Kobe University Repository : Kernel

PDF issue: 2026-10-14

A Study on Process-Variation-Adaptive Design for Robust and High-Performance VLSI Processor

Nakata, Yohei

(Degree)
博士（工学）

(Date of Degree)
2013-03-25

(Date of Publication)
2014-02-03

(Resource Type)
doctoral thesis

(Report Number)
甲5787

(URL)
https://hdl.handle.net/20.500.14094/D1005787

※ 当コンテンツは神戸大学の学術成果です。無断複製・不正使用等を禁じます。著作権法で認められている範囲内で、適切にご利用ください。

Doctoral Dissertation

A Study on Process-Variation-Adaptive Design for Robust and High-Performance VLSI Processor

(プロセスばらつきを考慮した

高信頼・高性能 VLSI プロセッサの設計技術研究)

January 2013

Graduate School of System Informatics

Kobe University

Yohei Nakata

中田 洋平

Abstract

This dissertation reports process-variation-aware robust and high-performance techniques of a very large scale integrated circuit (VLSI) processor design in a scaled semiconductor process technology. More and more electronic equipment and devices have been widely used in ubiquitous computing environments by incorporating a high-performance and high-reliability VLSI System on a Chip (SoC) device that integrates billions of transistors fabricated with advanced semiconductor process technology.

As the background of this research area, the objective of this study and an overview of this dissertation are presented. Then issues related to the VLSI system in the advanced process technology are noted. The main issues are explained as four parts: 1) operating stability degradation caused by degradation of SRAM operating reliability, 2) processing performance degradation in the VLSI with the synchronous clock design, 3) the degradation of scalabilities in the operating stability and the processing performance caused by the process variation, and 4) difficulty in analyzing VLSI system stability. The description of each part emphasizes the objectives of this study.

The third part of this paper describes a cache memory that can operate at low voltage under the effect of the process variation in a scaled process technology. The static random access memory (SRAM) is a vulnerable circuit component in the VLSI processor against process variation. Therefore, a large-capacity SRAM macro determines the minimum operating voltage (V_{min}) of the entire VLSI processor. The cache memory leverages 7T/14T SRAM, which can improve its operating reliability: two pMOS transistors are appended between internal nodes in a pair of the conventional 6T SRAM bitcells. To mitigate the variation of operating stability of the SRAM in the large-capacity SRAM cache macro, 32-bit word-level fine-grain mode control of the 7T/14T SRAM is introduced. The proposed scheme, designated as 7T/14T word-enhancing, also introduces a testing method that improves the efficiency of the 14T word-enhancing scheme. In a 65-nm process technology, the 4-MB cache implemented with the proposed scheme can operate at 0.5 V that is 42% and 21% lower, respectively, than a conventional 6T SRAM and a cache word-disable scheme. As a result of a measurement of the fabricated silicon chip in a 65-nm process, it was

confirmed that the 14T word-enhancing scheme can operate at 0.4 V and reduce V_{min} of the 6T SRAM and 14T dependable modes respectively by 25% and 19%. The respective dynamic power reductions are 89.2% and 73.9%. The respective degrees of 44.8% and 20.9% represent the total power reduction.

In the fourth part of this paper, a network-on-a-chip (NoC) is reported: it can reconfigure its composition considering the process variation. Because NoC generally adopts a synchronous network design across the silicon chip, NoC is strongly affected by process variation, which produces different effects depending on the location in the silicon chip. The operating frequency of the network is degraded while syncing the slowest network component in the silicon chip. A process-variation-adaptive NoC design is proposed to adapt process variation in individual locations of network routers. The proposed NoC introduces a variation-adaptive variable-cycle router (VAVCR) and a variable-cycle pipeline adaptive routing (VCPAR). The proposed VAVCR adaptively configures its processing latency of router pipeline corresponding to the process variation of its location. The operating frequency of the network degraded by the process variation is improved by an adaptive reconfiguration of the proposed VAVCR. The proposed VCPAR is a routing algorithm that can consider processing cycle variation of the NoC with VAVCR. The VCPAR preferentially passes through low-cycle latency routers to minimize the packet transmission latency. The total execution time reduction of the proposed VAVCR with VCPAR is 15.7%, on average, for five task graphs. The proposed scheme can contribute to synchronous network fabrics such as shared bus, ring bus, and crossbar, not limited to NoCs.

The fifth part of this paper describes a new system-level fault-injection scheme that can consider device level behaviors of SRAM. In the robustness evaluation of VLSI processor system under severe operating conditions, consideration of vulnerable SRAM blocks in the VLSI processor is necessary. An SRAM operating stability under severe operating condition is determined by a circuit level behavior and transistor device level variability. In the proposed system-level evaluation environment, the circuit level behavior and the transistor level variability of each individual SRAM are considered. Failures of the SRAM block in the severe operating condition can be injected to the evaluation environment. In the middle of this discussion, details of the modeling of the SRAM circuit behavior are described, along with consideration of the variability of the

transistor device and a fault case generator (FCG) that can generate failure patterns injectable to the system-level evaluation environment. Subsequently, evaluations of the vehicle engine control system are presented. It is confirmed that a dependable processor with 7T/14T dependable SRAM improves system-level dependability compared with the conventional 6T SRAM in the end of this part.

Finally, the conclusion of this study is presented in the last part. In this paper, three techniques governing the process variation are described. The three techniques will be much more valuable in more-scaled CMOS process technology, post-CMOS technology, and other promising future semiconductor technologies that have much more device characteristic variation.

Keywords: VLSI, Process variation, Adaptive circuits, Low voltage, SRAM, Cache memory, Fine-grain control, Network-on-Chip, Routing algorithm, Fault injection, System-level verification, Dependable processor

Table of Contents

Abstract i

Table of Contents v

List of Figures ix

List of Tables xiii

Chapter 1 Introduction 1

1.1 Background of Research Area 1

1.2 Objective of This Study 1

1.3 Overview of This Dissertation 2

Chapter 2 Issues of VLSI System in Advanced Process Technology 5

2.1 Process Variation 5

2.2 Issue of Stability Degradation 8

2.3 Issue of Performance Degradation 9

2.4 Issue of Scalability Degradation 10

2.5 Issue of Stability Analysis of VLSI Processor System 10

2.6 Summary 11

2.7 References 11

Chapter 3 Process-Variation-Aware Cache Architecture Using 7T/14T SRAM 15

3.1 Introduction 15

3.2 Related Work 16

3.3 7T14T SRAM 17

3.3.1 Failures in SRAM 17

3.3.2 7T/14T SRAM 17

3.3.3 Bit Error Rates (BERs) 19

3.4 Implementation of the 14T Word-Enhancing Scheme 22

3.4.1 Conventional word-disable scheme 22

3.4.2 Proposed 14T word-enhancing scheme with a divided control line .. 23

3.4.3 Incremental testing for the 14T word-enhancing scheme 24
3.4.4 Improved BER in the 14T word-enhancing scheme..... 28
3.4.5 Implementation 28
3.4.6 Measurement result..... 29
3.5 Performance, Energy, and Power Comparison 30
3.5.1 Performance evaluation..... 30
3.5.2 Energy and power comparison 32
3.6 Summary 36
3.7 Appendix: Probabilistic BER calculations 36
3.8 References..... 38

Chapter 4 Process-Variation-Adaptive NoC with VAVCR and VCPAR. 41
4.1 Introduction..... 41
4.2 Background..... 42
4.2.1 Process Variation in NoC..... 42
4.2.2 Impact of Variation in Processor Core..... 44
4.2.3 Impact of Variation in On-Chip Networks 45
4.3 Proposed Process-Variation-Adaptive Variable-Cycle Router and its Proposed Routing Algorithm..... 48
4.3.1 Process-Variation-Adaptive Variable-Cycle Router 48
4.3.2 Variable-Cycle Pipeline Adaptive Routing 51
4.4 Evaluation 52
4.4.1 Evaluation Methodology of Routing Methods 53
4.4.2 Evaluation of the Routing Method 53
4.4.3 Evaluation Methodology of VAVCR with VCPAR 59
4.4.4 Evaluation of VAVCR with VCPAR 60
4.4.5 Area overhead of the VAVCR w/ VCPAR..... 65
4.5 Discussion on the network frequency optimization..... 66
4.6 Summary 68
4.7 References..... 68

Chapter 5 System-Level Fault-Injection Scheme and Evaluation of Dependable Processor..... 71
5.1 Introduction..... 71

5.2 Fault-Injection System 72
5.3 Modeling of Failures in SRAM 73
5.3.1 Failures in SRAM 74
5.3.2 Behavior of SRAM failures on a device level 74
5.3.3 Proposed Fault-Injection Flow for System-Level Verification 76
5.3.4 Modeling Failures for System-Level Fault Injection 77
5.3.5 Fault Case Generator 78
5.3.6 Failure data pattern generation in FCG 79
5.4 7T/14T Dependable SRAM 81
5.4.1 Bit Error Rate (BER) 81
5.5 System-level Evaluation 83
5.5.1 Evaluation Methodology 83
5.5.2 Evaluation Result 85
5.6 Summary 87
5.7 References 88

Chapter 6 Conclusion 91

List of Publications and Presentations 95
Publications in journals and transactions 95
Presentations at international conferences 95
Presentations at domestic conferences 97
Patents 100

Acknowledgements 101

List of Figures

Fig. 1.1 Overview of this thesis. 4
Fig. 1.2 Technical layers of VLSI implementation and design techniques. 4
Fig. 2.1 Category of process variation. 6
Fig. 2.2 Pelgrom plots of different processes. The standard deviation of V_{th} becomes larger as the process technology is scaled down. 6
Fig. 2.3 Major impacts of process variation on VLSI processor design. 7
Fig. 2.4 Operating voltage scaling trend of VLSI processor with process technology scaling down. 8
Fig. 3.1 A 7T/14T bitcell pair. 18
Fig. 3.2 Conventional 6T bitcells. 19
Fig. 3.3 BERs for 32-bit cache: “6T”, “1-bit ECC”, “bit-fix” and ”word-disable” use conventional 6T bitcell schemes;“7T normal” and “14T dependable” use 7T/14T bitcells. 21
Fig. 3.4 BERs: magnifying the area bounded by the dashed line in Fig. 3.3. 21
Fig. 3.5 Comprehensive view of the cache word-disable scheme. 23
Fig. 3.6 Block diagram of a word shifter. 23
Fig. 3.7 7T/14T SRAM bitcell (BC) array with the divided control line (DCL) scheme. 24
Fig. 3.8 BERs: including the 14T word-enhancing scheme with conventional testing. 25
Fig. 3.9 Applying the 14T dependable mode in testing. These examples use eight-word cache lines for simplicity. Only asserted bitlines are shown. (a) Dependable mode is applied uniformly to all word pairs. (b) Conventional testing by which the 14T dependable mode is not applied to all word pairs. (c) Incremental testing, where the 14T dependable mode is applied to all word pairs. 26
Fig. 3.10 Flow chart of incremental testing (this figure shows the case of an eight-word cache line). 27
Fig. 3.11 Bit error rates (BERs): applying 14T word-enhancing with incremental testing. 28

Fig. 3.12 Layout plot of a proposed 4-MB cache implemented with a 65-nm process. 29
Fig. 3.13 Measured BERs of 6T, 14T dependable, and 14T word-enhancing scheme in 512-kb SRAM macro. 30
Fig. 3.14 Normalized IPCs in SPEC CPU2000 and SPLASH2 benchmarks. 32
Fig. 3.15 Dynamic energy and dynamic power in high-voltage operation and low-voltage operation in SPEC2000 and SPLASH2 benchmarks. Each figure is normalized by 6T normal in high voltage operation and sums up the energies and powers on the L1D cache, L1I cache, and L2 cache: (a) normalized total energy in each benchmark, (b) normalized total power in each benchmark. 35
Fig. 4.1 Operating frequency variation in a GALS NoC. The operating frequencies of processor cores (FMAX_Pmn) vary. The network frequency ($F_{network}$) is determined by the minimum operating frequency in routers. 43
Fig. 4.2 Distribution of operating frequencies of 20 FO4 inverters. The dashed line signifies the fitted normal distribution curve. 45
Fig. 4.3 Organization of the router. Parameters are the same as those of Table 4.3. 46
Fig. 4.4 Gantt chart of each pipeline stage of the router. 46
Fig. 4.5 Delay of each pipeline stage: NRC, next routing computation; VA, virtual channel allocation; SA, switch allocation; ST, switch traversal; and LT, link traversal. 47
Fig. 4.6 Timing diagrams of the conventional and proposed router pipelines. Here, (a) and (b) respectively correspond to the worst and best delays in the conventional router pipeline; (c) and (d) respectively correspond to the worst and best delays in the proposed router pipeline. 49
Fig. 4.7 Distributions of the router latencies for 8 × 8 mesh networks: (a) conventional router and the (b) proposed VAVCR. 50
Fig. 4.8 Overview of the proposed VCPAR method. 52
Fig. 4.9 Evaluation results of routing methods (packet size is four flits): uniform random traffic. 54
Fig. 4.10 Evaluation results of routing methods (packet size is four flits):

transpose traffic. 55
Fig. 4.11 Evaluation results of routing methods (packet size is four flits): bit reverse traffic. 55
Fig. 4.12 Evaluation results of routing methods (packet size is four flits): hot spot traffic with one hot spot node (the hot spot percentage is 6%). 56
Fig. 4.13 Evaluation results of routing methods (packet size is four flits): hot spot traffic with four hot spot nodes (the hot spot percentage is 6%). 56
Fig. 4.14 Evaluation results of routing methods (packet size is 16 flits): uniform random traffic. 57
Fig. 4.15 Evaluation results of routing methods (packet size is 16 flits): transpose traffic. 57
Fig. 4.16 Evaluation results of routing methods (packet size is 16 flits): bit reverse traffic. 58
Fig. 4.17 Evaluation results of routing methods (packet size is 16 flits): hot spot traffic with one hot spot node (the hot spot percentage is 6%) 58
Fig. 4.18 Evaluation results of routing methods (packet size is 16 flits): hot spot traffic with four hot spot nodes (the hot spot percentage is 6%). 59
Fig. 4.19 Evaluation results of STG-random: (a) normalized execution time and (b) normalized latency. 61
Fig. 4.20 Evaluation results of STG-robot: (a) normalized execution time and (b) normalized latency. 62
Fig. 4.21 Evaluation results of STG-sparse: (a) normalized execution time and (b) normalized latency. 63
Fig. 4.22 Evaluation results of STG-fpppp: (a) normalized execution time and (b) normalized latency. 64
Fig. 4.23 Evaluation results of TGFF: (a) normalized execution time and (b) normalized latency. 65
Fig. 4.24 (a) reduction of the total execution time versus $F_{network}$ (averaged by 100 variation maps) and (b) increase in the packet latency versus $F_{network}$ (averaged by 100 variation maps). 67
Fig. 5.1 Overview of processor-in-the-loop simulation (PILS) and a simple diagram of a controller LSI composed of a logic block and SRAM block. ... 72

Fig. 5.2 Proposed system-level verification environment has a vehicle engine model and controller (ECU) model in which a micro-controller model is included. Faults are injected through a fault-injectable bus bridge on an SH-2A CPU. 74
Fig. 5.3 Failure pattern examples in SRAM memory cell: read margin failure, write margin failure, and soft error. 75
Fig. 5.4 Proposed fault injection scheme flowing from a device level to a system level. 76
Fig. 5.5 Virtual chip: SRAM failures are randomly distributed across a chip. The data-dependence is also randomized as “0” or “1”..... 77
Fig. 5.6 Large-scale verification using the 10,000 virtual chips. 78
Fig. 5.7 Fault Case Generator. 79
Fig. 5.8 SRAM failure data pattern generation flow. 81
Fig. 5.9 Bit error rates (BERs): (a) read operation and (b) write operation. The respective “6T” and “14T” signify the conventional 6T memory cell and 14T dependable mode in the 7T/14T memory cells. Note that the performance of 7T is the same as 6T..... 82
Fig. 5.10 Abnormal termination rates (%) of engine control ECU system: (a) and (c) 6T SRAM, (b) and (d) 14T dependable mode of 7T/14T SRAM. (c) and (d) are plotted on logarithmic graphs. 85
Fig. 5.11 Comparison of ECU system abnormal termination rates (error rate) and SRAM bit error rates. 87

List of Tables

Table 3.1 Two modes in 7T/14T bitcell 19
Table 3.2 Cache architecture configuration parameters dependent on V_{dd}: Energies of single operation are derived from 65-nm SPICE simulations and CACTI 31
Table 3.3 Architecture configuration parameters independent of V_{dd} 31
Table 3.4 Performance comparison: V_{min}, area, frequency, IPC, and power during low-voltage operation 36
Table 4.1 Parameters used for operating frequency estimation 44
Table 4.2 Operating frequency characteristics 45
Table 4.3 Parameters for router delay estimation 47
Table 4.4 Frequencies and ratios in the frequency bins and router latencies in the proposed VAVCR 50
Table 5.1 Read failure information example of a virtual chip for 128-KB SRAM 80
Table 5.2 Parameters for System-level Evaluation 84
Table 5.3 μ_{ER} and σ_{ER} of approximated error rate functions 87

Chapter 1 Introduction

1.1 Background of Research Area

Recently, more and more electronic equipment and devices are widely used in ubiquitous computing society and various fields. The market keeps growing. New devices/solutions such as smartphones, tablets, intelligent vehicles, energy management system have also emerged and are shipped in large quantities. VLSI processors, which comprise the core of such electronic equipment, devices, and solutions, are therefore also produced in vast amounts. VLSI processors that serve as the core of safety critical systems must also maintain high reliability. The VLSI processors for safety critical systems also must be shipped in large amounts while maintaining high reliability. From different aspects, maintaining high yield is also important to ship in large quantities.

In recent advanced Complementary Metal-Oxide Semiconductor (CMOS) process technology, variations in the characteristics of the MOS transistor device become too great to ignore. The large variation engenders many issues related to VLSI processor design, such as operating stability degradation, processing performance degradation, and randomness of failure location. Keeping high reliability and high yield become more difficult because of the presence of the large process variation. Especially, reliability of the large capacity SRAM block is degraded by the large process variation, because SRAM, which is comprised by small-sized transistors, has larger standard deviation for the threshold voltage than the other block. The processing performance degradation caused by the large process variation degrades the yield of the high-performance VLSI processor. Therefore, schemes that can mitigate reliability degradation in the large SRAM block and processing performance degradation in the high performance VLSI processor are required.

1.2 Objective of This Study

In this research, to mitigate the issues caused by the large process variation, process variation adaptive designs and schemes are introduced. Reliability degradation of the large SRAM block is caused mainly by the random component of the process variation. Handling the randomness of the process variation in the large SRAM block is necessary

to mitigate the degradation appropriately. The objective of the mitigating scheme for the reliability degradation is to handle the random variation. Processing performance degradation of the VLSI processor is caused mainly by the systematic spatial component of the process variation. Dealing with the variation in each location of the VLSI processor component is necessary to mitigate this degradation. The objective of this mitigating scheme for the performance degradation is to deal with the systematic spatial variation. The randomness of failure location caused by the random process variation introduces difficulty in verification of the VLSI processor system level environment. An analytical scheme is required that can analyze the effect of the randomness of failure location to system stability. The objective of the analyzing scheme for this difficulty in verification is to consider the failures caused by the random process variation in the system level verification environment.

1.3 Overview of This Dissertation

An overview of this dissertation is presented in Fig. 1.1 with clear correlation between the issues and solutions. First, the background and objective of this study is described. Relations between technical layers of VLSI implementation and techniques described in Chapter 3, 4, 5 are presented in Fig. 1.2. Issues of the VLSI system in the advanced process technology are noted in Chapter 2. The main issues are summarized as four parts: 1) VLSI operating stability degradation caused by degradation of SRAM operating reliability, 2) processing performance degradation in the VLSI with the synchronous clock design, 3) the degradation of scalabilities in the operating stability and the processing performance caused by the process variation, 4) and difficulty in analyzing VLSI system stability. The description of each part enhances the objective of this study.

For the next three chapters, VLSI processor design techniques mitigating or analyzing the large process variation are demonstrated. Chapter 3 presents a cache memory that can operate at low voltage under the effect of the process variation in an advanced process technology. The cache memory leverages 7T/14T SRAM, which can improve its operating reliability: two pMOS transistors are added between internal nodes in a pair of the conventional 6T SRAM bitcells. Adaptively to mitigate the variability of operating stability of the SRAM in the large capacity SRAM cache macro, 32-bit

word-level fine-grain mode control of the 7T/14T SRAM is introduced. The proposed scheme, named 7T/14T word-enhancing, also introduces a testing method that improves the efficiency of the 14T word-enhancing scheme. The improvements in the minimum operating voltage are confirmed as a result of circuit simulations and measurements of fabricated chip. Power and energy reductions are also shown in the evaluation result part of this chapter.

In Chapter 4, a Network-on-a-Chip (NoC) is presented. It can reconfigure its composition considering the process variation in individual location of network routers. The proposed NoC introduces a variation-adaptive variable-cycle router (VAVCR) and a variable-cycle pipeline adaptive routing (VCPAR). The proposed VAVCR adaptively configures its processing latency of router pipeline corresponding to the process variation of its location. The operating frequency of the network degraded by the process variation is improved by an adaptive reconfiguration of the proposed VAVCR. The proposed VCPAR is a routing algorithm that can consider processing cycle variation of the NoC with VAVCR. The execution time reduction with the proposed VAVCR with VCPAR is summarized in the evaluation section of this chapter.

Chapter 5 describes a new system-level fault-injection scheme that can consider device level behaviors of SRAM. In the robustness evaluation of VLSI processor system under severe operating conditions, consideration of vulnerable SRAM blocks in the VLSI processor is necessary. An SRAM operating stably under the severe operating conditions is determined by circuit level behavior and transistor device level variability. In the proposed system-level evaluation environment, the circuit level behavior and the transistor level variability of each individual SRAM are considered. Failures of the SRAM block in the severe operating condition can be injected to the evaluation environment. Subsequently, evaluations of the vehicle engine control system are presented. It was confirmed that a dependable processor with 7T/14T dependable SRAM improves system-level dependability compared with the conventional 6T SRAM in the end of this chapter.

The conclusions of this study are presented in Chapter 6. The overall contributions are summarized briefly.

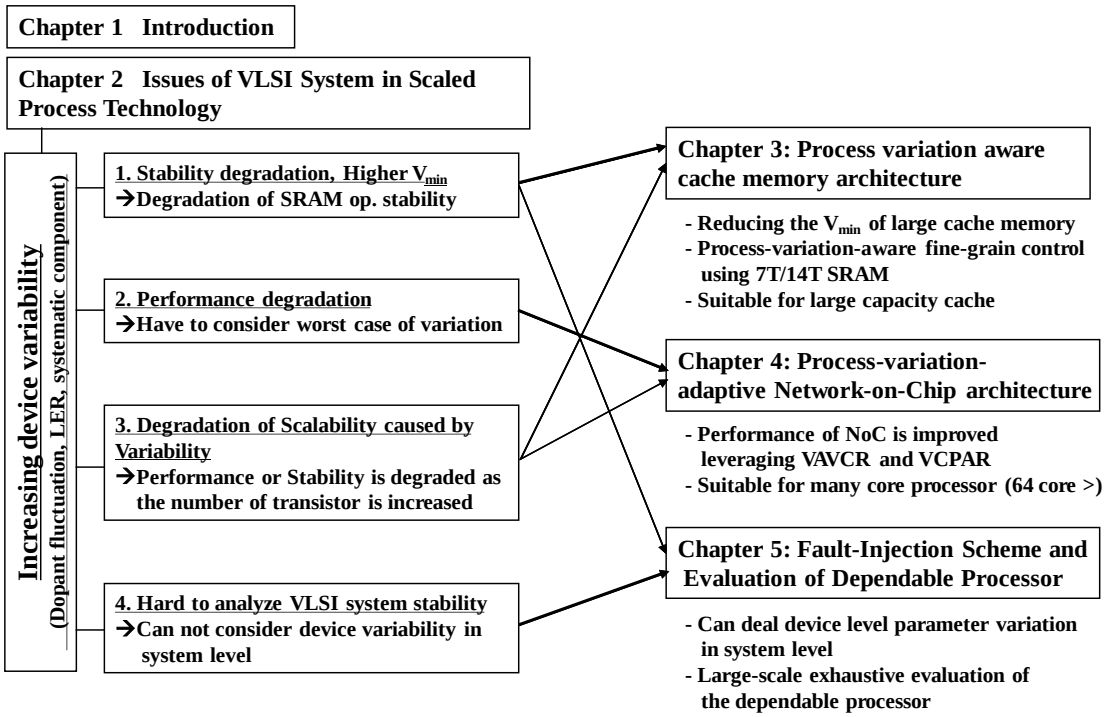


Fig. 1.1 Overview of this thesis.

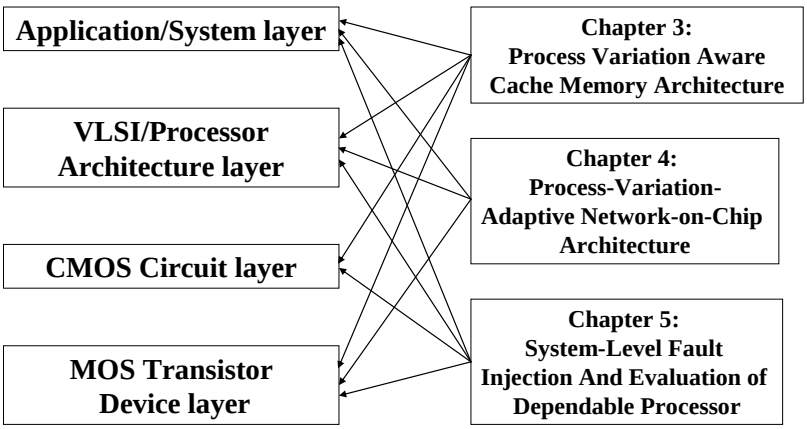


Fig. 1.2 Technical layers of VLSI implementation and design techniques.

Chapter 2 Issues of VLSI System in Advanced Process Technology

The issues of VLSI system in advanced CMOS process technology approached in this dissertation are summarized in this chapter.

First, in Section 2.1, principles and trends of the increasing process variation in an advanced CMOS process technology are described. A description of the process variation in this section enhances comprehension of following sections in this chapter. In Section 2.2, the operating stability degradation caused by degradation of SRAM operating reliability is described. Section 2.3 describes the processing performance degradation in the VLSI with the synchronous clock design. In Section 2.4, the degradations of scalabilities in the operating stability and the processing performance caused by the process variation are described. In Section 2.5, the difficulty in analyzing VLSI system stability is described.

2.1 Process Variation

Fig. 2.1 shows the category of process variation in the CMOS process technology.

Technology scaling increases the threshold-voltage (V_{th}) variation of MOS transistors composed of die-to-die (D2D) and within-die (WID) variations, of which the WID variation is divided into systematic and random variations. Systematic variation results mainly from lens aberration and has a spatial correlation [2.1]. Therefore, neighboring transistors have similar characteristics. In contrast, random variation results mainly from random dopant fluctuation (RDF) and line-edge roughness (LER) [2.2]: random variations show no spatial correlation. For that reason, individual transistors have different characteristics from those of neighboring transistors.

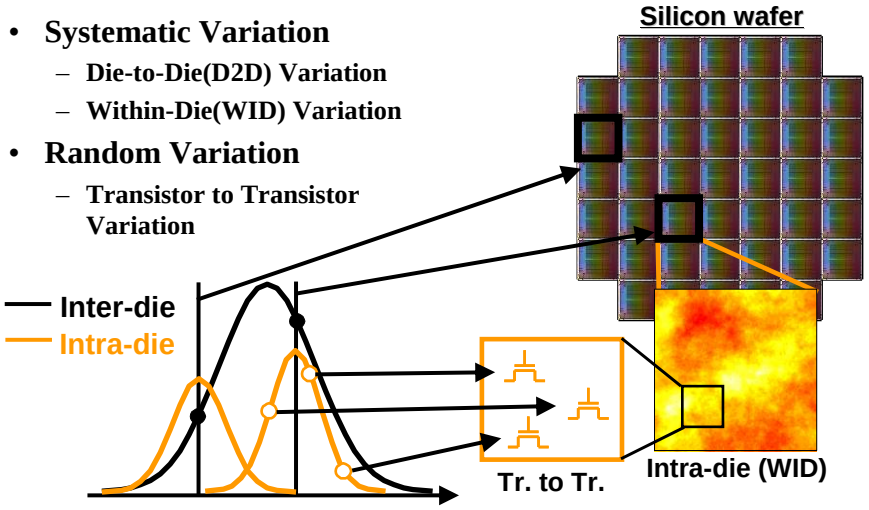


Fig. 2.1 Category of process variation.

As process technology is scaled down, the V_{th} variation of MOS transistors is increased (presented in Fig. 2.2) [2.3] because the channel area ($L_{\text{eff}} \times W_{\text{eff}}$) is shrunk as manufacturing processes advance. Therefore, the negative impacts of the process variation on individual circuits and VLSI processor system are increased in a scaled advanced process technology.

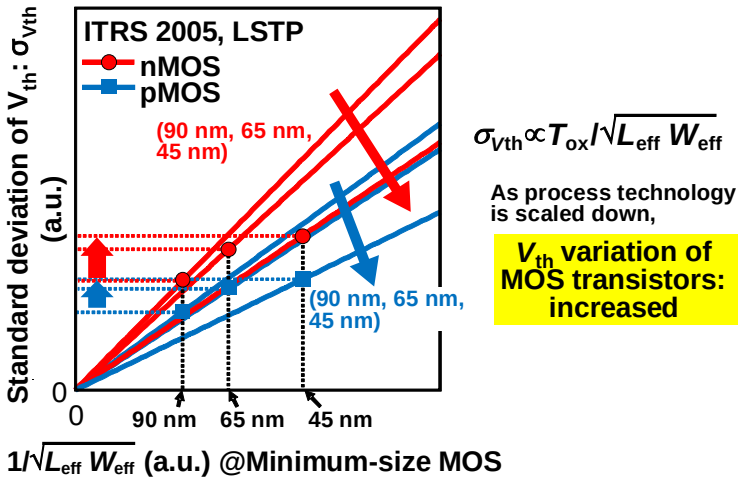


Fig. 2.2 Pelgrom plots of different processes. The standard deviation of V_{th} becomes larger as the process technology is scaled down.

Fig. 2.3 presents major challenges of VLSI processor design attributable to the process variation. The challenges are broadly classifiable into two categories: circuit design level and system design level. Furthermore, the circuit design level challenges are divided into performance degradations, reliability degradation, and difficulty in analog circuit design. Difficulty in failure effect analysis in system level is also shown by the process variation. In this dissertation, operating frequency degradation, marginal fault in memory device, and difficulty in failure effect analysis underlined in Fig. 2.3 will be tackled.

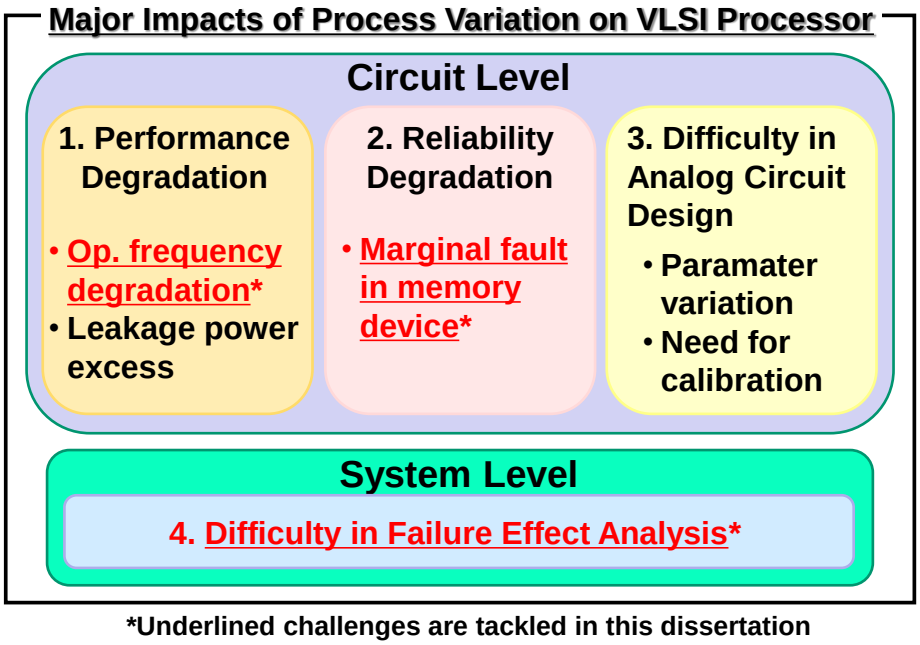


Fig. 2.3 Major impacts of process variation on VLSI processor design.

