fig. 3.2 Schematic of proposed 8T three-port SRAM with single ended read ports.

Table3.1 Transistor W/L sizes in the proposed SRAM cell. The proposed 8T SRAM cell is designed on the logic rule bases.

Tr. size	Pull-up Tr. (M1, M3)	Pull-down Tr. (M2, M4)	Write pass gate Tr. (M5, M6)	Read pass gate Tr. (M7, M8)
Width [nm]	80	142	80	80
Length [nm]	30	30	30	30

Logic Rule based design

and has asymmetric cell structure. This asymmetrical 8T SRAM cell achieves higher density than the conventional three-port SRAM cell.

The β ratio in the SRAM cell indicates a strength of the pull-down transistor against to the pass-gate transistor. The β ratio must be remained sufficiently to ensure the proper read operation. In our design, the W/L size of the pull-down transistor in the bitcell is chosen to remain a sufficient SNM (static noise margin) even when the both read ports are activated. All transistor W/L sizes in the bitcell are shown in Table 3.1. In this case, the β ratio in the proposed three port SRAM cell is 1.77 (= 142 nm/80 nm).

Figure 3.3(a) presents front-end-of-line (FEOL) of the proposed 8T three-port SRAM. Read ports comprising M7 and M8 transistors are arranged separately from a 6T SRAM cell, which share a common contact located at the middle as the QB node. This layout achieves a smaller cell area than in symmetrical layout in which the additional read ports are arranged at both ends [29]. However, a vertical distance between upper and lower gates (gate pitch) is increased because the M-3 metal for additional read wordlines (RWLs) are inserted.

Figure 3.3(b) shows the back-end-of-line (BEOL) of the proposed SRAM. The SRAM cell size is determined by the number of horizontal and vertical wires. In our proposed SRAM, two read ports consisting of M7 and M8 transistors are configured as two single-ended read ports having three bitlines and three wordlines shown as Metal 2 and Metal 3. The cell area is 0.56 μm^2 on a logic rule base, which is as small as the dual-port 8T bitcell [34], although the number of read ports is increased. The small-area bitcell contributes to reduce parasitic capacitance on the WLs and BLs in SRAM matrix. This feature of proposed SRAM is beneficial for low-energy operation by saving the leakage and the operating energy.

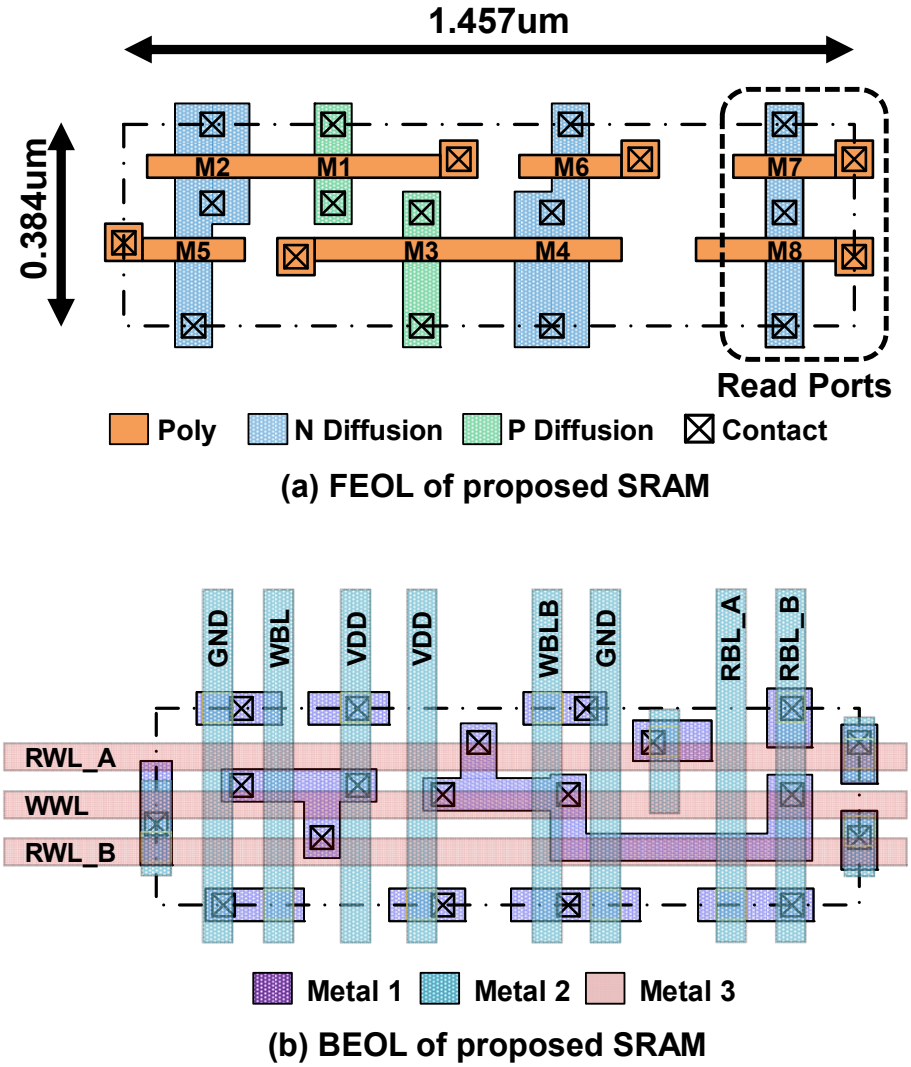


Fig. 3.3 Bitcell layout of proposed three-port SRAM: (a) FEOL and (b) BEOL.

The operating waveforms in the read operation are depicted in Fig. 3.4(a)-(c). Initially, the wordline pulse is inputted to the selected row, as shown in Fig. 3.4(a). If the QB node holds the “0” data, the read current flows through the pass-gate transistor M7 or M8 to the QB node, as shown in the case of Fig. 3.4(b). On the other hand, no read current flows through the read bitlines (RBL_A and RBL_B) when the internal node, node QB, is “1”, as shown in Fig. 3.4(c). Figures 3.5(a) and (b) show the behavior of the current flow at the read operation in the proposed SRAM. Fig. 3.5(a) shows a read current flow when the QB node stores “0” data and RWLs are activated. In this situation, read current is pulled from RBL_A, RBL_B to the QB node. The RBLs are precharged right before the read cycle. The charge/discharge energy on the RBLs are consumed every read cycle. On the other hand, because “1” data is stored in the QB node, the source node of M7 and M8 is equally driven to the VDD. Therefore, the read current flow from the RBLs to the QB node does not consumed, as depicted in Fig. 3.5(b). From this reason, maximizing the number of “1”s at node QB is important to reduce dynamic energy in the read operation.

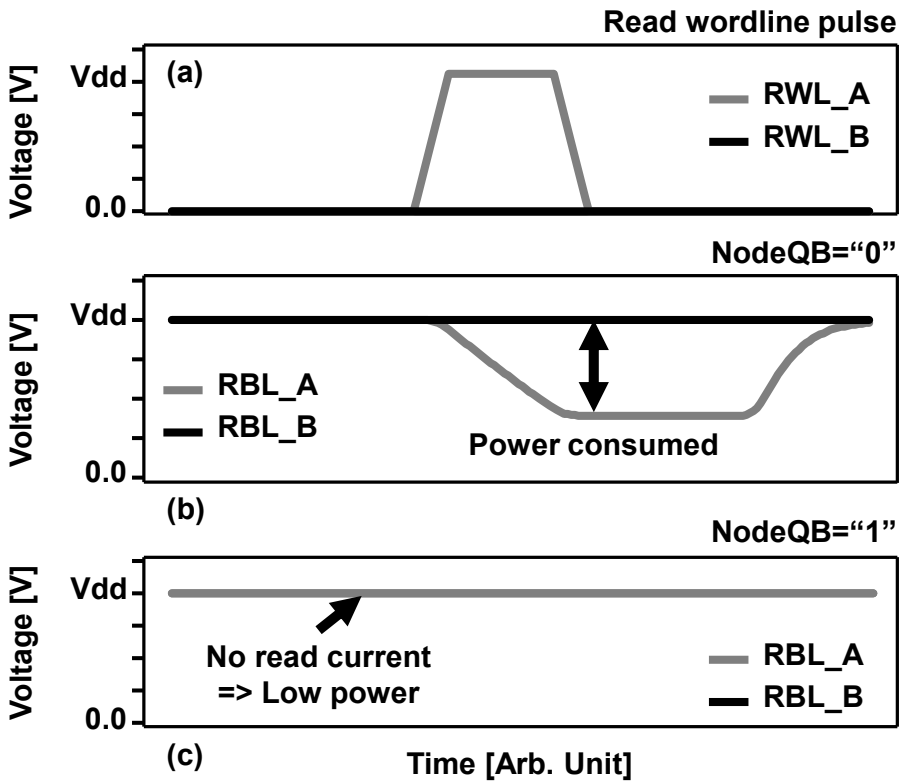
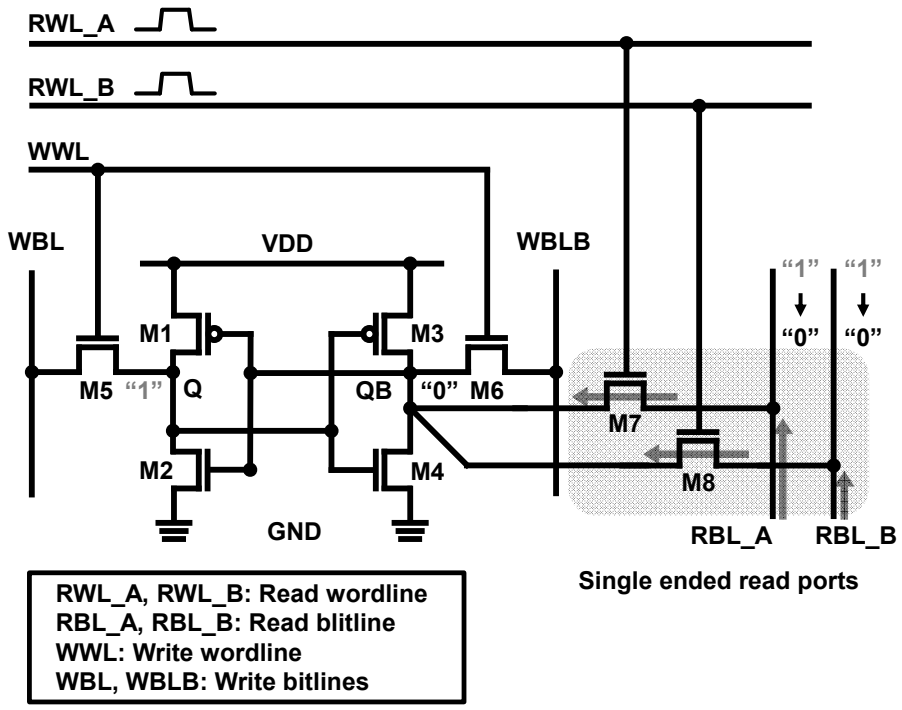
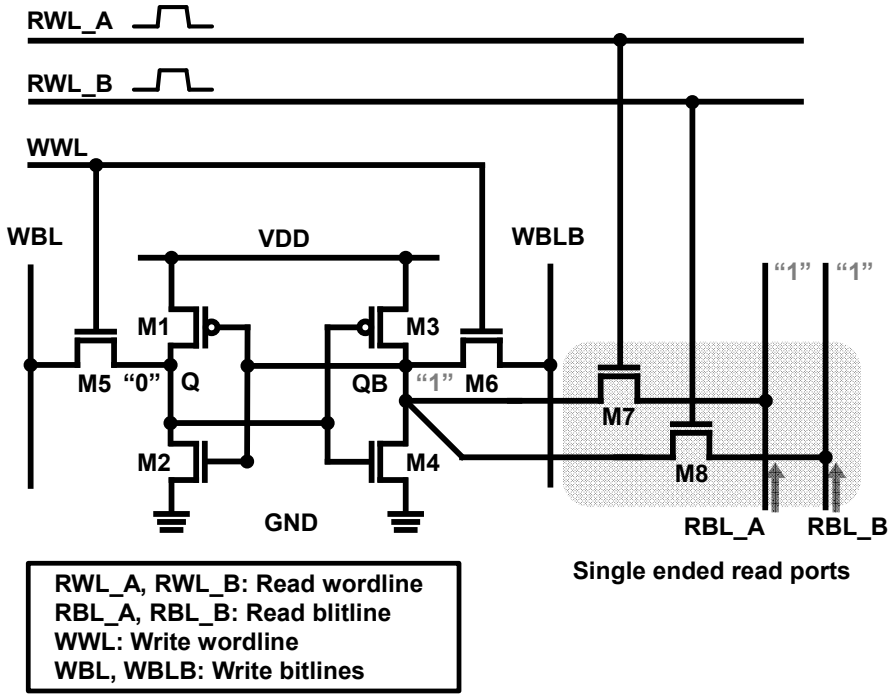


Fig. 3.4 Waveforms of proposed 8T three-port SRAM in read operation; (a) Wordline pulse, (b) when the Node QB holds “0” data, (c) when Node QB holds “1” data.



(a)



(b)

Fig. 3.5 Schematic of the proposed 8T three-port SRAM and write/read current flow model, (a) when the Node QB holds "0" data, (b) when Node QB holds "1" data.

3.1.3 Precharge-less write circuit

Figures 3.6(a)-(b) presents comparison of write circuit between the conventional 6T SRAM and the proposed 8T three-port SRAM. Figure 3.6(a) depicts the conventional write circuit with write bitline precharge scheme which as shown with pMOS transistors to charge a bitline pair. The conventional write circuit must have a precharge scheme to maintain stability of read operations because both read and write operations use the common bitline pair. Figure 3.6(b) depicts the precharge-less write circuit. Successive writes of the same data consume less energy because the proposed 8T SRAM does not need a precharge scheme on the write bitlines because of the dedicated read ports for the read operation. However, it incurs the well-known half-select problem along the write wordline. In our design, the divided wordline structure is therefore adopted to avoid the half-select problem [35]. However, the divided word line structure entails the large area overhead in SRAM macro, therefore only write wordlines have a divided architecture.

Figures 3.7(a)-(c) portrays simplified waveforms during four ‘0’ write cycles and an “1” write cycle. Figure 3.7(a) shows the waveform of the write wordline (WWL) pulse commonly used in the conventional SRAM and the proposed SRAM. Figure 3.7(b) shows waveforms of the write bitlines (WBL and WBLB) in the conventional write scheme. The either bitline is dropped to the ground voltage in write cycle. Therefore,

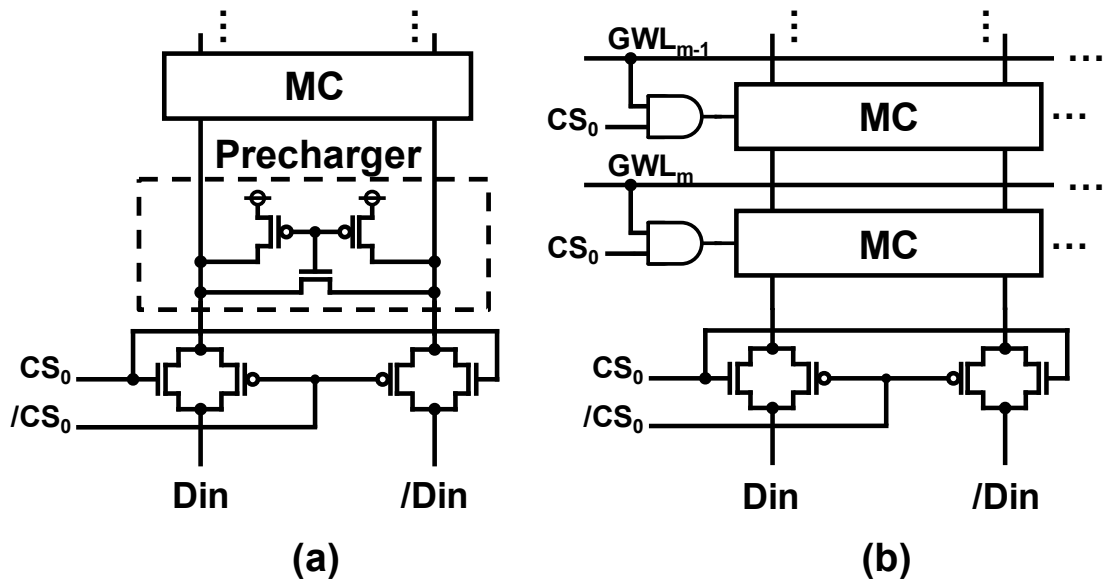


Fig. 3.6 Schematics of write circuits between conventional SRAM and proposed SRAM architecture: (a) conventional circuit and (b) precharge-less circuit.

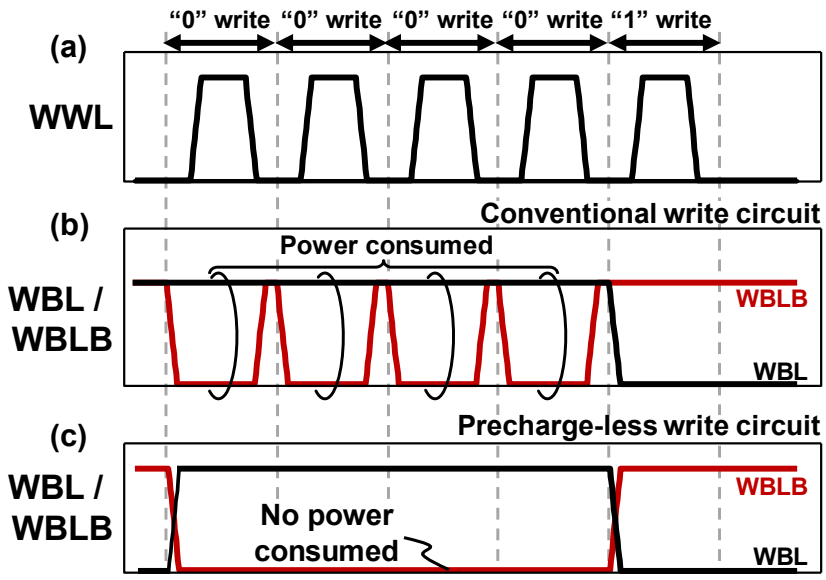


Fig. 3.7 Waveforms in the write operation: (a) Write wordline (WWL) pulse, (b) Write bitline (WBL and WBLB) signals in a conventional write circuit and (c) a precharge-less write circuit.

the charge/discharge power on the WBL is consumed in every cycle by the precharge scheme. Figure 3.7(c) portrays waveforms of the write bitlines in the proposed SRAM. By virtue of the precharge-less write scheme, which reduces the write energy, the charge/discharge power on WBLs is consumed only when a write datum is changed.

3.1.4 Static noise margin (SNM) in 8T 1W2R three-port SRAM

A multi-port SRAM supports simultaneous accesses from plural cores through the multiple read and write ports. Particularly in a one-write two-read (1W2R) three-port SRAM cell, the two read ports are both available for simultaneous read outs, which implies that simultaneous read outs occur on the single SRAM cell [36]. Figure 3.8 shows a variety of read situations in the 1W2R three-port SRAM cell when both read ports are enabled simultaneously. Figure 3.8(a) depicts two SRAM cells on different row addresses and different column addresses, designated independently. No issues emerge relative to the access conflict. However, the simultaneous dual-port read outs to a single SRAM cell activates both RWL_A and RBL_B, as presented in Fig. 3.8(b), which might worsen the static noise margin (SNM) because of double read currents.

Figures 3.9(a)-(b) presents simulation results of the SNM in the proposed 1W2R 8T three-port SRAM cell at several supply voltages of Vdd = 0.4–1.0 V. Figure 3.9(a)

depicts the standard butterfly curves in the single port read situation: the SNM of 171 mV are achieved at 1.0 V, leaving 85% of the SNM in the conventional 6T SRAM [11]. Figure 8(b) depicts the worst-case butterfly curves in the simultaneous dual-port reads. The SNM is reduced to 101 mV at 1.0 V. An interesting point is that the maximum SNM of 102 mV is observed at 0.8 V.

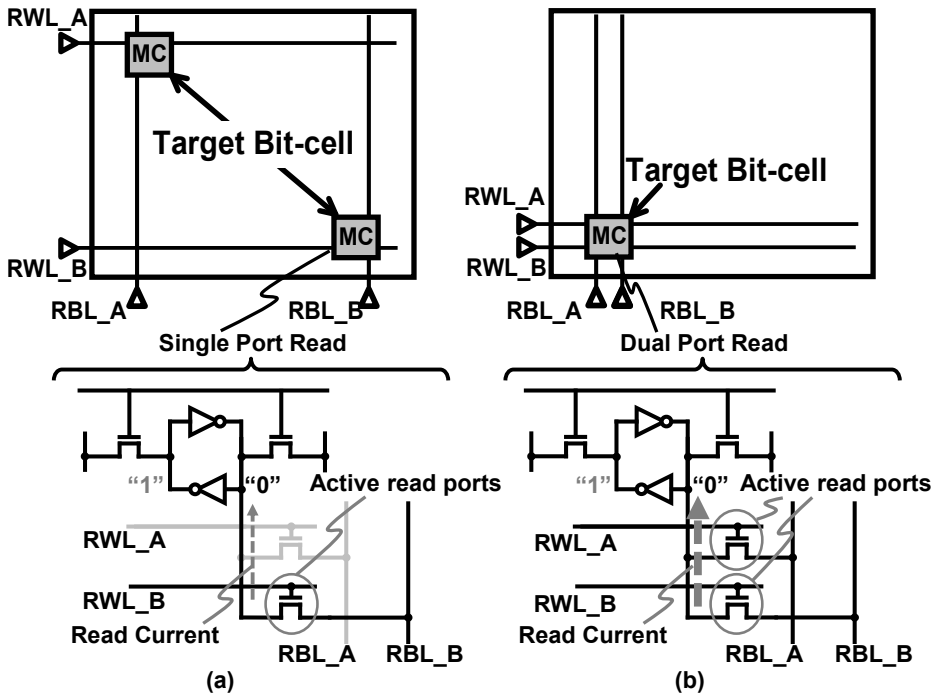


Fig. 3.8 Variety of access situations in the proposed 1-W/2-R three-port SRAM.

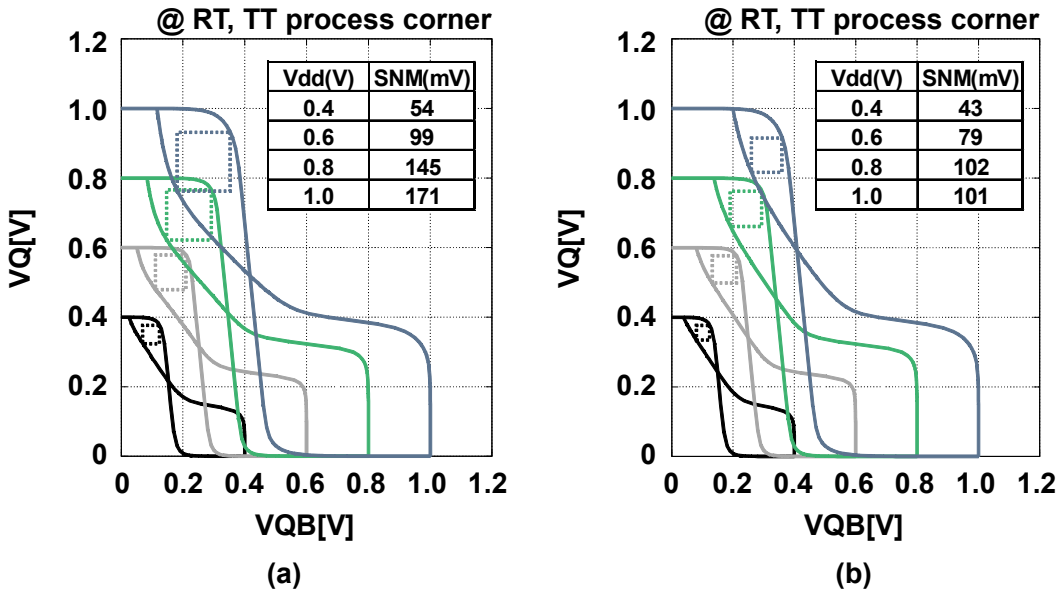


Fig. 3.9 Simulated butterfly curves at several Vdd from 1.0 V down to 0.4 V: (a) single-port read out and (b) dual-port read out.

3.1.5 Combination with majority logic

Our earlier study demonstrated that the majority logic circuit can conserve charge/discharge power on the read bitlines [37]. Fig. 3.10(a) shows a process flow of the proposed SRAM with the majority logic circuit. Image data reflect luminance information: bright pixels have many “1” data and dark pixels have many “0” data. For read energy reduction, having many “0”s should be inverted by the majority logic. To maximize the number of “1”s, the majority-logic circuit counts “1”s and decides if input data should be inverted in a write cycle, so that “1”s are in the majority. The information of inversion (“1” denotes inversion) is stored in an additional flag bit, as depicted in Fig. 3.10(b). In a read cycle, this procedure is reversed in order. Output data are inverted if a flag bit is true, so that the original data can be read. Note that the majority logic does not reduce write energy because the “1” write energy and the “0” write energy are the same in the SRAM.

The mechanism on the RBL power reduction is depicted in Fig. 3.11. We assume the bit width of the input data is eight. If the number of “1”s in the input data is five and over, the data is not inverted, and one “0” is stored as additional flag bit. This means one read current is made by RBL, in witch is a power overhead. If the number of

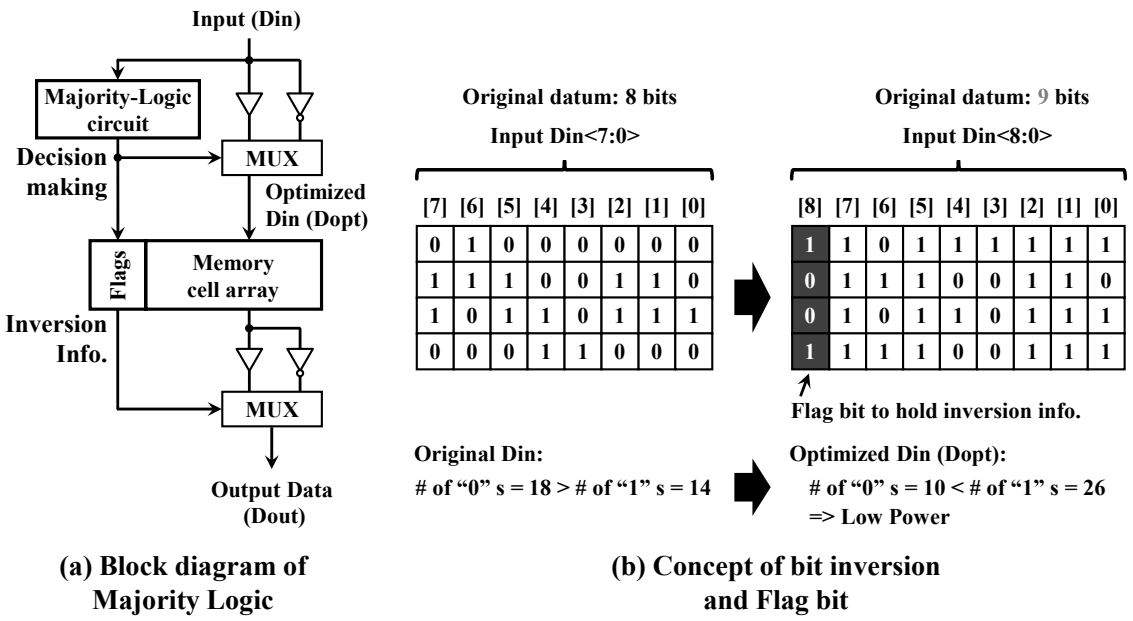


Fig. 3.10 Concept of SRAM with majority logic.(a) Block diagram, and (b) flag bit.

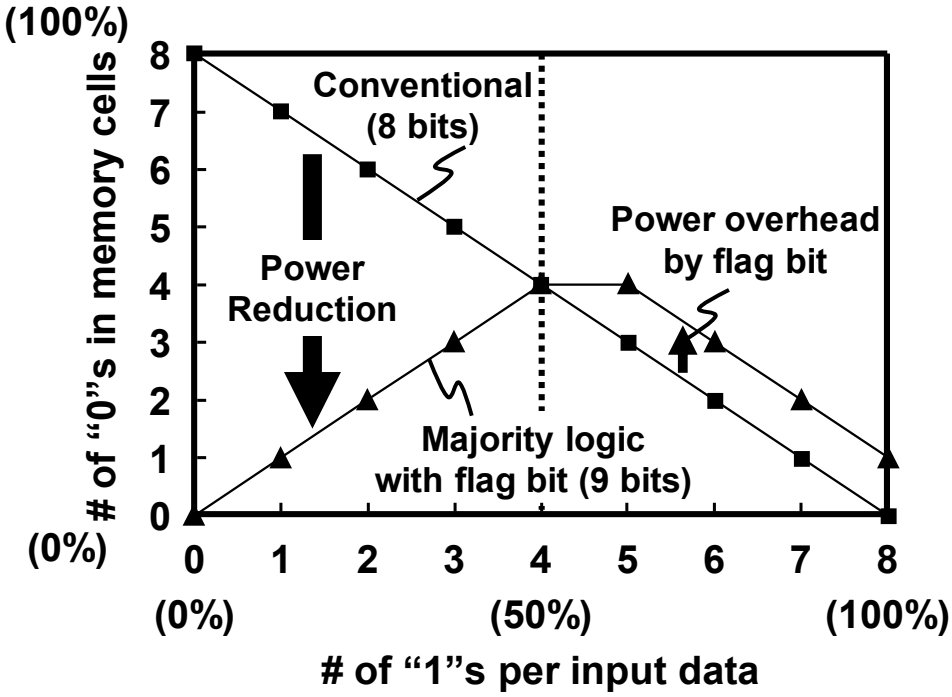


Fig. 3.11 Power reduction ratio at each digit in majority logic.

"1"s in the input data is four or less, the data are inverted by the majority logic to maximize the number of "1"s, and reduces the read power. When input data have a random pattern, the number of charges/discharges is four out of eight RBLs. 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 are random. In our proposed SRAM, majority logic conserves charge/discharge power effectively on the read bitlines because the number of "1"s in the input data is maximized.

3.1.6 Chip implementation and measurement results

We fabricated a 64-kb 8T three-port SRAM macro using 28-nm FD-SOI process technology. Figure 3.12 shows a test chip micrograph. The proposed 64-kb macro consists of 2 × 32 kb sub-blocks. The macro area is 0.058 mm^2. Figure 3.13 presents a measured read Shmoo plot of the proposed SRAM macro. We verified that it can operate with supply voltage of 0.46 V and access time of 140 ns. At room temperature (=25 celsius degree), the operating point that achieves the minimum energy per cycle is a supply voltage of 0.54 V and a cycle time of 55 ns (= 18.2 MHz).

Figure 3.14 shows the Shmoo plot in write operations. The test chip can operate at write pulse width of 4 ns. Figure 3.15 portrays a schematic of the proposed 8T three-port SRAM array and its peripheral circuits. Figure 3.16 shows the measured leakage and active energies. In the write operation, the test pattern of the “ALL0” write pattern means successive “0” writes to all bitcells in the memory macro. “ALL1” means successive “1” writes. In those cases, bitcell data do not change, and the bitline charge/discharge energies are saved. The “01-pat.” write pattern signifies the alternately writing “0”s and “1”s to the bitcells. Then the charge/discharge power occurs on the WBLs. This is the worst case in the write operation. The worst-case write energy is 484 fJ/cycle, which is 69% smaller than that in the 6T SRAM (see Fig. 3.16).

