Model-Based Fault Injection for Failure Effect Analysis

– Evaluation of Dependable SRAM for Vehicle Control Units –

Yohei Nakata[1], Yasuhiro Ito[2], Yasuo Sugure[2], Shigeru Oho[2], Yusuke Takeuchi[1], Shunsuke Okumura[1], Hiroshi Kawaguchi[1], and Masahiko Yoshimoto[1, 3]
[1]Graduate School of System Informatics, Kobe University, Kobe, 657-8501 Japan
[2]Central Research Laboratory, Hitachi, Ltd., Kokubunji, 185-8601 Japan
[3]Japan Science and Technology Agency (JST), CREST
Email: [1]nkt@cs28.cs.kobe-u.ac.jp, [2]{yasuhiro.ito.kp, yasuo.sugure.mq, shigeru.oho.ku}@hitachi.com,
[1]{takeuchi_u, s-oku, kawapy}@cs28.kobe-u.ac.jp, [1, 3]yosimoto@cs.kobe-u.ac.jp

***Abstract*—We propose a fault-injection system (FIS) that can inject faults such as read/write margin failures and soft errors into a SRAM environment. 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 are applicable for system-level verification. For evaluation, an abnormal termination rate in vehicle engine control was adopted. We confirmed that the vehicle engine control system with the 7T/14T SRAM improves system-level dependability compared with the conventional 6T SRAM.**

Keywords- fault injection; SRAM; system-level verification; dependable processor

I. INTRODUCTION

Recently, VLSI is increasingly becoming a key part in various industrial products. Therefore, its 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 variability in the transistor worsens its reliability and LSI yield. On the LSI, SRAM is comprised of the smallest-size transistors, which is thus the dominant factor that determines the LSI's reliability. Accordingly, high reliability is required for SRAM on the system LSI [1–3].

There have been many studies and implementations of fault injection into the LSI [4–6]. These studies injected stuck-at faults and transient faults due 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 a large number of physical LSIs each with different characteristics due to the random process variation.

II. FAULT-INJECTION SYSTEM

To exhaustively verify operating stability on a system LSI integrating a large number of vulnerable SRAMs, we must consider the impacts of its reliability on the operating stability. 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.

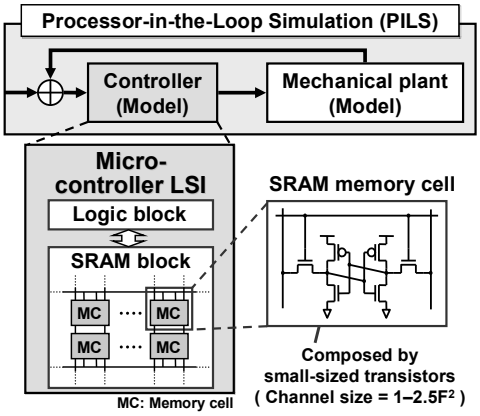

Figure 1. An overview of processor-in-the-loop simulation (PILS) and a simple diagram of a controller LSI composed of a logic block and SRAM block.

Fig. 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} [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. 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®.[1] The SH-2A processor is emulated by CoMET®.[2]

As shown in Fig. 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.

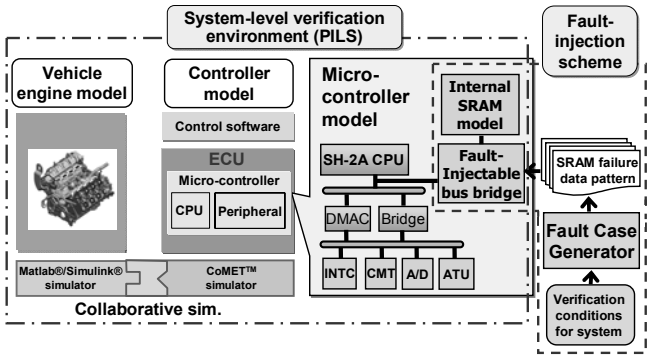

Figure 2. The proposed system-level verification environment has a vehicle engine model and controller (ECU) model in which a micro-controller model is included. Faults are injected through a fault-injectable bus bridge on an SH-2A CPU.

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.

III. 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 Sections III.A and III.B, respectively. Subsequently, the proposed fault injection flow is described in Section III.C. Next, a modeling method for the SRAM behavior at the device level is presented in Section III.D. Finally, the FCG that generates failure memory data patterns is stated in Section III.E.

A. Failures in SRAM

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

• **Read margin failure**: read operation is signified by a read static noise margin (read SNM) [7]. If the read SNM becomes zero due to a low V_{dd}, noise source, or destructive readout, the stored datum then flips.

• **Write margin failure**: write operation is explained by a write-trip point (WTP) as a metric (= write margin) [8]. The WTP indicates the maximum voltage that can write "0" to a memory cell and hen 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 violation**: occurs when a differential voltage between bitlines is small and a sense amplifier cannot sense it within a predetermined acceptable time. The access time violation is dependent on the clock frequency and timing guard band. This failure type was not considered in this study because it is dependent on the clock frequency. The read SNM, write margin, and soft error are dominant at low operating frequencies.

B. 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. 3 shows the failure pattern examples of the read/write margin failure and soft error.

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 a 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.

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

[2] CoMET® is a registered trademark of Synopsys®, Inc.