Systematic WID variations become problematic in large scale SoC with logic circuits, distributed around a chip, which must be synced to a clock. These logic circuits intrinsically have different maximum operating frequencies. The differences of the maximum operating frequency of these logic circuits are additionally increased by the systematic WID variations. In synchronous design, these logic circuits which have different maximum operating frequencies must be synced to the slowest one. Therefore, systematic WID variations degrade the processing performance of the VLSI processor. A detailed description of the issue on performance degradation caused by the systematic

WID variation is presented in Section 2.3.

D2D variation is not particularly considered in this study because it appears as chip-wide offset to the V_{th} and needs only to be treated as well as the conventional corner case aware design.

Random variation does not provide a significant impact on the processing speed of logic circuit because logic circuits generally comprise multiple stages of logic gates that average positive and negative impacts of logic gate of each stage on the processing speed. The random variation significantly degrades the operating margin of SRAM in the VLSI processor. Because of the randomness of the random variation, the deteriorated SRAM cell is distributed randomly in the silicon chip plane. A detailed description of the issue of stability degradation is described in Section 2.2.

2.2 Issue of Stability Degradation

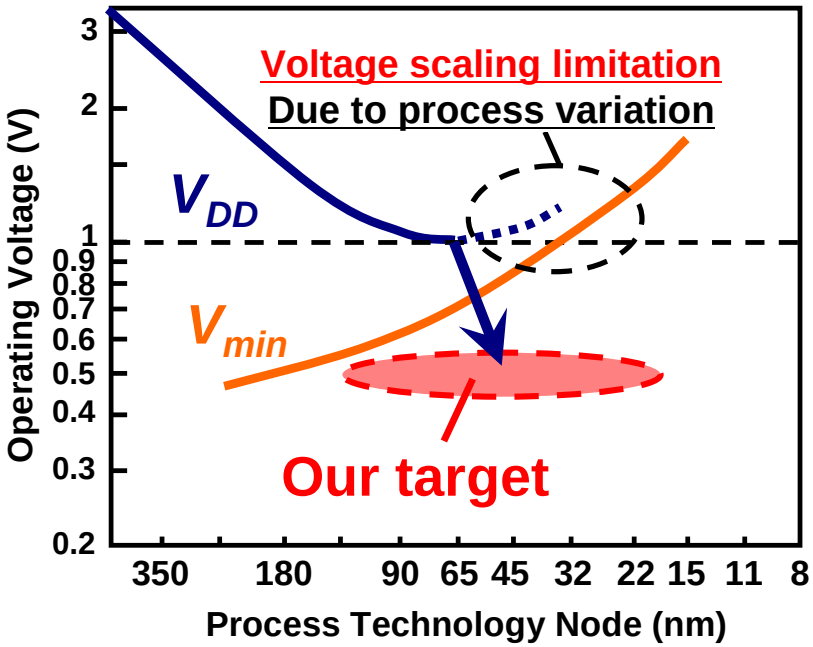


Fig. 2.4 Operating voltage scaling trend of VLSI processor with process technology scaling down.

Fig. 2.4 depicts the operating voltage scaling trend with process technology scaling down [2.4]. The operating voltage (V_{DD}) scaling was continued until the problem of the process variation becomes apparent. The V_{DD} scaling is limited by the rise of the minimum operating voltage (V_{min}) caused by the increasing process variation. The

increase of V_{min} degrades the transistor device reliability because of power supply noise, IR drops, and/or soft errors. Degradation of the transistor device reliability leads to the degradation of the VLSI processor stability. Reduction of the V_{min} is required for acquisition of the adequate VLSI processor stability.

The V_{min} on an entire processor including logic blocks and memory components is determined by the circuit that has the highest value of V_{min} [2.5]. The SRAM has a larger standard deviation of threshold voltage than logic blocks because its transistors are smaller. To make matters worse, the capacity of SRAM bitcells on a processor is huge. Consequently, large SRAM blocks such as L1 data/instruction caches and last level cache (LLC) determine the V_{min} of the processor.

The random V_{th} variation in each SRAM bitcell is distributed randomly throughout the whole SRAM block. Therefore, failures in the whole SRAM block or in the entire VLSI processor are distributed. Coarse-grain control on an SRAM block level basis or a cache way level basis cannot prevent these failures efficiently. Therefore, to reduce V_{min}, fine-grain control that adaptively addresses the V_{th} variations must be applied to the SRAM block.

2.3 Issue of Performance Degradation

The expanded process variation strongly affects the SoC circuit characteristics. As stated in Section 2.1 of this chapter, the systematic WID variation degrades the processing performance of synchronous circuit components considerably. Especially, many-core processors that have many homogeneous components (cores) synced to same clock period are affected strongly by the systematic WID variation.

Network-on-Chip (NoC), which is emerging as a highly efficient network fabric for many-core processors [2.6–2.8], commonly adopts a synchronous design for a network overall across the chip. The NoC in a many-core processor has many network components, each of which is affected by process variation. The network component delays are varied considerably as the number of network components increases. Therefore, the frequency of the large-scale chip-wide synchronous network is degraded to the level of the slowest network component. Many studies have sought means to mitigate the variations of many-core processors using dynamic voltage and frequency scaling (DVFS) [2.9] and application scheduling [2.10], fine-grain body biasing

(FGBB) [2.9], and dynamic voltage frequency-core scaling (DVFCS) [2.11]. However, they did not specifically address variation in a large-scale chip-wide synchronous network.

2.4 Issue of Scalability Degradation

The degradations of operating stability and processing performance caused by the process variation are described in Section 2.2 and 2.3, respectively. These degradations are further worsened if the process variation is larger or the scale of the VLSI processor is larger. The larger scale VLSI processor has larger deviations of operating stability of SRAM and processing performance because it must consider process variation of the larger number of transistors. In fact, the larger the scale of the VLSI processor, the greater the degradations of operating stability and processing performance become. A larger SRAM block operates at higher V_{min} and has lower operating stability. According as the scale of VLSI processor becomes larger or has many more components (cores), the processing performance degradation becomes larger. Finally, the processing performance is saturated. These are scalability degradations.

To prevent scalability degradations, a variation mitigating scheme that can keep its effectiveness in the larger process variation is required.

2.5 Issue of Stability Analysis of VLSI Processor System

Recently, VLSI processors are increasingly becoming key components in various industrial products. Therefore, their reliability is important. However, a transistor is more vulnerable and sensitive to soft errors and negative bias temperature instability (NBTI) because the process technology is scaled down. In addition, increasing variation in the transistor worsens its reliability and VLSI yield. On the VLSI, SRAM comprises the smallest-size transistors, which is therefore the dominant factor determining VLSI's reliability. Accordingly, high reliability is necessary for SRAM on the VLSI processor [2.5, 2.12–2.13].

Many studies and implementations of fault injection into the VLSI have been performed [2.14–2.16]. These studies injected stuck-at faults and transient faults attributable to single event upsets (SEUs) and supply voltage fluctuations. However,

these fault-injection schemes do not consider the physical characteristics of the vulnerable SRAM. In addition, they cannot perform large-scale verification considering the large number of physical VLSIs, each one with different characteristics because of the random process variation.

To analyze operating stability on a VLSI processor exhaustively integrating numerous vulnerable SRAMs, we must consider the impacts of its reliability on the operating stability of a VLSI system.

2.6 Summary

For future robust and high-performance VLSI processor systems in an advanced process technology, the key issues can be summarized as the following four items:

1) VLSI operating stability degradation caused by degradation of SRAM operating reliability.
2) Processing performance degradation in the VLSI with the synchronous clock design.
3) Degradation of scalabilities in the operating stability and processing performance caused by process variation.
4) Difficulty in analyzing VLSI system stability

The following Chapters 3, 4, and 5 respectively focus on issues 1) and 3), issues 2) and 3), and issues 1) and 4), as shown in Fig. 1.1.

2.7 References

[2.1] P. Friedberg, Y. Cao, J. Cain, R. Wang, J. Rabaey, and C. Spanos, "Modeling within-die spatial correlation effects for process-design co-optimization", Proceedings of IEEE Int. Symp. on Quality of Electronic Design, pp. 516-521, 2005.

[2.2] A. Asenov, S. Kaya, A.R. Brown, "Intrinsic parameter fluctuations in decananometer MOSFETs introduced by gate line edge roughness," IEEE Trans. on Electron Devices, vol. 50, No. 5, pp. 1254 - 1260, May 2003.

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

[2.4] K. Itoh, "Adaptive circuits for the 0.5-V nanoscale CMOS era," Int. Solid-State Circuit Conf. Dig. Tech. Papers, pp. 14-20, Feb. 2009.

[2.5] K. Itoh, “Low-voltage scaling limitations for nanoscale CMOS LSIs,” Proceedings of IEEE Int. Conf. on Ultimate Integration of Silicon, pp. 3-6, Mar. 2008.

[2.6] L. Benini and G. De Micheli, "Networks on chips: a new SoC paradigm”, IEEE Computer, vol. 35, no. 1, pp. 70-78, Jan. 2002.

[2.7] W. Dally, and B. Towles, “Principles and Practices of Interconnection Networks”, Morgan Kaufmann, 2004.

[2.8] A. Kumary, P. Kunduz, A.P. Singhx, L.-S. Pehy, and N.K. Jhay, "A 4.6 Tbits/s 3.6 GHz single-cycle NoC router with a novel switch allocator in 65 nm CMOS”, Proceedings of IEEE Int. Conf. on Computer Design, pp. 63-70, 2007.

[2.9] R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas, "Mitigating Parameter Variation with Dynamic Fine-Grain Body Biasing”, Proceedings of ACM/IEEE Int. Symp. on Microarchitecture, pp. 27-42, 2007.

[2.10] R. Teodorescu and J. Torrellas, “Variation-Aware Application Scheduling and Power Management for Chip Multiprocessors”, Proceedings of ACM/IEEE Int. Symp. on Computer Architecture, pp. 363-374, 2008.

[2.11] S. Dighe, S.R. Vangal, P. Aseron, S. Kumar, T. Jacob, K.A. Bowman, J. Howard, J. Tschanz, V. Erraguntla, N. Borkar, V.K. De, and S. Borkar, "Within-Die Variation-Aware Dynamic-Voltage-Frequency-Scaling With Optimal Core Allocation and Thread Hopping for the 80-Core TeraFLOPS Processor”, IEEE J. of Solid-State Circuits, vol. 46, no. 1, pp. 184-193, Jan. 2011.

[2.12] L. Chang, Y. Nakamura, R. K. Montoye, J. Sawada, A. K. Martin, K. Kinoshita, F. H. Gebara, K. B. Agarwal, D. J. Acharyya, W. Haensch, K. Hosokawa and D. Jamsek, “A 5.3 GHz 8T-SRAM with Operation Down to 0.41 V in 65 nm CMOS,” Symp. on VLSI Circuits Dig. Tech. Papers, pp. 252-253, 2007.

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

[2.14] V.K. Reddv, A.S. Al-Zawawi, and E. Rotenberg, “Assertion-Based Microarchitecture Design for Improved Fault Tolerance,” Proceedings of IEEE Int. Conf. on Computer Design, pp.362-369, 2006.

[2.15] B. Eklow, A. Hosseini, C. Khuong, S. Pullela, T. Vo, and H. Chau, “Simulation Based System Level Fault Insertion Using Co-verification Tools,” Proceedings of IEEE Int. Test Conference, pp.704-710, 2004.

[2.16] C.R. Elks, M. Reynolds, N. George, M. Miklo, S. Bingham, R. Williams, B.W. Johnson,

M. Waterman, and J. Dion, “Application of a fault injection based dependability assessment process to a commercial safety critical nuclear reactor protection system,” Proceedings of IEEE/IFIP Int. Conf. on Dependable Systems and Networks, pp.425-430, 2010.

Chapter 3 Process-Variation-Aware Cache Architecture Using 7T/14T SRAM

In this chapter, a novel cache architecture using 7T/14T SRAM, which can improve its reliability with control lines, is introduced. Our proposed 14T word-enhancing scheme can enhance its operating margin in word granularity by combining two words in a low-voltage mode. Furthermore, a new testing method that maximizes the efficiency of the 14T word-enhancing scheme is proposed. In a 65-nm process, it can reduce the minimum operation voltage (V_{min}) to 0.5 V to a level that is 42% and 21% lower, respectively, than those of a conventional 6T SRAM and a cache word-disable scheme. Measurement results show that the 14T word-enhancing scheme can reduce V_{min} of the 6T SRAM and 14T dependable modes by 25% and 19%, respectively. The respective dynamic power reductions are 89.2% and 73.9%. The respective total power reductions are 44.8% and 20.9%.

3.1 Introduction

A word-level enhancing scheme using 7T/14T SRAM for a large-capacity cache is presented in this chapter. The proposed 14T word-enhancing scheme is implemented with leveraging the word cut-off and with combining a 7T less-marginal bitcell to an adjacent 7T bitcell. The 14T word-enhancing scheme can reduce V_{min} lower than the cache word-disable scheme proposed by [3.2] because it can enhance the operating margin of the defective bitcell by making use of the 14T structure.

In the next section, works related to the cache for low-voltage operation or yield enhancement are described. Then, the 7T/14T SRAM bitcell and its operating modes, and compare bit error rates (BERs) of 7T/14T SRAM with other conventional schemes are introduced in Section 3.3. Section 3.4 presents a description of the proposed 14T word-enhancing scheme and the proposed incremental testing scheme. Then, the simulated and measured improvements of V_{min} compared with the conventional scheme are reported. Detailed descriptions of the physical implementation of the 14T word-enhancing scheme are also presented. In Section 3.5, a comparison of performance, energy, and power between the conventional scheme and the proposed

scheme are described. Finally, Section 3.6 concludes the chapter.

3.2 Related Work

Wilkerson et al., proposed the cache word-disable scheme (‘the word-disable scheme’ hereinafter) and the cache bit-fix scheme (‘bit-fix scheme’) enabling low-voltage operation [3.2]. The word-disable scheme disables defective words and selects four workable words from eight words. A defect word map (one-bit information per word), which shows which words are defective and valid, is stored in a cache tag. The word-disable scheme purges the remaining four words. Therefore, the cache size and associativity must be halved. The number of ways is reduced to four from eight in studies described in the literature.

The bit-fix scheme exploits one strategy for redundancy: it stores locations of defective bits in the remaining three ways along with patch bits for them. Then, the defective bits are replaced with the patch bits. The number of ways results in six from eight, which means that the area overhead is smaller than the word-disable scheme. However, the bit-fix scheme suffers a three-cycle penalty, whereas that in the word-disable scheme suffers only a one-cycle penalty. In low-voltage operation, the reliability in the redundant way is lowered as much as the other three ways, where slow error correction coding (ECC) must be implemented. The bit-fix scheme cannot operate at a lower voltage than the word-disable scheme because the failure rate is increased rapidly in the redundancy way. Even ECC cannot fix it.

That earlier study applied a word-disable scheme and the bit-fix scheme to L1 caches and L2 cache, respectively, achieving V_{min} reduction to 0.5 V. Nevertheless, detailed conditions of the failure rate in their 6T SRAM were not described clearly. The failure rate for the redundancy way was not considered in their report.

Ozdemir et al. proposed a yield-aware cache architecture and specifically addressed cache access latency and leakage power [3.3]. They developed four schemes: The first one disables cache ways that have timing failures or excess leakage to improve the cache yield. The second also disables horizontal regions in the cache. The third one changes cache access latency in each cache way. The fourth is a hybrid scheme of the first, second, and/or third schemes. They reduced the yield losses by 81.1% using the fourth hybrid scheme. However, they evaluated the yield only with access latencies and

leakage power, although margin analysis in SRAM is fundamental to the yield evaluation at a low voltage.

3.3 7T14T SRAM

3.3.1 Failures in SRAM

Failures in SRAM are categorized as read margin failure, write margin failure, soft error, and access time violation.

• **Read margin failure**: a read operation is signified by a read static noise margin (read SNM) [3.9]. If the read SNM becomes zero by a low V_{dd}, a noise source, or destructive readout, then the stored datum flips.

• **Write margin failure**: a write operation is explainable by a write-trip point (WTP) as a metric (= write margin) [3.10]. The WTP represents the maximum voltage that can write '0' to a bitcell and can then flip an internal datum.

• **Soft error:** an alpha ray or neutron collides against SRAM on an LSI at a certain probability. As a result, a noise current flows through transistors. Data inversions often occur in SRAM around the collision point.

• **Access time violataion** occurs when a differential voltage between bitlines is small and a sense amplifier cannot sense it in a predetermined acceptable time. The access time violation is dependent on the clock frequency and a timing guard band. This failure type is not incorporated into the discussion presented in this chapter because it is dependent on the clock frequency. The read SNM and write margin are dominant at low operating frequencies and low operating voltages.

3.3.2 7T/14T SRAM

Fig. 3.1 depicts the 7T bitcell (14T for two bitcells) [3.5]. Two pMOSes are added to internal nodes ('N00 and N10', 'N01 and N11') in a pair of the conventional 6T bitcells presented in Fig. 3.2. The area overhead in the 7T bitcell is 11% greater than that of the conventional 6T bitcell.

Table 3.1 shows that the 7T/14T bitcells have two modes.

• Normal mode (7T): The additional transistors are turned off (CL = 'H'); the 7T cell acts as a conventional 6T cell.

• Dependable mode (14T): The additional transistors are turned on (CL = ‘L’); the internal nodes are shared by the bitcell pair. In a write operation, both WL0 and WL1 are driven, but in a read operation, either WL0 or WL1 is asserted, which ensures stable operations.

In the normal mode, a one-bit datum is stored in one bitcell, which means that it is more area-efficient. In the dependable mode, a one-bit datum is stored in two bitcells, although the reliability of the information differs from that of the normal mode. The ‘more dependable with less failure rate’ information is obtainable by combining two bitcells [3.5]. In addition, the 14T dependable mode has better soft-error tolerance than the 7T normal mode because its internal node has more capacitance.

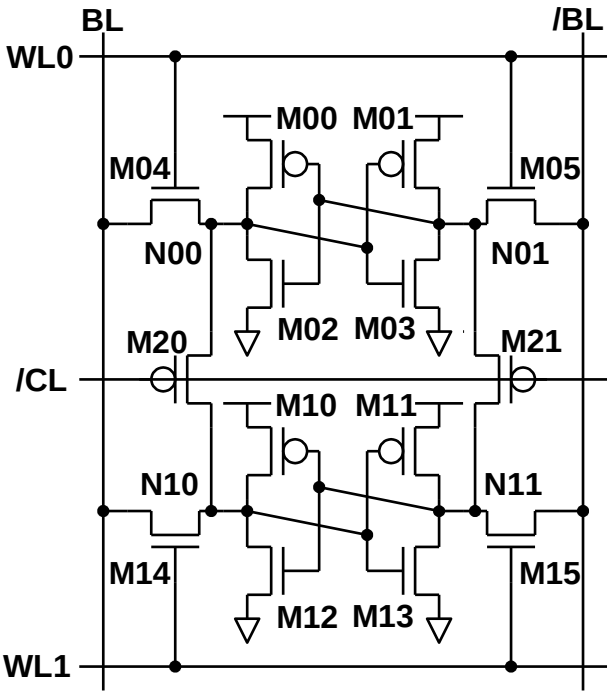


Fig. 3.1 A 7T/14T bitcell pair.

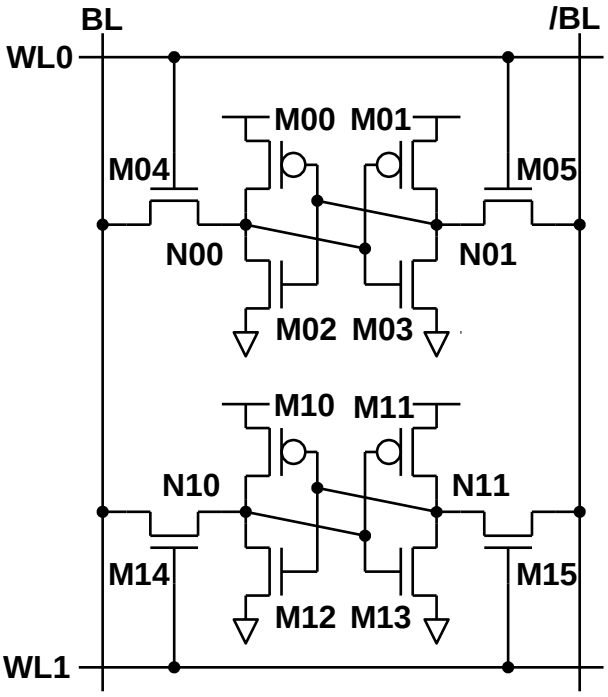


Fig. 3.2 Conventional 6T bitcells.

Table 3.1 Two modes in 7T/14T bitcell

	# of bitcells comprising 1 bit	# of WL drives	CL
Normal	1 (7T/bit)	1	Off (“H”)
Dependable (write)	2 (14T/bit)	2	On (“L”)
Dependable (read)	2 (14T/bit)	1	On (“L”)

3.3.3 Bit Error Rates (BERs)

Fig. 3.3 presents the bit error rates (BERs) simulated in a commercial 65-nm process. As described herein, the BER is referred as a metric in terms of the failure rate. The BERs in the 7T normal bitcell and the 14T dependable bitcell were obtained through Monte Carlo circuit simulation. The BERs in other scheme were obtained by probabilistic calculations using the above BERs in the 7T and 14T bitcells. Detailed descriptions of the probabilistic calculations are presented in the Appendix section. We

also consider the worst-case parameters: temperature and a process corner.

Fig. 3.4 portrays a magnified view of the area bounded by the dashed line in Fig. 3.3. Assuming 99.9% yield in 32-KB caches (999 good 32-KB caches out of 1,000), the respective V_{min} in the conventional 6T bitcell, one-bit ECC for a 32-bit word (= 32 bits + 6 correction bits) using 6T bitcells, the word-disable scheme, the bit-fix scheme, and the 14T dependable mode are 0.8 V, 0.685 V, 0.61 V, 0.615 V, and 0.620 V. Furthermore, assuming 99.9% yield in 4-MB cache, their V_{min} values respectively become 0.855 V, 0.72 V, 0.63 V, 0.645 V, and 0.66 V. The BER curve in the 7T normal mode is the same as that of the conventional 6T bitcells. The word-disable scheme can operate at lower V_{min} than the other schemes at both 32 KB and 4 MB sizes. In this simulation, the 14T dependable mode is applied uniformly to the entire cache (see Fig. 3.9(a)); its BER slope is gentler than that of the word-disable scheme and the bit-fix scheme that exploits the word-grain control and the bit-grain control. Fine-grain control such as the word-grain control or the bit-grain control is more efficient than uniform control for a low BER at a low voltage because it can choose superior bitcells selectively and can abandon less-margin bitcells in the fine-grain region. However, the uniform control of the 14T dependable mode in this simulation uses all pairs of bitcells. Therefore, we apply fine-grain control to the 14T dependable mode in the next section.

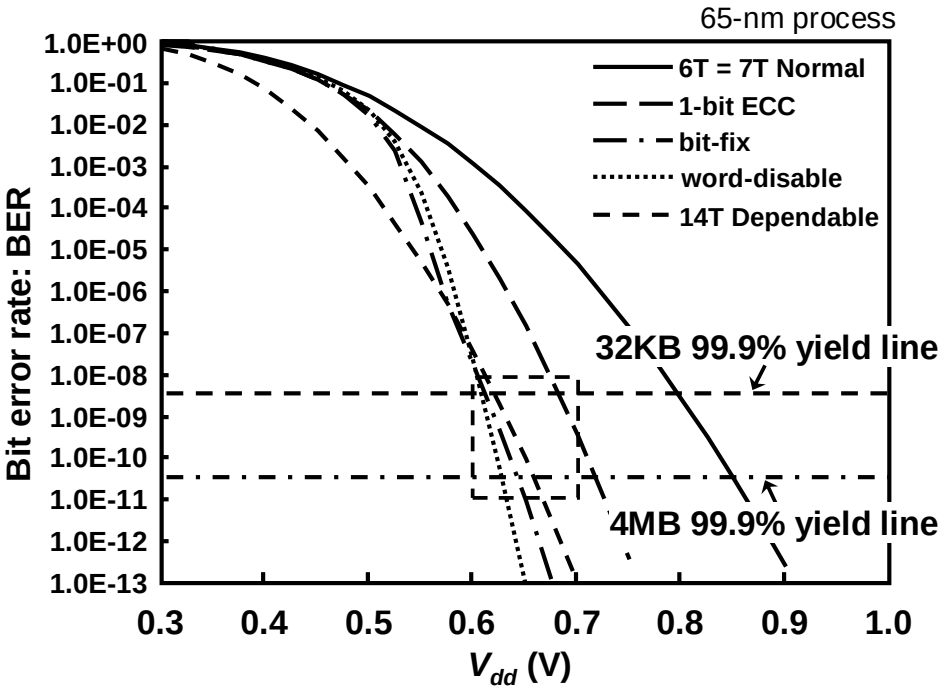


Fig. 3.3 BERs for 32-bit cache: “6T”, “1-bit ECC”, “bit-fix” and ”word-disable” use conventional 6T bitcell schemes;“7T normal” and “14T dependable” use 7T/14T bitcells.

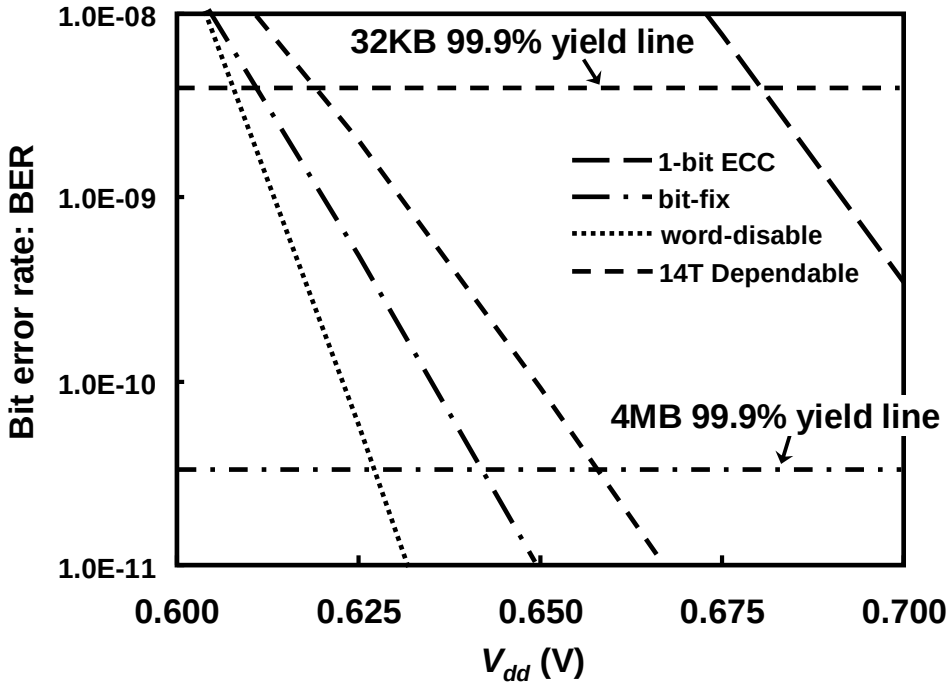


Fig. 3.4 BERs: magnifying the area bounded by the dashed line in Fig. 3.3.

3.4 Implementation of the 14T Word-Enhancing Scheme

In this section, we describe the proposed 14T word-enhancing scheme that enhances the operating margins of bitcells on the word-grain level. Then we will introduce incremental testing to improve the yield further. That is to say, the degree to which V_{min} is reduced using the proposed schemes will be demonstrated through comparison with the conventional word-disable scheme.

3.4.1 Conventional word-disable scheme

As described in Section 3.2, the word-disable scheme was proposed in an earlier report in the literature 2). The word-disable scheme purges defective words, combines two cache lines in two consecutive ways, and thereby produces one logical cache line. Consequently, this scheme halves the cache size and associativity by cutting out the defective words. Each way's tag has a defect word map as one-bit information that signifies a defective word (1) or a valid word (0). In a single 64-B cache line, it includes 16 sets of 32-bit words, which means that each cache line has an additional 16-bit defect word map in its tag.

Fig. 3.5 portrays a comprehensive view of the cache word-disable scheme. A 16-word cache line is halved (Word0–Word7 and Word8–Word15). In every stage, a word shifter removes a defective word (or weak word). That is, four defective words are removed in all through the four stages. Four defect-free words (strong words) remain in each path. Eventually, 8 defect-free words are obtainable out of 16 by merging the two sets of 4 defect-free words.

Fig. 3.6 presents a block diagram of a word shifter that removes defective words, and presents an example in which the second word is defective and removed. First, a defect vector ('01000') is extracted from the defect word map. The converting logic, similarly to a decoder, converts the 1-hot defect vector into a multiplexer control vector (0111) that controls four 32-bit 2:1 multiplexers to shift out the defective word.

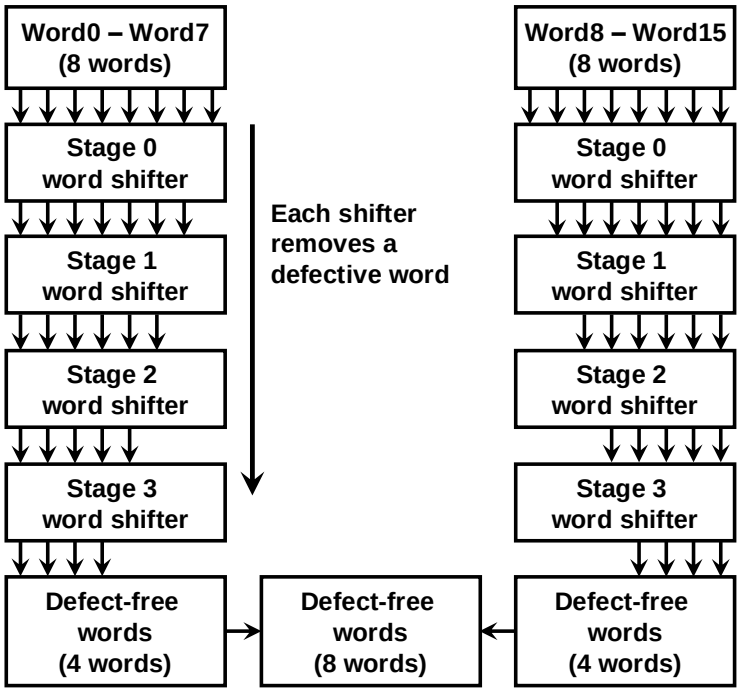


Fig. 3.5 Comprehensive view of the cache word-disable scheme.

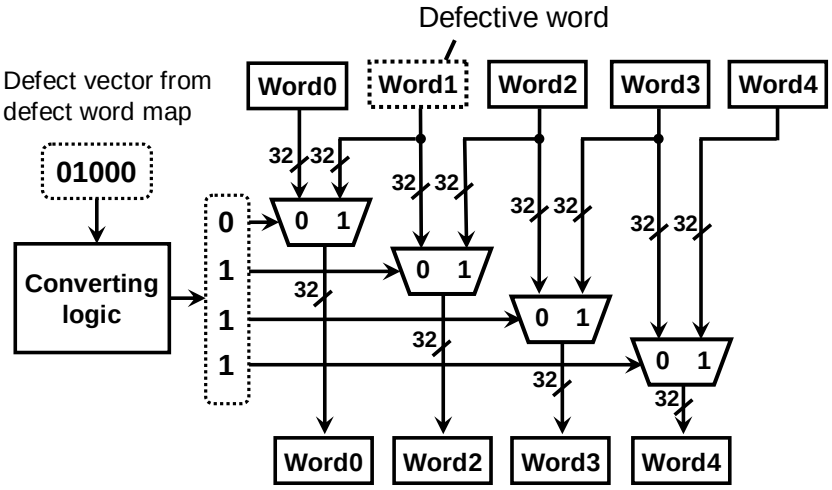


Fig. 3.6 Block diagram of a word shifter.

3.4.2 Proposed 14T word-enhancing scheme with a divided control line

The proposed 14T word-enhancing scheme is a method to use of the 14T dependable mode for word-grain control. We assert a control line using a divided control line (DCL) scheme to select either the 7T normal mode or the 14T dependable mode on the word-grain level. The circuit function of the DCL scheme resembles the divided

word-line (DWL) scheme 11). The DCL scheme divides a global control line (GCL) into local control lines (LCLs) dedicated to each word. Fig. 3.7 depicts a schematic of the 7T/14T SRAM with the DCL scheme. A GCL and a control line selection (CLS) signal control an LCL on row-by-row and column-by-column bases. In addition, a global word line (GWL) is divided into local word lines (LWL), one of which is asserted by the GWL and a word line selection (WLS) signal in the same way. Dedicated decoders, which are controlled by a defect vector from the defect word map, assert a CLS and WLS signals.