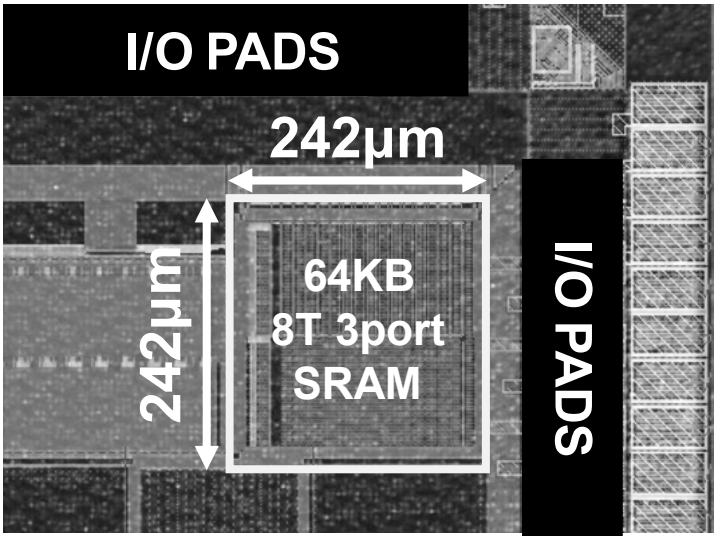


Fig. 3.12 Test chip micro photograph of proposed 8T three-port SRAM.

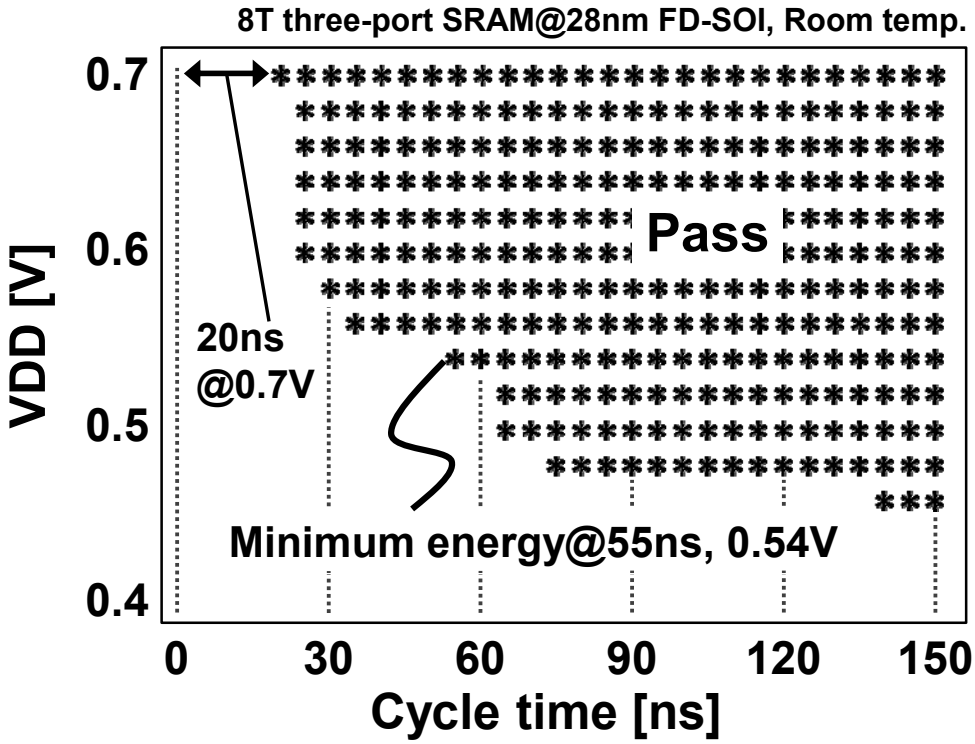


Fig. 3.13 Measured read Shmoo plot.

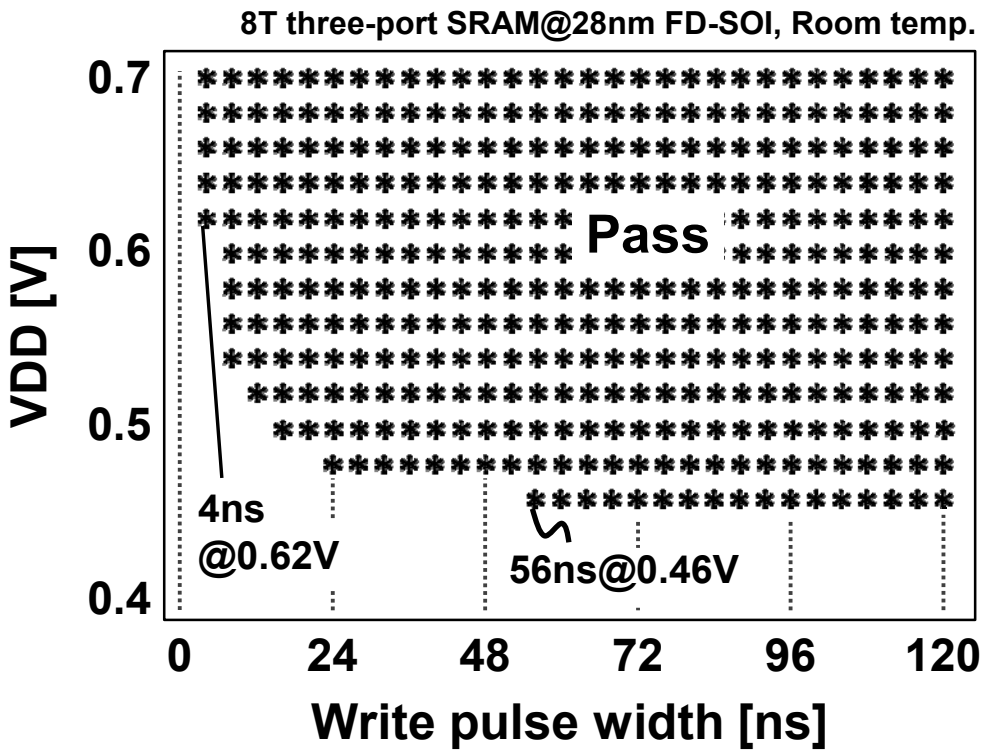


Fig. 3.14 Measured write Shmoo plot.

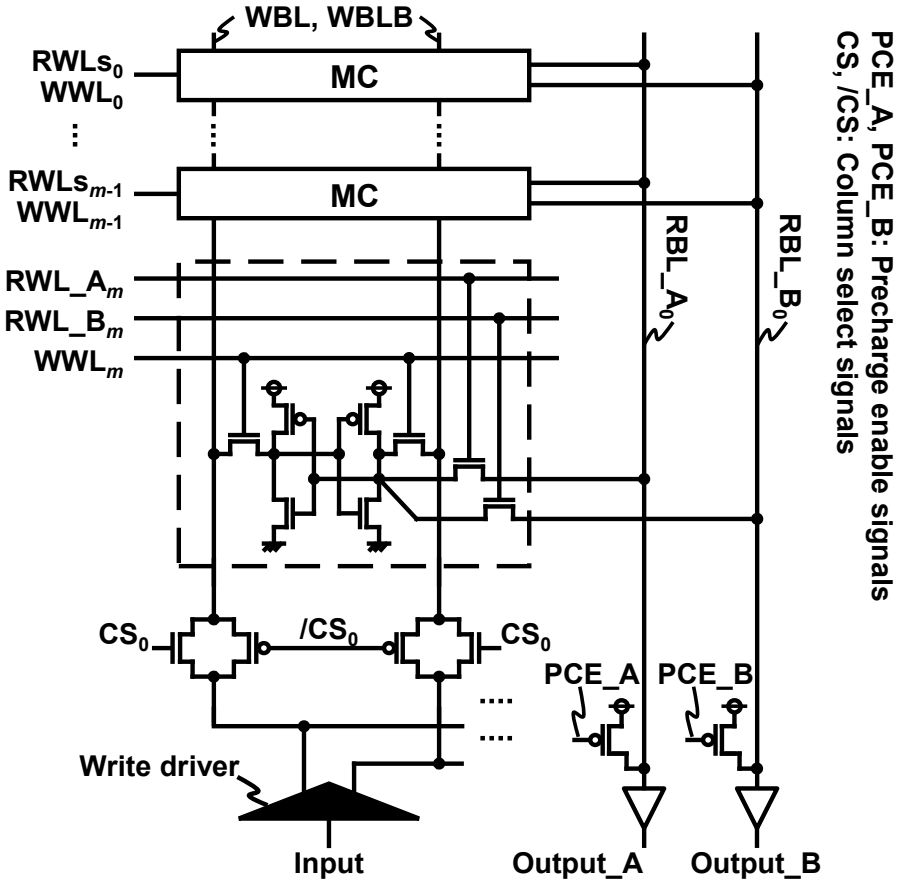


Fig. 3.15 Schematic of proposed 8T three-port SRAM array and its peripheral circuits.

The BL lengths of the proposed three-port SRAM are 1.3 times longer than the conventional 6T SRAM because of the three WLs (1 WWL / 2 RWLs) drawn through the 8T bitcell. However, the proposed 8T three-port SRAM does not require the WBL precharge scheme in the 6T SRAM. Furthermore, its WLs are divided by every 16 rows. Therefore, the proposed SRAM can reduce needless energy in the half-selected bitcells; As a result, the write energy of the proposed SRAM turns out lower than that of the conventional 6T SRAM.

It is noteworthy that the read circuit must have the RBL precharge scheme because of the single-ended read ports. In the read operation, the test patterns of the “ALL0” and “ALL 1” mean successive “0” and “1” read operations, respectively. The “01-pat.” read pattern results in the average dynamic energy of “ALL0” and “ALL1”. The respective “0” and “1” read energies are 1663.2 fJ/cycle (a read dynamic energy of 1449 fJ/cycle + a read leakage energy of 213.2 fJ/cycle) and 361.7 fJ/cycle (a read dynamic energy of

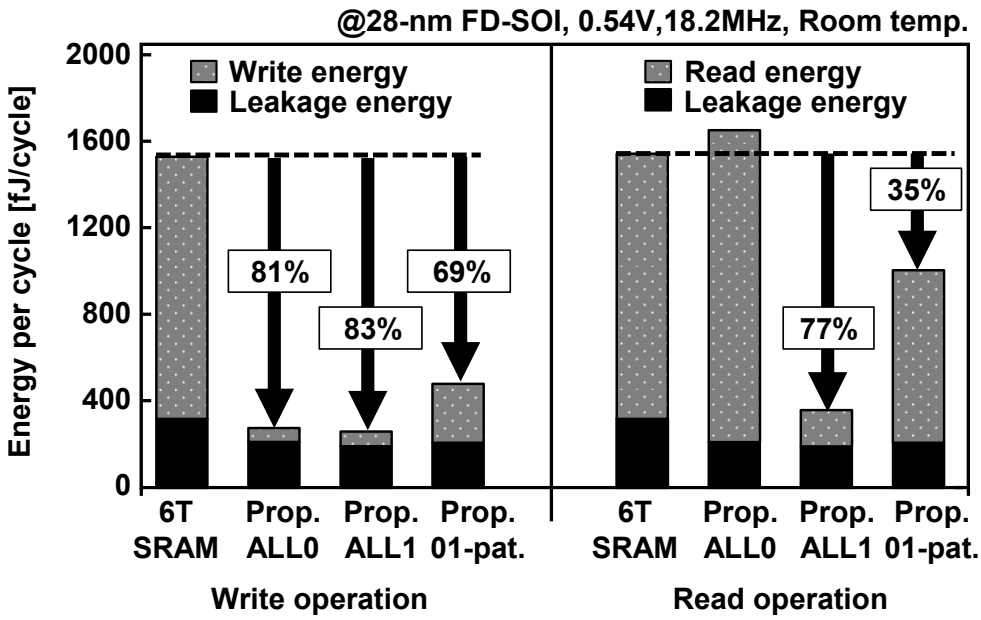


Fig. 3.16 Measured write energies, read energies, and comparisons with conventional 6TSRAM.

Table 3.2 Overview of the configurations of the implemented test chip.

Technology	28-nm FD-SOI
Supply voltage	0.46-0.7V (Memory macro)
	1.8V (I/O)
Chip area	1.0x1.0mm^2
Macro size	242x242μm^2
Macro configulation	64Kb (32Kb X 2), 16bits/word
Cell size	0.384x1.457μm^2
Frequency	7.14MHz@0.46V, 50MHz@0.7V
Write active energy	298fJ@0.54V, 18.2MHz, RT
Read active energy with majority logic	650fJ@0.54V, 18.2MHz, RT

168.5 fJ/cycle + a read leakage energy of 193.2 fJ/cycle). Consequently, the energy saving in the “1” read operation is 77%. The read energy improvement is, however, merely 35%, on average with no majority logic.

Figure 3.17 portrays the impact of the majority logic on the read energy saving. In bright Image 1, the read energy was reduced by 23%, whereas, in the dark Image 6, it reaches a 47% saving. As one might expect, the dark image is more appropriate and effective for the majority logic. In this case, the read energy is 650 fJ/cycle. Table 3.3 presents test SRAM characteristics.

Figure 3.18 shows the estimated power consumption when the proposed 8T three-port SRAM with the majority logic is applied to our prior work, ME264 motion estimation processor [22]; the values are scaled by the process node, supply voltage and operating frequency (28-nm process node, 0.54 V supply voltage and 50-MHz operation frequency). The ME264 processor has SIMD systolic-array architecture, and a 10T three-port SRAM is used as a search window and a template block. The energy consumed on the proposed SRAM is saved by 290 μW, which signifies 24% of energy.

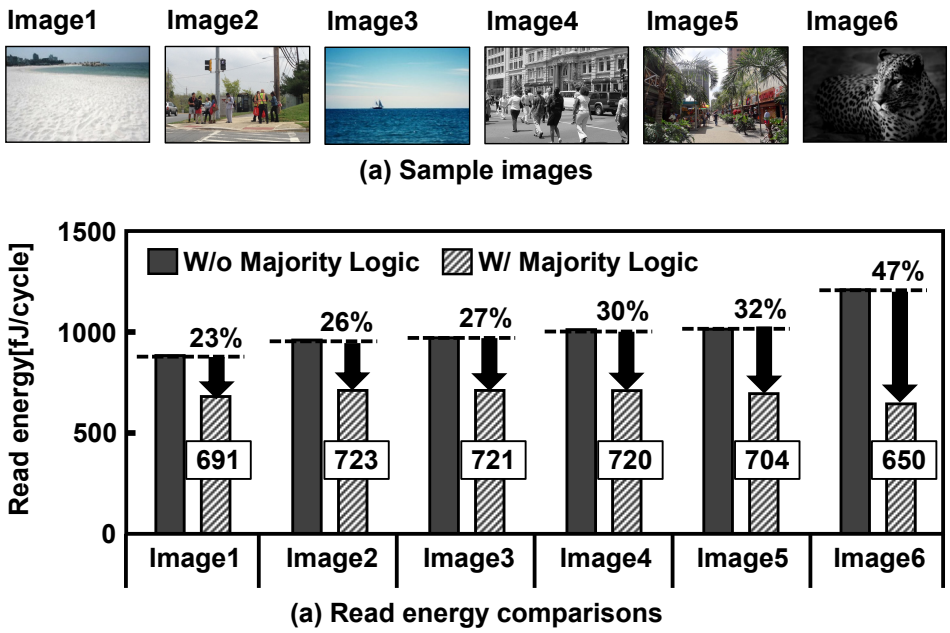


Fig. 3.17 Read energies saved by majority logic in actual image data.

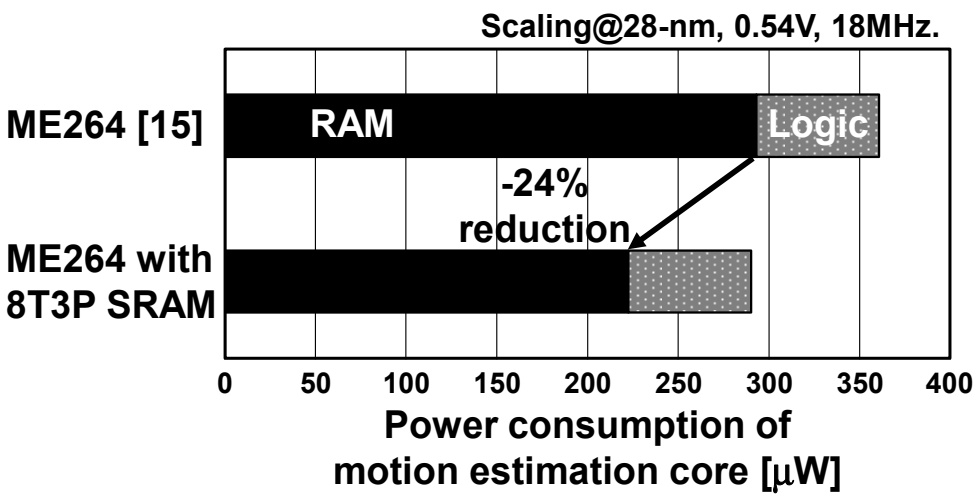


Fig. 3.18 Estimated power consumption of motion estimation image processor.

3.2 Summary

In this chapter, we presented 1-write/2-read 8T SRAM design in 28-nm FD-SOI: the asymmetric eight transistor (8T) SRAM cell for small cell area and combination with majority logic for low energy operation.

1) As described in this chapter, we presented an 8T three-port SRAM for an image processor. The proposed SRAM comprises one-write/two-read ports and a majority logic circuit to save active energy. We fabricated a 64-kb 8T three-port SRAM using 28-nm FD-SOI process technology. The test chip exhibits 0.46 V operation and access time of 140 ns. The energy minimum point is a supply voltage of 0.54 V at a frequency of 18.2 MHz, at which 484 fJ/cycle in a write operation and 650 fJ/cycle in a read operation are achieved, assisted by the majority logic. These factors are 69% and 47% smaller than those in a 28-nm FD-SOI 6T SRAM.

Chapter 4 BL Swing and Leakage Reduction for 8T Dual-Port SRAM

In this chapter, we are focus on the 1W/1R dual-port SRAM. This type of SRAM bitcell is generally used to enhance multi-thread operation. Furthermore, the single ended read port structure achieves better area efficiency. Therefore, this type of dual-port SRAM is suitable for the image processor which has plural cores and large memory capacity. As described in chapter 2, the second issue of multi-port SRAM is energy efficiency degradation effected by the unnecessary current flow and leakage. Especially, the SRAM bitcell with single-ended read port is significantly effected. In this chapter, we designed the 8T dual-port SRAM with improved energy efficiency by the novel circuitry. In addition, we implemented the proposed SRAM in the 28-nm FD-SOI process. For this purpose, we studied and proposed following contents:

1) To cutoff the unnecessary RBL charge/discharge, the selective sourceline drive (SSD) scheme as a novel footer circuitry is proposed.
2) To reduce the RBL swing, a novel MSB-based inversion logic for image processing is proposed.

4.1 Proposed Dual-Port SRAM Design

4.1.1 Overview of dual-port SRAM structure

Figure 4.1 portrays the memory system in an image processing unit. The SRAM array stores sequential data such as images, feature maps, and specific parameters for its

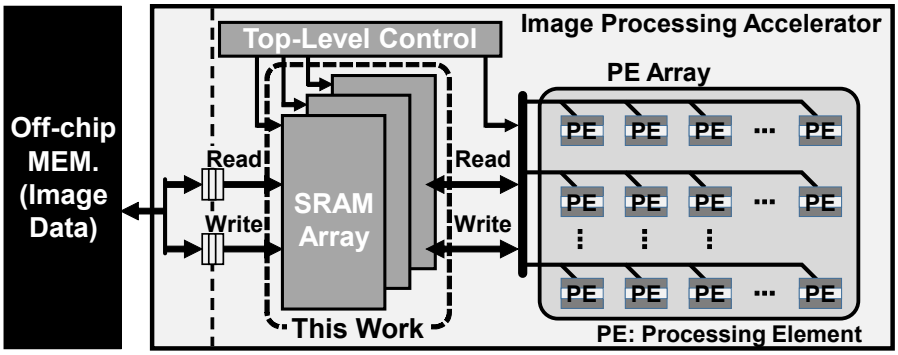


Fig. 4.1 Read energies saved by majority logic in actual image data.

processing on the many processing cores. Demand for multi-port SRAMs has increased to accommodate high-speed, low-energy image processing. The total chip performance and/or memory bandwidth can be improved with dual port bitcell by enabling multiple simultaneous operations [3]. Parallel processing is a key technology for real-time applications that require embedded memories with plural access ports [4]–[6]. To date, multiport SRAMs that support simultaneous write and read operations have been studied for use in the image processors [8]–[10].

Several important earlier works have examined dual-port SRAM architectures. In an earlier report of the literature [38], the authors proposed 40-nm 512-kb pipelined 8T SRAM for a high-speed image processor. The pipeline design enables high-frequency yet low-power operation. In another report [39], researchers explained dual-port 8T SRAM with a differential reference-based sense amplifier (SA). The motivation of their work is to obtain the benefits of small signal sensing in the context of a single-ended read path; then the work addressed the half-select problem. This SRAM design achieved lower operating voltage of 360 mV using the differential reference-based SA, but it required dummy circuits for the WL timing optimization. An SRAM design in a 28-nm fin field-effect transistor (Fin-FET) technology, which adopted a differential sense amplifier for one-write/one-read (1W1R) dual-port SRAM bitcells, was presented in another report [40]. The differential SA scheme divides a memory array into two (upper and lower) memory matrices (MATs). The differential voltage between a read bitline (RBL) of the selected MAT and that of an unselected MAT is amplified in read out operation at higher frequency. Reportedly [41], a 1W1R dual-port SRAM in a 16-nm Fin-FET technology was achieved by 6T single-ported SRAM bitcell with the double-pumping internal clock for high-speed and high-density. The double-pumping clock scheme virtually expands the bitcell function from single-port to dual-port by generating the double-clock in a single cycle. The double-pumping clock scheme for an internal clock generator achieves robust timing design without strict severe setup and hold margins, and achieves lower operating voltage of 680 mV using a negative-level write driver. Nevertheless, these devices and methods all require dedicated signal timing and entail greater area cost in SRAM [38, 40, 41].

As described previous chapter, an 8T 1R1W dual-port SRAM is typically used for leveraging disturb-free access because of the dedicated read port. A conventional 8T

dual-port SRAM cell consists of six transistors as a 6T SRAM cell and a decoupled two-transistor read port. This structure can eliminate the well-known read disturb problem by preventing charge sharing with internal storage nodes when a read wordline is activated. A single-ended read port of an 8T dual-port SRAM employs a sourceline as a footer line, which is shared in the same row address to perform low-energy operations. This 8T 1R1W dual-port SRAM reduces leakage current through unselected read bitlines. Some read bitlines are, however, discharged slightly in unselected columns because the floating sourceline of the conventional scheme [30] degrades energy efficiency in the read out operation.

4.1.2 Selective Sourceline drive (SSD) scheme

The conventional 8T 1R1W dual-port SRAM with the selective sourceline control (SSLC) scheme in Figure 4.2 presents an illustration of a memory matrix and a conventional SL control scheme [30]. The memory matrix commonly employs an interleaving structure. The SL in this scheme acts as a "virtual ground line" for a single column. The SL has two states: a grounded state and a floating state. A selected read bitline (RBL) is connected to the ground through a transfer gate in the conventional structure, whereas SLs at unselected read ports become floating. The floating node of the unselected SL is charged up when a read out datum is "0" on an RBL. The RBL voltage is not a full swing because of the cutoff SL, but it is unnecessary and consumes certain energy in the conventional scheme.

Fig. 4.3 presents an illustration of the concept of the proposed 1R1W dual-port SRAM with the selective sourceline drive (SSD) scheme. It has a pair of nMOS switches (M1 and M2), an inverter, as a footer circuit in every column. Those switches keep the SL on the ground for the selected column, or drive the SL to VDD−Vth (Vth is a threshold voltage of M1) when the column is not selected. In the standby mode, all the SLs are grounded to prepare for upcoming random read access. The right panel also depicts the proposed SSD scheme behavior. In the read operation, the SL discharge is enabled (SDE). The signal for all columns is "0". The column select signal (COL_E) from column-decoder activates a target column switch. In the selected column, the output of the OR gate becomes "1." The M2 transistor is activated to read out the stored data. At this time, the charge and discharge power is consumed only at the selected RBL because the SL at the selected column is grounded.

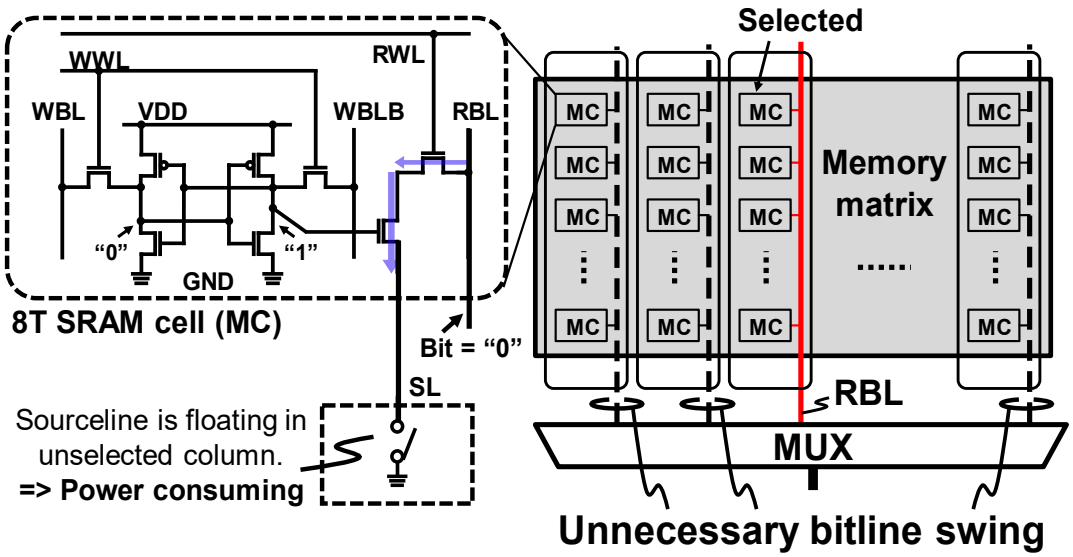
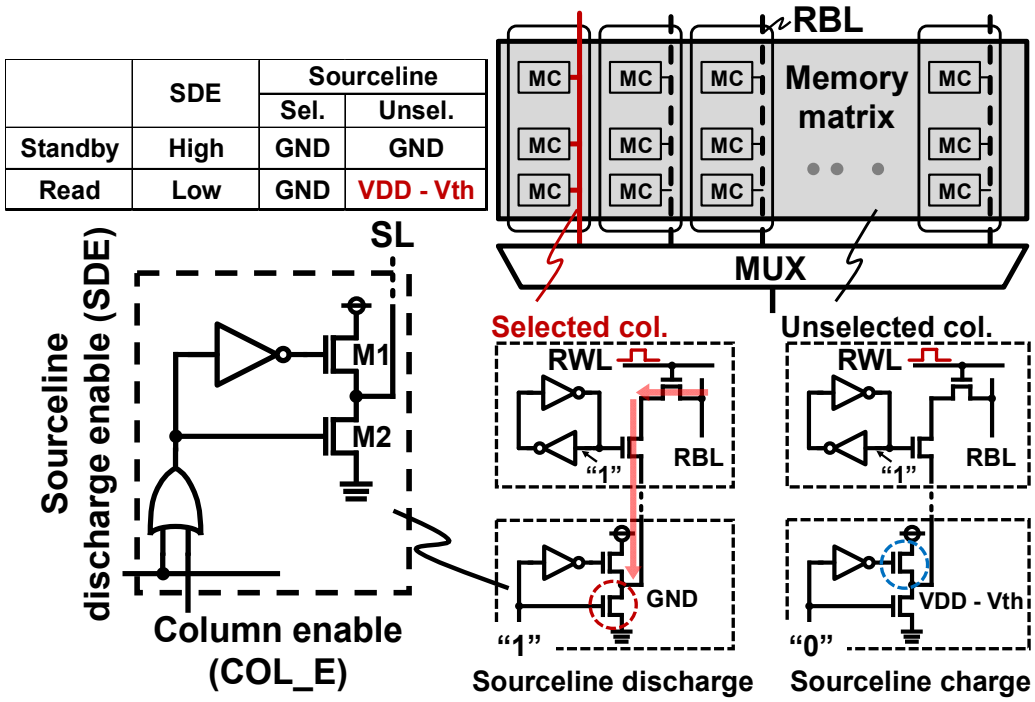


Fig. 4.2 Conventional 8T SRAM memory matrix with the SSLC scheme. Unselected SLs are floating because of an nMOS switch. They consume unnecessary energy.



	SDE	Sourceline	
		Sel.	Unsel.
Standby	High	GND	GND
Read	Low	GND	VDD - Vth

Fig. 4.3 Concept of proposed 8T SRAM with the selective sourceline drive (SSD) scheme in the read operation.

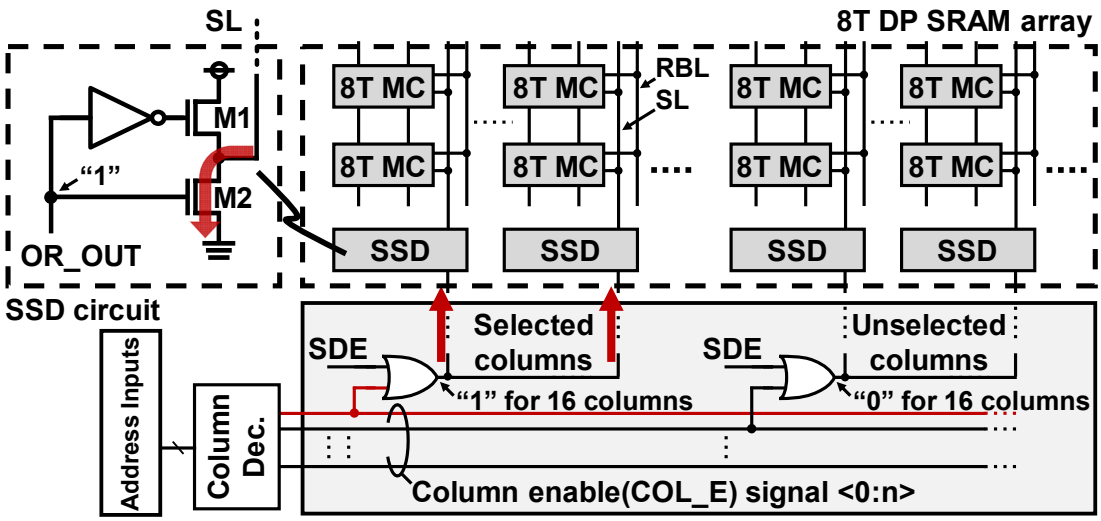


Fig. 4.4 Column control in the proposed SSD scheme in read operation.

In the unselected column, the OR gate output is low. The M1 transistor drives the SL voltage to VDD−Vth. Under these circumstances, no read current is flowing through the RBL. The nMOS transistors and an inverter circuit in the SSD scheme consume switching power to maintain the SL voltage (= VDD − Vth). However, the switching power is sufficiently smaller than the RBL discharge that is connected to the 256 cells/column.

Fig. 4.4 shows the concept of column control under the proposed SSD scheme. An OR gate is inserted as a column controller in every 16 columns to enable the SSD scheme. An inverter and two nMOS transistors are needed for every column. By contrast, in the conventional SSLC scheme, an nMOS transistor as a sourceline (SL) switch and an OR gate as a column selector are deployed in every column to control the SL connectivity. The proposed SSD scheme has less area overhead than the SSLC scheme. It is noteworthy that the column address inputs and a column decoder are used as well as the conventional column addressing circuitry. Therefore, no additional circuitry is necessary for column addressing in the SSD scheme. Each OR gate is activated by the SDE signal and a column address decoder output as the column enable (COL_E) signal. The SL voltage is discharged to the ground by activation of the M2 transistor if the output of the OR gate is set to "1".

