A Study on Memory and Digital Architectures for Low-Power Signal Processing

Noguchi, Hiroki

(Degree)
博士（工学）

(Date of Degree)
2011-03-25

(Date of Publication)
2012-01-11

(Resource Type)
doctoral thesis

(Report Number)
甲5251

(URL)
https://hdl.handle.net/20.500.14094/D1005251

Doctoral Dissertation

A Study on Memory and Digital Architectures for Low-Power Signal Processing

(低消費電力信号処理のためのメモリと
デジタルアーキテクチャに関する研究)

January 2011

Graduate School of Engineering
Kobe University

Hiroki Noguchi

Abstract

In this dissertation, low-power techniques of memory and digital architecture are presented for wearable and ubiquitous signal-processing applications. First, the background of this research area and objective of this study are described. In the second part, critical issues related to hardware implementation for low-power signal processing system are explained in the second part. The main issues of an advanced architecture and memory designs can be regarded as four limitations of memory bandwidth, power dissipation, standby electricity, and low-voltage operation. An explanation of each limitation is provided to enhance comprehension of the study objective.

In the third part, the low-power two-port SRAM design technique against process variation is discussed for low-power CMOS technology. The signal-processing demand for memory IP blocks as not only single-port SRAM but also dual-port SRAM to perform parallel operations. Multi-port SRAM is effective to reduce the total cycle times because of its parallel access mechanism. A conventional 8T SRAM comprises a precharge circuit and keeper circuit, and these circuits restrict low-voltage operation and operation frequency. To reduce the active power of the SRAM bitline, the non-precharge structure and novel cell topology are proposed. Two pMOS transistors are added to the conventional 8T memory cell, which creates the combination of the conventional 6T memory cell, an inverter, and a transmission gate. It is not necessary to prepare a precharge circuit because the inverter can independently charge and discharge the local read bitline. In the proposed SRAM, the precharge power on bitlines is eliminated and readout power is consumed only when the readout datum is changed. Measurement results verified that, at an operating frequency of 120 MHz, the proposed 64-kb video memory in a 90-nm process operates properly at 0.77 V, whereas the conventional SRAM does not function below 0.85 V. Those results demonstrate that the proposed SRAM achieves an 85% power saving on the read bitline, when regarded as an H.264 reconstructed image memory. We also examine dual-port SRAM design in terms of its area, speed, and readout power in a 45-nm process technology. Although the 8T SRAM has the lowest transistor count, and although it is the most area-efficient, the readout power consumption is high and the cycle time is notably increased because of peripheral circuits. The 10T differential-port SRAM would operate fastest if the differential

voltage were set to 50 mV. The 10T SRAM with a single-end read port consumes the least power.

The fourth part of this paper introduces low-voltage-operatable two-port SRAM design for dynamic voltage and frequency scaling (DVFS) techniques. The 7T/14T SRAM has been proposed to enhance SRAM dependability: two pMOS transistors are added between internal nodes in a pair of the conventional 6T bitcells. In this chapter, by adding a dedicated read port, we propose a 9T/18T dual-port SRAM that is presented. The additional read port is disturb-free. It can therefore operate at a lower voltage than the 7T/14T SRAM can. Moreover, the proposed SRAM has a 9T normal mode and 18T dependable mode. To achieve the 9T/18T SRAM architecture, an interleaved bitline scheme is incorporated for the dedicated read port. The 9T/18T dual-port SRAM can scale its speed, operating voltage, and power dynamically by combining two bitcells for one-bit information. We designed and fabricated the proposed SRAM using a 65-nm process. The measurement results show that the 18T dependable read mode can reduce the operation voltage to 0.45 V at a frequency of 1 MHz because of the disturb-free read port, although the dependable 14T SRAM demands 0.54 V at the same frequency.

The fifth part of this paper describes one low-power wearable digital signal processing architecture, memory-bandwidth reduction techniques for large-vocabulary real-time continuous speech recognition (LVRCS), which are critically dependent upon the bus frequency on VLSI. In the conventional speech recognition architecture, the memory bandwidth is an important issue for increasing the vocabulary. In the proposed architecture, parallel computing using speech signal correlation contributes to reduction of the memory bandwidth on a Gaussian Mixture Model setup. Two-stage language model search and specialized cache were also introduced for Viterbi processing to reduce the memory bandwidth. The two-stage language model search also reduces the computing amounts. The pipelining architecture is demonstrated for reducing the required frequency for real-time operations. An evaluation result obtained using hardware description language (HDL) and a verilog simulator shows the effectively reduced memory bandwidth and required frequency obtained when using the proposed techniques. For 60-k-word speech recognition, the required frequency can be reduced to 66.74 MHz. The required memory bandwidth can be reduced to 549.91 MB/s for real-time recognition, although the speech-recognition accuracy is maintained at

86.42%.

In the sixth part, the decentralized sound acquisition system is reported for a low-power ubiquitous digital signal processing system. The system uses a microphone-array-based sensor node network for sound acquisition. The sensor node has a voice activity detector (VAD), a sound-source localization module, and a sound-source enhancement module. Introducing a VAD circuit and power manager reduces the stand-by power. The VAD circuit outputs whether an input signal includes speech data or not. When the VAD module detects a speech signal, a main application module and signal-processing module are connected to a power source. When a speech signal is not detected, these circuits are blocked off. According to the speech-signal emergence ratio, such power management can save energy. To increase this saving factor, an extremely low-powered VAD was designed. The proposed VAD hardware achieves 3.49 μW at a frequency of 100 kHz. The sound acquisition performance of the proposed system is also presented in this part.

Finally, the conclusion of this study is presented. It is hoped that this research will help many designers to develop advanced digital signal processing architecture in the deep submicron era using low-power SoC as well as 32-nm or 22-nm devices.

Keywords: VLSI, SRAM, Low power, Low voltage, Speech recognition, Human interface, HMM, GMM, Viterbi, Sound source localization, Sound source separation

Table of Contents

Abstract i

Table of Contents v

List of Figures viii

List of Tables xiii

Chapter 1 Introduction 1
1.1 Background of this Research Area 1
1.2 Objective of this Study 3
1.3 Overview of this Thesis 3

Chapter 2 Issues of Low-Power Signal Processing VLSI 7
2.1 Introduction 7
2.2 Issue on SRAM Power Consumption 8
2.3 Issue of Minimum Operating Voltage of SRAM 11
2.4 Issue of Memory Bandwidth 11
2.5 Issue of Sound Acquisition Standby-Power 14
2.6 Summary 15

Chapter 3 Low-Power Two-Port SRAM Design 17
3.1 Introduction 17
3.2 8T SRAM 18
3.3 10T Single-End SRAM (10T-S SRAM) 20
3.4 10T Differential SRAM (10T-D SRAM) 23
3.5 Reducing the Number of Charge/Discharge Times 24
3.5.1 Application to Video Images 24
3.5.2 Block Size Optimization 25
3.6 Design in 90-nm Process Technology 27
3.6.1 Cell and Macro Layouts 27
3.6.2 Delay Model of Read-Bitline RC Trees 30
3.6.3 Chip Overview 33

3.6.4 Operating Frequency and Supply Voltage 35
3.6.5 Power 36
3.7 Design in 45-nm Process Technology 38
3.7.1 Cell and Macro Layouts 38
3.7.2 Operating Frequency versus Supply Voltage 41
3.7.3 Power 44
3.8 Summary 46

Chapter 4 Two-Port SRAM Design for DVFS 49
4.1 Introduction 49
4.2 Dependable SRAM: overview 50
4.3 7T/14T SRAM 51
4.4 9T/18T SRAM 53
4.5 Chip Implementation and Measurement Results 56
4.6 Summary 59

Chapter 5 Memory-Bandwidth Reduction for LVRCS 61
5.1 Introduction 61
5.2 Speech Recognition Overview 61
5.2.1 MFCC Feature Extraction 62
5.2.2 GMM Computation 62
5.2.3 Time-Synchronous Viterbi Beam Search 65
5.2.4 N-gram Language Model Search 66
5.3 Referential Hardware Design 67
5.3.1 Computation Amounts and Memory Bandwidth 67
5.4 Proposed Schemes 68
5.4.1 Burst GMM Calculation 68
5.4.2 Modified Unigram Language Model 69
5.4.3 Threshold-Cutting Scheme 70
5.4.4 Two-stage Language Model Search 71
5.5 VLSI Architecture 72
5.5.1 Implementation of GMM Computation 74
5.5.2 Viterbi and N-gram Architecture 76
5.5.3 Bigram Cache 78

5.5.4 Token List Cache 80
5.6 Implementation Results 80
5.6.1 Required Frequency and Memory Bandwidth 81
5.6.2 Comparison with other Architectures 81
5.7 Summary 83

Chapter 6 Low-Standby-Power Decentralized Sound Acquisition 85
6.1 Introduction 85
6.2 Intelligent Ubiquitous Sensor Network and Its Node 86
6.3 Voice Activity Detection 88
6.3.1 VAD Algorithms 89
6.3.2 Zero-Crossing VAD Algorithm 89
6.3.3 Modification the Zero-Crossing Algorithm 90
6.3.4 Hardware Implementation of VAD 92
6.3.5 Experimental Results 94
6.4 Proposed Sound Acquisition Scheme 95
6.4.1 Proposed Data Aggregation Scheme 96
6.4.2 Sound Source Localization Algorithm 97
6.4.3 Three-Dimensional Sound Source Localization 98
6.4.4 Simulation Results 100
6.5 Implementation of the Microphone Array System 100
6.5.1 Implementation 100
6.6 Future Work 103
6.7 Summary 104

Chapter 7 Conclusion 105

References 109

List of Publications and Presentations 117
Publications in journals and transactions 117
Presentations in international conferences 118
Presentations in domestic conferences 120

Acknowledgments 125

List of Figures

Fig. 1.1 Various applications of speech recognition. 2
Fig. 1.2 Front end and back end of a speech recognition system. 3
Fig. 1.3 Outline of this thesis. 5
Fig. 2.1 Trend of the memory area in future SoCs 9
Fig. 2.2 Pelgrom plots of different processes. The standard deviation of V_{th} increases as process technology is scaled down. 10
Fig. 2.3 RBL operation waveforms of (a) 90-nm and (b) 45-nm technologies at the SS corner (25°C). 10
Fig. 2.4 Trend of nominal operating voltage and minimum voltage of SRAM. ...11
Fig. 2.5 Conventional processor used for LVRCS. 12
Fig. 2.6 Vocabulary cover rate on various media, as examined by the National Institute for Japanese Language and Linguistics. 12
Fig. 2.7 Conventional architectures for real-time speech recognition and our target performance. 13
Fig. 3.1 8T dual-port SRAM: (a) a schematic and (b) waveforms in a read operation. 19
Fig. 3.2 10T SRAM with a single-end read bitline (10T-S SRAM): (a) a schematic and (b) waveforms in read operation. 21
Fig. 3.3 Charging–discharging times on an RBL in a 10T-S SRAM when a sense amplifier drive transistor width is changed at the SS corner (25°C). 22
Fig. 3.4 10T SRAM with differential read bitlines (10T-D SRAM): (a) a schematic and (b) waveforms in read operation. 23
Fig. 3.5 Circuit schematic of a sense amplifier in the 10T-D SRAM. 24
Fig. 3.6 Example of H.264 image data and its mapping onto eight LRBLs. 25
Fig. 3.7 HD video sequences. Each sequence comprises 100 frames and 1920 × 1024 pixels. 26
Fig. 3.8 H.264 encoding process. 26
Fig. 3.9 Transition possibilities (the normalized quantities of charge–discharge times) on an LRBL between the conventional 8T, MJ, and proposed 10T-S SRAMs when a block size is changed. 27

Fig. 3.10 Layouts of (a) the conventional MC, (b) the proposed MC without shared WL structure, and (c) the proposed MC with a shared WL structure.. 28
Fig. 3.11 Shared WL structure. 29
Fig. 3.12 A π-type RC model of the SRAM read port. 31
Fig. 3.13 Elmore delays by numeric calculation using ASPLA 90-nm process parameters. (a) Conventional 8T SRAM and (b) proposed 10T-S SRAM with the shared WL structure. 32
Fig. 3.14 Block diagram of a memory cell block in the proposed 10T-S SRAM.33
Fig. 3.15 Chip micrograph of the proposed 10T-S SRAM and the conventional 8T SRAM in a 90-nm process technology. The total memory size of each SRAM is 64 kb. 34
Fig. 3.16 Operation waveforms of the proposed 10T-S SRAM when (a) "0" and (b) "1" are read out in a 90-nm process technology (CC corner, 25°C). 34
Fig. 3.17 Area comparison of 64-kb SRAMs in a 90-nm process technology. ... 35
Fig. 3.18 Operating frequencies versus supply voltage with 90-nm process technology. Dotted lines show simulation results, and solid lines show the measurement results in this frequency comparison with conventional architecture. 36
Fig. 3.19 Measured readout power at 120 MHz when reading the original images and reconstructed images. 37
Fig. 3.20 (a) Readout power versus operating frequencies in a 90-nm process technology. (b) Magnified view. 38
Fig. 3.21 Cell layouts of (a) 8T, (b) 10T-S, and (c) 10T-D SRAMs, in a 45-nm process technology. 39
Fig. 3.22 Macro layouts of (a) 8T, (b) 10T-S, and (c) 10T-D SRAMs, in a 45-nm process technology. The total memory capacity of each macro is 64 kb. 40
Fig. 3.23 Operation waveforms of (a) 8T, (b) 10T-S, and (c) 10T-D SRAMs at the SS corner (25°C). 41
Fig. 3.24 Operating frequencies when a supply voltage is changed at the SS corner (25°C). 43
Fig. 3.25 Leakage power comparison of 8T, 10T-S, and 10T-D SRAMs at the CC corner (25°C). 43

Fig. 3.26 Density function of discharging time on RBL variation of 10T-D SRAMs. 44
Fig. 3.27 Readout power versus operating frequencies in a 45-nm process technology at the CC corner (25°C). 46
Fig. 4.1 Dependable SRAM. 51
Fig. 4.2 Schematics of (a) a conventional 7T/14T bitcell pair and (b) the proposed 9T/18T bitcell pair. 53
Fig. 4.3 Interleaved BL structure for read port in proposed 9T/18T SRAM. An RBL is shared by an upper left and lower right (or upper right and lower left) bitcells. An RWL is also shared but it is connected to every other bitcell. 54
Fig. 4.4 Layout of 9T/18T bitcell pair in a 65-nm process. 54
Fig. 4.5 Readouts of four kinds: (a) normal-mode differential read via inside BLs, (b) normal-mode single-ended read via outside read port, (c) dependable-mode differential read via inside BLs, and (d) dependable-mode differential read via outside read port. Here, SA denotes a sense amplifier. .. 55
Fig. 4.6 Chip micrograph and SRAM macro layout. 57
Fig. 4.7 Measured BERs of 9T/18T SRAM for four read operations. Frequency is 1 MHz. 57
Fig. 4.8 Measured characteristics in access time versus supply voltage: (a) normal-mode differential readout, (b) normal-mode single-ended readout, (c) dependable-mode inside differential readout, and (d) dependable-mode outside differential readout. 58
Fig. 5.1 Speech recognition flow with the HMM algorithm. 62
Fig. 5.2 Calculation flow to obtain MFCC from the waveform data. 63
Fig. 5.3 Left–right HMM. 63
Fig. 5.4 Left–right HMM calculation flow. 64
Fig. 5.5 Viterbi computation. 65
Fig. 5.6 Required frequency in a real-time process with the referential hardware [48, 49]. 68
Fig. 5.7 Required memory bandwidth in a real-time process with the referential hardware [48, 49]. 68
Fig. 5.8 Tree dictionary for Viterbi search. 70

Fig. 5.9 Beam width variation with the threshold-cut scheme. 71
Fig. 5.10 Appearance ratios of three transition types in Viterbi search in Julius 4.0. 72
Fig. 5.11 Cache architecture concept in Viterbi search processing. 73
Fig. 5.12 Block diagram of proposed processor architecture for a 60-k word speech recognition system. 74
Fig. 5.13 GMM computation flow. 75
Fig. 5.14 GMM processor data path. 76
Fig. 5.15 Viterbi computation flow. 77
Fig. 5.16 Proposed Viterbi architecture. 77
Fig. 5.17 Correlation rate of bigram accesses between adjacent frames profiled using Julius 4.0. 78
Fig. 5.18 Bigram cache hit rate profiled using Julius 4.0. 79
Fig. 5.19 Proposed two-way cache. 79
Fig. 5.20 Token list cache hit rate. 80
Fig. 5.21 Required frequency comparison with conventional architecture. 82
Fig. 5.22 Required memory bandwidth using the real-time process. 82
Fig. 6.1 Intelligent ubiquitous sensor network (IUSN) and block diagram of a sub-array node. 87
Fig. 6.2 Flow chart of intelligent ubiquitous sensor nodes. 88
Fig. 6.3 Zero-crossing point example. The offset line shows the direct current (DC) component. 90
Fig. 6.4 Zero-crossing VAD algorithm flow. 91
Fig. 6.5 Block diagrams of the integrated devices. The D flip-flop (DFF) circuits keep up each input data asynchronously. 92
Fig. 6.6 FPGA board with a microphone and a current tester. 93
Fig. 6.7 Layout plot of the zero-crossing algorithm integrated using CMOS 0.18-μm process technology. 93
Fig. 6.8 The false acceptance rate (FAR) in VAD outputs using the number of non-speech frames of the recorded condition as normalized criteria. 94
Fig. 6.9 The false rejection rate (FRR) in VAD outputs using the speech frames of the recorded condition as normalized criteria. 94

Fig. 6.10 Network traffic with (a) lossless and (b) lossy multi-hop networks..... 95
Fig. 6.11 Example of perfect aggregation among neighboring nodes. 96
Fig. 6.12 Delay-and-sum beam-forming mechanism. 96
Fig. 6.13 Three-dimensional sound-source localization. 99
Fig. 6.14 Sound-source localization experiment. 99
Fig. 6.15 Sound-source localization accuracy. 99
Fig. 6.16 System photographs: intelligent ubiquitous sensor node and a microphone array comprising sub-arrays. 101
Fig. 6.17 Experiment diagrams. 101
Fig. 6.18 SNR improvements vs. the number of microphones. 102
Fig. 6.19 Examples of traffic data sizes: (a) without and (b) with the proposed perfect data aggregations. 102
Fig. 6.20 Normalized traffic cost vs. the number of microphones. 103

List of Tables

Table 3.1 Simulation conditions in the H.264 encoder. 26
Table 3.2 Values of M, CL, CG, RL, RG, RMC, and RD, as obtained using the ASPLA 90-nm process parameters. 31
Table 4.1 Three modes in 7T memory cells. 52
Table 4.2 Body bias settings: ΔV_{tn} and $\Delta|V_{tp}|$ 57
Table 5.1 Memory bandwidth in 20-k word Viterbi search 72
Table 5.2 Comparison with other hardware-based systems. 83
Table 6.1 Device utilization summary. 92

Chapter 1 Introduction

1.1 Background of this Research Area

The remarkable progress of digital signal processing technology attained in recent years has allowed for practical application of multimedia applications using voice and images. These applications now span widely various fields such as car navigation equipment, portable appliances, robots, automatic writing at conferences, indexing for retrieval of images and voices, and so on. Of those, mobile equipment, wearable equipment, and intelligent robots have shown remarkable progress up to the present. Consequently, human interfaces have been attracting the attention of the people who are tasked with maintaining good natural communications between human beings and artificial systems that exist in various environments. Especially, speech recognition is remarkable as a human interface between a man and systems, allowing natural communications by anyone (presented in Fig. 1.1). Modern speech recognition systems are usually based on Hidden Markov Models (HMM). These are statistical models which output a sequence of symbols or quantities according to a probability function. Actually, HMMs are used in speech recognition because a speech signal can be treated as a piecewise stationary signal or as a short-time stationary signal, which is represented as triphones. Hierarchically, words are broken up into triphones and triphones are broken up into states. Speech recognition based on HMMs employs a large-scale trained dictionary, which consists of within-triphone (state-to-state) transitions and within-word (triphone-to-triphone) transitions. Large-vocabulary (more than 5,000 words) real-time continuous speech (LVRCS) recognition is necessary to achieve such a human interface. Actually, LVRCS supports completely hands-free interfaces because speech is the fundamental mode of human communication; moreover, speech interfaces offer a much broader range of application. In order for the speech recognition system to develop functions fully as a human interface, a continuous speech recognition system with extensive vocabulary and real-time features must be produced. At the same time, for implementation of a human interface for use in wearable equipment and robots, requirements for low power consumption and low cost are necessary. Although some software-based LVRCS exist, these solutions necessitate the use of a high-performance

processor, which consumes much power. For this reason, it is difficult to implement LVRCS on a mobile device, on wearable equipment, or in robots. Consequently, an application-specific processor for use in an LVRCS recognition system, a processor that can achieve high speed and low power consumption, is requested. A hardware approach such as those of VLSI or FPGA, can achieve more compact, more low-powered and more battery-friendly speech recognition because of their advantages for processing speed and power consumption: a high-speed but low-power speech-recognition processor is a necessity for mobile applications. For such a processor, the innovative low-power memory and digital architecture are necessary.

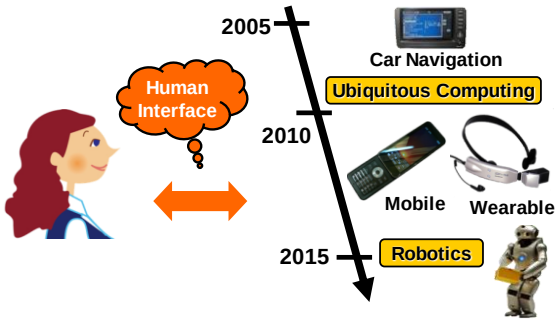

Fig. 1.1 Various applications of speech recognition.

In this dissertation, the research targets include both a back-end speech recognition system and a front-end speech acquisition system. Figure 1.2 presents some front-end and back-end system examples. A front-end system incorporates sound-acquisition functions such as sound-source localization, sound enhancement, and sound source separation. A back-end includes speech signal processing functions such as speech recognition, speaker identification, and context understanding. In the near future, an era of ubiquitous computing is coming, with numerous cameras and microphones located on walls and roofs of living spaces. They will obtain speech data and visual information automatically and support absolutely hands-free systems. The front-end of a speech recognition system usually involves multiple signal processing using a multi-channel interface such as a microphone array. A microphone array can localize sound sources and separate multiple sources using spatial information that can be collected along with the acquired sounds. The computational effort of these operations increases exponentially with the number of microphones, but the operating performance is known to increase as well. To reduce the increased power of a microphone array and to satisfy

recent demands for ubiquitous sound acquisition, it is necessary to realize a large sound processing system that uses little power. For such a system, the innovative low-voltage operable memory and digital architecture are necessary to reduce the stand-by power at the system level.

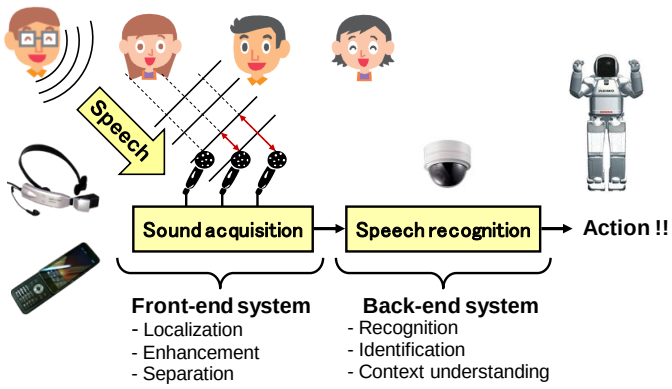

Fig. 1.2 Front end and back end of a speech recognition system.

1.2 Objective of this Study

In this research, hardware approaches are introduced to realize a wearable or ubiquitous speech recognition system that uses little power, as described above. Particularly, memory and digital architecture techniques are presented. Hardware approaches such as those using very large scale integrated (VLSI) circuits or a field programmable gate array (FPGA), can achieve more compact and more battery-friendly speech recognition using less power because of its advantages of high processing speed and low power consumption. The proposed architecture overcomes various issues related to speech signal processing and VLSI circuits, such as memory bandwidth, parallel computing, reduced operation voltage, low-power operation, power management, and so on. The main objective of these hardware techniques is to expand the dictionary size of speech recognition and the number of microphone arrays of the front-end sound acquisition with low power. Thereby, the system can assist human beings effectively.

1.3 Overview of this Thesis

In this dissertation, low-power techniques of memory and digital architecture are presented for wearable and ubiquitous signal processing applications. Figure 1.3 shows

that the outline of this thesis can be visualized with remarkable lucidity for this ambitious exposition. First, the background and objective of this study are described. For LVRCS and ubiquitous sound acquisition systems, critical issues related to hardware implementation are pointed out in Chapter 2. The main issues of an advanced architecture and memory designs are summarized as four limitations: memory bandwidth, power dissipation, standby electricity, and low-voltage operation. An explanation for each limitation is provided to enhance understanding of the study objective.

In the next four parts of this thesis, practical hardware design techniques against each limitation are demonstrated. In Chapter 3, the low-power two-port SRAM design technique against process variation is discussed for low-power CMOS technology. The speech signal processing demands memory IP blocks as not only single-port SRAM but also as dual-port SRAM to perform parallel operations. To reduce the active power of the SRAM bitline, a non-precharge structure and novel cell topology are proposed. The new SRAM has an advantage in terms of the number of charge and discharge times. In this part. Results show that the adequate compensate SRAM topologies are performed against the variation of threshold voltage of transistors and temperature. It is designed and fabricated using 90-nm and 45-nm advanced CMOS technology. Test results show that the active power is lowered by this circuitry.

Chapter 4 introduces a two-port SRAM design for a dynamic voltage and frequency scaling (DVFS) environment considering the proposed sound acquisition system described in chapter 5. The SRAM expands the operable-voltage width for reduced low-standby-power. The new SRAM presents advantages in terms of improvement of the minimum operating voltage lowering. The SRAM designed and fabricated using 65-nm CMOS technology and test results show that the minimum operating voltage is lower.

In Chapter 5, the memory-bandwidth reduction techniques for LVRCS, which are critically dependent on the VLSI bus frequency, are discussed first. It is presented that parallel computing using speech signal correlation contributes to reduction of the memory bandwidth on Gaussian Mixture Model setting up. The two-stage language model search and specialized cache were also introduced for Viterbi processing to reduce memory bandwidth. The two-stage language model search reduces the

computing amounts as well. The pipelining architecture is demonstrated for reducing the required frequency for real-time operation. An evaluation result obtained using hardware description language (HDL) and a verilog simulator shows the reduction of the memory bandwidth and required frequency using the proposed techniques.

In Chapter 6, the decentralized sound acquisition system is reported. The stand-by power is reduced by introducing a VAD circuit and power manager. The VAD circuit outputs whether an input signal includes speech data or not. When the VAD module detects a speech signal, a main application module and signal-processing module are connected to a power source. When a speech signal is not detected, these circuits are blocked off. According to the speech signal emergence ratio, the power management described above can save energy. To increase this saving factor, an extremely low-powered VAD is presented. The sound acquisition performance of the proposed system is also shown in this part.

The conclusion of this study is described in Chapter 7. The overall contribution is summarized briefly.

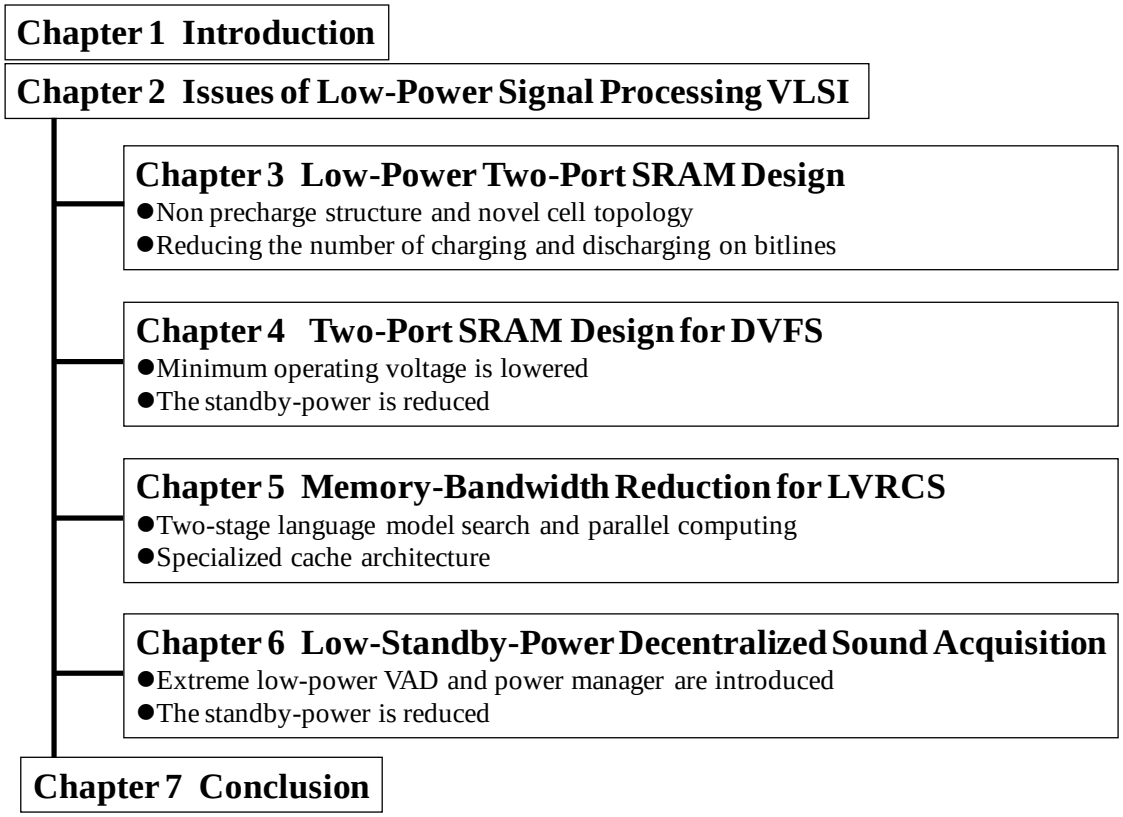

Fig. 1.3 Outline of this thesis.

Chapter 2 Issues of Low-Power Signal Processing VLSI

2.1 Introduction

Miniaturization of semiconductors has been a driving impetus for improvement of computer performance and the eventual progress of information technology. At International Electron Devices Meeting (IEDM) 2010, Taiwan Semiconductor Manufacturing Company Ltd., a major semiconductor foundry, disclosed logic CMOS technology for the 22 nm/ 20 nm generation [1]. In addition a research team with members from Intel Corp. and Micron Technology announced 64 Gb NAND flash memory of MLC type, which uses 25-nm manufacturing technology [2].

At present, when practical application of 22 nm generation is at hand, as described above, the International Technology Roadmap for Semiconductor (ITRS) predicts that miniaturization of channel length of 4.5 nm will be realized steadily in 2022. When miniaturization technology advances, the number of transistor devices which can be integrated increases. Given the same number of devices, the chip area is less, the number of chips obtainable from one sheet of wafer increases, production costs decrease, power consumption decreases because of reduction in operating voltage, and the operation clock is improved.

In recent years, however, problems have arisen despite the clear benefits obtained through miniaturization. When insulating film is thinned to obtain sufficient electric field effect with reduced operating voltage, tunnel effects arise, by which electrons pass through the insulating film that is thinned to its extremity. As a result, leakage current and scattering of the threshold voltage became remarkable. The leakage current increases the semiconductor power consumption and simultaneously creates difficulties in increasing the operation clock.

When problems such as leakage current and scattering of the threshold voltage are resolved as research and development of the process advance, operations of logic LSI of gigahertz-order will be realized in fine processes in the future. Even if such is not the case, it is anticipated that as the technology to improve performance becomes independent of the increase in the operation clock, then the use of micro-architecture to

improve IPC and of multi-core structures will be promoted, thereby drastically improving the VLSI computing performance. Particularly from the perspectives of multimedia applications, benefits from miniaturization are improvement of computing performance and large-capacity memory. Along with these two points, a system that has not been attained to date will be realized for use with image-processing and voice-processing systems in the near future. It is also considered that research and development of VLSI for image and speech processing will be promoted in two directions of exclusive use and general-purpose use, with encouragement by further development of VLSI miniaturization technology.

Although VLSI for general-purpose use has already been realized in the form of GPGPU and DSP, general-purpose VLSI remains absolutely necessary because image-processing and speech-processing uses diverse algorithms. It is expected that general-purpose use VLSI can provide higher computing performance with low power consumption because of miniaturization. Actually, VLSI for dedicated use is advantageous to VLSI for general-purpose use in terms of real-time performances and ultra-low power consumption. It is necessary in such a case in which interactive and real-time performances that can not be attained by VLSI for general-purpose use are required. Applications which show the benefits of miniaturization most are image processing and speech processing, which demand real-time performance in applications with limited use of batteries such as portable appliances.

For high-performance signal-processing systems, the use of digital-architecture and embedded SRAMs is confronting a crisis of increasing power dissipation. The number of accesses to the external memory increases according to the data size of signal processing. Total computation is also increasing according to the development of the application. These demands induce a system bottleneck. In this chapter, the fundamental principles of the VLSI signal-processing system and scaling trends of SRAM are described briefly.

2.2 Issue on SRAM Power Consumption

As the ITRS Roadmap predicts, the memory area is becoming larger (presented in Fig. 2.1). That area is expected to occupy 90% of the system on a chip by 2013 [3]. For example, an H.264 encoder used for a high-definition television requires at least a

500-kb memory as a search-window buffer, which consumes 40% of its total power [4]. As multimedia applications have become more complex and memory-demanding, large-capacity SRAMs will be adopted as frame buffers and/or restructured-image memory on video chips. The large-capacity SRAM potentially dissipates a larger share of its total power, and dominates the circuit speed. Therefore, low-power and high-speed dual-port SRAMs are strongly demanded for signal processing. Particularly, the power and operating frequency in a read operation are crucial because the readout takes place more frequently than write-in in a video codec. For instance in motion estimation, once picture data are written in memory, full-search algorithms or other motion compensation algorithms read out the data many times.

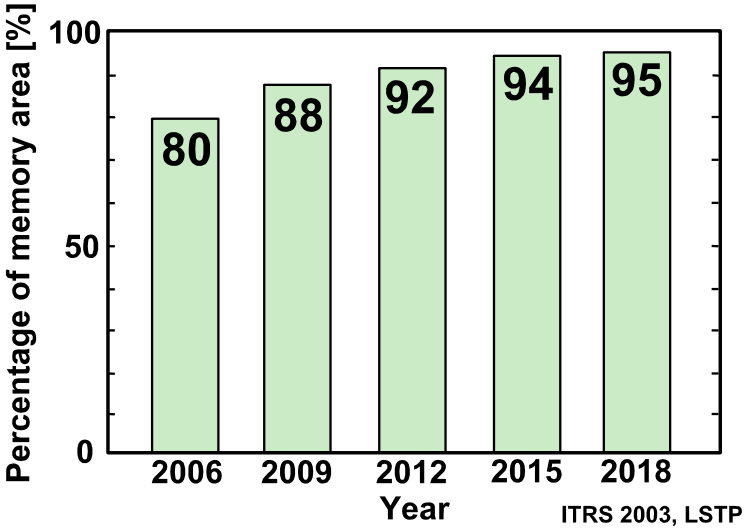

Fig. 2.1 Trend of the memory area in future SoCs

In fact, as process technology is scaled down, the threshold-voltage (V_{th}) variation of MOS transistors is increased (presented in Fig. 2.2) [3] because the channel area ($L_{eff} \times W_{eff}$) is shrunk as manufacturing processes advance. The readout current on the read bitline (RBL) is sensitive to the V_{th} variation. Figure 2.3 presents the readout operation waveforms of the single-end SRAM of 90-nm and 45-nm technologies. The SS corner, which denotes slow nMOS and slow pMOS, is one process corner, representing the extremes of fabrication-parameter variations within which a circuit that has been etched onto the wafer must function correctly. Designers examine the expected process range using ‘worst case’ analyses to verify that circuits will operate correctly under the V_{th} variation. The classic worst-case situation is the asymmetrical assignment of the V_{th}

variation to nMOS and pMOS, which worsens the charge speed or discharge speed. However, such a worst case involves an impossible situation in terms of probability. Monte Carlo simulation with a statistical die average model yields much more realistic results of how the circuits and especially how an SRAM will operate over the expected die average process variations. During the deep submicron era, it will be increasingly important to design the SRAM read-port while remaining cognizant of the V_{th} variation tolerance [5]. Monte Carlo simulation reveals the readout-timing variation and sense-timing difficulties and these lead to power dissipation in read operation.

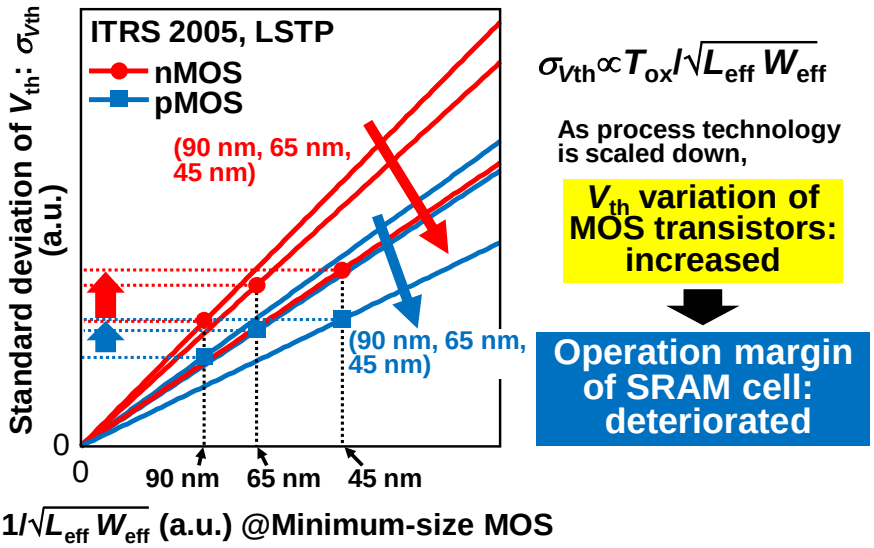

Fig. 2.2 Pelgrom plots of different processes. The standard deviation of V_{th} increases as process technology is scaled down.

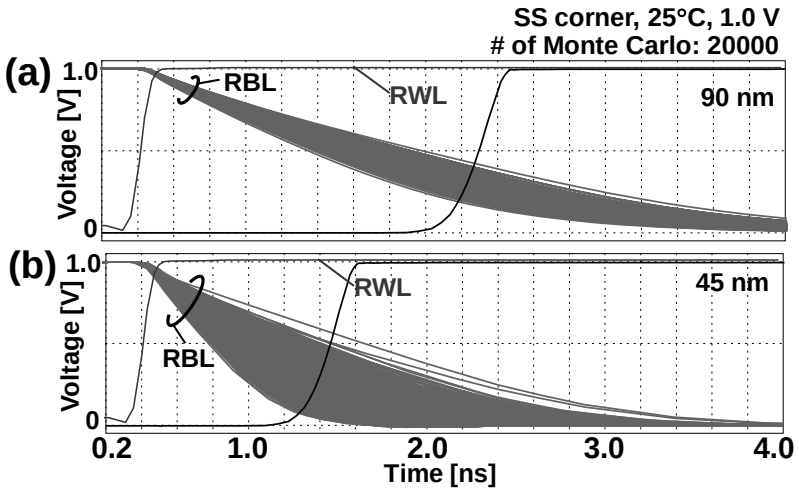

Fig. 2.3 RBL operation waveforms of (a) 90-nm and (b) 45-nm technologies at the SS corner (25°C).