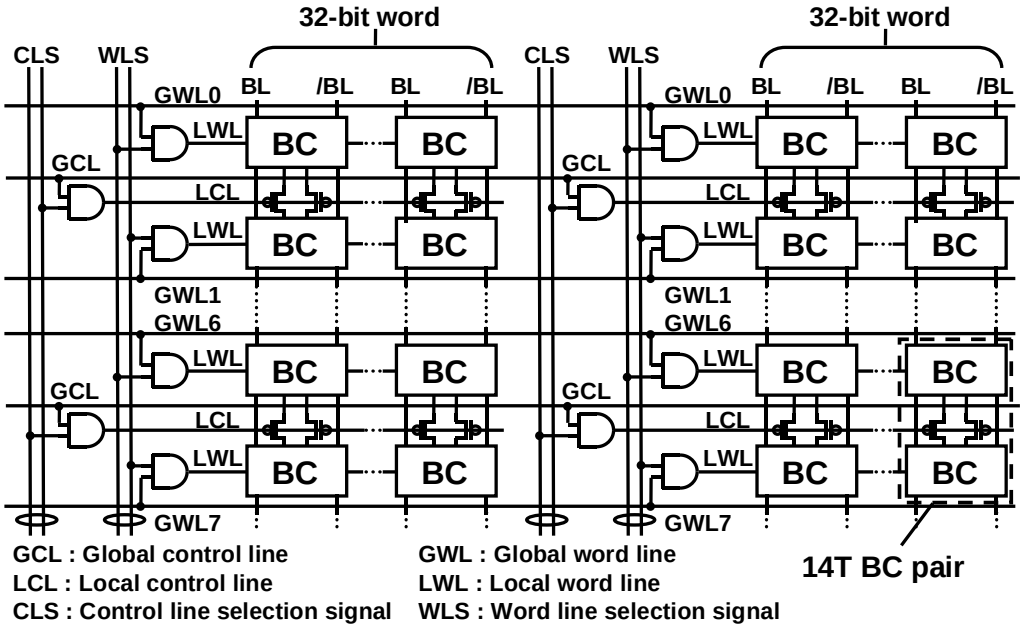


Fig. 3.7 7T/14T SRAM bitcell (BC) array with the divided control line (DCL) scheme.

3.4.3 Incremental testing for the 14T word-enhancing scheme

Fig. 3.8 portrays BERs including a word-level BER of the 14T word-enhancing scheme. The BER of the bit-fix scheme is removed. It is not included in the following comparison because the word-disable scheme is superior to the bit-fix scheme in terms of low-voltage operation and the cycle penalty.

On the 32-KB and 99.9% yield line, V_{min} of the 14T word-enhancing scheme is 0.605 V. On the 4-MB and 99.9% yield line, V_{min} is 0.62 V. As this figure shows, the 14T word-enhancing scheme yields only a small benefit compared to the conventional word-disable scheme because the BER of the 14T word-enhancing scheme is extracted

from conventional testing without consideration of its features. Conventional testing means testing by lowering voltage, with subsequent checking to determine whether each bitcell fails or not.

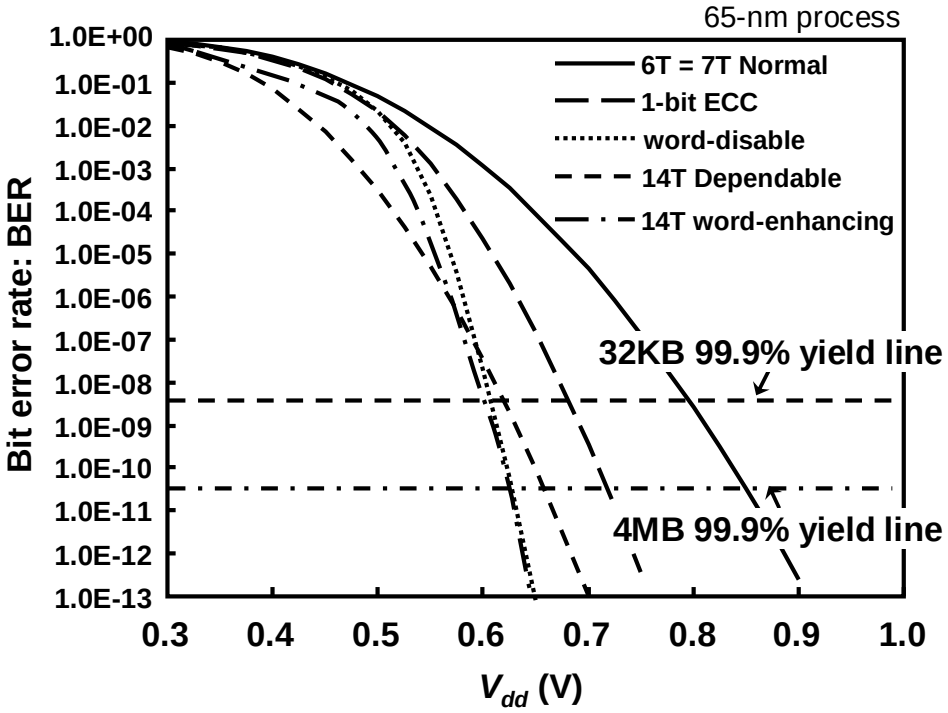


Fig. 3.8 BERs: including the 14T word-enhancing scheme with conventional testing.

The conventional scheme, which performs control on a whole block level, applies the 14T dependable mode uniformly to all word pairs, as portrayed in Fig. 3.9(a), whereas the 14T word-enhancing scheme reinforces a defective word using another half of a pair connected to the word in a testing phase. In low-voltage testing, however, if both words in a 14T pair are recognized as defective words simultaneously at a certain voltage, then such a word pair cannot be applied to the 14T dependable mode, as shown in Fig. 3.9(b). In fact, the 14T word-enhancing scheme can reduce its V_{min} efficiently in the case in which the 14T dependable mode is applied to all word pairs, as presented in Fig. 3.9(c). To do so, we propose incremental testing that exploits the salient feature of the 14T dependable mode.

Incremental testing is based on the idea of applying the 14T dependable mode incrementally to the word pairs to maximize the number of word pairs. Incremental testing adopts one word pair on even and odd lines for the 14T dependable mode within

a single execution of testing.

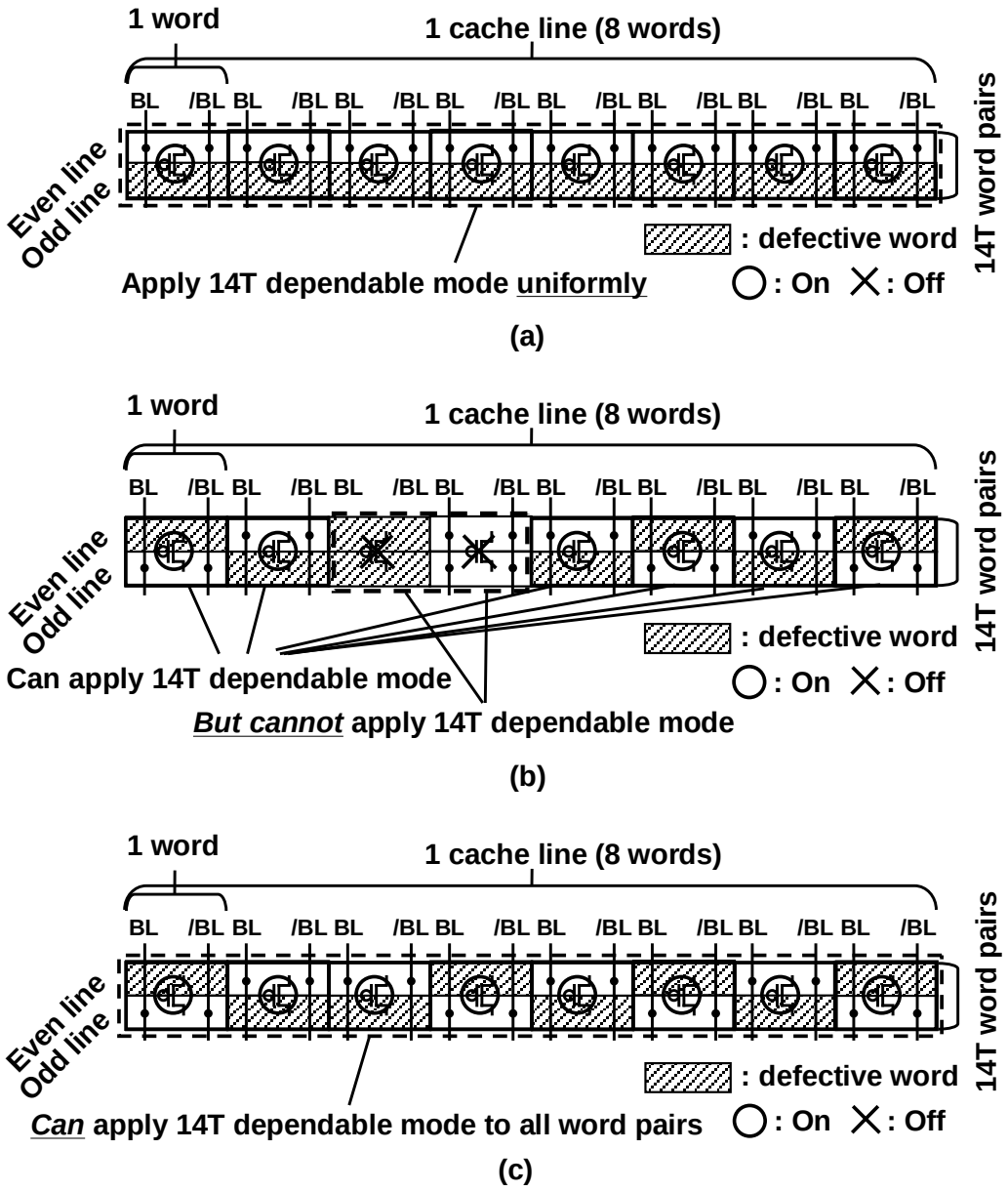


Fig. 3.9 Applying the 14T dependable mode in testing. These examples use eight-word cache lines for simplicity. Only asserted bitlines are shown. (a) Dependable mode is applied uniformly to all word pairs. (b) Conventional testing by which the 14T dependable mode is not applied to all word pairs. (c) Incremental testing, where the 14T dependable mode is applied to all word pairs.

Fig. 3.10 portrays a flow chart showing the incremental testing process. We take a step of an incremental V_{dd} as 50 mV 7). First, the testing V_{dd} is set to a nominal voltage. Next, testing is executed to evaluate whether defective words are detected or not. If

detected, then the 14T dependable mode is applied to the defective words: one word in a pair at most. Then testing is executed again for the updated 14T pair. If defective words are not detected, then the testing V_{dd} is decreased by 50 mV and testing continues. Before every testing execution, the number of disable words is checked to determine whether it equals or exceeds eight words (= half of the whole words in a cache line) or not. The incremental testing finishes if it is equal. If it is greater, then the number of disable words is limited to half for the cache line function, so that the 14T dependable mode is not applied to the excess words.

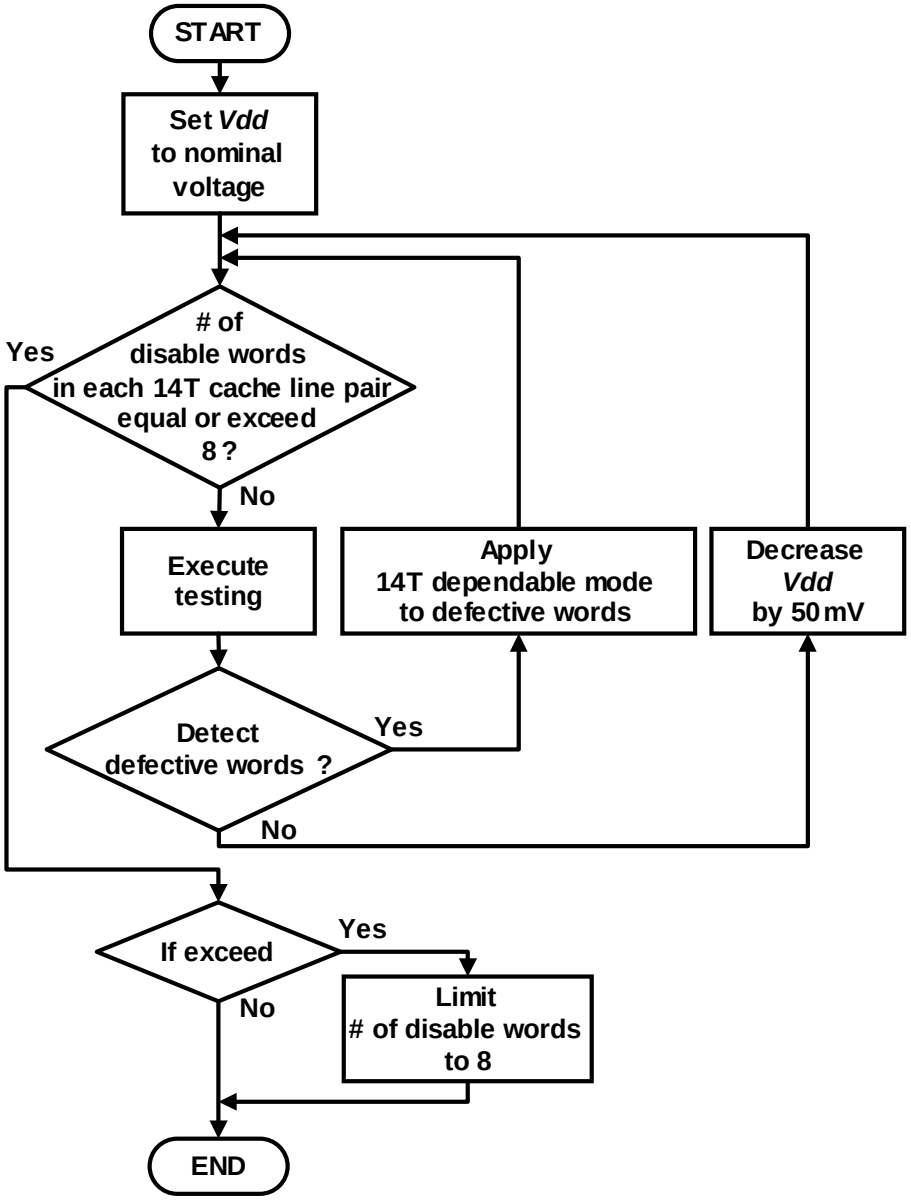


Fig. 3.10 Flow chart of incremental testing (this figure shows the case of an eight-word cache line).

3.4.4 Improved BER in the 14T word-enhancing scheme

Fig. 3.11 shows the BER of the 14T word-enhancing scheme with incremental testing. On the 32-KB and 99.9% yield line, V_{min} in the 14T word-enhancing scheme is improved further to 0.49 V. On the 4-MB and 99.9% yield line, it is 0.5 V, which is 42% and 21% lower, respectively, than the conventional 6T SRAM and the word-disable scheme. The figure shows that the 14T word-enhancing scheme with the incremental testing can reduce V_{min} effectively and that incremental testing is necessary for the 14T word-enhancing scheme.

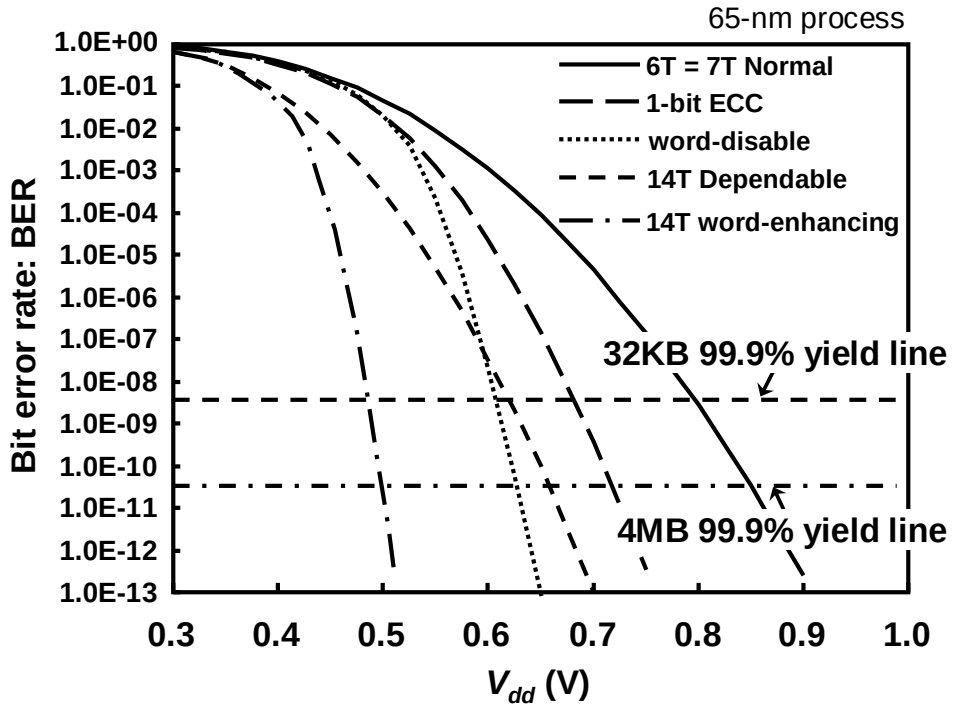


Fig. 3.11 Bit error rates (BERs): applying 14T word-enhancing with incremental testing.

3.4.5 Implementation

Fig. 3.12 shows a layout plot of a 4-MB cache implemented with the 14T word-enhancing scheme using the 65-nm design rule.

The tags must also operate under 0.5 V. The word-disable scheme guarantees low-voltage operation capability in the tags by application of 10T sub-threshold (ST) bitcells [3.6]. The ST 10T bitcells, however, constitute a large area overhead. Instead, we implement a tag with large 6T bitcells that can suppress random (local) variation.

The 6T bitcells for the tags are 1.3 times larger than normal 6T cells, which is 35% smaller than the ST 10T bitcell. The large 6T bitcell can operate reliably at 0.5 V.

The respective area overhead values attributable to the tags and DCL with the dedicated decoders are 4% and 8.9% of those in the conventional 6T SRAM. The total area overhead including the tags, the DCL with the dedicated decoders, and the 7T/14T SRAM, is 24% and 8% of the respective overhead values of the conventional 6T SRAM and the word-disable scheme.

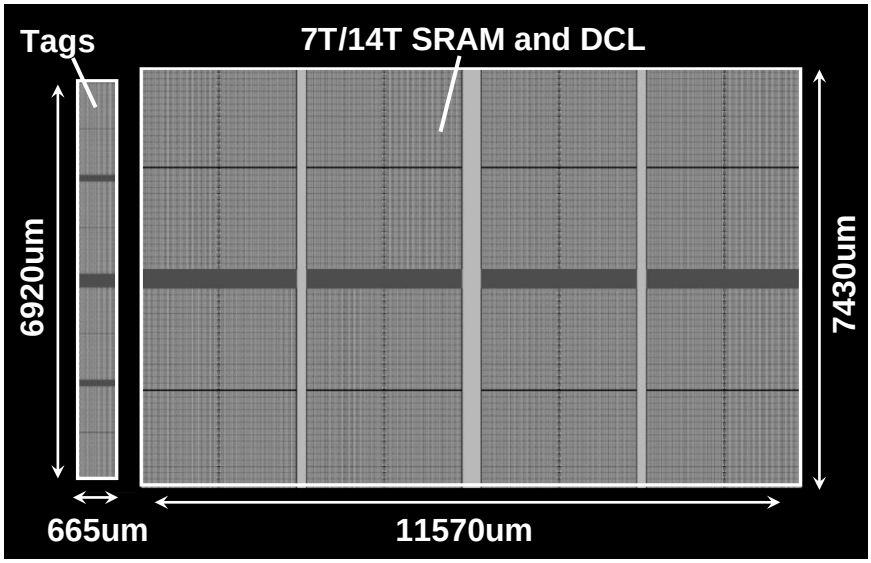


Fig. 3.12 Layout plot of a proposed 4-MB cache implemented with a 65-nm process.

3.4.6 Measurement result

To show the voltage reduction in our scheme, we fabricated a 512-kb SRAM macro with the proposed 14T word-enhancing scheme in a 65-nm process.

Fig. 3.13 shows the measured BERs of the 6T normal, 14T dependable, and 14T word-enhancing schemes. The function of the incremental testing is conducted off the chip in this evaluation environment. The respective first failure bits of the 6T normal, 14T dependable, and 14T word-enhancing schemes come out at 0.53 V, 0.49 V, and 0.3975 V (i.e., the respective V_{min} are 0.5325 V, 0.4925 V, and 0.4 V). From this measurement, it is apparent that the 14T word-enhancing scheme can function effectively in a low-voltage region and reduce V_{min} under the variation of the fabricated 65-nm chip.

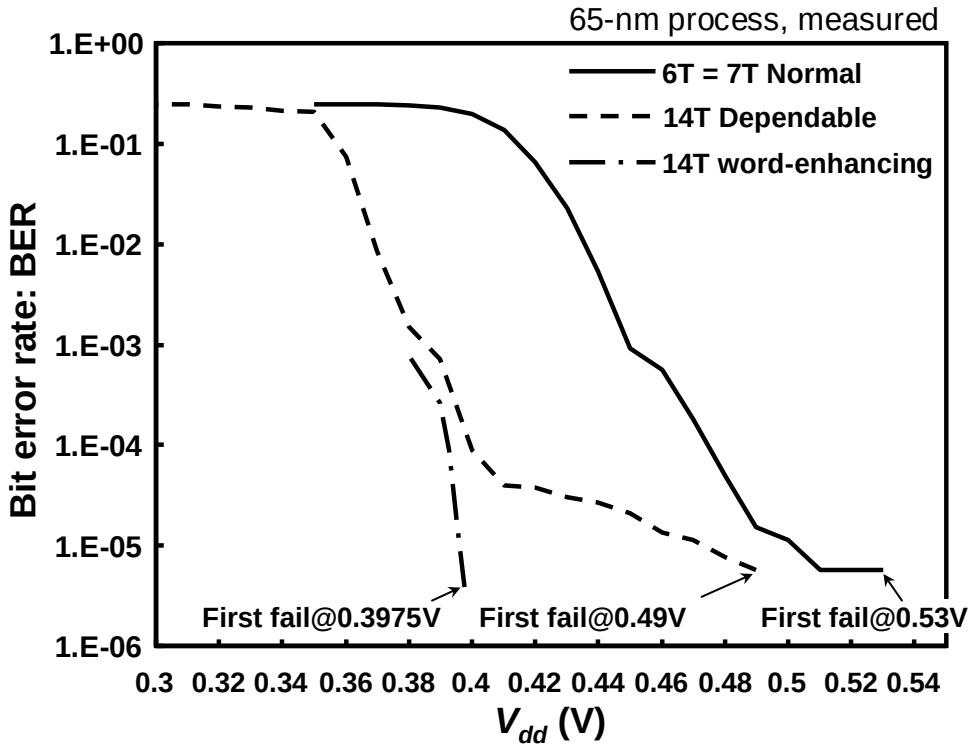


Fig. 3.13 Measured BERs of 6T, 14T dependable, and 14T word-enhancing scheme in 512-kb SRAM macro.

3.5 Performance, Energy, and Power Comparison

3.5.1 Performance evaluation

In this section, we will make a performance comparison between the conventional scheme and the proposed scheme. The performance degradation derived from the additional latencies and the cache capacity reduction must be evaluated quantitatively. We used the SESC [3.8] cycle-accurate simulator. Table 3.2 presents the architectural configuration parameters dependent on V_{dd} and the energies consumed on the cache in a single operation. The cache energies presented in Table 3.2 will be explained in Section 3.5.2.

We assumed a 20 FO4 gate delay for a single pipeline stage and obtained the operating frequencies in these 65-nm SPICE simulations. Table 3.3 presents the architectural configuration parameters that are independent of V_{dd}. The 14T word-enhancing and the word-disable have access time overhead derived respectively from the dedicated decoder assertion of the CLS and WLS signals and the word-disable

circuitry. Consequently, the 14T word-enhancing scheme and the word-disable scheme have a one-cycle penalty each for all cache accesses over the 6T normal mode.

Table 3.2 Cache architecture configuration parameters dependent on V_{dd}: Energies of single operation are derived from 65-nm SPICE simulations and CACTI

	6T normal		Word-disable		14T word-enhancing	
	High-voltage operation	Low-voltage operation	High-voltage operation	Low-voltage operation	High-voltage operation	Low-voltage operation
Vdd (supply voltage)	1.2 V	0.855 V	1.2 V	0.63 V	1.2 V	0.5 V
Frequency	2.6 GHz	1.7 GHz	2.6 GHz	900 MHz	2.6 GHz	500 MHz
DRAM access latency	260 cycles	170 cycles	260 cycles	90 cycles	260 cycles	50 cycles
L1$ read op. energy	0.187 nJ	0.095 nJ	0.267 nJ	0.072 nJ	0.188 nJ	0.033 nJ
L1$ write op. energy	0.181 nJ	0.092 nJ	0.256 nJ	0.071 nJ	0.183 nJ	0.032 nJ
L2$ read op. energy	0.984 nJ	0.500 nJ	1.059 nJ	0.299 nJ	0.992 nJ	0.172 nJ
L2$ write op. energy	0.892 nJ	0.453 nJ	0.969 nJ	0.298 nJ	0.895 nJ	0.155 nJ

Table 3.3 Architecture configuration parameters independent of V_{dd}

# of cores	**2**
Technology	**65-nm CMOS**
L1 Instruction cache	**32KB, 8-way, 2-cycle latency**
L1 Data cache	**32KB, 8-way, 2-cycle latency**
Shared L2 cache	**4MB, 8-way, 14-cycle latency**
Cache line size	**64B**
Fetch / Issue / Retire	**4/4/4**
INT / FP registers	**128/128**

We conducted SPEC2000 CINT (gzip, vpr, gcc, mcf, crafty, parser, gap, vortex, twolf) / CFP (wupwise, swim, mesa, ammp, equake) benchmarks and SPLASH2 benchmark [3.13] (fft, fmm, ocean, lu, radix, barnes, raytrace) as a performance evaluation. Fig. 3.14 presents normalized IPCs in the conventional scheme and the proposed scheme. The IPC reductions in the word-disable and 14T word-enhance schemes are, respectively, 3.8% and 3.7%, on average. They are almost identical.

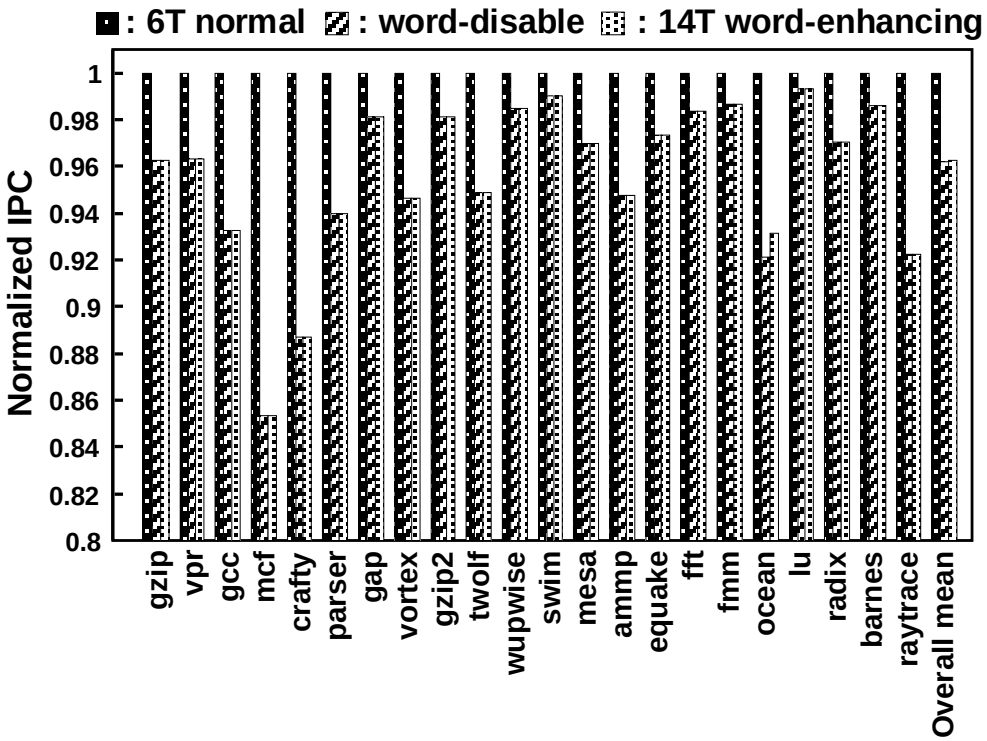


Fig. 3.14 Normalized IPCs in SPEC CPU2000 and SPLASH2 benchmarks.

3.5.2 Energy and power comparison

In the 14T dependable mode, internal nodes of the bitcell have almost double the capacitance of the 7T normal mode. However, the read energy in the 14T dependable mode does not increase from the 7T normal mode because the bitline current is the same as that of the 7T normal mode because the number of asserted wordlines is the same. Nevertheless, the write energy increases because charging and discharging the capacitance associated with the internal node increases. The energy consumed on the wordline is also increased because the number of asserted wordlines is doubled.

CACTI [3.12] was used to estimate energy overheads in the 14T dependable mode, word-disable and 14T word-enhancing schemes for the entire cache. Before the evaluation of cache energy, we first evaluated the write energies in the 7T normal mode and 14T dependable mode for a single 7T/14T bitcell by 65-nm SPICE simulations. The write energies per bitcell in the 7T normal mode and 14T dependable mode were, respectively, 5.5214 fJ at 1.2 V and 11.208 fJ at 1.2 V. Furthermore, we evaluated additional peripheral circuitry including the word shifter and additional dedicated decoders in the word-disable scheme, plus driving circuits of GCL, LCL, GWL, and

LWL and additional dedicated decoders in the 14T word-enhancing scheme. By feeding back the energies in the bitcells and additional peripheral circuits to CACTI, the read and write operation energy per cache access is obtainable.

In Table 3.2, we assumed L1I, L1D, L2 caches in the 65-nm technology (LSTP for cell array and HP for peripheral circuitry) for the cache energy evaluation. The read and write energies of each cache are shown in Table 3.2. During high-voltage operation, compared with the read energy overhead of the 6T normal mode, those of the word-disable scheme and 14T word-enhancing scheme are, respectively, 40.05% and 0.32% for L1 caches, and 7.62% and 0.83% for L2 cache. The write energy overheads are, respectively, 41.48% and 1.22% for L1 caches, and 8.72% and 0.32% for L2 caches. In the word-disable scheme, the word shifter consumes great amounts of energy: 75.4 pJ per cache operation for each cache. Consequently, the word-shifter is a major contributor to the large energy overhead of the word-disable scheme. In contrast, the 14T word-enhancing scheme has a reasonable energy overhead even if the 14T dependable mode's write operation and the additional peripheral circuitry are considered.

Figs. 3.15(a) and 3.15(b) portray dynamic energy and dynamic power of 6T normal mode, word-disable, and 14T word-enhancing schemes in the high-voltage operation and low-voltage operation. The SPEC2000 and SPLASH2 benchmarks are used. Each figure is normalized by the 6T normal mode in the high-voltage operation and sums up energies and powers of the L1D, L1I, and L2 caches.

3.5.2.1 Overheads in high-voltage operation

The word-disable scheme in the high-voltage operation has 42.43% energy overhead and 34.33% power overhead on average, against the 6T normal mode. The word-disable consumes large amounts of dynamic energy and dynamic power because of its word shifter for the variation-aware low-voltage operation. In stark contrast, the 14T word-enhancing scheme in the high-voltage operation consumes 7.8% less energy and 10.54% less power on average, compared with the 6T normal mode. This difference results from the increase in cache access latency, which reduces the energy and power used in the 14T word-enhancing scheme.

3.5.2.2 Energy and power reduction in low-voltage operation

During low-voltage operation, the word-disable and 14T word-enhancing schemes

respectively reduce dynamic energy usage by 22.23% and 63.1% compared with the 6T normal mode. The dynamic power reductions are, respectively, 58.66% and 89.22%. Each scheme in the low-voltage operation has a different frequency.