Figs. 4.5(a)–(c) portray a signal timing flow comparison between the conventional

SSLC scheme and the proposed SSD scheme in read operation. Fig. 4.5(a) shows SL control circuits that have an OR gate to control a target column commonly used in both the conventional SSLC and the proposed SSD. The SDE signal as an input signal for OR gate is initially set to “1” to disable the switch control, but it is turned to “0” as the first step of the read operation. Then, the column select (COL_E) signal chooses a target column. The switches are activated. In the selected column, the switches pull down the SL voltage to the ground in both SSLC and SSD. At this cycle, the RBL voltage falls down to the ground after the read wordline (RWL) is enabled in the “1” read operation. However, in an unselected column, the single nMOS switch separates the SL from the ground in the SSLC. The SL becomes a floating node. However, the RBL voltage is discharged slightly at every cycle because of leakage current, as presented in Fig. 4.5(b). By contrast, the proposed SSD scheme drives the SL voltage to VDD−Vth. The charge/discharge energy on the RBLs is eliminated even in unselected columns, presented in Fig. 4.5(c).

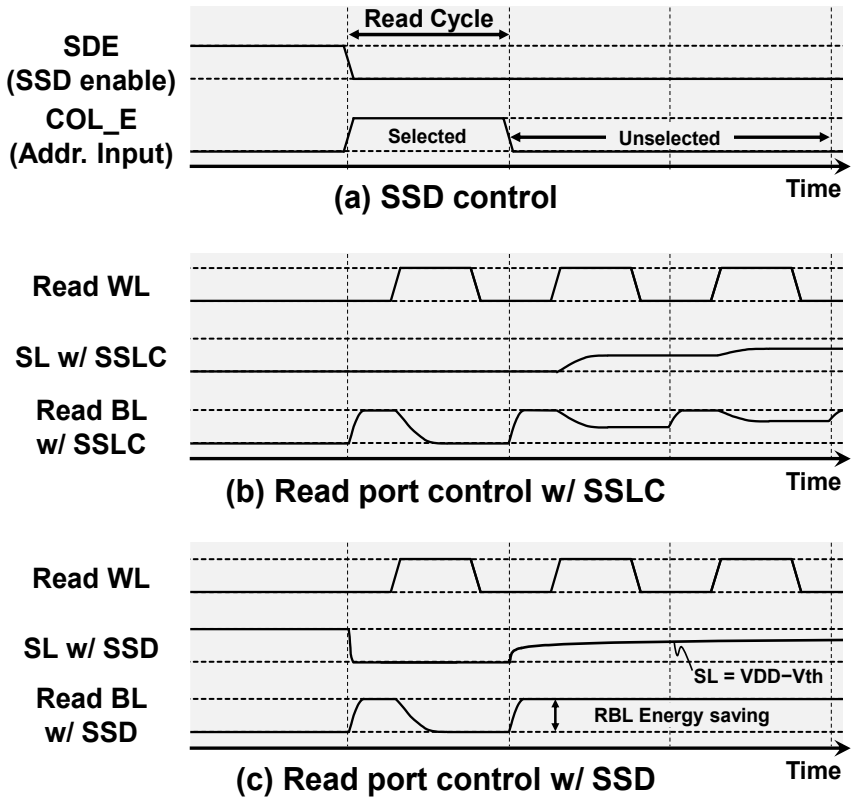


Fig. 4.5 Signal timing control flow in the conventional SSLC and the proposed SSD scheme in read operation: (a) SSD control signals, (b) read port control signals with the SSLC scheme, and (c) read port control signals with the SSD scheme.

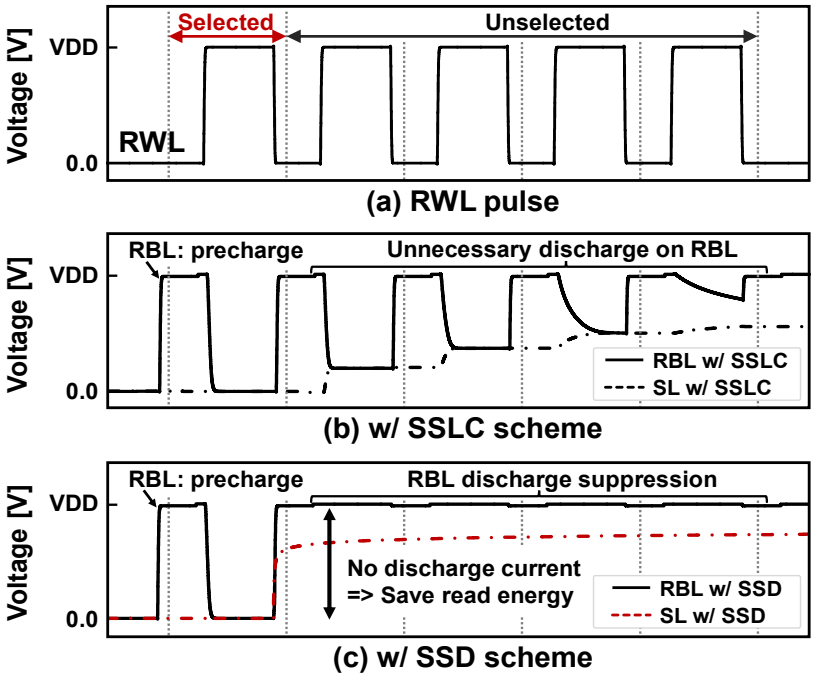


Fig. 4.6 Simulated "0"-read operating waveforms: (a) RWL waveform commonly used in the conventional SSLC scheme and the proposed SSD proposed scheme, (b) RBL and SL waveforms in the conventional SSLC scheme, and (c) RBL and SL waveforms in the proposed SSD scheme.

Figs. 4.6(a)–4.6(c) present a comparison of the simulated waveforms for "0"-read operation in the conventional SSLC scheme and the proposed SSD scheme. The RWL waveform is commonly used in the conventional SSLC scheme and the proposed SSD scheme depicted in Fig. 4.6(a).

Fig. 4.6 (b) shows the RBL and the SL waveforms in the "0"- read operation simulated with the conventional SSLC scheme. In the selected column, the SL is connected to the ground. The RBL voltage is discharged to the ground voltage after input of the RWL pulse. In the unselected column, the SL is separated by the nMOS switch from ground. The dedicated read port is activated by the RWL pulse. Then, the RBL voltage is pulled down slightly because of leakage current. It consumes unnecessary power at every cycle. Fig. 4.6(c) depicts the RBL and the SL waveforms with the proposed SSD scheme in the "0"-read operation. In the selected column, the SSD scheme connects the SL to the ground. Then the RBL voltage is discharged to the ground voltage after the RWL pulse input. However, the SSD scheme drives the SL voltage to VDD−Vth when the columns are not selected. The RBL voltage maintains the

VDD voltage and saves energy. In terms of power consumption, the SSD scheme is better than the conventional SSLC scheme. The proposed SSD scheme has no voltage swing on the local RBL. The output voltage amplitude of the SL is restricted to VDD−Vth because it minimizes the dynamic power consumption of the driver switches (M1 and M2) and its leakage current flowing through M2. The proposed SSD scheme therefore eliminates the unnecessary read current in the unselected columns. The dynamic power consumption on the two driver nMOSs in SSD scheme is made only when the selected column is changed. Therefore, the read operation in vertical memory addressing is effective to a considerable degree for the proposed SRAM.

4.1.3 The 1R1W 8T SRAM bitcell design

Figs. 4.7(a)–4.7(c) show the proposed 8T 1R1W dual-port SRAM cell schematic and custom layout designs with a separated SL architecture in the 28-nm FD-SOI process technology. Fig. 4.7(a) portrays a schematic of the proposed SRAM cell design. It has additional pass gate transistors PG3 and PG4 with additional RWL and RBL metals to draw the read current flowing through the dedicated read port.

The dedicated read port enables simultaneous but separate read and write operations. The gate node of PG4 transistor is connected to the cell internal node V2. Conventionally, the source node of PG4 transistor is connected to ground. In our design, the source node of PG4 transistor is comprises the SL, which is vertically connected to the SSD scheme to control the SL voltage on a column basis.

Fig. 4.7 (b) shows the FEOL layout of proposed SRAM cell. The bit area is 0.423 μm^2 designed on the logic rule base. Read ports comprising PG3 and PG4 transistors are arranged at the right side from a 6T SRAM cell. The PG4 transistor shares the same poly-gate metal with PU2 and PD2 transistors as the V2 node. Fig. 4.7(c) depicts the BEOL layout design of proposed SRAM cell. Conventionally, the source node of the dedicated read port can be shared with an adjacent cell because all source nodes are grounded. In the proposed SRAM cell, the source nodes are connected vertically to the SSD scheme. Thereby adjacent source nodes must be separated. In our design, the additional SL metal is located at the right end of figure with the Metal3 layer instead of the conventional ground line. The SL metal can be accommodated on the cell area. There is no area overhead for the additional SL metal on the cell design.

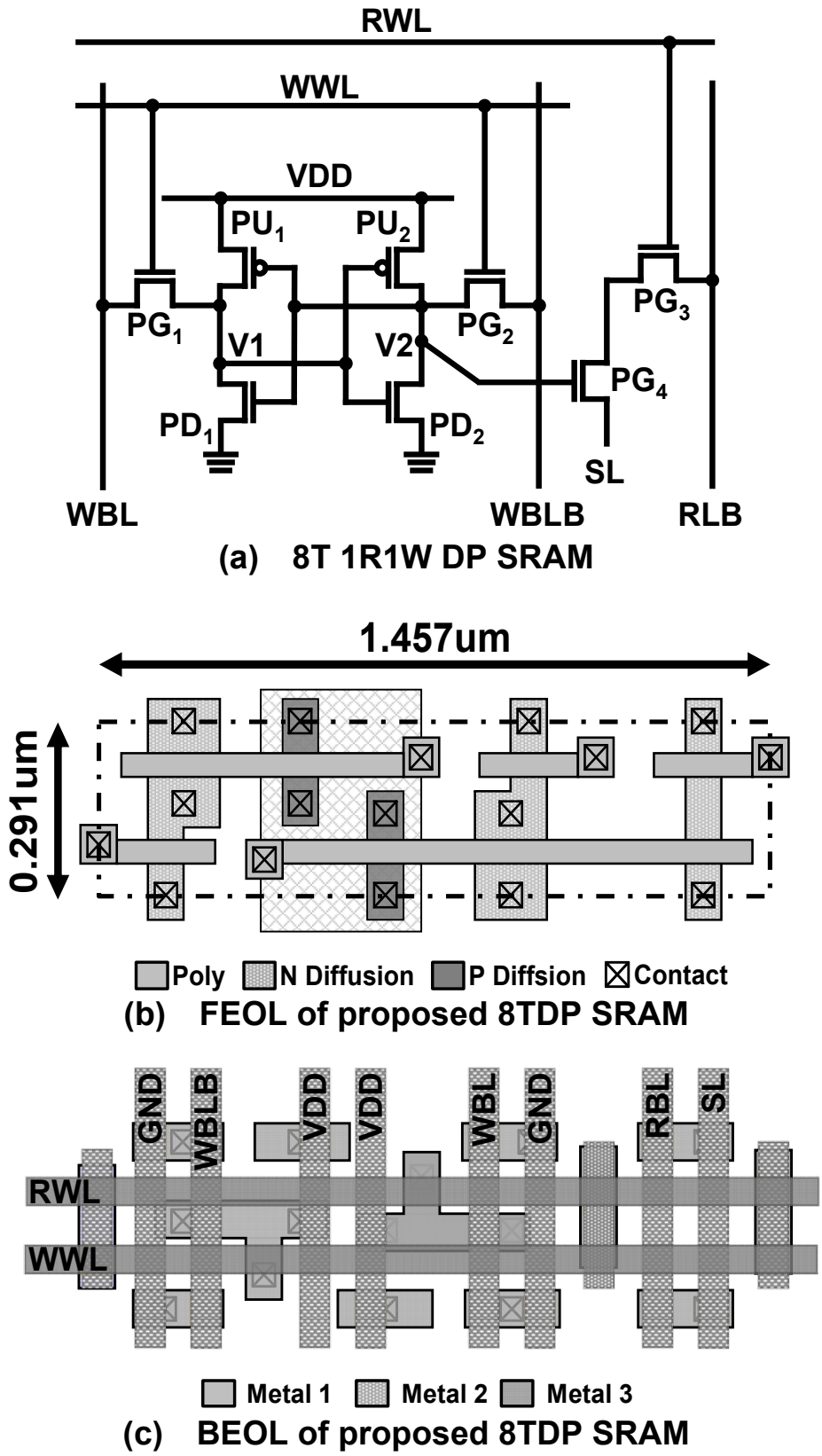


Fig. 4.7 Proposed 8T 1R1W dual-port SRAM cell in a 28-nm FD-SOI: (a) circuit design, (b) FEOL layout design, and (c) BEOL layout design.

4.1.4 RBL delay and area optimization in SSD scheme

The single-ended 1R1W dual-port SRAM generally uses an inverter circuit as a sense amplifier (SA). This type of SA is beneficial for high-density single-ended SRAM design by virtue of its simple structure. In a "0"-read operation, the RBLs are discharged by the activated read ports. In the single-ended read port, RBL voltage must be fully discharged to the ground to sense the stored datum. The RBL delay performance with local/global variations affects the read out timing setup. The RBL delay depends mainly on the SRAM cell transistor sizing and the pull down nMOS switch size in the SSD scheme.

Fig. 4.8 illustrates the RBL delay versus the width of the SL pulldown nMOS switch simulated with 1M Monte Carlo at SS-corner/−40°C. A smaller pulldown nMOS switch in the SSD scheme slows the RBL delay, whereas a larger size increases the capacitance on the RBL. The 5σ point of the RBL shows delays at each nMOS size in Fig. 4.8. The slowest RBL delay is shorter by 24.9% at 640 nm width, compared with the case of 80 nm width. The RBL delay improvement is saturated even if the switch size is increased further.

The area overhead is another factor; it increases linearly according to the nMOS width. In our design, we choose the pulldown nMOS switch width of 640 nm for the SSD scheme implementation with only 0.9% area overhead.

Fig. 4.9 presents the RBL delays simulated at FF-corner/125°C. The −5σ of RBL delays at FF-corner/125°C in Fig. 4.9. It is improved by 20.7% at the switch width of 640 nm with the area overhead of only 0.9% in the whole memory macro, compared with the case of 80-nm width. Herein, the nMOS switch width is optimized by considering the tradeoff between the RBL delay and the area overhead.

Figs. 4.10(a)–4.10(b) present read out waveforms of the slowest cells in the conventional grounded SL scheme, the SSLC scheme, and the SSD scheme when Monte Carlo analyses are executed at TT, 25°C. Fig. 4.10(a) shows simulated waveforms of the SA output signal V(SAout). Fig. 4.10(b) shows the simulated RBL waveforms in the read operation in the SSD scheme with 1 million iterations of Monte Carlo analyses. As described in this paper, t_{delay} is defined by the time to which V(SAout) rises to 0.45 V at supply voltage of 0.5 V. Although the fastest RBL is fully discharged, the slowest RBL is still in a half VDD voltage. The read out time in the

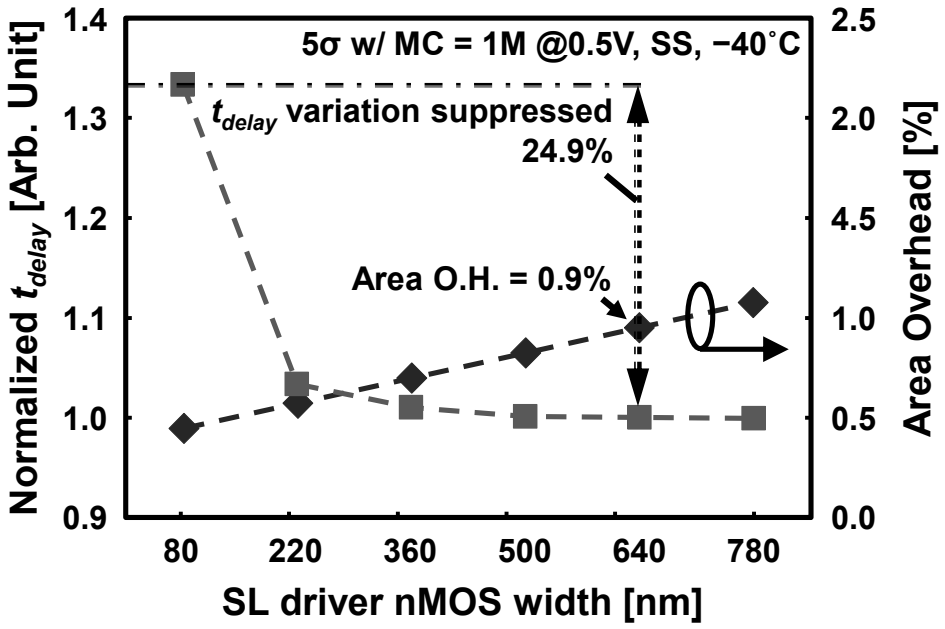


Fig. 4.8 Read bitline (RBL) delay t_{delay} and area overhead versus the gate width option of the SL pull-down nMOS transistor M2 (SS corner, -40°C). The slowest RBL delays at each gate width size are shown.

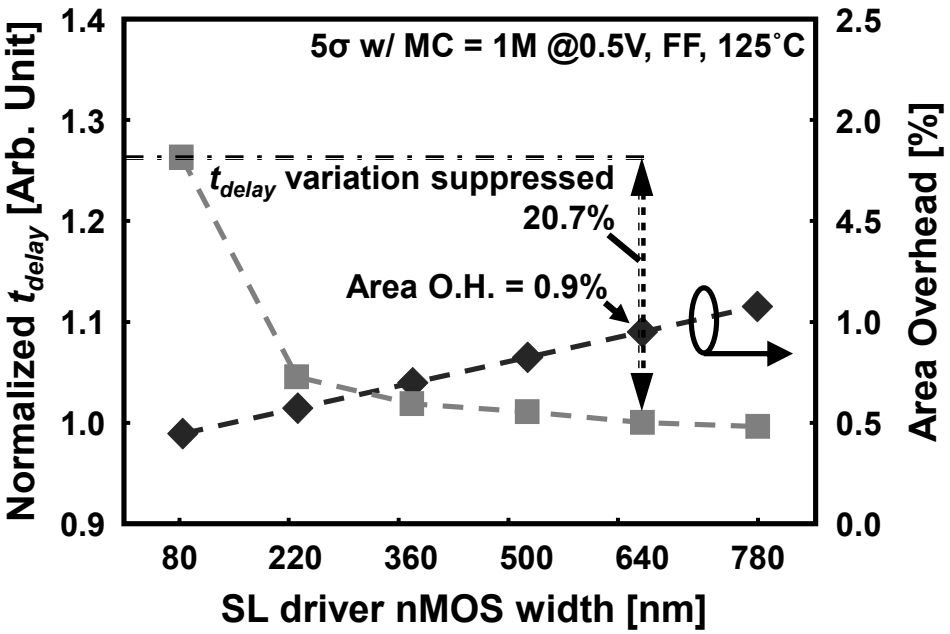


Table 4.9 Read bitline (RBL) delay t_{delay} and area overhead versus the gate width option of the SL pull-down nMOS transistor M2 (FF corner, 125°C): The fastest RBL delays at each gate width size are shown.

proposed SSD scheme is affected by the transistor size of the nMOS switches in the SSD scheme because the SL in the SSD turns out to be grounded immediately after a column is selected, which is regarded as a setup time. In any case, the SL discharge delay in the proposed SSD scheme is expected to be much shorter than the RBL discharge delay time. The t_{delay} in the grounded SL scheme, which value is 43.45 ns, is most strongly affected by the *V*th variations in the SRAM cell transistors. In the SSLC scheme, t_{delay} is 36.46 ns; in the SSD scheme, t_{delay} is 38.95 ns, which are 16.1% and 10.3% shorter, respectively, than that on the grounded SL at TT, 25°C.

Additionally, one must consider the hold time until V(SAout) becomes 50 % of operating voltage. The t_{delay} variations should be evaluated for a hold time margin in read out operation. Table 4.1 presents a summary and statistical evaluation of t_{delay} on RBL between the SL control schemes of three types when executing 1M Monte Carlo analyses at SS, −40°C, and varied operating voltage change from 0.4 V to 0.7 V. The SL circuit which adopted a voltage control scheme as transistor stacking should increase the average or median value of t_{delay} time because of the capacity increase, as presented in

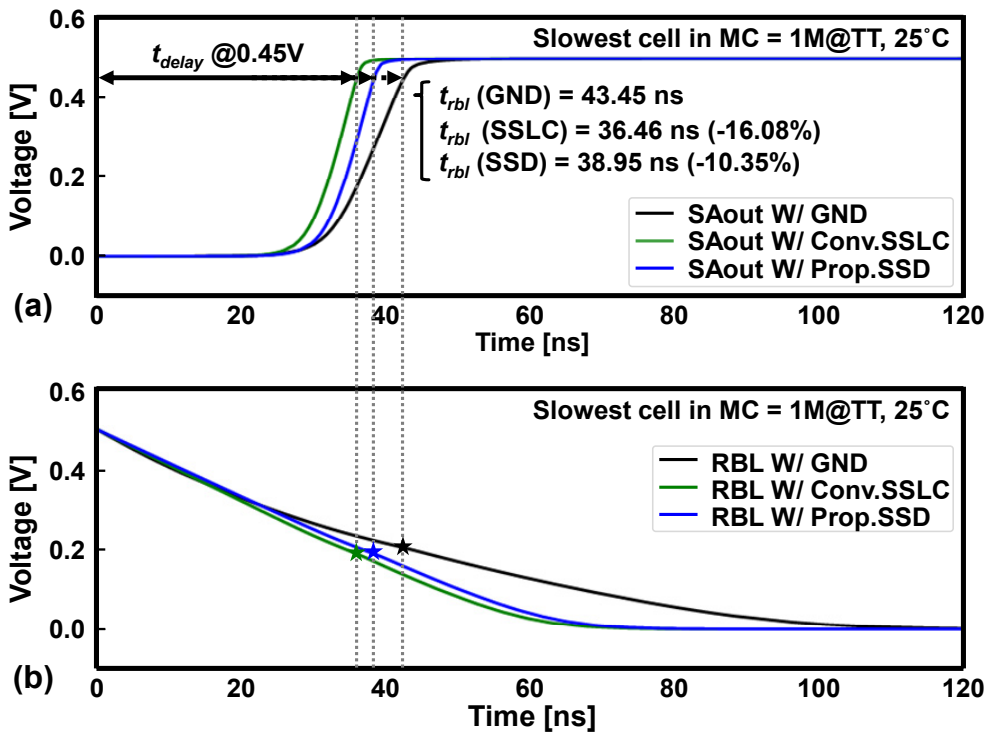


Table 4.10 Simulated read out delay comparison between conventional grounded SL, conventional SL with SSLC scheme, and proposed SL with SSD scheme.

presented in TABLE 4.1. It is apparent that the average t_{delay} increases by 4.2%. The

Table 4.1. It is apparent that the average t_{delay} increases by 4.2 %. The median value increases by 7.5 % at 0.5 V operation in the proposed scheme. However, statistics such as skewness and standard deviation of the t_{delay} distribution are slightly lower because of the nMOS transistor stack forcing condition [42].

The standard deviation is shown to decrease by 3.2%; the skewness decreases by 9.2 % at 0.5 V operation in the proposed scheme. The −5σ and 5σ values are shown as the t_{delay} *Min* and the t_{delay} *Max* in Table 4.1. Actually, t_{delay} *Min* increases 0.25 ns. Also, t_{delay} *Max* decreases 53.70 ns with SSLC compared with grounded SL. In addition, in the SSD, t_{delay} *Min* increases 0.31 ns and t_{delay} *Max* decreases 29.90 ns compared with conventional grounded SL.

Fig. 4.11 shows normalized delay on RBL at varied supply voltages of 0.4 V – 0.7 V for hold margin evaluation. Each plot extracted from the slowest cell which is supported 5σ coverage at SS, −40 °C. The t_{delay} on RBL is much lower than that of grounded SL: 30.2 % at a voltage of 0.4 V. Actually, t_{delay} does not have a normal distribution, and thus skewed. Fig. 4.12 shows t_{delay} distributions at 0.4 V, SS, and −40 °C, in the conventional grounded SL, the conventional SL with the SSLC scheme, and the proposed SL with the SSD scheme on the quantile–quantile plot. To statistically analyze the t_{delay} distribution, the horizontal t_{delay} is converted by the logarithmic function, $\log_{10}(t_{delay})$, to best fit its skewness to a straight line, which implies that t_{delay} is determined by near-subthreshold current under this condition. The respective mean values of the conventional grounded SL and the proposed SSD scheme are −6.533 and −6.513, respectively. Results show that the proposed SSD is slower than the grounded SL, on average. However, the standard deviations for the conventional grounded SL and the proposed SSD are, respectively, 0.328 and 0.313.

Table4.1 Statistical data comparison between different SL-structure in read operation.

Voltage	Corner, Tmp.	SL structure	Mean [ns]	Median [ns]	Std. [ns]	Skewness	Kurtosis	t_{delay} Min [ns]	t_{delay} Max [ns]
0.4 V	SS, –40°C	SL w/ GND	394.40	258.30	385.80	5.68	117.90	15.84	28,760.00
		SL w/ SSLC	404.20	276.40	380.60	5.03	75.01	20.02	20,660.00
		SL w/ SSD	405.20	285.40	382.40	4.92	70.00	21.24	20,070.00
0.5 V	SS, –40°C	SL w/ GND	17.26	13.26	12.58	5.04	81.02	2.59	681.20
		SL w/ SSLC	17.96	14.03	12.51	4.63	62.88	2.84	627.50
		SL w/ SSD	17.99	14.26	12.54	4.57	59.38	2.90	651.30
0.6 V	SS, –40°C	SL w/ GND	2.62	2.45	0.79	1.94	14.18	0.96	24.33
		SL w/ SSLC	2.75	2.58	0.81	1.85	12.40	1.01	23.49
		SL w/ SSD	2.76	2.59	0.81	1.86	12.14	1.06	24.17
0.7 V	SS, –40°C	SL w/ GND	0.99	0.97	0.17	0.86	4.60	0.52	3.20
		SL w/ SSLC	1.04	1.02	0.17	0.85	4.53	0.53	3.03
		SL w/ SSD	1.04	1.02	0.17	0.85	4.53	0.55	2.92

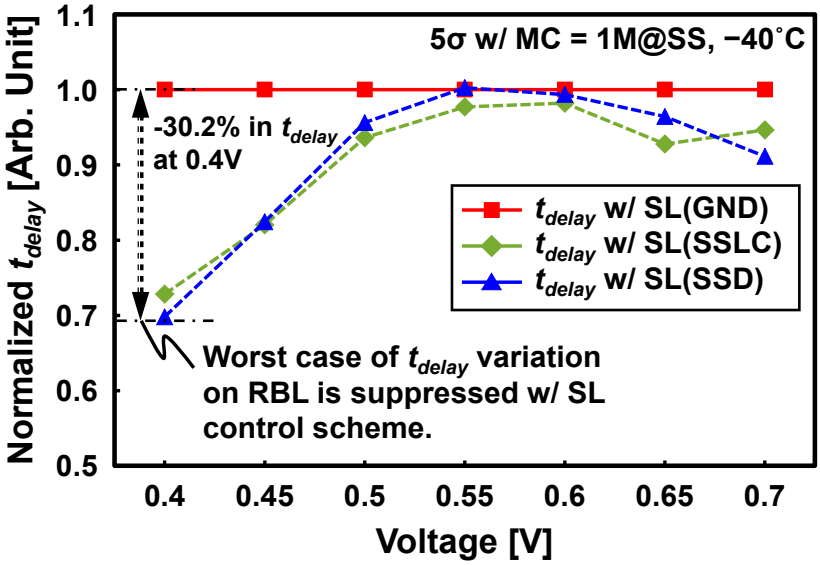


Fig. 4.11 Simulated read out delay comparison on slowest cells at varied supply voltage between conventional grounded SL, conventional SL with SSLC scheme, and proposed SL with SSD scheme.

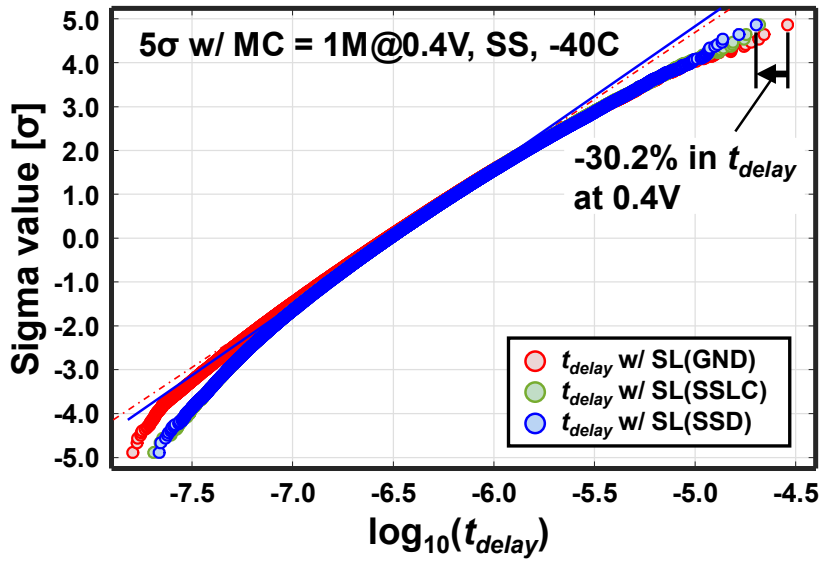


Fig. 4.12 t_{delay} distributions in the conventional grounded SL, the conventional SL with the SSLC scheme, and the proposed SL with the SSD scheme on a quantile-quantile plot.

The standard deviation of the proposed SSD is smaller. As depicted in Fig. 4.8, the standard deviation is significantly impacted by a size of the pull-down transistor M2 connected to the SL. If it is sufficiently large, then the mean value of t_{delay} is close to the conventional grounded SL. Its variation is suppressed in the proposed SSD. According to the probability theory, the standard deviation is suppressed by an increase in the

number of transistors [42, 43, 44]. The proposed SSD has one more transistor on the SL. For these reasons, t_{delay} of the proposed SSD is slightly smaller than the conventional grounded SL at 5σ of the percentile value.