2.3 Issue of Minimum Operating Voltage of SRAM

In order to save a power of an SoC, dynamic voltage and frequency scaling (DVFS) techniques that adaptively controls an operating frequency and supply voltage (V_{dd}) has been implemented in a mobile system [6]. However, a minimum operation voltage (V_{min}) is becoming higher as a fabrication technology is scaled down, since operation margins of memory cells in an embedded SRAM are degraded under both read and write conditions due to threshold-voltage (V_{th}) variation of MOSFETs. Figure 2.4 presents the nominal operating voltage and minimum voltage of SRAM as the fabrication technology is scaled down, as presented at International Solid-State Circuits Conference (ISSCC). Because the technology is advanced, the V_{th} variations of MOS transistors become readily apparent. Because of the V_{th} variation, operation margins of memory cells cannot be sustained in the conventional memory cells. A degradation of operation margins limits a lower bound of a supply voltage. Producing a low-voltage operating SRAM under the DVS environment is a key issue.

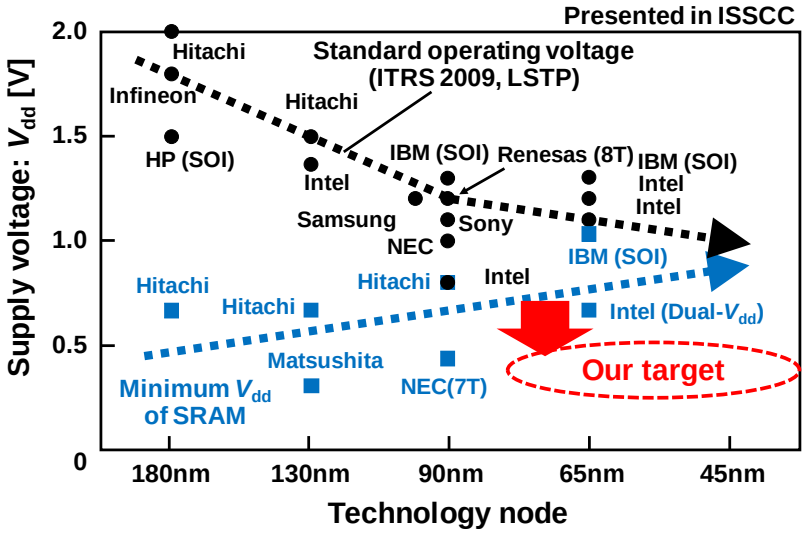

Fig. 2.4 Trend of nominal operating voltage and minimum voltage of SRAM.

2.4 Issue of Memory Bandwidth

Speech recognition technology has been used recently in various applications such as cellular telephones, car-navigation systems, PDAs, wearable computers, and robotics. Nevertheless, large vocabulary—more than 60-k words—real-time continuous speech recognition (LVRCSR) with an accurate model is too resource-hungry and

power-sensitive for use in software applications [7]. Figure 2.5 shows conventional hardware processors for LVRCS versus the vocabulary size. In terms of a hardware approach, the vocabulary size has remained around 5-k word recognition. Figure 2.6 presents the vocabulary cover rate in three situations: TV audio, TV display, and magazines. To process human daily conversation properly, more than 10-k word recognition is necessary considering general TV audio. Furthermore, more than 20-k word recognition is necessary to meet the need to process more natural conversation.

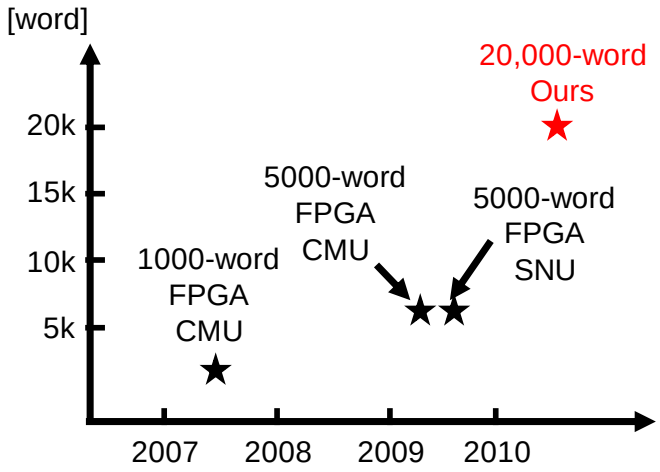

Fig. 2.5 Conventional processor used for LVRCS.

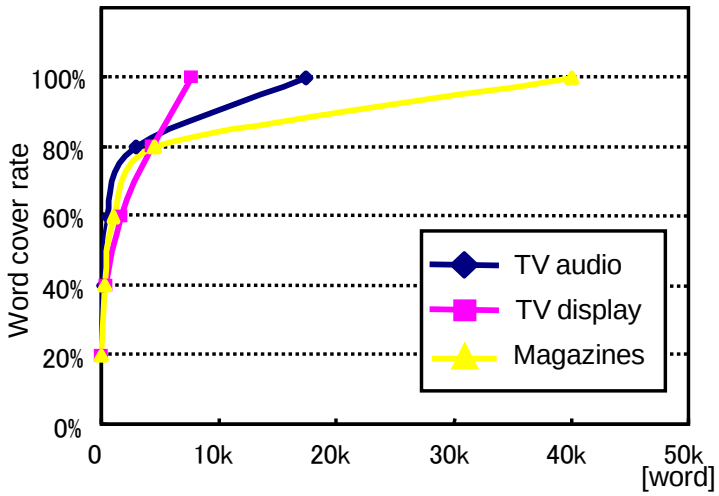

Fig. 2.6 Vocabulary cover rate on various media, as examined by the National Institute for Japanese Language and Linguistics.

A hardware approach, with implementation by VLSI or an FPGA, can achieve more compact and more battery-friendly speech recognition because of its advantageous

processing speed and power consumption. To enhance the speech-recognition performance of different systems, some studies have applied hardware approaches. Lin *et al*. investigated FPGA implementations for 5-k word continuous speech recognition [8, 9], but the applications did not run in real time. Choi *et al*. investigated FPGA implementations for 20-k word speech recognition [10, 11], but both consumed slightly greater memory bandwidth (BW) and power. Ma *et al*. reported memory-bandwidth reduction of Gaussian Mixture Models (GMM) processing for real-time 20-k word speech recognition [12], but that method did not treat Viterbi processing. Therefore, memory bandwidth reduction on a Viterbi processor remains as an important task because it requires high memory bandwidth.

A comparison of external memory bandwidth among recently described hardware-based speech recognizers is presented in Fig. 2.7. To date, the hardware approach has never achieved real-time operation with a 60-k word language model because numerous computations and external memory bandwidth degrade as the vocabulary is increased. In the ubiquitous-computing era that is expected to prevail in the future, with further development of robotics technology, speech recognition systems are expected to become the main technique used for human interface devices for mobile, wearable, and intelligent robots. Actually, LVRCSR is anticipated as a key technology for such applications.

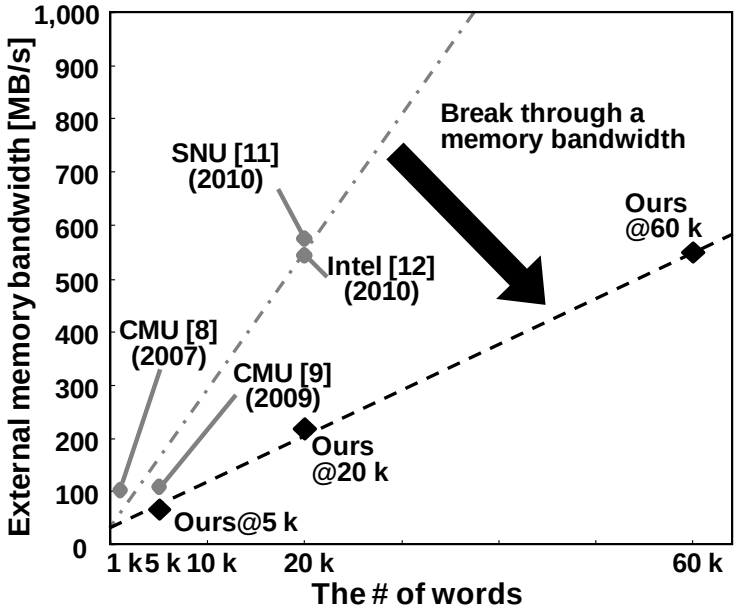

Fig. 2.7 Conventional architectures for real-time speech recognition and our target performance.

2.5 Issue of Sound Acquisition Standby-Power

Recent improvements in information processing technologies have produced real-time sound processing systems using microphone arrays [13]. One application is a meeting system with a 128-ch square microphone array [14], which captures speech data from every microphone. The microphone array processes signal recording and noise reduction, sound-source separation, speech recognition, speaker identification, and other tasks.

A microphone array can localize sound sources and separate multiple sources using spatial information of the acquired sounds. Huge microphone arrays have been widely investigated: arrays have been built at Tokyo University of Science (128 ch) [14], the University of Electro-Communication (156 ch) [15], Brown University and Rutgers University (512 ch) [16], and the Massachusetts Institute of Technology (1,020 ch) [17]. However, the problems of increasing computation, power consumption, and network costs render their practical use difficult, particularly in terms of sound-data acquisition. The main problem of conventional microphone array systems is that all microphones are connected to a single base station (high-performance sound server) with large-scale multi-channel sound recording devices. In conventional systems, the concentrative connection of a huge number of microphones engenders heavy network traffic. The computational effort increases polynomially as the number of microphones increases. If more than 1,000 microphones are used to collect data, then the signal-noise ratio (SNR) can be improved remarkably [17], but the network traffic and computational requirements would explode. To reduce the increased network traffic and computational power of a microphone array system and to satisfy recent demands for ubiquitous sound acquisition, it is necessary to realize a large sound-processing system covering wide-ranging human environments at low power.

2.6 Summary

For future low-power but high-performance signal processing systems, the key issues can be summarized as the following four items.

- active power consumption, especially on Dual-Port SRAM
- minimum SRAM operating voltage for a dynamic voltage and frequency scaling (DVFS) power controllable sensor node,
- memory bandwidth considering the large vocabulary language model, and
- standby-power of sound acquisition system including network traffic

The following four chapters explain solutions for the issues listed above.

Chapter 3 Low-Power Two-Port SRAM Design

As process technology is scaled down, large-capacity SRAMs will be used. Their power requirements must be lowered. In this chapter, we propose a two-port non-precharge SRAM comprising 10 transistors that is suitable for use in processing a real-time video image that has statistical similarity. The simulation reveals that the proposed SRAM can operate at a 65% higher frequency than a conventional 8T SRAM because it has no precharge period. The area overhead is 14.4% in a 90-nm process technology. Measurements demonstrats that the proposed SRAM saves 85% of the readout power when used as an H.264 reconstructed-image memory.

We also examine dual-port SRAM design in terms of its area, speed, and readout power in a 45-nm process technology. Although the 8T SRAM has the lowest transistor count, and although it is the most area-efficient, the readout power consumption is high and the cycle time is notably increased because of peripheral circuits. The 10T differential-port SRAM would operate fastest if the differential voltage were set to 50 mV. The 10T SRAM with a single-end read port consumes the least power.

3.1 Introduction

As the ITRS Roadmap predicts, the memory area used on a chip is becoming larger. It will occupy 90% of an SoC's area by 2013 [3]. Even on a real-time video SoC, this trend is progressing. An H.264 encoder for a high-definition television requires at least a 500-kb memory as a search-window buffer, which consumes 40% of its total power [4]. As process technology is scaled down, a large-capacity SRAM will be adopted as a frame buffer and a restructured-image memory on a video chip, and might dissipate a larger share of its power. To save power in a real-time video application, this chapter describes a low-power two-port SRAM.

Furthermore, this chapter presents a comparison of dual-port SRAMs of three kinds in a 45-nm process technology. A dual-port SRAM is extremely useful for real-time video processing because it can make one read and one write access simultaneously in a clock cycle [4, 18–20]. We handle the three kinds of dual-port SRAMs: the 8T SRAM,

10T SRAM with a single-end read port (hereinafter, '10T-S SRAM'), and 10T SRAM with differential read ports (hereinafter, '10T-D SRAM').

3.2 8T SRAM

In the conventional eight-transistor (8T) two-port memory cell (MC) depicted in Fig. 3.1(a), two nMOS transistors (N5 and N6) for a read wordline (RWL) and a local read bitline (RBL) are added to a single-port 6T MC, which frees a static noise margin (SNM) in a read operation because it has a separate read port [21]. A precharge circuit must be implemented on the RBL so that the two nMOS transistors can sink a bitline charge to the ground and a certain power is dissipated by precharging (see Fig. 3.1(b)). Furthermore, the readout time becomes greater as the supply voltage (V_{dd}) decreases because of the bitline keeper on the RBL [22]. In addition to the precharge circuit, it is necessary to prepare a bitline keeper on the RBL in the conventional two-port SRAM. Many MCs connecting to the RBL draw bitline leakage even if they are not selected as a readout bit. Even when a selected MC does not discharge the RBL ("1" readout), the RBL voltage would be decreased by the bitline leakage in such the case if no bitline keeper existed. The bitline keeper compensates this bitline leakage and maintains the voltage level on the RBL during a "1" readout [22]. Otherwise, we cannot distinguish a readout current from the bitline leakage, which is a readout malfunction.

In the 8T SRAM, an inverter circuit is used as a sense amplifier connecting to an RBL. When a datum "1" is read out, the sense amplifier inverter need not pay a delay overhead. In contrast, when a datum "0" is read out, the sense amplifier inverter takes a certain access time by discharging the readout node. The access time in the read operation is therefore determined by the "0" readout. In other words, the logical threshold voltage of the sense amplifier inverter should be adjusted higher to minimize the discharge time.

As process technology has advanced, the supply voltage and threshold voltage of transistors have decreased. Because the low threshold voltage increases the bitline leakage, we must increase the bitline keeper; then pay area overhead. The large bitline keeper imparts a negative influence on the readout time as well. To make matters worse, the delay overhead increases as the supply voltage decreases.

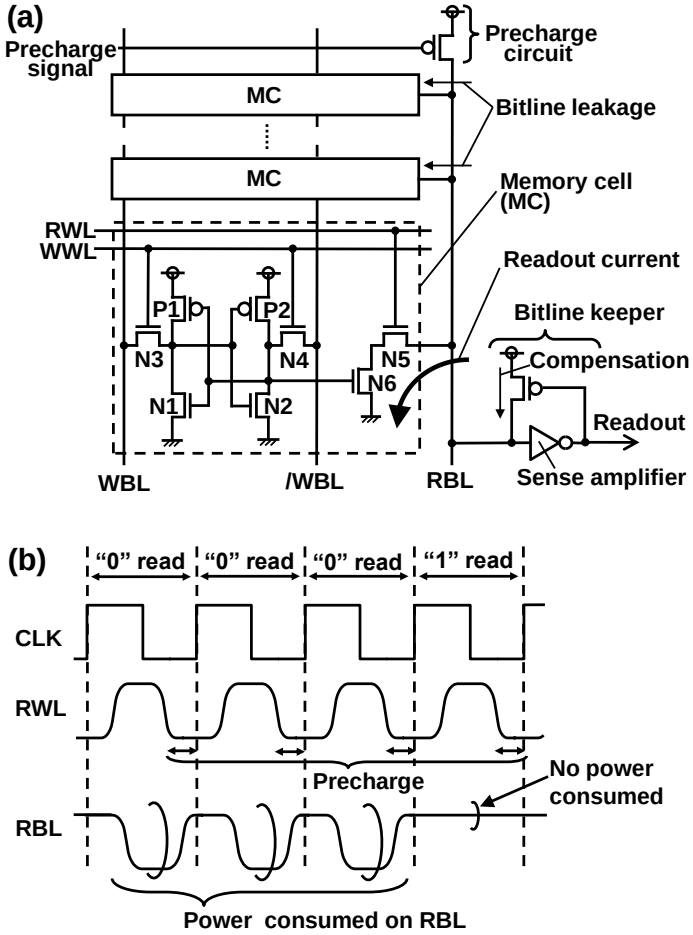

Fig. 3.1 8T dual-port SRAM: (a) a schematic and (b) waveforms in a read operation.

Figure 3.1(b) simplified operation waveforms in read cycles in the conventional 8T precharge-type SRAM. A precharge scheme is adopted and an RBL must be precharged to a supply voltage by the start time of a clock cycle. Therefore, a charge–discharge power is consumed on the RBL when "0" is read out. In contrast, no power is consumed when "1" is read out because the RBL keeps the supply voltage level and we need not precharge the RBL.

In an earlier study, which examined saving the charge–discharge power on a read bitline, a majority logic circuit and data-bit reordering were accommodated to write as many "1"s as possible [23] (the prior SRAM is designated as 'MJ SRAM' in this chapter). The MC structure in the MJ SRAM is the same as the conventional 8T SRAM although the read and write circuits differ. Input data comprising eight pixels are reordered into digit groups (from the most-significant-bit group to the least-significant-bit group); then a flag bit is appended to each group. The "0" data are

inverted to “1”s by the majority logic circuit if the number of “0”s in a group is more than that of “1”s. Thereby, we can maximize the number of “1”s in the input data. The inversion information (“1” signifies inversion) is stored in the additional flag bit. In a read cycle, the group data are inverted if a flag bit is true. Then they are put back in the original order so that it is possible to read out the original data. This mechanism reduces the power of the read bitline because we can statistically increase the probability that “1” is read where no power is dissipated.

For further power reduction, we propose a novel non-precharge-type SRAM (10T-S SRAM) in this chapter [24]. The 10T-S SRAM reduces the bitline power in both cases in which consecutive “0”s are read out and consecutive “1”s are read out because no precharge circuit exists on bitlines. The charge–discharge power is consumed only when a readout datum is changed. In contrast, in a conventional 8T SRAM, a consecutive-“0” readout requires large amounts of bitline power. In addition to the power reduction with the consecutive readout, the 10T-S SRAM operates in a shorter cycle time because a precharge period is not required. Furthermore, this structure obviates the bitline keeper, which improves operation in the low-voltage region. In comparison to the MJ SRAM, the 10T-S SRAM eliminates the flag bit that causes the power overhead.

3.3 10T Single-End SRAM (10T-S SRAM)

To improve the 8T SRAM, we have proposed a 10T non-precharge SRAM with a single-end read bitline [24–26], as depicted in Fig. 3.2(a) (10T-S SRAM). Two pMOS transistors are appended to the 8T SRAM cell, which engenders the combination of the conventional 6T single-port MC, an inverter, and a transmission gate. The additional signal (/RWL) is an inversion signal of a read wordline (RWL); it controls the additional pMOS transistor (P4) at the transmission gate. The additional pMOS transistor (P4) increases a RBL capacitance compared to the conventional 8T SRAM. Although the RWL and /RWL are asserted and the transmission gate is on, a stored node is connected to an RBL through the inverter. It is not necessary to prepare a precharge circuit because the inverter can charge–discharge the LRBL independently. No precharge circuit exists on either differential write bitline (WBL and /WBL) because they are dedicated for a write port.

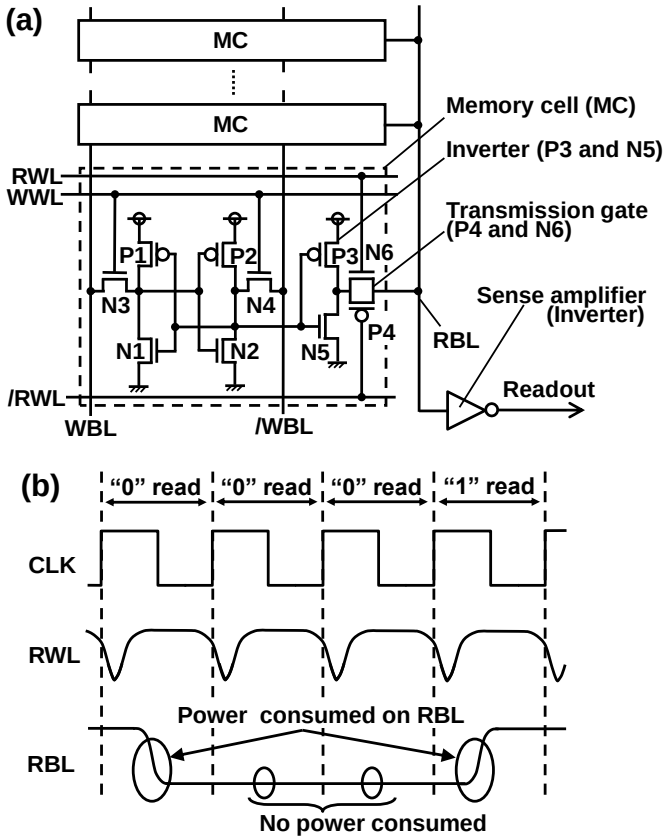

Fig. 3.2 10T SRAM with a single-end read bitline (10T-S SRAM): (a) a schematic and (b) waveforms in read operation.

Figure 3.2(b) depicts operation waveforms in the 10T-S SRAM in read cycles. A charge–discharge power on the RBL is consumed only when the RBL is changed. Consequently, no power is dissipated on the RBL if an upcoming datum is the same as the previous state. The 10T-S SRAM is suitable for a real-time video image that has statistical similarity [24–26].

The 10T-S SRAM theoretically reduces the power on the RBL to half that of a conventional 8T SRAM in a read operation if the readout data are random and the bitline capacitance is equal. The transient probability in a sequence of random data is 50% in the 10T-S SRAM; in the 8T SRAM, the number of charge–discharge times becomes one as an expected value. In the 8T SRAM, a charge and discharge pair takes place when "0" is read out. The RBL power is thereby reduced to about 50% in the read operation used for the 10T-S SRAM.

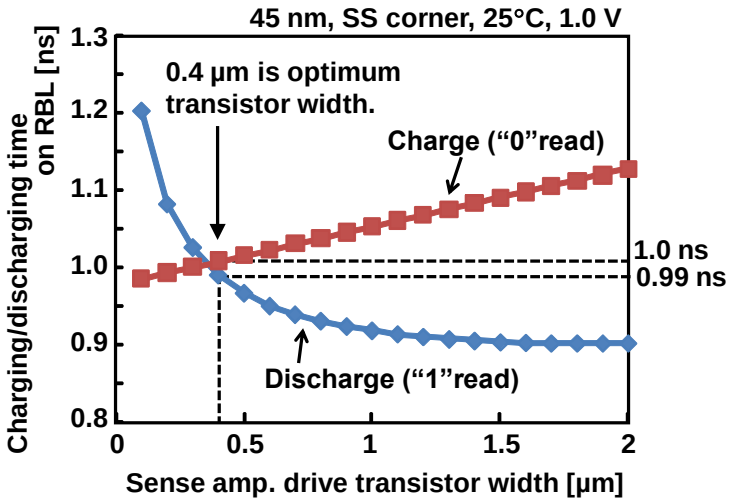

Fig. 3.3 Charging–discharging times on an RBL in a 10T-S SRAM when a sense amplifier drive transistor width is changed at the SS corner (25°C).

In the 10T-S SRAM, an inverter is connected to an RBL as a sense amplifier, just as with the 8T SRAM. The logical threshold voltage of the sense amplifier inverter should be adjusted in the middle, considering the charge–discharge on an RBL and maintaining their balance. Figure 3.3 shows the charging–discharging times on the RBL in the 10T-S SRAM when the drive transistor (nMOS) width in the sense amplifier inverter is changed. In the figure, the load transistor (pMOS) width in the sense amplifier is set to the minimum size—0.1 µm for the middle logical threshold voltage—because in the 10T-S SRAM, the drive power of the nMOS transistor N5 (see Fig. 3.2(a)) is stronger than that of the pMOS transistor P3 (see Fig. 3.2(a)) when the transistor sizes are the same. Therefore, the charging time is longer than the discharging one on the RBL. When the drive transistor width in the sense amplifier inverter is 0.4 µm, the propagation delay of the sense amplifier inverter becomes the shortest. Consequently, Fig. 3.3 indicates that the optimum ratio of the transistor widths between nMOS and pMOS in the sense amplifier inverter is four. In this chapter, we used 0.4-µm nMOS and 0.1-µm pMOS for the sense amplifier inverter of 10T-S SRAM. For large-capacity SRAM, in terms of reducing the threshold-voltage (V_{th}) variation, the minimum size transistor should be avoided for use as a sense amplifier because the deterioration on a sense amplifier affects the access time for all memory cells connected to it.

3.4 10T Differential SRAM (10T-D SRAM)

Figure 3.4(a) presents a schematic of a 10T SRAM with differential read bitlines (RBL and /RBL) [27] (10T-D SRAM). Two nMOS transistors (N5 and N7) for the RBL and the other additional nMOS transistors (N6 and N8) for /RBL are appended to the traditional 6T SRAM. As is true also for the 8T SRAM, precharge circuits must be implemented on the RBL and /RBL.

Figure 3.4(b) depicts operation waveforms in the 10T-D SRAM in read cycles. The differential bitlines must be precharged to VDD by the start time of a clock cycle. To sense a difference voltage between the RBL and /RBL correctly, the difference voltage must be, at least, more than 50 mV [28–30].

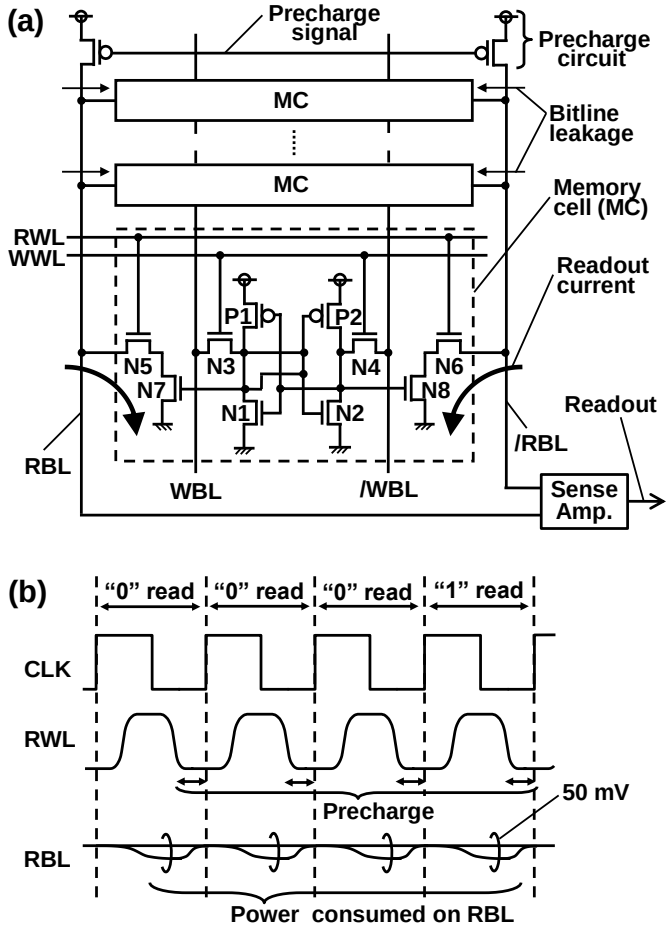

Fig. 3.4 10T SRAM with differential read bitlines (10T-D SRAM): (a) a schematic and (b) waveforms in read operation.

Figure 3.5 presents an illustration of a sense amplifier circuit for the 10T-D SRAM.

This is a commonly used latch type sense amplifier. The use of low-threshold-voltage transistors (P3-P5 and N3-N5) enables sensing of the differential voltage faster, although the precise control of the sense enable signal is needed [31] because timing generator circuits are easily affected by the V_{th} variation. Consequently, the differential voltage when the sense enabled signal is enabled is varied, which varies the readout power as well.

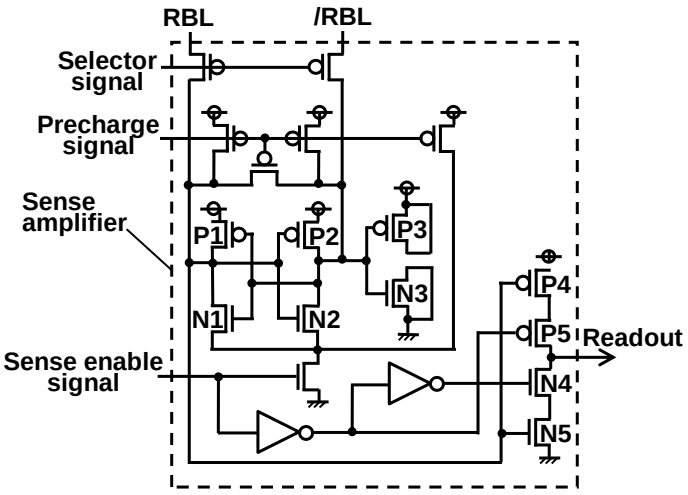

Fig. 3.5 Circuit schematic of a sense amplifier in the 10T-D SRAM.

3.5 Reducing the Number of Charge/Discharge Times

3.5.1 Application to Video Images

In the 10T-S SRAM, the charge–discharge power consumed on the LRBLs is proportional to the number of times that a datum flips (i.e., the number of transitions: "0" to "1" and "1" to "0") along the time axis. Therefore, we can exploit the 10T-S SRAM for video processing as well as the MJ SRAM because adjacent pixels have strong correlation with one another in a video image.

In the H.264 codec, the YUV format is adopted as a pixel datum. An example is presented in Fig. 3.6. One pixel comprises an 8-bit luma (Y signal) and 4-bit chroma (U and V signals). In this chapter, only luma data are considered. The most significant bits (MSBs) in consecutive data tend to be lopsided to either "0" or "1" with high probability, while in the least significant bits (LSBs), the values of the bits are random. In other words, the correlation becomes stronger in a more significant bit, which is well exploited in the MJ SRAM.

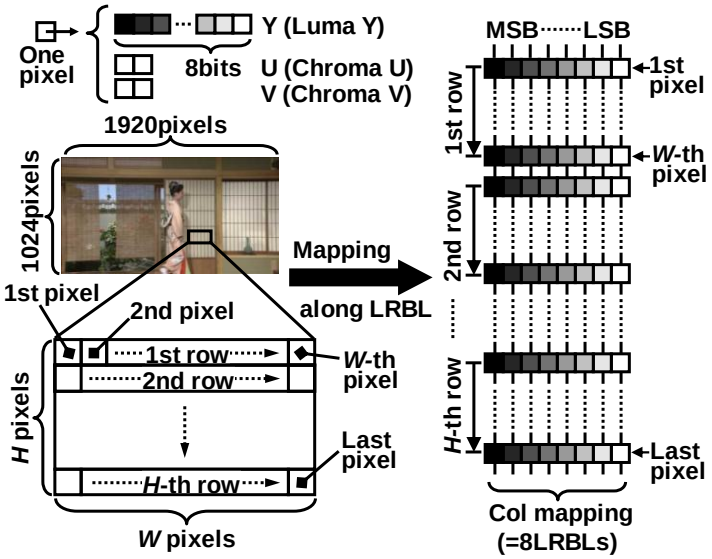

Fig. 3.6 Example of H.264 image data and its mapping onto eight LRBLs.

As described in Section 3.3, the power reduction on the LRBLs is theoretically expected because of the non-precharge scheme, even if input data are random. Furthermore, additional power reduction is promising because image data are lopsided to "0"s or "1"s with higher probability in a more significant digit. We exploit these characteristics in the 10T-S SRAM to reduce the LRBL power as well as the MJ SRAM.

3.5.2 Block Size Optimization

In this subsection, we discuss the optimum data mapping that uses the spatial correlation in an image. In a video image, the correlations among local pixels are presumed to be different in the vertical and lateral directions. It is important to determine the block size mapped onto an LRBL because a scan path affects the effective use of the spatial correlation and power. Assuming an H.264 encoder, we made a simulation under the condition shown in Table 3.1 to fix the block size. In the simulation, statistic analyses were conducted with the original images and reconstructed images, extracted from ten high-definition test sequences shown in Fig. 3.7: 'Bronze with Credit', 'Building along the Canal', 'Church', 'Intersections', 'Japanese Room', 'European Market', 'Yachting', 'Street Car', 'Whale Show', and 'Yacht Harbor'. The original image is encoded. Then its reconstructed image is generated in a local decoding loop. It is then used for motion estimation and motion compensation. The encoding process is depicted in Fig. 3.8.

Table 3.1 Simulation conditions in the H.264 encoder.

Profile	Main profile
Frame rate	30 fps
Bit rate	7.5 Mbps
Search range	±128 × ±128
Symbol mode	CABAC
JM version	9.8

Fig. 3.7 HD video sequences. Each sequence comprises 100 frames and 1920 × 1024 pixels.

Figure 3.6 portrays an example of the block size and its scan path. We set the number of pixels in a block to 256 because the search range is ±128 × ±128 in the H.264 encoder and a burst access over 256 pixels is possible if a full-search algorithm is considered. Therefore, in the simulation, a pixel block ($W \times H$ pixels) has 256 pixels. The scan path from the first pixel to the W-th pixel is mapped onto eight LRBLs.

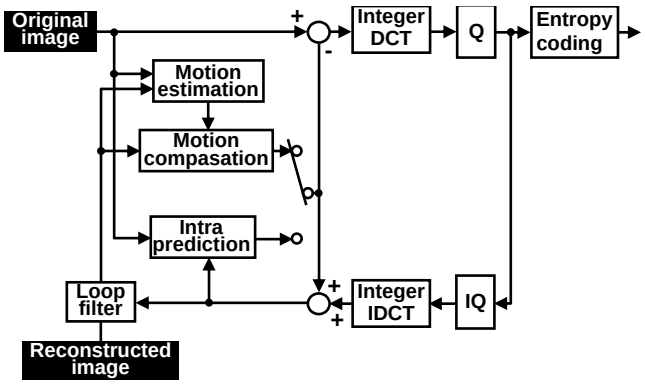

Fig. 3.8 H.264 encoding process.

Figure 3.9 presents a comparison of the transition possibilities (the normalized quantities of charge–discharge times) on an LRBL between the conventional 8T SRAM, MJ SRAM, and proposed 10T-S SRAM when the block size is changed. The values are the averages of 10 sequences. For both the original image and reconstructed image, the

block size of 256 × 1 pixels is optimum in terms of power reduction. The graph indicates that the 10T-S SRAM saves 73% of a dynamic power on an LRBL compared to the conventional 8T SRAM when the original image is read out.

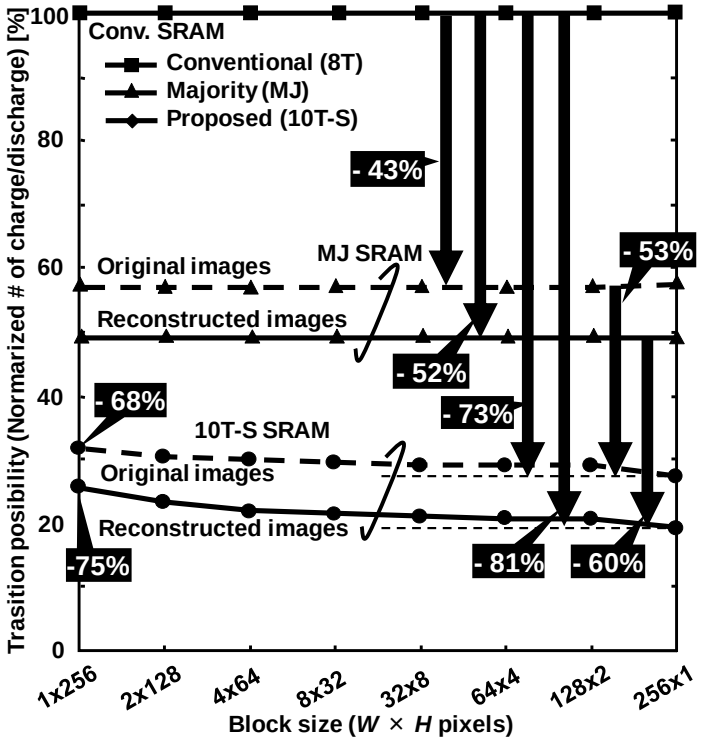

Fig. 3.9 Transition possibilities (the normalized quantities of charge–discharge times) on an LRBL between the conventional 8T, MJ, and proposed 10T-S SRAMs when a block size is changed.

The maximum power saving is achieved when a reconstructed image that has a stronger correlation than the original image is considered. The saving factor is extended to 81% compared to the conventional 8T SRAM, which indicates that the statistical characteristic of the reconstructed image is well exploited. It can be said that the proposed 10T-S SRAM is suitable for a real-time video codec such as MPEG2, MPEG4, and H.264 that require a large-capacity reconstructed-image memory.

3.6 Design in 90-nm Process Technology

3.6.1 Cell and Macro Layouts

Figure 3.10 depicts the layout patterns of conventional and proposed MCs in a 90-nm process technology. In addition, the transistor sizes are shown in this same figure.

Figure 3.10(a) portrays the conventional 8T MC layout. The schematic is portrayed in Fig. 3.1(a). The cell area is 3.15 × 0.76 μm^2, which is the smallest of the three. Because this memory cell frees an SNM, the driver transistors' (N1, N2) width can be minimized; then the load transistors' (P1, P2) length can be enlarged to extend the write margin. Therefore, the operation margin is sufficient at the nominal supply voltage of 1.0 V.

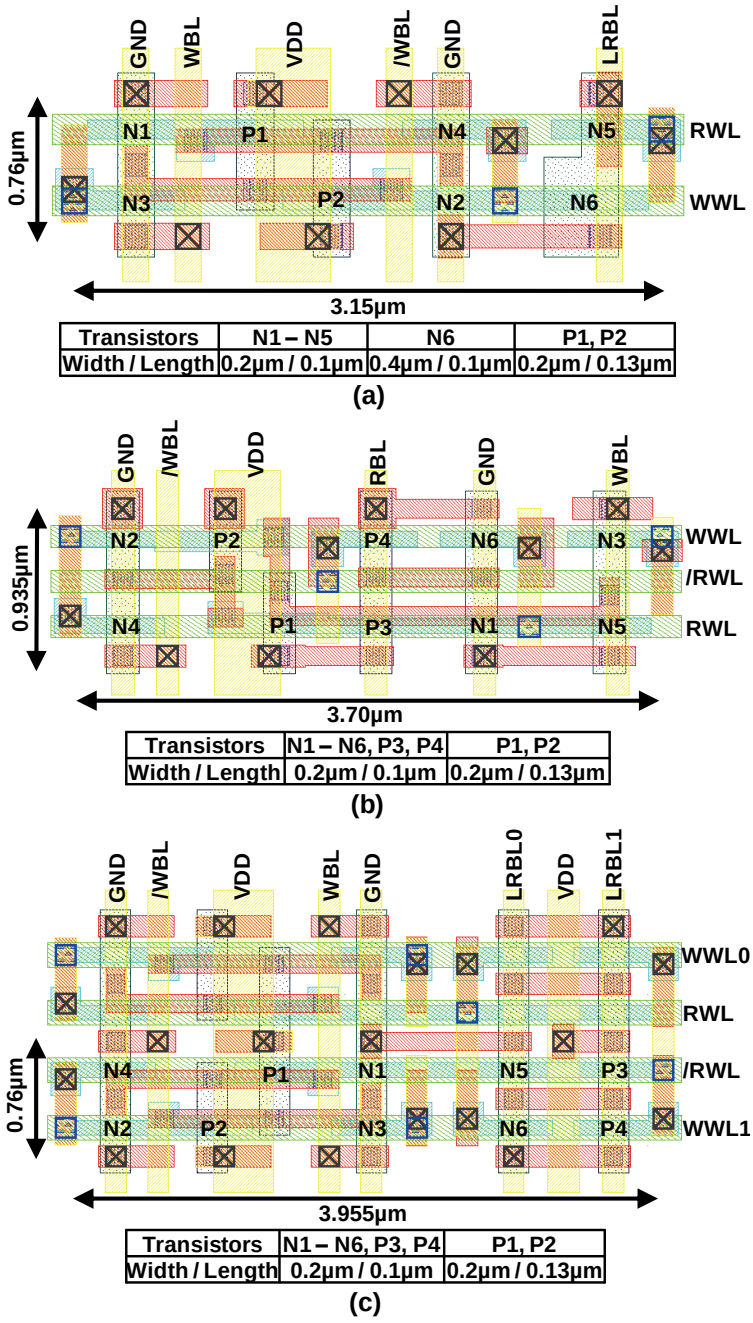

Fig. 3.10 Layouts of (a) the conventional MC, (b) the proposed MC without shared WL structure, and (c) the proposed MC with a shared WL structure.