3.5.2.3 Leakage power

Leakage power is also calculated using CACTI, augmented with the data obtained in SPICE simulations. We also assumed a 65-nm LSTP process for a cell array and a 65-nm HP process for peripheral circuitry, as in the energy calculation. During high-voltage operation, the word-disable and 14T word-enhancing schemes consume 14.9% and 25.0% more leakage power than 6T normal consumes. The respective increase in leakage power of the word-disable and 14T word-enhancing schemes is caused mainly by the increase in the number of transistors and area. During low-voltage operation, the respective leakage power reductions of the word-disable and the 14T word-enhancing schemes are 27.1% and 40.0%.

3.5.2.4 Total power

Total power includes the dynamic power and leakage power. During high-voltage operation, the total power used by the word-disable is higher by 17.9% and that used by 14T word-enhancing schemes is higher by 19.6%. During low-voltage operation, however, they are lower, respectively, by 30.2% and 44.8%.

Additionally, we estimate the total power considering the 65-nm LSTP process for both the cell array and peripheral circuitry assuming a low-power mobile processor. During high-voltage operation, the total power of the word-disable scheme is higher by 34.3%, and the total power of the 14T word-enhancing scheme is lower by 10.2%. During low-voltage operation, they are reduced, respectively, by 58.5% and 88.9%. The average ratio of the dynamic power to the leakage power is 1:8.48 for the LSTP process for the cell array in the HP process for the peripheral circuitry. The average ratio is 3400:1 for the LSTP process for both the cell array and peripheral circuitry. The leakage power is dominant in the former case. The dynamic power is dominant in the latter case.

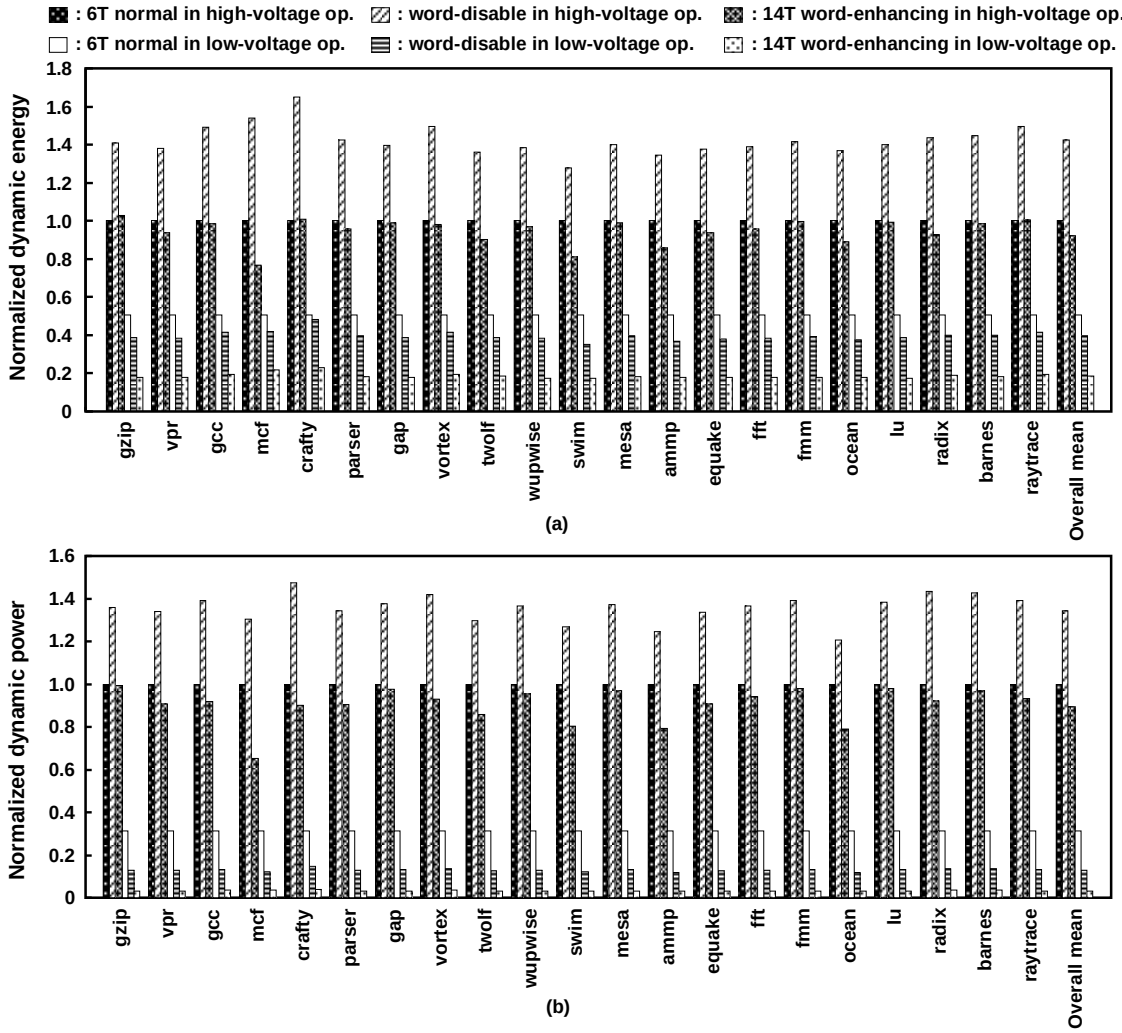


Fig. 3.15 Dynamic energy and dynamic power in high-voltage operation and low-voltage operation in SPEC2000 and SPLASH2 benchmarks. Each figure is normalized by 6T normal in high voltage operation and sums up the energies and powers on the L1D cache, L1I cache, and L2 cache: (a) normalized total energy in each benchmark, (b) normalized total power in each benchmark.

Table 3.4 presents a comparison of the performance of the conventional schemes and the proposed 14T word-enhancing scheme during low-voltage operation. Our proposed scheme can reduce the minimum dynamic power significantly, by 89.2% and 73.9%, respectively, compared to the conventional 6T cell and the word-disable scheme. It can also reduce the total power consumption of the LSTP cell array and the HP peripheral by 44.8% and 20.9%, respectively, and reduce the total power consumption of the LSTP

cell array and the LSTP peripheral by 88.9% and 73.2%, respectively.

Wider-range power scaling is possible when using the proposed scheme, which is suitable for low-power mobile devices that have a low-power operation mode with DVFS.

Table 3.4 Performance comparison: Vmin, area, frequency, IPC, and power during low-voltage operation

	6T cell	Word-disable	14T word-enhancing
V_{min} (mV)	855	630	500
Normalized area	1	1.15	1.24
Frequency (MHz)	1700	900	500
IPC	1.357	1.310	1.309
Normalized dynamic power	1	0.413	0.108
Normalized total power w/ HP peripheral	1	0.698	0.552
Normalized total power w/ LSTP peripheral	1	0.415	0.111

3.6 Summary

We proposed a 14T word-enhancing scheme that lowers $V_{min.}$ It uses a 7T/14T SRAM with divided control lines. The proposed incremental testing expands the efficiency of the 14T word-enhancing scheme, and it can further reduce $V_{min.}$ The proposed architecture achieves V_{min} reduction of 42% and 21%, respectively, for a 4-MB cache compared to the conventional 6T SRAM and the word-disable scheme. Measurement of a 512-kb macro implemented with the 14T word-enhancing scheme revealed 25% and 19% lower V_{min}, respectively, than in the 6T normal mode and 14T dependable mode. The minimum dynamic power was 89.2% and 73.9% lower, and the minimum total power was lower by 44.8% and 20.9%.

3.7 Appendix: Probabilistic BER calculations

Procedures used for probabilistic BER calculations for the one-bit ECC, the bit-fix scheme [3.2], the word-disable scheme, and the 14T word-enhancing scheme are

explained below.

First, the procedure for the one-bit ECC is introduced and explained. The one-bit ECC can fix a one-bit error in a single word. The BER of the one-bit ECC for an n-bit word can be expressed as a binomial expression

$$\begin{aligned} &BER(1-bit_ECC(n)) \\ &= 1-((1-BER(6T))^{n} + n\times((1-BER(6T))^{n-1}\times BER(6T)))^{1/n}, \end{aligned} \tag{1}$$

where BER(6T) denotes the BER for a single 6T bitcell.

Second, in the probabilistic BER calculation of the bit-fix scheme, the bit-fix scheme in the literature has 10 sets of two patch bits per 512-bit cache line. Therefore the bit-fix scheme can repair 10 defects per 512-bit cache line by replacing the 10 defects with the 10 sets of two patch bits. In principle, one patch bit is sufficient to fix one defect, but the address pointing to the defect requires nine bits (512 = 29) in this case. Therefore, in the literature, the two patch bits are adopted, which can repair two consecutive bits with an eight-bit address (512/2 = 28). The 10 bits (two patch bits and eight address bits are further encoded to one-bit-correction ECC, in which four bits are added to the ten bits and the total bits becomes 14 per defect (a one-bit defect and a two-consecutive-bit defect can be corrected by the 14 bits). The BER of the bit-fix scheme is expressed as shown below.

$$\begin{aligned} &BER(Bit-fix) \\ &= 1-(\sum_{i=0}^{10}\binom{256}{i}\times(1-BER(6T))^{2\times(256-i)}\times BER(6T)^{2\times i} \\ &\times BER(1-bit_ECC(14))^{2\times i})^{1/512} \end{aligned} \tag{2}$$

Next, the probabilistic BER calculation for the word-disable scheme is introduced. The word-disable scheme can remove eight defective words from 16 words in one way. In a 512-bit cache line in one way, 16 sets of 32-bit words exist. The 16-word cache line is divided into two halves. The word-disable scheme can then remove four defective words from eight words in the two halves. The BER for the word disable scheme is therefore expressed as follows.

$$BER(Word_disable) = 1 - \left(\sum_{i=0}^{4} \binom{8}{i} \times \left(1 - BER(6T)\right)^{32\times(8-i)} \times BER(6T)^{32\times i}\right)^{2/512} \quad (3)$$

Finally, the probabilistic BER calculation for the 14T word-enhancing scheme is introduced. Actually, the BER of the 14T word-enhancing scheme can be expressed similarly to that for the word-disable as

$$BER(14T_word_enhancing) = 1 - \left(\sum_{i=0}^{4} \binom{8}{i} \times \left(1 - BER(14T)\right)^{32\times(8-i)} \times \left(BER(14T)\right)^{32\times i}\right)^{2/512}, \quad (4)$$

where BER(14T) denotes a BER for a single 14T bitcell.

3.8 References

[3.1] K. Itoh, "Low-voltage scaling limitations for nanoscale CMOS LSIs," Proceedings of Int. Conf. on Ultimate Integration of Silicon, pp. 3-6, Mar. 2008.

[3.2] C. Wilkerson, H. Gao, A.R. Alameldeen, Z. Chishti, M. Khellah, and S.-L. Lu, "Trading off Cache Capacity for Reliability to Enable Low Voltage Operation," Proceedings of ACM/IEEE International Symposium on Computer Architecture, pp. 203-214, Jun. 2008.

[3.3] S. Ozdemir, D. Sinha, G. Memik, J. Adams, and H. Zhou, "Yield-Aware Cache Architectures," Proceedings of ACM/IEEE Int. Symp. on Microarchitecture, pp. 15-25, Dec. 2006.

[3.4] A. Agarwal, B.C. Paul, H. Mahmoodi, A. Datta, K. Roy, "A process-tolerant cache architecture for improved yield in nanoscale technologies," IEEE Trans. on Very Large Scale Integration Systems, vol. 13, no. 1, pp. 27-38, Jan. 2005.

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

[3.6] J.P. Kulkarni, K. Kim, and K. Roy, "A 160 mV Robust Schmitt Trigger Based Subthreshold SRAM," IEEE J. of Solid-State Circuits, vol. 42, no. 10, pp. 2303-2313, Oct. 2007.

[3.7] B. Stackhouse, S. Bhimji, C. Bostak, D. Bradley, B. Cherkauer, J. Desai, E. Francom, M. Gowan, P. Gronowski, D. Krueger, C. Morganti, and S. Troyer, “A 65 nm 2-Billion Transistor Quad-Core Itanium Processor,” IEEE J. of Solid-State Circuits, vol. 44, no. 1, pp. 18-31, Jan. 2009.

[3.8] J. Renau, B. Fraguela, J. Tuck, W. Liu, M. Prvulovic, L. Ceze, K. Strauss, S. Sarangi, P. Sack, and P. Montesinos, “SESC Simulator,” Jan. 2005. http://sesc.sourceforge.net .

[3.9] E. Seevinck, F.J. List, and J. Lohstroh, “Static-noise margin analysis of MOS SRAM cells,” IEEE J. of Solid-State Circuits, vol. 22, no. 5, pp. 748-754, Oct. 1987.

[3.10] R. Heald and P. Wang, “Variability in sub-100 nm SRAM designs,” Proceedings of IEEE/ACM International Conference on Computer Aided Design, pp. 347-352, Nov. 2004.

[3.11] M. Yoshimoto, K. Anami, H. Shinohara, T. Yoshihara, H. Takagi, S. Nagao, S. Kayano, and T. Nakano, "A divided word-line structure in the static RAM and its application to a 64K full CMOS RAM," IEEE J. of Solid-State Circuits, vol. 18, no. 5, pp. 479-485, Oct. 1983.

[3.12] S. Thoziyoor, N. Muralimanohar, J. H. Ahn, and N. Jouppi, ”CACTI 5.1”, Technical Report HPL-2008-20, Hewlett Packard Labs, April 2008.

[3.13] S.C. Woo, M. Ohara, E. Torrie, J.P. Singh, and A. Gupta, "The SPLASH-2 programs: characterization and methodological considerations," Proceedings of ACM/IEEE Int. Symp. on Computer Architecture, pp. 24-36, Jun. 1995.

[3.14] 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 ACM/IEEE Int. Symp. on Low-Power Electronics and Design, pp. 219-224, Aug. 2010.

Chapter 4 Process-Variation-Adaptive NoC with VAVCR and VCPAR

In this chapter, a process-variation-adaptive network-on-chip (NoC) is proposed. As process technology is scaled down, a typical system on a chip (SoC) becomes denser. In scaled process technology, process variation becomes greater and increasingly affects the SoC circuits. Moreover, the process variation strongly affects NoCs that have a synchronous network across the chip. Therefore, its network frequency is degraded. We propose a process-variation-adaptive NoC with a variation-adaptive variable-cycle router (VAVCR). The proposed VAVCR can configure its cycle latency adaptively on a processor core basis, corresponding to the process variation. It can increase the network frequency, which is limited by the process variation in a conventional router. Furthermore, we propose a variable-cycle pipeline adaptive routing (VCPAR) method with VAVCR; the proposed VCPAR can reduce packet latency and has tolerance to network congestion. The total execution time reduction of the proposed VAVCR with VCPAR is 15.7%, on average, for five task graphs.

4.1 Introduction

The minimum feature size of a CMOS process technology is scaled down, which enables higher density and lower chip fabrication cost. However, process variation is increased by technology scaling. Process variation strongly affects system-on-a-chip (SoC) circuit characteristics. A network-on-chip (NoC), which is one SoC that is emerging as a highly efficient network fabric for many-core processors [4.1, 4.12, 4.13], commonly adopts a synchronous design for a network across the chip. The NoC in a many-core processor has many network components, each of which is affected by process variation. The network component delays vary considerably as the network components become more numerous. Therefore, the frequency of a large-scale chip-wide synchronous network is degraded to the level of the slowest network component. Many studies have been undertaken to find means to mitigate the variations of many-core processors using dynamic voltage and frequency scaling (DVFS) [4.2], application scheduling [4.11], fine-grain body biasing (FGBB) [4.2], and dynamic

voltage frequency-core scaling (DVFCS) [4.3]. However, no study has specifically addressed variation in a large-scale chip-wide synchronous network. In this chapter, we examine process variation in an NoC.

The contribution of this chapter is a proposal for a process-variation-adaptive NoC using a variation-adaptive variable-cycle router (VAVCR) and a novel routing scheme named variable-cycle pipeline adaptive routing (VCPAR) for the NoC with the VAVCR. The proposed VAVCR can configure the cycle latency of the router in adaptation to the spatial process variation. Thereby, the NoC with the proposed VAVCR can enhance the network frequency and the overall throughput. The proposed VCPAR is adaptive to the variable cycle latency of the proposed VAVCR. The VCPAR can reduce packet latency and can be tolerant of network congestion.

This chapter is organized as follows. Section 4.2 describes the background of our work including the impact of process variation on NoC circuits. Section 4.3 presents the proposed VAVCR and VCPAR. In Section 4.4, we evaluate the proposed VCPAR method with the proposed VAVCR, and exhibit their effectiveness. Section 4.5 presents discussion of the settings of the network frequency. In Section 4.6, we conclude this chapter.

4.2 Background

4.2.1 Process Variation in NoC

Process variation in an NoC shows up as variation of operating frequencies of individual cores. Considering synchronous designs for entire NoC processor cores in situations where operating frequencies vary, each core in the NoC must synchronize with the slowest core. Therefore, the throughput of the entire NoC processor degrades with increasing impact of the process variation. Global-asynchronous local-synchronous (GALS) designs, in which the fabric with individual cores and network elements operate at their own maximum frequencies, are widely adopted in NoC design. The network portion composed of routers, wires, and buffers is designed frequently at a single frequency and in a single voltage domain [4.3, 4.15, 4.16] because the design of the network portion in an NoC is too complicated and too costly when adopting multi-frequency and multi-voltage design. However, when the network portion is with a

single frequency and a single voltage domain, its operating frequency is determined by the slowest component (such as a router and a buffer) because operating frequencies of routers and buffers distributed across the entire chip vary according to process variation. This issue is extremely important in a large NoC fabricated using scaled process technology.

Fig. 4.1 portrays the operating frequency variation in a GALS NoC. A processor core and a router communicate asynchronously with each other at a different frequency. An operating frequency in a processor core is determined by each maximum operating frequency (FMAX_Pmn). The network frequency on the entire NoC ($F_{network}$) is determined by the minimum (= worst) operating frequency among all routers. Detailed discussion of the variations in a processor core and an NoC are presented respectively in Subsections 4.2.3 and 4.2.4.

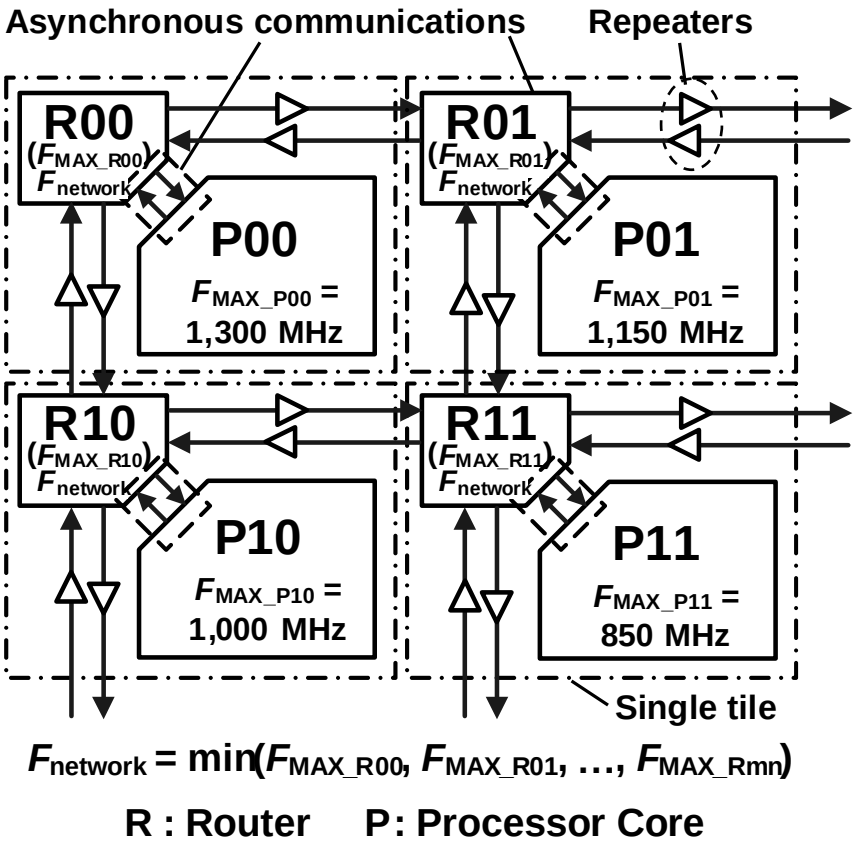


Fig. 4.1 Operating frequency variation in a GALS NoC. The operating frequencies of processor cores (FMAX_Pmn) vary. The network frequency (Fnetwork) is determined by the minimum operating frequency in routers.

4.2.2 Impact of Variation in Processor Core

This section presents a description of the impact of the process variation to the processor core. Assuming a 20 FO4 inverter chain delay as a single pipeline stage in the processor core, we conducted Monte Carlo simulations in a 65-nm process technology using a SPICE circuit simulator. The systematic variation in a threshold voltage (V_{th}) arises as C2C variation. In this simulation, the standard deviation of the systematic variation, σ_{system}, is calculated with [4.4], as 6.3% of the average V_{th}. Random variation is apparent at individual transistors. Consequently, it affects all circuits in the core. We use standard deviations of random variations in NMOSes and PMOSes from actual measurement [4.5]. We set the respective standard deviations, σ_{rnd_NMOS} and σ_{rnd_PMOS}, to 43 mV and 28 mV (in sizing of L = 60 nm and W = 140 nm). The parameters used for estimation of the operating frequency are presented in Table 4.1.

Table 4.1 Parameters used for operating frequency estimation

Technology	**65-nm CMOS**	σ_{system}	**0.063 / μ_{Vth}**
Process corner	**TT**	σ_{rnd_NMOS}	**43 mV**
Temperature	**25℃**	σ_{rnd_PMOS}	**28 mV**
# of Monte Carlo	**10,000**		

Fig. 4.2 shows the distribution of the operating frequencies obtained through simulations of 20 FO4 inverters. We set four frequency bins: 800 MHz, 1,100 MHz, 1,200 MHz, and 1,300 MHz. Details of the frequency bins are presented in Table 4.4. Table 4.2 shows summary statistics of the operating frequency distribution. From Fig. 4.2, the operating frequency variation derived from the V_{th} variation is apparent as a normal distribution. The standard deviation of the operating frequencies, $\sigma_{frequency}$, in the 20 FO4 inverters is 145.4 MHz. Accordingly, the individual processor cores in an NoC under the V_{th} variation represent mutually differing operating frequency characteristics.

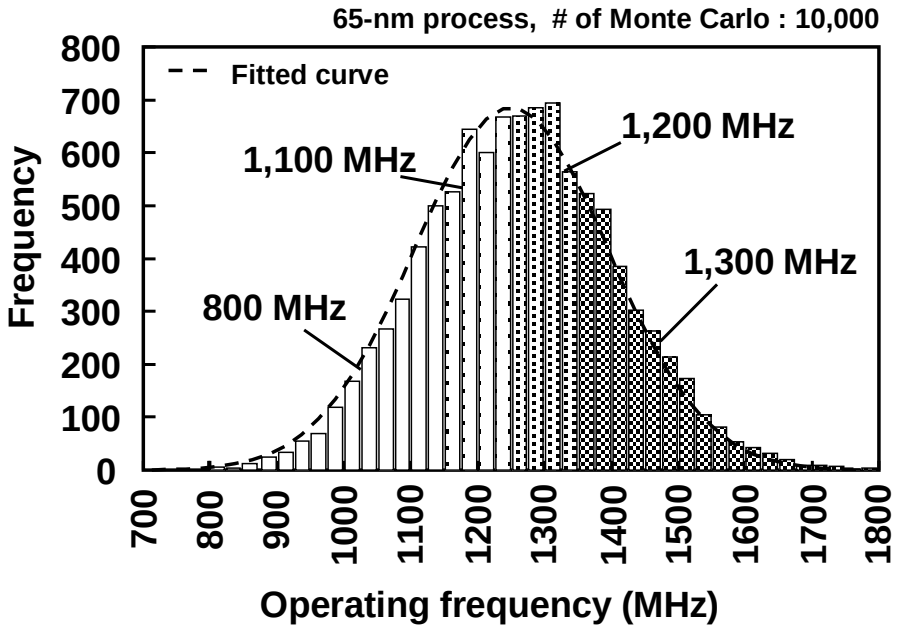


Fig. 4.2 Distribution of operating frequencies of 20 FO4 inverters. The dashed line signifies the fitted normal distribution curve.

Table 4.2 Operating frequency characteristics

$\mu_{frequency}$	1,237.7 MHz	$\mu_{frequency} + 3\sigma_{frequency}$	1,653.6 MHz
$\sigma_{frequency}$	145.4 MHz	$\mu_{frequency} - 3\sigma_{frequency}$	801.5 MHz
Maximum	1,802.1 MHz	Minimum	775.3 MHz

4.2.3 Impact of Variation in On-Chip Networks

Fig. 4.3 depicts the organization of a router with virtual channels (VCs) [4.12]. Fig. 4.4 presents a Gantt chart of each pipeline stage of the router. The pipeline stages are described as follows. The next routing computation stage (NRC) determines a hop direction for the next router, not for the current router. The virtual channel allocation stage (VA) allocates output VCs to the input packets. The switch allocation stage (SA) arbitrates the crossbar switch for the flit. The switch traversal stage (ST) delivers the packet across the crossbar to the output buffer. The link traversal stage (LT) traverses the packet from the output buffer to the next router. The SA, ST, and LT stages operate on every flit of the packet, differently from the NRC and VC stages, which compute once per packet.

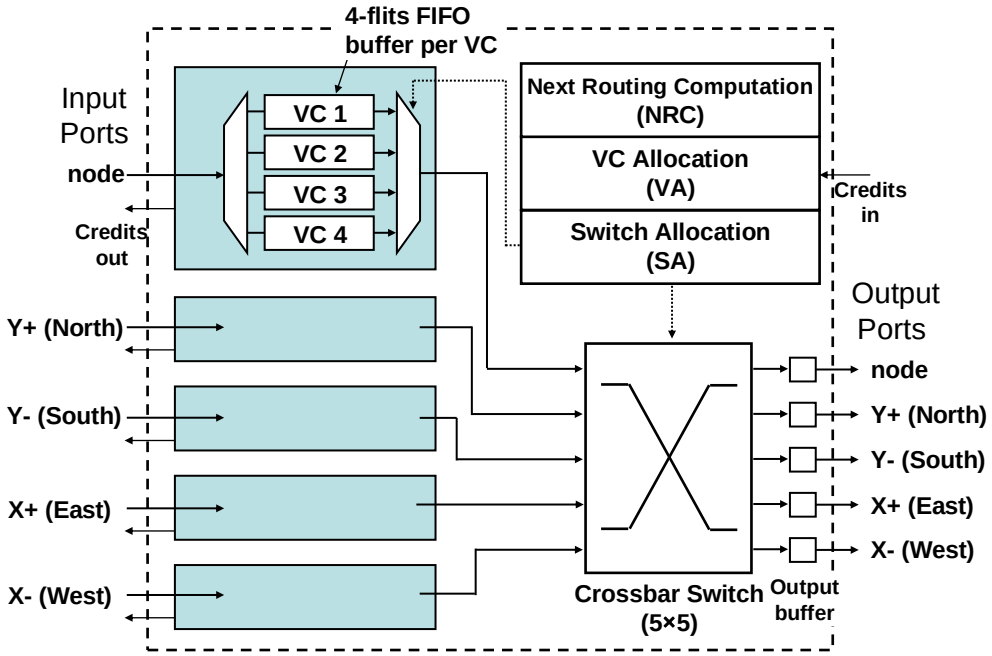


Fig. 4.3 Organization of the router. Parameters are the same as those of Table 4.3.

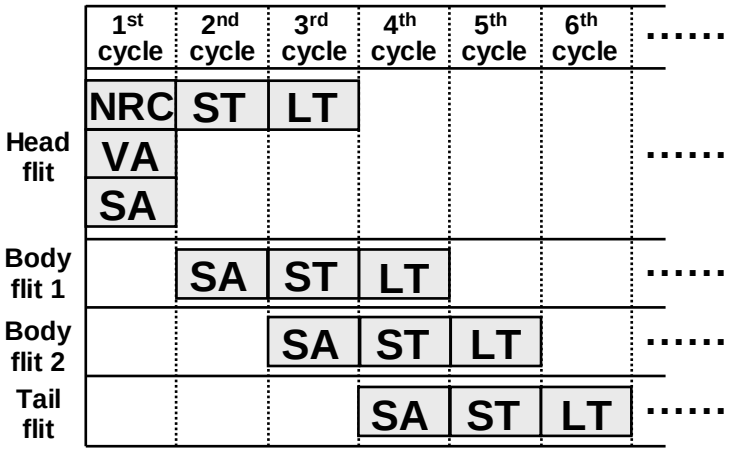


Fig. 4.4 Gantt chart of each pipeline stage of the router.

As described in Subsection 4.2.2, $F_{network}$ is degraded by a single frequency domain for the entire network portion because all components of the network portion must be synchronized with the slowest one. In this section, the delay variation in each pipeline stage of the router is evaluated. We used an open-source RTL of a router [4.6]. The router was synthesized using a 65-nm process technology with Synopsys Design Compiler. The configurations of the router synthesis are shown in Table 4.3. Then, the

synthesized netlist was evaluated using a SPICE circuit simulator, and the delay variation was obtained. As parameters for the variation, the parameters shown in Table 4.1 were used as described in Subsection 4.2.1. We assumed the link length between nodes as 1 mm for the delay evaluation.

Table 4.3 Parameters for router delay estimation

Topology	**8 × 8 mesh**
Flit size	**64 bits**
Routing	**X-Y DOR [19]**
Router type	**Speculative, look ahead**
# of VCs	**4**
VC buffer size	**4 flits**
# of input/output ports	**5 (X+/-, Y+/- and own node)**

The evaluated result for the delays of each pipeline stage is depicted in Fig. 4.5. The upper bound (i.e. the worst delay) of each stage is assumed as 99.7% of the whole. The longest delay in the pipeline stage is the virtual channel allocation (VA) stage. The delay of the VA stage varies: 627–1319 ps.

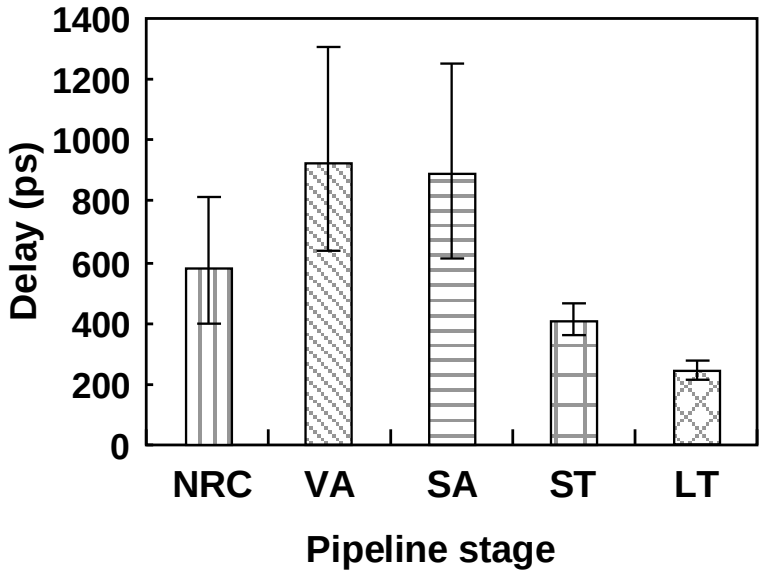


Fig. 4.5 Delay of each pipeline stage: NRC, next routing computation; VA, virtual channel allocation; SA, switch allocation; ST, switch traversal; and LT, link traversal.

4.3 Proposed Process-Variation-Adaptive Variable-Cycle Router and its Proposed Routing Algorithm

4.3.1 Process-Variation-Adaptive Variable-Cycle Router

In this section, a process-variation-adaptive variable-cycle router (VAVCR) is proposed. The proposed VAVCR can configure the cycle latency of the router corresponding to spatial process variation. The VAVCR can realize a variation-adaptive NoC configuration. Fig. 4.6 presents timing diagrams of the conventional and proposed router pipelines. The values of the delays are brought from Fig. 4.5. Figs. 4.6(a) and 4.6(c) show the worst delays (i.e. combination of the upper-bound delays in Fig. 4.5). Figs. 4.6(b) and 4.6(d) show the best delays (i.e. the lower bounds in Fig. 4.5).

In the conventional router pipeline, the router frequency is determined by the worst delay (Fig. 4.6(a)). Accordingly, a great amount of slack emerges at the conventional router pipeline that operates in the best delay (Fig. 4.6(b)). Consequently, the larger the process variation, the greater is the slack at the conventional router pipeline.