4.1.5 Operating speed evaluation in the write cycle

The proposed SRAM employs the precharge-less write circuit to reduce the energy consumption on the WBL. Fig. 4.13 shows the precharge-less write waveforms of "0"-write operation in 1M Monte Carlo analyses. In this figure, the "1" data are initially stored in the SRAM cell. In the write operation, input data transfer to the WBL/WBLB without the precharge sequence. The WBLs have no equalization because of the precharge-less write scheme. Therefore, the same input data consume less energy on the WBLs. The write delay to flip the internal node voltage shown with the internal node V1 and V2 waveforms on the fastest and slowest cell is focused in internal nodes V1 and V2. Inversion of the internal node is started when the WWL pulse is inputted. In this figure, inversion time t_{write} is defined by the time to node V1 rises to 0.45 V. In

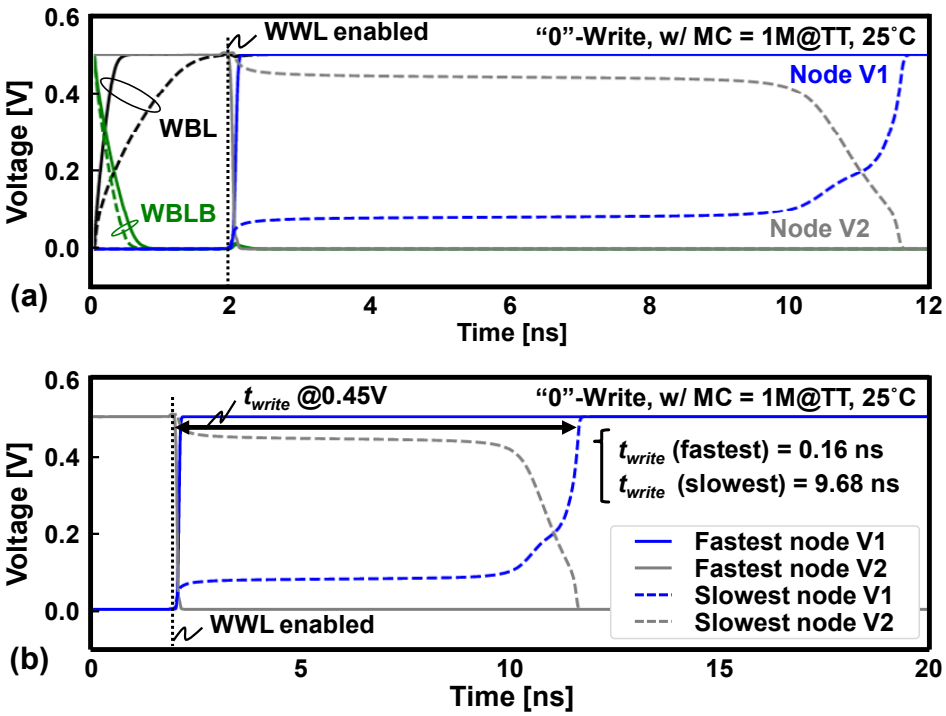


Fig. 4.13 Consecutive memory access in video processing: (a) block diagram and (b) waveforms in write operation.

addition, t_{write} for the fastest cell at −5σ is only 160 ps, whereas the t_{write} for the slowest cell at 5σ is 9.68 ns. In the whole write sequence, write access time with 11.56 ns is achieved at TT, 25°C, which is much shorter than that of the read operation.

4.1.6 Consecutive memory access in video processing

Image data such as people or landscapes reflect luminance information. They have similar brightness in adjacent pixels. Figure 4.14 presents the switching possibility of a read out bit between a present pixel and a next pixel. The averaged switching possibility obtained from the three sample images is 49.8% on the least-significant bit (LSB = 1st digit), meaning that the value of the LSB is random, which is reasonable. The most-significant bit (MSB = 8th digit) has a switching possibility of 7.8% on average because it has much stronger correlation between adjacent pixels.

Memory mapping for image data in the proposed video processing is performed on a channel-by-channel basis. Correlation between adjacent pixels is retained on RGB channels. Our consecutive memory access is beneficial for optimizing power consumption even if image data have multiple dimensions of channels. Therefore, in the consecutive accesses, it is better to map their addresses along the row direction, as presented in Fig. 4.15(a), where a column address is not changed often. Fig. 4.15(b) depicts the waveform of the proposed SRAM in write operation. By virtue of the precharge-less and incremental write operation, the proposed SRAM reduces the write energy; the charge/discharge on a pair of write bitlines (WBL and WBLB) is consumed only when a write datum is changed. The consecutive writing of the same datum consumes less energy because the proposed dual-port SRAM has a dedicated write port and needs no precharge scheme on the WBLs.

However, it is adversely affected by the well-known half-select problem along the write wordline (WWL). The divided wordline structure is therefore adopted only for the write port to avoid the half-select problem [35]. The read port has the common interleaving structure with no divided RWLs because an image processor often requires a greater number of read ports and therefore exerts strong effects on its area.

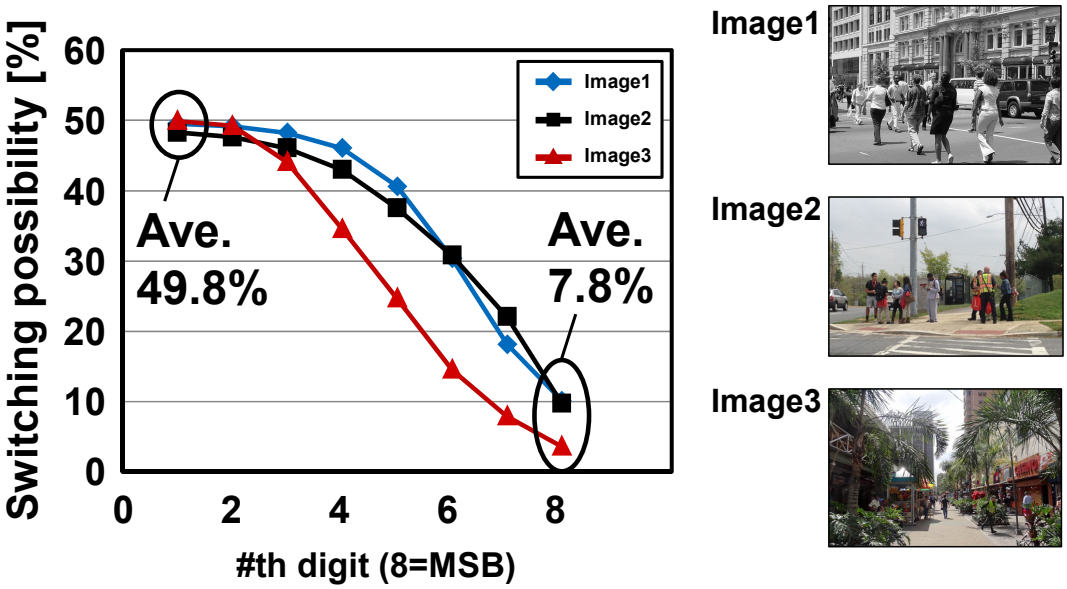


Fig. 4.14 Switching possibility in image data.

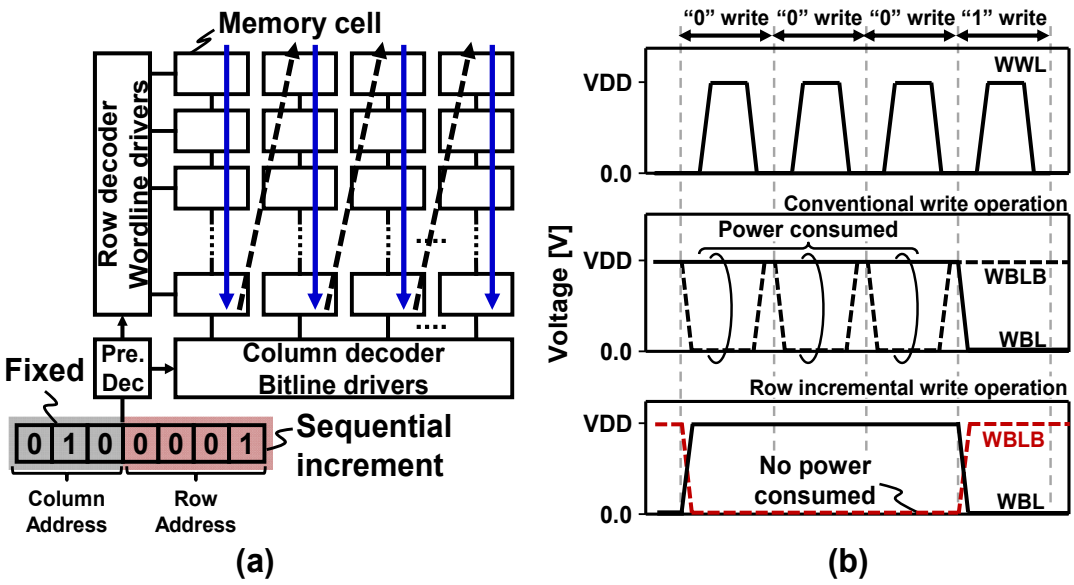


Fig. 4.15 Consecutive memory access in video processing: (a) block diagram and (b) waveforms in write operation.

4.1.7 Chip implementation and measurement results

Fig. 4.16 shows a chip layout of the proposed 64-kb SRAM macro configured with 32 kb × 2 banks with X/Y decoders, read/write and I/O circuit. The macro size is 242 × 189 μm^2 (= 0.045 mm^2). Each 32-kb bank consists of 4 kb × 8 subarrays, which are configured with 256 rows and 16 columns. The area size of the 4-kb subarray is 2,157 μm^2 (= 1962.38 μm^2 memory array + 194.77 μm^2 peripheral circuit). The circuit for the SSD scheme is located under the Y decoder. Its area is 27.75 μm^2.

In our design, the area overhead of the proposed SSD circuit is only 0.9% of the entire macro. Fig. 4.17 presents a test chip micrograph. We fabricated a 64-kb 8T dual-port SRAM macro in the 28-nm FD-SOI process technology.

Fig. 4.18 presents a measured Shmoo plot in read operation. We verified that the test chip can operate at a supply voltage of 0.48 V and with access time of 135 ns (= 7.4 MHz) at a room temperature: 25°C. The operating point that achieves the minimum energy per cycle is at a supply voltage of 0.56 V and a cycle time of 35 ns (= 28.6 MHz). In addition, Fig. 4.19 presents a measured Shmoo plot in write operations. The test chip

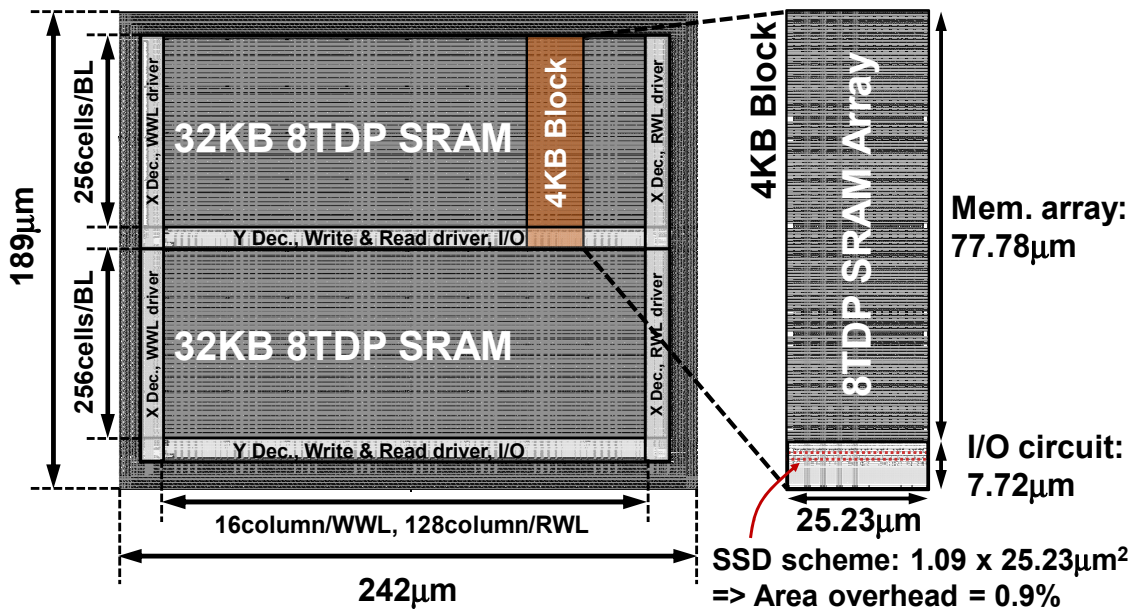


Fig. 4.16 A 64-kb SRAM (32 kb × 2 bank) macro layout design comprises 16 × 4-kb subarray block (= 256 × 16 cells).

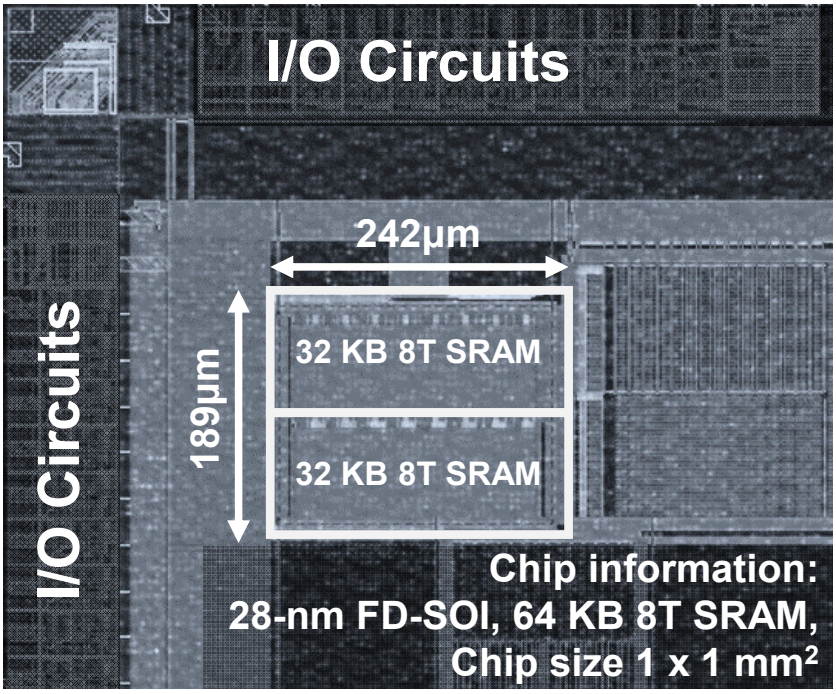


Fig. 4.17 Chip micrograph of the test chip.

can operate at a supply voltage of 0.46 V and a write pulse width of 56 ns. The shortest write pulse width is 4 ns at a supply voltage of 0.62 V.

Fig. 4.20 portrays a schematic of the proposed 8T dual-port SRAM array with the SSD scheme. The proposed 8T SRAM has a precharge-less write circuit. Consequently, successive writes of same data consume less energy. However, bit interleaving incurs the well-known half-select problem along the write wordline. The divided wordline structure is therefore adopted to avoid the half-select problem [35] in our design. The OR gate of the SSD scheme is connected to 16 read out circuits; it selects a target column for SL discharge or charge SLs to VDD − Vth in unselected columns.

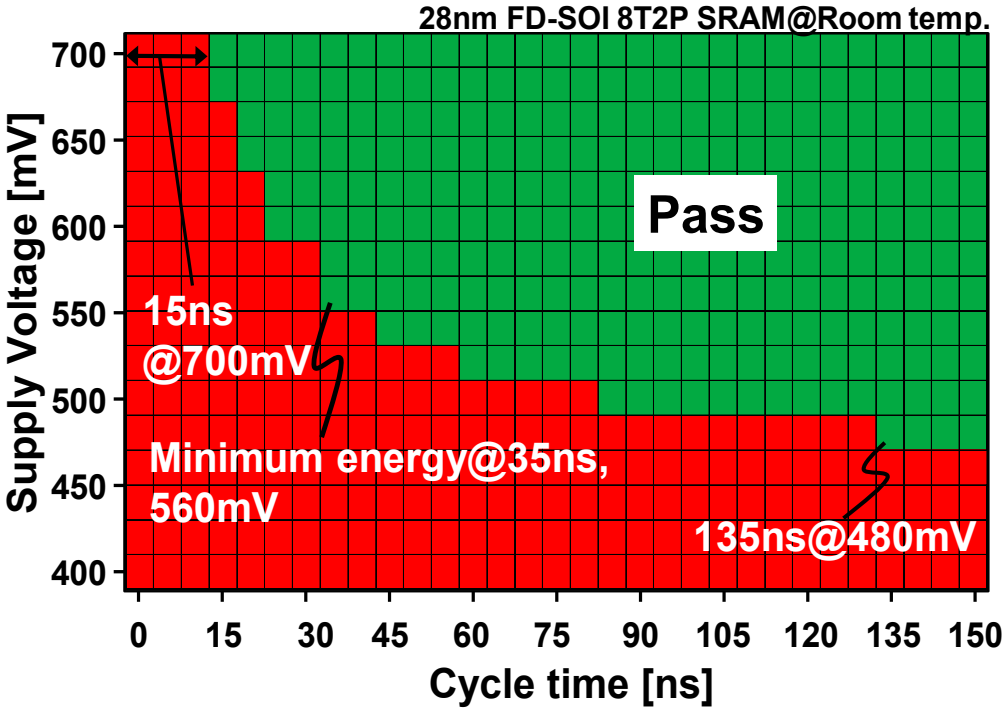


Fig. 4.18 Measured Shmoo plot in read operation.

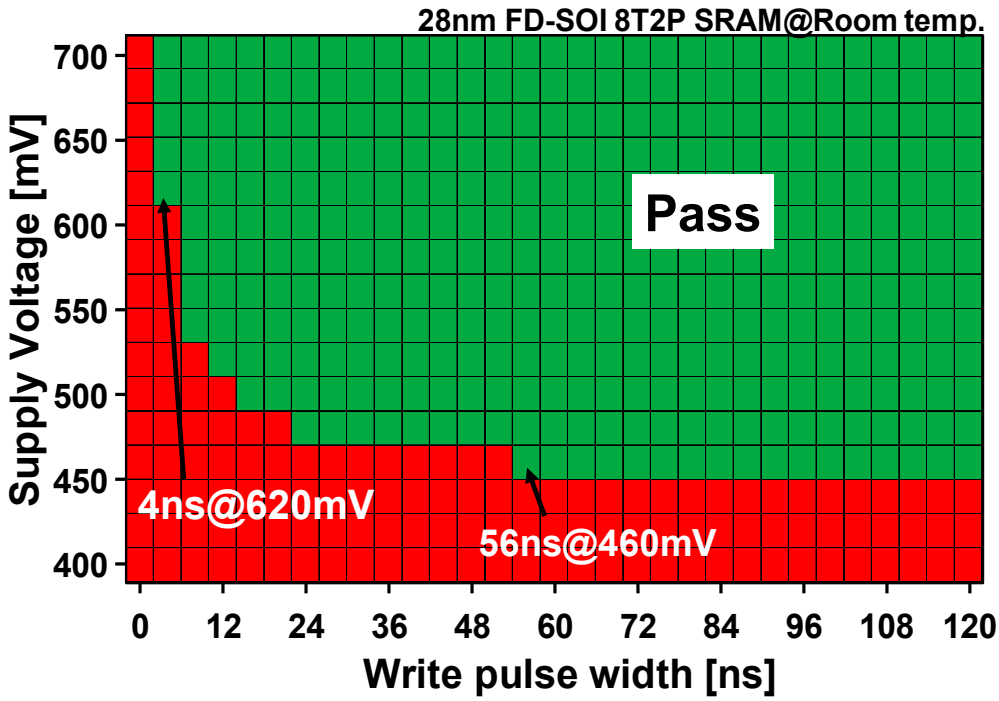


Fig. 4.19 Measured Shmoo plot in Write operation.

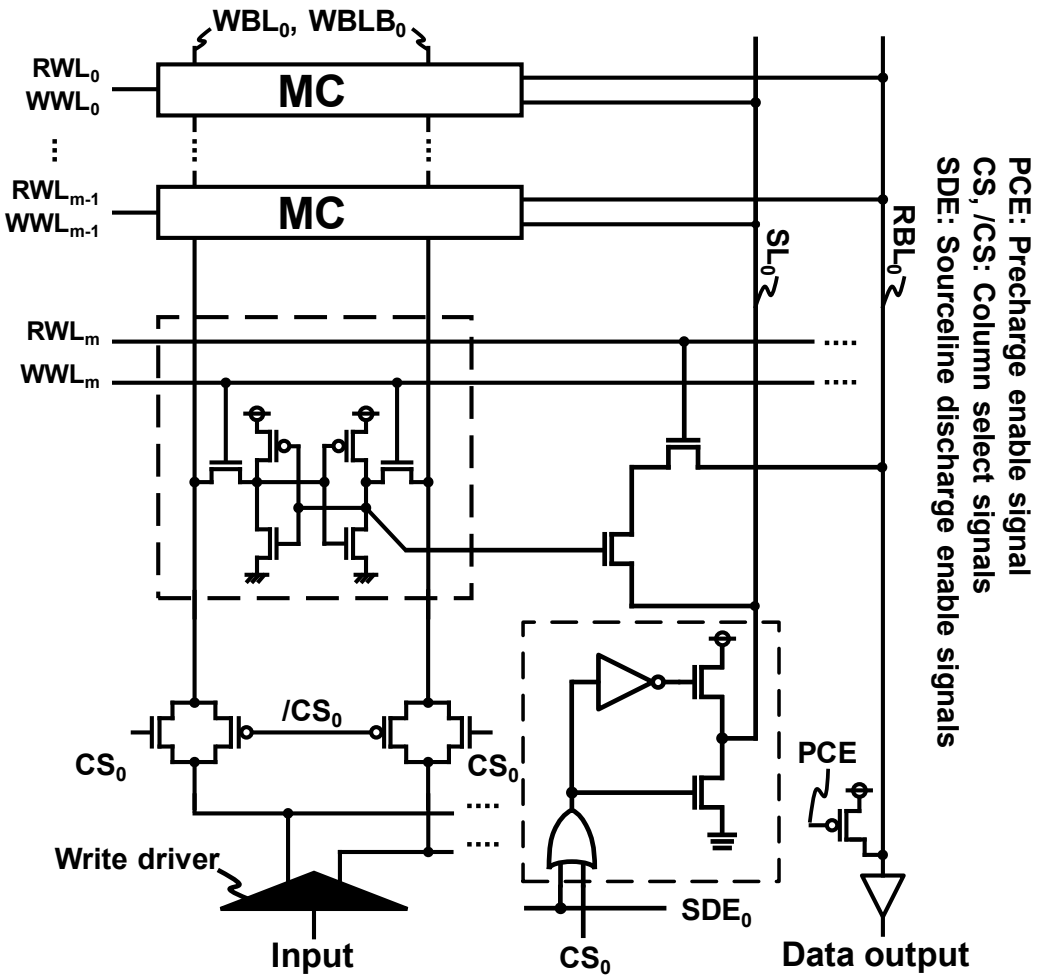


Fig. 4.20 Schematic of the proposed 8T dual-port SRAM with the SSD scheme.

Fig. 4.21 shows the simulated and measured active/leakage energy comparisons between the conventional SSLC scheme and the proposed SSD scheme for read operation. It is noteworthy that both read circuits must have the RBL precharge scheme because of their associated single-ended read ports. In the read operation, the test patterns of the “ALL 0” and “ALL 1” respectively denote the mean consecutive “0” and “1” read operations of the incremental row address accesses. The checkerboard patterns using the incremental row address (CKB X+) have 50.0 % “0” and 50.0 % “1” data with energies that are intermediate of “ALL0” and “ALL1”. In the CKB using the incremental column address (CKB Y+), the column address is changed at every cycle. The energy comparison demonstrates that the proposed SSD scheme improves the read energy by 26.0 % on average, which is 389.6 fJ/cycle.

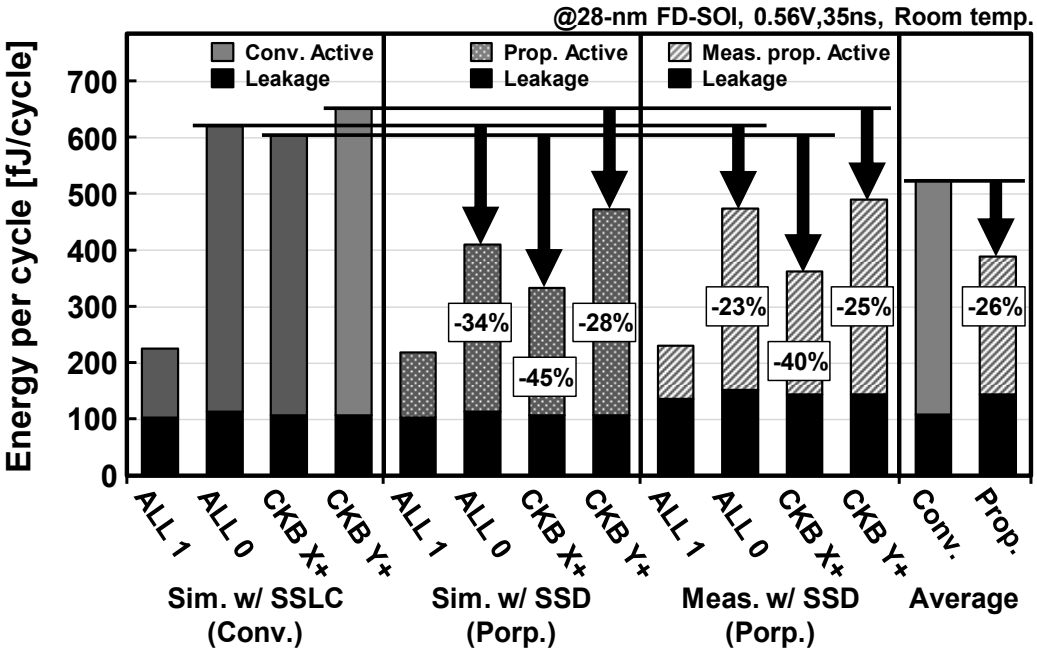


Fig. 4.21 Simulated and measured energy comparisons between the conventional SSLC scheme and the proposed SSD scheme in read operation.

Figure 4.22 again shows the simulated energy breakdown and a comparison between the conventional SSLC scheme and the proposed SSD scheme in “ALL 1” and “ALL 0” read operations. Although an RWL must be enabled at every cycle, its RBL charge/discharge does not occur in the “ALL 1” read operation because the read port is cut off with PG4 in the 8T cell. Unnecessary current is reduced in unselected columns. However, Fig. 4.22(a) shows that its energy saving is small because no RBL is discharged in this case. However, Fig. 4.22(b) shows the “ALL 0” read operation. The RBL charge/discharge takes place at every cycle. The RBL and the selected SL energy are increased drastically. Floating SLs in the unselected columns are discharged in the SSLC scheme, which consumes unnecessary read out energy. The SSD scheme can eliminate the energy wasted in the unselected column by 96.5% compared with the SSLC scheme.

In the write operation, the proposed 8T dual-port SRAM requires no precharge on the WBLs. Additionally, its WWL has a divided structure. Therefore, the proposed SRAM can reduce the unneeded write energy because of the charge/discharge in the half-selected columns. The measured “0” and “1” write energies are, respectively, 196.2 fJ/cycle and 215.2 fJ/cycle.

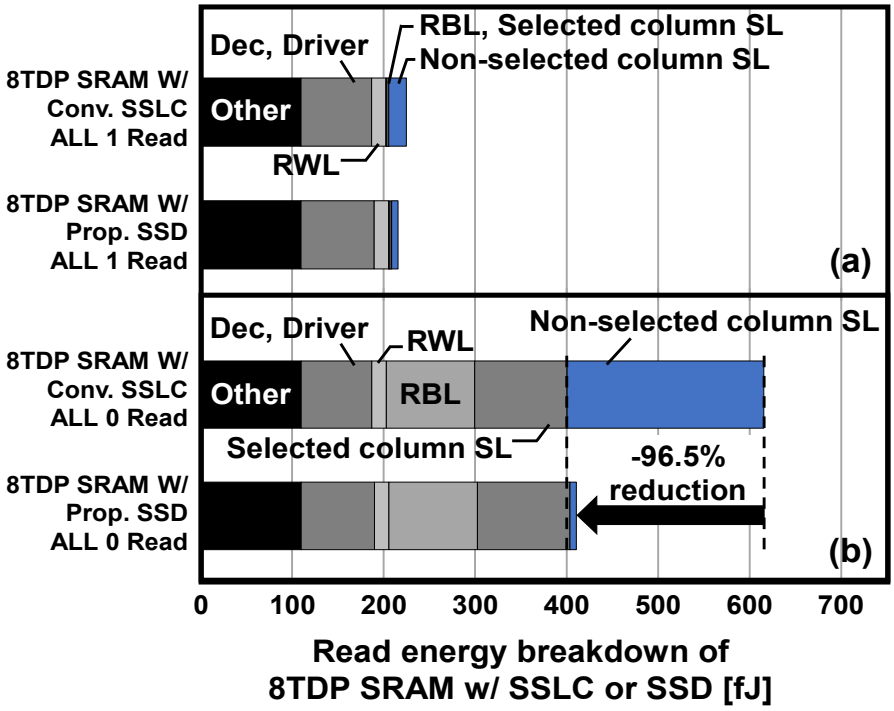


Fig. 4.22 Simulated energy breakdown comparison between the conventional SSLC scheme and the proposed SSD scheme in (a) “ALL 1” and (b) “ALL 0” read operations.

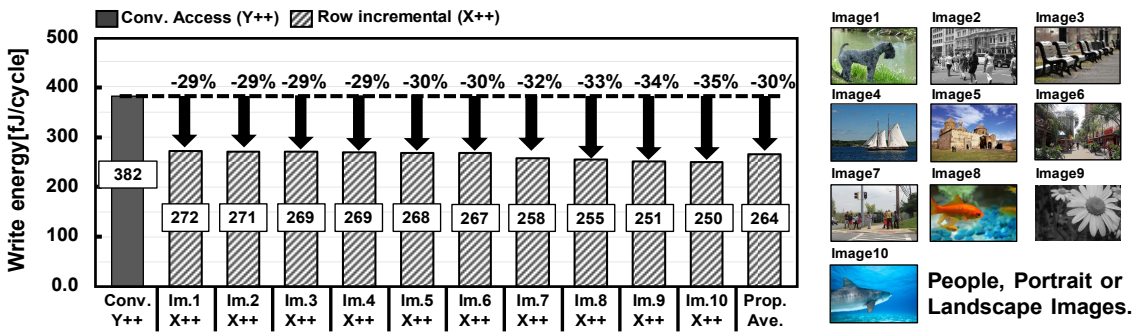


Fig. 4.23 Write energy saving in incremental accesses.

Table4.2 Tast chip features.

Technology	28-nm FD-SOI
Supply voltage	0.48-0.7V (Memory macro)
	1.8V (I/O)
Chip area	1.0x1.0mm^2
Macro size	189x242μm^2
Macro configulation	64Kb (32Kb X 2), 16bits/word
Cell size	0.291x1.457μm^2
Frequency	7.4MHz@0.48V, 66.7MHz@0.7V
Read active energy	389.6fJ@0.56V, 28.6MHz, RT
Write active energy	265.0fJ@0.56V, 28.6MHz, RT

Fig. 4.23 portrays the impact of the incremental write operation for write energy saving. As a counterpart, the write energy becomes 382.0 fJ/cycle in the consecutive column access in the CKB test pattern. Write energies are reduced according to the spatial frequency on its images. In the “color” Image 1 and monochrome Image 2, the write energy is reduced by 29%, whereas in the monochrome Image 9 and the color image 10, they respectively reach 34% and 35% of energy saving. Those features are beneficial in that the image has high similarity among all pixels. In Image 10, the write energy of the 250.4 fJ/cycle achieves 264.0 fJ/cycle, on average, which is 30% lower than that in the consecutive column access. Therefore, our proposed 8T dual-port SRAM with the SSD scheme can reduce both read and write energies. Moreover, it is suitable for low-power image processing devices. Table 4.2 presents a summary of the characteristics of the proposed SRAM test chip implemented in the 28-nm FD-SOI technology.

Table 4.3 presents a summary of a performance comparison among the state-of-the-art 1R1W dual-port SRAMs taken from recent conference and journal papers, as introduced in Section 1. For our test chip, we designed the SRAM bitcell on a logic rule basis. Therefore, the bit cell density is lower than [30] using a similar technology node. As described previously, we take an inverter as a large-signal sensing scheme. It is generally used for a lower area cost by virtue of its simple structure. However, the inverter incurs a full-swing signal when “1” is read out. A small-signal sensing scheme using a differential sense amplifier is often adopted [38, 40, 41], which consequently achieves much higher operating frequency. However, such circuitry

Table4.3 Dual-Port 8T SRAM Comparison.