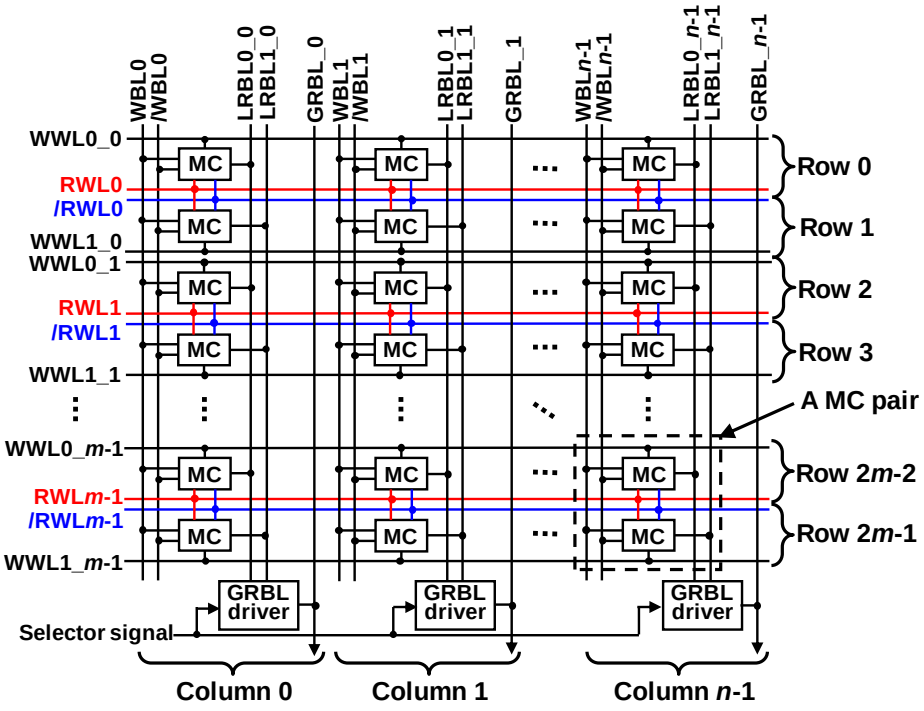

Fig. 3.11 Shared WL structure.

Figure 3.10(b) shows the proposed 10T-S MC layout. The schematic is portrayed in Fig. 3.2(a). The cell area is 3.70 × 0.935 μm^2. This MC, at 0.935 μm, is higher than the conventional one (0.76 μm) because the 10T-S MC requires three wordlines: WWL, RWL, and /RWL. The minimum metal pitch to align these three wordlines is larger than a transistor pitch. Therefore, the height of this 10T-S MC is restricted by the metal lines. The coupling noise between the wordlines would be larger than that in the conventional one because the metal pitch must be minimized for a small MC area.

We propose a shared WL structure to shrink the area of the proposed MC. Figure 3.11 presents the shared WL structure. A pair of a top and bottom MCs shares an RWL and its complementary signal: /RWL. Instead, two LRBLs must be interconnected in each column, as the figure shows. For instance, when RWL0 and /RWL0 are asserted, the MCs in Row 0 and Row 1 become active. The data stored in Row 0 are read out to the LRBL0 group (LRBL0_0, …, LRBL0_n-1), and the data stored in Row 1 are read out to the LRBL1 group (LRBL1_0, …, LRBL1_n-1). Consequently, the additional drivers are prepared to choose which data are read out to the global read bitlines (GRBLs); the GRBL driver selects either LRBL0 group or the LRBL1 group using the selector signal.

Figure 3.10(c) shows the layout of an MC pair with the shared WL structure. The cell area is 3.955 × 0.76 μm^2. By introducing the shared WL structure, the 10T-S MC can be

designed to be the same height as the conventional 8T MC because the RWL and the /RWL are shared by each MC pair. Therefore, the quantities of RWLs and /RWLs are reduced to a half that of the WWLs, which reduces the MC area overhead of the proposed 10T-S SRAM. Although, Fig. 3.10(b) shows that the MC height is restricted by the metal wordlines, the MC height with the shared wordline structure is restricted by the transistor pitch as depicted in Fig. 3.10(c). Therefore, the metal pitch of the wordlines in Fig. 3.10(c) is relaxed. The coupling noise between the wordlines can be reduced [32].

In the 10T-S MC, all transistors are aligned on two lines, so this MC layout improves lithographic quality and is better in terms of manufacturability than the MC without the shared wordline structure. Furthermore, no bent polysilicon pattern was found in the proposed 10T-S MC, which might reduce variations in transistors' finished dimensions. In the 10T-S MC pair, because each bitline is shielded by a VDD line or GND line, it is tolerant of coupling noise [32].

3.6.2 Delay Model of Read-Bitline RC Trees

In our proposed 10T-S SRAM, because of the additional pMOS transistor, MCs can fully charge–discharge each LRBL. However, the MC transistors are too small to charge–discharge each long RBL quickly. Therefore, in our design, we adjust the length of the RBLs which are charged–discharged by MCs in the hierarchical read-bitline structure (double-bitline structure: The LRBLs and GRBLs). The hierarchical read-bitline structure is effective to avoid a speed overhead of a single-bitline scheme, which is applicable to the 10T-S SRAM [21]. In our proposed shared WL structure, when an address is asserted, the quantities of the active MCs differ in write and read operations because, only in the read operation, the wordlines are shared. The hierarchical read-bitline structure also solves this addressing problem.

We model the BL structure to minimize a propagation delay from the LRBLs to the GRBLs. Elmore delays are obtainable node-by-node on the bitline: all resistances and all capacitances from the input node to the output node. Figure 3.12 shows a π-type RC model of the SRAM read port when the total number of bits on each GRBL is set to 512 and when each GRBL is divided into LRBLs by a factor N (N is a natural number) [33]. The respective widths of the LRBL and GRBL using the metal-1 and metal-2 lines are

set to 0.14 μm.

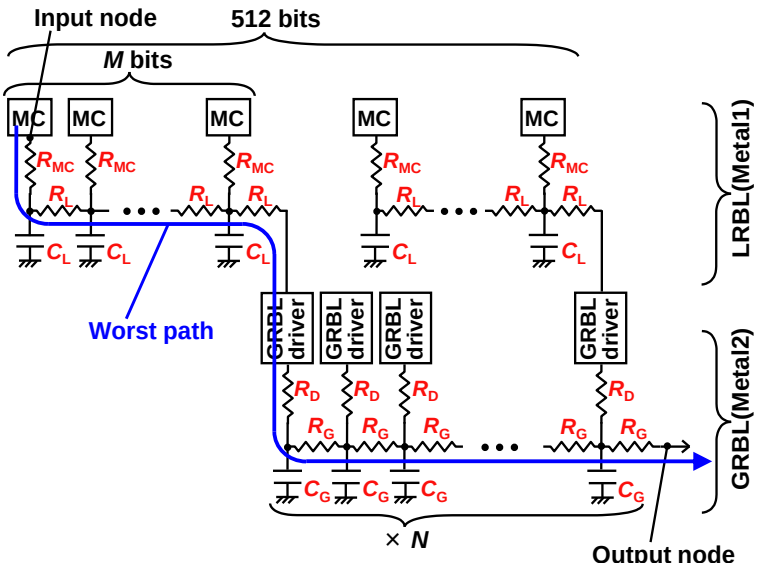

Fig. 3.12 A π-type RC model of the SRAM read port.

In Fig. 3.12, M, C_L, C_G, R_L, R_G, R_{MC}, and R_D respectively represent the number of bits on each LRBL, the capacitance of the LRBL per bit, the capacitance of the GRBL per 512/N bits, the resistance of the LRBL per bit, the resistance of the GRBL per 512/N bits, the output resistance of each MC to the LRBL, and the output resistance of each GRBL driver.

Table 3.2 Values of M, CL, CG, RL, RG, RMC, and RD, as obtained using the ASPLA 90-nm process parameters.

	Conventional 8T SRAM	Proposed 10T SRAM with shared WL structure
M [bits]	512 / N	256 / N
C_L [fF]	0.408	0.748
C_G [fF]	0.0546 × M + 2	0.1248 × M + 2
R_L [Ω]	0.0789	0.1580
R_G [Ω]	0.0789 × M	0.1580 × M
R_{MC} [Ω]	1.2 × 104	1.5 × 104
R_D [Ω]	2.5 × 103	2.5 × 103

Table 3.2 presents the values of M, C_L, C_G, R_L, R_G, R_{MC}, and R_D, obtained using the ASPLA 90-nm process parameters. The parameter of C_L presents the sum of a wiring capacitance and a drain capacity in an MC. The drain capacitance depends on a MC topology. When considered the conventional 8T MC in Fig. 3.1(a), the drain capacitance corresponds to the drain capacity of N6 only. Similarly, when considering the proposed

10T-S MC in Fig. 3.2(a), the drain capacitance is the sum of N6 and P4, so the C_L in the proposed MC is larger than that in the conventional 8T MC. The C_G parameter presents a sum of the wiring capacitance of the GRBLs and the drain capacitance of the GRBL driver. In this model, the drain capacity of the GRBL driver circuits equals 2 fF. The R_{MC} and R_D are obtained respectively by analyzing the transistors characteristic that connected to the LRBL and GRBL.

When the total number of bits on each GRBL and each LRBL is set to 512 and M, respectively, and when each GRBL is divided into LRBLs by a factor, N, the Elmore delay $\tau_{Elmore}(M, N)$ is expressed as shown below [34].

$$\begin{aligned}\tau_{\text{Elmore}}(M,N) &= R_{\text{MC}}\left(M\cdot C_{\text{L}}+N\cdot C_{\text{G}}\right)+R_{\text{L}}\sum_{k=0}^{M-1}\left\{k\cdot C_{\text{L}}+N\cdot C_{\text{G}}\right\}\\ &\quad +R_{\text{D}}\cdot N\cdot C_{\text{G}}+R_{\text{G}}C_{\text{G}}\sum_{k=0}^{N-1}k\\ &= R_{\text{MC}}\left(M\cdot C_{\text{L}}+N\cdot C_{\text{G}}\right)+R_{\text{L}}\left\{\frac{M(M-1)}{2}\cdot C_{\text{L}}+M\cdot N\cdot C_{\text{G}}\right\}\\ &\quad +R_{\text{D}}\cdot N\cdot C_{\text{G}}+R_{\text{G}}\cdot\frac{N(N-1)}{2}\cdot C_{\text{G}}\\ &= \frac{1}{2}R_{\text{L}}C_{\text{L}}\cdot M^{2}+\left(R_{\text{MC}}C_{\text{L}}-\frac{1}{2}R_{\text{L}}C_{\text{L}}\right)\cdot M+\frac{1}{2}R_{\text{G}}C_{\text{G}}\cdot N^{2}\\ &\quad +\left(R_{\text{MC}}C_{\text{G}}+R_{\text{D}}C_{\text{G}}-\frac{1}{2}R_{\text{G}}C_{\text{G}}\right)\cdot N+R_{\text{L}}C_{\text{G}}\cdot M\cdot N\end{aligned} \tag{3.1}$$

The values in Table 3.2 are substituted for (3.1); $\tau_{Elmore}(M, N)$ is obtained by calculation. Figure 3.13 shows $\tau_{Elmore}(M, N)$. When the total number of bits on each GRBL is set to 512, the optimum N is 8 in both the conventional and the proposed SRAMs.

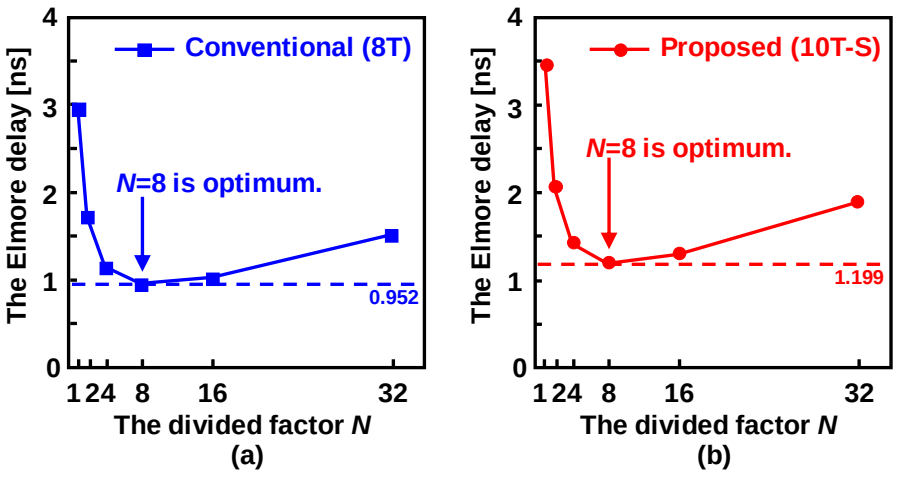

Fig. 3.13 Elmore delays by numeric calculation using ASPLA 90-nm process parameters. (a) Conventional 8T SRAM and (b) proposed 10T-S SRAM with the shared WL structure.

3.6.3 Chip Overview

Figure 3.14 shows a block diagram of the proposed SRAM. A hierarchical read-bitline structure, already discussed in the previous subsection, is applied. A GRBL driver drives a GRBL with a block selector signal from the X decoders.

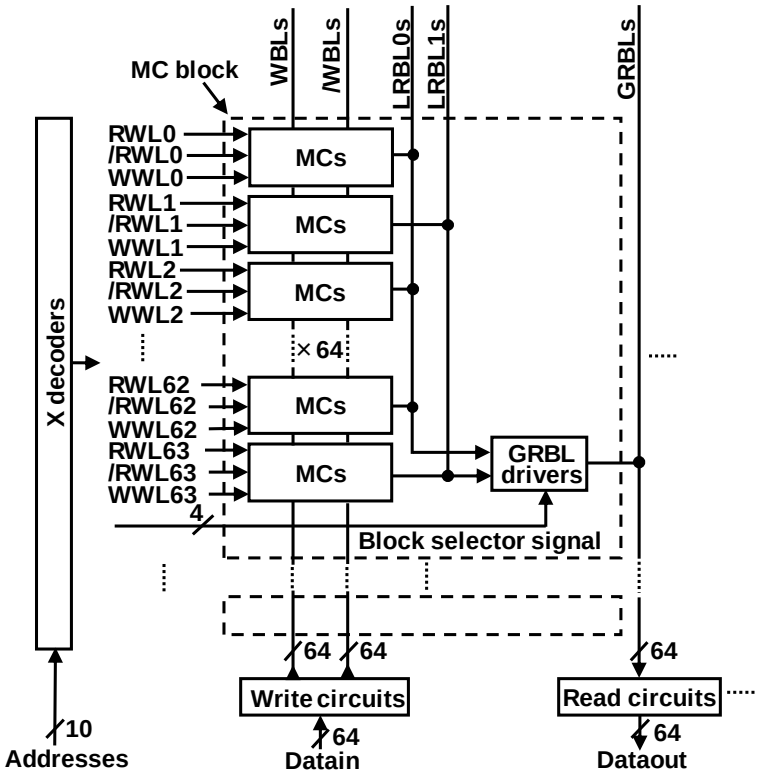

Fig. 3.14 Block diagram of a memory cell block in the proposed 10T-S SRAM.

Figure 3.15 presents a chip micrograph of the 64-kb 10T-S SRAM in a 90-nm process technology. The MC area, which comprises 10 transistors, is 3.96 × 0.76 μm^2. An MC block is 64 words by 64 bits, into which two 256-pixel blocks can be put.

Figure 3.16 portrays operation waveforms of the proposed 10T-S SRAM when "0" and "1" are read out. After a block selector signal is asserted, a GRBL is discharged–charged as Dataout. The access times at the "0" and "1" readouts are, respectively, 0.93 ns and 1.16 ns. The "0" readout is faster than the "1" readout because nMOS transistors in the GRBL driver and the read circuit are stronger than the pMOS ones. Figures 3.16(a) and 3.16(b) show that the proposed 10T-S SRAM shortens the cycle time to 1.16 ns because of the precharge-less structure. This access time corresponds to an 862-MHz (= 1/1.16 ns) operation because the proposed 10T-S SRAM requires no precharge period.

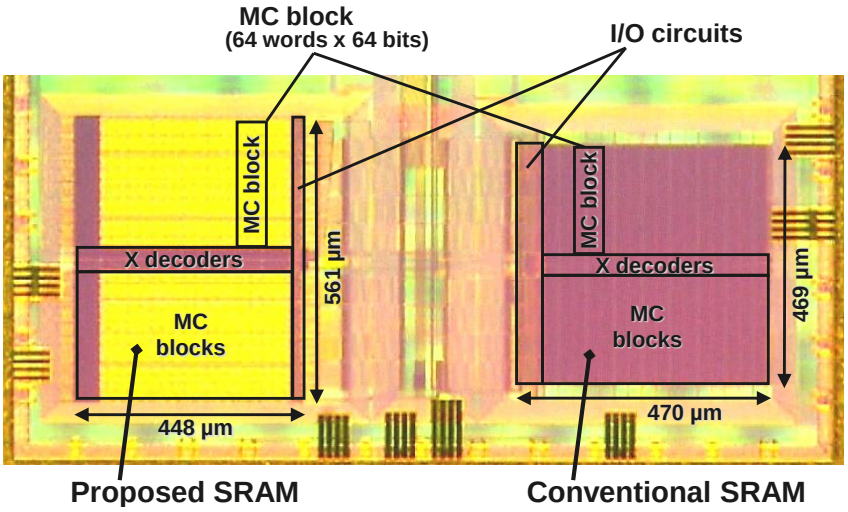

Fig. 3.15 Chip micrograph of the proposed 10T-S SRAM and the conventional 8T SRAM in a 90-nm process technology. The total memory size of each SRAM is 64 kb.

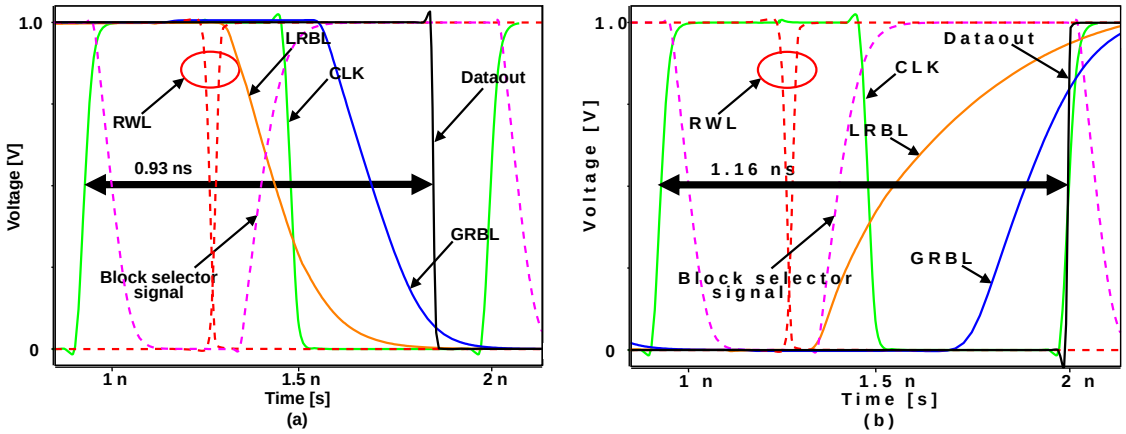

Fig. 3.16 Operation waveforms of the proposed 10T-S SRAM when (a) "0" and (b) "1" are read out in a 90-nm process technology (CC corner, 25°C).

An area comparison of a conventional 8T SRAM, an MJ SRAM, and the proposed 10T-S SRAM is portrayed in Fig. 3.17, showing that the SRAM areas include three parts: The MC part, the read and write circuit part, and the other part. The MC part represents the entire MC array area, and the additional flag bits in MJ SRAM. The read and write circuit part includes the write drivers, precharge circuits, sense amplifiers, and the GRBL drivers in the proposed 10T-S SRAM. The other part contains the address decoders, word line drivers, flip-flops for input, data bus, and timing control circuits. The area overhead in the proposed 10T-S SRAM is 14.4% because two pMOS transistors are added to the conventional 8T MC. However, the read and write circuits

are smaller than the conventional SRAM by 1% because of elimination of the precharge and bitline keeper circuits.

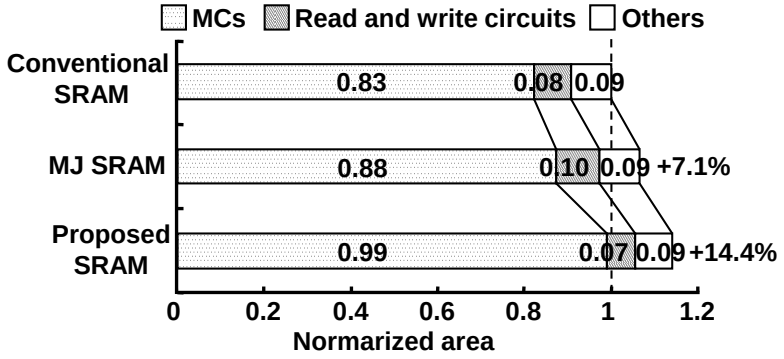

Fig. 3.17 Area comparison of 64-kb SRAMs in a 90-nm process technology.

3.6.4 Operating Frequency and Supply Voltage

As described above, no precharge period exists in the proposed 10T-S SRAM, which can shorten the cycle time compared to those of other precharge-type SRAMs, thereby providing higher performance in terms of the operating frequency. Furthermore, in the proposed 10T-S SRAM, the readout speed is fundamentally improved by eliminating the bitline keeper, as described in Section 3.3. Furthermore, using the shared WL structure, we can set the number of bits on each LRBL to half of the conventional 8T SRAM. Despite the additional pMOS transistors, which increase the amount of the LRBL capacitance, the proposed 10T-S SRAM can operate faster than a conventional one. Figure 3.18 depicts the frequency dependence on supply voltage in simulations. At a supply voltage of 1 V, the proposed 10T-S SRAM improves the operating frequency by 315 MHz (65% faster) compared with the conventional 8T SRAM. In other words, the proposed 10T-S SRAM can run at a lower supply voltage when an operating frequency is the same as others. In the conventional 8T SRAM and MJ SRAM, the bitline keepers hinder low-voltage operation, as described in Section 3.2. In contrast, the proposed 10T-S SRAM functions at a lower voltage, this greatly reduces the power requirement because the dynamic power is proportional to the square of a supply voltage. At an operating frequency of 300 MHz, the proposed 10T-S SRAM operates properly at 0.69 V, whereas the MJ SRAM does not operate properly below 0.85 V.

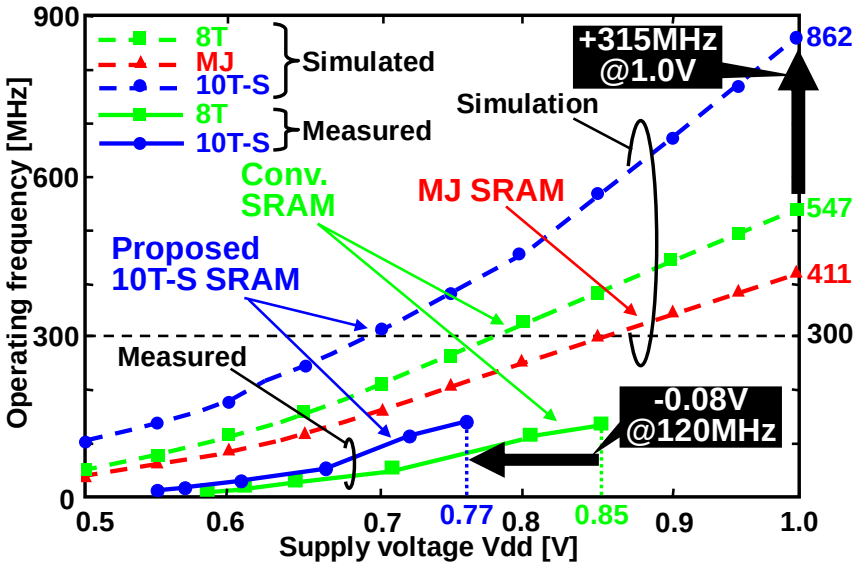

Fig. 3.18 Operating frequencies versus supply voltage with 90-nm process technology. Dotted lines show simulation results, and solid lines show the measurement results in this frequency comparison with conventional architecture.

Figure 3.18 also shows the frequency dependence on supply voltage in the measurement. The measured operating frequency is not greater than 120-MHz operation because of LSI tester limitations. According to the measured results, at an operating frequency of 120 MHz, the proposed 10T-S SRAM operates properly at 0.77 V, although the conventional 8T SRAM does not work below 0.85 V.

3.6.5 Power

In the proposed 10T-S SRAM, the power overhead is obviated in a write operation because the 6T structure at the write port is identical to that of the conventional one. However, in a read operation, the additional pMOS transistor, P4 in Fig. 3.2(a), increases the LRBL capacitance by 83%. However, the shared WL structure reduces the number of bits on each LRBL to half that of the conventional SRAM. Therefore, the speed overhead by the LRBL capacitance does not exist. Furthermore, the number of charge–discharge times is halved in comparison to the conventional case. Thereby, the readout power is theoretically reduced in the proposed SRAM even if data are random.

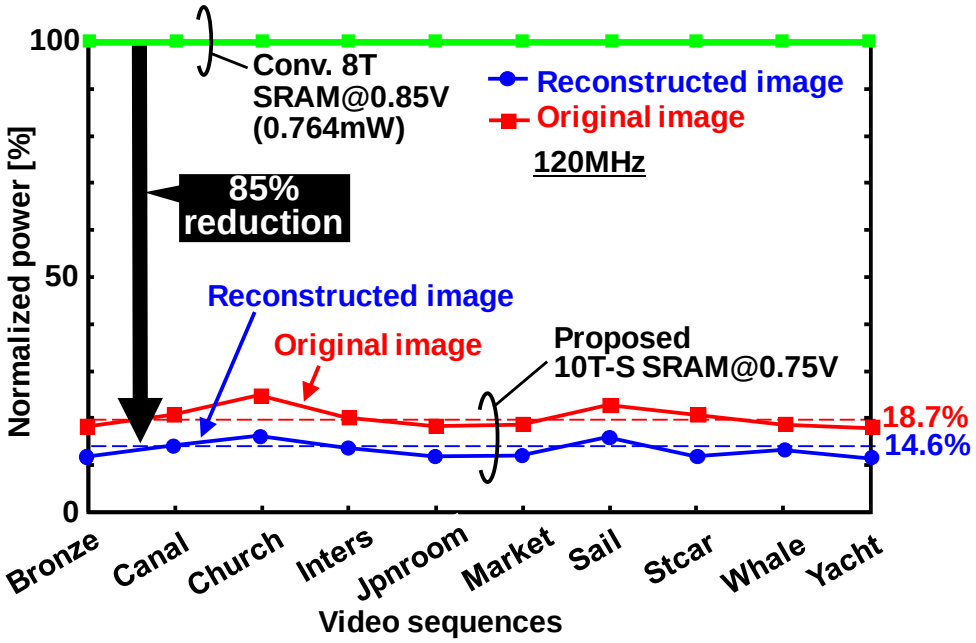

Fig. 3.19 Measured readout power at 120 MHz when reading the original images and reconstructed images.

Figure 3.19 presents comparisons of the measured readout powers when varying the content stored in the SRAMs. For video memory, power reduction in a read operation is important because the readout is performed more frequently than a write-in. The supply voltages are set to 0.85 V and 0.75 V in a conventional 8T SRAM and the proposed 10T-S SRAM, respectively, based on Fig. 3.18. In the conventional 64-kb 8T SRAM, the readout power is measured as 0.764 mW on average of the 10 video sequences described in Subsection 3.5.2, at the supply voltage of 0.85 V and the frequency of 120 MHz. Our proposed SRAM saves 85% of a total readout power at the lower supply voltage when it is used as a reconstructed image buffer. Its power dissipation is 14.6% on average.

Figure 3.20 presents a comparison of the readout power in the conventional 8T SRAM and the proposed 10T-S SRAM when the supply voltage is changed according to the operation frequencies. The proposed 10T-S SRAM reduces the readout power by 65% compared with the conventional 8T SRAM at the 120-MHz operation in the measurement if random data are used. That savings factor increases to 79% compared to the conventional 8T SRAM if the memory content is an H.264 original image. In a reconstructed image, we can maximize the power improvement, where we can save 85% of the readout power.

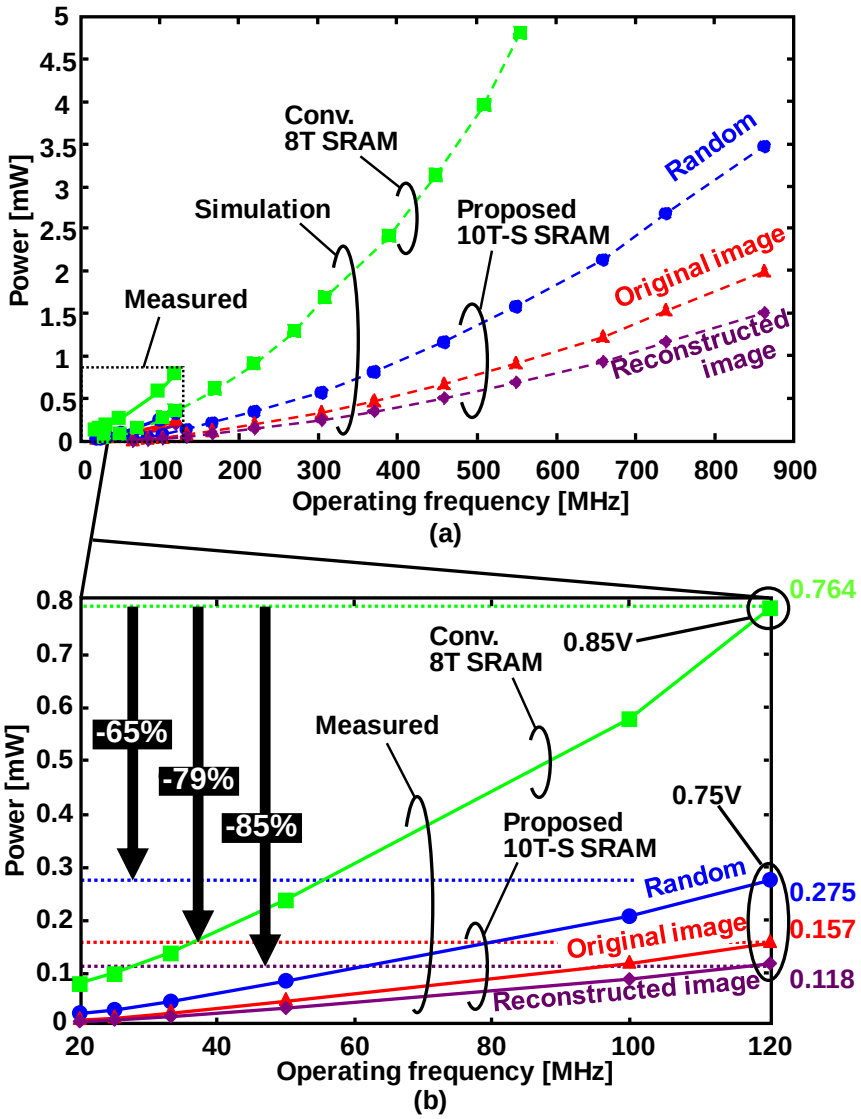

Fig. 3.20 (a) Readout power versus operating frequencies in a 90-nm process technology. (b) Magnified view.

3.7 Design in 45-nm Process Technology

3.7.1 Cell and Macro Layouts

Figure 3.21 portrays the layouts of the dual-port SRAMs of three kinds in a 45-nm process technology. Schematics are shown in the previous figures. The areas of the 8T, 10T-S, and 10T-D SRAM cells are, respectively, 1.55 × 0.41 μm^2, 1.97 × 0.41 μm^2 and 1.95 × 0.41 μm^2. In 8T and 10T-D SRAMs, we use 0.2-μm nMOS drive transistors in the read port because the drive current of 0.2-μm nMOS is larger than that of 0.1-μm nMOS. Furthermore, for these additional read ports, the poly space rule restricts the

memory cell width whether we use 0.2-μm nMOS or not.

In Fig. 3.21, our memory cell design is based on a logic-design rule. When considering an SRAM-design rule, we can use a shared contact to an inverter couple, which saves the height of the memory cell and which engenders shorter RBL and faster read operation. The effects of adopting the SRAM-design rule are absolutely identical for the cells of three kinds; and the tendency of performance comparison does not vary in terms of the logic-design rule or with the SRAM-design rule.

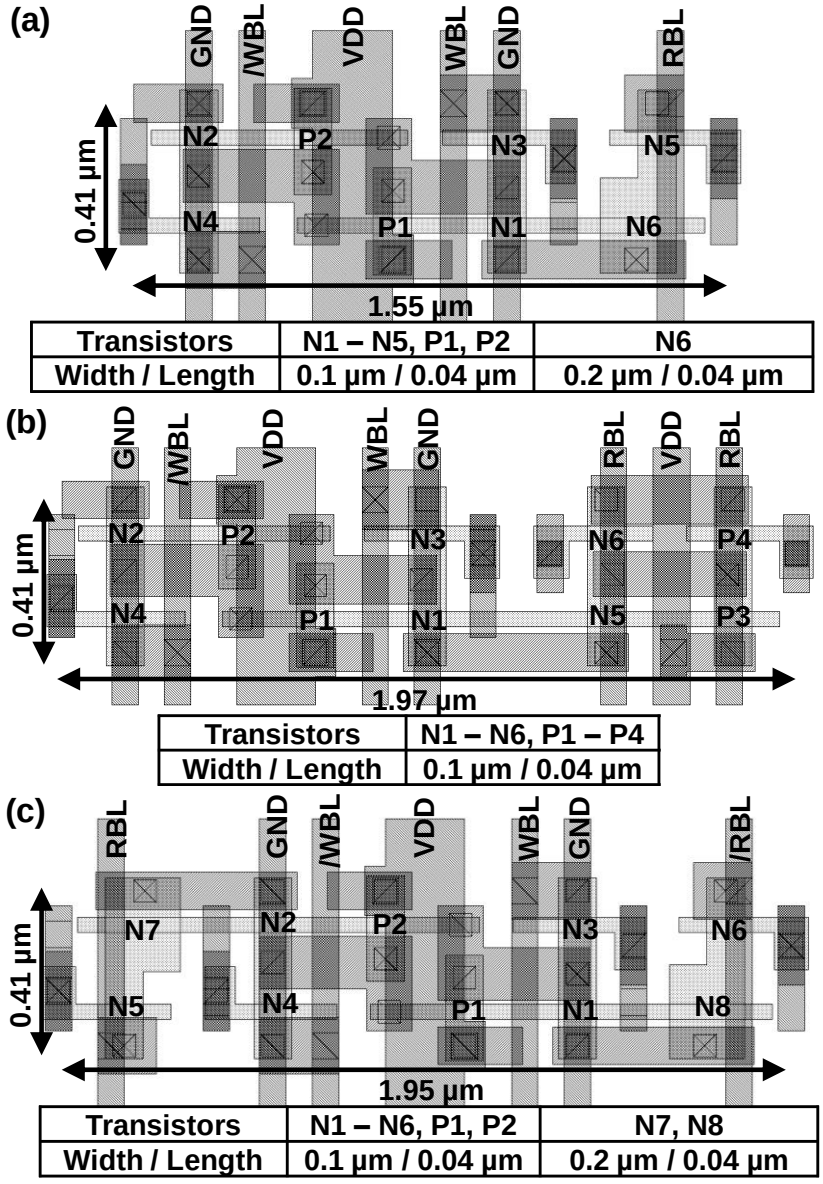

Fig. 3.21 Cell layouts of (a) 8T, (b) 10T-S, and (c) 10T-D SRAMs, in a 45-nm process technology.

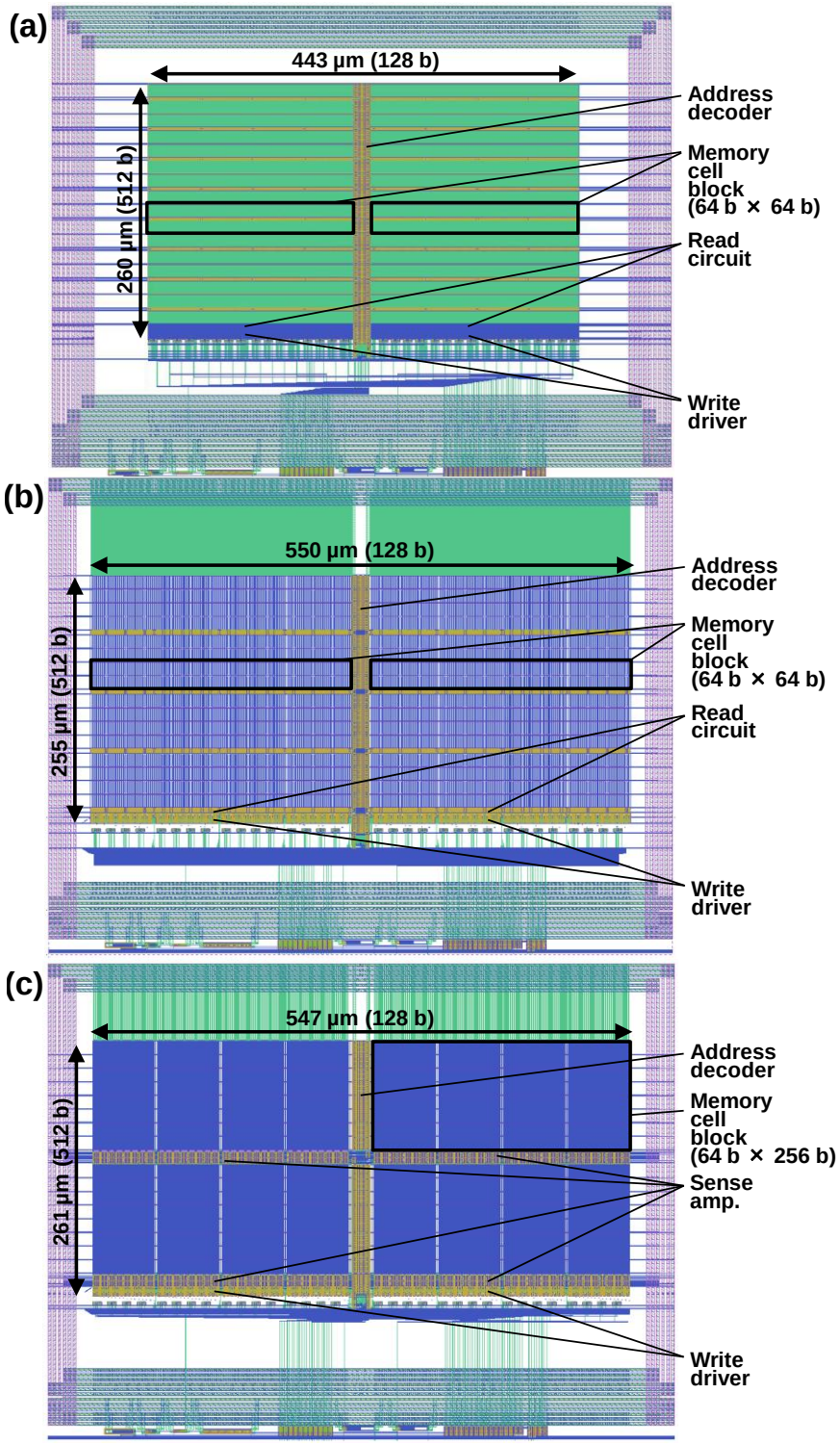

Fig. 3.22 Macro layouts of (a) 8T, (b) 10T-S, and (c) 10T-D SRAMs, in a 45-nm process technology. The total memory capacity of each macro is 64 kb.