Figs. 4.6(c) and 4.6(d) portray timing diagrams of the proposed VAVCR pipeline. The VAVCR pipeline applies multi-cycle paths to NRC, VA, and SA stages (Fig. 4.6(c)) when a delay in the stages exceeds the predefined cycle time. The cycle time is set to 1/1,050 MHz in this example. For the case in which no delay of the pipeline stage exceeds the predefined cycle time, the VAVCR pipeline does not apply the multi-cycle paths; it operates in the same way as the conventional pipeline, but it can do so at a higher frequency (compare Fig. 4.6(d) to Fig. 4.6(b)). Therefore, the proposed VAVCR pipeline can reduce the large slack at the conventional router pipeline, and can realize greater network throughput.

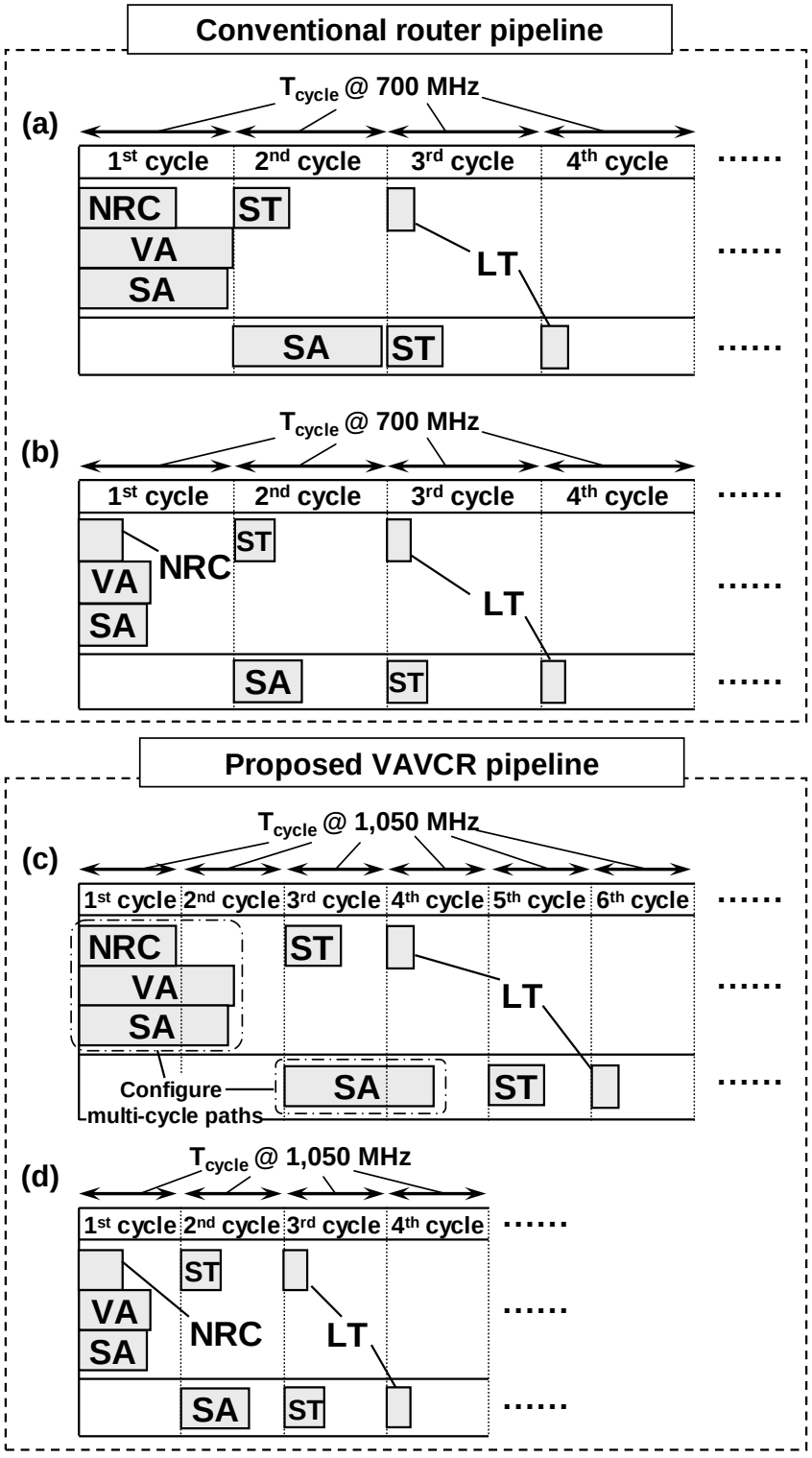


Fig. 4.6 Timing diagrams of the conventional and proposed router pipelines. Here, (a) and (b) respectively correspond to the worst and best delays in the conventional router pipeline; (c) and (d) respectively correspond to the worst and best delays in the proposed router pipeline.

Table 4.4 Frequencies and ratios in the frequency bins and router latencies in the proposed VAVCR

Frequency bin	Frequency 0	Frequency 1	Frequency 2	Frequency 3
Frequency	800 MHz	1,100 MHz	1,200 MHz	1,300 MHz
Ratio	27.6%	25.8%	24.7%	23.2%
Router latency of the proposed VAVCR	4 cycles	4 cycles	3 cycles	3 cycles

Fig. 4.7 portrays a distribution of the router cycle latency for an 8 × 8 mesh network. The number in the circle is the latency of the router. Figs. 4.7(a) and 4.7(b) respectively depict the networks of the conventional router and proposed VAVCR. In the proposed VAVCR, the routers are configured as a three-cycle latency router or a four-cycle latency router corresponding to the spatial process variation on the chip. The pipeline delay of each router can be measured in a burn-in test, and can configure the cycle latency in a testing process. $F_{network}$ can be increased to 1,050 MHz from 700 MHz by applying the proposed VAVCR.

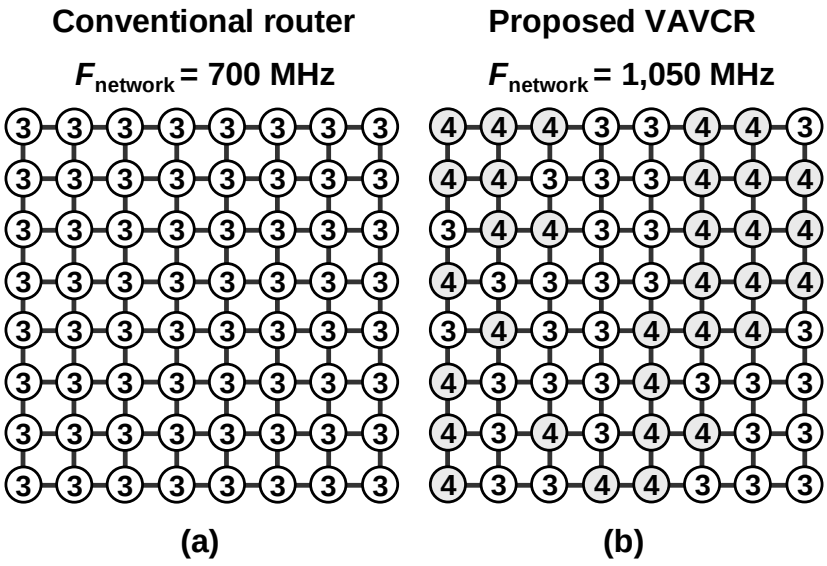


Fig. 4.7 Distributions of the router latencies for 8 × 8 mesh networks: (a) conventional router and the (b) proposed VAVCR.

4.3.2 Variable-Cycle Pipeline Adaptive Routing

In the proposed VAVCR, the packet latency is increased by the variable-cycle router pipeline. In this section, we propose a specific routing algorithm, considering the spatial distribution of the router latency.

The proposed variable-cycle pipeline adaptive routing (VCPAR) employs the odd–even turn model [4.18] to avoid deadlocks; it can select a hop direction adaptively considering their pipeline latencies with neighboring VAVCRs. VCPAR aims for low-latency routing in an NoC with the VAVCRs. The detailed procedure in the VCPAR algorithm is described as follows:

1 Each VAVCR has a distribution of the router latency similar to that shown in Fig. 4.7(b). The distribution information on the mutual router latencies is stored in a testing process.
2 Five ports of a VAVCR have five transmission counters storing the number of packet transmissions.
3 On the way to the destination router, if a next router position (NRP) is at the same row or at the same column, then the hop direction is set to a straight-ahead direction (the row address and the column address are increased or decreased monotonically).
4 In a false case of Procedure 3 and if the destination is toward east:
  4.1 If the NRP is at an even column, then the available direction can be set to east.
  4.2 If the NRP is at an odd column, then the available direction can be set to east or either north or south according to the destination direction.
5 In a false case of Procedure 2 and if the destination is toward the west:
  5.1 If the NRP is at an odd column, then the available direction can be set to west.
  5.2 If the NRP is at an even column, then the available direction can be set to west or either north or south according to the destination direction.
6 If only one direction is available, then the packet is transmitted to that direction.
7 If two directions are available, then the VAVCR checks the transmission counters of the two ports.
  7.1 If the two transmission counters have equal values, then the packet is transmitted to the direction which has the least pipeline latency.

7.2 If the two transmission counters have different values, then the packet is transmitted to the direction which has a lower value

8 The transmission counter in the transmitted direction is incremented by a size of the packet. All transmission counters are decremented by one in each cycle.

The proposed VCPAR reduces packet latency with preferential selection of three-cycle latency routers unless the routers are congested (Fig. 4.8). The VCPAR uses only two-hop-ahead routers' latencies and makes less complexity routing than other routing methods that compute global-variation-adaptive routing paths on an entire NoC. In addition, the transmission counter avoids congestion through specific paths by preferential routing and enhances the communication efficiency.

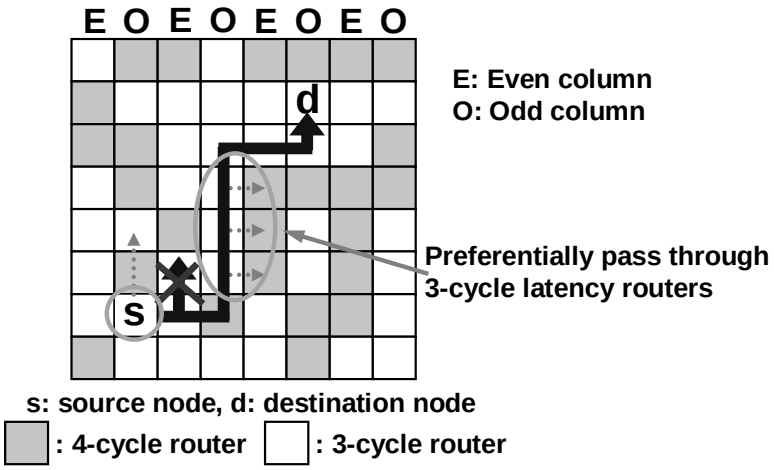


Fig. 4.8 Overview of the proposed VCPAR method.

4.4 Evaluation

In this section, we present an evaluation of the proposed VAVCR and the proposed VCPAR. First, we present the evaluation of routing methods for the NoC with the VAVCR including the conventional routing and the proposed VCPAR in Subsections 4.4.1 and 4.4.2. From this evaluation, the routing method suitable for the NoC with the proposed VAVCR and the effectiveness of the proposed VCPAR can be obtained. Second, we evaluate the proposed VAVCR with VCPAR using task graphs in Subsections 4.4.3 and 4.4.4. Lastly, we estimate the area overhead of the proposed

VAVCR with the VCPAR in Subsection 4.4.5.

4.4.1 Evaluation Methodology of Routing Methods

We used a BookSim simulator [4.7] to evaluate the entire NoC implemented with the proposed VAVCR. The BookSim simulator was modified to evaluate the proposed VAVCR and proposed VCPAR. The router configuration is identical to that shown in Table 4.3, except for the routing method.

The spatial process variation is modeled using a simplified VARIUS model [4.4]. The spatial correlation parameter is assumed as $\Phi = 0.5$ We used a simple spatial process variation model that has the same V_{th} value within a single tile, which includes a processor core, router, and repeater buffers, as shown in Fig. 4.1. The variation parameters in Table 4.1 are used as explained in Section 4.2. The processor core frequencies are determined by the V_{th} variation map and the frequency bins in Table 4.4. The router latencies of the proposed VAVCR are four cycles for Frequency 0 and 1 bins, and three cycles for Frequency 2 and 3 bins. Ten variation maps (chips) are taken in this evaluation. We evaluate the conventional routing method including X-Y DOR [4.19], ROMM [4.20], Toggle X–Y (TXY) [4.21], Odd–Even Random, and the proposed VCPAR method. Odd–Even Random routing uses the Odd–Even turn model in which the next hop direction is determined randomly if it has two available directions. All evaluation in this section, the NoC with the proposed VAVCR is used.

Traffic patterns used for the routing evaluation are uniform random, transpose, bit reverse, hot spot with one hot spot, and hot spot with four hot spots (the hot spot percentage is 6%) [4.22]. The packet size is four flits and 16 flits.

4.4.2 Evaluation of the Routing Method

Figs. 4.9 and 4.10 present the evaluation results of routing methods (respective packet size are four flits and 16 flits). The X-axis shows the injected traffic (packets/ cycle/ node); the Y-axis specifies the average packet latency (= cycles) for the ten variation maps.

In the case of uniform random traffic (Fig. 4.9 and Fig. 4.14), X-Y DOR outperforms the other routing methods, which is reasonable because the uniform random traffic is uniform and suitable for X-Y DOR [4.18]. In the transpose traffic (Fig. 4.10 and Fig.

4.15) and the bit reverse traffic (Fig. 4.11 and Fig. 4.16), the proposed VCPAR yields the lowest latency and exhibits the best tolerance to network congestion (except the transpose for the 16 flit packet size (Fig. 4.15). TXY can avoid congestion on the specific paths in the transpose because it can select the next hop direction randomly from two available directions). This fact demonstrates that preferentially selecting three-cycle latency routers can reduce the packet latency; adaptability based on direction selection with the transmission counter (described in Subsection 4.4.1) alleviates network congestion. The same tendency is observed for hot spot traffic (Figs. 4.12 and 4.13, Figs. 4.17 and 4.18). The proposed VCPAR outperforms the other routing methods in the hot spot traffic. The proposed VCPAR with the VAVCR makes use of process variation and has a low-latency feature even in the network congestion.

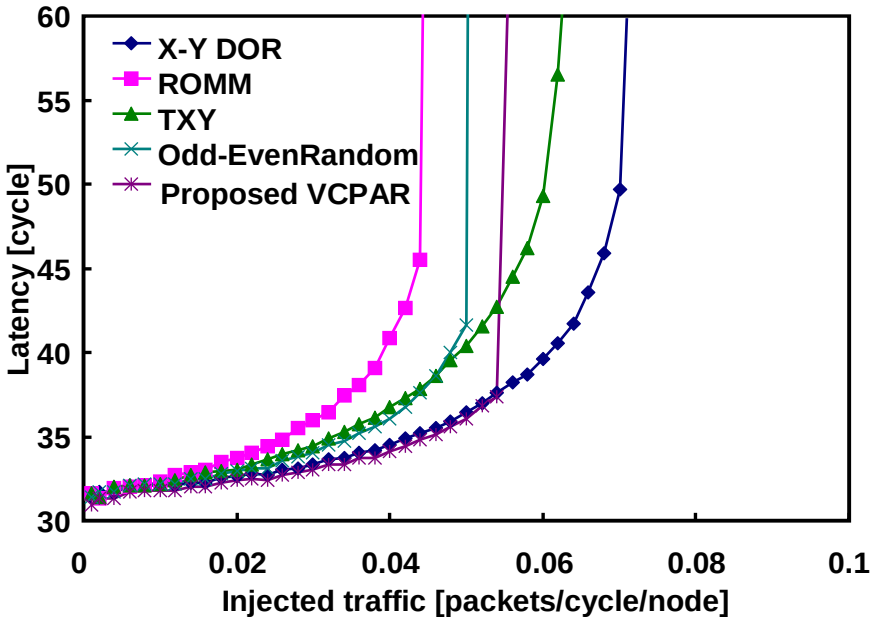


Fig. 4.9 Evaluation results of routing methods (packet size is four flits): uniform random traffic.

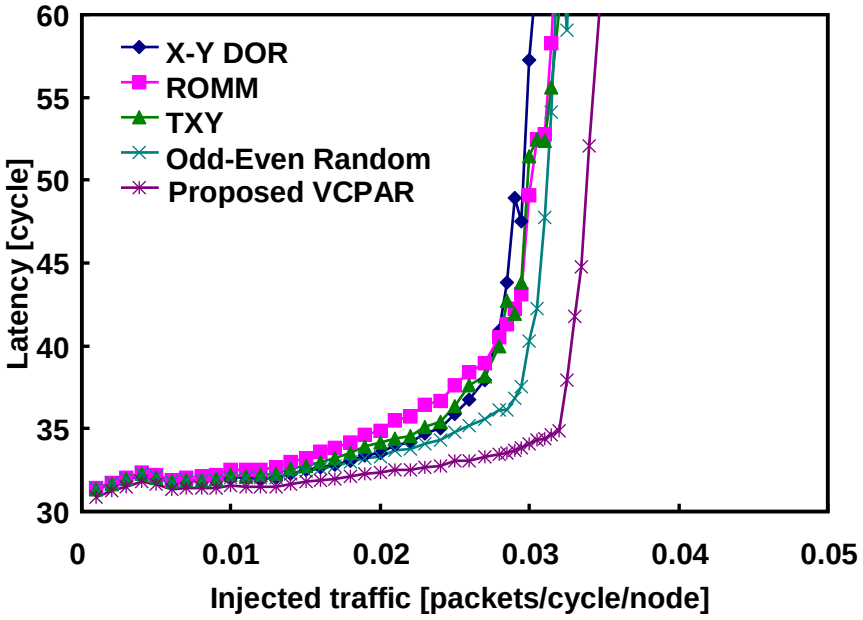


Fig. 4.10 Evaluation results of routing methods (packet size is four flits): transpose traffic.

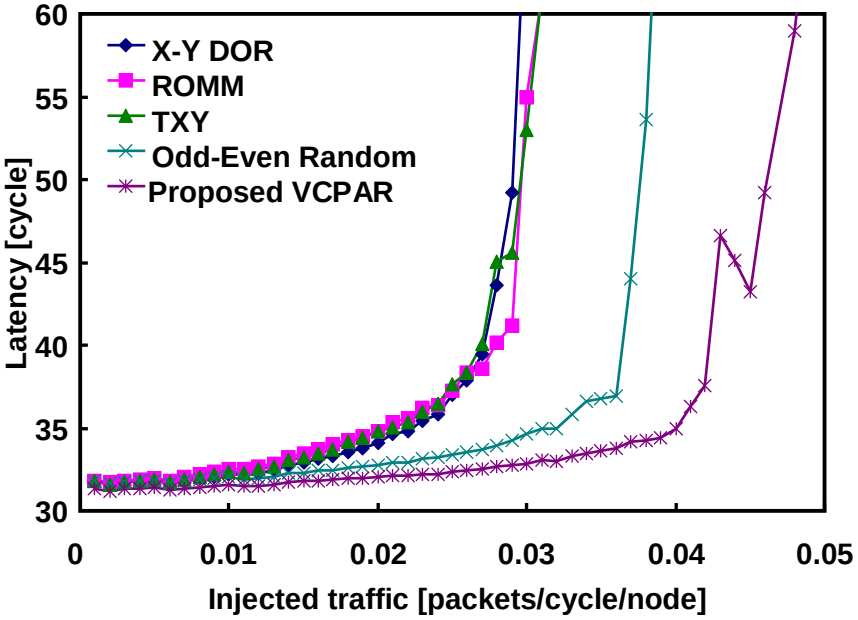


Fig. 4.11 Evaluation results of routing methods (packet size is four flits): bit reverse traffic.

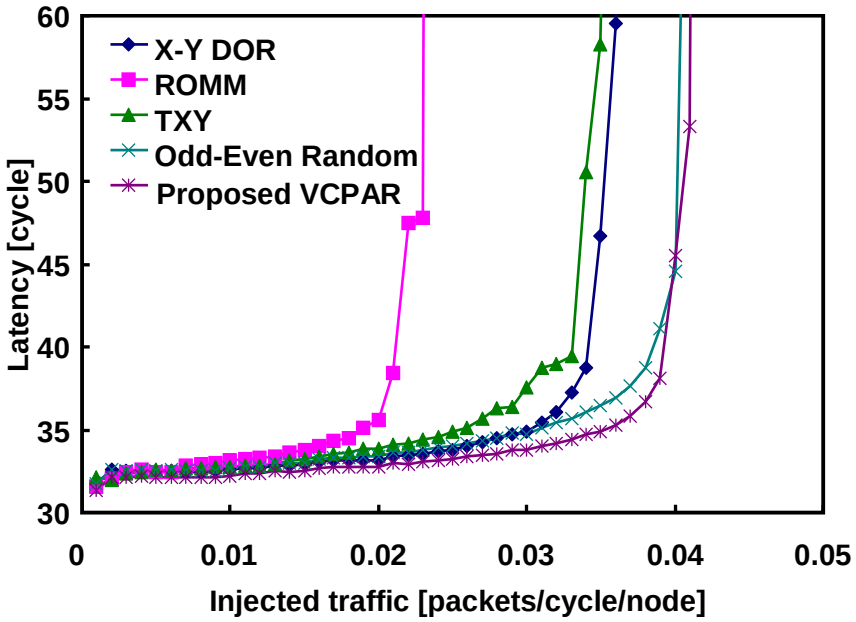


Fig. 4.12 Evaluation results of routing methods (packet size is four flits): hot spot traffic with one hot spot node (the hot spot percentage is 6%).

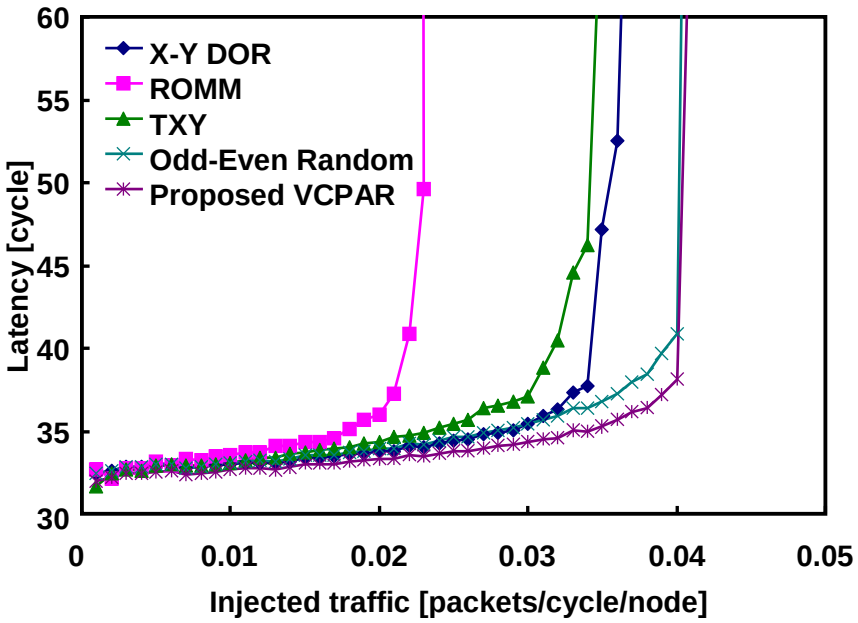


Fig. 4.13 Evaluation results of routing methods (packet size is four flits): hot spot traffic with four hot spot nodes (the hot spot percentage is 6%).

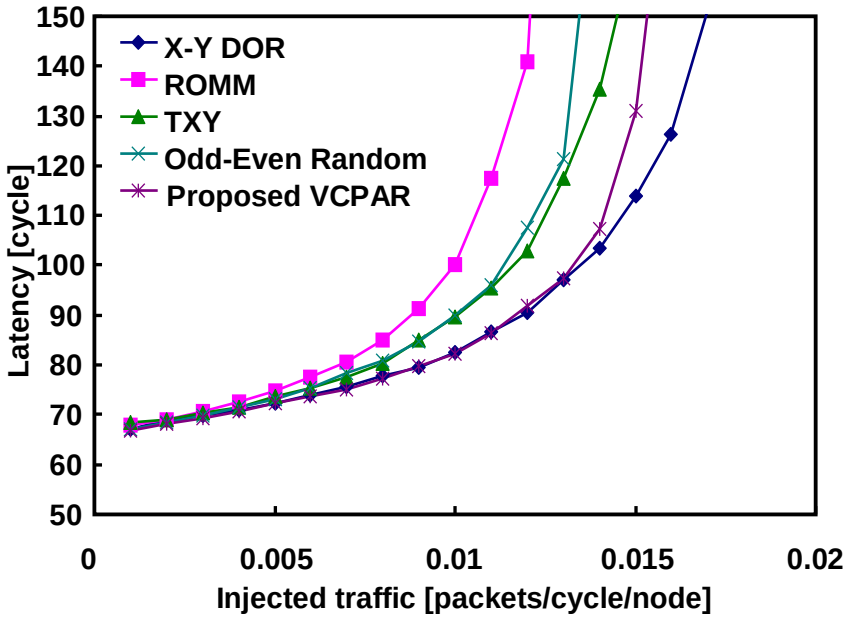


Fig. 4.14 Evaluation results of routing methods (packet size is 16 flits): uniform random traffic.

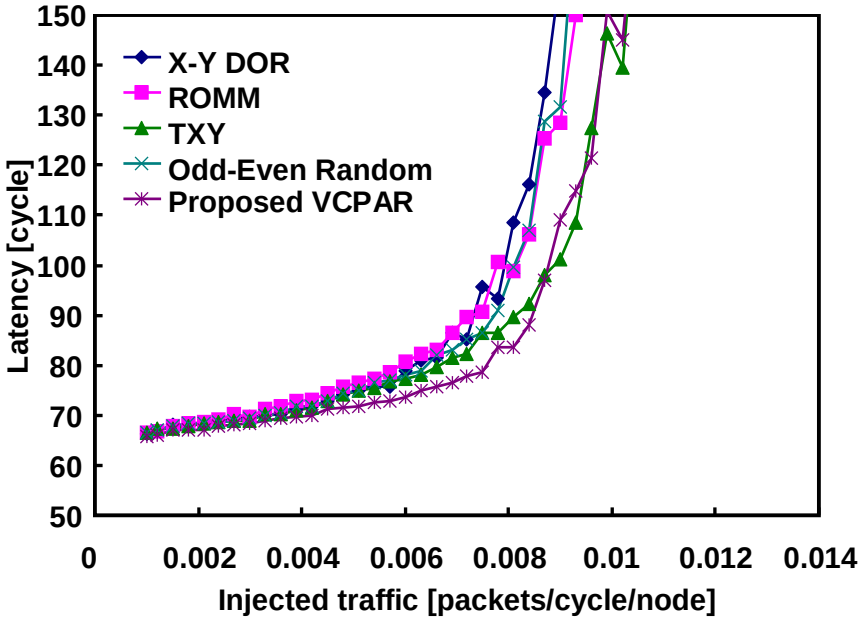


Fig. 4.15 Evaluation results of routing methods (packet size is 16 flits): transpose traffic.

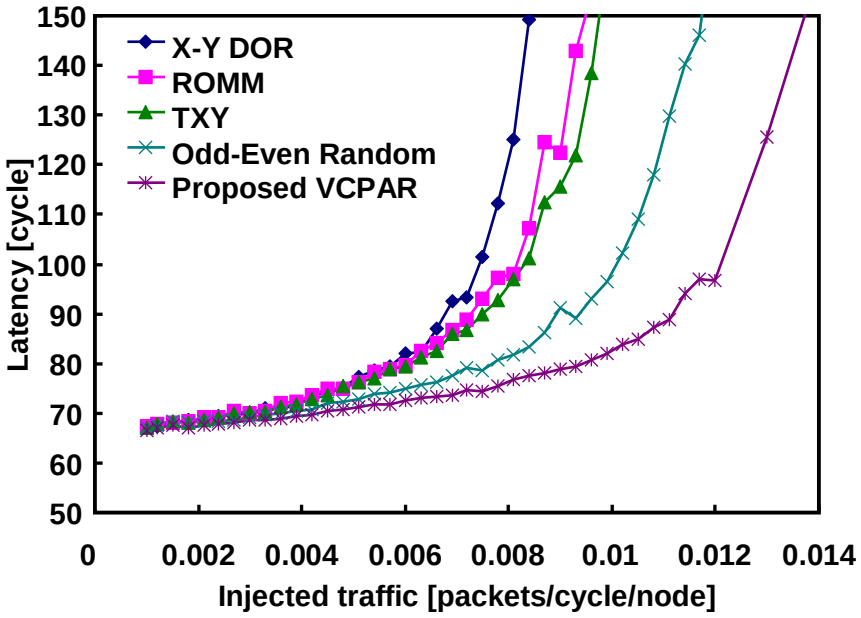


Fig. 4.16 Evaluation results of routing methods (packet size is 16 flits): bit reverse traffic.

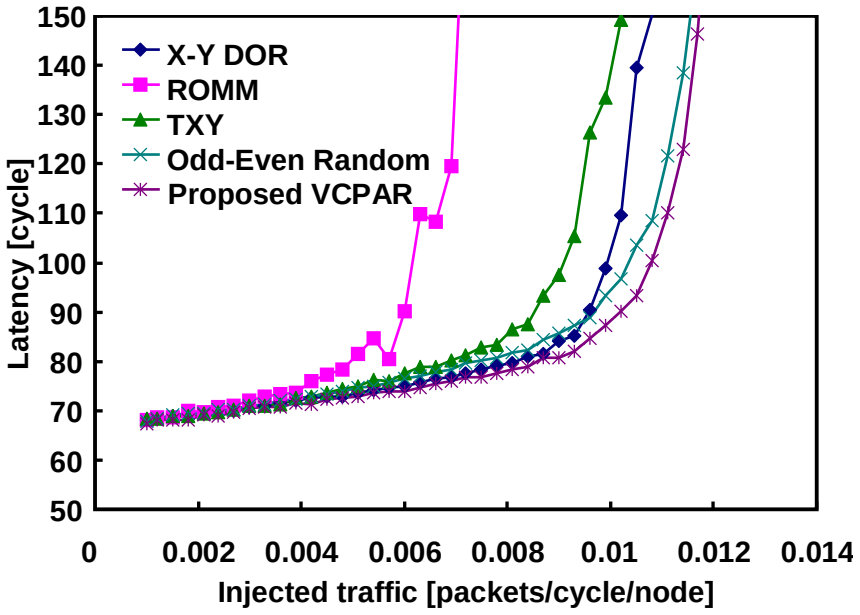


Fig. 4.17 Evaluation results of routing methods (packet size is 16 flits): hot spot traffic with one hot spot node (the hot spot percentage is 6%)

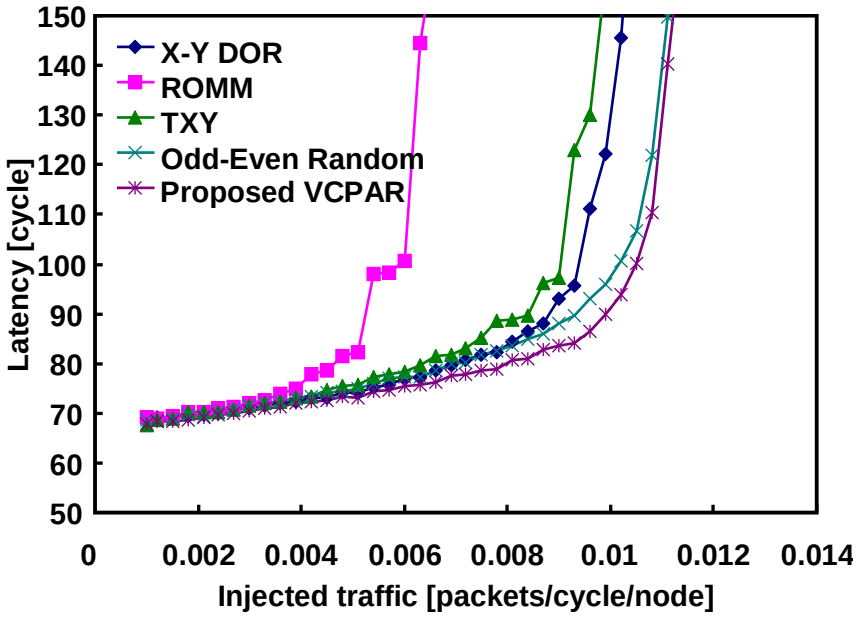


Fig. 4.18 Evaluation results of routing methods (packet size is 16 flits): hot spot traffic with four hot spot nodes (the hot spot percentage is 6%).

4.4.3 Evaluation Methodology of VAVCR with VCPAR

To evaluate the proposed VAVCR with the VCPAR, we use the same methodologies and parameters described in Subsection 4.4.1. In this evaluation, 100 different variation maps (chips) are assessed. The network frequency for the conventional router and proposed VAVCR are 700 MHz and 1,050 MHz, respectively. The conventional router adopts X-Y DOR as a routing method.