Source	[30]	[38]	[39]	[40]	[41]	This Work
Technology	40nm bulk CMOS	40nm LP CMOS	65nm bulk CMOS	28nm bulk CMOS	16nm FinFET	28nm FD-SOI
Memory Size	16 kb	512 kb	96 kb	512 kb	256 kb	64 kb
Cell Size [um^2]	1.01*	0.8496	-	-	-	0.423*
Bit density (Mb/mm^2)	0.457	1.17	-	3.16	6.05	2.35
Sensing Scheme	Large Signal	Small Signal	Large Signal	Small Signal	Small Signal	Large Signal
Cell Type	8T-1R1W	8T-CP-1R1W	8T-1R1W	8T-1R1W	6T-1R1W	8T-1R1W
IO Size	16	64	8	32	64	16
# Bits/BL	128	32	128	256	128	256
Performance	10.0 MHz (0.5V/25°C)	800.0 MHz (1.1V/25°C)	1.8 MHz (0.55V/25°C)	1.69 GHz (1.0V/125°C)	1.2 GHz (0.88V/ -)	66.7 MHz (0.7V/25°C)
Functional Frequency	10.0 MHz (0.5V/25°C)	200.0 MHz (0.65V/25°C)	125.0 KHz (0.36/25°C)	1.69 GHz (1.0V/125°C)	1.2 GHz (0.88V/ -)	28.0 MHz (0.56V/25°C)
Power [uW/Access]	17.8 (0.5V/25°C)	902.0 (0.65V/25°C)	5.1 (0.36V/25°C)	16375.7 (1.0V/125°C)	14503.2 (0.88V/125°C)	9.16 (0.56V/25°C)
FoM [fJ/bit]	2.43	0.19	11.44	0.15	0.26	0.08

*Logic rule based SRAM cell design

requires dedicated signal timing and a greater area cost. The figure of merit (*FoM*) represents the energy per bit that includes standby and active energy, which is scaled by technology node scale factor *k*, operating frequency *freq*, the number of cells on a bitline l_{bl}, I/O bit width w_{io}, and the entire memory capacity *Cap*. In this case, *FoM* is as expressed as shown below equation: Eq. (13).

$$FoM = \frac{Power}{Cap * \ l_{bl} \ * w_{io} * freq * k} \tag{13}$$

Because of the unnecessary read energy reduction by the proposed SL voltage control scheme and because of the WBL charge and discharge energy saving with consecutive memory access, the *FoM* number is much more beneficial than other cutting-edge schemes. These results demonstrate the utility of the SL voltage control with the SSD scheme for low-power and low-voltage performance on 1R1W dual-port SRAMs.

4.2 Summary

In this section, I demonstrated two bitline limitation technology; the proposed selective sourceline drive scheme and a consecutive memory access technique.

1) As described in this section, we presented an 8T dual-port SRAM with the selective sourceline drive (SSD) scheme for an image processor. Our proposed SRAM drives the sourceline (SL) to VDD−Vth at unselected columns in read operation and exploits the consecutive row accesses in write operation for improving energy efficiency at low voltage. We fabricated a 64-Kb 8T dual-port SRAM using 28-nm FD-SOI process technology. The test chip exhibits 0.48 V operation with 135 ns access time. The energy minimum point is a supply voltage of 0.56 V at a 28.6 MHz frequency, at which 265.0 fJ/cycle in the write operation and 389.6 fJ/cycle in the read operation were achieved.

4.3 MSB Based Inversion Logic with Dual-Port 8T SRAM

In the previous section, we focused on the 1R1W dual-port SRAM. To reduce unnecessary current and leakage energy on the unselected RBL, we proposed footer circuitry. In this section, we address the another issue in 1R1W dual-port SRAM. To reduce active energy in read operation, we explain the MSB based inversion logic and circuit design for further active energy reduction on the selected RBLs.

4.3.1 Overview of dual-port SRAM and data-bit reordering

Fig. 4.24 (a)-(b) shows the schematic of 8T 1R1W dual-port SRAM and its simplified waveforms in the read operation. The charge/discharge energy is consumed in every cycle when the “0” data read due to the precharge scheme is adopted. In contrast, the deactivated transistor PG4 cut off the current flow from the read bitline (RBL) to the sourceline (SL) when the “1” data read. From this reason, increasing the possibility of the “1” read is lead to better energy efficiency because of the RBL charge/discharge current is reduced. Our earlier study demonstrated the majority logic with the data-bit reordering that is adopted on the I/O module in the conventional dual-port SRAM [45].

Normally, image data has luminance information. 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. To maximize the number of “1”s, the majority logic circuit considers the number of “1”s and judges if input data should be inverted in a write cycle. However, the conventional majority logic entail a huge area overhead, because the additional flag bits require the additional columns in the SRAM array. Furthermore, the sensing scheme to count the number of “1”s has considerable area overhead. In particular, an image processor requires a larger multiport SRAM capacity, which gives a serious impact on its cost [17–41].

In this section, the MSB-based inversion logic for an real-time and low-energy image processor is described. To minimize the area overhead, the proposed MSB-based

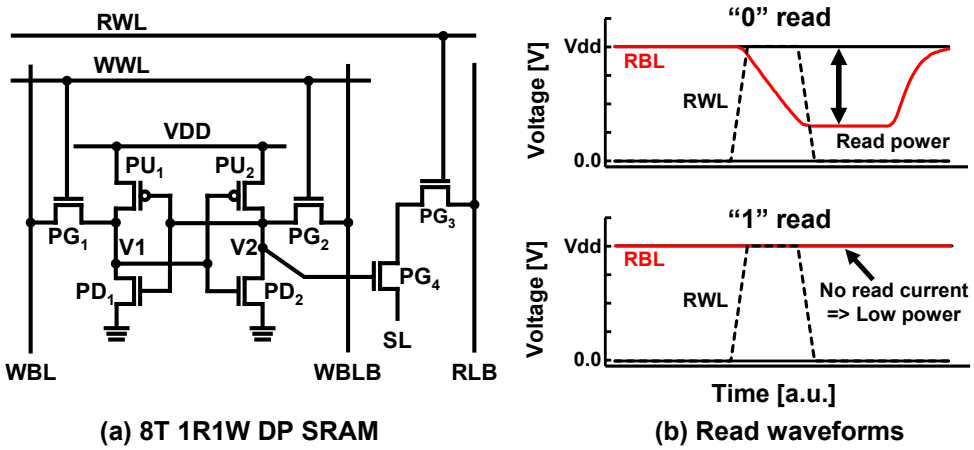


Fig. 4.24 8T 1R1W Dual-port SRAM (a) schematic and (b) Operating waveforms when "0" or "1" read operation.

inversion logic eliminates the additional flag bit. Within this chapter, detail of the proposed MSB-based bit inversion and the circuit design consideration in the 28-nm FD-SOI process technology is explained. Also, we discussed the total read energy reduction when we considered the computations of the convolutional neural network (CNN) for deep learning application with VGG-F CNN network model.

4.3.2 Proposed MSB based inversion logic

In the image datum, the strong correlation appears between the adjacent pixels. This tendency is predominant at the most significant bits (MSBs) in successive data, which lopsided to either "0" or "1" with high probability. The distributions of the number of "1"s in different digit groups are shown in Fig. 4.25. The LSB group takes a normal distribution and the similar distribution tendencies are observed even in the 2nd- and 3rd-digit groups. On the other hand, the strong correlation is observed in the MSB group. Probabilities taking on the 7th digit reach 58 % and 89 % in the Image 1 and Image 2, respectively. The data bit reordering exploits the correlation in image data [45]. In this study, we rearrange the data bit reordering for further power reduction on the RBLs.

Fig. 4.26 illustrates the concept of the conventional majority logic and the proposed MSB-based inversion logic, both with data bit reordering. Fig. 4.26 (a) shows the optimization flow in conventional majority logic. In the write cycle, input data

comprised of *m* pixels (8*m* bits) are reordered in each digit group. To maximize the number of "1"s, the majority logic counts number of "1"s and judges whether input data have to be inverted in each write cycle. The inversion information is stored in an additional flag bit. Although the majority logic optimizes the number of "0"s, it requires huge area overhead for additional circuitry to count the number of '1's and the additional flag bits to store the inversion information. Actually, the flag bits have to be accommodated in additional columns. Those overheads give serious impacts on its area and overall performance.

Fig. 4.26 (b) shows the proposed MSB based inversion logic. Input data comprised of *m* pixels (8*m* bits) are reordered in each digit group. In the proposed scheme, the MSB in the same digit group judges whether to invert the input data or not. Because the proposed scheme does not need to count the number of "1"s in the bit string, thus, it can decide the bit inversion immediately after the input of the write datum. The degree of the correlation in the input datum is important to optimize the RBL charge/discharge energy. As mentioned in Fig. 4.25, the image data with a strong correlation between adjacent pixels should be a good feature to improve the effectiveness of our proposed inversion logic.

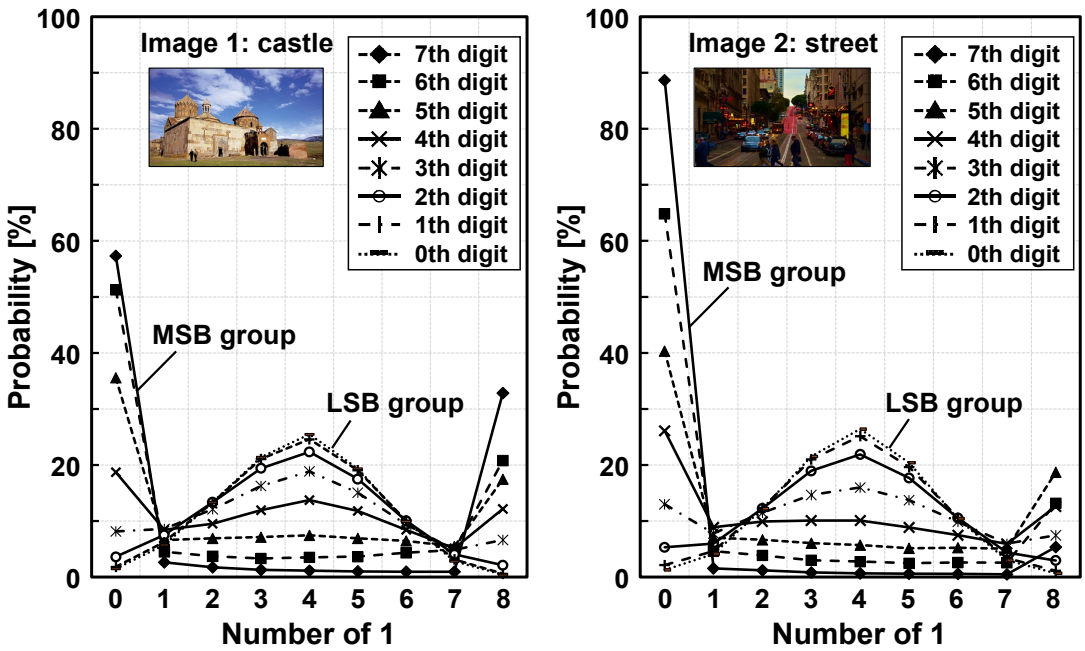


Fig. 4.25 Distributions of the number of "1"s in different digit groups, analyzed with HD size image (Image 1: castle, and Image 2: street).

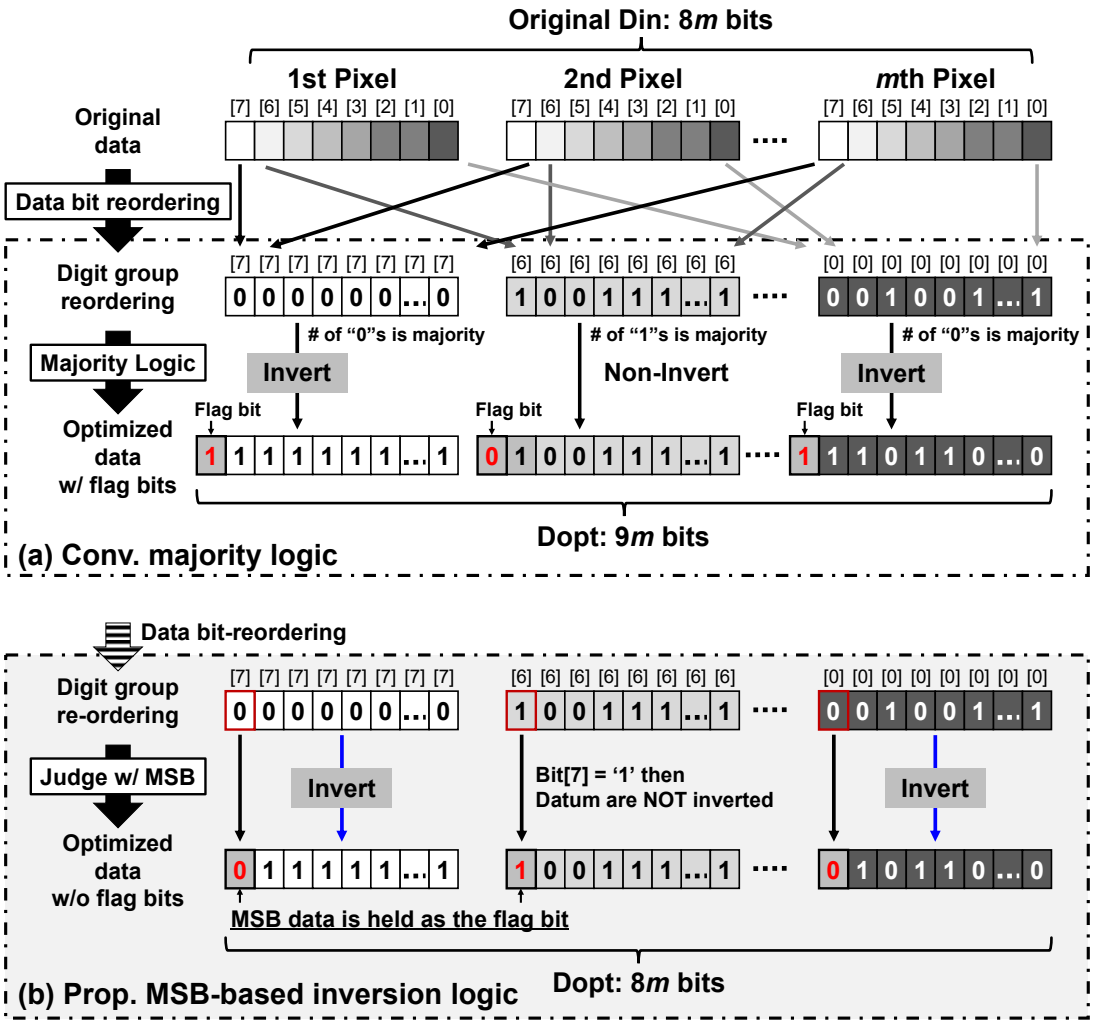


Fig. 4.26 (a) Conventional majority logic, and (b) proposed MSB-based inversion logic: both logic use the data bit reordering.

4.3.3 Parameter optimization

The bit-width parameter m in data bit reordering affects the correlation degree in each digit group. In order to obtain the optimum value of m, the read energy reduction was analyzed statistically with a set of images. Fig. 4.27 illustrates the read energy reduction ratio when the bit-width parameter m is varied. When we set m = 4, the correlation among the same digit groups is maximized. However, the proposed MSB-based inversion logic must hold the MSB value and does not change it, even if a "0"-bit is stored as the MSB. Thus, the bit inversion ratio is slightly decreased in comparison with the case of m = 8. On the other hand, in the case of m = 16, although the impact of the

MSB bits is minimized, it has weaker correlation among digit groups. Herein, the *m* should be optimized by considering the tradeoff between correlation degree and impact of the flag bit; we take *m* = 8.

As illustrated in Fig. 4.28, the read energy reduction is maximized at the MSB digit group that has the strongest correlation; its saving ratio reaches 41.7 % at the 7-th digit. On the other hand, the read energy is slightly improved at the LSB group. The proposed scheme reduces a 14.76 % of the read energy on average although additional flag bits are eliminated. Fig. 4.29 presents the area overhead comparison when *m* = 4, 8, and 16. In the conventional majority logic, the additional columns are inserted every m column in an SRAM array; thus, it conducts a huge amount of area overhead, which is 16.1 % in the case of *m* = 8. The proposed MSB-based inversion logic eliminates this area overhead in the SRAM array. Its area overhead is constantly 1.8 % in peripheral circuitry, regardless of *m*'s value.

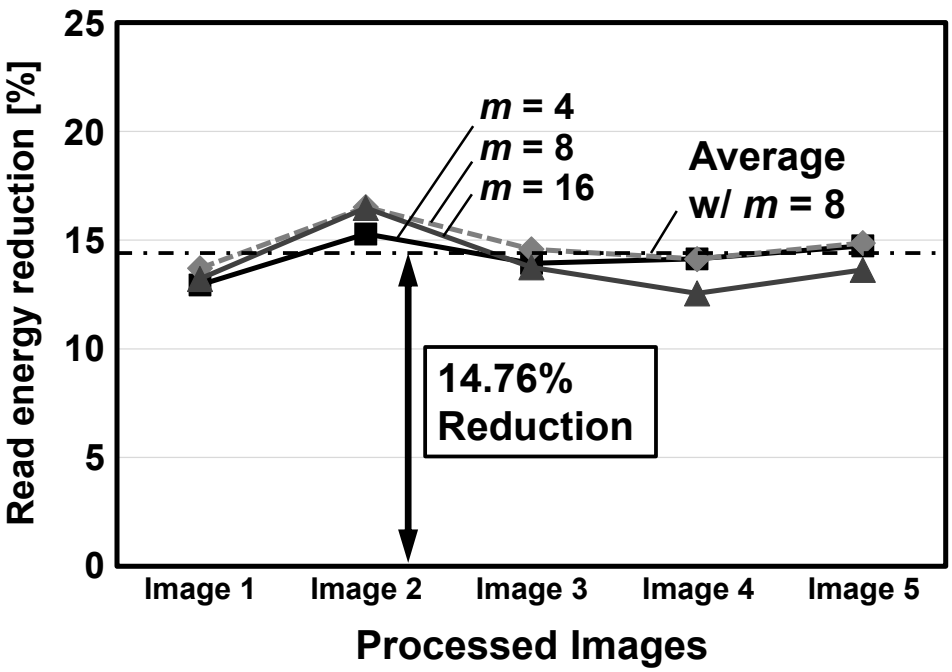


Fig. 4.27 Normalized read energy reduction ratio in the proposed MSB-based bit inversion with data bit reordering: comparison in each image.

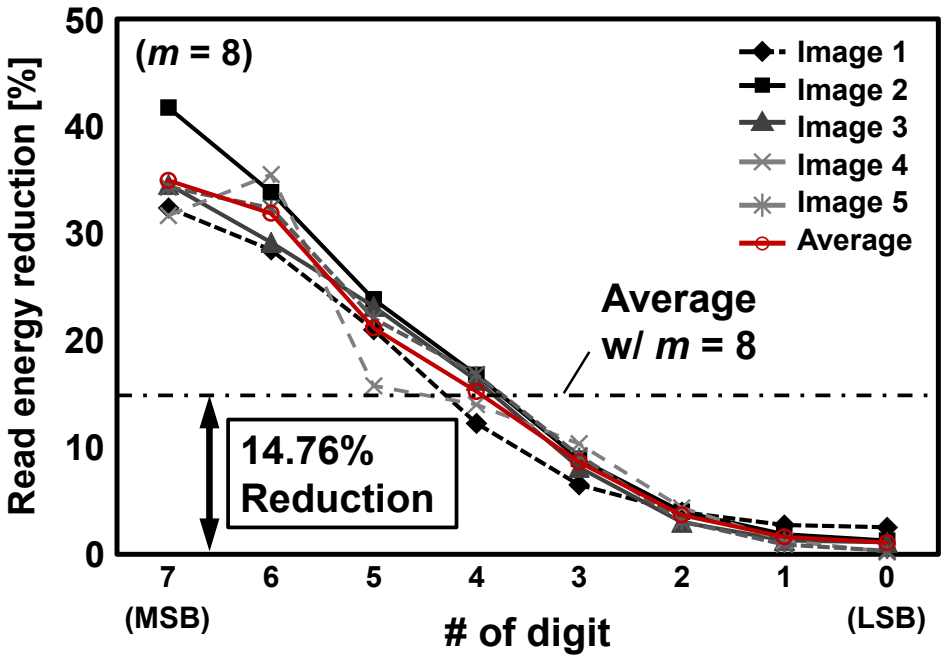


Fig. 4.28 Normalized read energy reduction ratio in the proposed MSB-based bit inversion with data bit reordering: comparison at each digit.

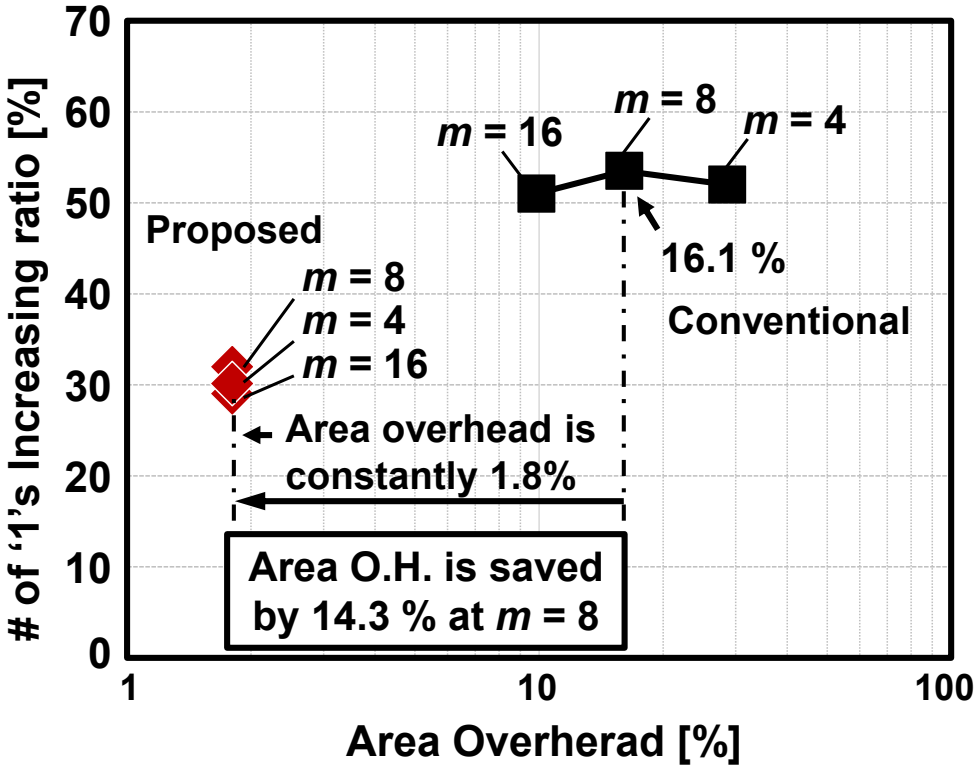


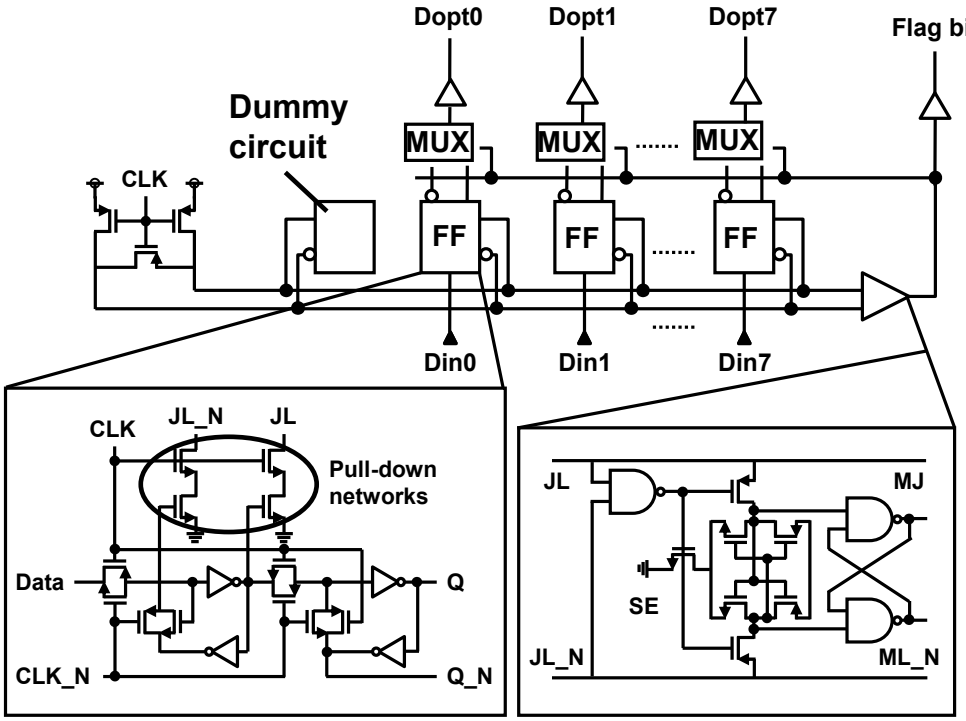
Fig. 4.29 Area overhead comparison between the conventional majority logic and the proposed MSB-based inversion logic in peripheral circuitry.

4.3.4 Circuit design for MSB-based inversion logic

We designed and evaluated dual-port SRAMs with the conventional majority logic and the proposed MSB-based inversion logic. Fig. 4.30 depicts the conventional majority logic circuit schematic and simulated waveforms in write operation at 0.7 V, TT, 25°C. The majority logic is comprised of a precharge circuit, pull-down networks in flip-flops, and a sense amplifier (SA). This SA senses a voltage difference between majority signals, JL and JL_N. When the number of "0"s is majority, pull-down signals and one-dummy signal rapidly sink the voltage of JL_N node to "L" as shown in Fig. 4.31 (a); the MJ node is switched to "L". When the "1"s are in a majority, the voltage of JL node becomes "L", and the MJ node becomes "H" (Fig. 4.31 (b)).

Fig. 4.32 illustrates schematic and simulated waveforms of the proposed MSB-based inversion logic in write operation at 0.7V, TT, 25°C. The proposed logic is comprised of transmission-gate multiplexers (MUX). The input Din0 at the MSB undertakes a role as a flag bit; thus, it signifies whether or not to invert Din1 to Din7. The transistor size of each MUX is denoted at under the schematic, which gate width/Length size are Wp/Lp = 0.64μm/0.03μm for pMOS transistors, and Wn/Ln = 0.32μm/0.03μm for nMOS transistors, respectively. Figs. 4.33(a)-(b) presents the operating waveforms of the inversion logic in the write cycles. Once a datum is input, the bit inversion is immediately determined by Din0. The input Din0 is always hold and not inverted, but stored as a flag bit. Because the proposed scheme does not calculate the majority logic in the bit string, the proposed inversion logic achieves smaller access time than the conventional scheme.

Fig. 4.34 shows a chip layout of the proposed 64-kb SRAM macro configured with 32-kb × 2 banks. The macro size is 243 × 202 μm^2 (= 0.049 mm^2). Each bank consists of 4-kb × 8 subarrays, which are configured with 256 rows and 16 columns. The area size of the 4-kb subarray is 2,214 μm^2 (= 1962.3 μm^2 memory array + 251.5 μm^2 peripheral circuits).



(a) Schematic of the Majority Logic

Fig. 4.30 The conventional majority logic: write circuitry schematics.

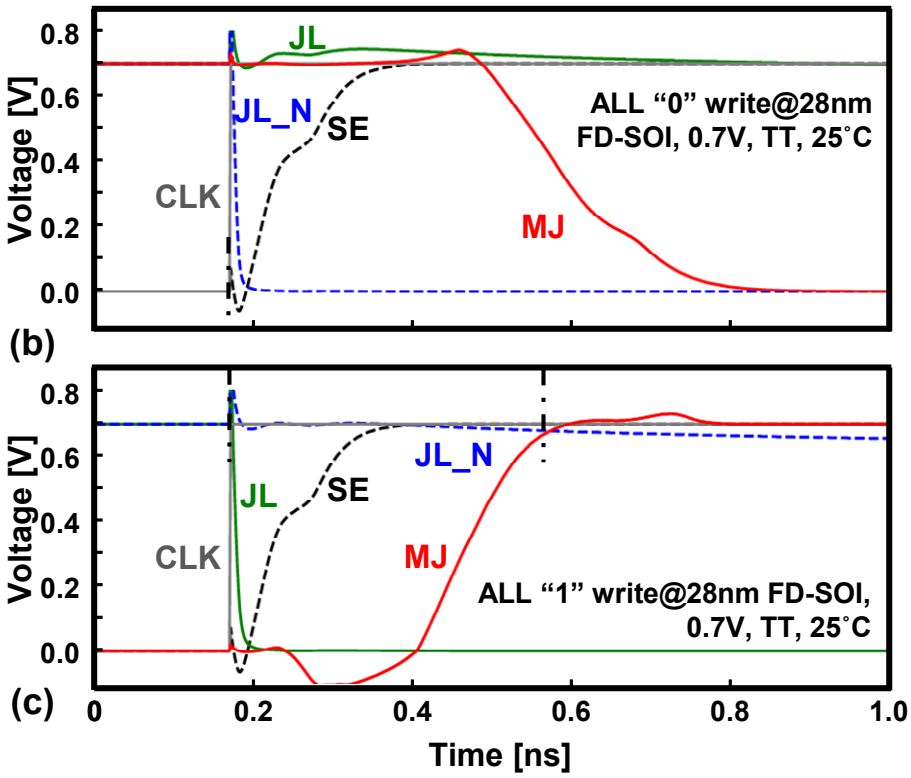
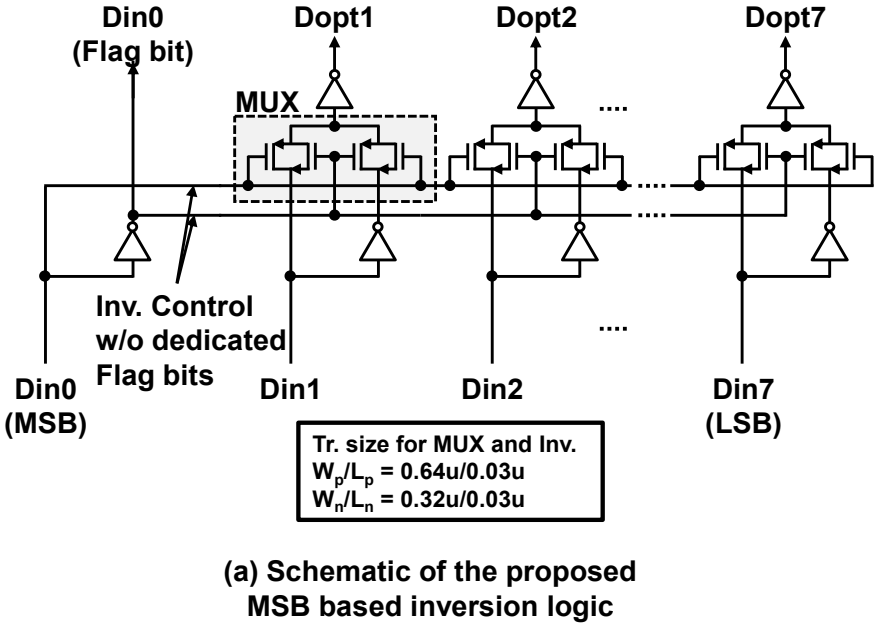


Fig. 4.31 The conventional majority logic: (a) simulated waveforms when "0" is the majority, and (b) waveforms when "1" is the majority in write operation.



(a) Schematic of the proposed MSB based inversion logic

Fig. 4.32 The proposed MSB-based inversion logic: (a) write circuitry schematics.