We also designed 64-kb SRAM macros in the 45-nm process technology for macro-level area comparison. Figure 3.22 shows the macro layouts. The core sizes of

the 8T, 10T-S, and 10T-D SRAM macros are, respectively, 260 × 443 μm^2, 255 × 550 μm^2, and 261 × 547 μm^2. Each macro is 64 kb (128 b × 512 b). The 8T and 10T-S SRAM macros have 16 memory cell blocks (64 b × 64 b), and the divided factor between local RBL and global RBL is eight, which has been optimized using the Elmore delay model [26]. The 10T-D SRAM macro has four memory cell blocks (64b × 256 b) and the divided factor between local RBL and global RBL is two. The 8T SRAM macro is the most area-efficient because it has the lowest transistor count. The 10T-D SRAM macro has, compared to the 10T-S SRAM, a 2% area overhead that is attributable to differential sense amplifiers and precharge circuits.

3.7.2 Operating Frequency versus Supply Voltage

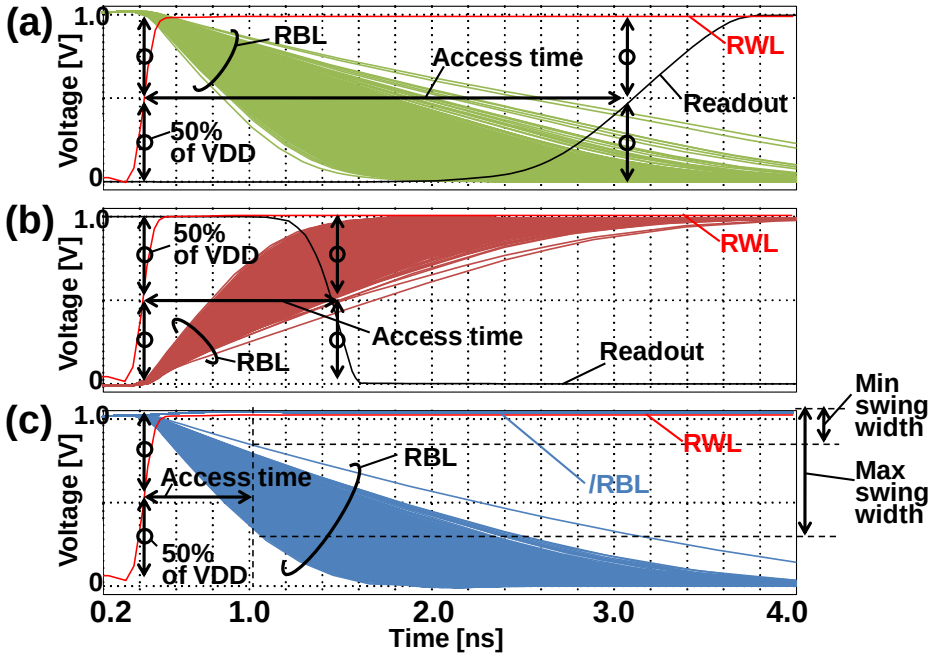

Fig. 3.23 Operation waveforms of (a) 8T, (b) 10T-S, and (c) 10T-D SRAMs at the SS corner (25°C).

To obtain an operating frequency, we conducted Monte Carlo simulations considering the threshold voltage variation of each transistor. The number of Monte Carlo samples was 20,000, which is sufficient to assess the local variation with 20-kb SRAM. When considered for an SRAM having more than 20-kb capacity, the Monte-Carlo samples must be increased according to the capacity. The standard deviation σ for V_{th} variation of nMOS and pMOS are, respectively, $\sigma[\mathrm{V}] = 3.6/\sqrt{L_{\mathrm{eff}}[\mathrm{nm}] \cdot W_{\mathrm{eff}}[\mathrm{nm}]}$ and

$\sigma[\mathrm{V}] = 2.7/\sqrt{L_{\mathrm{eff}}[\mathrm{nm}] \cdot W_{\mathrm{eff}}[\mathrm{nm}]}$, which are obtained from the Pelgrom plots based on ITRS 2005 [3]. In the Monte Carlo simulation, all transistors of an accessed memory cell and sense amplifier inverter are given V_{th} variation according to their L_{eff} and W_{eff}.

Figure 3.23 portrays operating waveforms for the SRAMs of three kinds. In the figure, we adopt the SS corner model to simulate the worst-case delay. As shown in Section 3.3, in the 10T-S SRAM, the sense-amplifier circuit optimization shows that the charging time on RBL is 1.0 ns and the discharging time on RBL is 0.99 ns. Consequently, a "1" readout is 0.01 ns longer than a "0" readout; for a 10T-S SRAM, the worst case in a read operation is a "1" readout (Fig. 3.23(b)).

The following are criteria used to calculate the access times:

- In the 8T SRAM, the access time is a period from the time at which an RWL rises to VDD/2 to the time at which output of the sense amplifier is charged up to VDD/2.
- In the 10T-S SRAM, the access time is a longer one: periods from a time at which an RWL rises to VDD/2 to a time at which an output of the sense amplifier is charged up to 50% of VDD, or the period from the time at which an RWL rises up to VDD/2 to the time that an output of the sense amplifier is discharged down to VDD/2.
- In the 10T-D SRAM, the access time is the period from a time at which an RWL rises to VDD/2 to the time at which the differential voltage between an RBL and /RBL is expanded to 50 mV, 100 mV, or 200 mV.

In all SRAMs, the worst cell with the worst threshold-voltage combination determines the critical-path delay and operating frequency. Figure 3.24 shows characteristics of the operating frequency when the VDD is changed. The operating frequency is calculated as the inverse of a cycle time, which is the sum of a bitline charge–discharge time plus propagation delays in decoder circuits, a wordline, and sense amplifier circuits. The propagation delays in decoder circuits and the wordline are set as equal for all SRAMs. In this simulation of the operating frequency, the precharge periods in the 8T and 10T-D SRAMs are not considered because they can be overlapped completely with the decoder operation. The quantities of memory cells connected to a local RBL and a sense amplifier circuits are set to 64 for 8T and to 10T-S SRAM and

256 for 10T-D SRAM. The sense amplifier circuits connected to a global RBL. In the simulation, the stored datum of accessed memory cell and the other memory cells are set as opposite to consider worst cell leakage from un-accessed memory cells to the local RBL. The metal capacitances, according to the wire length, are appended to the local RBL and the global RBL. In the simulation, all transistors of an accessed memory cell and sense amplifier inverter are given the worst V_{th} combination according to 20,000-sampled Monte-Carlo simulation.

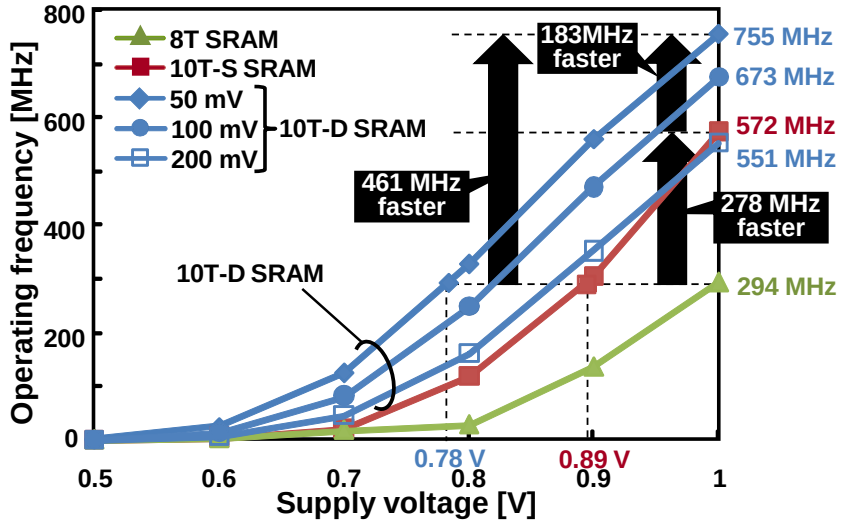

Fig. 3.24 Operating frequencies when a supply voltage is changed at the SS corner (25℃).

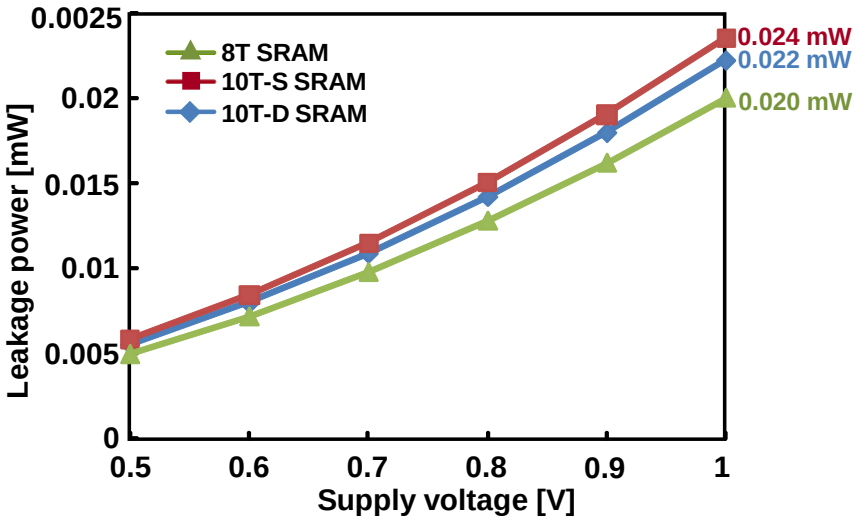

Fig. 3.25 Leakage power comparison of 8T, 10T-S, and 10T-D SRAMs at the CC corner (25℃).

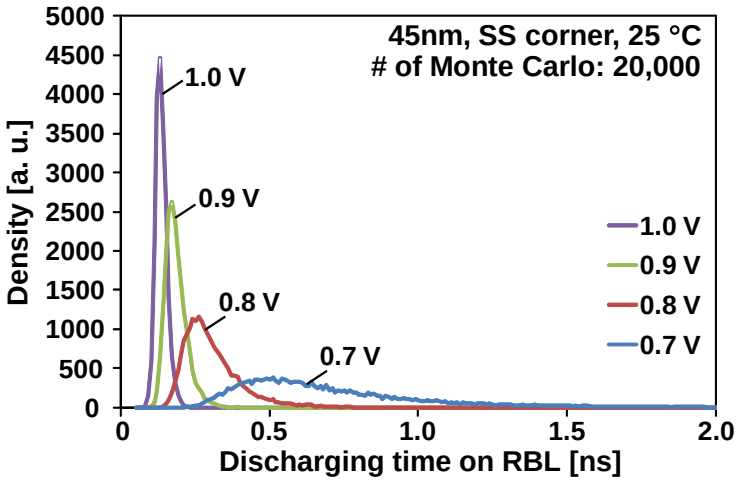

Fig. 3.26 Density function of discharging time on RBL variation of 10T-D SRAMs.

At supply voltages of 1.0 V, the 8T, 10T-S, and 10T-D SRAMs can run respectively at 294 MHz, 572 MHz, and 755 MHz. The maximum 755 MHz is achieved in the 10T-D SRAM at a differential voltage of 50 mV. Consequently, probably the small differential voltage of 50 mV achieves high-speed operation. However, as described in Section 3.4, in the 10T-D SRAM, even if the sense point is targeted to 50 mV, most cells sink more than 50 mV on the bitline. Eventually, the differential voltage gives a large value at a low-voltage operation. Although the additional transistor (P4) is appended in the 10T-S SRAM (see Fig. 3.2(a)) and although it increases the RBL capacitance, the 10T-S SRAM is faster than the 8T SRAM because it obviates the precharge circuit and the keeper circuit.

3.7.3 Power

Figure 3.25 presents a comparison of the leakage power in a 45-nm process technology when the stored data of 64 kb are random. The 8T SRAM cell has the lowest leakage power of the three because it has the fewest transistors. The 10T-S SRAM consumes the most leakage power.

Figure 3.26 portrays a density function of discharging period of 10T-D SRAM from the time at which an RWL rises to VDD/2 to the time at which a differential voltage between RBL and /RBL is expanded to 50 mV when the number of Monte-Carlo samples is set to 20,000. The figure shows that, for the 10T-D SRAM, the discharging time variation deteriorates concomitantly with the decrease in the supply voltage. To ensure the statistically weak cell operation, the sense enable signal comes to step away

from mean timing as the supply voltage is decreased. For example at 0.7-V operation, the worst cell needs 5.98 ns to get 50-mV differential voltage, although the mean discharge time is 0.717 ns. This is a 5.98–0.717 = 5.263 ns mismatch, which engenders much larger bitline amplitudes than 50 mV. For a 10T-D SRAM, this sense enable timing mismatch becomes considerably large as the supply voltage is decreased because the σ value of this density function expands as the supply voltage is decreased. We conducted the cyclopedic simulation and statistical analysis at operation voltages of 0.78 V, 0.7 V, 0.6 V, and 0.5 V to obtain the mean bitline amplitudes and readout power. The results are, respectively, 576.4 mV, 662.3 mV, 599.8 mV, and 499.9 mV at 0.78 V, 0.7 V, 0.6 V, and 0.5 V. These results indicate that, at low-voltage operation, the 10T-D SRAM needs an almost full-swing readout in spite of its differential operation.

Figure 3.27 presents a comparison of the readout powers in the 8T, 10T-S, and 10T-D SRAMs. Actually, VDD is changed in the lines, according to Fig. 3.24. The 10T-S SRAM uses the least power because the transition possibility of the RBL is 50% when a sequence of random data is considered. However, in the 10T-D SRAM, as the supply voltage is decreased, the average voltage differential between the RBL and /RBL becomes greater than 80% of VDD, as described above, even if the sense point is set to 50 mV. The readout power in the 10T-S SRAM is 25% lower than that of the 10T-D SRAM at the operating frequency of 294 MHz when random data are considered. The saving factor is maximized to 63% if the readout data have statistical similarity to H.264 reconstructed image data [22]. For 8T SRAM, a power saving scheme proposed with majority logic and data-bit reordering [35] can save 28% readout power when image data are considered.

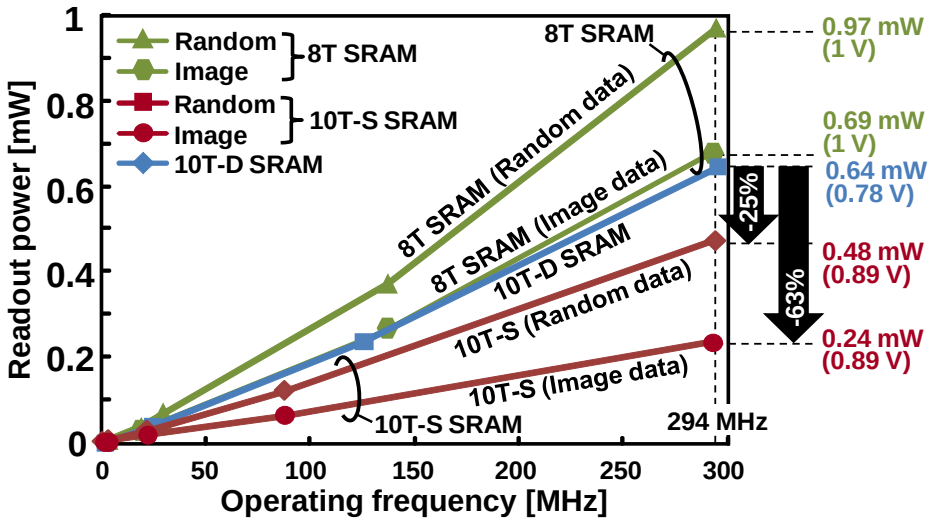

Fig. 3.27 Readout power versus operating frequencies in a 45-nm process technology at the CC corner (25°C).

3.8 Summary

As described in this chapter, a low-power non-precharge-type dual-port SRAM for video processing that exploits statistical similarity in images has been proposed. To minimize the charge–discharge power on a read bitline, the proposed memory cell (MC) has 10 transistors (10T) comprising the conventional 6T MC, a readout inverter and a transmission gate for a read port. In addition to three incorporated wordlines, we have proposed a shared wordline structure, with which the vertical cell size of the 10T MC is fitted to the same size as the conventional 8T MC. The readout inverter fully charges/discharges a read bitline. For that reason, no precharge circuit exists on the read bitline. Consequently, power is not consumed by precharging, but is consumed only when a readout datum is changed. This feature is suitable for use in video processing because image data have a spatial correlation and similar data are read out in consecutive cycles. In addition to power reduction, the prechargeless structure shortens the cycle time by 38% compared with the conventional 8T SRAM because the structure obviates a precharge period. This feature, in turn, demonstrated that the proposed SRAM operates at a lower voltage, which realizes further power reduction. Compared to the conventional 8T SRAM, the proposed SRAM reduces the charge–discharge possibility to 19% (81% saving) on the bitlines.

Measurement results confirmed that the proposed 64-kb video memory in a 90-nm process achieves an 85% power saving on the read bitline when regarded as an H.264

reconstructed image memory. The area overhead of the 10T single-end SRAM is 14.4% of the overhead taken by the 8T SRAM. In 45-nm process technology, the 8T SRAM has the lowest transistor count; it is also the most area-efficient. However, the readout power becomes large and the access time increases because of peripheral circuits. The 10T single-end SRAM can reduce the readout power by 74%. The operating frequency is improved by 195%, over the 8T SRAM. However, the 10T differential SRAM can operate fastest (256% faster than the 8T SRAM) because its small differential voltage of 50 mV achieves high-speed operation. In terms of the power efficiency, however, the readout current is affected by the threshold-voltage (V_{th}) variation and the timing of sense cannot be optimized singularly among all memory cells using 45-nm technology. The readout power remains 34% lower than that of the 8T SRAM (33% higher than the 10T single-end SRAM). Moreover, its operating voltage is the lowest of the three. The 10T single-end SRAM always consumes less readout power than the 8T or 10T differential SRAM.

Chapter 4 Two-Port SRAM Design for DVFS

This chapter presents a dependable dual-port SRAM with 9T/18T bitcell structure and two operating modes: a 9T normal mode and an 18T dependable mode. The 9T bitcell has an outside single-ended bitline as a dedicated read port along with a pair of conventional differential inside bitlines. Therefore, the 18T bitcell has two differential pairs of the outside bitlines and inside bitlines. For the dedicated read port, the 18T bitcell can exploit a differential sense amplifier operating at low voltage, but the 9T bitcell must have a single-ended readout inverter at high voltage. To achieve the 9T/18T SRAM architecture, an interleaved bitline scheme is incorporated for the dedicated read port. The 9T/18T dual-port SRAM can scale its speed, operating voltage, and power dynamically by combining two bitcells for one-bit information.

We designed and fabricated the proposed SRAM using a 65-nm process. The measurement results show that the dependable read mode using the pair of the single-ended bitlines can reduce the operation voltage to 0.45 V at 1 MHz because of the disturb-free read port, although the dependable read mode using the inside bitlines needs 0.54 V at the same frequency.

4.1 Introduction

The minimum feature size in transistors continues to decrease along with the advance of process technology, which achieves higher density and lower cost. Technology scaling, however, increases the threshold-voltage (V_{th}) variation of transistors mainly because of a random dopant fluctuation (RDF). Consequently, in the recent deep submicron era, it is important to design SRAM with both read and write margins considering the V_{th} variation tolerance [3, 36]. Therefore, dependable computing systems have been specifically examined because silicon LSIs support social infrastructure to a remarkable degree. In addition to the V_{th} variation, the advanced process technology tends to cause accidental errors such as soft errors and negative bias temperature instability (NBTI), more frequently. Furthermore, some errors might persist from the design, manufacturing, or test phase. Presumably, it is almost impossible to eliminate these human-induced errors perfectly from a future complicated LSI: a product will invariably be shipped with some errors, and will subsequently malfunction

accidentally. We no longer expect error-free LSIs with sufficient operating margins.

Reliability varies according to operating conditions (speed, supply voltage, temperature, and even altitude corresponding to a soft error). Therefore, it is desirable to improve the reliability dynamically on worse conditions. Furthermore, necessary reliability depends on the application software, which indicates that the reliability should be changed in accordance with the application.

In light of this background, we propose an SRAM that can control its reliability and speed dynamically. An SRAM has recently dominated operating margins of a chip because of numerous transistors [37−41]. The proposed SRAM can change the quality of its information in terms of reliability, speed, and power.

4.2 Dependable SRAM: overview

Operating conditions affect the reliability of an SRAM, although the reliability depends on application software that uses the SRAM. For example, an encryption program and a screen saver program demand different levels of reliability. Therefore, reliability can be chosen according to the operating conditions and application.

However, to conserve power used for an SoC, dynamic voltage and frequency scaling (DVFS) that adaptively controls an operating frequency and supply voltage has been implemented [11]. However, an SRAM presents the possibility of not working correctly at a minimum operating voltage because operating margins of memory cells are degraded attributable to V_{th} variation of MOSFETs as a fabrication technology is scaled down. Therefore, in minimum voltage operation, it is necessary to maintain reliability.

In the dependable SRAM, the SRAM reliability can be changed dynamically on a block-by-block basis, as illustrated in Fig. 4.1. In blocks with typical dependability (Blocks 0–3), assignment is usually that by which one memory cell has one bit. However, in high-dependability blocks (Blocks 4 and 5), one-bit information is stored in two memory cells by combining a pair of memory cells. This arrangement yields high dependability, but the memory capacity becomes half of that in the high-dependability blocks.

However, this dynamic switching between the typical dependability and high dependability opens up new resource allocation in an SRAM. For instance, an operating system (OS) can allocate an encryption program to the high-dependability block.

Application software can also change the reliability of its data by a system call. Encryption data or personal information should be in the high-dependability block. If memory utilization of programs and data is 50% or less, the high-dependability mode can be exploited aggressively by the OS without the memory-capacity overhead. A small code with small data always runs in the high-dependability mode.

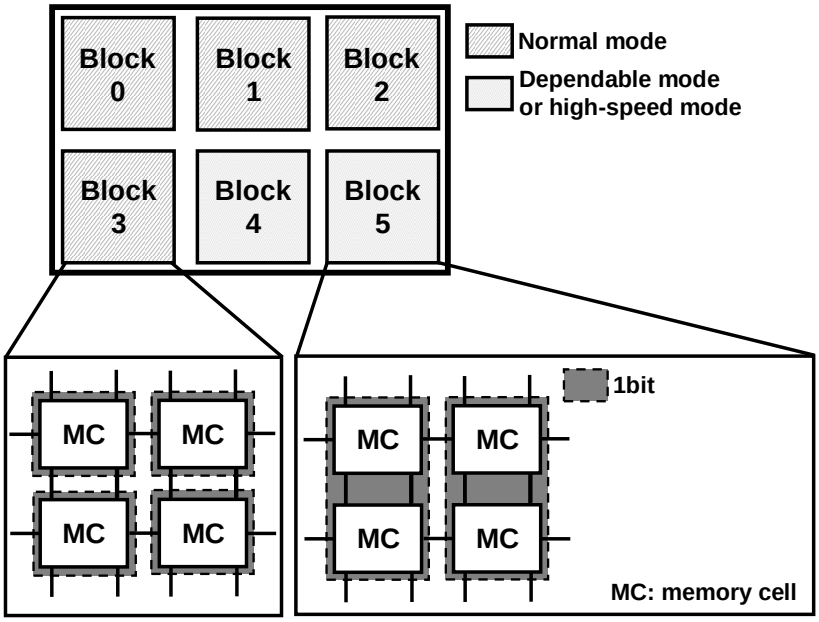

Fig. 4.1 Dependable SRAM.

4.3 7T/14T SRAM

The 7T/14T SRAM presented in Fig. 4.2(a) has been proposed to enhance SRAM dependability: two pMOS transistors (P3 and P4) are added between internal nodes in a pair of the conventional 6T bitcells [42−44]. The structure thereby achieves a dependable mode that features margin enhancements by combining two bitcells, especially at low voltage, in both read and write operations. In the dependable read mode, only one wordline is asserted to gain a large β ratio (a ratio of two driver transistors' total size to one access transistor size). A bitcell with no static noise margin (SNM) is recovered by the other bitcell through the two connecting pMOS transistors. In the dependable write mode, a datum is written into a pair of bitcells by asserting both wordlines, which averages and mitigates the write margin degradation. If the 7T/14T SRAM has sufficient operation margins, for instance, then this recovery feature can be disabled when it operates at high voltage by negating the two connecting pMOS

transistors. It is used as a normal mode. The dependable mode and normal mode can be switched according to the operating voltage, power limit, and required dependability in an application. This concept is 'quality of a bit (QoB)', in which the operating voltage, power, and a bit error rate (BER) are controlled as attributes of one-bit information. The 7T/14T memory cells have three modes, as presented in Table 4.1.

- Normal mode: the additional transistors are turned off (CTRL = "L" or /CTRL = "H"), and the 7T cells act as the conventional 6T cell.
- High-speed mode: the additional transistors are turned on (CTRL = "H" or /CTRL = "L"); then the internal nodes are shared by the memory cell pair. Both WL0 and WL1 are driven, which enables a faster readout.
- Dependable mode: the additional transistors are activated, but either WL0 or WL1 is asserted. By doing so, a larger static noise margin is obtainable because a β ratio is doubled.

Table 4.1 Three modes in 7T memory cells.

	# of MCs comprising 1 bit	# of WL drivers	CTRL (/CTRL)
Normal	1 (7T/bit)	1	"L" ("H")
High speed	2 (14T/bit)	2	"H" ("L")
Dependable	2 (14T/bit)	1	"H" ("L")

In the typical mode, a one-bit datum is stored in one memory cell, which is the most area-efficient. In the high-speed mode and dependable mode, a one-bit datum is stored in two memory cells, although the quality of the information differs from that of the typical mode. Consequently, 'higher-speed' or 'more dependable' information is obtainable. We designate this concept as 'quality of a bit (QoB)'. The information quality is scalable in the 7T/14T memory cell [11].

The QoB concept [42−44] and its applications [45, 46] have been studied widely. The 7T/14T SRAM is suitable for low-voltage cache architectures [45]. Furthermore, it realizes block-level instantaneous copying that requires only four cycles to copy all data in a 32-kb memory block [46]. This block-level copying feature is particularly eligible for high-speed data transfer among multi-core processors.

As described in this chapter, by adding a dedicated read port, we propose the 9T/18T dual-port SRAM presented in Fig. 4.2(b). The additional read port is disturb-free. It can therefore operate at a lower voltage than the 7T/14T SRAM can. The proposed SRAM

also has a 9T normal mode and an 18T dependable mode, in which a single-ended inverter and differential sense amplifier are used, respectively, for readout. To incorporate the two modes, an interleaved bitline (BL) scheme is adopted. This proposed 9T/18T SRAM is suitable for use in a multimedia processor and multi-core DSP architecture.

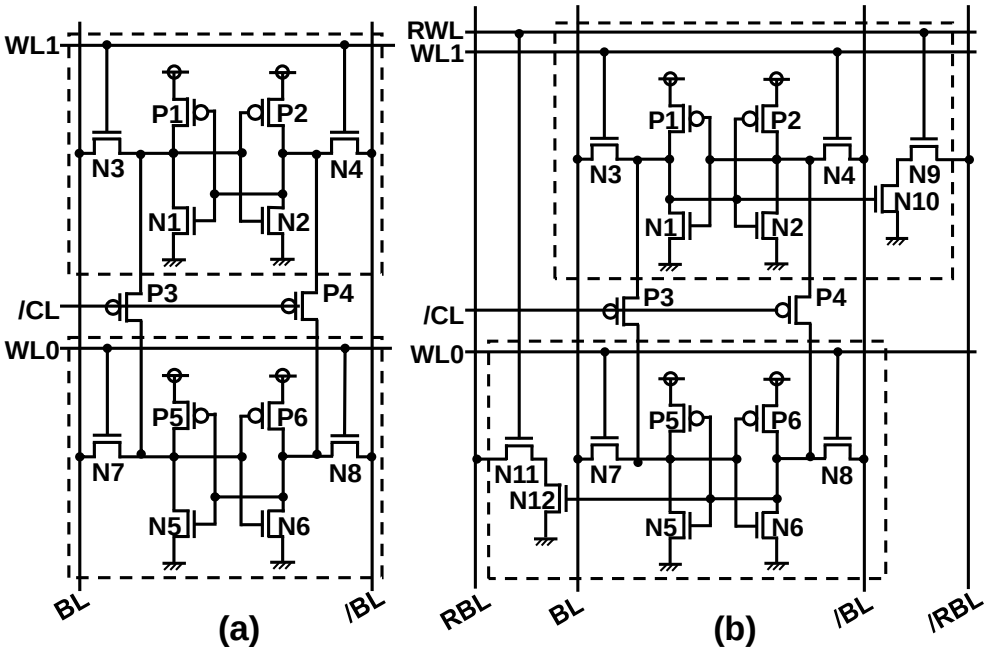

Fig. 4.2 Schematics of (a) a conventional 7T/14T bitcell pair and (b) the proposed 9T/18T bitcell pair.

4.4 9T/18T SRAM

Figure 4.2 again depicts schematics of the 7T/14T and 9T/18T bitcell pairs. The control signal, /CL, is for switching between the normal mode (/CL = "H") and dependable mode (/CL = "L"). In the dependable mode, a 14T or 18T bitcell pair acts as a single bitcell. In the 9T/18T bitcell, four nMOS transistors (N9, N10, N11, and N12) and dedicated read bitlines (RBLs) are appended to the 7T/14T bitcell. Read wordlines (RWLs) are appended for the read ports' control.

To shrink an area of the proposed 9T/18T SRAM and reduce its area overhead, we use an interleaved BL structure for the read port. Figure 4.3 depicts the interleaved bitcell array structure. A pair of right and left bitcells shares an RBL. Instead, two RWLs must be interconnected through each row of the bitcell array.

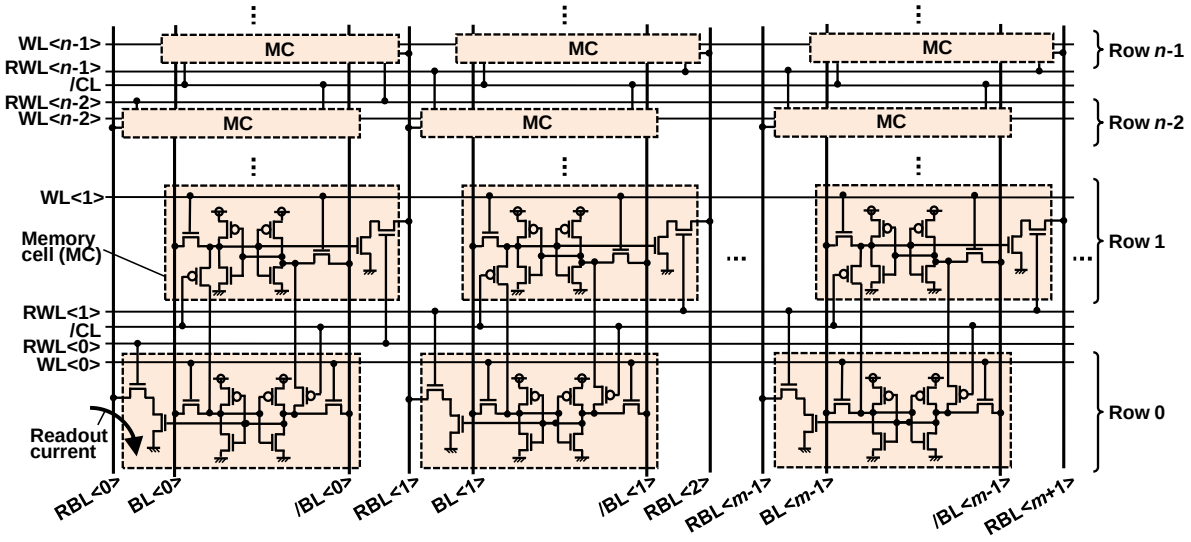

Fig. 4.3 Interleaved BL structure for read port in proposed 9T/18T SRAM. An RBL is shared by an upper left and lower right (or upper right and lower left) bitcells. An RWL is also shared but it is connected to every other bitcell.

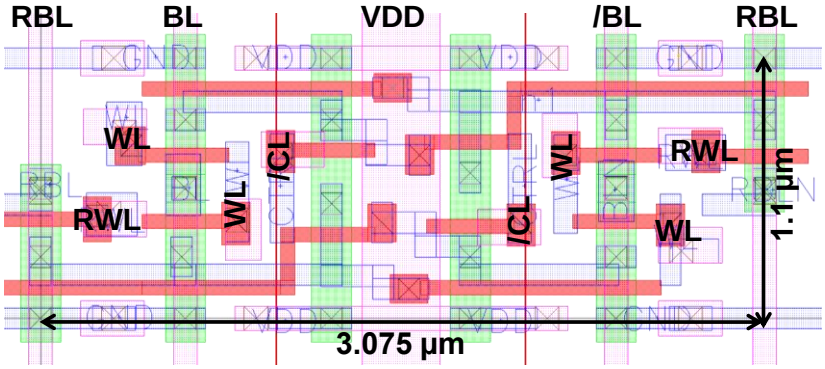

Fig. 4.4 Layout of 9T/18T bitcell pair in a 65-nm process.

Figure 4.4 portrays the layout of the 9T/18T bitcell pair. The design is based on a logic design rule; all transistors have a minimum size (W/L = 170/60 µm). The area of this pair cell is 3.075 × 1.100 µm^2, which has 26.54% and 12.20% area overhead compared to the 7T/14T (= 2.43 × 1.1 µm^2) and 2-b 8T cells (= 2.70 × 1.1 µm^2), respectively.

When RWL<0> in Fig. 4.3 is asserted, the even-numbered column's read ports in Row 0 and Row 1 become active. The stored data are read out to RBLs. When /CL is asserted, each pair of RBLs comes to a differential readout. Consequently, additional multiplexers must be prepared to choose a single-ended or differential readout. Furthermore, the proposed 9T/18T SRAM has the other inside read port because it is a

dual-port SRAM. A datum is read out through the inside BL pairs as well as the outside read port. Therefore, the 9T/18T SRAM achieves higher memory bandwidth than the 7T/14T SRAM does. It is useful for multimedia processors performing tasks such as video processing and multi-core DSP architecture.

Consequently, as portrayed in Fig. 4.5, the proposed 9T/18T SRAM possesses readouts of four kinds. Figures 4.5(a) and 4.5(c) are read operations with the inside BLs in the respective normal and dependable modes, which corresponds to those in the 7T/14T SRAM. Figures 4.5(b) and 4.5(d) show other read operations using the additional read ports.

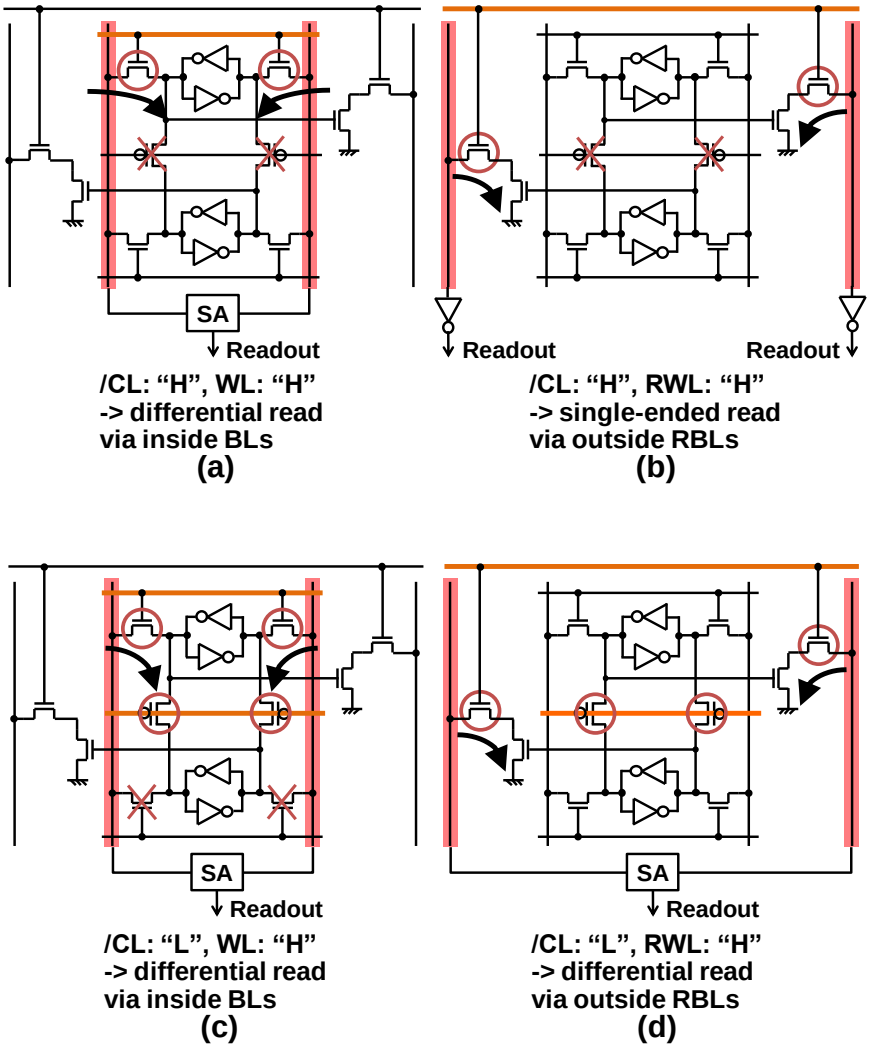

Fig. 4.5 Readouts of four kinds: (a) normal-mode differential read via inside BLs, (b) normal-mode single-ended read via outside read port, (c) dependable-mode differential read via inside BLs, and (d) dependable-mode differential read via outside read port. Here, SA denotes a sense amplifier.

As for write operations, no difference exists between the 7T/14T SRAM and 9T/18T SRAM. In the normal write mode, its condition is the same as that of the classic 6T SRAM. In the dependable write mode, to assure the conductance of the access transistors, both wordlines in the bitcell pair are enabled. Results show that the conductance of the access transistors is averaged, and that the V_{th} variation is suppressed. Thereby, the write margin becomes larger.

4.5 Chip Implementation and Measurement Results

We designed and fabricated a 128-kb SRAM macro using 65-nm process technology for measurement and verification. Figure 4.6 portrays a chip micrograph and SRAM macro layout. The core size of the 9T/18T SRAM macro is 1130 × 413 μm^2. The macro comprises eight blocks (block size, 141 × 413 μm^2), each with a 16-kb array (128 rows × 8 columns × 16 b), address decoders, write drivers, inverters for the outside single-end BLs, and two sets of sense amplifiers for the inside and outside differential BLs. For the sense-amplifier circuit, we adopt a commonly used latch-type sense amplifier.

Figure 4.7 presents the measured bit error rates (BERs) of the 9T/18T SRAM. The minimum operating voltages in the normal mode are 0.67 V and 0.72 V, respectively, with the inside differential BLs and outside single-ended BL; the minimum operating voltage in the single-ended BL is worse by 50 mV than that in the differential BLs because the single-ended BL must be a full swing.

However, in the dependable mode, the outside differential BLs lowers the minimum operating voltage to 0.45 V (90-mV reduction) by virtue of the disturb-free differential readout, whereas that in the inside differential BL needs 0.54 V. This minimum operating voltage reduction can achieve lower power in a low-voltage region.

To apply body biasing and to simulate behaviors at various process corners, the SRAM macro is designed with a triple-well process. In other words, the body biasing control gives global V_{th} variations, meaning that we can estimate reliabilities under the global V_{th} variations. To guarantee the V_{th} control accuracy, we implemented pMOS and nMOS test transistors on the chip for characteristic measurements.

Table 4.2 presents the body bias settings at which four process corners (FF, FS, SF, and SS) are emulated. Furthermore, ΔV_{tn} ($\Delta|V_{tp}|$) represents an nMOS (pMOS) transistor's threshold voltage difference from the fabricated CC transistor.