In reality, the optimal network frequency for the proposed VAVCR, which maximizes throughput, depends on characteristics of the traffic pattern. In this evaluation, we took 1,050 MHz as $F_{network}$. The detailed discussion of the optimal network frequency will follows in Section 4.5.

As the traffic pattern used in this evaluation, we used the standard task graph set (STG) [4.8] and task graphs for free (TGFF) [4.9]. For the STG, we used random (500 tasks, the task graph number is 0000), robot[1], sparse[2], and fpppp[3]. We set the packet size

[1] STG-robot is a task graph for Newton–Euler dynamic control calculation.

[2] STG-sparse is a task graph for a random sparse matrix solver of an electronic circuit simulation.

[3] STG-fpppp is a task graph for subroutine of SPEC95fp fpppp.

of each edge as 16 ± 8 flits. For TGFF, we set parameters as follows: number of tasks = 500, processing cycle of tasks = 3,000 ± 1,500; and packet size = 32 ± 16 flits [4.10]. Each task in the task graph is assigned to the processor core based on the critical path method [4.23].

4.4.4 Evaluation of VAVCR with VCPAR

Figs. 4.19–4.23 present the evaluation results for the conventional router and the proposed VAVCR with the VCPAR. They signify the total execution times of the task graphs. Each result includes the evaluation of 100 variation maps (chips): The index number is from 0 to 99 in the figures. The execution times are normalized by the average of the conventional router. The dashed and chained lines respectively represent the average of the conventional router and the proposed VAVCR with the VCPAR.

In Fig. 4.19, the proposed VAVCR is shown to reduce the total execution time of the STG-random by 14.6% on average. The packet latency of the STG-random is increased by 31% on average. The execution time is reduced because of the increase in the network frequency. The packet latency of the STG-random is increased because of the existence of the four-cycle routers. Irrespective of the amount of the increase in the packet latency, the proposed VAVCR reduces the total execution time. Similarly, the proposed VAVCR reduces the total execution times of the STG-robot (Fig. 4.20(a)), STG-sparse (Fig. 4.21(a)), STG-fpppp (Fig. 4.22(a)), and TGFF (Fig. 4.23(a)) by 12.3%, 29.3%, 22.1%, and 0.3% on average, respectively. The packet latencies of the STG-robot (Fig. 4.20(b)), STG-sparse (Fig. 4.21(b)), STG-fpppp (Fig. 4.22(b)), and TGFF (Fig. 4.23(b)) were increased by 17.5%, 8.6%, 11.1%, and 32.5% on average, respectively.

The proposed VAVCR can efficiently reduce the total execution time necessary for executing network-bound tasks such as STG-random, STG-sparse, and STG-fpppp. In contrast, the proposed VAVCR reduces it inefficiently when executing computation-bound tasks such as TGFF.

Table 4.5 presents a summary of the reductions of the total execution time, the increases in the packet latency, the standard deviations of the total execution times in the conventional router, and the proposed VAVCR with the VCPAR, the standard deviations of the packet latencies in the conventional router, and the proposed VAVCR

with the VCPAR. They are 15.7%, 20.2%, 2.85%, 2.79%, 3.29%, and 4.87% on average of the five task graphs (TGs), respectively.

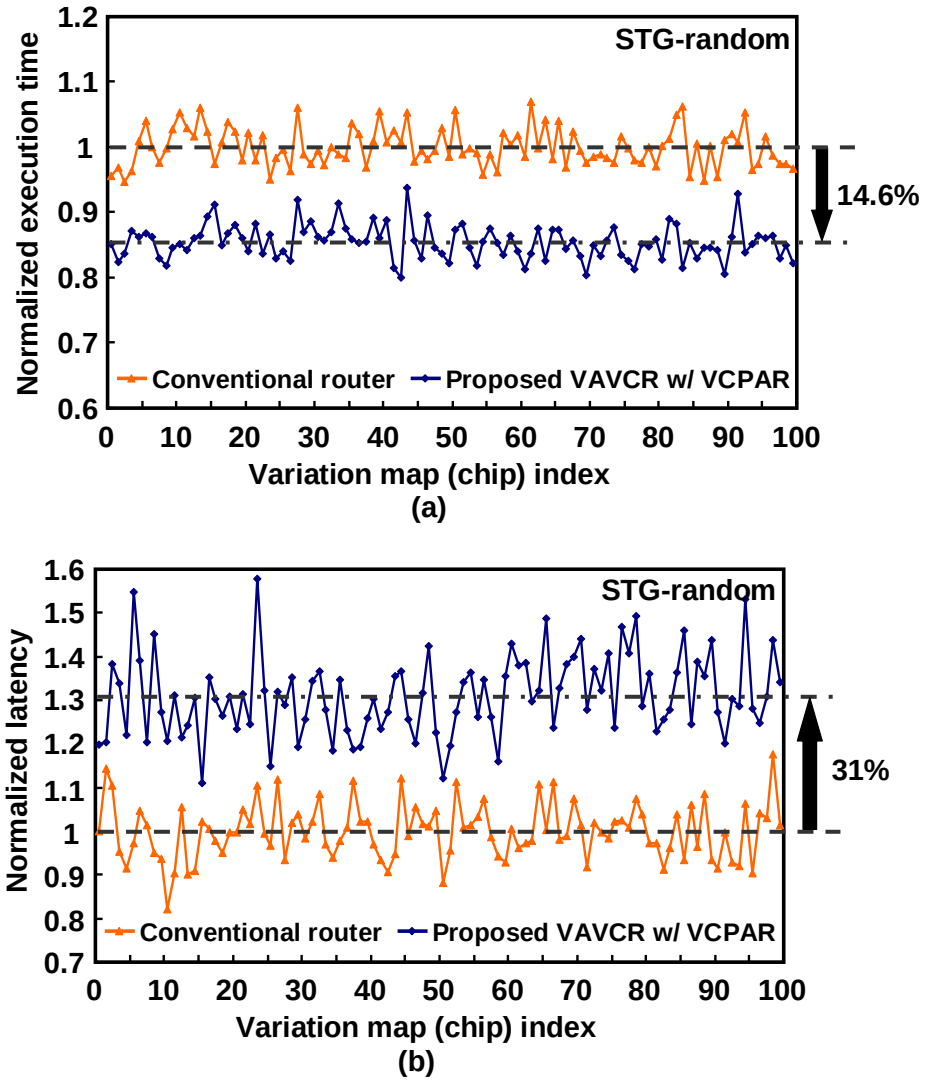


Fig. 4.19 Evaluation results of STG-random: (a) normalized execution time and (b) normalized latency.

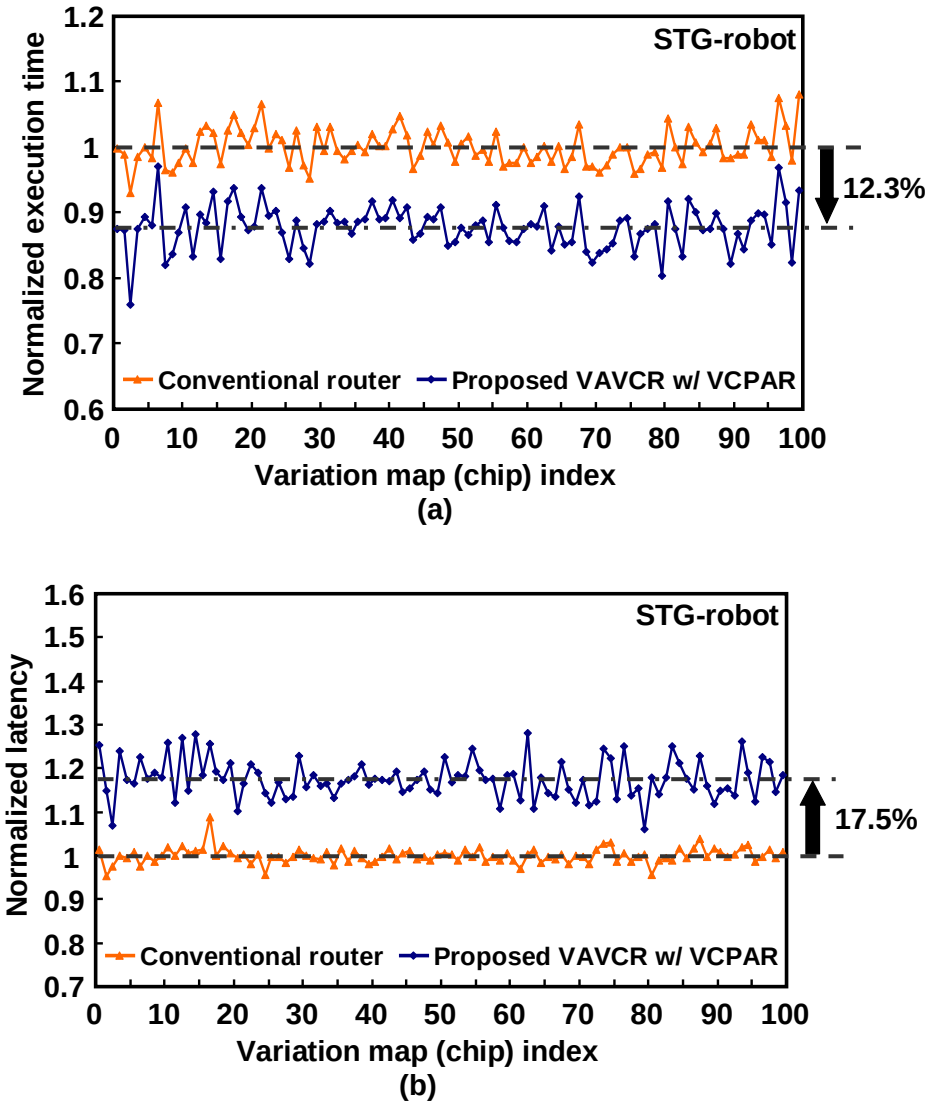


Fig. 4.20 Evaluation results of STG-robot: (a) normalized execution time and (b) normalized latency.

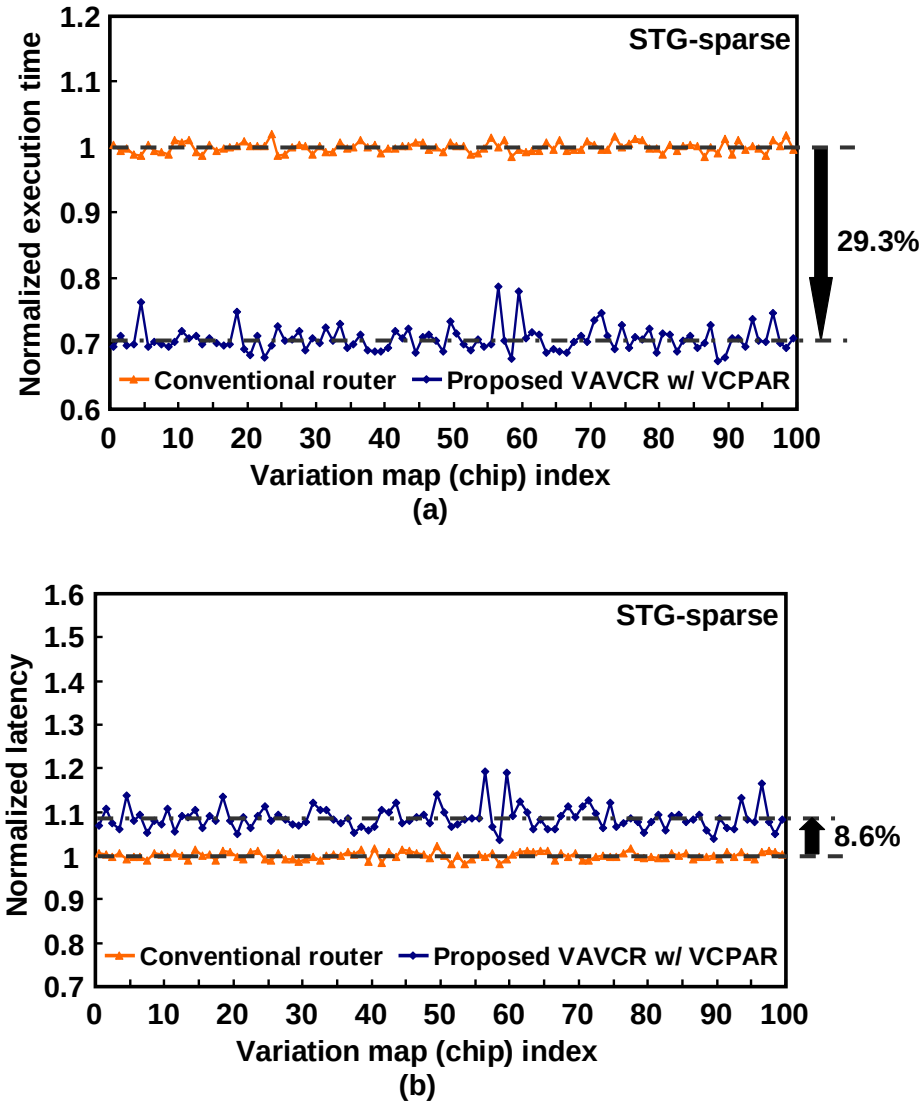


Fig. 4.21 Evaluation results of STG-sparse: (a) normalized execution time and (b) normalized latency.

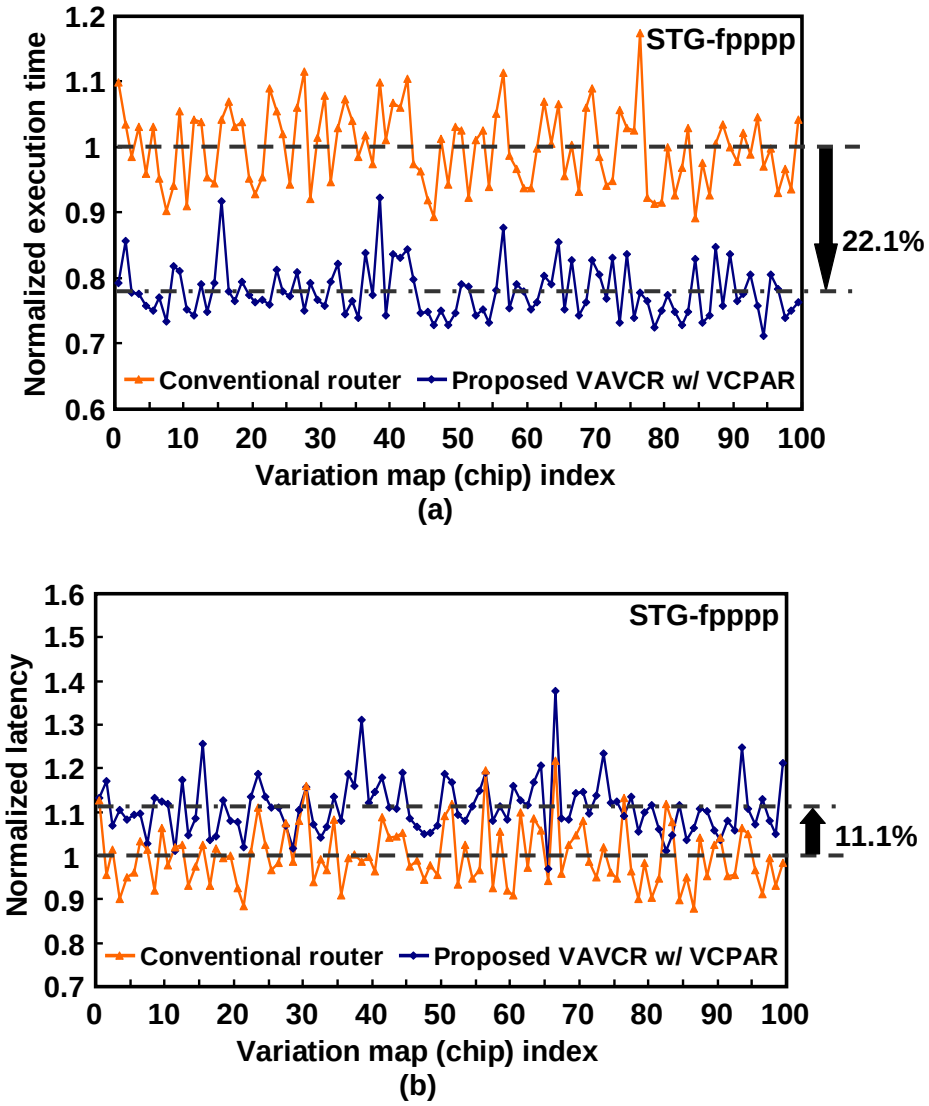


Fig. 4.22 Evaluation results of STG-fpppp: (a) normalized execution time and (b) normalized latency.

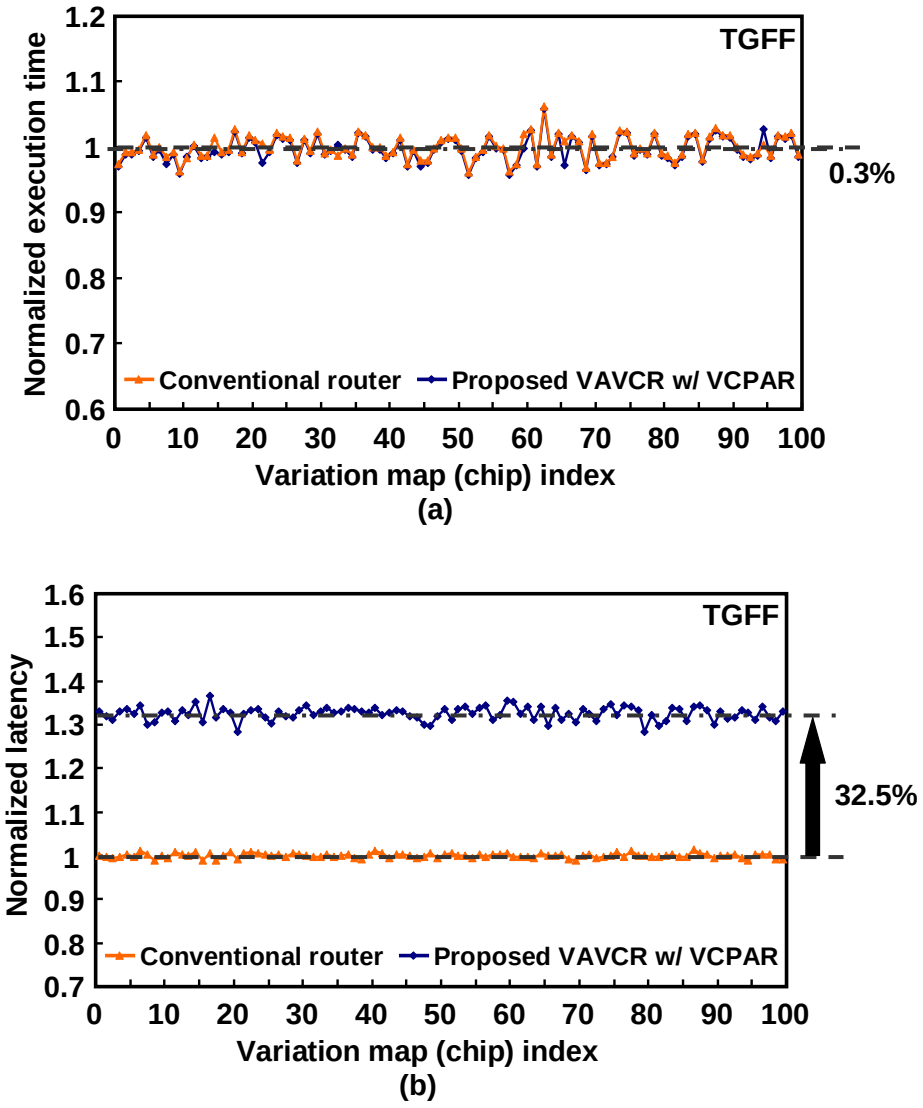


Fig. 4.23 Evaluation results of TGFF: (a) normalized execution time and (b) normalized latency.

4.4.5 Area overhead of the VAVCR w/ VCPAR

To estimate the area overhead of the proposed VAVCR with the VCPAR, its transistor count is to be compared with that of the conventional router. The transistor count of the conventional router and the proposed VAVCR with the VCPAR are 618.1 k and 629.1 k, respectively. The area overhead of the proposed VAVCR with the VCPAR is 1.78% as a single router, which implies that the total area overhead will turn out almost negligible because a router portion is much smaller than a processor portion in an NoC.

4.5 Discussion on the network frequency optimization

In this section, the optimization of the network frequency is discussed. The execution time and packet latency depends on $F_{network}$. Figs. 4.24(a) and 4.24(b) show the reduction of the total execution time and increase in the packet latency when $F_{network}$ is varied. 100 variation maps are again utilized as well as in Section 4.4. "Static" in the figure means "full use of the network", in which a single packet has 16 flits. We utilize "Static" as a reference to be compared with the other five task graphs.

The reductions in the total execution times of the STG-random and STG-fpppp monotonically increase with $F_{network}$ because they do not incur network congestion; a faster $F_{network}$ is better in these cases. The STG-random presents degradation of the execution time at a network frequency of 850 MHz or less because its traffic is uniform and thus X-Y DOR is eligible for it (we have already discussed this point in Subsection 4.4.2). In contrast, the STG-sparse and STG-robot that incur network congestion have local maximums at 1,100 MHz and 950 MHz, respectively, in terms of reduction in the execution time. It is noteworthy that they have similar shapes to "Static" that fully use the network. The TGFF is not affected by $F_{network}$ at all because it is a compute-bound task; the computation occupies over 99% of the total execution cycles.

From this discussion, an appropriate $F_{network}$ should be set by designers based on characteristics of a traffic pattern.

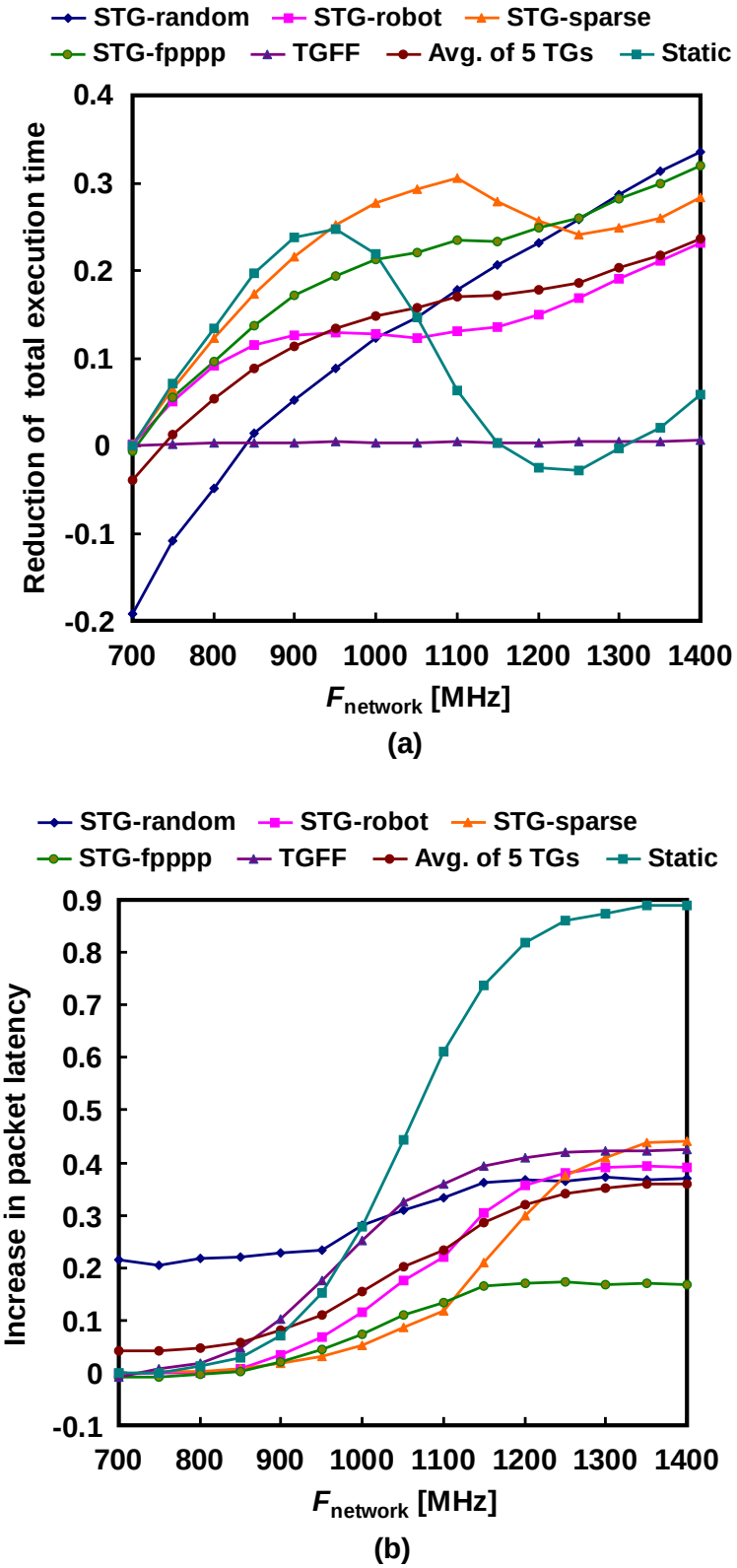


Fig. 4.24 (a) reduction of the total execution time versus Fnetwork (averaged by 100 variation maps) and (b) increase in the packet latency versus Fnetwork (averaged by 100 variation maps).

4.6 Summary

As described in this chapter, we proposed a process-variation-adaptive NoC with a variation-adaptive variable-cycle router (VAVCR) and a variable-cycle pipeline adaptive routing method (VCPAR). The proposed VAVCR can configure its cycle latency adaptively corresponding to the spatial process variation. It increases the network frequency, which is limited by the slowest network component in the conventional router. The proposed VAVCR can reduce the total execution time by 15.7% based on an average of the five task graphs at a network frequency of 1,050 MHz. The proposed VCPAR can reduce packet latencies in the NoC adaptively with variable cycle router and can efficiently suppress network congestion.

4.7 References

[4.1] L. Benini and G. De Micheli, "Networks on chips: a new SoC paradigm", IEEE Computer, vol. 35, no. 1, pp. 70-78, Jan. 2002.

[4.2] R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas, "Mitigating Parameter Variation with Dynamic Fine-Grain Body Biasing", Proceedings of ACM/IEEE Int. Symp. on Microarchitecture, pp. 27-42, 2007.

[4.3] S. Dighe, S.R. Vangal, P. Aseron, S. Kumar, T. Jacob, K.A. Bowman, J. Howard, J. Tschanz, V. Erraguntla, N. Borkar, V.K. De, and S. Borkar, "Within-Die Variation-Aware Dynamic-Voltage-Frequency-Scaling With Optimal Core Allocation and Thread Hopping for the 80-Core TeraFLOPS Processor", IEEE J. of Solid-State Circuits, vol. 46, no. 1, pp. 184-193, Jan. 2011.

[4.4] S.R. Sarangi, B. Greskamp, R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas, "VARIUS: A Model of Process Variation and Resulting Timing Errors for Microarchitects", IEEE Trans. on Semiconductor Manufacturing, vol. 21, no. 1, pp. 3-13, Feb. 2008.

[4.5] T. Tsunomura, A. Nishida, and T. Hiramoto, "Analysis of NMOS and PMOS Difference in VT Variation with Large-Scale DMA-TEG", IEEE Trans. on Electron Devices, vol. 56, no. 9, pp. 2073-2080, Sept. 2009.

[4.6] Stanford Univ. – Open Source Network-on-Chip Router RTL https://nocs.stanford.edu/

[4.7] Booksim http://sourceforge.net/projects/booksim/

[4.8] T. Tobita and H. Kasahara, “A standard task graph set for fair evaluation of multiprocessor scheduling algorithms”, J. of Scheduling, vol. 5, pp. 379-394, Sep. 2002.

[4.9] R. P. Dick, D. L. Rhodes, and W. Wolf, “TGFF: task graphs for free”, Proceedings of IEEE Int. Workshop on Hardware/Software Codesign, pp. 97-101, March 1998.

[4.10] J. Chan and S. Parameswaran, “NoCEE: Energy Macro-Model Extraction Methodology for Network on Chip Routers”, Proceedings of IEEE Int. Conf. on Computer-Aided Design, pp. 254-259, Nov. 2005.

[4.11] R. Teodorescu and J. Torrellas, “Variation-Aware Application Scheduling and Power Management for Chip Multiprocessors”, Proceedings of ACM/IEEE Int. Symp. on Computer Architecture, pp. 363-374, 2008.

[4.12] W. Dally, and B. Towles, “Principles and Practices of Interconnection Networks”, Morgan Kaufmann, 2004.

[4.13] A. Kumary, P. Kunduz, A.P. Singhx, L.-S. Pehy, and N.K. Jhay, "A 4.6 Tbits/s 3.6 GHz single-cycle NoC router with a novel switch allocator in 65 nm CMOS”, Proceedings of IEEE Int. Conf. on Computer Design, pp. 63-70, 2007.

[4.14] P. Friedberg, Y. Cao, J. Cain, R. Wang, J. Rabaey, and C. Spanos, "Modeling within-die spatial correlation effects for process-design co-optimization”, Proceedings of IEEE Int. Symp. on Quality of Electronic Design, pp. 516-521, 2005.

[4.15] J. Howard, S. Dighe, S.R. Vangal, G. Ruhl, N. Borkar, S. Jain, V. Erraguntla, M. Konow, M. Riepen, M. Gries, G. Droege, T. Lund-Larsen, S. Steibl, S. Borkar, V.K. De, and R. Van Der Wijngaart, "A 48-Core IA-32 Processor in 45 nm CMOS Using On-Die Message-Passing and DVFS for Performance and Power Scaling”, IEEE J. of Solid-State Circuits, vol. 46, no. 1, pp. 173-183, Jan. 2011.

[4.16] D.N. Truong, W.H. Cheng, T. Mohsenin, Yu Zhiyi, A.T. Jacobson, G. Landge, M.J. Meeuwsen, C. Watnik, A.T. Tran, Xiao Zhibin, E.W. Work, J.W. Webb, P.V. Mejia, and B.M. Baas, "A 167-Processor Computational Platform in 65 nm CMOS”, IEEE J. of Solid-State Circuits, vol. 44, no. 4, pp. 1130-1144, 2009.

[4.17] Y. Nakata, Y. Takeuchi, H. Kawaguchi, and M. Yoshimoto, "A Process-Variation-Adaptive Network-on-Chip with Variable-Cycle Routers”, Proceedings of Euromicro Conference on Digital System Design, pp. 801-804, Aug. 2011.

[4.18] Ge-Ming Chiu, “The odd–even turn model for adaptive routing”, IEEE Trans. on Parallel and Distributed Systems, vol. 11, pp. 729-738, July 2000.

[4.19] W.J. Dally and C.L. Seitz, “Deadlock-Free Message Routing in Multiprocessor Interconnection Networks”, IEEE Trans. on Computers, Vol. C-36, pp. 547-553, May

1987.

[4.20] T. Nesson and S.L. Johnsson, “ROMM Routing: A Class of Efficient Minimal Routing Algorithms”, Proceedings of Int. Workshop on Parallel Computer Routing and Communication, pp. 185-199, 1994.

[4.21] D. Seo, A. Ali, W.T. Lim, and N. Rafique, “Near-optimal worst-case throughput routing for two-dimensional mesh networks”, Proceedings of ACM/IEEE Int. Symp. on Computer Architecture, pp. 432-443, June 2005.

[4.22] M.L. Fulgham and L. Snyder, “Performance of Chaos and Oblivious Routers under Non-Uniform Traffic”, Tech. Report UW-CSE-93-06-01, Univ. of Washington, July 1993.

[4.23] E.G. Coffman, “Computer and Job-shop Scheduling Theory”, John Wiley & Sons, 1976.

Chapter 5 System-Level Fault-Injection Scheme and Evaluation of Dependable Processor

In this chapter, a fault-injection system (FIS) that can inject faults such as read/write margin failures and soft errors into a SRAM environment is proposed. The fault case generator (FCG) generates time-series SRAM failures in 7T/14T or 6T SRAM, and the proposed device model and fault-injection flow is applicable for system-level verification. For evaluation, an abnormal termination rate in vehicle engine control was adopted. It was confirmed that the vehicle engine control system with the 7T/14T SRAM improves system-level dependability compared with the conventional 6T SRAM.

5.1 Introduction

We propose a novel fault-injection scheme using physical characteristics of the SRAM for the system-level verification. In addition, a SRAM fault-injection flow from the device level to the system level is introduced: The proposed fault-injection system can evaluate SRAM reliability in terms of operating stability for a system LSI. Large-scale verification considering the random process variation of each physical LSI can be performed by the proposed fault-injection system.