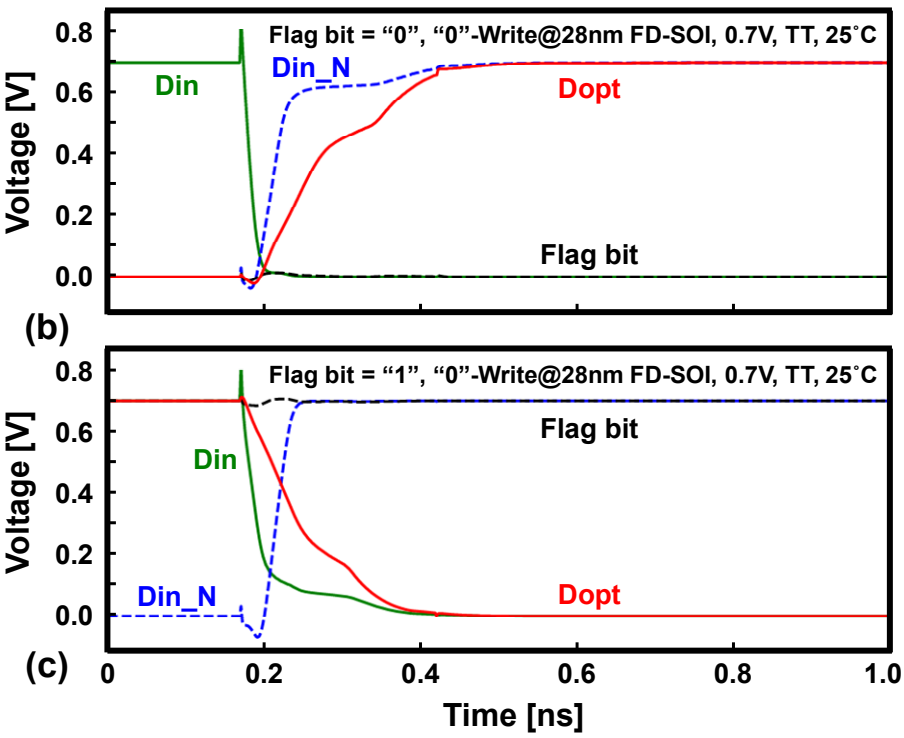


Fig. 4.33 The proposed MSB-based inversion logic: (b) when the flag bit = "0", and (c) when the flag bit = "1".

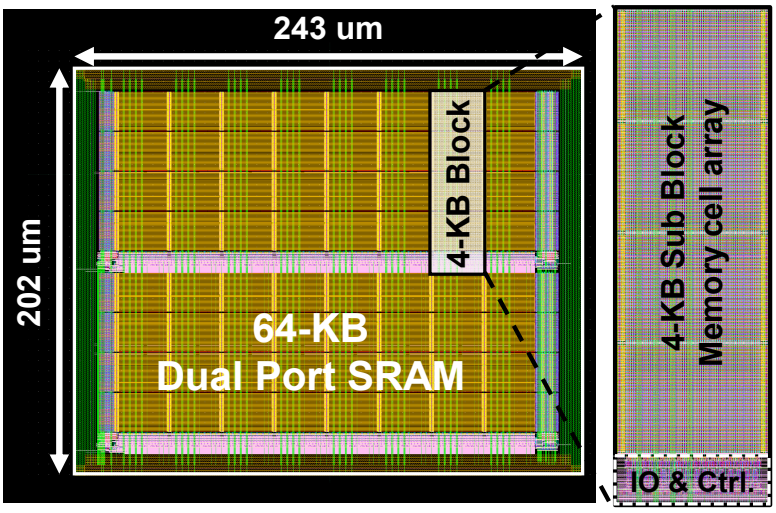


Fig. 4.34 Chip layout design of the proposed SRAM.

4.3.5 Performance evaluation in DL tasks

As presented previous section, the MSB based inversion logic advantageous for image or video sequences. In this section, we carried out further energy reduction tests with the deep learning task; we use convolutional neural networks (CNN) and the ImageNet benchmark [46] for evaluation. In this analysis, the proposed SRAM stores image data such as input sample images, and output activations of convolutional layers. Fig. 4.35 (a) shows the architectural model of VGG-F network taken from the MatConvNet [47], which consists of five CNN layers and three fully connected (FC) layers. Fig. 4.35(b) exhibits a samples of input images and output activations of the CNN layers which is generated by the VGG-F network.

Table 4.4 summarizes the number of dimensions of input image and activations, and bit precision information. We took various bit precision types for energy evaluation. Here, I note the meaning of each notation; single: means FP32, half: FP16, and mini: FP8, those are used for the entire computation consistently. Fig. 4.36 shows the simulated active and leakage energy comparisons between the conventional and the proposed SRAM in read operation in a 28-nm FD-SOI; both cases carried out the data bit reordering before storing data into the SRAM. In the original image, the proposed scheme effectively reduces the RBL charge/discharge energy at the FP32 precision, which value is 38.5 %; this fact insists that the energy reduction is much more effective in the FP32 precision, rather than the INT8 one. The activations of the CNN layer

contain lots of minus value; in the FP32 precision, the correlations among a digit group are deteriorated. However, the proposed scheme constantly reduces the energy from the CNN layer 1 and the layer 5. Fig. 4.37 depicts the read energy reduction ratios in the precision options. The reduction ratio is improved more in the original image with the bit length increased. Although the randomness is diffusing to the CNN layers, the proposed SRAM constantly reduces the read energy. The FP8 has a certain rounding error in decimal; bit correlation is slightly improved than the other precisions in the layers 1-5. This energy comparison demonstrates that the proposed scheme improves the read energy by 13.7%, 14.2%, and 17.3% in the FP8, FP16, and FP32 precisions on average, which respective values are 311.5 fJ/cycle, 314.1 fJ/cycle, and 312.4 fJ/cycle.

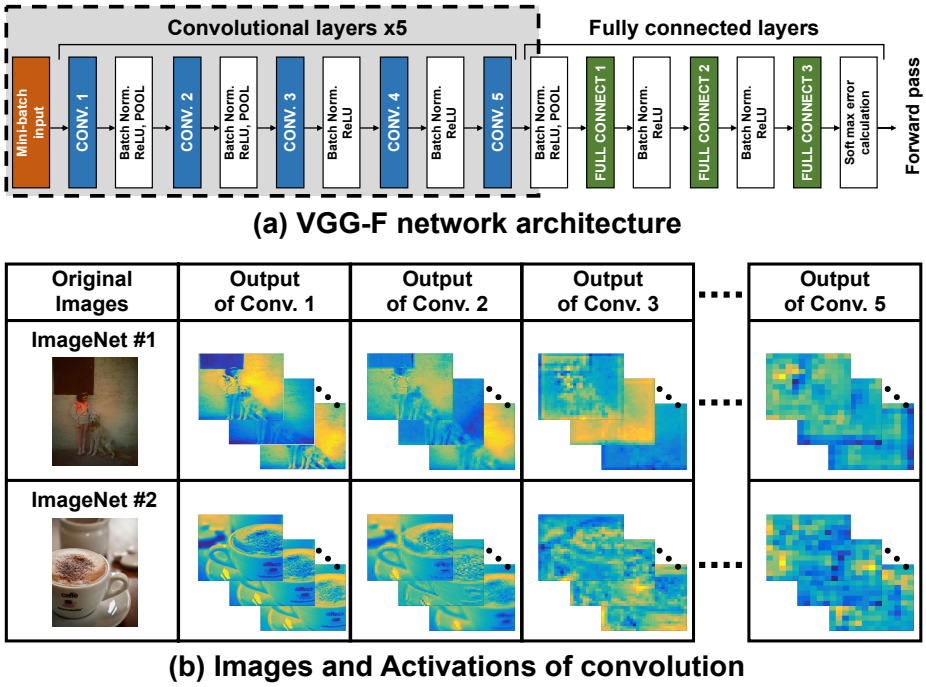


Fig. 4.35 (a) VGG-F with CNNs, and (b) samples of input images and activations of the convolutional layers generated by the VGG-F.

Table4.4 Specifications: input images and activation of convolutions.

Input data	Original Image	Conv. 1 output	Conv. 2 output	Conv. 3 output	Conv. 4 output	Conv. 5 output
Dimension of activations	224 × 224 × 3	54 × 54 × 64	27 × 27 × 256	13 × 13 × 256	13 × 13 × 256	13 × 13 × 256
Bit-precision Option	Floating point (FP32, 16, 8)	Floating point (FP32, 16, 8)	Floating point (FP32, 16, 8)	Floating point (FP32, 16, 8)	Floating point (FP32, 16, 8)	Floating point (FP32, 16, 8)

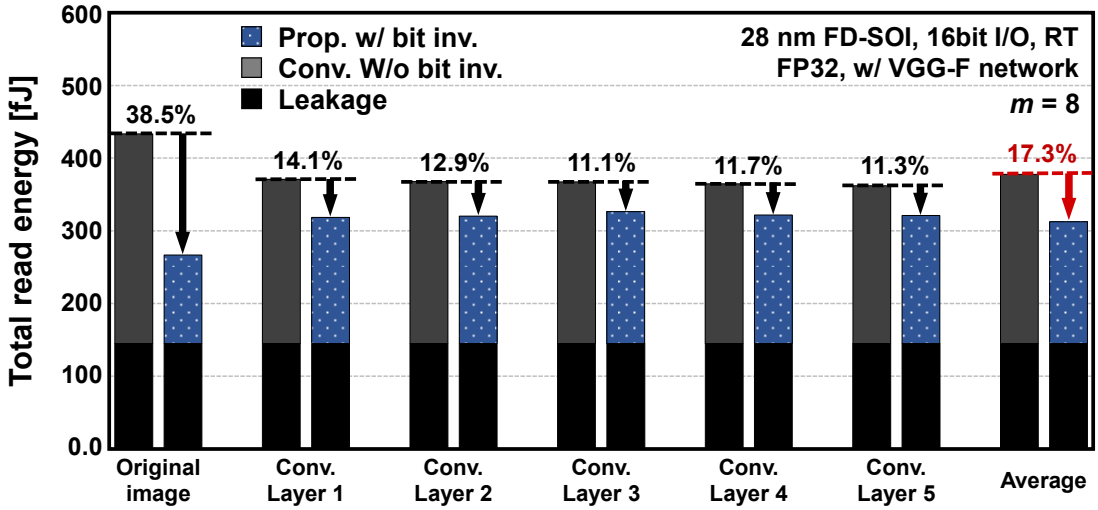


Fig. 4.36 Measured read energy comparison with FP32 in 28nm FD-SOI.

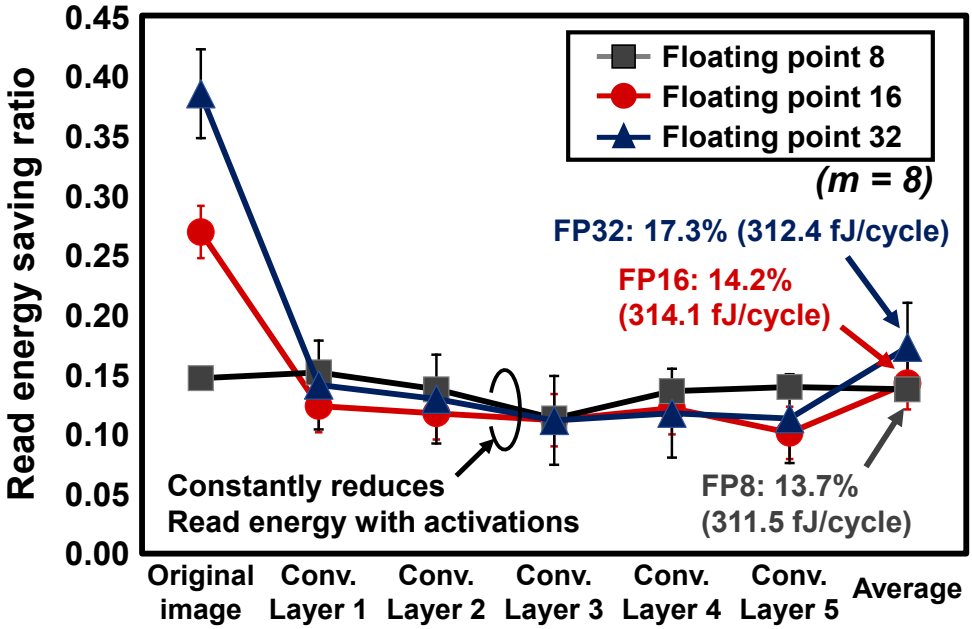


Fig. 4.37 Read energy reduction ratios in FP8, FP16, and FP32 precisions.

4.4 Summary

As described in this chapter, the techniques in leakage reduction and limiting BL swing for low-energy 8T SRAM are presented; 1) a selective source drive (SSD) scheme with dual-port 8T SRAM, and the consecutive memory access 2) an MSB-based inversion logic with dual port SRAM:

1) This study presents a low-energy and low-voltage 64-kb 8T dual-port image memory in the 28-nm FD-SOI process technology. A novel 8T Dual-Port SRAM adopts the selective sourceline drive (SSD) scheme and the consecutive data write technique for improving active energy efficiency at the low voltage. We fabricated a 64-kb 8T dual-port SRAM in the 28-nm FD-SOI process technology; the test chip exhibits 0.48 V operation and an access time of 135 ns. The energy minimum point is at a supply voltage of 0.56 V and an access time of 35 ns, where 265.0 fJ/cycle in write operation and 389.6 fJ/cycle in read operation are achieved; these factors are 30 % and 26 % smaller than those in the 8T dual-port SRAM with the conventional selective sourceline control (SSLC) scheme, respectively.

2) A low-energy 8T dual-port SRAM with a novel MSB-based (most-significant-bit-based) inversion logic for an image processor such a deep-learning processor. Our proposed SRAM is suitable for real-time and low-power image processing, in which data have statistical correlation and data bit reordering are exploited. The proposed MSB-based inversion logic eliminates an additional flag bit in a majority logic; the MSB digit in an input datum judges whether or not to invert the datum. Thus, the area overhead of 16.1 % for the 8-bit conventional majority logic is dramatically saved. The area overhead of the proposed SRAM is merely 1.8 % for the MSB-based inversion logic. We verified that, with the proposed technique, 14.7 % of total energy can be saved in a 28-nm 64-kb FD-SOI SRAM when a set of images is read out. Furthermore, the saving factor is extended to 17.3 % when image processing in the VGG-F convolutional neural network (CNN) is considered, where 312.4 fJ/cycle in the read operation is achieved.

Chapter 5 Co-design of the Distributed Deep Learning Accelerator

This chapter presents a high-scalable, energy-efficient, and low-cost deep learning accelerating system. In this work, the co-designed deep learning system as a novel layer-block-wise-pipeline with pipelined stochastic gradient decent (SGD) algorithm and its hardware architecture is proposed. The study of proposed layer-block-wise pipeline with distributed memory and segmented data bus structure aim memory capacity and bandwidth reduction in highly distributed deep learning.

5.1 Layer Block Wise Pipeline

5.1.1 Overview of distributed deep learning

As described in chapter 2, DNNs have generality with a deeper and larger-scale network, therefore, their error rates of cognition continue to improve. However, computational time become much longer, particularly those for training purpose. For example, AlexNet took 5–6 days to training 90 epochs of 1.2-M ImageNet picture data sets on two NVIDIA GTX580 GPUs [26]. Also, ResNet-200 (ResNet with 200 layers) designed for the ImageNet classification, took 3 weeks for the network training, even with 8 GPGPUs used in parallel computing [48]. The deep learning (DL) with a large training-data set has an issue in its computational time. Therefore, the distributed deep learning with data-parallelization is often adapted to accelerate the large network training. Figure 5.1 depicts the variety of the parallelization models. There are two-concepts of parallelism for a deep learning task to shorten the training time for an enormous network model [49]:

- Data-parallelization has divided dimensions of sample data. Each worker trains with a different data example, but on the same network.
- Model parallelization has divided dimensions of a network (model). Each worker trains a different part of the network (model).

A mini-batch stochastic gradient descent (SGD) algorithm can be faster in error rate convergence than pure-SGD algorithm because a matrix-matrix operation should be

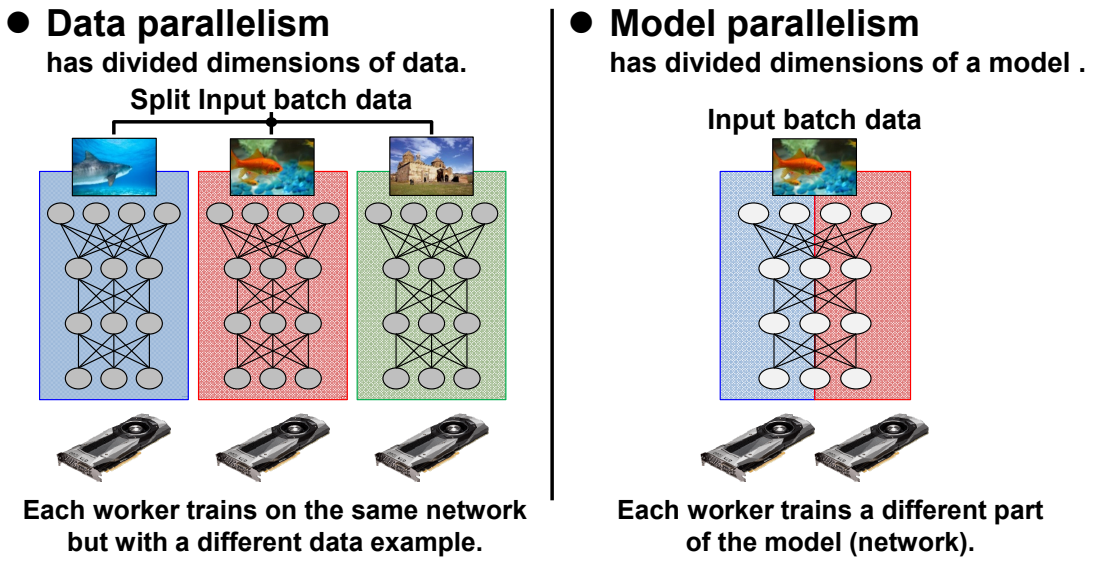


Fig. 5.1 The conceptual models of data parallelism and model parallelism.

better optimized than the matrix-vector one. Multithreaded mini-batch-SGD algorithm is frequently exploited as a data parallelism for additional speeding up of the training. Deploying homogeneous parallel workers and implementing the same software for them are simple.

Each worker has the same network model, but processes a different mini-batch. In other words, a single-network is trained with different mini-batch samples. Each worker updates different part of weight parameters. All workers must unify their own weights. The unified weight parameters usually obtained by averaging their own weights that received from all workers. Then, the unified weights are sent-back to each worker for the upcoming mini-batch step. The weight unification procedure and replication process are applied repeatedly. They invariably consume the memory bandwidth in communication data bus. As the number of parallel workers is increased, memory-bandwidth turns out to be linearly wider [50], [51]. In terms of internal memory capacity, each worker must hold all weight parameters and the activations of a whole network in the multi-threaded mini-batch SGD. The multi-threaded mini-batch SGD tends to be less effective in the convergence than a single threaded one, because its effective mini batch size is scaled by the data parallelism. Therefore, parameter updates per epoch result in a lower number [52, 53, 54]. To reduce the memory capacity and bandwidth, and to maintain scalability of parallelism, the model parallelism is effective. Of course, the model parallelism can be mixed with the data parallelism.

Pipelined back propagation with a distributed memory structure has been studied for decades as a kind of the data parallelism. In 1990s, node parallelism [55] and the layer-based pipeline backpropagation model [56, 57] were proposed for the epoch training. Since 2010, a pipelined DNN algorithm combined with a hidden Marcov-model has been proposed for the speech recognition to shorten the GPGPU training period [58−61]. As described in this paper, we revisit the use of pipelined deep neural network (DNN) for image recognition tasks with the ImageNet dataset, and we propose another model parallelism with layer blocks. The proposed layer-block-wise pipeline with segmented data bus hardware architecture suppresses whole memory capacity and transferred memory bandwidth to maintain the scalability of parallelism. In this study, the layer-block-wise pipeline algorithm with software implementation and its hardware architecture is presented to improve the entire memory capacity and the data communication amount on the communication data bus.

5.1.2 Software design of layer-block-wise-pipeline

Figure 5.2 depicts the proposed layer-block-wise pipeline as a conceptual diagram. In the figure, each pipeline stage with multiple layers has a worker for both forward propagation and backpropagation. A worker takes charge of one or more layers called "layer blocks", shown in square model. The layer-block-wise pipeline is categorized as a kind of model parallelism. Its layer dimensions are divided by m or a smaller integer (where m is the number of layers including a layer for error calculation scheme). The number of pipeline divisions depends on the DNN models. Each worker executes a different task in parallel. A worker keeps a single weight matrix corresponding to its own layer network.

Hereinafter, parameters P, T, and S respectively denote the number of pipeline stages (number of layer blocks), a current mini-batch step, and a current pipeline stage (current layer block). In each pipeline stage, forward propagation is conducted on a stage-by-stage basis. After certain latency, backpropagation is executed with corresponding activations; then weights are updated. Therefore, each pipeline stage processes a different but consecutive mini batch, and respectively propagates its activations and deltas down and up simultaneously to adjacent pipeline stages.

In forward propagation, the $(T-S+1)$-th mini batch is processed at pipeline stage S. Its

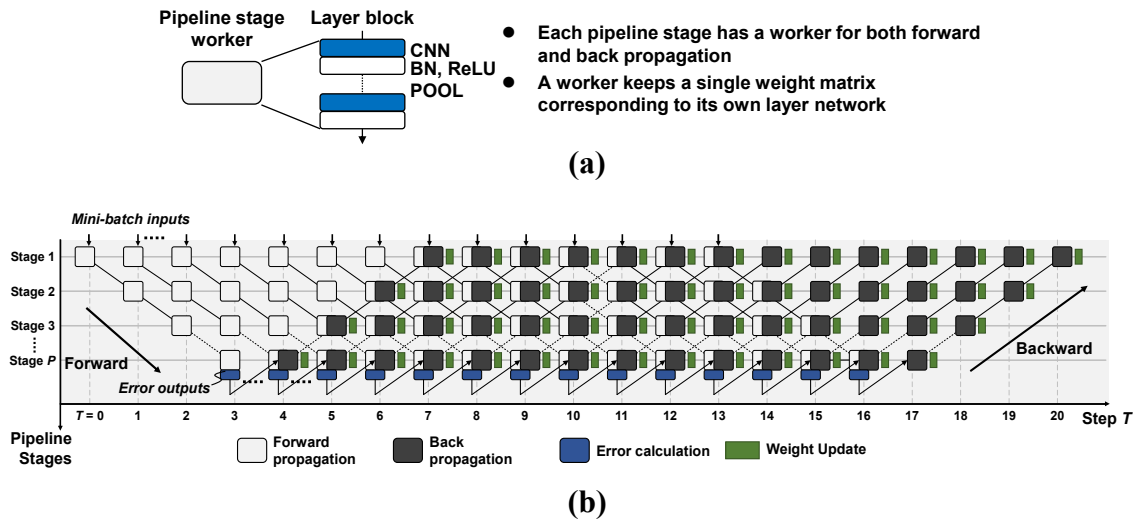


Fig. 5.2 Concept of layer block wise pipeline; (a) Pipeline stage and layer-block, (b) Conceptual data-flow diagram of the proposed layer-block-wise pipeline with weight update latency.

activations are transferred down to the next pipeline S+1. Finally, a mini batch is processed at the last pipeline stage P, where errors are calculated. The errors are going to be backpropagated at the next time step T+1.

In backpropagation, the (T−2P+S)-th mini batch is processed at a pipeline stage S. It is noteworthy that a worker at the pipeline stage S must save 2P−2S+2 datasets of forward activations for the current and upcoming backpropagations. Deltas are calculated with the oldest dataset of activations. Weights are updated with the deltas. The deltas in the shallowest layer in the pipeline stage S are transferred up to the upstream pipeline stage S−1 for a next time step T+1. In this manner, plural mini batches are propagated back and forth simultaneously without waiting for a naive parameter update. The weights are updated with a latency of 2P−2S+1. It can be said that the proposed pipeline has a concept of approximate computing instead of the naive SGD.

To evaluate the accuracy and to verify training convergence in the proposed layer-block-wise pipeline model, we implemented Algorithm 1 with MatConvNet [47]. N signifies the total number of input mini-batch steps (the total number of time steps for input mini batches). The input is mini-batch data. In forward propagation, a vector of activations $Y_{S,\,x}$ for a pipeline stage S is calculated first, where x is a dataset of activations to be saved ($0 \le x \le 2P-2S+2$). After forward propagation is completed, an

error vector is prepared as dY_{P+1} for backpropagation. Then, a vector of delta dY_S and a matrix of delta weights dW_S for a pipeline stage S are calculated. A matrix of weights W_S is updated with dW_S.

To demonstrate the layer-block-wise pipeline, we adopted VGG-F as a network model [46]. VGG-F has five CNN layers and three fully-connected layers, as presented in Fig. 5.3(a), which classifies 1,000 categories. Figs. 5.3(b)-(d) present two-stage, four-stage, and eight-stage pipeline cases.

Algorithm 1 Software implementation of the proposed layer-block-wise pipeline

Input: $\text{MiniBatchInput}_0 \ldots \text{MiniBatchInput}_{N-1}$
Output: $W_1 \ldots W_P$

for $T = 0 \ldots N+2P-2$ **do**
  $Y_{0,\,\mathrm{mod}(T/2P)} = \text{MiniBatchInput}_T$
  for $S = 1 \ldots P$ **do**
    $Y_{S,\,\mathrm{mod}((T-S+1)/(2P-2S+2))} = \text{Forward}(Y_{S-1,\,\mathrm{mod}((T-S+1)/(2P-2S+4))}, W_S)$
  end for
  $dY_{P+1} = \text{Error}(Y_{P,\,\mathrm{mod}((T-P+1)/2)})$
  for $S = P \ldots 1$ **do**
    $[dY_S, dW_S] = \text{Backward}(dY_{S+1}, Y_{S,\,\mathrm{mod}((T-2P+S)/(2P-2S+2))}, W_S)$
    $W_S = \text{Update}(W_S, dW_S)$
  end for
end for

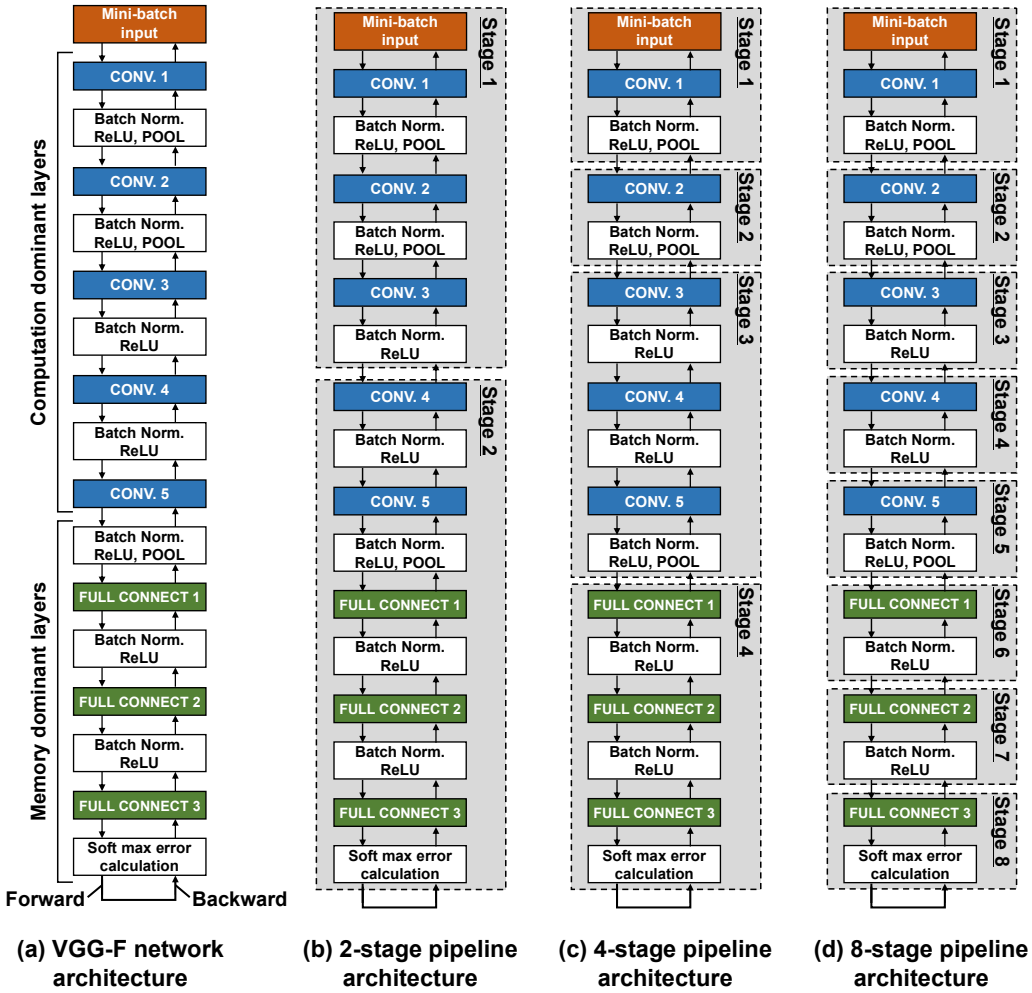


Fig. 5.3 Partitioning variations for VGG-F in the layer-block-wise pipeline; (a) VGG-F network architecture, (b) Partitioning for 2-stage pipeline architecture, (c) Partitioning for 8-stage pipeline architecture.

5.1.3 Hardware model and evaluation

The main purpose of this paper is to reduce memory bandwidth and capacity for scalability of parallelism. Memory performance gives large impacts to speedup. We evaluate the hardware performance using the bus models. The layer-block-wise pipeline potentially reduces weight parameters and memory bandwidth on an I/O data bus. Fig. 5.4 presents a typical multithreaded SGD architecture. In this model, each processing unit has the same network model duplicated for multithreading. The dedicated parameter server for weight update is on a shared I/O data bus to communicate with the processing units. Each processing unit holds weights W and delta weights dW in internal

memory. Actually, delta weights *dW* become 243.4 MB per processing unit in VGG-F. This amount of memory is pushed to and pulled from the parameter server by a DMA controller. An important issue related to a multithreaded architecture is data traffic concentration on the shared I/O data bus. The memory bandwidth comes to 243.4×*n* MB at every mini-batch step (where *n* is the number of workers). The sheared bus brims over with communication among multiple workers, which has restricted system throughput in data parallelism [50]–[54].

Fig. 5.5 depicts a model for the layer-block-wise pipeline with distributed memory and segmented I/O data buses. The proposed architecture divides a network across a layer dimension. Layers are put together to an arbitrary number of blocks. Each worker performs different tasks in parallel. The segmented I/O data bus is used for communication only between two adjacent workers. The bus direction is always fixed to a single side (a sender side or a receiver side). Each worker receives and sends partial activations *Y* in forward propagation; again, each worker receives and sends partial deltas *dY* in the backpropagation. No communication exists on weights *W* and delta weights *dW*. The layer-block-wise pipeline prevents traffic concentration and improves memory bandwidth.

Delay to external data communication depends on a transfer data size. Table 5.1 and Table 5.2 show the respective memory performance comparison between the multithreaded SGD architecture and the proposed layer-block-wise pipeline. In the conventional multithread, the weights W and the delta weights dW increase linearly according to the number of network model duplication (parallelization degree). It is noteworthy that, in the layer block wise pipeline, memory capacity and memory bandwidth for activations and deltas are scaled up linearly with a batch size, although they are reasonable values at a typical batch size of 32 (BS = 32). As explained previously, the multithreaded SGD requires a memory bandwidth of 1.21 GB (four unicasts and a broadcast) to unify the weights when a parallelization degree is four. The memory bandwidth per batch against the BS in the proposed pipeline increases linearly with increasing mini-batch size. The proposed pipeline has different values of bandwidth on forward and backward processes. When the input batch size is 32, the memory bandwidth for both forward and back propagation are, respectively, 36.3 MB and 17.0 MB.