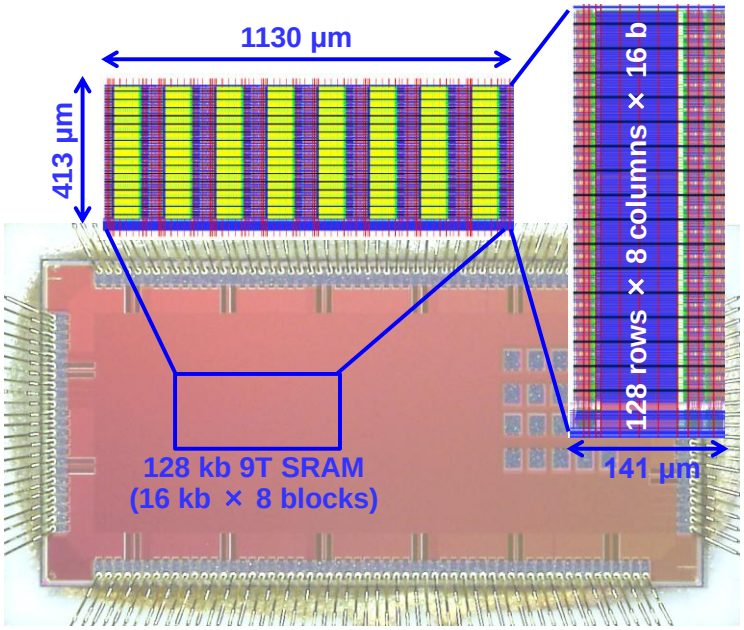

Fig. 4.6 Chip micrograph and SRAM macro layout.

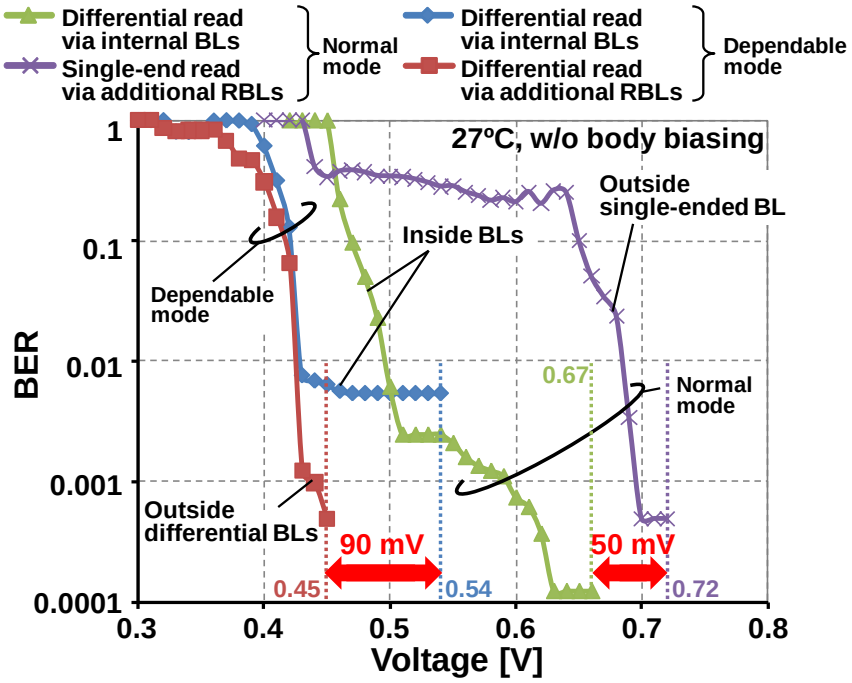

Fig. 4.7 Measured BERs of 9T/18T SRAM for four read operations. Frequency is 1 MHz.

Table 4.2 Body bias settings: ΔV_{tn} and $\Delta|V_{tp}|$

Corner	ΔV_{tn} [mV]	$\Delta\|V_{tp}\|$ [mV]
CC	±0	± 0
FF	−146	−92
FS	−97	+67
SF	+74	−61
SS	+108	+99

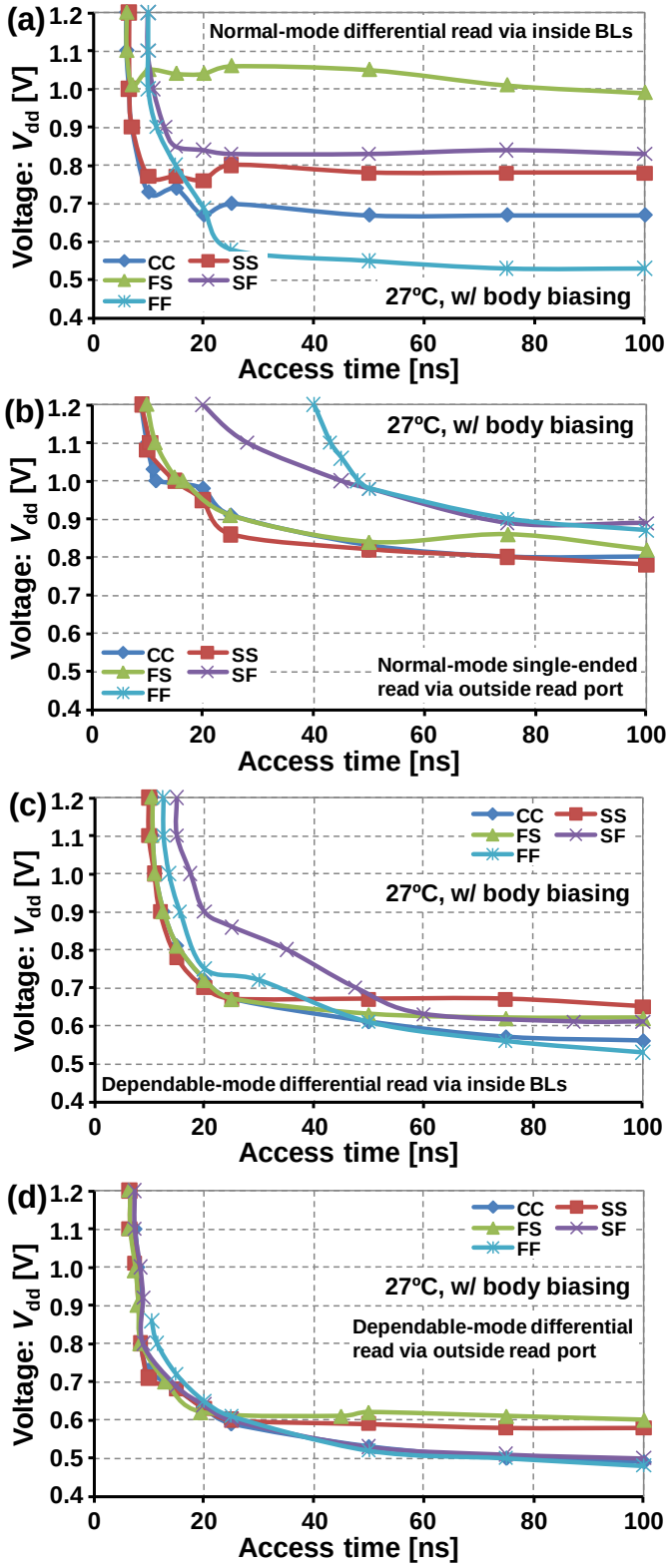

Fig. 4.8 Measured characteristics in access time versus supply voltage: (a) normal-mode differential readout, (b) normal-mode single-ended readout, (c) dependable-mode inside differential readout, and (d) dependable-mode outside differential readout.

Figure 4.8 portrays curves of access time versus supply voltage when body biasing is applied according to Table 1. The access time is the period from the time at which a clock rises up to the time at which an output is fixed: it includes delays in the decoder, wordline, BL charging/discharging, sense amplifier, and terminal I/O buffer. Body biasing is applied not only to the bitcell array but also to peripheral circuits aside from the I/O buffer.

Figure 4.8(a) shows the inside differential readout in the normal mode, which suffers most from the V_{th} variation. The operating voltages are widely varied. Figure 4.8(b) shows the out single-ended readout in the normal mode. The readout inverter needs a full swing, causing slower operation especially at the FF and SF corners because a stronger pMOS discharges a bitline even with an "H" readout. It also engenders fail readout operation. Figures 4.8(c) and 4.8(d) respectively show the inside and outside differential readouts in the dependable mode. The outside BLs exhibit better performance with the disturb-free feature, although the inside differential readout becomes particularly slower at the SF corner.

4.6 Summary

We proposed the 9T/18T SRAM, which provides a better read margin than that of the conventional 7T/14T SRAM. The 9T/18T bitcell topology comprises a conventional 7T/14T cell and an additional read port. In the 9T/18T SRAM, the additional read port can operate without care of the static noise margin because of the disturb-free read operation. Consequently, the 9T/18T SRAM is more stable than the 7T/14T SRAM under threshold-voltage (V_{th}) variation. We fabricated the proposed SRAM using a 65-nm triple-well process. The measurement results show that the dependable read mode using the additional read port can suppress the V_{th} variation, and reduce the operation voltage to 0.45 V, although the dependable read mode using internal bitlines needs 0.54 V. Body biasing control reveals the weakness of the conventional 7T/14T SRAM read process and clarifies that the 9T/18T SRAM has greater reliability against unbalanced process corners, particularly at the SF corner, than the 7T/14T SRAM.

Chapter 5 Memory-Bandwidth Reduction for LVRCS

This chapter proposes a low-memory-bandwidth, high-efficiency VLSI architecture for 60-k word real-time continuous speech recognition. The architecture includes a cache architecture using the locality of speech recognition, beam pruning using a dynamic threshold, two-stage language model searching, a parallel Gaussian Mixture Model (GMM) architecture based on the mixture level and frame level, a parallel Viterbi architecture, and pipeline operation between Viterbi transition and GMM processing. Results show that our architecture achieves 88.24% of the necessary frequency reduction (66.74 MHz) and 84.04% memory bandwidth reduction (549.91 MB/s) for real-time 60-k word continuous speech recognition.

5.1 Introduction

In this chapter, a novel architecture is proposed to reduce the required computational cycle time and memory bandwidth. The architecture comprises a specialized cache, threshold-cut beam pruning, two-stage language model search, parallel processing, and pipeline operation between Viterbi transition and GMM processing. Using that architecture, high-efficiency and low memory-bandwidth Viterbi and GMM processing can be implemented. Thereby, more sophisticated LVRCSR is applicable to VLSI.

This chapter is organized as follows. Section 5.2 introduces the theory and algorithms of speech recognition with the HMM algorithm. Section 5.3 presents the reference hardware design. Section 5.4 presents novel architectural techniques for GMM and Viterbi processors. Section 5.5 specifically describes the GMM and Viterbi VLSI architectures. Section 5.6 presents an assessment of the performance of the architecture. Finally, Section 5.7 summarizes this chapter.

5.2 Speech Recognition Overview

Figure 5.1 presents the speech recognition flow with the Hidden Markov Model (HMM) algorithm. The following items describe concrete stages. Step 1, Feature vector extraction: a feature vector is extracted on a frame-by-frame basis. Step 2, GMM

calculation: a phonemic-model GMM is read and GMM probability, log [b_j (x_t)], is calculated for all active state nodes. Step 3, Viterbi transition: $\delta_t(j)$ is calculated for all active state nodes using GMM probabilities. Step 4, Beam pruning: according to the beam width, active state nodes having a higher score (accumulated probability) are selected; the others are dumped. Step 5, Output sentence: The word-end state having the maximum score is output as a speech recognition result after final-frame calculation and determination of the transition sequence.

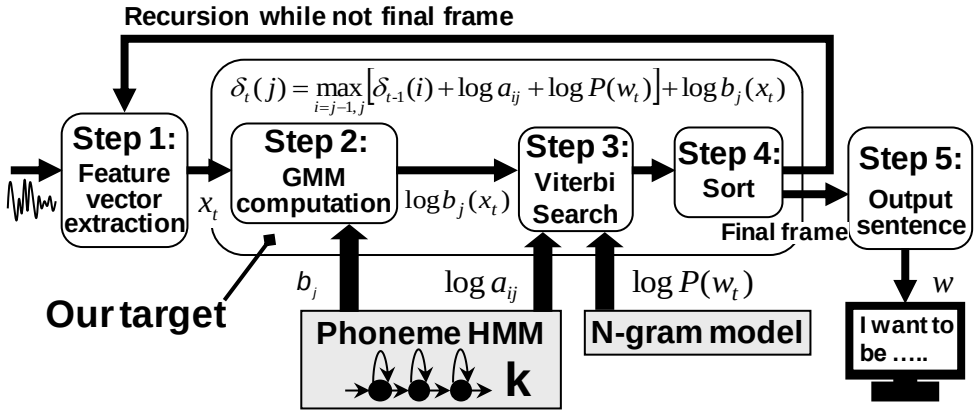

Fig. 5.1 Speech recognition flow with the HMM algorithm.

5.2.1 MFCC Feature Extraction

Mel-frequency cepstral coefficients (MFCCs) are coefficients that collectively make up an MFC. Figure 5.2 shows the MFCC calculation flow. They are derived from a type of cepstral representation of the audio clip (a nonlinear 'spectrum-of-a-spectrum'). The difference between the cepstrum and the mel-frequency cepstrum is that in the MFC, the frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly spaced frequency bands used in the normal cepstrum. This frequency warping can allow better representation of sound, for example, in speech recognition.

5.2.2 GMM Computation

The HMM algorithm is used most frequently, and virtually in all cases of speech recognition [47]. The left–right HMM, which is a kind of the HMM algorithm, is depicted in Fig. 5.3. GMM scoring is performed to calculate the output probabilities of all the possible sounds that could have been pronounced. Output probability is

expressed as $b_j(x_t)$, where x_t is a time-series feature vector extracted from speech. Figure 5.4 presents the concept of the output probabilities. If it is assumed that a feature vector sequence is x_1, x_2, x_3 and that the transition sequence is q_1, q_2, q_3, q_4, then the probability of the transition from q_1 to q_4, $P(q_1 \rightarrow q_4)$, is calculable with (5.1). The solid line in Fig. 5.4 is the transition in this case.

$$P(q_1 \rightarrow q_4) = \pi \times a_{01} b_1(x_1) \times a_{12} b_2(x_2) \times a_{23} b_3(x_3) \times a_{34} \tag{5.1}$$

In the HMM algorithm, each HMM corresponds to a phone. Each word is expressed as a sequence of phones. Each sentence is represented as a sequence of words.

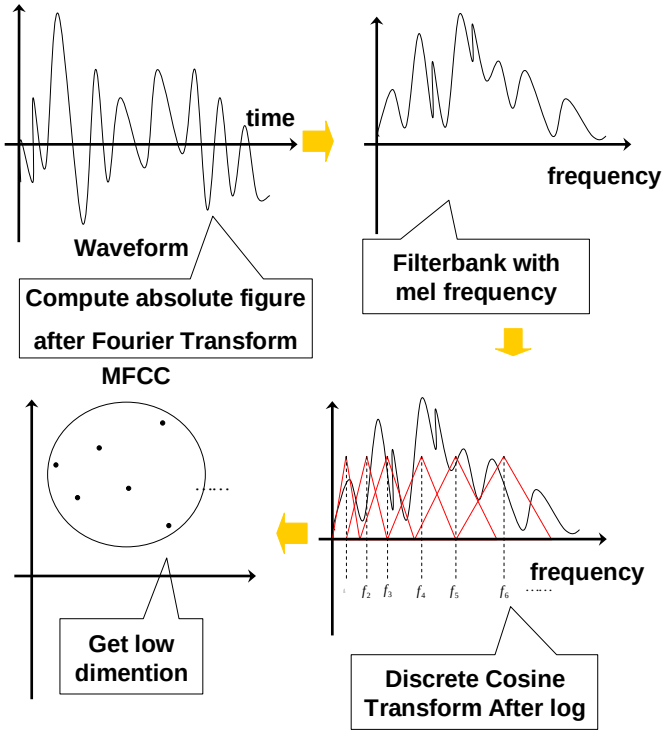

Fig. 5.2 Calculation flow to obtain MFCC from the waveform data.

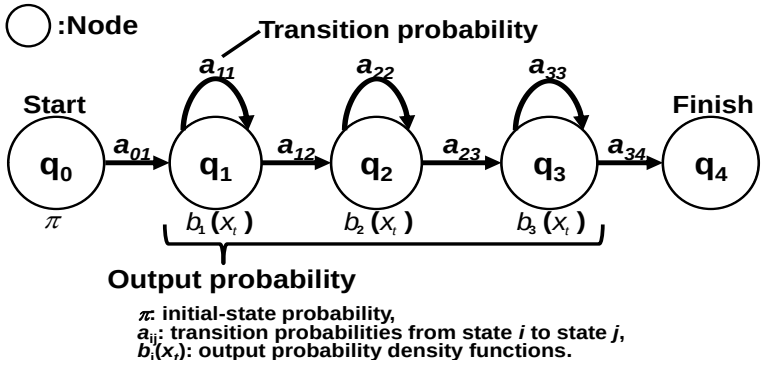

Fig. 5.3 Left–right HMM.

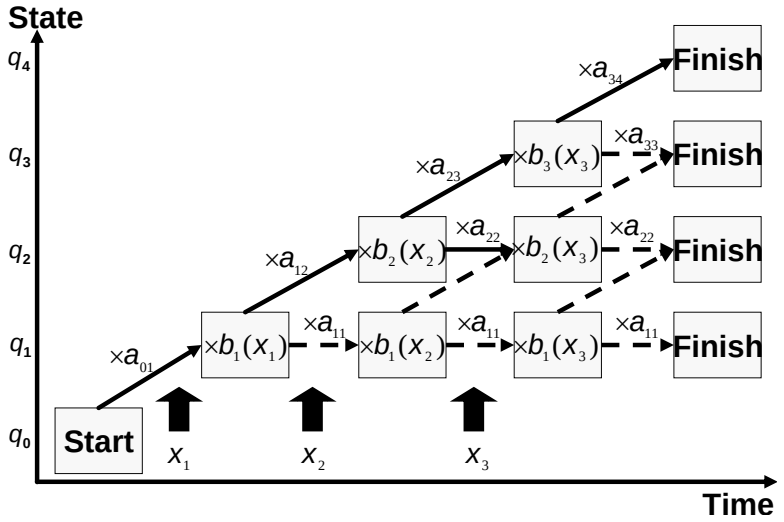

Fig. 5.4 Left–right HMM calculation flow.

The GMM computation obtains $\log [b_j(x_t)]$ from a feature vector x_t and parameters of a GMM, which is used in the Viterbi search algorithm. As expressed in (5.2), $\log [b_j(x_t)]$ is expressed as a logarithm of a sum of the Gaussian distribution multiplied by weight functions. We assume that Σ_i is a diagonal matrix and we simplify it.

$$\log b_j(x_t) = \sum_i^{mix} \lambda_i N(x_t, \mu_i, \Sigma_i) \tag{5.2}$$

$$= \log\left[\sum_i^{mix} \lambda_i \left[\frac{1}{(2\pi)^{\frac{P}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x_t - \mu_i)^t \Sigma_i (x_t - \mu_i)\right\}\right]\right]$$

$$= add \log\left[w_{ij} + \sum_{s=1}^{P} (x_{ts} - \mu_{ijs})^2 \sigma_{ijs}\right] \tag{5.3}$$

$$w_j = \log \lambda_i + \log\left[\left((2\pi)^{\frac{P}{2}} \left(\prod_{s=1}^{P} \Sigma_{ijs}\right)^{\frac{1}{2}}\right)^{-1}\right], add \log[X_i] = \log \sum_{i=1}^{mix} X_i$$

In those equations, the following parameters are used: $b_j(x_t)$ signifies a GMM probability density function (PDF), N denotes a Gaussian distribution PDF, P represents the number of dimensions in a feature vector, *mix* stands for the number of mixtures in the GMM, x_t is a feature vector, μ is a mean parameter, Σ represents a variance–covariance matrix, and λ denotes a weight function. In addition, w_{ij} is a constant number that can be computed offline before speech recognition. Actually, (5.3)

shows that the GMM computation at one dimension consists of one addition, one subtraction, two multiplications, *P* summations, and their respective logarithms.

5.2.3 Time-Synchronous Viterbi Beam Search

The following formulas show the log-Viterbi algorithm. To prevent underflow, logarithms are usually taken.

Initialization:

$$\delta_0(0) = \log \pi \tag{5.4}$$

Recursion:

$$\delta_t(j) = \max_{i=j-1,\ j} [\delta_{t-1}(i) + \log a_{ij}] + \log b_j(x_t) \tag{5.5}$$

for $1 \le t \le T, 1 \le j \le N_{state}$

Termination:

$$P(w \mid x_1, x_2, \cdots, x_T) = \max_{N_f} [\delta_T(i)] \tag{5.6}$$

In those equations, T represents the number of frames, N_{state} denotes the number of all HMM states, N_f stands for the states set that correspond to word-end, and i and j are state indexes. In addition, $\delta_t(j)$ is a likelihood value at a time index t and state j; w denotes a recognition output sentence.

The speech waveform is divided into frames (15–25 ms). Then a feature vector is calculated for each frame. In fact, (5.5) shows that, once a feature vector is obtained, each state in the HMM move to the next state that maximizes the likelihood value. For that reason, the transition sequence is uniquely determined (depicted in Fig. 5.5).

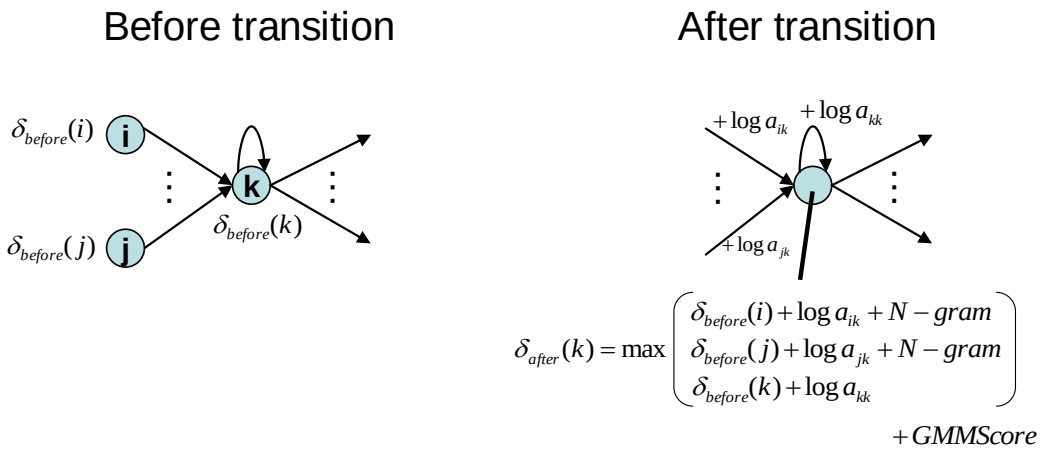

Fig. 5.5 Viterbi computation.

In actual speech recognition, N_{state} represents 1,000–5,000 states. It is too large to calculate all likelihood values. To address this problem, after all transitions in one frame are over, only a few ('beam width') nodes with large likelihood values are considered. The remaining nodes are terminated. We designate this process as *beam pruning*. Nodes that are unpruned in later stages are designated as active state nodes. In the next frame, only the likelihood values in the active state nodes are computed. After the final frame of computation, the maximum likelihood value in the word-end state is output as the recognition result.

5.2.4 N-gram Language Model Search

To achieve high-accuracy recognition, a language model is adopted. A language model represents grammatical accuracy of sentences. Generally, an *N*-gram model is used. In the *N*-gram model, a probability of a sentence, $w = \{w_{0,}, w_1, w_2, \ldots, w_{n-1}, w_n\}$, which is uttered, is expressed as $P(w)$, as shown in (5.7).

$$P(w) = P(w_0)P(w_1 \mid w_0)P(w_2 \mid w_0 w_1), \cdots, P(w_n \mid w_0 w_1, \cdots, w_{n\text{-}1}) \tag{5.7}$$

In practice, the number of *N* is cut out as two or three. The following equations show the log-Viterbi algorithm in the *N*-gram model.

$$P(w) = P(w_0)P(w_1 \mid w_0)\prod_{k=3}^{n} P(w_k \mid w_{k-2} w_{k-1}) \tag{5.8}$$

Here, to obtain $P(w)$, it is necessary to calculate $P(w_i), P(w_i \mid w_{i-1}), P(w_i \mid w_{i-2}, w_{i-1})$ in advance. In the trained dictionary, these values are obtainable in a polynomial time. These values are represented as the following equations, using the number of times of appearance in the trained space of a sentence, $w = \{w_{0,}, w_1, w_2, \ldots, w_{n-1}, w_n\}$, as $C(w)$.

$$P(w_i) = \frac{C(w_i)}{\sum_{w_i} C(w_i)} \tag{5.9}$$

$$P(w_i \mid w_{i-1}) = \frac{C(w_{i-1}, w_i)}{C(w_{i-1})} \tag{5.10}$$

$$P(w_i \mid w_{i-2}, w_{i-1}) = \frac{C(w_{i-2}, w_{i-1}, w_i)}{C(w_{i-1}, w_{i-1})} \tag{5.11}$$

5.3 Referential Hardware Design

We implemented referential hardware of a speech recognition system based on Julius 4.0 [7], which is well-known Japanese speech recognition system software using Verilog HDL [48, 49]. This architecture comprises a GMM processor and a Viterbi processor. The GMM processor adopts a four-core parallelism, pipelined processing, and a vector look-ahead scheme [48]. In the vector look-ahead scheme, several feature vectors are buffered in advance. Their output probabilities are computed in parallel. Then, answers are stored in a cache. The answer stored in the cache is read out if a duplicated state appears in a subsequent frame. The Viterbi processor adopts a dual-core parallelism and asynchronous scheduling to reduce the required minimum operating frequency and the idle cycles [49].

5.3.1 Computation Amounts and Memory Bandwidth

We first profiled the referential hardware (described in Subsection 5.3) of a speech recognition system, using 5-k, 20-k, and 60-k word speech recognition models. The beam widths were, respectively, set to 500, 2,000, and 4,000 with 5-k, 20-k, and 60-k word models. We used an FPGA (Stratix *II*; Altera Corp.) to obtain the necessary frequencies and the memory bandwidths needed for the three models.

Figures 5.6 and 5.7 respectively portray the required frequency and the required memory bandwidth in the referential hardware to achieve real-time speech recognition. The GMM processing dominates a large share of the total computation time, of which the computing output probabilities occupy 81.9% considering a 20-k word LVRCSR. For LVRCSR recognition, minimizing the computation time of the output probabilities is effective in terms of the computational workload. In contrast, Viterbi search consumes a larger share of the total memory bandwidth as the number of words increases. When developing a VLSI chip for LVRCSR, a salient issue is the high memory bandwidth of speech recognition processing, which engenders inefficient power consumption. It is necessary to reduce the memory bandwidth to develop a low-power VLSI chip for use with LVRCSR. Figure 5.7 shows that, when considering a 20-k word LVRCSR, memory bandwidths of the Viterbi, GMM, and sort processing are estimated respectively as 534.32 MB/s, 485.71 MB/s, and 138.65 MB/s. Furthermore, when considering a 60-k word LVRCSR, the memory bandwidth of Viterbi search

increases by 483% (2580.99 MB/s) compared to a 20-k word LVRCSR. As described in this chapter, to realize a low-power and low-memory-bandwidth 60-k word recognition system, we propose several ideas to reduce the workload and memory bandwidth.

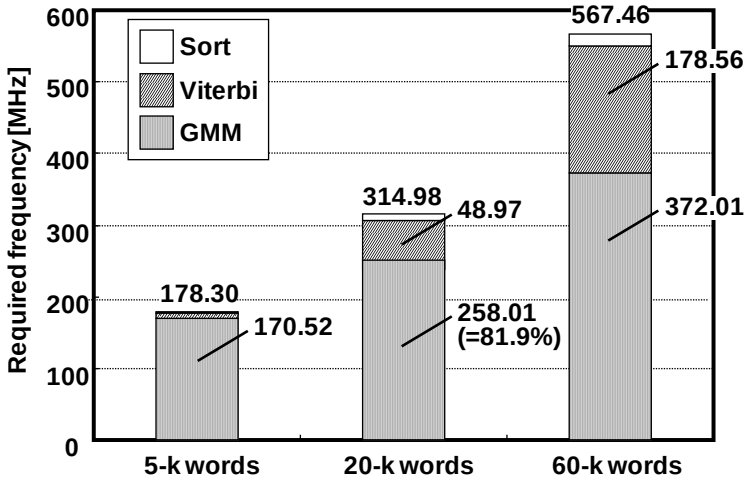

Fig. 5.6 Required frequency in a real-time process with the referential hardware [48, 49].

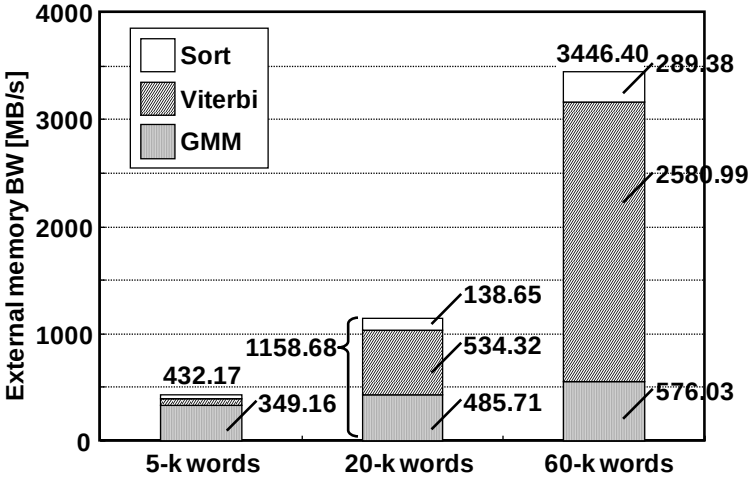

Fig. 5.7 Required memory bandwidth in a real-time process with the referential hardware [48, 49].

5.4 Proposed Schemes

5.4.1 Burst GMM Calculation

The memory bandwidth in GMM calculations results from large GMM parameters. Each state in the HMM has a specific GMM. To compute the likelihood log $[b_j(x_t)]$, the memory controller must read the Gaussian parameters, the mean μ_{ijs}, and the standard deviation σ_{ijs}, from external memory. However, each phonemic HMM has a

self-transition. Fortunately the GMM data used in the present frame will be reused in the next frame at high probability. This probability reaches more than 90%. To reduce the external memory bandwidth for reading the acoustic model, we share the Gaussian parameters to compute GMM probability among several contiguous frames at a time [48]. If we were to share the Gaussian parameters for 50 frames, then we would need to increase the GMM result RAM used for storing the calculated scores, and the input buffer, which is stored the MFCC feature vectors, from 15 kB to 750 kB and from 200 B to 10 kB. However, the external memory bandwidth can be reduced to 1/50 if all Gaussian parameters can be shared. For 20-k and 60-k word recognition, it is necessary to maintain sufficient beam width according to the number of words to achieve highly accurate recognition. Furthermore, it engenders a large amount of GMM processing among approximately all GMM states. In our novel GMM architecture, every GMM probability is computed for all input feature vectors. In doing so, a two-stage pipeline between GMM and Viterbi is readily applicable.

5.4.2 Modified Unigram Language Model

A unigram language model was used for computing word-internal transitions. The unigram language model comprises HMM probabilities; each value corresponds individually to a state of HMM trees (depicted in Fig. 5.8). In the conventional scheme, if the HMM state transitioned, then the probability of the previous state subtracted from the temporal score before being added the new probability of the current state to the temporal score. In terms of a unigram language model, the previous state of every state is individually identifiable. For that reason, we modified the unigram language model to hold only difference values between the probability of a new HMM state and the probability of its previous HMM state. Using our modified unigram language model, the extra memory access to the previous state can be reduced. Furthermore, because the unigram update process can be eliminated, word-internal transitions, cross-word transitions to the isolated trees, and cross-word transitions to shared trees can be treated simply using the same process module. Furthermore, the internal memory usage for storing unigram transitions can be saved.

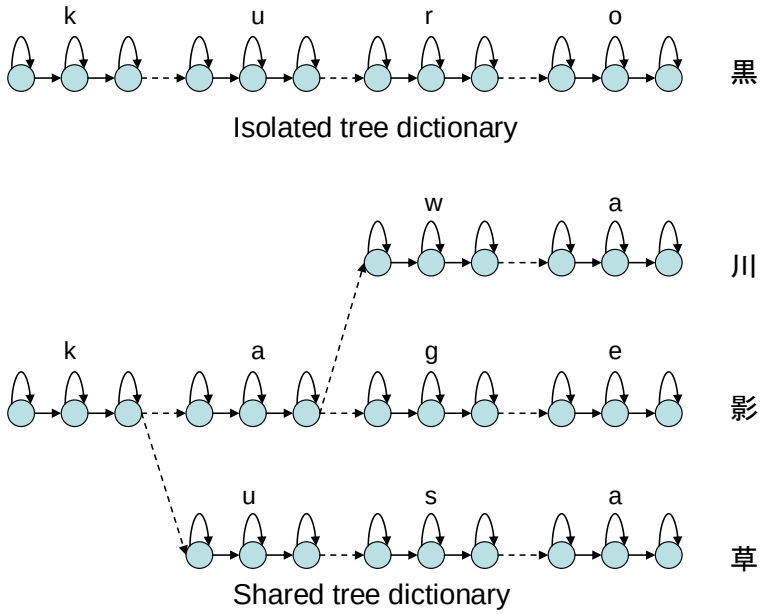

Fig. 5.8 Tree dictionary for Viterbi search.

5.4.3 Threshold-Cutting Scheme

In the conventional architecture, the sort is processed after a Viterbi search at every frame before pruning the lower score transitions. This necessitates a large workspace because all temporal scores that are generated by the Viterbi transition must be retained until the Viterbi search of the current frame is finished at every frame, although almost all scores are pruned by the beam-cutting process at every frame. Moreover, sort processing requires computational amounts greater than 10 MIPS and consumes memory bandwidth of more than 400 MB/s for 60-k word recognition (shown in Figs. 5.6 and 5.7).

We introduce the threshold-cutting scheme to reduce the workspace and memory bandwidth instead of sort processing. In this scheme, the threshold is set adaptively to a constant value at every frame. All transitions that have a lower score than the threshold are pruned while processing Viterbi search of current frame. Only selected transitions with a higher score than the threshold are stored in workspace memory. Therefore, the proposed threshold-cutting scheme cut off the superfluous workspace.

Why is the threshold changed adaptively? This prevents degradation of the beam cutting accuracy. An improper threshold gives rise to inconvenient cases in which too many nodes remain or too many nodes are cut off in comparison to the beam width. An adaptive threshold is set based on the difference between the average scores of the

previous frame and the current frame and the number of selected transitions between the previous frame and the current frame.

Figure 5.9 shows beam width variation with the threshold-cut scheme when the target width is set to 1,500. The threshold cut results have ± 500 variations. Regarding the actual speech recognition results obtained with a 20-k word language model, the speech recognition accuracy is unaffected by this variation of beam width because almost all transitions that engender the final speech recognition output trellis output higher scores than the others.

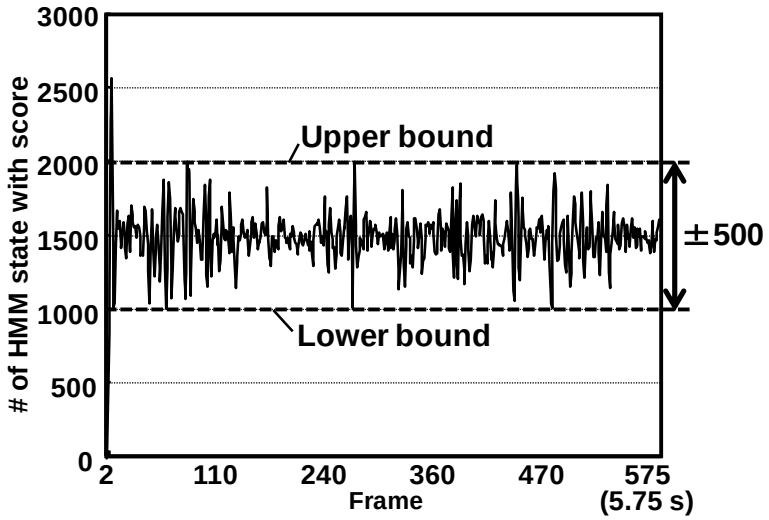

Fig. 5.9 Beam width variation with the threshold-cut scheme.

5.4.4 Two-stage Language Model Search

The Viterbi transition comprises word-internal transitions, cross-word transitions to isolated trees, and cross-word transitions to shared trees. Figure 5.10 presents a comparison of these transition appearance ratios derived from profiling with Julius 4.0 and Visual Studio C++ (Microsoft Corp.). In the figure, the cross-word transitions to isolated trees are dominant. Consequently, to reduce the computational amount for the cross-word transitions to isolated trees, we propose a novel two-stage language model search scheme. This scheme is derived from the transition frequency difference between phonemic HMM and language HMM: in actual human speech, the appearance ratio of syllabic transitions is much lower than the MFCC frame rate: 100 Hz. In this scheme, the cross-word transition search is divided into two stages. The first stage is a simplified language model search for the top 10 important transitions of bigram probability. The

second stage is a detailed language model search for all cross-word transitions. In the traditional language model search, only our second search treated every frame. However, in our proposed language model search, the second stage is treated at every five frames. By applying this proposed search, when the frequency of detail language search is set to 1/5, the computational amount and memory bandwidth can be reduced to 1/5, corresponding to a 78% reduction in the total Viterbi processing. The accuracy degradation that occurs when using this scheme is less than 1%, as shown by actual speech recognition measurements with the 20-k word language model.

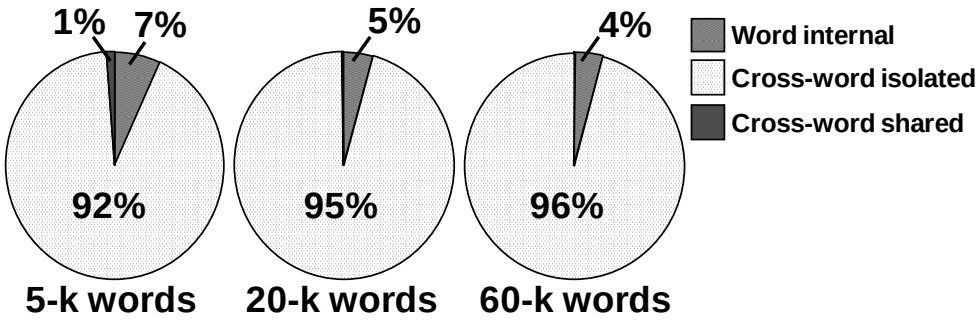

Fig. 5.10 Appearance ratios of three transition types in Viterbi search in Julius 4.0.

5.5 VLSI Architecture

The GMM memory bandwidth was reduced as explained in Subsection 5.4.1. Nevertheless, this reduction of GMM processing is not efficient when considering the total LVRCSR because the GMM memory bandwidth does not increase depending on the vocabulary library scale. However, as explained in this subsection, we specifically examine the Viterbi beam search algorithm and reduce its memory-bandwidth because the memory bandwidth of the Viterbi processor increases with the vocabulary number.

Table 5.1 Memory bandwidth in 20-k word Viterbi search

Item	Memory bandwidth [MB/s]
Tree dictionary	39.41
Transported token list	144.21
Transport information	3.09
Trellis	0.48
Bigram	346.60 (29.26)
Bigram address table	0.12
Unigram	0.35
Total	534.32(216.98)

Table 5.1 shows the memory bandwidth of each component in Viterbi processing when considering 20-k word recognition, as obtained from the hardware simulation using the referential hardware. We applied the proposed two-stage language model search to this hardware simulation. The dominant memory accesses are to the transported token lists (144.21 MB/s) that are address tables of the external memory for active state nodes, and to the bigram database (346.60 MB/s). Therefore, we introduce custom caches to these memory accesses and reduce memory bandwidth. Here, the memory bandwidth of the bigram was reduced to 29.26 MB/s, as described in the previous subsection.