This chapter is organized as follows. Section 5.2 introduces the overview of the proposed fault-injection system. In Section 5.3, the detail of modeling of the SRAM behavior and nature is described. In Section 5.4, bit error rates of the 7T/14T SRAM are presented. Vehicle control system level evaluations including system error rate evaluation and its relation to SRAM bit-error rate are evaluated in Section 5.5. Finally, Section 5.6 concludes this chapter.

5.2 Fault-Injection System

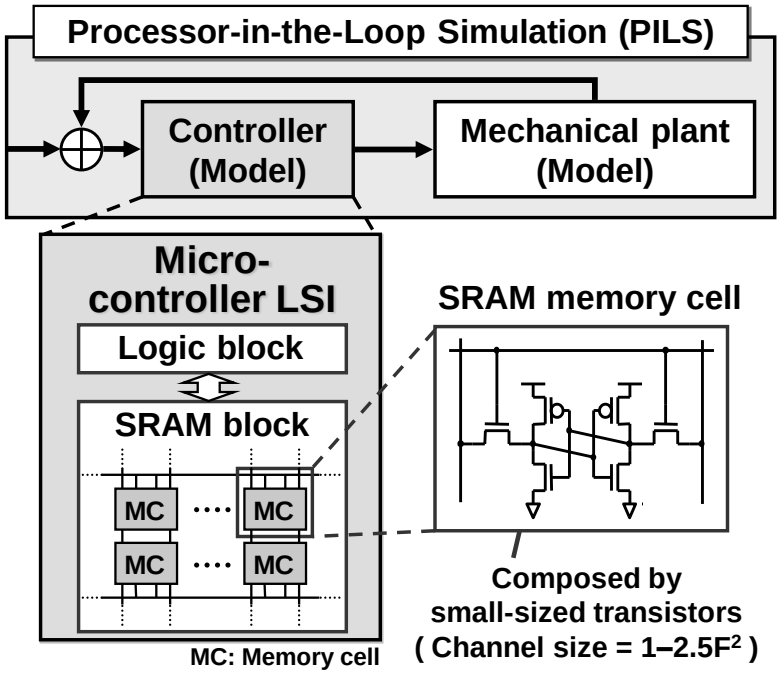


Fig. 5.1 Overview of processor-in-the-loop simulation (PILS) and a simple diagram of a controller LSI composed of a logic block and SRAM block.

Fig. 5.1 shows an overview of a processor-in-the-loop simulation (PILS) and a simple diagram of the controller LSI composed of a logic block and SRAM block.

The PILS can provide information on hardware features and perform high-accuracy simulation in a prototype system; it tests actual control software running on a dedicated processor with the virtual prototype of the mechanical plant.

The increase in minimum operation voltage (V_{min}) on an LSI degrades its device reliability due to power supply noise, IR drops (voltage drop caused by current × resistance), and/or soft errors. V_{min} on the entire micro-controller, including the logic block and SRAM block, is determined by the circuit with the highest value of V_{min} [5.1]. SRAM has a larger standard deviation for the threshold voltage than the logic block because its transistor size is smaller. To make matters worse, the SRAM capacity on the micro-controller is huge. Consequently, large SRAM blocks such as the cache memory or internal local memory determine V_{min} on the micro-controller.

Fig. 5.2 shows an overall view of the proposed fault-injection system (FIS). The FIS

integrates a system-level verification environment and the fault-injection scheme.

In this study, we handled an electric control unit (ECU) system for vehicle engine control that consists of a vehicle engine with sensors/actuators and the ECU with an SH-2A processor; it can simulate engine revolution control. The mechanical system including the engine, sensors, and actuators is emulated by MATLAB®/Simulink®[4]. The SH-2A processor is emulated by CoMET™ [5].

As shown in Fig. 5.2, the fault-injection scheme can inject failures based on a precalculated bit error rate (BER) into the internal. Several various failure modes are supported as described in the next section. The fault-injectable bus bridge (FIB) is allocated between the SH-2A core and internal SRAM in the micro-controller; it arbitrates a normal access and false access (injected failure). The FIB intervenes in the memory transactions to destroy access data to the internal SRAM and switches to the failure data pattern when a failure occurs.

The fault case generator (FCG) uses various device parameters such as a supply voltage, temperature, aging, etc.; it generates time-series failure data patterns according to the parameters. The time-series failure data patterns are stored once in the FIB, which then injects the failure data into the memory transactions when accessing the failure address.

5.3 Modeling of Failures in SRAM

In this section, the proposed method for modeling the SRAM failure and implementing the FCG are described in detail. By injecting SRAM's physical behavior from the device level to the system level, the proposed model can reflect the SRAM well as an actual silicon chip.

First, SRAM failures and their behaviors at the device level are described in Subsections 5.3.1 and 5.3.2, respectively. Subsequently, the proposed fault injection flow is described in Subsection 5.3.3 Next, a modeling method for the SRAM behavior at the device level is presented in Subsection 5.3.4. Then, the FCG that generates failure memory data patterns is stated in Subsection 5.3.5. Details of virtual chips and failure

[4] MATLAB®/Simulink® is a registered trademark of The MathWorks, Inc.

[5] CoMET™ (currently Virtualizer™) is a registered trademark of Synopsys®, Inc.

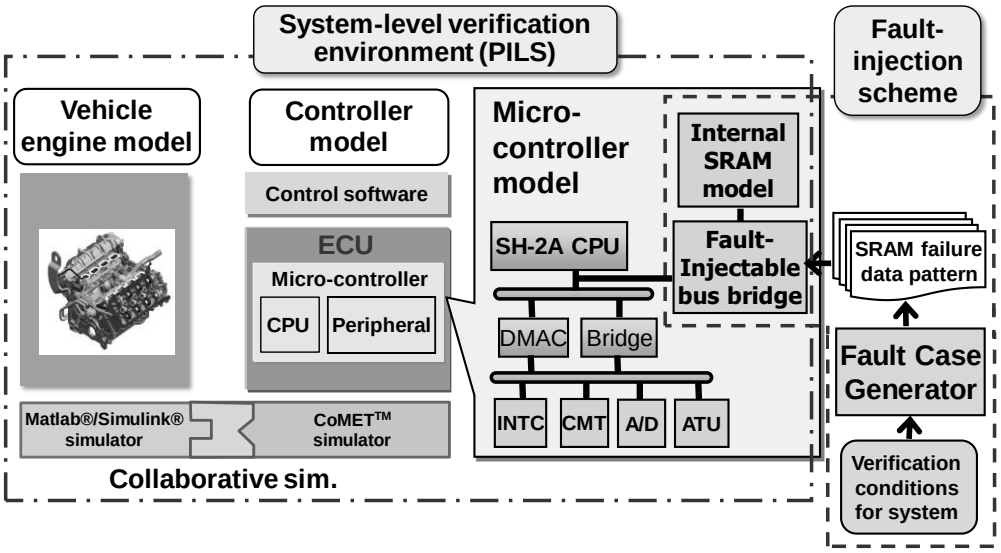


Fig. 5.2 Proposed system-level verification environment has a vehicle engine model and controller (ECU) model in which a micro-controller model is included. Faults are injected through a fault-injectable bus bridge on an SH-2A CPU.

data pattern generation flow are described in Subsection 5.3.6. The final section concludes the chapter.

5.3.1 Failures in SRAM

Failures in SRAM are categorized as read margin failure, write margin failure, soft error, and access time violation as described in Subsection 3.3.1. The read and write margin failures, and soft error are considered in this chapter because they are dominant at low operating frequencies.

5.3.2 Behavior of SRAM failures on a device level

To inject the SRAM failure and estimate the system-level verification, modeling the SRAM failures are necessary. Fig. 5.3 shows the failure pattern examples of the read/write margin failure and soft error; the models in the figure are derived from physical SRAM behaviors.

The read margin failure emerges as a destructive readout; the stored datum in a memory cell flips when the datum with no read margin is read out. The failure (flipped

datum) lasts until it is rewritten.

The write margin failure occurs when there is an attempt to write a memory cell with no write margin. In the write operation, the memory cell with no write margin cell does not flip to the write datum. This failure lasts until the flipped memory cell is normally written, similar to the read margin failure.

The read/write margin failure is mainly caused by process variations including random and systematic variations, aging of the transistor device, and fluctuations in the supply voltage and temperature. In addition, the read/write margin failure has datum dependence: either “0” failure or “1” failure for each memory cell. It is determined by the random variation of transistors in every SRAM memory cell.

The soft error is modeled as a temporarily failure; a datum stored in a memory cell suddenly flips. The failure also lasts until it is rewritten.

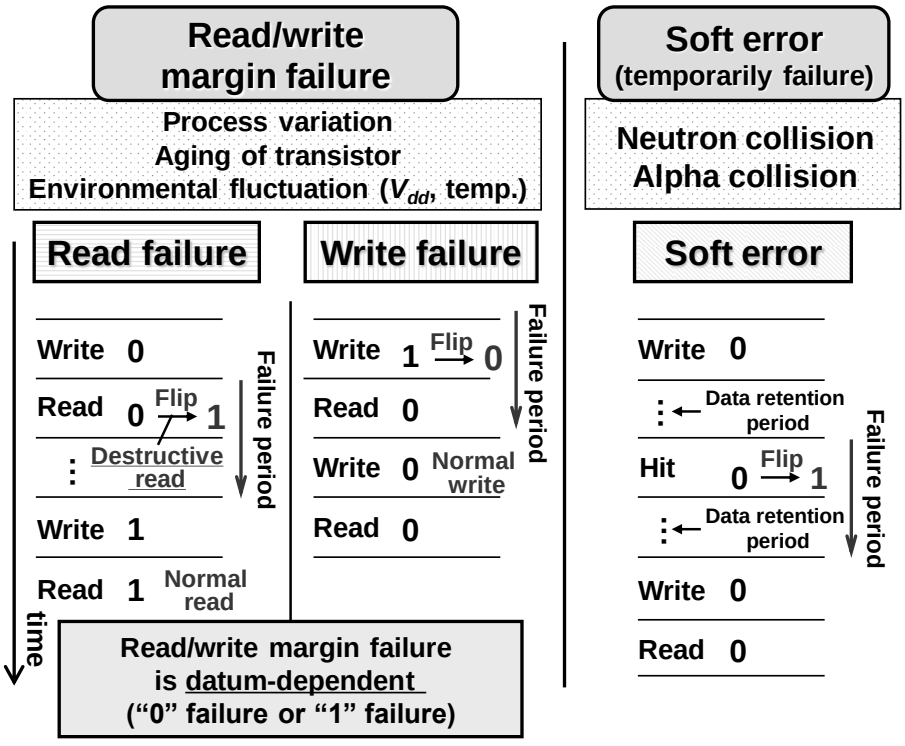


Fig. 5.3 Failure pattern examples in SRAM memory cell: read margin failure, write margin failure, and soft error.

5.3.3 Proposed Fault-Injection Flow for System-Level Verification

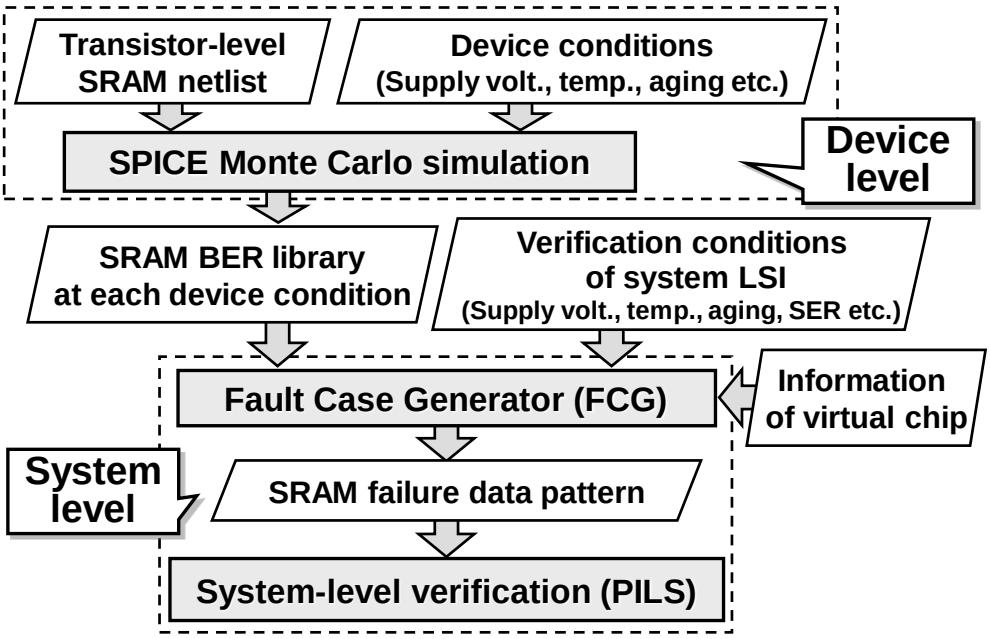


Fig. 5.4 Proposed fault injection scheme flowing from a device level to a system level.

Fig. 5.4 portrays the proposed fault-injection flow for the system-level verification, which starts at the device level and ends at the system level. First, on the device level, SPICE Monte Carlo simulations using a transistor-level SRAM netlist are conducted considering various device parameters. In the following subsection, we mention the device parameter. As a result of the Monte Carlo simulations, an SRAM BER library including BERs on various device conditions is obtained. Next, the generated SRAM BER library, the verification condition under which a system LSI designer wants to verify, and information of the virtual chip are used as inputs to the FCG. The virtual chip has information about failure addresses, which are described in detail in the next subsection. Eventually, the FCG calculates and outputs SRAM failure data patterns, which are fed to the PILS as system-level verification.

In this way, the device-level behavior of the SRAM is injected into the system-level verification environment. If another kind of SRAM must be evaluated on a system level, it can achieved by creating a new SRAM BER library, and the same fault injection flow is then carried out.

5.3.4 Modeling Failures for System-Level Fault Injection

In this subsection, the modeling method for generating SRAM failures is proposed. Fig. 5.5 shows the basic concept of the virtual chip. In an actual silicon chip, read/write margin failures and soft errors are randomly distributed across the chip because of the random variation derived from transistor physics. The datum-dependence of the read/write margin failure is also determined randomly by the random variation.

In other words, the virtual chip can reproduce the features on an actual silicon chip and thus has repeatability. The failure addresses are determined to be random spatially. The datum-dependences of the read/write margin failure are randomly determined as "0" or "1". The largest advantage of using the virtual chip is the large-scale verification capability. Fig. 5.6 shows an example of a large-scale verification using 10,000 virtual chips. Each virtual chip has different addresses of failures and thus different reliabilities. The failure addresses may make the virtual chip fail or sometimes not. The FIS with the virtual chip concept can easily perform large-scale verification using a large number of virtual chips without a large number of actual chip samples.

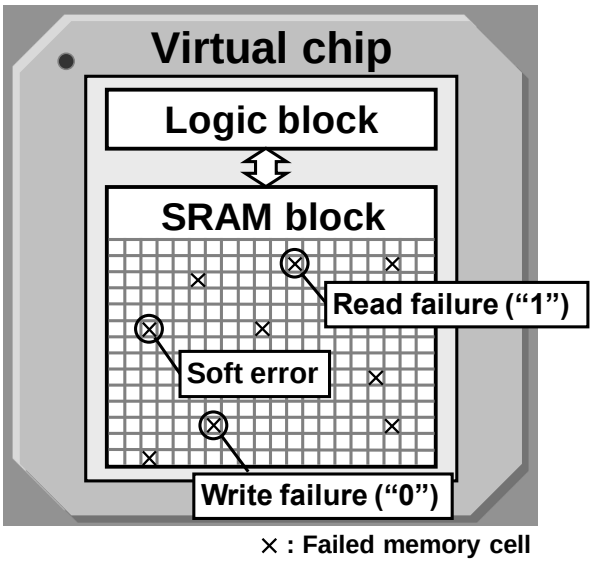


Fig. 5.5 Virtual chip: SRAM failures are randomly distributed across a chip. The data-dependence is also randomized as "0" or "1".

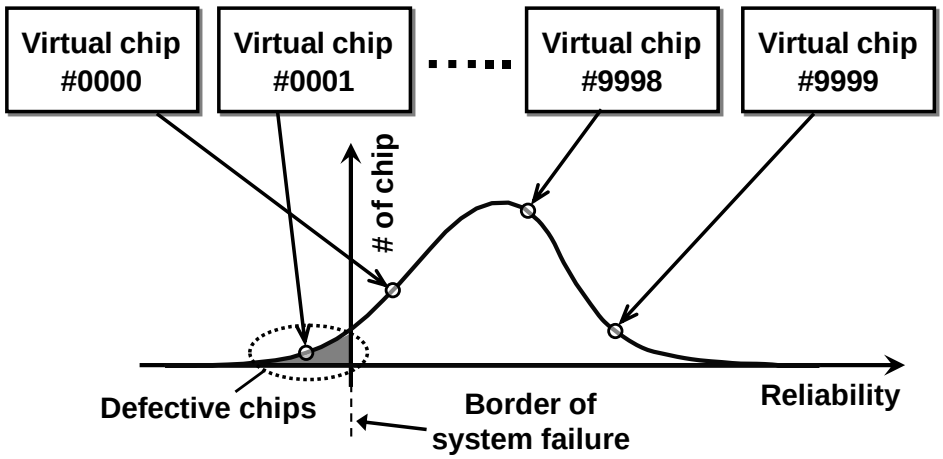


Fig. 5.6 Large-scale verification using the 10,000 virtual chips.

5.3.5 Fault Case Generator

Fig. 5.7 shows a block diagram of the FCG. The FCG generates time-series memory failure data patterns as outputs; device parameters can be input to it, including the supply voltages, operating temperature, process variation (standard deviation of threshold voltage σ_{Vth}), aging in the PMOS transistor (decrease in threshold voltage ΔV_{th}), soft error rate, SRAM capacity, information of a virtual chip, and BER library obtained by SPICE Monte Carlo simulations. The supply voltage and operating temperature are time-series parameters; the others are fixed. Arbitrary waveforms for the power supply noise and operating temperature can be used as inputs to the FCG.

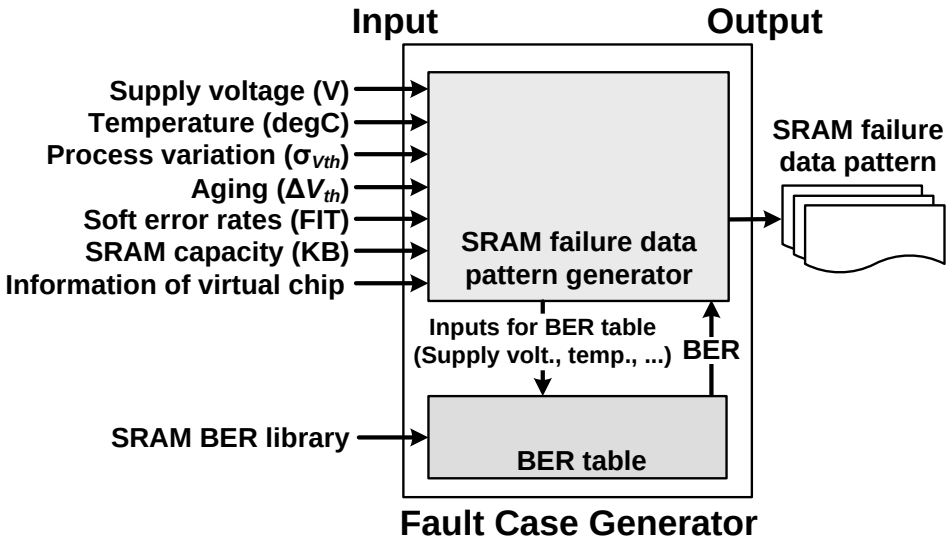


Fig. 5.7 Fault Case Generator.

After receiving inputs, the FCG stores the SRAM BER library in the BER table, and the BER table queries a BER that corresponds to the input device parameters. The SRAM failure data pattern generator generates time-series failure data patterns based on the BER coming from the BER table. The read/write margin failures and soft errors are generated at random addresses.

5.3.6 Failure data pattern generation in FCG

Table 5.1 shows read failure examples in a virtual chip for a 128-KByte SRAM. A virtual chip holds the descending orders (ranks) of read and write margins of memory cells, which indicate memory cell weaknesses for the read and write operations. By ranking margins of memory cells, it can reproduce the feature in an actual silicon chip that memory cells fail in the weak order. Each entry (row) includes a descending order of the cell margin, a bit address, and a data-dependence of the read/write margin failure. Each virtual chip can store 10% entries of the total addresses (i.e., for 128-KByte (1-Mbit) SRAM, the number of entries is 104,858 (see Table 5.1)).

Fig. 5.8 illustrates the failure data pattern generation flow conducted in the FCG. First, a supply voltage and temperature are read as time-series change information. A BER is obtained by querying the BER library obtained by SPICE Monte Carlo simulations using the supply voltage, temperature, and other device condition (process variation,

aging parameter, and soft error rate). Next, the number of the no read/write margin cells (N) is calculated by using the BER obtained in the previous procedure and SRAM capacity. Then, the failure data patterns are generated by N, information of virtual chip, and an address offset which specifies a base address of a scratch pad memory (SRAM) determined by the address assignment. The failure data patterns include the failures from the entries from “1” to “N” in Table 1. Next, we assess whether the time-series change information of the supply voltage and temperature is remained or not. If it is remained, then the next information (i.e. the next changing point of the supply voltage or temperature) of the time-series change information of supply voltage and temperature are read; then the SRAM data pattern generation flow is repeated. The time-series SRAM failure data patterns are updated for new supply voltage and temperature at the next changing point. Thus, the proposed SRAM failure data pattern generation in the FCG can generate the SRAM failure data pattern as arbitrary waveforms for the power supply noise and operating temperature.

Table 5.1 Read failure information example of a virtual chip for 128-KB SRAM

Descending order of the cell margin	Bit address	Data-dependence
1	00086DEB	"0"
2	00002EEB	"1"
3	0006C4CE	"1"
4	000A42CA	"0"
5	00047273	"1"
⋮	⋮	⋮
104,857	0008361A	"0"
104,858	000E67CC	"0"

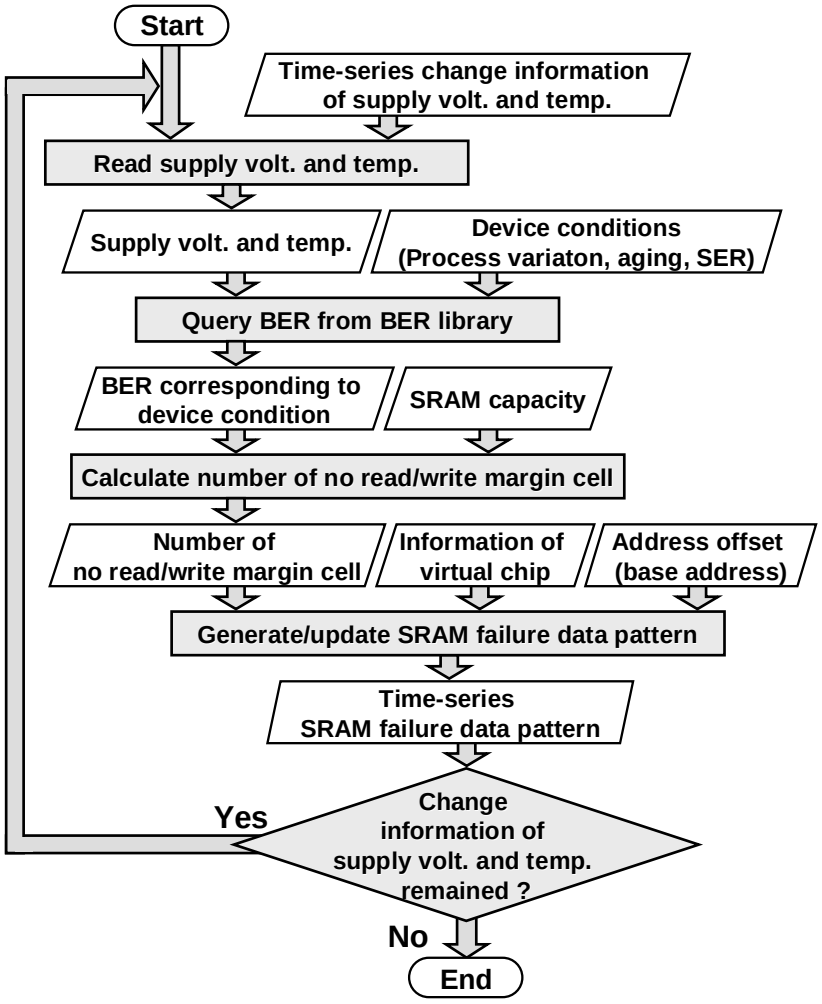


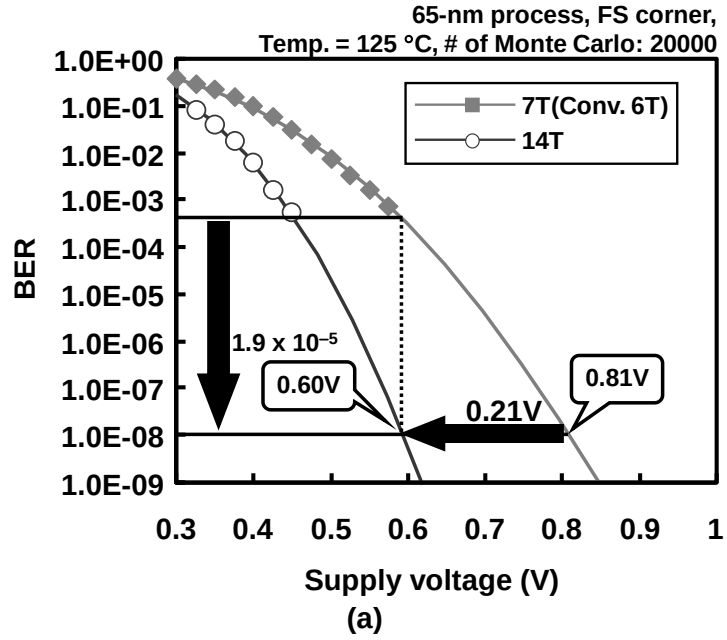
Fig. 5.8 SRAM failure data pattern generation flow.

5.4 7T/14T Dependable SRAM

5.4.1 Bit Error Rate (BER)

In the normal mode, a one-bit datum is stored in one memory cell, which means it is more area-efficient. In the dependable mode, a one-bit datum is stored in two memory cells, although the reliability of the information differs from that of the normal mode. The "more dependable with less failure rate" information is obtainable by combining two memory cells [5.7]. In addition, the 14T dependable mode has better soft-error tolerance than the 7T normal mode because its internal node has more capacitance.

Fig. 5.9(a) illustrates a bit error rate in the read operation. The SNM is used as a metric to evaluate read BERs. The dependable mode works fine below 0.60 V with a



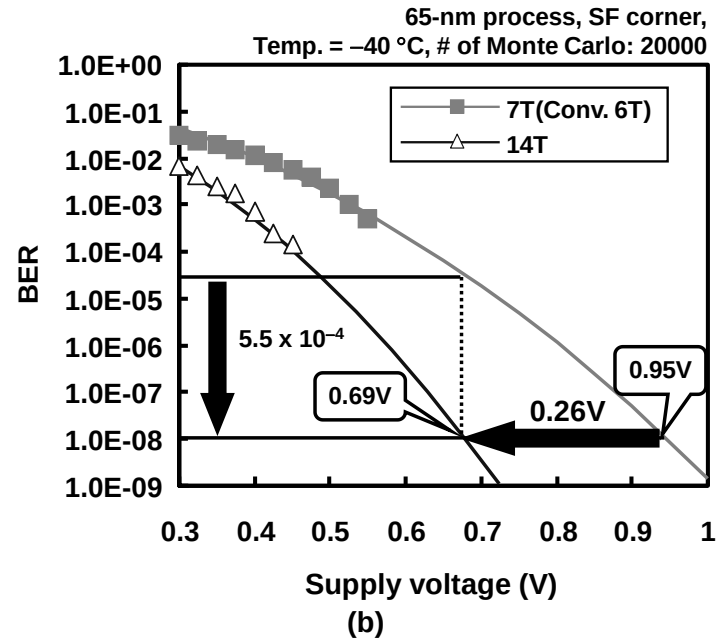


Fig. 5.9 Bit error rates (BERs): (a) read operation and (b) write operation. The respective "6T" and "14T" signify the conventional 6T memory cell and 14T dependable mode in the 7T/14T memory cells. Note that the performance of 7T is the same as 6T.

BER of 10^{-8} kept even in the worst-case condition (FS corner, 125°C). The minimum operating voltage and BER are improved by 0.21 V and 1.9×10^{-5} in comparison with the 6T cell (and thus with the 7T normal mode). The dependable mode is the most reliable in the read operation.

Fig. 5.9(b) is a BER in the write operation (worst-case condition: FS corner, –40°C).

The WTP is used as a metric to evaluate write BERs. In the dependable mode, the conductance of the access transistors is doubled, and variation is suppressed. Thereby, the write margin becomes larger. The proposed memory cell functions at 0.69 V with a BER of 10^{-8} kept. The minimum operating voltage and BER are improved by 0.26 V and 5.5×10^{-4} compared with the normal mode.

5.5 System-level Evaluation

To evaluate the proposed FIS integrated with the fault-injection scheme and system-level verification environment, we used the vehicle engine control ECU system embedded SH-2A processor shown in Fig. 5.2. In this evaluation, we used the conventional 6T SRAM and 7T/14T dependable SRAM as internal SRAM of ECU. Vehicle engine control software first ran on the ECU, and faults were injected to the internal SRAM in the ECU running the vehicle engine control software. With the fault-injection to the internal SRAM, operating stabilities of vehicle engine control ECU system using the 6T SRAM and the 7T/14T SRAM can be evaluated and compared. It is noteworthy that the dependable mode is used in the evaluation using the 7T/14T SRAM.

5.5.1 Evaluation Methodology

Abnormal termination of the vehicle engine control software is judged in two ways: a watchdog timer interruption triggered by a runaway of the software and an access violation to an illegal address. A normal termination is judged as when no abnormal termination occurs within the predefined execution time. It is noteworthy that abnormal behavior of the mechanical system was not considered in this study, only the behavior of the electric system.

The BERs of the 6T SRAM and 7T/14T SRAM were calculated in a 65-nm process as presented in Section 5.5.1. The process corner is a TT corner. For the degree of aging of the transistor, we assumed a degradation of PMOS threshold voltage as –24 mV assuming a 10-year aging by NBTI [5.8].

Table 5.3 summarizes the parameters for the system-level evaluation of the FIS. In the actual silicon chip, mapping of SRAM failure points differed for each chip. As a result, the impact of SRAM failures in each chip to the operating stability of each system was quite unique. Thus, to evaluate the functional safety of the system,

exhaustive system-level failure analysis for a large number of chips is necessary. In this evaluation, we generated and evaluated 1,050 virtual chips. To verify the functional safety, evaluations for more virtual chips are required. We leave such a large-scale evaluation to future work.

In this evaluation, inputs of supply voltages and operating temperatures did not change in time. We evaluated static supply voltage (DC) and operating temperature characteristics for the abnormal termination of system. As a result of the evaluation, knowledge of the operating range for evaluating the functional safety of the system can be obtained. The ranges of supply voltage and operating temperature evaluated are shown in Table 5.2.