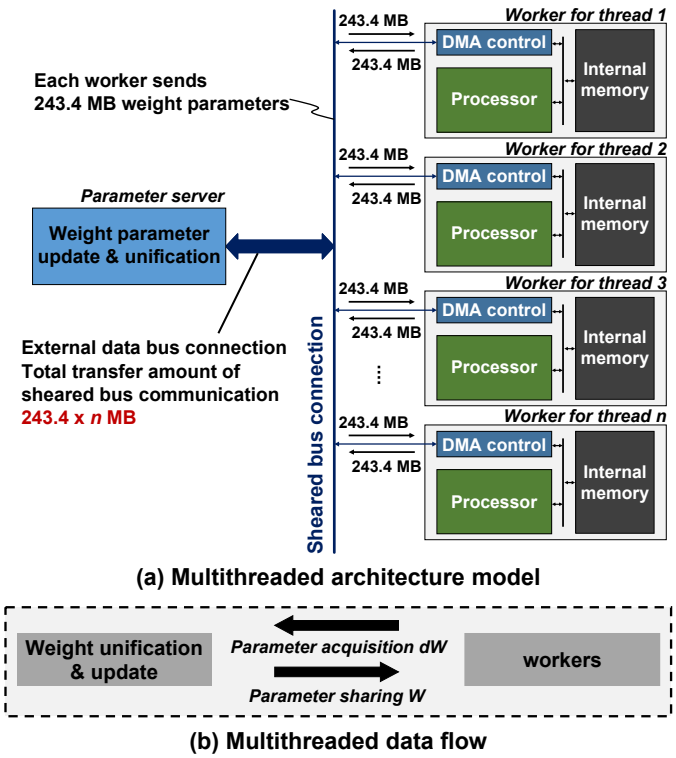


Fig. 5.4 (a) Architectural model of shared-bus multithreading and (b) its data flow.

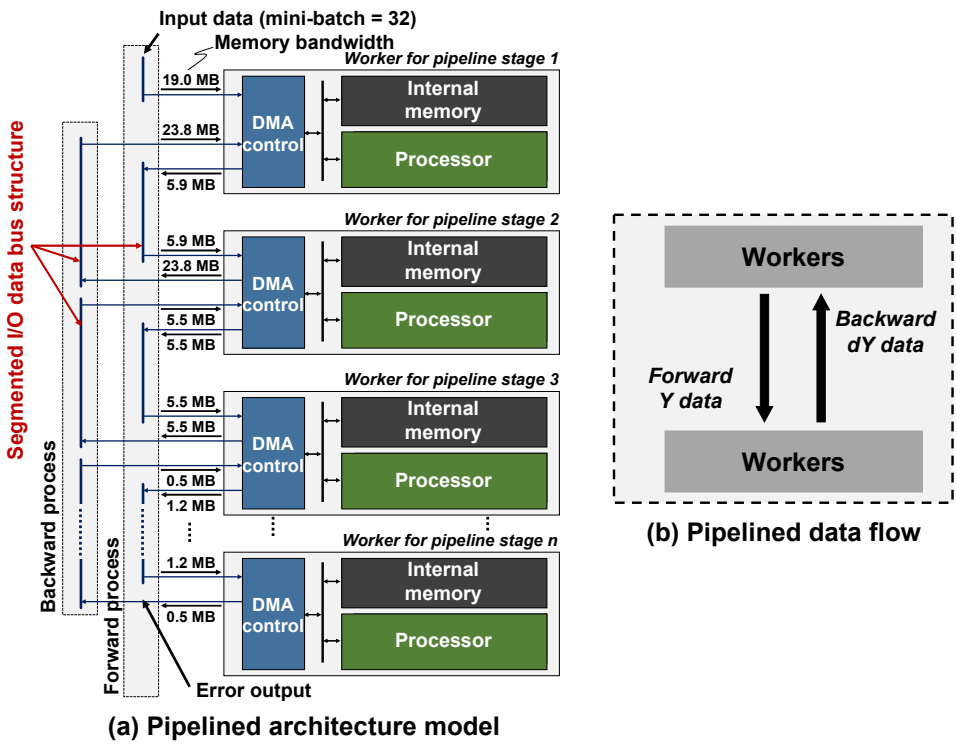


Fig. 5.5 (a) Architectural model of the layer-block-wise pipeline and (b) its data flow.

Table5.1 Memory capacity and memory bandwidth in conventional multithreads.

4-degree of multithreads	Internal memory capacity						Memory bandwidth				
	Weight parameter memory [MB]		I/O data memory [MB] BS = 1		I/O data memory [MB] BS = 32		Status	Transfer data amount [MB] BS = 1		Transfer data amount [MB] BS = 32	
	W	*dW*	*Y*	*dY*	*Y*	*dY*		*W*	*dW*	*W*	*dW*
Thread 1	243.45	243.45	7.19	7.19	230.23	230.23	Receive / Send		243.45		243.45
Thread 2	243.45	243.45	7.19	7.19	230.23	230.23	Receive / Send	243.45	243.45	243.45	243.45
Thread 3	243.45	243.45	7.19	7.19	230.23	230.23	Receive / Send	(broadcast)	243.45	(broadcast)	243.45
Thread 4	243.45	243.45	7.19	7.19	230.23	230.23	Receive / Send		243.45		243.45
Sub total	973.80	973.80	28.76	28.76	920.92	920.92	Total	243.45	973.80	243.45	973.80
Total	2005.12				3789.44						

Table5.2 Memory capacity and memory bandwidth in proposed pipeline.

4-stage pipeline	Internal memory capacity						Memory bandwidth					
	Weight parameter memory [MB]		I/O data memory [MB] BS = 1		I/O data memory [MB] BS = 32		Forward process			Backward process		
							Status	Transfer data amount [MB/batch] BS = 1	Transfer data amount [MB/batch] BS = 32	Status	Transfer data amount [MB/batch] BS = 1	Transfer data amount [MB/batch] BS = 32
	W	*dW*	*Y*	*dY*	*Y*	*dY*						
Stage 1	0.09	0.09	21.19	2.42	678.08	77.44	Receive	0.60	19.26	Receive	0.19	5.98
							Send	0.19	5.98	Send	-	-
Stage 2	4.01	4.01	14.65	2.93	468.80	93.76	Receive	0.19	5.98	Receive	0.17	5.53
							Send	0.17	5.53	Send	0.19	5.98
Stage 3	155.79	155.79	3.37	1.12	107.84	35.84	Receive	0.17	5.53	Receive	0.17	5.53
							Send	0.17	5.53	Send	0.17	5.53
Stage 4	83.56	83.56	0.07	0.07	2.24	2.24	Receive	0.17	5.53	Receive	-	-
							Send	-	-	Send	0.17	5.53
Sub total	243.45	243.45	39.28	6.54	1,256.96	209.28	Receive total	1.13	36.30	Receive total	0.53	17.04
Total	532.72				1,953.14		Send total	0.53	17.04	Send total	0.36	17.04

Figure 5.6 portrays memory bandwidth trends in the multithreaded SGD. A memory bandwidth of 974.0 MB is required overall for send and receive processes to unify the weight parameters when a parallelization degree is four. Fig. 5.7 presents memory bandwidth trends against the batch size in the layer-block-wise pipeline. The memory bandwidth per batch increases linearly with increasing mini-batch size. It is noteworthy that layer-block-wise pipeline has different values of memory bandwidth on forward and backward processes. As described above, our target batch size is 32, in which case the memory bandwidth both forward propagation and backward propagation are, respectively, 36.3 MB and 17.0 MB. The total memory bandwidth for the both directions does not exceed 100.0 MB.

.

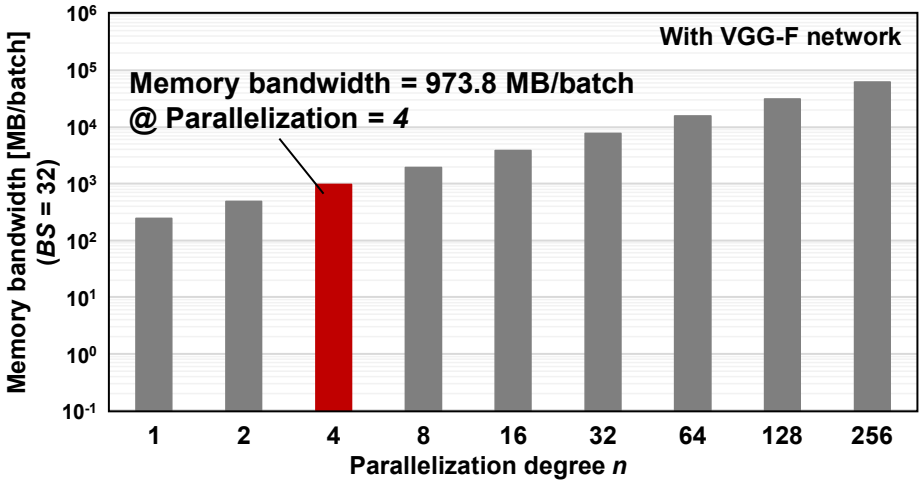


Fig. 5.6 Memory bandwidth trends against the parallelization degree in multithreading.

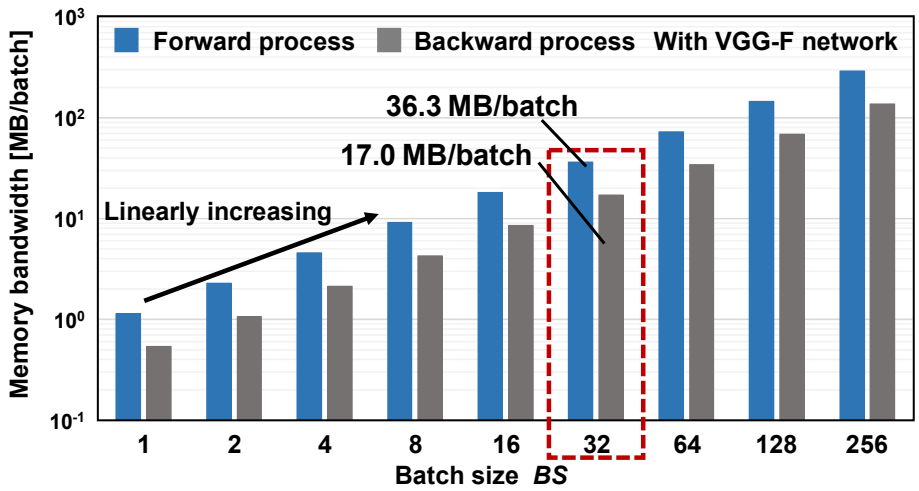


Fig. 5.7 Memory bandwidth trends against the batch size in the layer-block-wise pipeline.

5.1.4 Performance evaluation

Figs. 5.8(a)-(b) describe training convergence comparisons between the naive SGD and the proposed layer-block-wise pipeline. The convergence is the time when the 20 epoch training. A 1.28 M ImageNet dataset is used for training VGGF. We observed the top-1 accuracy is slightly lowered with momentum SGD at four-eight stage pipeline due to the weight update delays in each layer, as shown in Fig. 5.8(a). Thus, the update value which is scaled by learning rate (LR) must have variations with different number of delays in each layer. To align the differences of update value, Adagrad [62] is adopted for LR adaptation, which simply shown as below equations (Eq. 14, and Eq. 15).

$$\tau_{(t)} = \tau_{(t-1)} + \left(dW_{(t)}\right)^2. \tag{14}$$

$$W_{(t+1)} = W_{(t)} - \frac{\alpha}{\sqrt{\tau_{(t)}} + \epsilon} dW_{(t)}. \tag{15}$$

The Adagrad changes the LRs by τ (t) which accumulates square value of deltas: dW(t). Here, we took valuables; initial τ (t) 0.0, stabilization coefficient Ep= 0.1, and LR coefficient a= 0.01. Fig. 5.8(b) again shows training convergence comparisons between multithread and the pipeline with Adagrad.

Accuracy convergence is compensated by LR adaptation. The proposed pipeline is 2.0 and 4.2 times faster in the 2-stage and 4-stage pipeline, respectively. With eight pipeline stages, the acceleration factor is improved to 8.1.

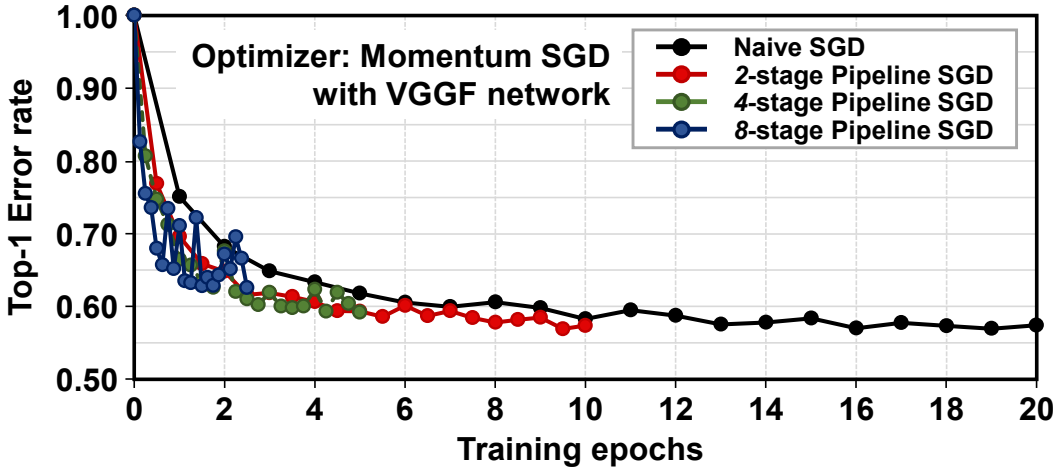
Figure 5.9 illustrates the total execution time T_{total} comparisons and breakdowns, which are evaluated with Caffe [63], and multiple GPGPUs linked by PCIe Gen3. For the breakdown mapping, T_{total} is calculated by summation of the parameters of a communication time T_{com}, and a computational time T_{exe} , as Eq. (16):

$$T_{total} = T_{com} + T_{exe}. \tag{16}$$

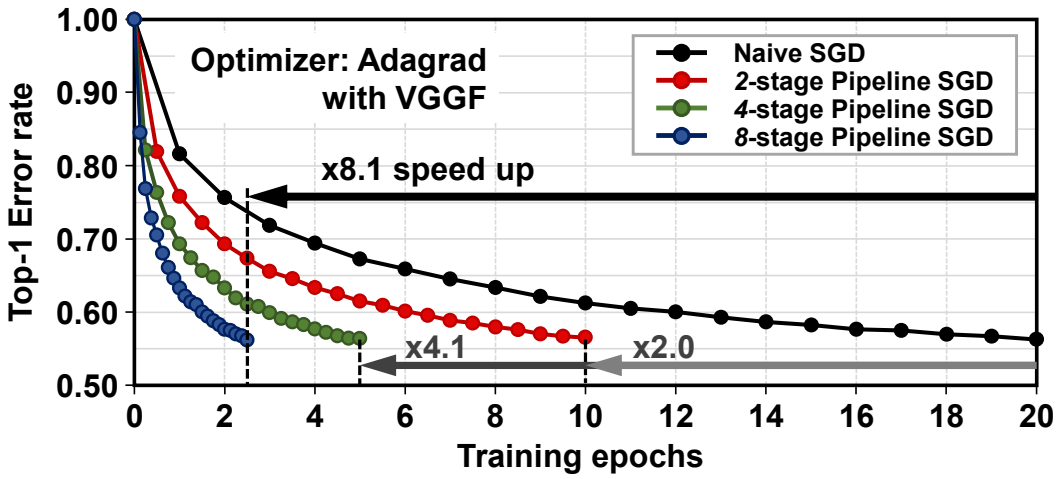
In the conventional multithread, T_{exe} in an epoch is divided by the number of workers: n. On the other hand, T_{exe} in the proposed pipeline is determined by the longest computational time among the divided stages; it occurs certain overhead on the computation time. We divided each network model to minimize the overhead of the worst-case computations in each stage.

In the conventional scheme, $T_{com(conv)}$ in an epoch can be modeled by a weight memory capacity Cap_Y, bus speed S_{bus}, the number of iterations l_{iter}, and the number of workers n, as expressed equation (17). In the equations, n and 1 are correspond to uploading unicasts (= n) and downloading broadcast (= 1) for weight unification and weight update (all reduction), respectively. In our PCIe linking, this model is better fit with our multithread system than a tournament weight unification model. In our

environment, the S_{bus} value settled as much as 32 GB/sec. Because of the non-parallelized part in a training function such as parameter unification, it seems that



(a)



(b)

Fig. 5.8 Training convergence comparison for parallelization degrees of 1, 2, 4, and 8 with (a) Momentum SGD, (b) SGD using the Adagrad LR adaptation.

the $T_{exe(conv)}$ has certain penalty time at higher parallelization degree. Thus, the acceleration factor in the $T_{exe(conv)}$ according to the parallelization degree is gradually decreased. In the proposed layer-block-wise pipeline, the $T_{com(prop)}$ can be modeled simply with the worst-case partial activations Cap_Y, and bus speed S_{bus} as shown in equations (18).

$$T_{com\,(conv\,)} = \frac{Cap_W * l_{iter} * (n+1)}{S_{bus}}. \tag{17}$$

$$T_{com\,(prop\,)} = \frac{Cap_Y}{S_{bus}}. \tag{18}$$

The value of Cap_Y depends on the partition scheme in each network model. The comparison results show that the proposed pipeline exhibited an improvement in T_{total} at the four-parallelization degree, which acceleration factors are VGGF: 1.76, VGG16: 1.23, Resnet18: 1.01, and Resnet50: 1.07, respectively.

Fig. 5.10 presents T_{com} reduction ratio in each network model when the parallelization degrees are 2, 4, and 8. The proposed pipeline is more beneficial than the conventional multithread for the T_{com} reduction; the higher T_{com} reduction ratio is achieved in VGG-F and VGG-16 since weight memory is dominant than the activations. The T_{com} reduction ratio is VGG-F: 93.1 % and VGG-16: 83.5 % at the eight-stage pipeline. On the other hand, in the Resnet18 or 50, the partial activation memory to communicate for pipeline is increased. In this case, the T_{com} reduction ratio is degraded, which value is Resnet18: 51.4 % and Resnet50: 55.4 %. The proposed pipeline, however, maintains over 50 % of T_{com} reduction ratio at the eight-stage pipeline. Fig. 5.11 presents the relationship between acceleration factors and the memory capacity. In multithreading, the memory capacity increases linearly with the parallelization degree. In the proposed pipeline, only activations and corresponding deltas are increased; thus, it has less memory than the multithread for the same degree of parallelization. The layer-block-wise pipeline has 25.4 % less memory in Resnet18 at eight-stage pipeline. In the VGGF, its value is extended to 52.1 % at eight-stage pipeline, with better acceleration performance per unit of memory capacity: 3.61 GB for pipeline and 7.56 GB for multithread.

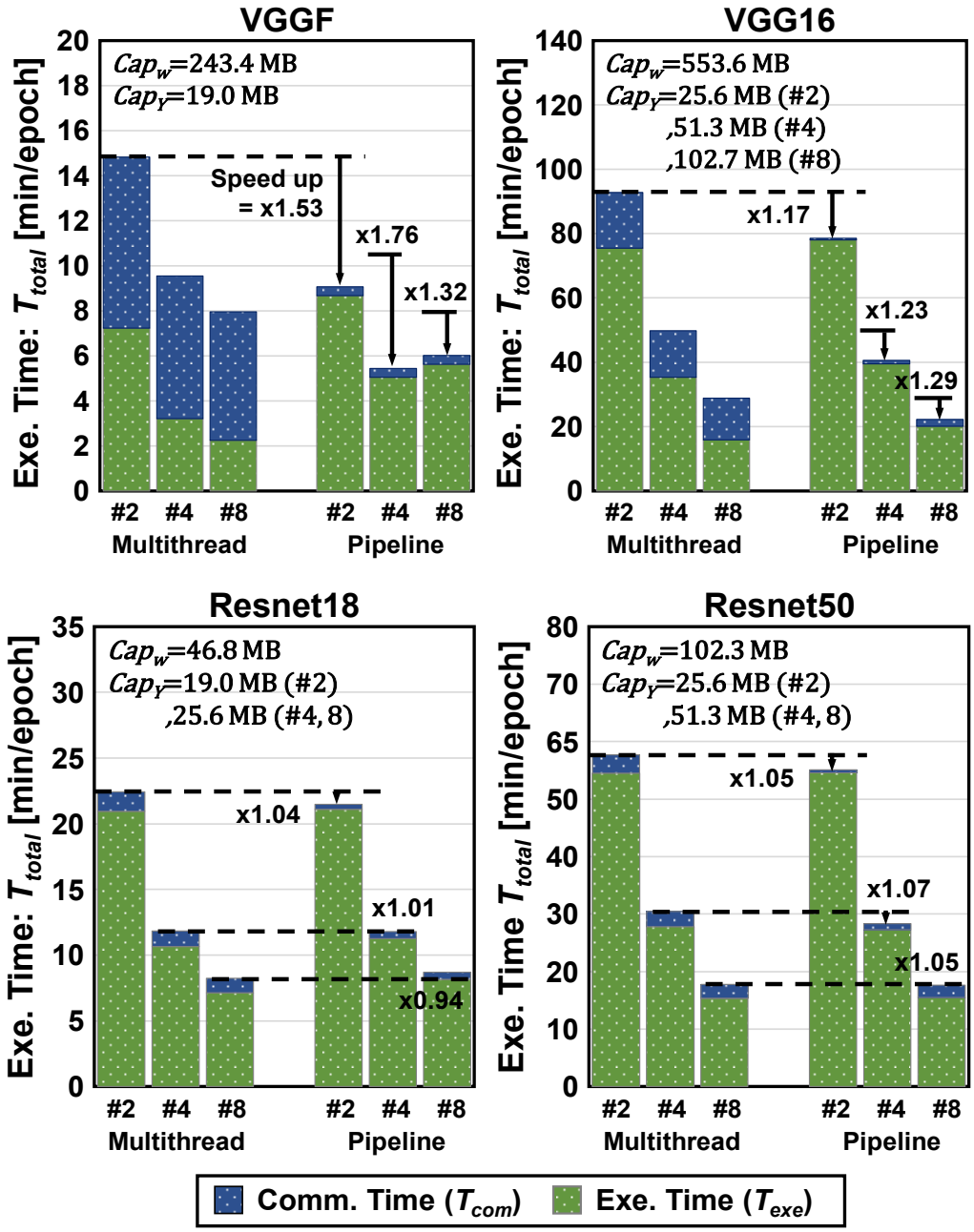


Fig. 5.9 Total execution time T_{total} comparisons and breakdown mapping for each networks between the conventional multithread and the proposed pipeline.

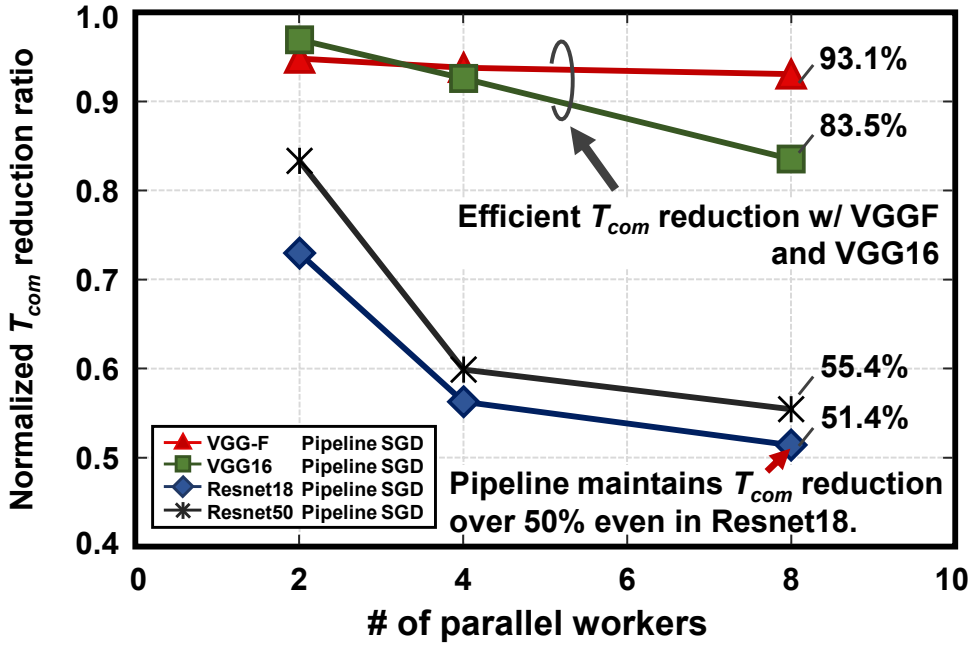


Fig. 5.10 Normalized Tcom reduction ratio comparisons between the conventional multithread and the proposed layer-block-wise pipeline in various network models with number of parallel workers 2, 4, and 8.

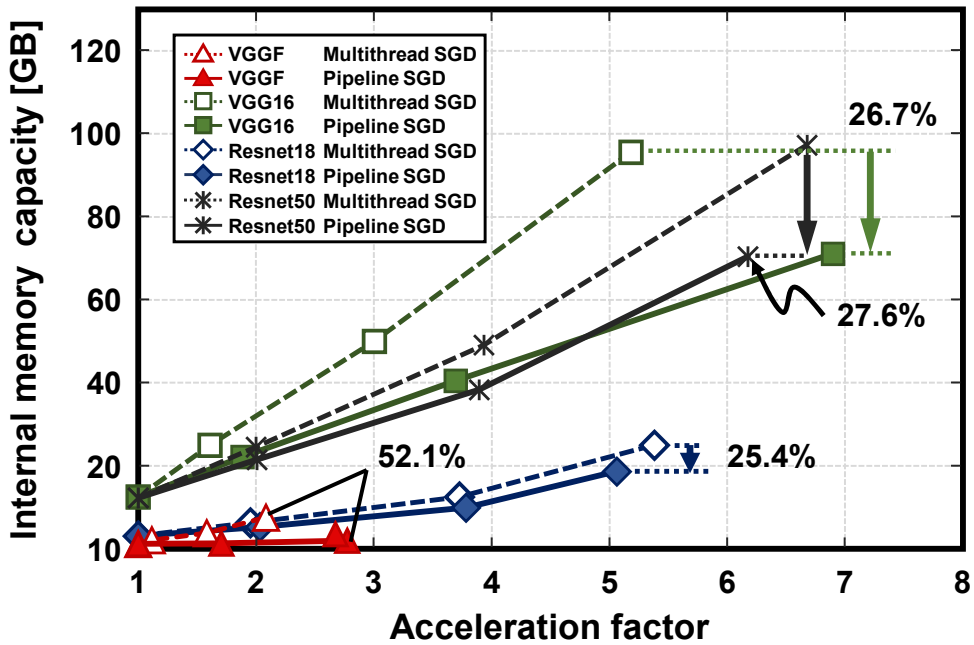


Fig. 5.11 Relationship between the internal memory capacity and the acceleration factor for the conventional multithread and the proposed pipeline.

5.2 Summary and Discussion

As described in this chapter, we proposed a layer-block-wise pipeline algorism and hardware architecture as memory bandwidth and capacity reduction techniques for scalability of parallelism with distributed deep learning:

1) We described a pipelined stochastic gradient descent (SGD) algorithm and its hardware architecture with a memory distributed structure. In the proposed architecture, a pipeline stage takes charge of multiple layers: a "layer block." The layer-block-wise pipeline has much less weight parameters for network training than conventional multithreading because weight memory is distributed to workers assigned to pipeline stages. The memory capacity of 1.95 GB for the four-stage proposed pipeline is about half of the 3.79 GB for multithreading when a batch size is 32 in VGG-F network model. Unlike multithreaded data parallelism, no parameter server for weight update or shared I/O data bus is necessary. Therefore, the memory bandwidth is drastically reduced. The proposed four-stage pipeline only needs memory bandwidths of 36.3 MB and 17.0 MB per batch, respectively, for forward propagation and backpropagation processes, whereas four-thread multithreading requires a bandwidth of 1.21 GB overall for send and receive processes to unify its weight parameters. At the parallelization degree of four, the proposed pipeline still maintaining training convergence by a factor of 1.76, compared with the conventional multithreaded architecture although the memory capacity and the memory bandwidth are decreased.

Chapter 6 **Conclusion**

This dissertation presents low-power and low-energy memory techniques for real-time and low-energy image recognition processors.

In Chapter 2, the intrinsic issues of the image memory and the image recognition processor are introduced as follows:

- Increased active energy
- Energy efficiency degradation by Leakage
- Increased memory bandwidth and capacity in distributed architecture

In this study, the solutions to these issues are presented in Chapter 3 to Chapter 5:

1) Low-energy multiple read-port 8T three-port SRAM design (Chapter 3)
2) Bitline swing and leakage reduction for 8T dual-port SRAM(Chapter 4)
3) Memory and bandwidth reduction in deep learning (Chapter 5)

In Chapter 3, low-power one-write and two-read (1W/2R) 8T three-port SRAM design is proposed. The proposed 8T three-port SRAM accommodates eight-transistor bit cells comprising one-write/two-read ports and a majority logic circuit to save active energy. We fabricated a 64-kb 8T three-port SRAM using 28-nm FD-SOI process technology. The test chip operates at a supply voltage of 0.46 V and access time of 140 ns. The minimum energy point is a supply voltage of 0.54 V and an access time of 55 ns (= 18.2 MHz), at which 484 fJ/cycle in a write operation and 650 fJ/cycle in a read operation are achieved assisted by majority logic. These factors are 69% and 47% smaller than those in a conventional 6T SRAM using the 28-nm FD-SOI process technology. Furthermore, the energy consumed on the proposed SRAM is saved by 290 μW, which signifies 24% energy reduction in total over the conventional H.264 motion estimation image processor.

Chapter 4 introduced the techniques in leakage reduction and limiting BL swing for low-energy 8T SRAM; 1) a selective source drive (SSD) scheme with dual-port 8T SRAM, and the consecutive memory access 2) a MSB-based inversion logic with dual port SRAM.

1) This work presented a low-energy and low-voltage 64-kb 8T dual-port image memory in a 28-nm FD-SOI process technology. A novel 8T Dual-Port SRAM adopts the selective sourceline drive (SSD) scheme and the consecutive data write technique

for improving active energy efficiency at the low voltage. We fabricated a 64-Kb 8T dual-port SRAM in the 28-nm FD-SOI process technology. The 8T SRAM cell size is $0.291 \times 1.457\ \mu m^2$. The test chip exhibits 0.48 V operation at access time of 135 ns. The energy minimum point is at a supply voltage of 0.56 V and an access time of 35 ns, where 265.0 fJ/cycle in write operations and 389.6 fJ/cycle in read operations are achieved. These factors are, respectively, 30% and 26% smaller than those of the 8T dual-port SRAM with the conventional scheme.

2) This work presented low-energy 8T dual-port SRAM with a novel MSB-based (most-significant-bit-based) inversion logic for an image processor such a deep-learning processor. Our proposed SRAM is suitable for real-time and low-power image processing, in which data have statistical correlation and data bit reordering are exploited. The proposed MSB-based inversion logic eliminates an additional flag bit in a majority logic; the MSB digit in an input datum judges whether or not to invert the datum. Thus, the area overhead of 16.1 % for the 8-bit conventional majority logic is dramatically saved. The area overhead of the proposed SRAM is merely 1.8 % for the MSB-based inversion logic. We verified that, with the proposed technique, 14.7 % of total energy can be saved in a 28-nm 64-kb FD-SOI SRAM when a set of images are read out. Furthermore, the saving factor is extended to 17.3 % when image processing in the VGG-F convolutional neural network (CNN) is considered, where 312.4 fJ/cycle in the read operation is achieved.