Figure 5.11 portrays the Viterbi cache architecture concept. Two caches are introduced to the bigram and the transported token list because their memory bandwidths are wider than those for other data in Viterbi processing (see Table 5.1). The stored data are bigram probabilities for bigram cache and the temporal calculated token list for the beam cache.

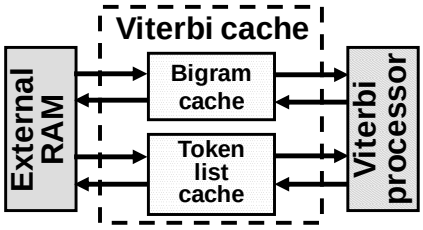

Fig. 5.11 Cache architecture concept in Viterbi search processing.

Figure 5.12 portrays a block diagram of a 60-k word speech recognition processor using our proposed schemes. To reduce the operation cycle time and external memory bandwidth, our architecture contains some schemes such as (1) cache architecture using the locality of speech recognition, (2) beam-pruning using a dynamic threshold, (3) two-stage language model search, (4) parallel GMM architecture based on mixture level and frame level, (5) parallel Viterbi architecture, and (6) pipelining operation between Viterbi transition and GMM processing. We used an FPGA (Stratix *II*; Altera Corp.) to verify the architecture at the RTL level.

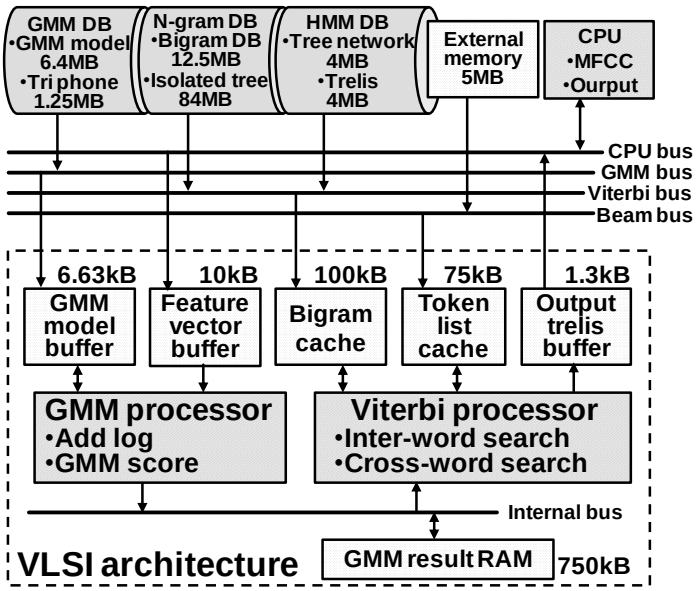

Fig. 5.12 Block diagram of proposed processor architecture for a 60-k word speech recognition system.

In Fig. 5.12, the memory sizes of DBs of three types are derived from the 2,000-state Gaussian four mixture tri-phone model and the 60-k word Japanese language model. The burst GMM calculation shares 50 frames, the two-stage language model search has 1/5 frequency of detail language search, and the Viterbi beam width is set to 4,000. These parameters engender the buffer sizes (6.63 kB, 10 kB, and 1.3 kB), cache sizes (100 kB and 75 kB), and GMM result RAM size (750 kB), as the figure shows. All external database sizes and the external memory size are also determined by the number of language-model vocabulary (60 k) and the Viterbi beam width (4,000).

5.5.1 Implementation of GMM Computation

To achieve speech recognition in real time, but at a lower operating frequency, we propose a parallel architecture with low memory bandwidth. Our proposed scheme features the following three points:

- parallel computing of Gaussian distributions as to the number of GMM mixtures and frame-based parallel processing,
- parallelization in taking logarithms based on a look-up table,
  and
- pipeline architecture for reading Gaussian distribution parameters and calculating them.

In the first feature, the parallelism can be increased theoretically to the number of mixtures in the GMM. However, it increases memory bandwidth linearly. For this reason, in our architecture, the parallelism in computing the Gaussian distributions is expanded to a frame base, as described in Subsection 5.4.1.

As the second feature, we prepare two-input add-log units, which calculate an approximate logarithmic value of a sum of two inputs. For instance, to carry out four 'add log's, four data are divided into two groups: each group is input to two two-input add-log units, and each is calculated individually but simultaneously. The two two-input add-log units output two results. Repeating this operation, we can obtain a desired output for any number of data. This method reduces the computation cycles. The number of parallelism in taking logarithms is 16.

The third one shows that memory reading and Gaussian distribution calculations are performed simultaneously in our dedicated hardware.

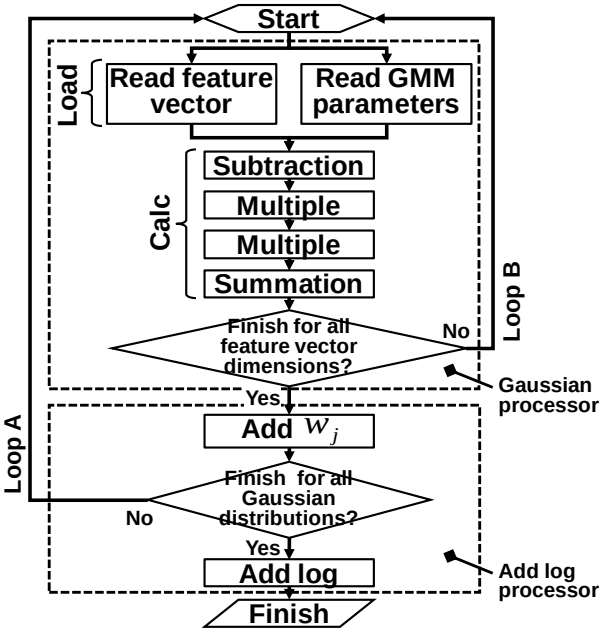

Fig. 5.13 GMM computation flow.

Figure 5.13 shows the operation flow of the GMM calculation. The GMM calculation is divisible into two steps. The first step comprises the data load and calculation processes. In our architecture, this step is treated in the Gaussian processor module. The second step is add-log computation; this step is calculated in the Add log processor module in our architecture.

Figure 5.14 portrays the proposed GMM architecture. Input data are feature vectors among 50 frames and GMM model parameters comprising 2,000 states. Output shows the output probabilities of 50 frames that correspond to every state.

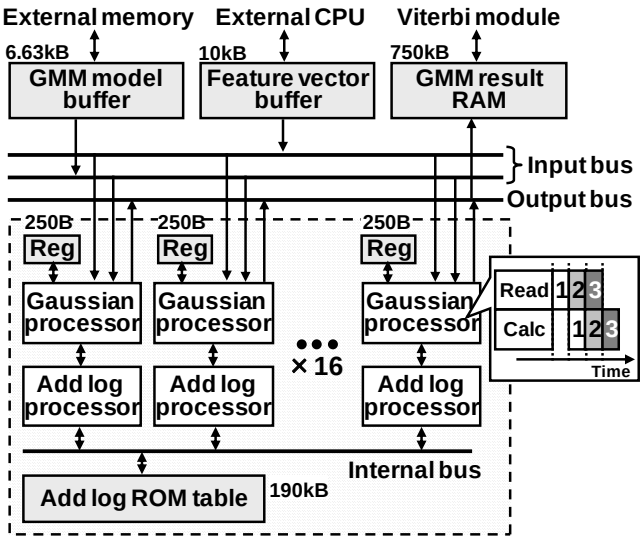

Fig. 5.14 GMM processor data path.

5.5.2 Viterbi and N-gram Architecture

Figure 5.15 presents the Viterbi transition flow. The Viterbi transition in a frame is divided roughly into three steps: word-internal transition, trellis-saving, and cross-word transition.

The Viterbi transition is performed for all active state nodes left in the previous frame. First, fetch an active state node from an active nodes queue. 1) *Word-internal* transition: perform word-internal transition when the transition source node and destination node belong in the same word. 2) *Trellis saves*: save a trellis when the active state node is the end of a word. The trellis is a dataset with a word history and a score of the word-end node. It is used to determine the recognition result in the last frame. 3) *Cross-word transition*: perform cross-word transition after a trellis save. In this step, the transition is performed from an active state node, which is a word-end state node, to all word-beginning nodes.

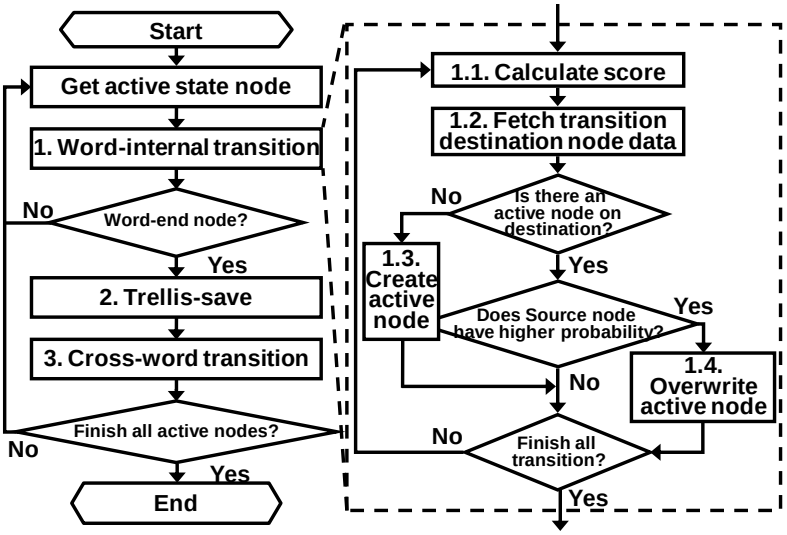

Fig. 5.15 Viterbi computation flow.

The word-internal transition and cross-word transition can be expressed with the same flow. 1.1) *Calculate score*: The transition probability from the HMM dictionary is added to the score of active state node. 1.2) *Fetch transition destination node date*: fetch the information of a transition destination node from a HMM dictionary and Result RAM. 1.3) *Create active state node*: create an active state node when no active state node exists on the transition destination. 1.4) *Overwrite active state node*: overwrite the active state node on destination when a lower probable active state node exists on the transition destination.

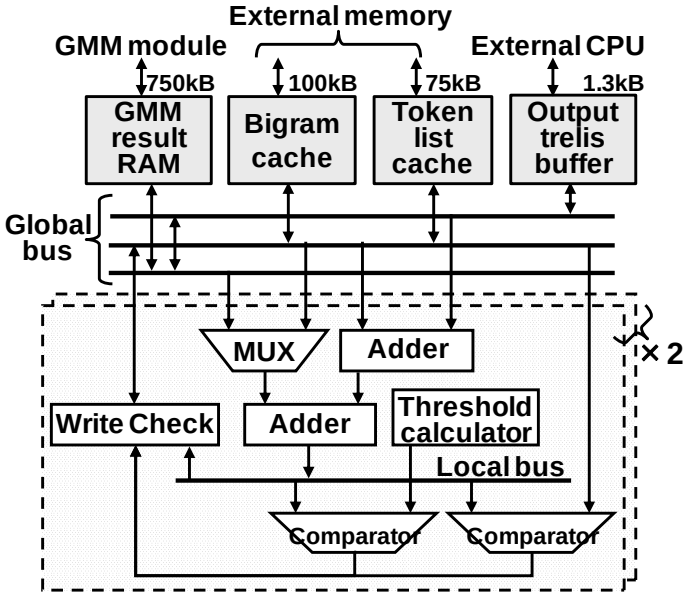

Fig. 5.16 Proposed Viterbi architecture.

Figure 5.16 portrays the proposed architecture for reducing the Viterbi processor memory bandwidth. The proposed architecture employs a specialized cache, threshold cut, and two-stage language model search. The Viterbi cache is set between the external memory and Viterbi processor.

The following subsections show the specialized cache architecture in detail. The number of words in the vocabulary dictionary is set to 20,000. The beam width is set to 2,000. The start nodes are 1,000. To limit the memory capacity of buffered RAM for VLSI, a Viterbi processor is necessary for efficient processing by implementing a specialized cache.

5.5.3 Bigram Cache

Locality of data is important to implement a cache. However, no chance of reading the same bigram probability exists in a frame in the HMM algorithm. Therefore, data correlation between frames is important to implement a bigram cache. Figure 5.17 shows the correlation that we researched using software profiling with Visual C++ (Microsoft Corp.) and Julius 4.0. The figure indicates that a correlation exists between frames, which accounts for about 60–90%. From this result, we inferred that the target value of the hit rate is about 70%.

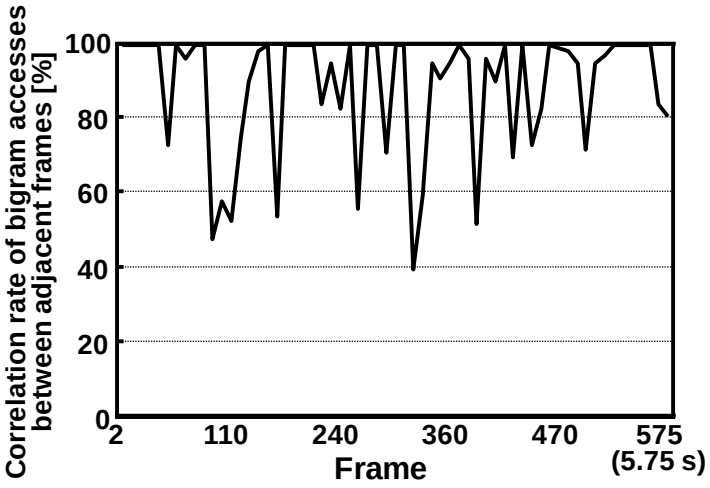

Fig. 5.17 Correlation rate of bigram accesses between adjacent frames profiled using Julius 4.0.

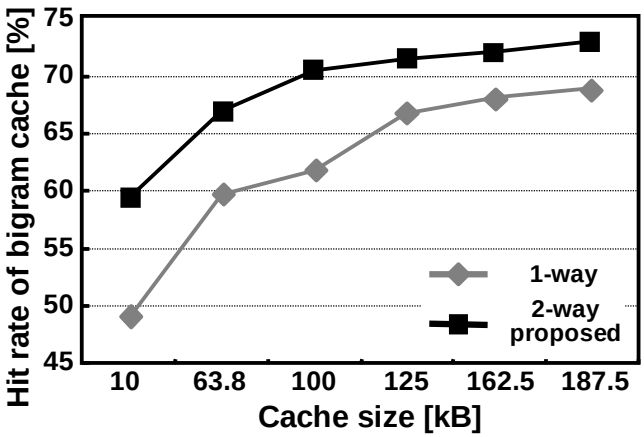

Fig. 5.18 Bigram cache hit rate profiled using Julius 4.0.

The top 10 data are read in using a two-stage language model search process. We then determine the cache line size as 40 B (= 10 × 4 B) to fit it. Figure 5.18 portrays the relation between the hit rate and cache size with a direct mapping scheme by hardware-emulated profiling using Visual C++ based on a custom array corresponding to cache architecture. Additionally, to achieve a higher hit rate, we employ another scheme using a feature of speech recognition flow. In the bigram step, a node with a higher score tends to survive in the next frame. To set the timing of writing a score to the cache as the score is overwritten, the higher score is stored in the cache late in the same frame because the existing score is overwritten when a higher score appears. Using such features of speech recognition, we propose a cache as presented below.

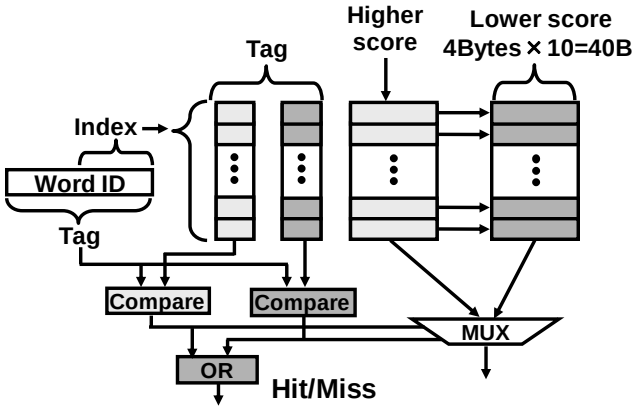

Fig. 5.19 Proposed two-way cache.

The proposed cache scheme is presented in Fig. 5.19. The cache adopts a two-way set associative scheme. In the two-way set associative scheme in that figure, a lower bit of

word ID is used to create an index. The whole word ID is used as a tag. The two-way consists of two parts: one for a higher score and one for a lower score. When a mis-hit occurs, a new bigram probability is stored in the high score part because the bigram probability of late in the frame is higher. On this occasion, the score in the higher part is settled to the lower part; a new bigram score is stored in the higher part. In doing so, a bigram probability computed late in the frame tends to remain in the cache. Therefore, the hit rate becomes high. Figure 5.18 shows the hit rate. Using the proposed architecture in this chapter, we decide that the cache size is 100 kB to achieve the target hit rate of 70%. By implementing the bigram cache and two-stage language model search, we can reduce the memory bandwidth of bigram probability by about 94%.

5.5.4 Token List Cache

We adopt a direct mapping cache for the token list cache. The cache line size is set to 4 bits, which is the same as the size of a transitioned token list data. In our architecture, to guarantee high-accuracy recognition, the beam width is targeted to 2,000. In fact, 60% of node information is occupied by start-node data, and the bigram calculates frequently accessed start-node data. Therefore, the start-node data always hold in the token list cache. The hit rate of the token list cache is presented in Fig. 5.20. Using the architecture proposed in this chapter, we decide that the cache size is 75 kB to achieve the target hit rate of 70%.

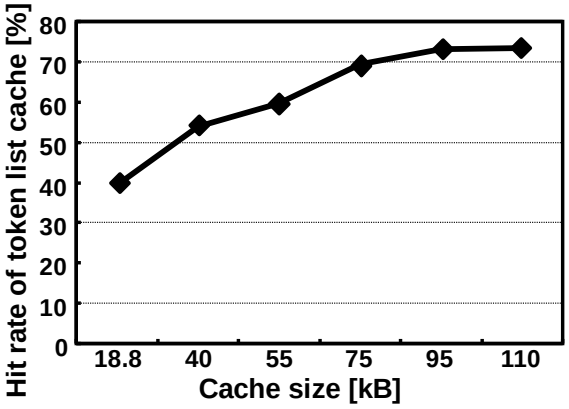

Fig. 5.20 Token list cache hit rate.

5.6 Implementation Results

As described in Section 5.3, we first implemented this architecture on an FPGA

(Stratix II; Altera Corp.) to verify the VLSI architecture on a register transfer level (RTL) basis. However, our VLSI architecture described in Section 5.5 must use a larger size of internal memory than the FPGA has. Therefore, to exploit the cache operation and its effect, we simulate SRAM models and corresponding logic operation in speech recognition using a Verilog simulator, and obtain a required frequency and memory bandwidth for real-time operation.

5.6.1 Required Frequency and Memory Bandwidth

Figure 5.21 shows the required operating frequencies of the proposed architecture and referential architecture (depicted in Fig. 5.6) for each number of vocabularies. The proposed architecture incorporates the proposed schemes described in Section 5.3; the referential hardware does not. Especially, the two-stage language model search in the proposed architecture contributes to reduction of the required frequency by almost one-fifth. In Fig. 5.21, our proposed architecture can operate real-time speech recognition at 21.71 MHz for 5-k words, at 41.71 MHz for 20-k words, and at 66.74 MHz for 60-k words. Our proposed architecture can reduce the required frequency for real-time speech recognition by 87.82% for 5-k words, by 86.76% for 20-k words, and by 88.24% for 60-k words.

Figure 5.22 shows the required memory bandwidth of our proposed architecture and referential architecture and referential architecture (depicted in Fig. 5.7). Our proposed architecture achieves 82.96% reduction (73.64 MB/s) for 5-k words, 81.12% reduction (218.63 MB/s) for 20-k words, and 84.04% reduction (549.91 MB/s) for 60-k words.

5.6.2 Comparison with other Architectures

Table 5.2 presents a comparison of specifications with other hardware-based systems. This table presents the vocabulary size, GMM model, Viterbi beam width, accuracy, real-time factor frequency, memory bandwidth, logic size, memory size, and platform. The vocabulary size represents the number of words in the language model dictionary. The accuracy is the recognition rate. The real-time factor represents how fast the hardware is: for instance, a real-time factor of ‘0.5’ corresponds to twice as fast as real-time operation. The frequency represents the operating frequency of the hardware speech-recognition system. The external memory bandwidth equals the data transfer

traffic between an FPGA/VLSI chip and external SDRAM/DRAM memory.

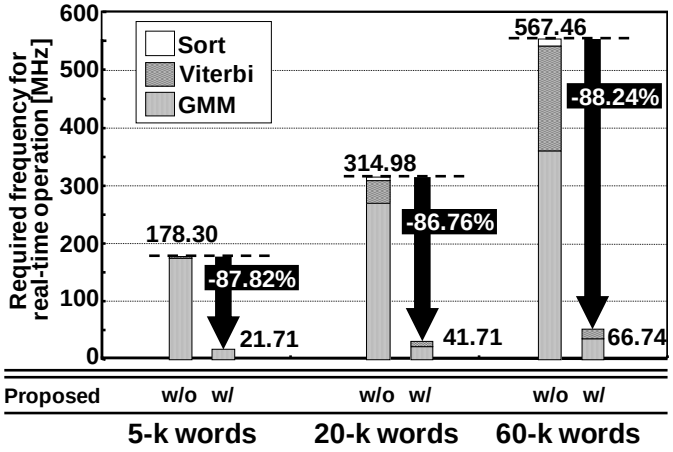

Fig. 5.21 Required frequency comparison with conventional architecture.

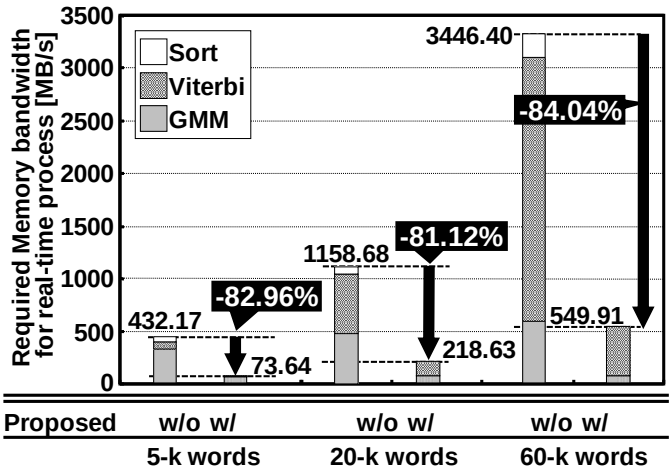

Fig. 5.22 Required memory bandwidth using the real-time process.

The comparison reveals that our architecture achieves the lowest external memory bandwidth with the same-size vocabulary. Our architecture reduces the required minimum operating frequency and external memory bandwidth for real-time operation to 58% (= 41.71/72) and 38% (= 218.63/576) in the 20-k word speech recognition, respectively, compared to the SNU architecture (required frequency: 72 MHz, and external memory bandwidth: 576 MB/s, considering the real-time factor: 0.72) [11] because, our architecture uses advantages derived from the novel techniques: The burst GMM calculation (described in Subsection 5.4.1), the two-stage language model search (Subsection 5.4.4), and the specific cache implementation (denoted in Subsections 5.5.3 and 5.5.4). These techniques enable our architecture to process 60-k word speech recognition in real time. Our architecture can be applied easily to limited

hardware-resource devices, such as ubiquitous/wearable applications, because of its low external memory bandwidth with a large vocabulary language model.

Table 5.2 Comparison with other hardware-based systems.

	This work			Intel (GMM only) [12]	SNU [11]	CMU [8]	CMU(Viterbi only)[9]	
							Single FPGA	Dual FPGAs
Vocabulary (k)	5	20	60	20	20	1	5	5
GMM model # of states	2,000			8,000	3,001	NA	4,147	
GMM model # of distributions	16			16	16	8	NA	
GMM model # of dimensions	25			39	39	39	NA	
Viterbi beam width	500	2,000	4,000	NA	500	NA	NA	
Accuracy (%)	80.13	87.81	89.99	90.10	87.31	89.10	92.28	92.28
Real-time factor	1	1	1	1.15	0.72	2.20	0.17	0.10
Frequency (MHz)	21.71	41.71	66.74	1600	100	50	100	100
External memory	73.64	218.6	549.9	480	800	100	558	950
Resources	75,490 LUTs			NA	13,835 slices	13,449 slices	25,973 slices	2,812,480 slices
Internal memory	458	778	1,133	NA	52	305	566	993
External memory	11.00	27.62	103.7	NA	49.44	3	88	176
Platform	Altera Stratix *II*			Atom Z530	Xilinx Virtex-4	Xilinx XUP	Xilinx Virtex *II* Pro70	

5.7 Summary

We proposed a VLSI architecture to support real-time continuous speech recognition. To reduce the operation cycle time and external memory bandwidth, our architecture contains cache architecture using the locality of speech recognition, beam pruning using a dynamic threshold, two-stage language model search, parallel GMM architecture based on mixture level and frame level, parallel Viterbi architecture, and pipeline operation between Viterbi transition and GMM processing. Results show that our architecture achieves 88.24% required frequency reduction (66.74 MHz) and 84.04% memory bandwidth reduction (549.91 MB/s) for real-time 60-k word continuous speech recognition.

Chapter 6 Low-Standby-Power Decentralized Sound Acquisition

In this chapter, a microphone array network that realizes ubiquitous sound acquisition is proposed. Several nodes with 16 microphones are connected to form a novel huge sound acquisition system to conduct VAD, sound-source localization, and enhancement. The three operations are distributed among nodes. Using the distributed network, we achieve a low-traffic data-intensive array network. To manage nodes' power consumption, VAD is implemented. Consequently, the system uses little power when speech is not active. For sound localization, a network-connected multiple signal classification (MUSIC) algorithm is used. The experimentally obtained result of the sound-source enhancement shows a signal-noise ratio (SNR) improvement of 7.75 dB using 112 microphones. Network traffic is reduced by 99.11% when using 1024 microphones.

6.1 Introduction

In recent years, digital human interfaces have been developed for living spaces, medical centers, robotics, and automobiles. Future applications will enable one person to control thousands of microprocessors without any consciousness of their presence or function. Some face-recognition and speech-recognition systems are in practical use, but most systems operate only in constrained environments or installation conditions such as angle or distance to a device. Users must confront a camera in a typical face-recognition system; a microphone must be near a person's mouth when using a speech-recognition system. For most people, these constraints are not convenient for everyday life. Therefore, various intelligent ubiquitous sensor systems have been developed as new human interfaces [2]. In future systems, numerous cameras and microphones will be located on walls and roofs of living spaces. They will obtain visual information and speech data automatically and support absolutely hands-free systems. As described herein, we specifically examine speech signal processing as a ubiquitous sensor system because a speech interface is a fundamental mode of human communication. Moreover, a speech interface has a much broader range of application.

To implement a microphone array as a realistic ubiquitous sound acquisition system with scalability, we propose division of the huge array into sub-arrays to produce a multi-hop network: an intelligent ubiquitous sensor network (IUSN) [50–53]. The sub-array nodes with some microphones can be set up on the walls and ceiling of a room. Reducing the amount of transmission can be accomplished after introducing multi-hop networking. Each relay node on a routing path must store all temporal multi-channel sound data that the node receives, but not send it. This engenders a large-size buffer memory and large total power dissipation in the system. For that reason, some breakthrough network solution is necessary to reduce the network traffic. In this chapter, we describe how our IUSN solves the problems described above. Multi-hop networking, specific data aggregation, and distributed processing are novel concepts that present great differences from conventional microphone array systems. The performance can be improved easily by increasing the node number, but communication among nodes does not increase much in our system.

As described in this chapter, the distributed sound acquisition system is presented in Section 6.2. The low-power technique with VAD hardware module is explained in Section 6.3. Sections 6.4 and 6.5 will discuss performances and accuracies of the proposed data acquisition scheme, which is based on sound-source localization and sound source separation, using measured data. Section 6.6 presents a summary of the chapter.

6.2 Intelligent Ubiquitous Sensor Network and Its Node

This section presents a description of the proposed perfect aggregation scheme, as implemented with a microphone array system. Figure 6.1 presents a brief description of the proposed IUSN and a functional block diagram of a sub-array node. Sixteen-microphone inputs are digitized with A/D converters. Then the sound information is stored in SRAM. Then, the information is used for sound-source localization and sound source separation. The sound-processing unit including them is activated by the power manager and VAD module to conserve power. The sound-processing unit is turned off if no sound exists around the microphone array. Power management is fundamentally required because enormous microphones waste much power when they are not in use. In our VAD, the sampling frequency can be

reduced to 2 kHz and the number of bits per sample can be set to 10 bits. These values are sufficient to detect human speech, in which case only 3.49 μW is dissipated on a 0.18-μm CMOS process [50]. By separating the low-power VAD module from the sound processing unit, it can turn off the sound processing unit using the power manager. A single microphone is sufficient to detect a signal: the remaining 15 microphones are tuned off as well. Furthermore, not all VAD modules in all nodes need operate. The VAD modules in only a limited number of nodes are activated in the system.

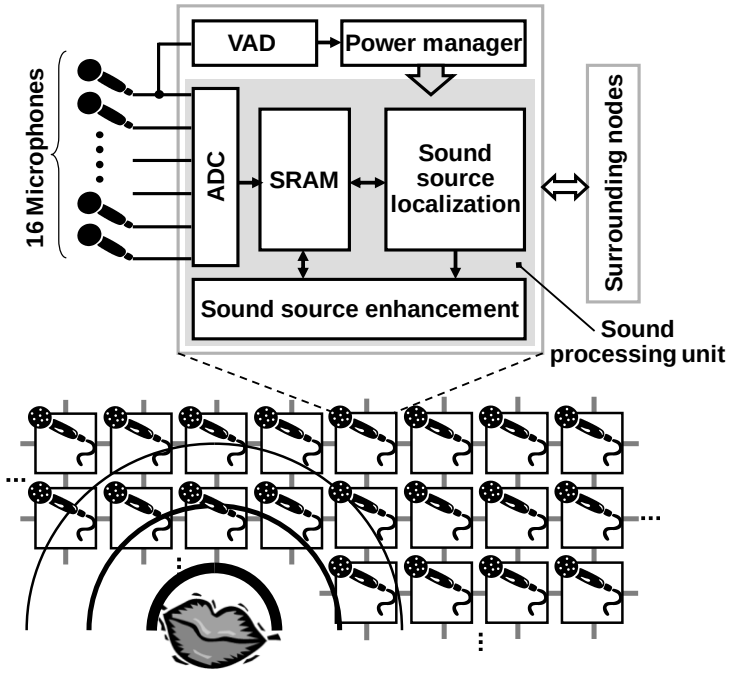

Fig. 6.1 Intelligent ubiquitous sensor network (IUSN) and block diagram of a sub-array node.

Figure 6.2 portrays a flow chart of our system. The salient features of the system are: 1) low-power VAD to activate the entire node, 2) sound-source localization to find sound sources, and 3) sound-source enhancement to reduce the noise level of the sound. The sub-array nodes are connected to support their mutual communication. Therefore, the sound gained by each node can be gathered to improve the sound source's SNR further. The system achieves a huge microphone array through interaction with surrounding nodes. Therefore, the computations can be distributed among nodes. The system has scalability in terms of the number of microphones. Each node preprocesses the acquired sound data. Subsequently, only compressed data—localized and enhanced sound—are communicated.

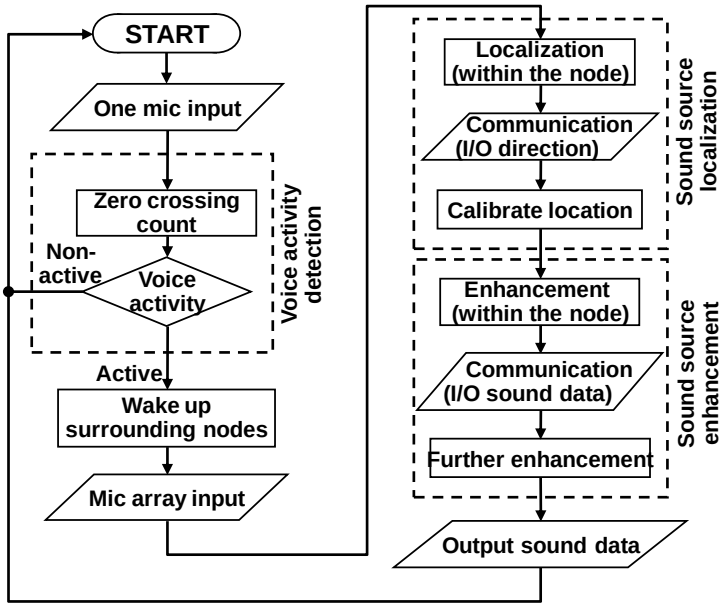

Fig. 6.2 Flow chart of intelligent ubiquitous sensor nodes.

We use low-power zero-crossing VAD, which will be described in Section 6.3. The following Sections 6.4 and 6.5 present discussion of the performances and accuracies of the sound-source localization and sound-source enhancement in our system using measured data. For the system, gathering and processing localization data are important to improve the localization accuracy. Distributed localization data obtained with the MUSIC algorithm can be processed using a communication network in our system. Regarding the sound-source enhancing, we use basic delay-and-sum beam forming both within a node and among nodes [54]. Therefore, the time accuracy between nodes strongly affects the final SNR of the sound source collected with the network.

6.3 Voice Activity Detection

A microphone array network consists of numerous microphones, whose power consumption would easily become high. Therefore, our intelligent ubiquitous sensor node must operate with a limited energy source and must conserve power to the greatest degree possible. Sound processing that does not waste power needlessly is effective because the sound-processing unit and microphone amplifiers consume a certain amount of power only when there is some utterance to record or amplify.

6.3.1 VAD Algorithms

The VAD algorithms determine the difference between noise wave patterns and speech signals, and find the beginning and end of speech. The VAD algorithms have been used progressively in speech recognition and voice over internet protocol (VoIP) applications [55]. For use in real-time applications, such as internet telephony, the complexity of the VAD algorithm must be low, but for almost all VAD algorithms, the power consumption is merely a secondary concern. Consequently, advanced VAD algorithms attract attention for their use of complicated algorithms such as Fourier Transforms, acoustic, and language model bases [56].

To minimize VAD circuit power consumption, the time-domain algorithm is the most suitable. Although the time-domain VAD algorithms' speech detection performance is poor, they are computationally less complex than frequency-domain algorithms. Frequency-domain algorithms have better immunity to low S/N than the time-domain algorithms, but they have higher computational complexity [55]. The zero-crossing VAD, which is a time-domain algorithm, is used to recover some low-energy phonemes that are rejected by an energy-based detector [55]. In Subsections 6.3.2 and 6.3.3, we describe the zero-crossing VAD mechanism and algorithm in detail.

6.3.2 Zero-Crossing VAD Algorithm

In our previous work, we proposed a low-power VAD hardware implementation using a single microphone [50]. This custom hardware uses a zero-crossing algorithm for the VAD. Figure 6.3 portrays the zero-crossing algorithm, which is implemented on an FPGA in the ubiquitous sensor node as well.

The zero crossing is the first intersection between an input signal and an offset line, after the signal crosses the trigger line: The high trigger line or low trigger line. Between a speech signal and non-speech signal, the appearance ratios of this zero crossing differ. The zero-crossing VAD detects this difference and outputs the beginning point and the end point of a speech segment.

For the zero-crossing VAD to detect speech, all requirements are to catch the crossing over the trigger line and the offset line. A detailed speech signal is unnecessary. For that reason, the sampling frequency and the number of bits can be reduced. Once the VAD module detects a speech signal, the main signal processor begins to run and the

sampling frequency and the number of bits are increased to sufficient values. These parameters, which decide the analog digital converter (ADC) specifications, are changeable depending on the specific applications that are integrated on the system. As described herein, we adopt standard parameters: The quality of 16 kHz sampling frequency and 16 bits per sample, for which most speech-recognition systems require continuous sensing [49]. Furthermore, only for the VAD algorithm, the sampling frequency is set to 2 kHz and the number of bits per sample is set to 10 bits. These values are sufficient to detect human speech, in which case only 3.49 μW is dissipated on a 0.18-μm CMOS process. By separating the low-power VAD module from the sound processing unit, it can turn off the sound processing unit using the power manager. A single microphone is sufficient to detect a signal. The remaining 15 microphones are turned off as well. Furthermore, not all VAD modules in all nodes need operate. The VAD modules in only a limited number of nodes are activated in the system.

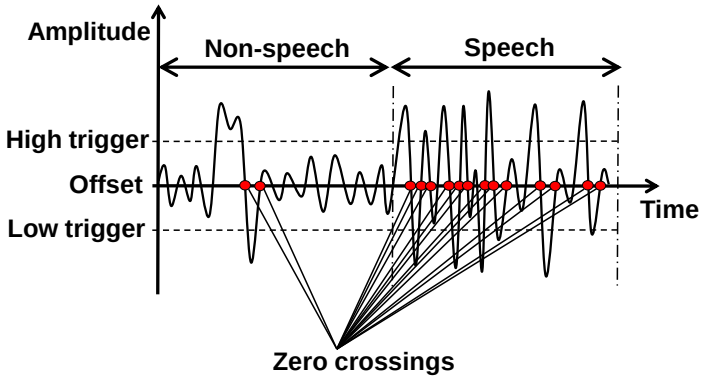

Fig. 6.3 Zero-crossing point example. The offset line shows the direct current (DC) component.

6.3.3 Modification the Zero-Crossing Algorithm

When considering the hardware implementation, it is important to adapt to the ADC circuits. The direct current (DC) offset presented in Fig. 6.3 is the mean value of the ADC outputs; it changes depending on the temperature, voltage, noise, and other operating parameters. Therefore, the output from ADC is usually normalized, such as to a range of 0 to 1, or -1 to 1, to operate correctly as a continuous system. However, to minimize the total computation of VAD, no floating point calculations can be used: all operations must be integer arithmetic. For that reason, we adopt a DC offset adjust process that is specialized for the zero-crossing algorithm.

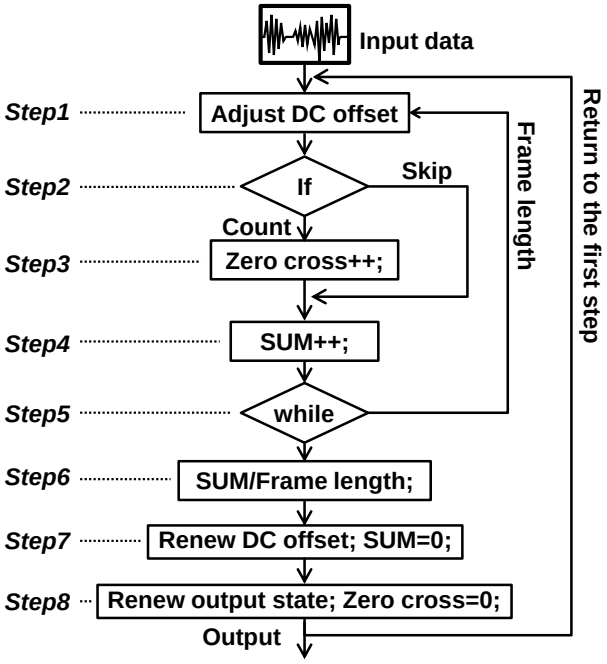

Fig. 6.4 Zero-crossing VAD algorithm flow.

Figure 6.4 portrays the VAD flowchart with the DC offset process and details of each step in this flow. The following items describe concrete steps.

Step 1: Input data are adjusted to avoid overflowing.

Step 2: Input data are judged as to whether they have a zero cross or not.

Step 3: The zero-cross count is incremented if the input data exceed a threshold.

Step 4: To calculate the mean value in the present frame, the input data are added to the temporary sum.