Table 5.2 Parameters for System-level Evaluation

# of virtual chips	1,050
Execution time	10 sec.
Range of supply voltage	0.4V to 0.8V
Range of temperature	–50 degC to 150 degC
σ_{Vth} of PMOS, NMOS (L=60nm, W=120nm)	40mV, 30mV
Delta *Vth* of PMOS (aging)	–24mV
SRAM capacity	128 Kbytes
Soft error rate	300 FIT

5.5.2 Evaluation Result

Fig. 5.10 signifies the evaluation result of the abnormal termination rates in the vehicle engine control ECU system. The evaluation results using the 6T SRAM and 7T/14T SRAM as the internal SRAM of ECU are shown in Figs. 5.10(a) and (c) and 5.10(b) and (d), respectively. Figs. 5.10(c) and (d) are plotted on three-dimensional logarithmic graphs. In all results, the abnormal termination rates have a trend of becoming higher as the operating temperature becomes higher. This may be because the

Temperature (degC)

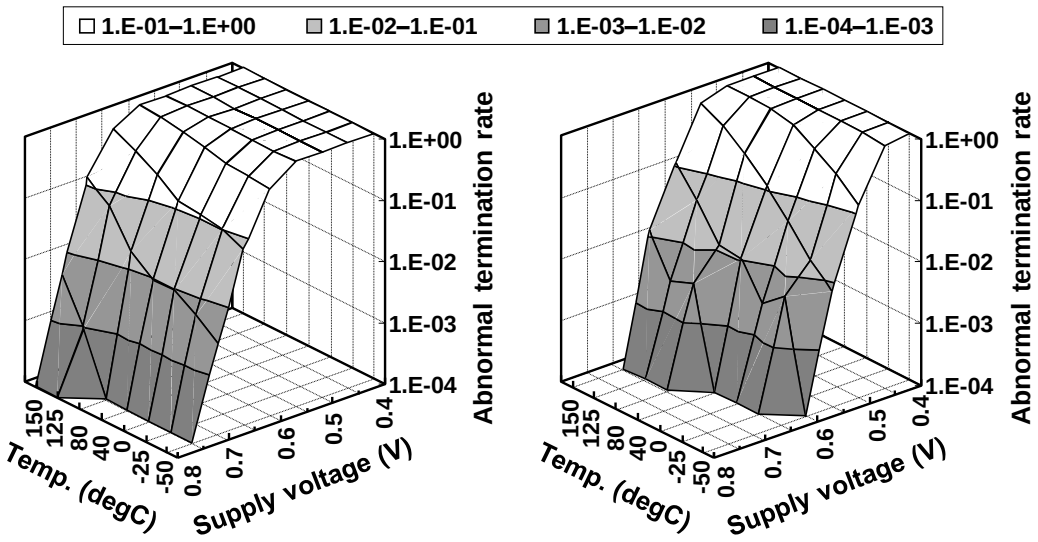
Supply voltage (V)	-50	-25	0	40	80	120	150
0.8	0	0	0	0	0	0	0
0.75	0	0	0	0	0	0.1	0.5
0.7	0.3	0.6	1.0	1.0	1.8	6.7	13.0
0.65	7.2	10.2	11.3	13.2	22.6	37.8	58.0
0.6	49.8	52.2	57.2	68.9	81.1	92.3	97.2
0.55	97.7	97.7	98.3	98.6	99.9	99.4	98.1
0.5	100	100	100	100	100	100	99.9
0.45	100	100	100	100	100	100	100
0.4	100	100	100	100	100	100	100

(a)

Temperature (degC)

Supply voltage (V)	-50	-25	0	40	80	120	150
0.8	0	0	0	0	0	0	0
0.75	0	0	0	0	0	0	0
0.7	0	0	0	0	0	0	0
0.65	0	0	0	0	0.4	0.3	1.2
0.6	0	0.6	0.3	1.1	1.1	4.2	11.3
0.55	0.8	1.6	4.5	11.2	25.7	47.5	72.8
0.5	13.3	28.4	49.6	78.5	94.9	99.0	99.7
0.45	85.5	97.0	99.3	100	100	100	100
0.4	99.9	100	99.8	100	100	100	100

(b)



(c) (d)

Fig. 5.10 Abnormal termination rates (%) of engine control ECU system: (a) and (c) 6T SRAM, (b) and (d) 14T dependable mode of 7T/14T SRAM. (c) and (d) are plotted on logarithmic graphs.

increase in the number of read margin failures affects the degradation of the abnormal termination rates because read margins of SRAM become worse at higher operating temperatures. The evaluation result using the 7T/14T SRAM (in dependable mode) improved V_{min} by 0.05–0.15 V compared with using the 6T SRAM. In addition, there was a trend that the V_{min} improvements provided by the 7T/14T SRAM are more in the lower operating temperature and less in the higher operating temperature. This is partly because the write margin failures, which are the dominant failure in the low operating temperature region, are reduced by 7T/14T SRAM. To analyze the reason for this, a statistical analysis is needed of a large amount of virtual chips to determine what kind of SRAM failure or where it is invokes the abnormal termination of the system. The cause–effect relationship between the SRAM failures and abnormal termination of the vehicle engine control ECU system is left to future work.

Fig. 5.11 shows the ECU system error rates (ECU system abnormal termination rates) and SRAM read/write BERs. To compare and discuss about the error rates, approximations of the normal distribution are used.

The approximations are found by an approximated error rate function, ER(V_{dd}), as shown below.

$$\begin{cases} f(x) = \begin{cases} \dfrac{1}{\sqrt{2\pi}\sigma} EXP\left[-\dfrac{(x-\mu)^2}{2\sigma^2} \right] & (x \geq \mu) \\ 0 & (x \leq \mu) \end{cases} \\ ER(V_{DD}) = \int_{V_{DD}}^{\infty} f(x)dx \end{cases} \quad (1)$$

Therein, f(x) is a probability density function and is assumed to be a normal distributed function. μ_{ER} is a voltage at which an error rate becomes 0.5 (i.e. random). σ_{ER} is a deviation, but in the figure, it is used for a fitting parameter.

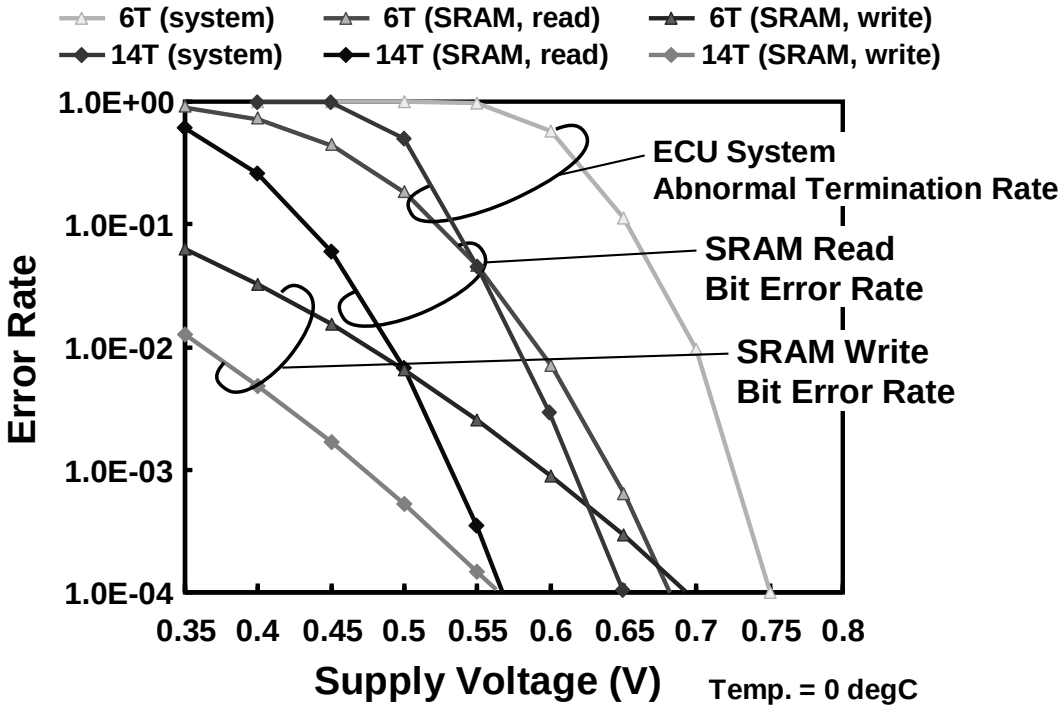


Fig. 5.11 Comparison of ECU system abnormal termination rates (error rate) and SRAM bit error rates.

Table 5.3 presents a summary of μ_{ER}s and σ_{ER}s of the approximated error rate functions. There seems to be stronger correlation between the abnormal termination rate and the SRAM read BER than that between the abnormal termination rate and the SRAM write BER.

Table 5.3 μ_{ER} and σ_{ER} of approximated error rate functions

	6T (system)	14T (system)	6T (SRAM, read)	14T (SRAM, read)	6T (SRAM, write)	14T (SRAM, write)
μ_{ER}	0.598	0.495	0.442	0.365	0.11	0.025
σ_{ER}	0.042	0.038	0.0645	0.0546	0.157	0.145

5.6 Summary

We propose a fault-injection system (FIS) that can inject well-device-conscious SRAM failures, including read/write margin failure and soft error, for system-level verification. The proposed fault-injection flow enables generation and injection of

SRAM failures from the device level to the system level. The proposed modeling method of failures in SRAM considers the physical characteristics of SRAM well and generates the SRAM failure that can be interpreted easily by the system-level verification. The fault case generator (FCG) can generate the time-series SRAM failure that can be injected for system-level verification. The detailed SRAM failure data pattern generation flow was described.

To evaluate the proposed FIS integrated with the fault-injection scheme and system-level verification environment, the abnormal termination rates of vehicle engine control ECU system using the 6T SRAM and the 7T/14T SRAM were evaluated. The vehicle engine control ECU using the 7T/14T SRAM was clearly observed to improve the system-level dependability compared with using the conventional 6T SRAM. By using the FIS, knowledge can be gained on how the dependability of SRAM affects the dependability of the processor system, and evaluation of the improvement in the dependability of a processor using SRAM with higher dependability can be performed easily.

5.7 References

[5.1] K. Itoh, R. Takemura, "Low-Voltage Limitations and Challenges of Memory-Rich Nano-Scale CMOS LSIs," Proceedings of IEEE Int. Conf. on Electronics, Circuits and Systems, pp.739-742, 2007.

[5.2] L. Chang, Y. Nakamura, R. K. Montoye, J. Sawada, A. K. Martin, K. Kinoshita, F. H. Gebara, K. B. Agarwal, D. J. Acharyya, W. Haensch, K. Hosokawa and D. Jamsek, “A 5.3 GHz 8T-SRAM with Operation Down to 0.41 V in 65 nm CMOS,” Symp. on VLSI Circuits Dig. Tech. Papers, pp. 252-253, 2007.

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

[5.4] V.K. Reddv, A.S. Al-Zawawi, and E. Rotenberg, “Assertion-Based Microarchitecture Design for Improved Fault Tolerance,” Proceedings of IEEE Int. Conf. on Computer

Design, pp.362-369, 2006.

[5.5] B. Eklow, A. Hosseini, C. Khuong, S. Pullela, T. Vo, and H. Chau, “Simulation Based System Level Fault Insertion Using Co-verification Tools,” Proceedings of IEEE Int. Test Conf., pp.704-710, 2004.

[5.6] C.R. Elks, M. Reynolds, N. George, M. Miklo, S. Bingham, R. Williams, B.W. Johnson, M. Waterman, and J. Dion, “Application of a fault injection based dependability assessment process to a commercial safety critical nuclear reactor protection system,” Proceedings of IEEE/IFIP Int. Conf. on Dependable Systems and Networks, pp.425-430, 2010.

[5.7] 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 Int. Symp. on Quality Electronic Design, pp. 98-102, 2008.

[5.8] R. Vattikonda, W. Wang, and Y. Cao, "Modeling and minimization of PMOS NBTI effect for robust nanometer design," Proceedings of the 43rd ACM/IEEE Design Automation Conference, pp. 1047-1052, 2006.

[5.9] Y. Nakata, Y. Ito, Y. Sugure, S. Oho, Y. Takeuchi, S. Okumura, H. Kawaguchi and M. Yoshimoto, "Model-Based Fault Injection for Failure Effect Analysis -Evaluation of Dependable SRAM for Vehicle Control Units-," Proceedings of the 5th Int. Workshop on Dependable and Secure Nanocomputing, in conjunction with the 41st Int. Conf. on Dependable Systems and Networks, pp. 91-96, Jun. 2011.

Chapter 6 Conclusion

This dissertation presents a description from the perspective of robust and high-performance design techniques of the VLSI processor under increasing process variation.

In Chapter 2, the main issues were pointed out as four: 1) degradation of operating stability caused by degradation of SRAM operating reliability, 2) processing performance degradation in the VLSI with the synchronous clock design, 3) degradation of scalabilities in the operating stability and the processing performance caused by the process variation, and 4) difficulty in analyzing VLSI system stability.

In this study, the solutions to these issues are presented as three techniques in this dissertation:

(1) Process-variation-adaptive memory design for reducing V_{min} (Chapter 3)
(2) Process-variation-adaptive large-scale many-core processor design for improving the processing performance (Chapter 4)
(3) System-level fault-injection scheme, which can be regarded as device level behaviors of SRAM (Chapter 5)

The subsequent three chapters described these practical designs in detail that contribute to overcoming issues in VLSI processors under increasing process variation.

In Chapter 3, a cache memory that can operate at low voltage under the effect of the process variation in a scaled process technology was described. A large-capacity SRAM macro determines the minimum operating voltage (V_{min}) of the entire VLSI processor. The cache memory leverages 7T/14T SRAM, which can improve its operating reliability: two pMOS transistors are appended between internal nodes in a pair of the conventional 6T SRAM bitcells. Adaptively, to mitigate the variability of operating stability of the SRAM in the large capacity SRAM cache macro, 32-bit word-level fine-grain mode control of the 7T/14T SRAM is introduced. The proposed scheme, named 7T/14T word-enhancing, also introduces a testing method that improves the efficiency of the 14T word-enhancing scheme. In a 65-nm process technology, the 4-MB cache implemented with the proposed scheme can operate at 0.5 V, which is 42%

and 21% lower, respectively, than a conventional 6T SRAM and a cache word-disable scheme. Measurements of the fabricated silicon chip in a 65-nm process confirmed that the 14T word-enahancing scheme can operate at 0.4 V and reduce V_{min} of the 6T SRAM and 14T dependable modes respectively by 25% and 19%. The respective dynamic power reductions are 89.2% and 73.9%. The respective total power reductions are 44.8% and 20.9%.

In Chapter 4, a network-on-chip (NoC) was reported: it can reconfigure its composition considering the process variation. Because NoC generally adopts a synchronous network design across the silicon chip, NoC is strongly affected by process variation, which brings different effects depending on the silicon chip location. The operating frequency of the chip-wide synchronous network is degraded as syncing the slowest network component in the silicon chip. A process-variation-adaptive NoC design is proposed to adapt process variation in individual locations of network routers. The proposed NoC introduces a variation-adaptive variable-cycle router (VAVCR) and a variable-cycle pipeline adaptive routing (VCPAR). The proposed VAVCR adaptively configures its processing latency of the router pipeline corresponding to the process variation of its location. The operating frequency of the network degraded by the process variation is improved by the adaptive reconfiguration of the proposed VAVCR. The proposed VCPAR is a routing algorithm that can accommodate processing cycle variation of the NoC with VAVCR. The VCPAR passes through low-cycle latency routers preferentially to minimize the packet transmission latency. The total execution time reduction of the proposed VAVCR with VCPAR is 15.7%, on average, for five task graphs.

Chapter 5 described a new system-level fault-injection scheme that can consider device-level behaviors of SRAM. In the robustness evaluation of VLSI processor system under severe operating conditions, consideration of vulnerable SRAM blocks in the VLSI processor is required. An SRAM operating stably under severe operating conditions is determined by a circuit level behavior and transistor device level variability. In the proposed system-level evaluation environment, the circuit level behavior and the transistor level variability of each individual SRAM are considered, and failures of the SRAM block in the severe operating condition can be injected to the evaluation environment. In the middle of this chapter, it is described that details of the

modeling of the SRAM circuit behavior, consideration of the variability of the transistor device, and a fault case generator (FCG) that can generate failure patterns injectable to the system-level evaluation environment. Subsequently, evaluations of the vehicle engine control system are presented. Results show that a dependable processor with 7T/14T dependable SRAM improves system-level dependability compared with the conventional 6T SRAM described at the end of this chapter.

The conclusion of this study is presented in this chapter. This dissertation presents the process-variation-aware robust and high-performance VLSI processor design techniques under increasing process variation. The three techniques described in this dissertation will be more valuable when applied in more scaled CMOS process technology, post-CMOS technology, and other promising future semiconductor technologies that have much more characteristic variation among devices.

List of Publications and Presentations

Publications in journals and transactions

1) Y. Nakata, H. Kawaguchi, and M. Yoshimoto, "A Process-Variation-Adaptive Network-on-Chip with Variable-Cycle Routers and Variable-Cycle Pipeline Adaptive Routing," IEICE Transactions on Electron., Vol. E95-C, No. 4, pp. 523-533, Apr. 2012.
2) S. Okumura, Y. Nakata, K. Yanagida, Y. Kagiyama, S. Yoshimoto, H. Kawaguchi, M. Yoshimoto, "Low-energy block-level instantaneous comparison 7T SRAM for dual modular redundancy," IEICE Electronics Express, Vol. 9, No. 6, pp.470-476, March, 2012.
3) Y. Nakata, S. Okumura, H. Kawaguchi, and M. Yoshimoto, “0.5-V 4-MB Variation-Aware Cache Architecture Using 7T/14T SRAM and Its Testing Scheme,” IPSJ Transactions on System LSI Design Methodology, Vol. 5, pp.32-43. Feb. 2012.
4) S. Okumura, Y. Kagiyama, Y. Nakata, S. Yoshimoto, H. Kawaguchi, and M. Yoshimoto, "7T SRAM Enabling Low-Energy Instantaneous Block Copy and Its Application to Transactional Memory," IEICE Transactions on Fundamentals, vol. E94-A, No. 12, pp. 2693-2700, Dec. 2011.

Presentations at international conferences

1) Y. Nakata, Y. Ito, Y. Takeuchi, Y. Sugure, S. Oho, H. Kawaguchi, and M. Yoshimoto, "Model-Based Fault Injection for Large-Scale Failure Effect Analysis with 600-Node Cloud Computers," DATE RIIF Workshop, Mar. 2013.
2) Y. Takeuchi, Y. Nakata, Y. Ito, Y. Sugure, S. Oho, H. Kawaguchi, and M. Yoshimoto, "SRAM Failure Injection to a Vehicle ECU and Its Behavior Evaluation," IEEE/ACM DATE RIIF Workshop, Mar. 2013.
3) J. Jung, Y. Nakata, M. Yoshimoto and H. Kawaguchi, "Energy-Efficient Spin-Transfer Torque RAM Cache Exploiting Additional All-Zero-Data Flags," Proceedings of IEEE International Symposium on Quality Electornic Design, Mar. 2013.

4) J. Jung, Y. Nakata, S. Okumura, H. Kawaguchi, and M. Yoshimoto, "A Variation-Aware 0.57-V Set-Associative Cache with Mixed Associativity Using 7T/14T SRAM," Proceedings of IEEE Faible Tension Faible Consommation, Jun. 2012.
5) Y. Nakata, S. Izumi, H. Kawaguchi, and M. Yoshimoto, "Trading off ECU Footprint for Reliability in X-by-Wire Application with Hybrid TMR Architecture," DAC International Workshop on System Level-Design of Automotive Electronics/Software, Jun. 2012.
6) Y. Kagiyama, S. Okumura, K. Yanagida, S. Yoshimoto, Y. Nakata, S. Izumi, H. Kawaguchi, and M. Yoshimoto, "Bit Error Rate Estimation in SRAM Considering Temperature Fluctuation," Proceedings of IEEE International Symposium on Quality Electronic Design, pp. 514-517, Mar. 2012.
7) K. Kugata, S. Soda, Y. Nakata, S. Okumura, S. Izumi, M. Yoshimoto, and H. Kawaguchi, "Processor Coupling Architecture for Aggressive Voltage Scaling on Multicores," Proceedings of ARCS Workshops 2012, pp. 375-384, Mar. 2012.
8) J. Jung, Y. Nakata, S. Okumura, H. Kawaguchi, and M. Yoshimoto, 256-KB Associativity-Reconfigurable Cache with 7T/14T SRAM for Aggressive DVS Down to 0.57 V ," Proceedings of IEEE International Conference on Electronics, Circuits, and Systems, pp. 524-527, Dec, 2011.
9) S. Okumura, Y. Nakata, K. Yanagida, Y. Kagiyama, S. Yoshimoto, H. Kawaguchi, and M. Yoshimoto, "Low-Power Block-Level Instantaneous Comparison 7T SRAM for Dual Modular Redundancy," Proceedings of IEEE Custom Integrated Circuits Conference, Sep. 2011.
10) Y. Nakata, Y. Takeuchi, H. Kawaguchi, and M. Yoshimoto, "A Process-Variation-Adaptive Network-on-Chip with Variable-Cycle Routers," Proceedings of the 14th Euromicro Conference on Digital System Design, pp. 801-804, Aug. 2011.
11) Y. Nakata, Y. Ito, Y. Sugure, S. Oho, Y. Takeuchi, S. Okumura, H. Kawaguchi, and M. Yoshimoto, "Model-Based Fault Injection for Failure Effect Analysis -Evaluation of Dependable SRAM for Vehicle Control Units-," Proceedings of the 5th Workshop on Dependable and Secure Nanocomputing, in conjunction with the 41st IEEE International Conference on Dependable Systems and

Networks, pp. 91-96, Jun. 2011.

12) M. Yoshikawa, S. Okumura, Y. Nakata, Y. Kagiyama, H. Kawaguchi, and M. Yoshimoto, "Block-Basis On-Line BIST Architecture for Embedded SRAM Using Wordline and Bitcell Voltage Optimal Control," Proceedings of IEEE International Symposium on Quality Electronic Design, pp. 322- 325, Mar. 2011.
13) S. Okumura, S. Yoshimoto, K. Yamaguchi, Y. Nakata, H. Kawaguchi, and M. Yoshimoto, "7T SRAM Enabling Low-Energy Simultaneous Block Copy," Proceedings of IEEE Custom Integrated Circuits Conference, pp. 1-4, Sep. 2010.
14) 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 ACM/IEEE International Symposium on Low Power Electronics and Design, pp. 219-224, Aug. 2010.
15) Y. Takeuchi, Y. Nakata, H. Kawaguchi, and M. Yoshimoto, "Scalable Parallel Processing for H.264 Encoding Application to Multi/Many-core Processor," Proceedings of IEEE International Conference on Intelligent Control and Information Processing (ICICIP), pp. 163-170, Aug. 2010.

Presentations at domestic conferences

1) 鄭晋旭, 中田洋平, 奥村俊介, 川口博, 吉本雅彦, "プロセスばらつきを考慮した低電圧動作混合連想度キャッシュ構造," 信学技報, vol. 112, no. 170, ICD2012-31, pp. 1-6, 2012 年 8 月.
2) 中田洋平, 川口博, 吉本雅彦, "プロセスばらつきを考慮した NoC アーキテクチャ," LSI とシステムのワークショップ 2012, pp. 204-206, 北九州市, 2012 年 5 月. （優秀ポスター賞受賞）
3) 鄭晋旭, 中田洋平, 奥村俊介, 川口博, 吉本雅彦, "低電圧動作マージン拡大機能を有する連想度可変キャッシュ," LSI とシステムのワークショップ 2012, pp. 207-209, 北九州市, 2012 年 5 月.
4) 柳田晃司, 奥村俊介, 中田洋平, 鍵山祐輝, 吉本秀輔, 川口博, 吉本雅彦, "低エネルギ比較機能を有する DMR 応用 7T SRAM," LSI とシステムのワークショップ 2012, pp.186-188, 北九州市, 2012 年 5 月.

5) 梅木洋平, 奥村俊介, 中田洋平, 柳田晃司, 鍵山祐輝, 吉本秀輔, 川口博, 吉本雅彦, "低エネルギ比較機能を有する DMR 応用 7T SRAM," 信学技報, vol. 112, no. 15, ICD2012-16, pp. 85-90, 2012 年 4 月, 岩手.
6) 北原佑起, 鍵山祐輝, 奥村俊介, 柳田晃司, 吉本秀輔, 中田洋平, 和泉慎太郎, 川口博, 吉本雅彦, "温度変化を考慮した SRAM の BER 導出手法の検討," 電子情報通信学会総合大会, 2012 年 3 月.
7) 藤川飛鳥, 吉川将弘, 奥村俊介, 中田洋平, 鍵山祐輝, 川口博, 吉本雅彦, "ディペンダブル SRAM のためのオンライン故障診断技術の開発," 電子情報通信学会総合大会, 2012 年 3 月.
8) 鄭晋旭, 中田洋平, 奥村俊介, 川口博, 吉本雅彦, "低電圧動作におけるマージン拡大機能を有する連想度可変キャッシュ," 信学技報, vol. 111, no. 388, ICD2011-139, pp.55-60, 2012 年 1 月
9) 竹内勇介, 中田洋平, 伊藤康宏, 勝康夫, 於保茂, 奥村俊介, 川口博, 吉本雅彦, "故障注入技術を用いたディペンダブル SRAM を搭載するプロセッサの信頼性評価・検証", 電子情報通信学会技術研究報告, CPSY2011-25, pp.1-6, 2011 年 10 月.
10) 鍵山祐輝, 奥村俊介, 吉本秀輔, 中田洋平, 川口博, 吉本雅彦, "ブロックデータ一括コピー機能を有する 7T SRAM," LSI とシステムのワークショップ 2011, pp.209-211, 北九州市, 2011 年 5 月.
11) 竹内勇介, 中田洋平, 伊藤康宏, 勝康夫, 於保茂, 川口博, 吉本雅彦, "システムレベル故障注入技術によるディペンダブルメモリを搭載したプロセッサの評価・検証," LSI とシステムのワークショップ 2011, pp.209-211, 北九州市, 2011 年 5 月.
12) 鄭晋旭, 中田洋平, 奥村俊介, 川口博, 吉本雅彦, "7T/14T SRAM の細粒度制御による低電圧動作キャッシュアーキテクチャ," LSI とシステムのワークショップ 2011, pp.209-211, 北九州市, 2011 年 5 月.
13) 中田洋平, 伊藤康宏, 勝康夫, 於保茂, 川口博, 吉本雅彦, "システムレベル故障注入技術を用いたディペンダブルプロセッサアーキテクチャの評価・検証," 電子情報通信学会技術研究報告, vol. 110, no. 317, VLD2010-74,

DC2010-41, pp.125-130, 2010 年 11 月.

14) 伊藤康宏, 中田洋平, 川口博, 吉本雅彦, 勝康夫, 於保茂 "非実機環境上での故障注入技術による車載システムレベル信頼性評価技術," 電子情報通信学会技術研究報告, vol. 110, no. 317, VLD2010-73, DC2010-40, pp.119-123, 2010 年 11 月.

15) 中田洋平, 竹内幸大, 川口博, 吉本雅彦, "プロセスばらつきを考慮した NoC アーキテクチャの検討," 情報処理学会研究報告 計算機アーキテクチャ(ARC), 2010-ARC-191(5), 1-5, Oct. 2010. （2011 年度 山下記念研究賞受賞）

16) 奥村俊介, 鍵山祐輝, 吉本修輔, 山口幸介, 中田洋平, 川口博, 吉本雅彦, "ブロック一括コピー機能を有する 7T SRAM," 電子情報通信学会 CEATEC JAPAN 2010 連携企画研究報告（Digital Harmony を支えるプロセッサと DSP, 画像処理の最先端）, pp.49-54, Oct. 2010.

17) 中田洋平, 竹内幸大, 川口博, 吉本雅彦, "マルチコアプロセッサにおける H.264/AVC 符号化処理の並列度とメモリアクセスに関する高効率実装," DA シンポジウム 2010, pp. 195-200, 豊橋市, Sep. 2010.

18) 中田洋平, 川口博, 吉本雅彦, "7T/14T SRAM を内部メモリに用いたマルチコア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 2010, pp. 209-211, 北九州市, May 2010. （最優秀ポスター賞受賞）

19) 中田洋平, 川口博, 吉本雅彦, "7T/14T SRAM を内部メモリに用いたマルチコアプロセッサアーキテクチャの検討," DA シンポジウム 2009, pp. 163-168, 金沢市, Aug. 2009. （優秀学生発表賞受賞）

20) 坂田義典, 中田洋平, 川上健太郎, 川口博, 吉本雅彦, "動的電源電圧/周波数制御によるフレームバッファ SRAM 内蔵型 H.264/AVC デコーダの低消費電力化," 第 11 回システム LSI ワークショップ, pp. 204-206, 北九州市, Nov. 2007.

Patents

1) Masahiko YOSHIMOTO, Hiroshi KAWAGUCHI, Yohei NAKATA, Shunsuke OKUMURA, "LOW-VOLTAGE SEMICONDUCTOR MEMORY," Publication Number: WO/2012/023277 (23.02.2012)
2) 吉本雅彦, 川口博, 中田洋平, “キャッシュメモリとそのモード切替方法”, 特開 2011-040010

Acknowledgements

I would like to express my most earnest appreciation to Professor Masahiko Yoshimoto of Kobe University for offering me his continued profound advice, appropriate guidance, and enthusiastic encouragement throughout the preparation of this dissertation. I am grateful as well to Associate Professor Hiroshi Kawaguchi of Kobe University for providing me meaningful advice and guidance based on his immense knowledge related to this research. I am at a loss to express my true gratitude to them.

I also would like to thank Professor Makoto Nagata and Professor Yusaku Yamamoto for giving me helpful advice for this dissertation. I also appreciate Professor Itsuro Kakiuchi for valuable discussion and helpful advice related to the contents of Chapter 3.

I am thankful for research support and helpful discussions with Professor Shigeru Oho of the Nippon Institute of Technology, Dr. Yasuo Sugure and Mr. Yasuhiro Ito of Central Research Laboratory, Hitachi Ltd., and Mr. Masafumi Shimozawa of Hitachi Solutions Ltd., which was instrumental in the preparation of Chapter 5.

I am deeply grateful to my colleagues of the DVS project: Mr. Yoshinori Sakata, Mr. Yukihiro Takeuchi, Mr. Jinwook Jung, Mr. Yusuke Takeuchi, Mr. Asuka Fujikawa, Ms. Mari Masuda, Mr. Yuta Kimi, and Mr. Go Matsukawa. I would particularly like to thank Dr. Hidehiro Fujiwara, Dr. Hiroki Noguchi, and Mr. Shunsuke Okumura for fruitful discussions of SRAM design and applications. My appreciation also goes to Assistant Professor Shintaro Izumi for valuable technical discussions. I acknowledge also those research members who discussed and supported my research: Dr. Koji Nii, Dr. Yuichiro Murachi, Dr. Yasuhiro Morita, Dr. Toshikazu Suzuki, Dr. Hiroaki Suzuki, Dr. Takashi Takeuchi, and Dr. Takashi Matsuda. I have enormous appreciation of Mr. Shunsuke Okumura, Mr. Toshihiro Konishi, and Mr. Kosuke Mizuno for discussion of VLSI design and spending time in the laboratory for 6 years as classmates and colleagues.

During the enriched time I spent in the laboratory, I was fortunate to meet Mr. Takahiro Iinuma, Mr. Tomokazu Ishihara, Ms. Fang Yin, Mr. Akihiro Gion, Mr. Mitsuhiko Kuroda, Mr. Yuhi Higuchi, Mr. Keiichi Yoshino, Mr. Hyeokjong Lee, Mr. Yusuke Iguchi, Mr. Yu Otake, Mr. Kenichiro Yagura, Mr. Tetsuya Kamino, Mr. Junichi Tani, Mr. Koh Tsuruda, Mr. Kazuo Miura, Mr. Yasuharu Sakai, Mr. He Guangji, Mr. Akihisa Oka, Mr. Yusuke Shimai, Mr. Tomoya Takagi, Mr. Tsuyoshi Fujinaga, Mr.

Masahiro Yoshikawa, Mr. Kosuke Yamaguchi, Mr. Shusuke Yoshimoto, Mr. Takuro Amashita, Mr. Yuki Kagiyama, Mr. Koji Kugata, Mr. Takanobu Sugahara, Mr. Masaharu Terada, Mr. Yosuke Terachi, Mr. Masanori Nishino, Mr. Keisuke Okuno, Mr. Shimpei Soda, Mr. Yuki Miyamoto, Mr. Koji Yanagida, Mr. Yohei Umeki, Mr. Yuki Kitahara, Mr. Kenta Takagi, Mr. Masanao Nakano, Mr. Ken Yamashita, Mr. Song Dae-Woo, Mr. Tomoki Nakagawa, Mr. Takahide Fujii, and Mr. Kumpei Matsuda. I would like express my appreciation to Ms. Emi Go, Ms. Keiko Matsuoka, Ms. Aya Tsuboi, and Ms. Yurie Izumi for their kindness. I also would like to thank my English teacher, Ms. Mitsu Tsukino for her hearty encouragement and advice for my presentations at international conferences.

I am grateful for helpful suggestions and chip fabrication support offered to me by Mr. Takuya Sawada and Mr. Taku Toshikawa of Kobe University, and Mr. Hirofumi Nakano, Mr. Makoto Yabuuchi, Dr. Hidehiro Fujiwara, Dr. Koji Nii, and Mr. Hiroyuki Kawai of Renesas Electronics.

I would like also to acknowledge financial support of this research. Chapters 3 and 5 are supported by Japan Science and Technology Agency (JST) CREST.

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

Finally, I wish to thank my parents, brothers, and family who raised me, encouraged me, and supported me.

Yohei Nakata