In Chapter 5, we presented memory bandwidth and capacity reduction techniques for scalability of parallelism with distributed deep learning; 1) a layer-block-wise pipeline stochastic gradient decent (SGD) algorithm and its hardware architecture for deep learning processors.

1) This work presented a pipelined stochastic gradient descent (SGD) algorithm and its hardware architecture with a memory distributed structure. In the proposed architecture, a pipeline stage takes charge of multiple layers: a “layer block.” The layer-block-wise pipeline has much less weight parameters for network training than conventional multithreading because weight memory is distributed to workers assigned to pipeline stages. The memory capacity of 1.95 GB for the four-stage proposed pipeline is about half of the 3.79 GB for multithreading when a batch size is 32 in VGG-F network model. Unlike multithreaded data parallelism, no parameter server for weight

update or shared I/O data bus is necessary. Therefore, the memory bandwidth is drastically reduced. The proposed four-stage pipeline only needs memory bandwidths of 36.3 MB and 17.0 MB per batch, respectively, for forward propagation and backpropagation processes, whereas four-thread multithreading requires a bandwidth of 1.21 GB overall for send and receive processes to unify its weight parameters. At the parallelization degree of four, the proposed pipeline still maintaining training convergence by a factor of 1.76, compared with the conventional multithreaded architecture although the memory capacity and the memory bandwidth are decreased.

Finally, the conclusion of this study is presented in this chapter. This thesis presents the low-energy and low-cost memory architecture for high-speed and low-energy image recognition application overlooking whole memory architecture. The work contributes to achieve an energy-efficient SRAM design for advanced technology and development of energy-efficient and high speed image processing flame work with higher scalability.

References

[1] J. Lim, N. B. Lakshminarayana, H. Kim, W. Song, S. Yalamanchili, W. Sung, "Power Modeling for GPU Architecture using McPAT," *ACM Trans. Des. Automat. Electron. Syst.*, vol. 19, no. 3, June 2014, Art, no. 26.

[2] "International Technology Roadmap for Semiconductor 2011 Edition System Drivers." [Online]. Available: http://www.itrs.net/.

[3] J. P. Kulkarni, J. Keane, K. H. Koo, S. Nalam, Z. Guo, E. Karl, and K. Zhang, "5.6Mb/mm^2 1R1W 8T SRAM arrays operating down to 560 mV utilizing small-signal sensing with charge sheared bitline and asymmetric sense amplifier in 14 nm FinFET CMOS technology" *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 229–239, Jan. 2017.

[4] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, Jan. 2017.

[5] S. W. Keckler, W. J. Dally, B. Khailany, M. Garland, and D. Glasco, "GPUs AND THE FUTURE OF PARALLEL COMPUTING" *IEEE Micro*, vol. 31, no. 5, pp. 7–17, Sep./Oct. 2011.

[6] B.-C. C. Lai, H. K. Kuo, J.-Y. Jou, "A Cache Hierarchy Aware Thread Mapping Methodology for GPGPUs" *IEEE Trans. Comput.*, vol. 64, no. 4, pp. 884–898, Apr. 2015.

[7] Y. Murachi, J. Miyakoshi, M. Hamamoto, T. Iinuma, T. Ishihara, F. Yin, J. Lee, H. Kawaguchi, and M. Yoshimoto, "A Sub 100uW H.264 MP@L4.1 Integer-Pel Motion Estimation Processor Core for MBAFF Encoding with Reconfigurable Ring-Connected Systolic Array and Segmentation-Free, Rectangle-Access Search-Window Buffer," IEICE Trans. Electron., vol. E91-C, no. 4, pp. 465–478, Apr. 2018.

[8] K. Nii, M. Yabuuchi, Y. Yokoyama, Y. Ishii, T. Okagaki, M. Morimoto, Y. Tsukamoto, K. Tanaka, M. Tanaka, S. Tanaka, "2RW dual-port SRAM design challenges in advanced technology nodes" in *IEDM Dig. Tech. Papers*, Dec. 2015, pp. 11.1.1–11.1.4.

[9] M. Yabuuchi, Y. Tsukamoto, M. Morimoto, M. Tanaka, and K. Nii, "20nm high-density single-port and dual-port SRAMs with wordline-voltage-adjustment system for read/write assists," in *IEEE Int. Solid State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2014, pp. 234–235.

[10] J. Kulkarni, M. Khellah, J. Tschanz, B. Geuskens, R. Jain, S. Kim, and V. De, "8T-bitcell SRAM Array in 22nm Tri-Gate CMOS for Energy-Efficient Operation across Wide Dynamic Voltage Range," in *Symp. VLSI Tech. Dig. Papers*, Jun. 2013, pp. C126–127.

[11] N. Planes, O. Weber, V. Barral, S. Haendler, D. Noblet D. Croain, M. Bocat, P. Sassoulas, X. Federspiel, A. Cros, A. Bajolet, E. Richard, B. Dumont, P. Perreau, D. Petit, D. Golanski, C. F.-Beranger, N. Guillot, M. Rafik, V. Huard, S. Puget, X. Montagner, M.-A. Jaud, O. Rozeau, O. Saxod, F. Wacquant, F. Monsieur, D. Barge, L. Pinzelli, M. Mellier, F. Boeuf, F. Arnaud and M. Haond, "28-nm FDSOI Technology Platform for High-Speed Low-Voltage Digital Applications" in *Symp. VLSI Technol. Tech. Dig.*, June 2012, pp. 133–134.

[12] P. Flatresse, B. Giraud, J. Noel, et al., "Ultra-Wide Body-Bias Range LDPC Decoder in 28-nm UTBB FDSOI Technology" in *IEEE Int. Sorid-State Circuits Conf. (ISSCC) Tech. Dig.*, Feb. 2013, pp. 424–425.

[13] L. Hutin, C. L. Royer, F. Andrieu, O. Weber, M. Casse, J.-M. Hartmann, D. Cooper, A. Béché, L. Brevard, L. Brunet, J. Cluzel, P. Batude, M. Vinet and O. Faynot, "Dual Strained Channel Co-Integration into CMOS, RO and SRAM cells on FDSOI down to 17nm Gate Length" in *IEDM Dig. Tech. Papers*, Dec. 2010, pp. 11.1.1–11.1.4.

[14] J. E. Husseini, X. Garros, J. Cluzel, A. Subirats, A. Makosiej, O. Weber, O. Thomas, V. Huard, X. Federspiel, G. Reimbold, "A complete Characterization and Modeling of the BTI-Induced Dynamic Variability of SRAM Arrays in 28-nm FD-SOI Technology," *IEEE Trans. on Electron Devices*, Vol. 61, no. 12, pp. 3991–3999, Dec. 2014.

[15] C. Fenouillet-Beranger, S. Denorme, B. Icard et al., "Fully-Depleted SOI Technology using High-K and Single-Metal Gate for 32nm Node LSTP Applications featuring 0.179µm2 6T-SRAM bitcell" in IEDM Dig. Tech. Papers, Dec. 2007, pp. 267–270.

[16] O. Thomas, B. Zimmer, B. P.-Prayer, N. Planes, K.-C. Akyel, L. Ciampolini, P. Flatresse and B. Nikolić, "6T SRAM Design for Wide Voltage Range in 28-nm FDSOI" in *IEEE Int. SOI Conf.*, Oct. 2012, pp. 1–2.

[17] H. Pilo, C. A. Adams, et al., "A 64Mb SRAM in 22nm SOI Technology Featuring Fine-Granularity Power Gating and Low-Energy

Power-Supply-Partition Techniques for 37% Leakage Reduction" in *IEEE Int. Solid State-Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2013, pp. 322–323.

[18] T. Sakurai, "Low power digital circuit design", *Proc. IEEE 30th Eur. Solid-State Conf. (ESSCIRC)*, Sep. 2004, pp. 11–18.

[19] M. Nomura, A. Muramatsu, H. Takeno, S. Hattri, D Ogawa, M. Nasu, K Hirairi, S. Kumashiro, S. Moriwaki, Y. Yamamoto, S. Miyano, Y. Hiraku, I. Hayashi, K. Yoshioka, A. Shikata, H. Ishikuro, M. Ahn, Y. Okuma, X. Zhang, Y. Ryu, K. Ishida, M. Takamiya, T. Kuroda, H. Shinohara, and T. Sakurai, "0.5V Image Processor with 563 GOPS/W SIMD and 32bit CPU Using High Voltage Clock Distribution (HVCD) and Adaptive Frequency Scaling (AFS) with 40nm CMOS" in *Symp. on VLSI Technol. Dig. Papers*, June 2013, pp. 118–119.

[20] K. Kang, H. Jeong, Y. Yang, J. Park, K. Kim, and S.-O. Jung, "Full-swing local bitline SRAM architecture based on the 22-nm FinFET technology for low-voltage operation," *IEEE Trans. Very Large Scale Intgr. (VLSI) Syst.*, vol. 24, no. 4, pp. 1342–1350, Apr. 2016.

[21] K.-H. Koo, L. Wei, J. Keane, U. Bhattacharya, E. A. Karl, and K. Zhang, "A 0.094μm2 high density and aging resilient 8T SRAM with 14nm FinFET technology featuring 560mV VMIN with read and write assist," in *Symp. VLSI Circuits, Dig. Tech. Papers*, Jun. 2015, pp. C266–C267.

[22] H. Mori, T. Nakagawa, Y. Kitahara, Y. Kawamoto, K. Takagi, S. Yoshimoto, S. Izumi, K. Nii, H. Kawaguchi, and M. Yoshimoto, "A 298-fJ/writecycle 650-fJ/readcycle 8T Three-Port SRAM in 28-nm FD-SOI Process Technology for Image Processor," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Sep. 2015, pp. 1–4.

[23] F. Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, Nov. 1958.

[24] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Proceedings of Neural Information*

Processing Systems (NIPS), pp. 1097–1105, Dec. 2012.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), pp. 770–778, June 2016.

[27] A. Karpathy, "What I learned from competing against a ConvNet on ImageNet," Blog at: http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/.

[28] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[29] Y. Ishii, H. Fujiwara, et al., "A 28 nm Dual-Port SRAM Macro with Screening Circuitry Against Write-Read Disturb Failure Issues" *IEEE J. Solid-State Circuits*, vol. 46, no. 11, pp. 2535–2544, Nov. 2011.

[30] S. Yoshimoto, S. Miyano et al., "A 40-nm 8T SRAM with Selective Source Line Control of Read Bitlines and Address Preset Structure," *Proc. Custom Integr. Circuits Conf. (CICC)*, Sep. 2013, pp. 1–4.

[31] J. Cong, "Neural Networks on FPGAs," in *IEEE Int. Sorid-State Circuits Conf. (ISSCC) Forum Papers*, Feb. 2017.

[32] Google Research Blog at: https://research.googleblog.com/2016/04/announcing-tensorflow-08-now-with.html.

[33] 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, no.9, pp. 1562–1570.

[34] S. Yoshimoto, M. Terada, S. Okumura, T. Suzuki, S. Miyano, H. Kawaguchi and M. Yoshimoto, "A 40-nm 0.5-V 20.1- W/MHz 8T SRAM with Low-Energy Disturb Mitigation Scheme" in *Symp. VLSI Circuits, Dig. Tech. Papers*, pp. 72–73, June 2011.

[35] H. Fujiwara, M. Yabuuchi, M. Morimoto and K. Tanaka, "A 20nm 0.6V 2.1μW/MHz 128-kb SRAM with no half select issue by interleave wordline and hierarchical bitline scheme" *in Symp. VLSI Circuits, Dig. Tech. Papers*,

June 2013, pp. 118–119.

[36] D. Wang, H. Lin, C. Chuang and W. Hwang, "Low-Power Multiport SRAM With Cross-Point Write Word-Lines, Shared Write Bit-Lines, and Shared Write Row-Access Transistors" *IEEE Trans. Circuits Syst. II Exp. Briefs*, vol. 61, no. 3, pp. 188–192, Mar. 2014.

[37] H. Fujiwara, K. Nii, H. Noguchi, J. Miyakoshi, Y. Murachi, Y. Morita, H. Kawaguchi and M. Yoshimoto, "Novel Video Memory Reduces 45% of Bitline Power Using Majority Logic and Data-Bit Reordering" *IEEE Trans. VLSI Systems*, vol. 16, no. 6, pp. 620–627, Jun. 2008.

[38] N.-C Lien, L. Chen, C. Chen, H. Yang, M. Tu, P. Kan, Y. Hu, C. Chung, S. Jou, W. Hung, "A 40 nm 512 kb Cross-Point 8 T Pipeline SRAM With Binary Word-Line Boosting Control, Ripple Bit-Line and Adaptive Data-Aware Write-Assist," *IEEE Trans. Circuits and Syst. I, Reg. Papers*, vol. 61, no. 12, pp. 3416–3425, Dec. 2014.

[39] L. Wen, X. Cheng, K. Zhou, S. Tian, and X. Zeng, "Bit-Interleaving-enabled 8T SRAM with shared data-aware write and reference-based sense amplifier," *IEEE Trans. Circuit Syst. II, Exp. Briefs*, vol. 63, no. 7, pp. 643–647, Jul. 2016.

[40] M. Yabuuchi, H. Fujiwara, Y. Tsukamoto, M. Tanaka, S. Tanaka, and K. Nii, "A 28nm High Density 1R/1W 8T SRAM Macro with Screening Circuitry against Read Disturb Failure," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Sep. 2013, pp. 1–4.

[41] M. Yabuuchi, Y. Sawada, T. Sano, Y. Ishii, S. Tanaka, M. Tanaka and K. Nii, "A 6.05-Mb/mm^2 16-nm FinFET Double Pumping 1W1R 2-port SRAM with 313-ps read access time," in *IEEE Symp. VLSI Circuits Tech. Dig. Papers*, Feb. 2016, pp. 14–15.

[42] D. N. da Silva, A. I. Reis, and R. P. Ribas, "CMOS logic gate performance variability related to transistor network arrangements," *Microelectronics Rel.*, vol. 49, nos. 9–11, pp. 977–981, Sep. 2009.

[43] M. Alioto, G. Palumbo, and M. Pennisi, "Understanding the Effect of Process Variations on the Delay of Static and Domino Logic," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 5, pp. 697–710, May 2010.

[44] D. N. da Silva, A. I. Reis, R. P. Ribas, "Gate delay variability estimation

method for parametric yield improvement in nanometer CMOS technology," *Microelectronics Rel.*, vol. 50, nos. 9–11, pp. 1223–1229, Aug. 2010.

[45] H. Mori, Y. Umeki, S. Yoshimoto, S. Izumi, K. Nii, H. Kawaguchi, and M. Yoshimoto, "A 28-nm 484-fJ/writecycle 650-fJ/readcycle 8T Three-Port FD-SOI SRAM for Image Processor," *IEICE Trans. Electron.*, vol. E99-C, no. 8, Aug. 2016.

[46] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv: 1409.1556, Sep. 2014.

[47] *MatConvNet* at http://www.vlfeat.org/matconvnet/.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," *Proceedings of European Conference on Computer Vision* (ECCV), *arXiv*:1603.05027, Jul. 2016.

[49] A. Krizhevsky, "One weird trick for parallelizing convolutional neural networks," *arXiv*:1404.5997, Apr. 2014.

[50] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M.'A. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Y. Ng, "Large Scale Distributed Deep Networks," *Proceedings of Neural Information Processing Systems (NIPS)*, pp. 1223–1231, Dec. 2012.

[51] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *arXiv*:1603.04467, Mar. 2016.

[52] Y. Bengio, "Practical Recommendations for Gradient-Based Training of Deep Architectures," *arXiv*:1206.5533, Sep. 2012.

[53] S. Gupta, W. Zhang, F. Wang "Model Accuracy and Runtime Tradeoff in Distributed Deep Learning: A Systematic Study," *Proceedings of IEEE International Conference on Data Mining* (ICDM), *arXiv*:1509.04210, Dec. 2016.

[54] J. Keuper, and F.-J. Pfreundt, “Distributed Training of Deep Neural Networks: Theoretical and Practical Limits of Parallel Scalability,” *arXiv*:1609.06870, Dec. 2016.

[55] H. Yoon, J. H. Nang, and S. R. Maeng, “A Distributed Backpropagation Algorithm of Neural Networks on Distributed- Memory Multiprocessors,” *Proc. of Symp. on Frontiers of Massiv. Parallel Comput.*, pp. 358–363, Oct. 1990.

[56] A. Petrowski, G. Dreyfus, and C. Girault “Performance Analysis of a Pipelined Backpropagation Parallel Algorithm,” *IEEE Trans. Neural Networks*, vol. 4, no. 6, pp. 970–981, Nov 1993.

[57] S. Zickenheiner, M. Wendt, B. Klauer, and K. Waldschmidt, “Pipelining and Parallel Training of Neural Networks on Distributed-Memory Multiprocessors,” in *IEEE Intr. Conf. on Neural Networks*, pp. 2052–2057, June 1994.

[58] K. Vesely, L. Burget, and F. Grezl, “Parallel Training of Neural Networks for Speech Recognition,” *Proc. of IEEE International Conference on Neural Networks*, pp. 2934–2937, Sep. 2010.

[59] F. Seide, G. Li, and D. Yu, “Conversational Speech Transcription Using Context-Dependent Deep Neural Networks,” *Proc. of ISCA Interspeech*, pp. 437–440, Aug. 2011.

[60] X. Chen, A. Eversole, G. Li, D. Yu, and F. Seide, “Pipelined Back-Propagation for Context-Dependent Deep Neural Networks,” *Proc. of ISCA Interspeech*, pp. 26–29, Sep. 2012.

[61] A. Harlap, D. Narayanan, A. Phanishayee, V. Seshadri, N. Devanur, G. Ganger, P. Gibbons, “PipeDream: Fast and Efficient Pipeline Parallel DNN Training,” *arXiv*:1806.03377, Jun 2018.

[62] J. Duchi, E. Hazan, and Y. Singer, “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization,” *Journal of Machine learning Reserch*, vol. 12, pp. 2121–2159, Nov. 2011.

[63] *nv-caffe at* https://github.com/NVIDIA/caffe/tree/caffe-0.16

[64] G. Huang, Y. Sun, Z. Liuy, D. Sedra, and K. Q. Weinberger, “Deep Networks with Stochastic Depth,” *arXiv*:1603.09382, July 2016.

[65] H. Mori, T. Youkawa, S. Izumi, H. Kawaguchi, and A. Inoue, “A layer-block-wise pipeline for memory and bandwidth reduction in distributed

deep learning," in *IEEE International Works. on Machine Learning for Signal Processing* (MLSP), pp. 1–6, Sep. 2017.

[66] K. Fukushima, "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position," *Biological Cybernetics*, vol. 36, no. 4, pp. 93–202, Apr. 1980.

List of Publications and Presentations

Publications in journals and transactions

1) **H. Mori**, Y. Umeki, S. Yoshimoto, S. Izumi, K. Nii, H. Kawaguchi and M. Yoshimoto, "A 28-nm 484-fJ/writecycle 650-fJ/readcycle 8T Three-Port FD-SOI SRAM for Image Processor," IEICE Trans. Electron., Vol. E99-C, No.8, pp. 901–908, Aug. 2016.
2) **H. Mori**, T. Nakagawa, Y. Kitahara, Y. Kawamoto, K. Takagi, S. Yoshimoto, S. Izumi, H. Kawaguchi and M. Yoshimoto, "A 28-nm FD-SOI 8T Dual-Port SRAM for Low-Energy Image Processor with Selective Sourceline Drive Scheme," IEEE Trans. on Circuit and Systems I, (Accepted).

Presentations at international conferences

1) T. Nakagawa, S. Izumi, K. Yanagida, Y. Kitahara, S. Yoshimoto, Y. Umeki, **H. Mori**, H. Kitahara, H. Kawaguchi, H. Kimura, K. Marumoto, T. Fuchikami, Y. Fujimori, and M. Yoshimoto, "A Low Power 6T-4C Non-volatile Memory using Charge Sharing and Non-precharge Techniques," IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2904–2907, May. 2015.
2) **H. Mori**, T. Nakagawa, Y. Kitahara, Y. Kawamoto, K. Takagi, S. Yoshimoto, S. Izumi, K. Nii, H. Kawaguchi and M. Yoshimoto, "A 298-fJ/writecycle 650-fJ/readcycle 8T Three-Port SRAM in 28-nm FD-SOI Process Technology for Image Processor," IEEE Custom Integrated Circuits Conference (CICC), pp. 1–4, Sep. 2015. (Intel/IBM/Catalyst Foundation CICC Student Scholarship Award)
3) Y. Umeki, K. Yanagida, H. Kurotsu, H. Kitahara, **H. Mori**, S. Izumi, M. Yoshimoto, H. Kawaguchi, S. Yoshimoto, K. Tsunoda, T. Sugii, "Process variation tolerant counter base read circuit for low-voltage operating STT-MRAM," DATE EMS Workshop, Mar. 2016.
4) **H. Mori**, T. Nakagawa, Y. Kitahara, Y. Kawamoto, K. Takagi, S. Yoshimoto, S. Izumi, H. Kawaguchi and M. Yoshimoto, "An Low-Energy 8T Dual-Port SRAM for Image Processor with Selective Sourceline Drive Scheme in 28-nm FD-SOI Process Technology," IEEE International Conference on Electronics, Circuits,

and Systems (ICECS), pp. 532–535, Dec. 2016. (Best Paper Award)

5) **H. Mori**, T. Youkawa, S. Izumi, M. Yoshimoto, H. Kawaguchi, and A. Inoue, "A Layer-block-wise Pipeline for Memory and Bandwidth Reduction in Distributed Deep Learning," IEEE Workshop on Machine Learning and Signal Processing (MLSP), pp. 1–6, Sep. 2017. (Best Student Paper Award)
6) K. Yamada, **H. Mori**, T. Youkawa, Y. Miyauchi, S. Izumi, M. Yoshimoto, and H. Kawaguchi, "Adaptive Learning Rate Adjustment with Short-Term Pre-Training in Data-Parallel Deep Learning," IEEE Workshop on Signal Processing Systems (SiPS), pp. 1–6, Sep. 2018.
7) T. Youkawa, **H. Mori**, Y. Miyauchi, K. Yamada, S. Izumi, M. Yoshimoto, and H. Kawaguchi, "DELAYED WEIGHT UPDATE FOR FASTER CONVERGENCE IN DATA-PARALLEL DEEP LEARNING," IEEE Global Conference on Signal and Information Processing (Global-SIP), pp. 1–4, Nov. 2018.
8) Y. Miyauchi, **H. Mori**, T. Youkawa, K. Yamada, S. Izumi, M. Yoshimoto, H. Kawaguchi, and A. Inoue, "Layer Skip Learning using LARS variables for 39% Faster Conversion Time and Lower Bandwidth," IEEE International Conference on Electronics, Circuits, and Systems (ICECS), pp. 1–4, Dec. 2018.
9) **H. Mori**, S. Izumi, M. Yoshimoto, and H. Kawaguchi, "28-nm FD-SOI Dual-Port SRAM with MSB-Based Inversion Logic for Low-Power Deep Learning," IEEE International Conference on Electronics, Circuits, and Systems (ICECS), pp. 1–4, Dec. 2018.

Invited presentations at domestic conferences

1) **H. Mori,** T. Nakagawa, Y. Kitahara, Y. Kawamoto, K. Takagi, S. Yoshimoto, S. Izumi, K. Nii, H. Kawaguchi and M. Yoshimoto, "A 298-fJ/writecycle 650-fJ/readcycle 8T Three-Port SRAM in 28-nm FD-SOI Process Technology for Image Processor," *ICD*, vol. 116, no. 3, pp. 13–16, April 2016, Tokyo Japan.

Presentations at domestic conferences

1) **H. Mori,** T. Nakagawa, Y. Kitahara, Y. Kawamoto, K. Takagi, S. Yoshimoto, S. Izumi, K. Nii, H. Kawaguchi and M. Yoshimoto, "A 298-fJ/writecycle

650-fJ/readcycle 8T Three-Port SRAM in 28-nm FD-SOI Process Technology for Image Processor," *ICD*, vol. 116, no. 3, pp. 13–16, April 2016, Tokyo Japan.

2) **H. Mori,** K. Yanagida, Y. Umeki, S. Yoshimoto, S. Izumi, M. Yoshimoto, H. Kawaguchi, K. Tsunoda, and T. Sugii, "A STT-MRAM Architecture for Improving Throughput," *ICD*, vol. 113, no. 419, ICD2013-110, p. 27, Jan. 2014, Kyoto Japan.
3) Y. Kawamoto, S. Yoshimoto, T. Nakagawa, Y. Kitahara, **H. Mori,** K. Takagi, S. Izumi, K. Nii, H. Kawaguchi, and M. Yoshimoto, "Low-Power SRAM in 28-nm FD-SOI for Image Processor," *ICD*, vol. 113, no. 419, ICD2013-116, p. 41, Jan. 2014, Kyoto Japan.
4) **H. Mori,** K. Yanagida, Y. Umeki, S. Yoshimoto, S. Izumi, H. Kawaguchi, M. Yoshimoto, K. Tsunoda, and T. Sugii, "A STT-MRAM Architecture for improving write throughput," *LSI and system Workshop 2014*, pp. 186–188, May 2014, Kokura Japan.
5) H. Kitahara, T. Nakagawa, S. Izumi, K. Yanagida, Y. Kitahara, S. Yoshimoto, **H. Mori,** H. Kawaguchi, H. Kimura, K. Marumoto, T. Fuchikami, Y. Fujimori, and M. Yoshimoto, "A Low Power 6T-4C Non-volatile Memory Techniques," *LSI and system Workshop 2015*, May 2015, Kokura Japan.
6) **H. Mori,** T. Nakagawa, Y. Kitahara, Y. Kawamoto, K. Takagi, S. Yoshimoto, S. Izumi, K. Nii, H. Kawaguchi and M. Yoshimoto, "A 298-fJ/writecycle 650-fJ/readcycle 8T Three-Port SRAM in 28-nm FD-SOI Process Technology for Image Processor," *ICD*, vol. 116, no. 3, pp. 13–16, April 2016, Tokyo Japan.
7) **H. Mori,** T. Nakagawa, Y. Kitahara, Y. Kawamoto, K. Takagi, S. Yoshimoto, S. Izumi, H. Kawaguchi and M. Yoshimoto, "A Low power 8T Dual-Port SRAM with Selective Sourceline Drive Scheme in 28-nm FD-SOI Process Technology for Image Processor," *LSI and system Workshop 2017*, May 2017, Tokyo Japan.
8) **H. Mori,** T. Youkawa, S. Izumi, M. Yoshimoto, H. Kawaguchi, and A. Inoue, "A Layer-block-wise Pipeline for Memory and Bandwidth Reduction in Distributed Deep Learning," *LSI and system Workshop 2018*, May 2018, Tokyo Japan.

Acknowledgement

First and foremost, I would like to express my sincere gratitude to my advisor Professor Hiroshi Kawaguchi of Kobe University for the continuous support of my Ph.D. study and related research. I appreciate all his contributions of time, patience, motivation, and immense knowledge to make my Ph.D. experience productive and stimulating. His guidance, insightful suggestions, invaluable comments, and grateful encouragements were motivational, and encouraged me in all the time of research and writing of this dissertation, even during tough times in the Ph.D. pursuit. I would like to pay attention my deepest gratitude to Professor Emeritus, Masahiko Yoshimoto of Kobe University for providing me constructive comments and warm encouragement related to this research. My deepest appreciation goes to them.

I also would like to offer my special thanks to Professor Makoto Nagata, Professor Mitsuo Yokokawa, and Associate Professor Noriyuki Miura for their helpful suggestions related to this dissertation.

The members of RAM project group and DNN project group have contributed immensely to my personal and professional time at Kobe University. I am grateful to the team members for their talents and invaluable supports: Dr. Shusuke Yoshimoto, Dr. Youhei Umeki, Mr. Yuki Kitahara, Mr. Tomoki Nakagawa, Mr. Naoki Okawa, Mr. Hiroto Kitahara, Mr. Hiroaki Kurotsu are with the RAM project, Mr. Tetsuya Youkawa, Mr. Yuki Miyauchi, Mr. Kazuki Yamada, Ms. Fuyuka Yamada are with the DNN project. I owe my deepest gratitude to Specially Appointed Associate Professor Shintaro Izumi of Osaka University for giving me meticulous comments. I am particularly grateful for the technical discussions and comments given by those research members: Dr. He Guangji, Dr. Keisuke Okuno, Dr. Go Matsukawa, Dr. Chikako Nakanishi, Dr. Motofumi Nakanishi, and Mr. Koji Yano.

I have received generous support from Mr. Kenta Takagi, Mr. Masanao Nakano, Mr. Asuka Fujikawa, Mr. Yuki Miyamoto, Mr. Ken Yamashita, Mr. Yuta Kimi, Mr. Song Dae-Woo, Mr. Kotaro Tanaka, Mr. Takahide Fujii, Mr. Kunpei Matsuda, Mr. Yuta Kawamoto, Mr. Yozaburo Nakai, Ms. Kana Nakamura, Mr. Shuhei Yoshida, Mr. Taisuke Kodama, Mr. Masanori Koyama, Mr. Yoshito Tanaka, Mr. Daichi Matsunaga, Mr. Yusuke Fujita, Ms. Mio Tsukahara, Mr. Ryota Nakamura, Mr. Yuki Nagasato, Ms.

Yuri Nishizumi, Mr. Yoshitaka Matsuda, Mr. Takaaki Okano, Mr. Koichi Kajihara, Mr. Takumi Katsuura, Mr. Yuki Nishikawa, Mr. Reiya Kawamoto, Mr. Seiya Yoshida, Mr. Kento Watanabe, Mr. Masaya Kabuto, Ms. Kana Sasai, Mr. Masakazu Taichi, Mr. Riki Narukage, and Mr. Daisuke Watanabe in our laboratory. I have had the warm encouragement of Ms. Emi Go, Ms. Yuna Tamura and Ms. Keiko Matsuoka. I also would like to express my great appreciation to Ms. Mitsu Tsukino for her enormous supports to enhance my presentation skills and brush up the language skill.

I have greatly benefited from Dr. Koji Nii, with Renesas Electronics Corporation, Dr. Atsuki Inoue with Fujitsu Laboratories Ltd., for chip implementation in the dissertation. Also, I have greatly benefited from Dr. Hidehiro Fujiwara and Dr. Hiroki Noguchi with Taiwan Semiconductor Manufacturing Company (TSMC) for technical discussions in my intern period at TSMC, Hsinchu, Taiwan.

I have received financial and technical supports of this research. Chapters 3 and 4 are carried as a part of the support by the Semiconductor Technology Academic Research Center (STARC), and partially supported by the New Energy and Industrial Technology Development Organization (NEDO), JSPS KAKENHI, under Grant 18J11572 and Grant 18H01500. Chapters 5 is supported by Fujitsu Laboratories Ltd., the New Energy and Industrial Technology Development Organization (NEDO), and JSPS KAKENHI, under Grant 18J11572 and Grant 18H01500. This work is also supported by Grant-in-Aid for Japan Society for the Promotion of Science (JSPS) Fellow. This dissertation was supported by VLSI Design and Education Center (VDEC), The University of Tokyo in collaboration with Cadence Design Systems Inc., Mentor Graphics Corp., and Synopsys Inc., CMP, Inc.

Finally, I would like to express my heartfelt gratitude to my family for giving me their persistent help, moral support, and considerable encouragement.

Haruki Mori

Doctor Thesis, Kobe University
"A study on low-energy memory architecture for image processors", 118 pages
Submitted on January, 25, 2019
The date of publication is printed in cover of repository version published in Kobe University Repository Kernel.