Step 5: Input data are counted to control the frame length.

Step 6: The temporary sum is divided by the frame length only with the shift operation; the mean value in the present frame is obtained.

Step 7: The DC offset is adjusted according to the mean value.

Step 8: The output state is renewed based on the zero crossing count; the processing returns to the first step.

In ***step 6***, the average of the input amplitudes is obtained using integer arithmetic alone. The condition precedent is that the frame length corresponds to a multiple of 2 to obtain the average simply using the adder and the shifter circuits. After obtaining the

average of the ADC outputs, VAD can count the zero crossings (***steps 2*** and ***3***). The total calculation amount from ***step 1*** to ***step 8*** is approximately 3 kops.

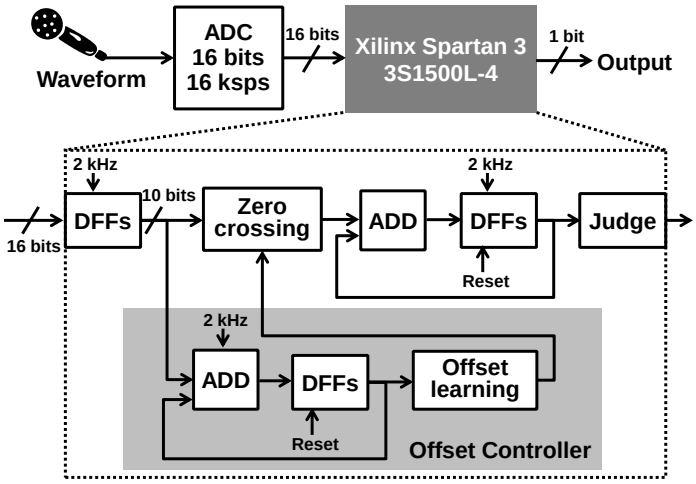

Fig. 6.5 Block diagrams of the integrated devices. The D flip-flop (DFF) circuits keep up each input data asynchronously.

6.3.4 Hardware Implementation of VAD

To clarify the proposed VAD performances, we implemented our proposed VAD algorithm using an FPGA (Spartan 3; Xilinx Inc.). We measured the FPGA board power consumption including the ADC, excepting the microphone. Figure 6.5 portrays the board block diagram. The supply voltage to the board is 5 V. The ADC that we used takes a sample of 10 bits at a sampling rate within 16 ksps. This sampling rate is controlled using a dedicated circuit configured on FPGA. In Fig. 6.5, the signals sampled by ADC input to the FPGA chip directly and FPGA chip output the state signal whether the input signal includes speech or not. The calculations executed in this FPGA chip are almost identical to the flow depicted in Fig. 6.4. The zero crossing, the offset controller and the judgment modules presented in Fig. 6.5 respectively correspond to ***steps 1*** and ***2***, ***steps 4***, ***6***, and ***7***, and ***step 8***, in Fig. 6.4. All calculations are integrated using integer arithmetic. Table 5.1 presents the device utilization summary. The slice flip flops and 4-input LUTs are, respectively, 1,015 and 3,831.

Table 6.1 Device utilization summary.

Logic utilization	Used	Available	Utilization
# of slice flip flops	1,015	26,624	3%
# of 4-input LUTs	3,831	26,624	14%

Figure 6.6 presents the equipment—the FPGA board, microphone, and tester—used for the experiments described herein. Measurement results show that the board, except the microphone, requires 0.42 mA electric current and 2.10 mW power consumption. Results show that the stand-alone VAD module can run about 70 hr with a 150 mAh battery.

All blocks of the zero-crossing VAD module are implemented using CMOS 0.18-μm process technology. Figure 6.7 depicts the layout plot. The power consumption is 3.49 μW at the 1.8 V supply voltage and 100 kHz frequency, which implies 1,700-day operation with the 150 mAh battery.

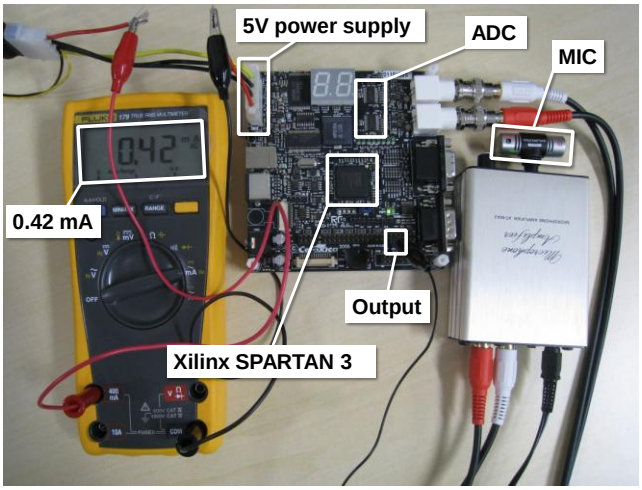

Fig. 6.6 FPGA board with a microphone and a current tester.

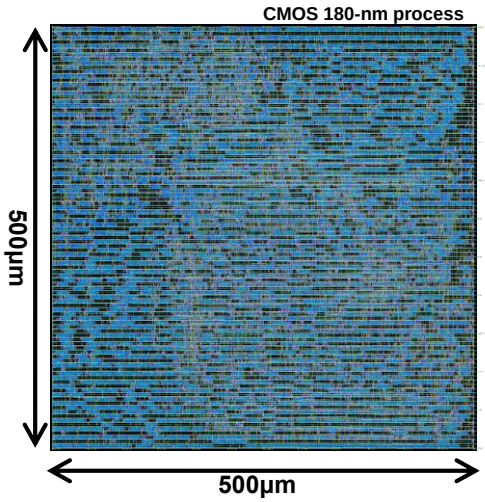

Fig. 6.7 Layout plot of the zero-crossing algorithm integrated using CMOS 0.18-μm process technology.

6.3.5 Experimental Results

The SNR easily affects the zero-crossing VAD algorithms because they are based only on changes in amplitude. For SNR dependencies of the VAD performance, we experiment using various SNR environments of -20 – 20 dB. Input signals in this experiment are generated by adding recorded environmental sound, which is used for continual noise, to original speech data with gain controlling. In every SNR condition, we use identical 15-min speech data comprising 24 ATR phoneme balanced sentences [57]. The frame length of the VAD algorithm is 256 samples. In each SNR condition experiment, the number of VAD results is 7,030 samples. For this experiment, we counted the quantities of surplus, and deficit VAD results. Each condition is defined as follows.

False acceptance (FA): A case in which the VAD output is speech, although the input frame is non-speech.

False rejection (FR): A case in which the VAD output is non-speech, although the input frame is speech.

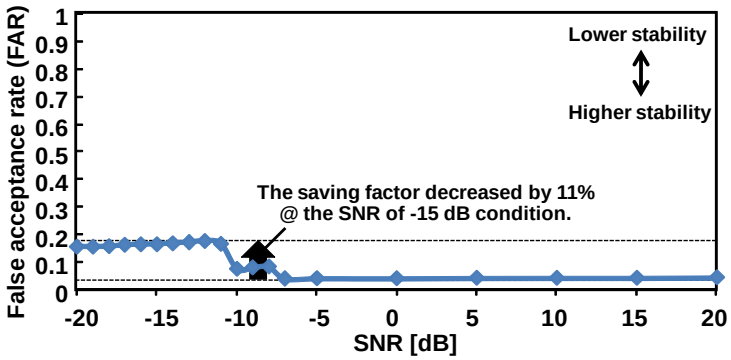

Fig. 6.8 The false acceptance rate (FAR) in VAD outputs using the number of non-speech frames of the recorded condition as normalized criteria.

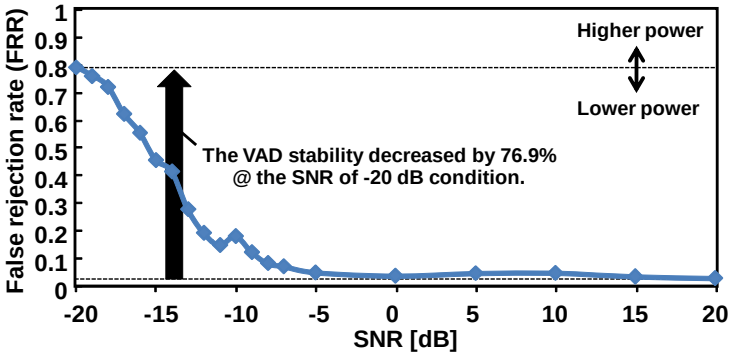

Fig. 6.9 The false rejection rate (FRR) in VAD outputs using the speech frames of the recorded condition as normalized criteria.

Figures 6.8 and 6.9 respectively present results of ***FA***, and ***FR*** VAD output quantities. The figures show that the power saving factor and stability of the zero-crossing VAD decreases according to deterioration of the SNR.

6.4 Proposed Sound Acquisition Scheme

As described in this chapter, we specifically examine microphone array networks to obtain high-SNR sound data. To enhance the sound data, it is necessary to gather the sound data among numerous nodes and the network traffic is bottleneck. Then we produce it as a multi-hop network. We propose a perfect aggregation solution that is specialized for obtaining high-SNR sound data in this section.

Some data aggregation techniques have been proposed to reduce network traffic for sensor networking. Figure 6.10 presents network traffic with and without data aggregation. Without data aggregation, the network traffic is concentrated around the base station. An aggregation scheme should be chosen carefully according to the application. Data aggregation is classifiable as lossy and lossless [58]. Our aggregation method is chosen according to the former application. For applications such as reproduction of sound fields, lossless aggregation is suitable. However, irreversible aggregation is sufficient for applications such as ours, which are intended solely to improve the SNR of sound. Perfect aggregation [59] and beam forming [60] are lossy aggregations. With perfect aggregation, a sensor node aggregates the received data into one unit of data and then sends it to the next hop [61]. Therefore, perfect aggregation can reduce traffic on a grand scale.

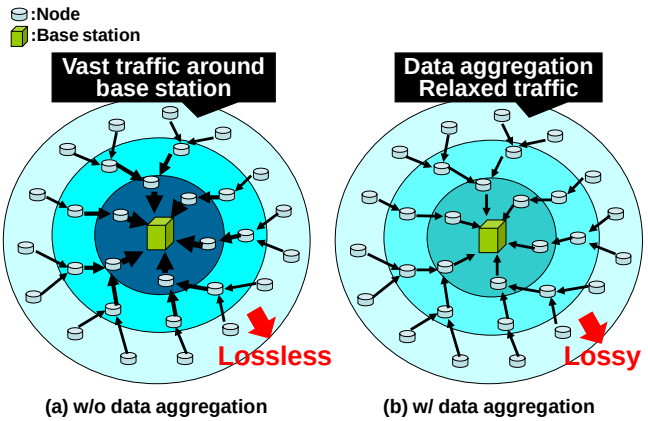

Fig. 6.10 Network traffic with (a) lossless and (b) lossy multi-hop networks.

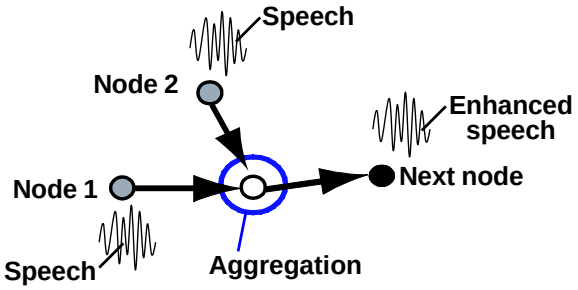

Fig. 6.11 Example of perfect aggregation among neighboring nodes.

6.4.1 Proposed Data Aggregation Scheme

In this subsection, we introduce the proposed perfect aggregation method. Figure 6.11 presents an example of aggregation. In the figure, speech data acquired in nodes 1 and 2 are aggregated to single enhanced speech data in the aggregation node. Then the speech data are sent to the next node.

To obtain high-SNR speech data, the aggregation algorithm must be eligible for a chosen sound-source enhancing method that lowers the noise signal level. Major sound-source enhancing methods are geometric techniques, which use position information, and statistical techniques, which use no position information. For the proposed system, delay-and-sum beam forming, which is categorized as a geometric method, is chosen because the node positions in the network are known. This method produces less distortion than statistical techniques do. Fortunately, it requires only a small amount of computation. For distributed processing in sound-source enhancing, it is applicable easily because it is based on summations (Fig. 6.12). The key point for delay-and-sum beam forming among distributed nodes is how to obtain time differences (W_i: phase differences in sound waves) among neighboring nodes.

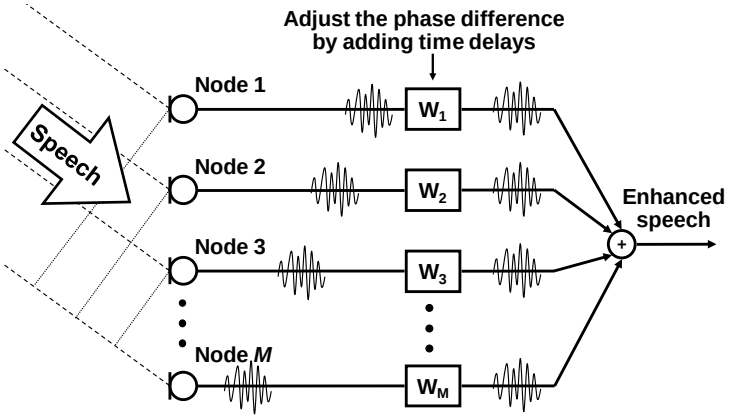

Fig. 6.12 Delay-and-sum beam-forming mechanism.

Time differences among neighboring nodes are calculable from header information in a packet, which comprises a sound-source coordinate and a coordinate of each node. As a matter of course, the coordinate origin must be calibrated to a unique point. In aggregation using the timing data described above, all temporal speech data are adjusted by adding time differences and summing them to a single speech datum for uploading the signal. Consequently, high-SNR speech data can be acquired at the base station.

6.4.2 Sound Source Localization Algorithm

However, without a precise sound source coordinate, the delay-and-sum beam-forming method does not operate effectively. For this reason, a basic sound-source localization algorithm with a high degree of accuracy is important to produce a perfect aggregation scheme. To achieve highly accurate sound-source localization, we have already proposed a hierarchical sound-source localization method [51] based on the multiple signal classification (MUSIC) algorithm [62–64].

The MUSIC algorithm is based subspace techniques for estimating the direction of arrivals (DOAs) of multiple signal sources. We adopt the basic MUSIC algorithm for intelligent ubiquitous sensor system. We assume an array composed of N sensors that receives signals from L (L < N) sources. The N × 1 array output at time t can be modeled as [63, 64].

$$\mathbf{x}(t) = \mathbf{A}(\boldsymbol{\theta})\mathbf{s}(t) + \mathbf{n}(t) \tag{6.1}$$

where $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_L]^T$ is the DOA vector, $\boldsymbol{s}(t)$ is the $L \times 1$ vector of signal waveforms, $\boldsymbol{n}(t)$ is the $N \times 1$ vector of noise and interference, and

$$\mathbf{A}(\boldsymbol{\theta}) = [\mathbf{a}(\theta_1), \cdots, \mathbf{a}(\theta_L)] \tag{6.2}$$

is the $N \times L$ signal steering matrix. We presume that no coherent signals exist and that the noise is spatially white. Consequently, the $N \times N$ array covariance matrix can be written as [39, 40].

$$\mathbf{R}_{\mathbf{x}} = \mathrm{E}\{\mathbf{x}(t)\mathbf{x}^H(t)\} = \mathbf{A}\mathbf{R}_{\mathbf{s}}\mathbf{A}^H + \sigma^2\mathbf{I} \tag{6.3}$$

where $\mathbf{R}_{\mathbf{s}} = \mathrm{E}\{\mathbf{s}(t)\mathbf{s}^H(t)\}$ is the source covariance matrix, σ^2 is the sensor noise variance, and $\boldsymbol{I}$ is the identity matrix. By a unitary matrix $\boldsymbol{E}$, the $\boldsymbol{R}_x$ can be orthogonal transformed as

$$\mathbf{E}^H\mathbf{R}_{\mathbf{x}}\mathbf{E} = \mathrm{diag}[\lambda_1, \cdots, \lambda_L, 0, \cdots, 0] + \sigma^2\mathbf{I} \tag{6.4}$$

where $\lambda_1,\cdots,\lambda_L$ are eigenvalues of $\boldsymbol{R}_x$. The unitary matrix $\boldsymbol{E}$ can be represented as $\mathbf{E}=[\mathbf{e}_1,\cdots,\mathbf{e}_N]$ where $\mathbf{e}_1,\cdots,\mathbf{e}_N$ are eigenvectors of $\boldsymbol{R}_x$. Consequently, equation (6.4) can be deformed as

$$\mathbf{R}_{\mathbf{x}}=\sum_{n=1}^{L}(\lambda_n+\sigma^2)\mathbf{e}_n\mathbf{e}_n^H+\sigma^2\sum_{n=L+1}^{N}\mathbf{e}_n\mathbf{e}_n^H \tag{6.5}$$

using the relation, $\mathbf{E}\mathbf{E}^H=\mathbf{E}^H\mathbf{E}=\mathbf{I}$. Between (6.3) and (6.5), $\mathbf{a}(\theta_1),\cdots,\mathbf{a}(\theta_L)$ and $\mathbf{e}_{L+1},\cdots,\mathbf{e}_N$ are orthogonal. Consequently, the source location can be found by plotting the following quantity as a function of θ:

$$P(\theta)=\frac{1}{\mathbf{a}(\theta)^H\sum_{n=L+1}^{N}\mathbf{e}_n\mathbf{e}_n^H\mathbf{a}(\theta)} \tag{6.6}$$

6.4.3 Three-Dimensional Sound Source Localization

We adopt the above MUSIC algorithm as the perfect aggregation method. We will divide the localization into two layers: 1) relative direction estimation within a node and 2) absolute location estimation by exchanging the results through the network. The MUSIC algorithm is chosen for direction estimation within a node because the number of microphones and their buffer memory on a node is limited; the MUSIC algorithm can achieve higher resolution using fewer microphones. To find a relative direction, the sound source probability $P(\theta, \varphi)$ must be calculated on each node. Once the relative direction to the sound source is obtained, its information is transferred to neighboring nodes to proceed to the next step.

For the system, gathering and processing localization data are important to improve the localization accuracy. Distributed localization data obtained with the multiple signal classification (MUSIC) algorithm [62–64] can be processed using a communication network in our system. We will localize the absolute sound-source location in the network layer. The authors have already proposed a calibration method with a three-dimensional coordinate of the sound source, as presented briefly in Fig. 6.13 [51]. First, the maximum $P(\theta, \varphi)$ and corresponding θ and φ are calculated on each node using the MUSIC algorithm. We alternatively adopt the shortest line segment connecting two lines because we can usually find no exact intersection in the three-dimensional space. We presume a point that divides the shortest line segment by

the ratios of $P(\theta, \varphi)$ as an intersection. The sound source is localized by calculating the center of gravity as well, with the obtained intersections.

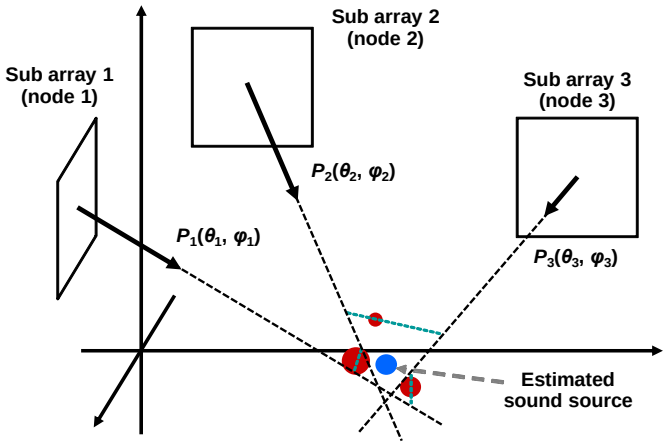

Fig. 6.13 Three-dimensional sound-source localization.

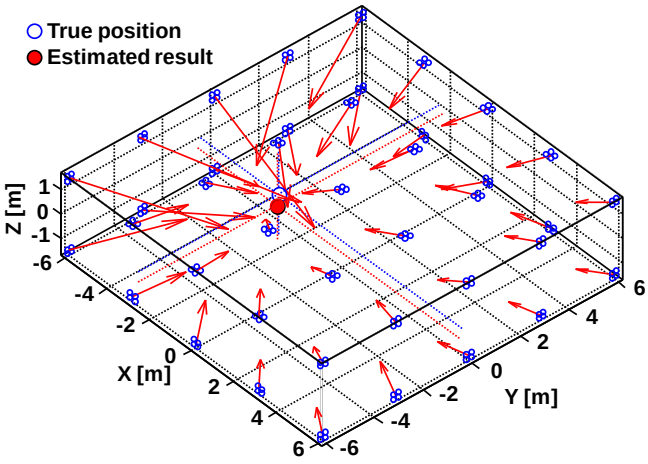

Fig. 6.14 Sound-source localization experiment.

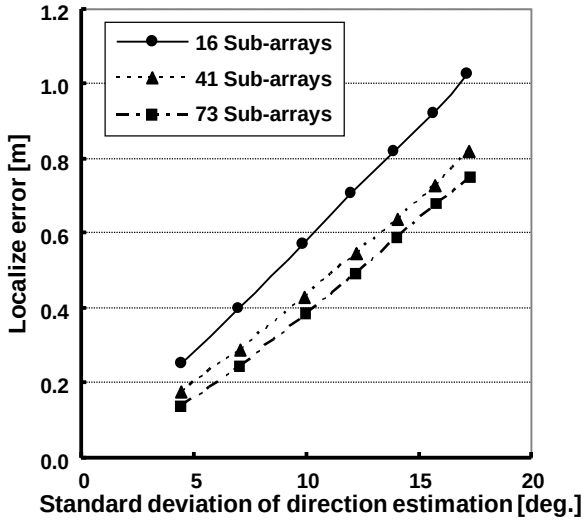

Fig. 6.15 Sound-source localization accuracy.

6.4.4 Simulation Results

We verified the hierarchical localization by simulation, assuming that an estimation result has a variation on every node. Figure 6.14 presents an example of the experiments, for which the observed range is 12 m × 12 m. The localization accuracy is portrayed in Fig. 6.15. The localization error is smaller when the number of arrays is large and the direction estimation is precise. Results show that the effective means to make the localization accurate is to minimize the direction error. However, the number of sub-arrays does not give much impact, although the number of sub-arrays strongly affects sound enhancement, as described later.

Although the coordinate data can be calibrated with nodes, the time stamp of each speech cannot be calibrated in this scheme. Time synchronization is an important issue for the delay-and-sum beam-forming method. Timers of each sensor node, even among neighboring nodes, have dispersion by various environmental and device-origin effects. For that reason, the time synchronization method among nodes in sensor networks is important for the perfect aggregation scheme. Various means of time synchronization for a sensor network have been examined: Reference Broadcast Synchronization (RBS) [65], Timing-sync Protocol for Sensor Networks (TPSN) [66], and Flooding Time Synchronization Protocol (FTSP) [67]. Using a time synchronization protocol, infection to the SNR by the timer variation can be disregarded. For low-power multi-hop sensor networks such as microphone array networks, FTSP is the most suitable in terms of power consumption.

6.5 Implementation of the Microphone Array System

6.5.1 Implementation

We implemented the proposed perfect aggregation scheme in an actual sensor network with microphone arrays. Using it, we verified the SNR performance. Regarding the sound-source enhancement, we use the basic delay-and-sum beamforming (denoted in Section 4) both within a node and among nodes [54]. Therefore, the time accuracy between nodes gives a great impact on the final SNR of the sound source collected with the network.

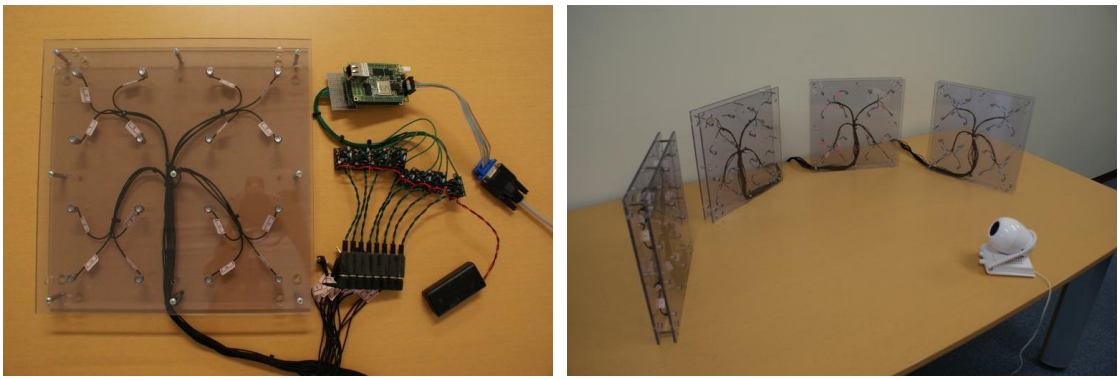

Fig. 6.16 System photographs: intelligent ubiquitous sensor node and a microphone array comprising sub-arrays.

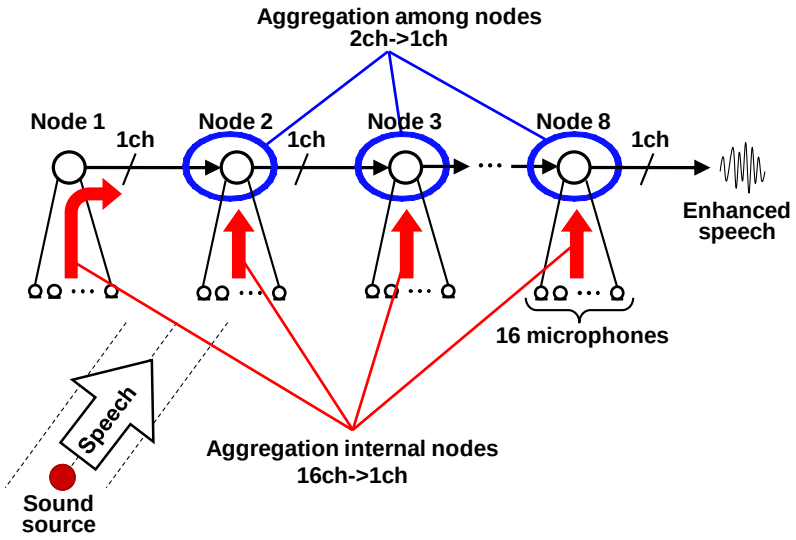

Fig. 6.17 Experiment diagrams.

As a real design, we implemented the intelligent ubiquitous sensor node on a field-programmable gate array board (FPGA, SZ410, Suzaku; Atmark Techno Inc.) and microphones (ECM-C10; Sony Corp.). Figure 6.16 portrays photographs of the prototype system. Each node operates the sound-source localization with its 16 microphones. Consequently, each node aggregates 16 sounds to a single sound using delay-and-sum beam-forming enhancing the objective sound. Then the sound is transmitted to neighboring nodes. In this experiment, seven nodes are connected linearly, as portrayed in Fig. 6.17. Then they aggregate the data of one side to the other side. One aggregated audio datum, which has higher SNR, is obtained at the last node. All sounds from all 112 microphones are aggregated to one channel.

Figure 6.18 shows that the SNR improvement of 7.75 dB was gained with 122 microphones. We expect to achieve 15 dB or greater improvement using several tens of sub-arrays and hundreds of microphones.

Next, we compared network-traffic costs with the proposed perfect aggregation and without data aggregation. Figure 6.19 depicts examples of the traffic data sizes with and without proposed perfect data aggregations. The network traffic is increased by 16 channels on every node if without the data aggregation, which enables lossless sound acquisition and realizes applications such as the reproduction of sound fields, but results in heavy traffic (Fig. 6.19(a)). However, using the proposed perfect aggregation, the network traffic is always 1 channel (Fig. 6.19(b)). This small-channel network implies lossy sound source acquisition. However, the sound-source localization algorithm and the sound-source enhancing algorithm achieve high SNR sound acquisition for an intended sound source.

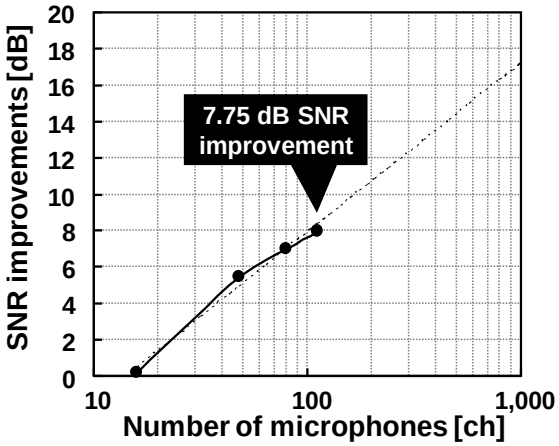

Fig. 6.18 SNR improvements vs. the number of microphones.

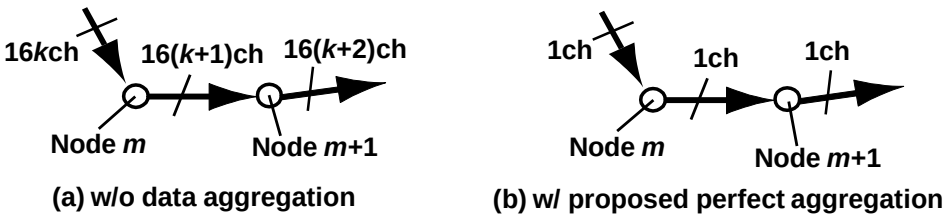

Fig. 6.19 Examples of traffic data sizes: (a) without and (b) with the proposed perfect data aggregations.

Figure 6.20 shows normalized (the criterion for normalization is the network cost in

the proposed 32-ch perfect data aggregation) network costs with and without the proposed perfect data aggregations. For 1024-ch microphones, the proposed perfect aggregation achieves 99.11% network traffic reduction, which demonstrates that the proposed scheme maintains low network traffic costs consistently. It is applicable to a future larger-scale microphone array for a sound acquisition system.

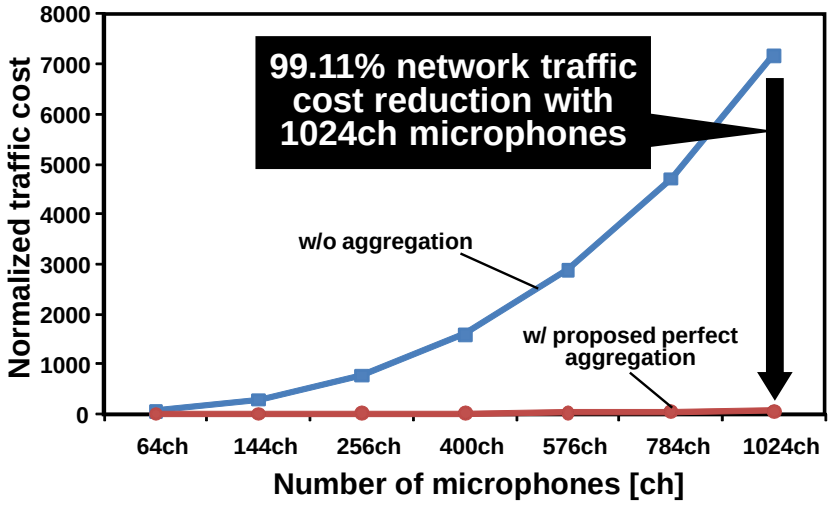

Fig. 6.20 Normalized traffic cost vs. the number of microphones.

6.6 Future Work

Our proposed system is implemented under the condition that the number of sound sources is one. Multiple sound sources and their separation are to be investigated as future work. Compared to single sound enhancement, multiple sound separation necessitates sound source tracking because enhanced performance depends on the time-series direction and because the system must recognize which sound source is the targeted source. In our proposed aggregation scheme, when considered with the multiple sound sources and their separations, the network traffic is increased linearly according to the number of sound sources: the issues are the multiple sound-source localization and tracking, plus increasing network traffic for our future system.

Although the performance of sound-source enhancement is known to be improved as the number of microphones increase [17]; the power consumption and traffic costs increase as well. Therefore, for actual implementation, the optimum deployment and power control (which nodes should be operated / which node can sleep) among nodes must be investigated further using a large-scale setup.

6.7 Summary

As described in this chapter, we propose a novel perfect aggregation scheme that is specialized for sound acquisition systems comprising numerous microphones. The microphone array network using 16-microphone sub-arrays performed the following three operations in a node and a network: 1) low-power VAD to activate the entire node, 2) sound-source localization to find sound sources, and 3) sound-source enhancement to improve the SNR. We implemented an actual microphone array network that realizes the ubiquitous sound acquisition system, and verified that the proposed scheme reduces the network traffic and saves resources such as power and memory size.

Low-power VAD was implemented to manage the node's power consumption. The system uses only very low power when speech is not active. The VAD module dissipates only 3.49 μW on a 0.18-μm CMOS process. Sound-source localization is processed with the distributed nodes. The proposed sound-source localization scheme uses a two-layered hierarchical algorithm. The experimentally obtained result of the sound-source enhancement shows SNR improvement of 7.75 dB using 112 microphones. The system will achieve an SNR of 15 dB if the entire microphone network has more than several hundred microphones. We confirmed that the system achieves a 99.11% traffic amount reduction when using 1024 microphones.

Chapter 7 Conclusion

This thesis describes terms of the low-power techniques of memory and digital architecture for signal processing systems. This study is presented in terms of four main points, as described below.

(1) Active power consumption, especially on SRAM (Chapter 3)
(2) Minimum operating voltage of SRAM for DVFS (Chapter 4)
(3) Memory bandwidth considering a large vocabulary (Chapter 5)
(4) Standby-power of sound acquisition system including network traffic (Chapter 6)

These architectural techniques contribute to resolution of some critical issues that remain as obstacles to high-performance hardware signal processing systems.

Before presenting the practical architectural techniques to improve the power efficiencies, the actual limitations or undesirable problems in developing advanced speech recognition systems were introduced in Chapter 2. Increasing memory bandwidth, which strongly affects the bus frequency on VLSI, was introduced and discussed in terms of its effects on the size of the language model dictionary. The issues of increasing the active power dissipations of SRAM were also introduced because of the slowdown in voltage scaling in spite of the increment of clock frequency. Consequently, a power reduction technique is demanded. From another perspective, the standby power of the sound acquisition system is increased. Furthermore, to reduce the minimum operating voltage of SRAM with maintenance of the performance of nominal supply voltage, dynamic voltage and frequency scaling (DVFS) are needed, and the minimum operating voltage of SRAM is a critical issue.

As explained in Chapter 3, dual-port SRAM design in terms of area, speed, and readout power in a 45-nm process technology were examined. Although the 8T SRAM has the lowest transistor count, and although it is the most area-efficient, the readout power is large and the cycle time increases because of peripheral circuits. The 10T differential-port SRAM would operate fastest if the differential voltage were set to 50 mV. The 10T SRAM with a single-end read port consumes the least power.

In Chapter 4, the 9T/18T SRAM, which provides a better read margin than that of the

conventional 7T/14T SRAM, was proposed. The 9T/18T bitcell topology comprises a conventional 7T/14T cell and an additional read port. In the 9T/18T SRAM, the additional read port can operate irrespective of the static noise margin because of the disturb-free read operation. Consequently, the 9T/18T SRAM is more stable than the 7T/14T SRAM under threshold-voltage (V_{th}) variation. We fabricated the proposed SRAM using a 65-nm triple-well process. Measurement results show that the dependable read mode using the additional read port can suppress V_{th} variation and reduce the operation voltage to 0.45 V, although the dependable read mode using internal bitlines needs 0.54 V. Body biasing control reveals the weakness of the conventional 7T/14T SRAM read process and clarifies that the 9T/18T SRAM provides greater reliability against unbalanced process corners, particularly the SF corner, than the 7T/14T SRAM does.

In Chapter 5, a VLSI architecture to support real-time continuous speech recognition was proposed. To reduce the operation cycle time and external memory bandwidth, our architecture includes a cache architecture using the locality of speech recognition, beam pruning using a dynamic threshold, two-stage language model search, parallel GMM architecture based on a mixture level and a frame level, parallel Viterbi architecture, and pipeline operation between Viterbi transition and GMM processing. Results show that our architecture achieves 88.24% required frequency reduction (66.74 MHz) and 84.04% memory bandwidth reduction (549.91 MB/s) for real-time 60-k word continuous speech recognition.

In Chapter 6, a novel perfect aggregation scheme was proposed. It is specialized for sound acquisition systems comprising numerous microphones. The microphone array network using 16-microphone sub-arrays performed the following three operations in a node and a network: 1) low-power VAD to activate the entire node, 2) sound-source localization to identify sound sources, and 3) sound-source enhancement to improve the SNR. We implemented an actual microphone array network that realizes the ubiquitous sound acquisition system. Results verified that the proposed scheme reduces the network traffic and efficiently uses resources such as power and memory size. Low-power VAD was implemented to manage the node's power consumption. The system achieves low power when speech is not active. The VAD module dissipates only 3.49 μW on a 0.18-μm CMOS process. Sound-source localization is processed with the

distributed nodes. The proposed sound-source localization scheme uses a two-layered hierarchical algorithm. The experimentally obtained result of the sound-source enhancement shows SNR improvement of 7.75 dB using 112 microphones. The system will achieve an SNR of 15 dB if the entire microphone network has more than several hundred microphones. We confirmed that the system can produce a 99.11% traffic amount reduction when using 1024 microphones.

It is hoped that these considerations will help any designer to develop advanced hardware speech recognition systems to accommodate a more-than-60-k word speech recognizer with a large-scale microphone array system involving the DVFS voltage control scheme as well as 32-nm or 22-nm next feature devices.

References

[1] C.C. Wu, D.W. Lin, A. Keshavarzi, C.H. Huang, C.T. Chan, C.H. Tseng, C.L. Chen, C.Y. Hsieh, K.Y. Wong, M.L. Cheng, T.H. Li, Y.C. Lin, L.Y. Yang, C.P. Lin, C.S. Hou, H.C. Lin, J.L. Yang, K.F. Yu, M.J. Chen, T.H. Hsieh, Y.C. Peng, C.H. Chiou, C.J. Lee, C.W. Huang, C.Y. Lu, F.K. Yang, H.K. Chen, L.W. Weng, P.C. Yen, S.H. Wang, S.W. Chang, S.W. Chuang, T.C. Gan, T.L. Wu, T.Y. Lee, W.S. Huang, Y.J. Huang, Y.W. Tseng, C.M. Wu, E. Ou-Yang, K.Y. Hsu, L.T. Lin, S.B. Wang T.M. Kwok, C.C. Su, C.H. Tsai, M.J. Huang, H.M. Lin, A.S. Chang, S.H. Liao, L.S. Chen, J.H. Chen, P.S. Lim, X.F. Yu, S.Y. Ku, Y.B. Lee, P.C. Hhieh, P.W. Wang, S.S. Lin, S.S. Lin, Y.H. Chiu, H.J. Tao, and M. Cao, "High Performance 22/20nm FinFET CMOS Devices with Advanced High-K/Metal Gate Scheme," Proceedings of IEEE International Electron Devices Meeting (IEDM), pp.27.1.1-27.1.4, December 2010.

[2] K. Prall and K. Parat, "25nm 64Gb MLC NAND Technology and Scaling Challenges," Proceedings of IEEE International Electron Devices Meeting (IEDM), pp. 5.2.1-5.2.4, December 2010.

[3] International Technology Roadmap for Semiconductors 2005 (online), available from <http://www.itrs.net/Links/2005ITRS/Home2005.htm> (accessed 2010-05-27).

[4] J. Miyakoshi, Y. Murachi, K. Hamano, T. Matsuno, M. Miyama, and M. Yoshimoto, "A Low-Power Systolic Array Architecture for Block-Matching Motion Estimation," IEICE Transactions on Electronics, vol. E88-C, no. 4, pp. 559-569, April 2005.

[5] S. Lin, Y.B. Kim, and F. Lombard, "Design and Analysis of a 32 nm PVT Tolerant CMOS SRAM Cell for Low Leakage and High Stability," Elsevier Science Publishers B. V. the VLSI Journal on Integration, vol. 43, no. 2, pp. 176-187, April 2010.

[6] H. Ohira, K. Kawakami, M. Kanamori, Y. Morita, M. Miya,a, and M. Yoshimoto, "A Feed-Forward Dynamic Voltage Control Algorithm for Low Power MPEG4 on Multi-Regulated Voltage CPU," IEICE Transaction on Electronics, vol. E87-C, no. 4, pp. 457-465, April 2004.

[7] A. Lee, T. Kawahara and K. Shikano, "Julius – an open source real-time large vocabulary recognition engine," Proceedings of European Conference on Speech Communication and Technology (EUROSPEECH), pp. 1691-1694, September 2001.

[8] E. C. Lin, K. Yu, R. A. Rutenbar, and T. Chen, "A 1000-Word Vocabulary, Speaker-Independent, Continuous Live–Mode Speech Recognizer Implemented in a Single FPGA," Proceedings of ACM/SIGDA 15th International Symposium on Field programmable gate arrays (FPGA), pp. 60-68, February 2007.

[9] E. C. Lin and R. A. Rutenbar, "A Multi-FPGA 10x-Real-Time High-Speed Search Engine for a 5000-Word Vocabulary Speech Recognizer," Proceedings of ACM/SIGDA 17th International Symposium on Field Programmable Gate Arrays (FPGA), pp.83-92, Februry 2009.

[10] Y. Choi, K. You, and W. Sung, "FPGA-based implementation of a real-time 5000-word continuous speech recognizer," Proceedings of 16th European Signal Processing Conference (EUSIPCO), August 2008.

[11] Y. Choi, K. You, J. Choi, and W. Sung, "A Real-Time FPGA-Based 20,000-Word Speech Recognizer with Optimized DRAM Access," IEEE Transactions on Circuits and Systems I, vol. 57, no. 8, pp. 2119-2131, February 2010.

[12] T. Ma and M. Deisher, "Novel CI-Backoff Scheme for Real-Time Embedded Speech Recognition", Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 1614-1617, March 2010.

[13] M. Brandstein and D. Ward: Microphone Arrays, "Signal Processing Techniques and Applications," Springer, 2001.

[14] Y. Tamai, S. Kagami, H. Mizoguchi, K. Sakaya, K. Nagashima, and T. Takano, "Circular Microphone Array for Meeting System," Proceedings of IEEE Sensors, vol. 2, pp. 1100-1105, October 2003.

[15] T. Wakabayashi, K. Takahashi, H. Iwakura, "Independent Component Analysis using Large Microphone Array," Technical report of IEICE EA, vol. 102, no. 322, pp. 29-34, September 2002, in Japanese.

[16] H. F. Silverman, W. R. Patterson III, and J. L. Flanagan, "The Huge Microphone Array," Journal of IEEE Concurrency, vol. 6, no. 4, pp. 36-46, Oct-Dec. 1998 and vol. 7, no. 1, pp. 32-47, Jan-Mar. 1999.

[17] E. Weinstein, K. Steele, A. Agarwal, and J. Glass: Loud, "A 1020-Node Modular Microphone Array and Beamformer for Intelligent Computing Spaces," MIT, MIT/LCS Technical Memo, MIT-LCS-TM-642, 2004.

[18] Y. Murachi, K. Hamano, T. Matsuno, J. Miyakoshi, M. Miyama, and M. Yoshimoto, "A 95 mW MPEG2 MP@HL Motion Estimation Processor Core for Portable High-Resolution Video Application," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol.E88-A, no.12, pp.3492-3499, December 2005.

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

[20] Y-W. Huang, T-C. Chen, C-H.Tsai, C-Y. Chen, T-W. Chen, C-S. Chen, C-F. Shen, S-Y. Ma, T-C. Wang, B-Y. Hsieh, H-C. Fang, and L-G. Chen, "A 1.3TOPS H.264/AVC Single-Chip Encoder for HDTV Applications," IEEE International Solid-State Circuits Conference (ISSCC) 2005 Digest of Technical Paper, pp.128-129, February 2005.

[21] K. Takeda, Y. Hagihara, Y. Aimoto, M. Nomura, Y. Nakazawa, T. Ishii, and H. Kobatake, "A Read-Static-Noise-Margin-Free SRAM Cell for Low-V_{dd} and High-Speed Applications," IEEE Journal of Solid-State Circuits, vol.41, no.1, pp.113-121, January 2006.

[22] R. K. Krishnamurthy, A. Alvandpour, G. Balamurugan, N. R. Shanbhag, K. Soumyanath, and S. Y. Borkar, "A 130-nm 6-GHz 256 × 32 bit leakage-tolerant register file," IEEE Journal of Solid-State Circuits, vol. 37, no.5, pp.624-632, May 2002.

[23] H. Fujiwara, K. Nii, J. Miyakoshi, Y. Murachi, Y. Morita, H. Kawaguchi, and M. Yoshimoto, "A Two-Port SRAM for Real-Time Video Processor Saving 53% of Bitline Power with Majority Logic and Data-Bit Reordering," Proceedings of ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED), pp.61-66, October 2006.

[24] H. Noguchi, Y. Iguchi, H. Fujiwara, Y. Morita, K. Nii, H. Kawaguchi, and M. Yoshimoto, "A 10T Non-Precharge Two-Port SRAM for 74% Power Reduction in Video Processing," Proceedings of IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp.107-112, May. 2007.

[25] H. Noguchi, S. Okumura, Y. Iguchi, H. Fujiwara, Y. Morita, K. Nii, H. Kawaguchi, and M. Yoshimoto, "Which is the Best Dual-Port SRAM in 45-nm Process Technology? – 8T, 10T Single End, and 10T Differential –," Proceedings of IEEE International Conference on IC Design and Technology (ICICDT), pp. 55-58, June 2008.

[26] H. Noguchi, Y. Iguchi, H. Fujiwara, S. Okumura, Y. Morita, K. Nii, H. Kawaguchi, and M. Yoshimoto, "A 10T Non-Precharge Two-Port SRAM Reducing Readout Power for Video Processing," IEICE Transactions on Electronics, vol. E91-C, no. 4, pp. 543-552, April 2008.

[27] N. Shibata, H. Kiya, S. Kurita, H. Okamoto, M. Tan'no, and T. Douseki, "A 0.5-V 25-MHz 1-mW 256-kb MTCMOS/SOI SRAM for Solar-Power-Operated Portable Personal Digital Equipment - Sure Write Operation by Using Step-Down Negatively Overdriven Bitline Scheme," IEEE Journal of Solid-State Circuits, vol. 41, no. 3, pp. 728-742, March 2006.

[28] N. Verma and A. P. Chandrakasan, "A 256 kb 65 nm 8T Subthreshold SRAM Employing Sense-Amplifier Redundancy," IEEE Journal of Solid-State Circuits, vol.43, no.1, pp.141-149, January 2008.

[29] R.E. Aly, M.A. Bayoumi, and M. Elgamel, "Dual Sense Amplified Bit Lines (DSABL) Architecture for Low-Power SRAM Design," Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), vol. 2, pp. 1650-1653, May 2005.

[30] S. Ohbayashi, M. Yabuuchi, K. Nii, Y. Tsukamoto, S. Imaoka, Y. Oda, T. Yoshihara, M. Igarashi, M. Takeuchi, H. Kawashima, Y. Yamaguchi, K. Tsukamoto, M. Inuishi, H. Makino, K. Ishibashi, and H. Shinohara, "A 65-nm SoC Embedded 6T-SRAM Designed for Manufacturability With Read and Write Operation Stabilizing Circuits," IEEE Journal of Solid-State Circuits, vol. 42, no. 4, pp. 820-829, April 2007.

[31] C. Qikai, H. Mahmoodi, S. Bhunia, and K. Roy, "Modeling and Testing of SRAM for New Failure Mechanisms due to Process Variations in Nanoscale CMOS," Proceedings of 23rd IEEE VLSI Test Symposium, pp. 292-297, May 2005.

[32] R. Arunachalam, E. Acar, and S.r. Nassif, "Optimal shilding/spacing metrics for low power design," Proceedings of IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp.167-172, February 2003.

[33] Chung-Yu, Wing-Chuen Shiau, "Delay Models and Speed Improvement Techniques for RC Tree Interconnections Among Small-Geometry CMOS Inverters," IEEE Journal of Solid-state Circuits, vol. 25, no. 5, pp.1247-1256, October 1990.

[34] W. C. Elmore, "The Transient Response of Damped Linear Networks with Paticular Regard to Wideband Amplifiers," Journal of Applied Physics, vol. 19, pp. 55-63, January 1948.

[35] 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 Transactions on Very Large Scale Integration (VLSI) Systems, vol. 16, no. 6, pp. 620-627, June 2008.

[36] H. Pilo, J. Barwin, G. Braceras, C. Browning, S. Burns, J. Gabric, S. Lamphier, and M. Miller, "An SRAM Design in 65 nm and 45 nm Technology Nodes Featuring Read and Write-Assist Circuits to Expand Operating Voltage," 2006 Symposium on VLSI Circuits Digest of Technical Papers, pp. 15-16, June 2006.

[37] N. Verma, A. P. Chandrakasan, "A 65nm 8T Sub-V_t SRAM Employing Sense-Amplifier Redundancy," International Solid-State Circuits Conference (ISSCC) 2007 Digest of Technical Paper, pp. 328-329, Febrary 2007.

[38] T. H. Kim, J. Liu, J. Keane, C. H. Kim, "A High-Density Subthreshold SRAM with Data-Independent Bitline Leakage and Virtural Ground Replica Scheme," International Solid-State Circuits Conference (ISSCC) 2007 Digest of Technical Papers, pp. 330-331, February 2007.

[39] I. J. Chang, J. J. Kim, S. P. Park, and K. Roy, "A 32kb 10T Subthreshold SRAM Array with Bit-Interleaving and Differential Read Scheme in 90nm CMOS," International Solid-State Circuits Conference (ISSCC) 2008 Digest of Technical Papers, pp. 398-300, February 2008.

[40] M. Yamaoka, N. Maeda, Y. Shinozaki, Y. Shimazaki, K. Nii, S. Shimada, K. Yanagisawa, And T. Kawahara, "90-nm process-variation adaptive embedded SRAM modules with power-line-floating write technique," IEEE Journal of Solid-State Circuits, vol. 41, no. 3, pp. 705-711, March 2006.

[41] Y. Morita, H. Fujiwara, H. Noguchi , K. Kawakami, J. Miyakoshi, S. Mikami, K. Nii, H. Kawaguchi, and M. Yoshimoto, "A V_{th}-Variation-Tolerant SRAM with 0.3-V Minimum Operation Voltage for Memory-Rich SoC under DVS Environment," 2006 Symposium on VLSI Circuits Digest of Technical Papers, pp. 16-17, June 2006.

[42] H. Fujiwara, S. Okumura, Y. Iguchi, H. Noguchi, Y. Morita, H. Kawaguchi, and M. Yoshimoto, "Quality of a Bit (QoB): A New Concept in Dependable SRAM," Proceedings of IEEE International Symposium on Quality Electronic Design (ISQED), pp. 98–102, March 2008.

[43] H. Fujiwara, S. Okumura, Y. Iguchi, H. Noguchi, H. Kawaguchi, and M. Yoshimoto, "A 7T/14T Dependable SRAM and Its Array Structure to avoid Half Selection," Proceedings of IEEE 22nd International Conference on VLSI Design, pp. 295-300, January 2009.

[44] H. Fujiwara, S. Okumura, Y. Iguchi, H. Noguchi, H. Kawaguchi, and M. Yoshimoto, "A Dependable SRAM with 7T/14T Memory Cells," IEICE Transactions on Electronics, vol. E92-C, no. 4, pp. 423-432, April 2009.

[45] Y. Nakata, S. Okumura, H. Kawaguchi, and M. Yoshimoto, "0.5-V Operation Variation-Aware Word-Enhancing Cache Architecture Using 7T/14T Hybrid SRAM," Proceedings of IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), pp. 219-224, August 2010.

[46] S. Okumura, S. Yoshimoto, K. Yamaguchi, Y. Nakata, H. Kawaguchi, and M. Yoshimoto, "7T SRAM Enabling Low-Energy Simultaneous Clock Copy," Proceedings of IEEE Custom Integrated Circuits Conference (CICC), pp. 13-16, September 2010.

[47] S. Yoshizawa, N. Wada, N. Hayasaka and Y. Miyanaga, "Scalable Architecture for Word HMM-Based Speech Recognition and VLSI Implementation in Complete System," IEEE Transaction on Circuits and Systems I, vol. 53, no. 1, pp. 70-77, January 2006.

[48] K. Miura, H. Noguchi, H. Kawaguchi, and M. Yoshimoto, "A Low Memory Bandwidth Gaussian Mixture Model (GMM) Processor for 20,000-Word Real-Time Speech Recognition FPGA System," Proceedings of IEEE International Conference on Field-Programmable Technology (FPT), pp. 341-344, December 2008.

[49] T. Fujinaga, K. Miura, H. Noguchi, H. Kawaguchi, and M. Yoshimoto, "Parallelized Viterbi Processor for 5,000-Word Large-Vocabulary Real-Time Continuous Speech Recognition FPGA System," Proceedings of ISCA Annual Conference of International Speech Communication Association (Interspeech), pp. 1483-1486, September 2009.

[50] H. Noguchi, T. Takagi, M. Yoshimoto, and H. Kawaguchi, "An Ultra-Low-Power VAD Hardware Implementation for Intelligent Ubiquitous Sensor Networks," Proceedings of IEEE Workshop on Signal Processing Systems (SiPS), pp. 214-219, October 2009.

[51] T. Takagi, H. Noguchi, K. Kugata, M. Yoshimoto, and H. Kawaguchi, "Microphone Array Network for Ubiquitous Sound Acquisition," Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1474-1477, March 2010.

[52] K. Kugata, T. Takagi, H. Noguchi, M. Yoshimoto, and H. Kawaguchi, "Live Demonstration: Intelligent Ubiquitous Sensor Network for Sound Acquisition," Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1413-1417, May 2010.

[53] H. Noguchi, T. Takagi, K. Kugata, M. Yoshimoto, and H. Kawaguchi: Low-Traffic and Low-Power Data-Intensive Sound Acquisition with Perfect Aggregation Specialized for Microphone Array Networks, Proceedings of International Conference on Sensor Technologies and Applications (SENSORCOMM), pp. 157-162, July 2010.

[54] J. Benesty, M.M. Sondhi, and Y. Huang, "Handbook of Speech Processing," Springer, 2007.

[55] R. Venkatesha Prasad, Abhijeet Sangwan, H.S. Jamadagni, Chiranth M.C, Rahul Sah, and Vishal Gaurav, "Comparison of Voice Activity Detection Algorithms for VoIP," Proceedings of the Seventh International Symposium on Computers and Communications (ISCC), p. 530, July 2002.

[56] H. Sakai, T. Cincarek, H. Kawanami, H. Saruwatari, K. Shikano, and A. Lee, "Voice activity detection applied to hands-free spoken dialogue robot based on decoding using acoustic and language model," Proceedings of ACM International Conference on Robot Communication and Coordination (ROBOCOMM), pp. 303-310, October 2007.

[57] T. Kobayashi, S. Itahashi, S. Hayamizu, and T. Takezawa, "ASJ Continuous Speech Corpus for Research," Journal of Acoustical Society of Japan, vol. 48, no. 12, pp. 888-893, 1992, in Japanese.

[58] T.F. Abdelzaher, T. He, and J.A. Stankovic, "Feedback Control of Data Aggregation in Sensor Networks," Proceedings of IEEE Conference on Decision and Control (CDC), vol. 2, pp. 1490-1495, December 2004.

[59] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, "Impact of Density on Data Aggregation in Wireless Sensor Networks," Proceedings of 22nd International Conference on Distributed Computing Systems (ICDCS), pp. 457-458, November 2001.

[60] A. Wang, W.B. Heinzelman, A. Sinha, and A.P. Chandrakasan, "Energy-Scalable Protocols for Battery-Operated MicroSensor Networks," Journal of VLSI Signal Processing systems, vol. 29, no. 3, pp. 223–237, November 2001.

[61] J. Zhao, R. Govindan, and D. Estrin, "Computing Aggregates for Monitoring Wireless Sensor Networks," Proceedings of IEEE International Workshop on Sensor Network Protocols and Applications (SNPA), pp. 139-148, May 2003.

[62] F. Asano, H. Asoh, and T. Matsui, "Sound Source Localization and Signal Separation for Office Robot (Jijo-2)," Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 1999), pp. 243-248, August 1999.

[63] H. Tanaka and T. Kobayashi, "Estimating Positions of Multiple Adjacent Speakers Based on MUSIC Spectra Correlation using a Microphone Array," Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 5, pp. 3045-3048, May 2001.

[64] K. Nakadai, H. Nakajima, M. Murase, S. Kaijiri, K. Yamada, T. Nakamura, Y. Hasegawa, H.G. Okuno, and H. Tsujino, "Robust Tracking of Multiple Sound Sources by Spatial Integration of Room and Robot Microphone Arrays," Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 4, pp. 929-932, May 2006.

[65] J. Elson, L. Girod, and D. Estrin, "Fine-Grained Network Time Synchronization using Reference Broadcasts," Proceedings of 5th ACM SIGOPS Symposium on Operating Systems Design and Implementation (OSDI'02), pp. 147-163, December 2002.

[66] S. Ganeriwal, R. Kumar, and M.B. Srivastava, "Timing-Sync Protocol for Sensor Networks," Proceedings of 1st ACM Conference on Embedded Networked Sensor Systems (SenSys'03), pp. 138-149, November 2003.

[67] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The Flooding Time Synchronization Protocol," Proceedings of 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys'04), pp. 39-49, November 2004.

List of Publications and Presentations

Publications in journals and transactions

1) H. Noguchi, K. Miura, T. Fujinaga, T. Sugahara, H. Kawaguchi, and M. Yoshimoto, "VLSI Architecture of GMM Processing and Viterbi Decoder for 60,000-Word Real-Time Continuous Speech Recognition," IEICE Transactions on Electronics, vol. E94-C, no. 4, April 2011. (In press)
2) H. Noguchi, T. Takagi, K. Kugata, S. Izumi, M. Yoshimoto, and H. Kawaguchi, "Data-Intensive Sound Acquisition System with Large-Scale Microphone Array," Journal of Information Processing Society of Japan (IPSJ), vol. 19, March 2011. (In press)
3) H. Noguchi, Y. Iguchi, H. Fujiwara, S. Okumura, K. Nii, H. Kawaguchi, and M. Yoshimoto, "Design Choice in 45-nm Dual-Port SRAM – 8T, 10T Single End, and 10T Differential –," IPSJ Transactions on System LSI Design Methodology, vol. 4, pp. 80-90, February 2011.
4) H. Fujiwara, S. Okumura, Y. Iguchi, H. Noguchi, H. Kawaguchi, and M. Yoshimoto, "A Dependable SRAM with 7T/14T Memory Cells," IEICE Transactions on Electronics, vol. E92-C, no. 4, pp. 423-432, April 2009.
5) 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 Transactions on Very Large Scale Integration (VLSI) Systems, vol. 16, no. 6, pp. 620-627, June 2008.
6) H. Noguchi, S. Okumura, Y. Iguchi, H. Fujiwara, Y. Morita, K. Nii, H. Kawaguchi, and M. Yoshimoto, "A 10T Non-Precharge Two-Port SRAM Reducing Readout Power for Video Processing," IEICE Transactions on Electronics, vol. E91-C, no. 4, pp. 543-552, April 2008.
7) Y. Morita, H. Fujiwara, H. Noguchi, Y. Iguchi, K. Nii, H. Kawaguchi, and M. Yoshimoto, "Area Comparison between 6T and 8T SRAM cells in Dual-V_{dd} Scheme and DVS scheme," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol.E90-A, no. 12, December 2007.

8) Y. Morita, H. Fujiwara, H. Noguchi, Y. Iguchi, K. Nii, H. Kawaguchi, and M. Yoshimoto, "Area Optimization in 6T and 8T SRAM Cells Considering V_{th} Variation in Future Processes," IEICE Transactions on Electronics, vol. E90-C, no. 10, pp. 1949-1956, October 2007.
9) Y. Morita, H. Fujiwara, H. Noguchi, K. Kawakami, J Miyakoshi, S. Mikami, K. Nii, H. Kawaguchi, and M. Yoshimoto, "A 0.3-V Operating, V_{th}-Variation-Tolerant SRAM under DVS Environment for Memory-Rich SoC in 90-nm Technology Era and Beyond," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E89-A, no. 12, pp. 3634-3641, December 2006.

Presentations in international conferences

1) H. Noguchi, S. Okumura, T. Takagi, K. Kugata, M. Yoshimoto and H. Kawaguchi, "0.45-V Operating V_t-Variation Tolerant 9T/18T Dual-Port SRAM," Proceedings of IEEE International Symposium on Quality Electronic Design (ISQED), March 2011.
2) H. Noguchi, J. Tani, Y. Shimai, M. Nishino, S. Izumi, H. Kawaguchi, and M. Yoshimoto, "A 34.7-mW Quad-Core MIQP Solver Processor for Robot Control," Proceedings of IEEE Custom Integrated Circuits Conference (CICC), pp. 513-516, September 2010.
3) K. Mizuno, H. Noguchi, G. He, Y. Terachi, T. Kamino, H. Kawaguchi, and M. Yoshimoto, "Fast and Low-Memory-Bandwidth Architecture of SIFT Descriptor Generation with Scalability on Speed and Accuracy for VGA Video," Proceedings of 20th International Conference on Field Programmable Logic and Applications (FPL), pp. 608-611, August 2010.
4) H. Noguchi, T. Takagi, K. Kugata, M. Yoshimoto, and H. Kawaguchi, "A Low-Traffic and Low-Power Data-Intensive Sound Acquisition System with Perfect Aggregation Scheme Specialized for Microphone Array Network," Proceedings of International Conference on Sensor Technologies and Applications (SENSORCOMM), pp. 157-162, July 2010.
5) H. Noguchi, J. Tani, Y. Shimai, H. Kawaguchi, and M. Yoshimoto, "Parallel-Processing VLSI Architecture for Mixed Integer Linear Programming,"

Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2362-2365, May 2010.

6) K. Kugata, T. Takagi, H. Noguchi, M. Yoshimoto, and H. Kawaguchi, "Intelligent Ubiquitous Sensor Network for Sound Acquisition," Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1413-1417, May 2010.
7) T. Takagi, H. Noguchi, K. Kugata, M. Yoshimoto, and H. Kawaguchi, "Microphone Array Network for Ubiquitous Sound Acquisition," Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1474-1477, March 2010.
8) Y. Shimai, J. Tani, H. Noguchi, H. Kawaguchi, and M. Yoshimoto, "FPGA Implementation of Mixed Integer Quadratic Programming Solver for Mobile Robot Control," Proceedings of IEEE International Conference on Field-Programmable Technology (FPT), pp. 447-450, December 2009.
9) H. Noguchi, T. Takagi, M. Yoshimoto, and H. Kawaguchi, "An Ultra-Low-Power VAD Hardware Implementation for Intelligent Ubiquitous Sensor Networks," Proceedings of IEEE Workshop on Signal Processing Systems (SiPS), pp. 214-219, October 2009.
10) T. Fujinaga, K. Miura, H. Noguchi, H. Kawaguchi, and M. Yoshimoto, "Parallelized Viterbi Processor for 5,000-Word Large-Vocabulary Real-Time Continuous Speech Recognition FPGA System," Proceedings of ISCA Annual Conference of International Speech Communication Association (Interspeech), pp.1483-1486, September 2009.
11) S. Okumura, Y. Iguchi, S. Yoshimoto, H. Fujiwara, H. Noguchi, K. Nii, H. Kawaguchi, and M. Yoshimoto, "A 0.56-V 128kb 10T SRAM Using Column Line Assist (CLA) Scheme," Proceedings of IEEE International Symposium on Quality Electronic Design (ISQED), pp. 659-663, March 2009.
12) H. Fujiwara, S. Okumura, Y. Iguchi, H. Noguchi, H. Kawaguchi, and M. Yoshimoto, "A 7T/14T Dependable SRAM and its Array Structure to Avoid Half Selection," Proceedings of IEEE International Conference on VLSI Design, pp. 295-300, January 2009.
13) K. Miura, H. Noguchi, H. Kawaguchi, and M. Yoshimoto, "A Low Memory

Bandwidth Gaussian Mixture Model (GMM) Processor for 20,000-Word Real-Time Speech Recognition FPGA System," Proceedings of IEEE International Conference on Field-Programmable Technology (FPT), pp. 341-344, December 2008.

14) H. Noguchi, S. Okumura, Y. Iguchi, H. Fujiwara, Y. Morita, K. Nii, H. Kawaguchi, and M. Yoshimoto, "Which is the Best Dual-Port SRAM in 45-nm Process Technology? - 8T, 10T Single End, and 10T Differential -," Proceedings of International Conference on IC Design and Technology (ICICDT), pp.55-58, June 2008.
15) H. Fujiwara, S. Okumura, Y. Iguchi, H. Noguchi, Y. Morita, H. Kawaguchi, and M. Yoshimoto, "Quality of a Bit (QoB): A New Concept in Dependable SRAM," Proceedings of IEEE 9th International Symposium on Quality Electronic Design (ISQED), pp. 98-102, March 2008.
16) Y. Morita, H. Fujiwara, H. Noguchi, Y. Iguchi, K. Nii, H. Kawaguchi, and M. Yoshimoto, "An Area-Conscious Low-Voltage-Oriented 8T-SRAM Design under DVS Environment," 2007 Symposium on VLSI Circuits Digest of Technical Papers, pp.256-257, June 2007.
17) H. Noguchi, Y. Iguchi, H. Fujiwara, Y. Morita, K. Nii, H. Kawaguchi, and M. Yoshimoto, "A 10T Non-Precharge Two-Port SRAM for 74% Power Reduction in Video Processing," Proceedings of IEEE Computer Society Annual Symposium on VLSI 2007 (ISVLSI), pp.107-112, May 2007.
18) Y. Morita, H. Fujiwara, H. Noguchi, K. Kawakami, J Miyakoshi, S. Mikami, K. Nii, H. Kawaguchi, and M. Yoshimoto, "A V_{th}-Variation-Tolerant SRAM with 0.3-V Minimum Operation Voltage for Memory-Rich SoC under DVS Environment," 2006 Symposium on VLSI Circuits Digest of Technical Papers, pp.16-17, June 2006.

Presentations in domestic conferences

1) 和泉慎太郎, 野口紘希, 高木智也, 久賀田耕史, 祖田心平, 吉本雅彦, 川口博, "ネットワーク型マイクロホンアレイ間のデータ集約による音声信号ビームフォーミング," 電子情報通信学会技術研究報告, vol. 110, no. 216,

ICD2010-81, pp. 95-100, 2010 年 10 月.

2) 祖田心平, 久賀田耕史, 高木智也, 和泉慎太郎, 野口紘希, 吉本雅彦, 川口博, “分散処理を用いた超低消費電力ネットワーク型マイクロホンアレーの研究,” 日本音響学会 2010 年秋季研究発表会, 2-10-9, pp. 585-588, 2010 年 9 月.
3) 久賀田耕史, 野口紘希, 高木智也, 祖田心平, 吉本雅彦, 川口博, “ネットワーク分散処理を用いた超低消費電力音声信号処理プロセッサ,” STARC フォーラム/シンポジウム, 2010 年 8 月.
4) 嶋井優介, 谷純一, 野口紘希, 川口博, 吉本雅彦, “知能ロボットのためのマルチコア MIQP ソルバープロセッサの FPGA 実装,” LSI とシステムのワークショップ, pp. 176-178, 2010 年 5 月.
5) 藤永剛史, 三浦和夫, 野口紘希, 川口博, 吉本雅彦, “大語彙連続音声認識のための並列 Viterbi プロセッサアーキテクチャ,” LSI とシステムのワークショップ, pp. 182-184, 2010 年 5 月.
6) 高木智也, 野口紘希, 久賀田耕史, 吉本雅彦, 川口博, “分散処理型ユビキタスセンサネットワークのための超低消費電力音声処理プロセッサ,” LSI とシステムのワークショップ, pp. 179-181, 2010 年 5 月. (IEEE SSCS Kansai Chapter Academic Research Award 受賞)
7) 谷純一, 野口紘希, 嶋井優介, 川口博, 吉本雅彦, “全整数計画問題のソルバーの FPGA 実装,” LSI とシステムのワークショップ, pp. 241-243, 2009 年 5 月.
8) 吉本秀輔, 井口友輔, 奥村俊介, 藤原英弘, 野口紘希, 新居浩二, 川口博, 吉本雅彦, “カラム線制御回路を用いた 0.56V 動作 128-kb10T 小面積 SRAM,” LSI とシステムのワークショップ, pp. 226-228, 2009 年 5 月.
9) 高木智也, 野口紘希, 吉本雅彦, 川口博, “マイクロホンアレイ・センサネットワークによるインテリジェント・ユビキタス音声処理システムと, その低消費電力 LSI の提案,” LSI とシステムのワークショップ, pp. 235-237, 2009 年 5 月.
10) 三浦和夫, 野口紘希, 藤永剛史, 川口博, 吉本雅彦, “リアルタイム 20,000

語彙連続音声認識のための GMM プロセッサの FPGA 実装," LSI とシステムのワークショップ, pp.244-246, 2009 年 5 月.

11) 奥村俊介, 藤原英弘, 井口友輔, 野口紘希, 川口博, 吉本雅彦, "高信頼性モードを有する 7T/14T ディペンダブル SRAM," LSI とシステムのワークショップ, pp. 229-231, 2009 年 5 月.
12) 藤原英弘, 奥村俊介, 井口友輔, 野口紘希, 川口博, 吉本雅彦, "7T/14T ディペンダブル SRAM およびそのセル配置構造," 電子情報通信学会総合大会, p. 92, 2009 年 3 月.
13) 谷純一, 野口紘希, 川口博, 吉本雅彦, "全整数計画問題ソルバーの FPGA 実装," 情報処理学会関西支部大会, D-05, pp. 233-234, 2008 年 10 月.
14) 野口紘希, 川口博, "発話推定を用いたインテリジェント認識システムの低消費電力化技術," STARC フォーラム/シンポジウム, 2008 年 7 月. (優秀ポスター賞受賞)
15) 奥村俊介, 藤原英弘, 井口友輔, 野口紘希, 森田泰弘, 川口博, 吉本雅彦, "高信頼性モードと高速アクセスモードを有するディペンダブル SRAM," システム LSI 設計技術 (SLDM) 研究報告, 2008-SLDM-135, vol. 2008, no. 38, pp.67-72, 2008 年 5 月.
16) 森田泰弘, 藤原英弘, 野口紘希, 井口友輔, 新居浩二, 川口博, 吉本雅彦, "DVS 環境下での小面積・低電圧動作 8T SRAM の設計-32nm 世代以降で 8T セルが小面積・低電圧動作を同時に実現-," 第 11 回システム LSI ワークショップ, pp.222-224, 2007 年 11 月. (IEEE システム LSI 技術賞受賞)
17) 井口友輔, 野口紘希, 奥村俊介, 藤原英弘, 森田泰弘, 新居浩二, 川口博, 吉本雅彦, "ビット線電力を削減する, 動画像処理応用 10T 非プリチャージ 2-port SRAM," VDEC デザイナーフォーラム, 2007 年 9 月. (IEEE SSCS Japan Chapter Outstanding Design Award 受賞)
18) 森田泰弘, 藤原英弘, 野口紘希, 井口友輔, 新居浩二, 川口博, 吉本雅彦, "DVS 環境下での小面積・低電圧動作 8T SRAM の設計," 電子情報通信学会技術研究報告, ICD2007-95, vol. 107, no. 195, pp. 139-144, 2007 年 8 月.
19) 奥村俊介, 野口紘希, 井口友輔, 藤原英弘, 森田泰弘, 新居浩二, 川口博,

吉本雅彦,“ビット線電力を74%削減する動画像処理応用 10T 非プリチャージ 2-port SRAM の設計,” 電子情報通信学会技術研究報告, ICD2007-53, vol.107, no.163, pp.95-100, 2007 年 7 月.

20) 藤原英弘, 新居浩二, 野口紘希, 宮越純一, 村地勇一郎, 森田泰弘, 川口博, 吉本雅彦, “ビット線の電力を削減する実時間動画像処理応用 2-port SRAM,” 電子情報通信学会技術研究報告, ICD2007-7, vol. 107, no. 1, pp. 35-40, 2007 年 4 月.

21) 井口友輔, 野口紘希, 藤原英弘, 森田泰弘, 新居浩二, 川口博, 吉本雅彦, “ビット線電力を 8 割削減する動画像処理応用 10T 非プリチャージ 2-port SRAM,” 2007 年電子情報通信学会総合大会, A-3-11, p. 101, 2007 年 3 月.

22) 野口紘希, 森田泰弘, 藤原英弘, 新居浩二, 川口博, 吉本雅彦, “しきい値電圧ばらつきを克服した DVS 環境下における 0.3V 動作 SRAM の開発,” 第 10 回システム LSI ワークショップ, pp. 219-222, 2006 年 11 月. (最優秀ポスター賞受賞)

23) 藤原英弘, 新居浩二, 野口紘希, 宮越純一, 村地勇一郎, 森田泰弘, 川口博, 吉本雅彦, “ビット線充放電電力を 53%削減する動画像処理応用 2-port SRAM,” 2006 電子情報通信学会ソサイエティ大会講演論文集, C-12-42, p. 103, 2007 年 9 月.

24) 野口紘希, 森田泰弘, 藤原英弘, 川上健太郎, 宮越純一, 三上真司, 新居浩二, 川口博, 吉本雅彦, “しきい値電圧ばらつきを克服した DVS 環境下における 0.3V 動作 SRAM の開発,” 電子情報通信学会技術研究報告, ICD2006-106, vol. 106, no. 206, pp. 155-160, 2006 年 8 月.

25) 森田泰弘, 藤原英弘, 野口紘希, 川上健太郎, 宮越純一, 三上真司, 新居浩二, 川口博, 吉本雅彦, “動的電圧制御環境下における 0.3-V 動作 64-kb SRAM,” 2006 年電子情報通信学会総合大会講演論文集, AS-2-2, pp. S-17-S-18, 2006 年 3 月.

Acknowledgments

I would like to express my gratitude to Professor Masahiko Yoshimoto of Kobe University for providing me the great opportunity to study in his laboratory and for providing appropriate guidance and valuable advice for this research. I am grateful as well to Associate Professor Hiroshi Kawaguchi of Kobe University, who gave me fruitful advice related to this research. I also would like to thank Professor Makoto Nagata and Professor Masahiro Numa for giving me valuable guidance in revising this dissertation. I also extend my sincere appreciation to Professor Ariki Yasuo, Professor Zhiwei Luo, Professor Masahiko Tsukamoto and Associate Professor Chikara Ohta for their expert advice on innumerable occasions.

We are grateful for helpful suggestions related to the various issues with Dr. Masunori Sugimoto, Dr. Hiroshi Nakayama, Dr. Kazumasa Suzuki, and Dr. Koki Okada, in Chapters 4 and 6.

I am exceedingly grateful to Dr. Koji Nii, Dr. Yasuhiro Morita, Dr. Hidehiro Fujiwara, Mr. Yusuke Iguchi, Mr. Shunsuke Okumura, for providing me valuable discussions about low-power and high-dependable SRAM designs, particularly related to Chapters 3 and 4.

I have enormous appreciation for useful advice and active discussions related to Chapters 5 and 6 from my colleagues of the hyper-project. I would like to thank the speech recognition team: Mr. Kazuo Miura, Mr. Tsuyoshi Fujinaga, Mr. He Guangji, and Mr. Takanobu Sugahara, and Mr. Yuki Miyamoto. My deepest appreciation goes to the ubiquitous team: Mr. Tomoya Takagi, Mr. Koji Kugata, and Mr. Shinpei Soda.

I am indebt to Dr. Junichi Miyakoshi, and Dr. Shintaro Izumi for his support in developing digital LSIs.

I received generous support from the object recognition team: Mr. Kosuke Mizuno, Mr. Yusuke Kurita, Mr. Mitsuhiko Kuroda, and Mr. Yosuke Terachi.

I am deeply grateful to the robotics controlling team: Mr. Junichi Tani, Mr. Yusuke Shimai, and Mr. Masanori Nishino.

Discussions related SRAM design, with Mr. Kousuke Yamaguchi, Mr. Syusuke Yoshimoto, Mr. Masahiro Yoshikawa, Mr. Takurou Amashita, Mr. Yuuki Kagiyama, Mr. Masaharu Terada, and Mr. Koji Yanagida, have been insightful.

I appreciate the feedback offered related to dynamic voltage and frequency scaling (DVFS) techniques and high-efficient computer architectures by Dr. Kentaro Kawakami, Mr. Jun Takemura, Mr. Yoshinori Sakata, Mr. Yohei Nakata, Mr. Yukihiro Takeuchi, Mr. Yusuke Takeuchi, and Mr. Jung Jin-Wook.

I have had the support and encouragement of the motion-estimation team and optical-flow team: Dr. Yuichiro Murachi, Mr. Yuki Fukuyama, Mr. Ryo Yamamoto, Mr. Masaki Hamamoto, Mr. Tomokazu Ishihara, Mr. Takahiro Iinuma, Ms. Yin Fang, Mr. Keiichi Yoshino, Mr. Jangchung Lee, Mr. Koji Takahashi, and Mr. Tetsuya Kamino.

During the rich time I spent in the laboratory, I was fortunate to meet so many research members in the laboratory. With them, I spent a fruitful time. I must acknowledge all the research members who have discussed and supported my research: Dr. Shinji Mikami, Dr. Toshikazu Suzuki, Dr. Hiroaki Suzuki, Dr. Takashi Takeuchi, Dr. Takashi Matsuda, Dr. Tetsuro Matsuno, Dr. Augusto Foronda, Mr. Hirotoku Yoshino, Mr. Takafumi Aonishi, Mr. Masumi Ichien, Mr. Kenichi Nagai, Mr. Yuhi Higuchi, Mr. Akihiro Gion, Mr. Tadayoshi Katagiri, Mr. Koji Hamano, Mr. Yu Otake, Mr. Kenichiro Yagura, Mr. Hyeokjong Lee, Mr. Toshihiro Konishi, Mr. Yasuharu Sakai, Mr. Kou Tsuruda, Mr. Akihisa Oka, and Mr. Keisuke Okuno. I also appreciate CS26 research members: Dr. Mitsuya Fukazawa, Dr. Takushi Hashida, Dr. Yoji Bando, and Mr. Yuki Araga. I cannot thank Ms. Emi Go, Ms. Keiko Matsuoka and Ms. Aya Tsuboi, Ms. Yurie Izumi enough for their kindness. I also thank my English teacher Ms. Mitsu Tsukino for her encouragement for my presentation at the international conferrences.

I also appreciate Professor Kazutoshi Kobayashi and Assistant Professor Akira Tsuchiya with Kyoto VDEC Sub-Center for their support in measurements of the test chips described in Chapter 3.

I must also thank the financial supporters of this research. Particularly, the work for Chapter 3 was supported by Renesas Electronics Corporation. Work associated with Chapters 4 and 6 was supported by the Semiconductor Technology Academic Research Center (STARC). The work described in Chapter 5 was supported by KAKENHI (18200003).

The VLSI test chip in Chapter 3, using 90-nm process technology, was facilitated by the chip fabrication program of VLSI Design and Education Center (VDEC), The

University of Tokyo in collaboration with Advanced SoC Platform (ASPLA) Corp. This dissertation is 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.

I am also grateful to the many researchers who participated in lively discussions with me at international conferences and symposia.

Finally, I appreciate my parents for nurturing me in my development and I thank them for all the sacrifices that they have made on my behalf.

Hiroki Noguchi