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.

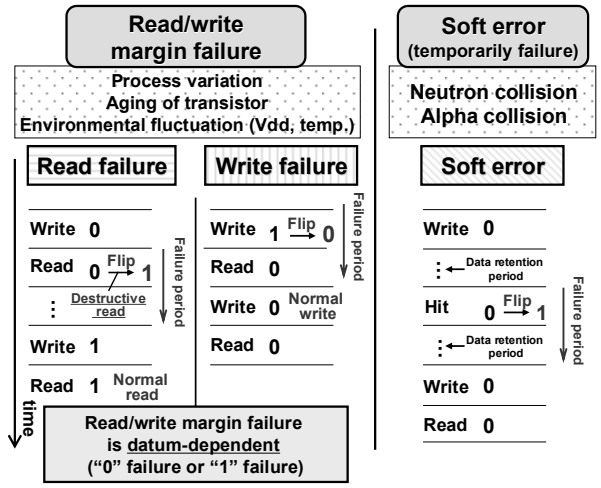

Figure 3. Failure pattern examples in SRAM memory cell: read margin failure, write margin failure, and soft error.

C. *Proposed Fault-Injection Flow for System-Level Verification*

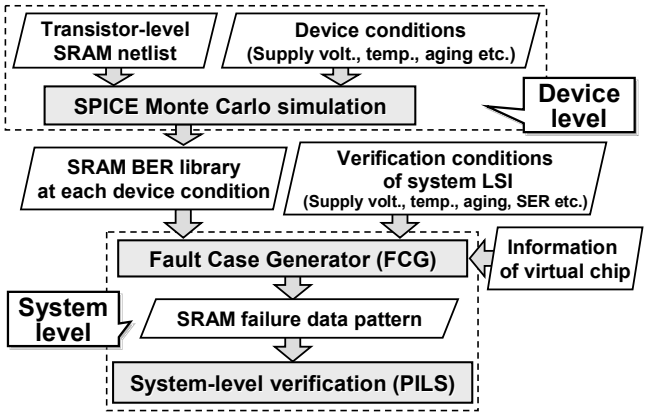

Figure 4. Proposed fault injection scheme flowing from a device level to a system level.

Fig. 4 illustrates 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, a 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 needs to 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.

D. *Modeling Failures for System-Level Fault Injection*

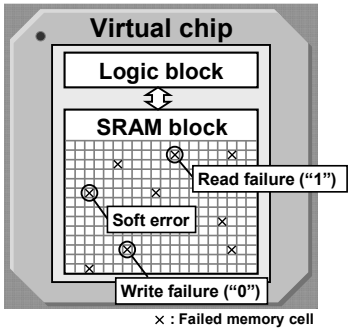

Figure 5. Virtual chip: SRAM failures are randomly distributed across a chip. The data-dependence is also randomized as "0" or "1".

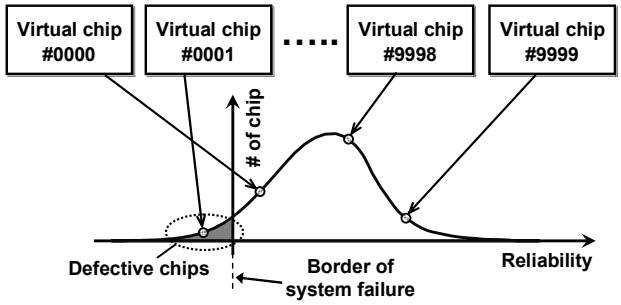

Figure 6. Large-scale verification using the 10,000 virtual chips.

In this subsection, the modeling method for generating SRAM failures is proposed. Fig. 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; this is 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. 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.

E. Fault Case Generator

Fig. 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.

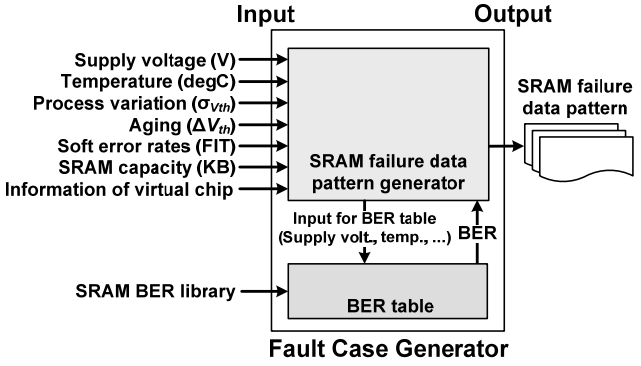

Figure 7. Fault Case Generator.

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.

IV. 7T/14T Dependable SRAM

A. 7T/14T SRAM

Fig. 8 depicts the 7T/14T SRAM memory cell (14T for two memory cells) [9]. Two PMOSs are added to internal nodes ("N00 and N10", "N01 and N11") in a pair of conventional 6T SRAM memory cells, as shown in Fig. 9. The area overhead the 7T memory cell is 11% greater than that of the conventional 6T memory cell.

The 7T/14T memory cells have two modes, as shown in Table I.

• 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 write operation, both WL0 and WL1 are driven, but in read operation, either WL0 or WL1 is asserted, which ensures stable operation.

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 [9]. In addition, the 14T dependable mode has better soft-error tolerance than the 7T normal mode because its internal node has more capacitance.

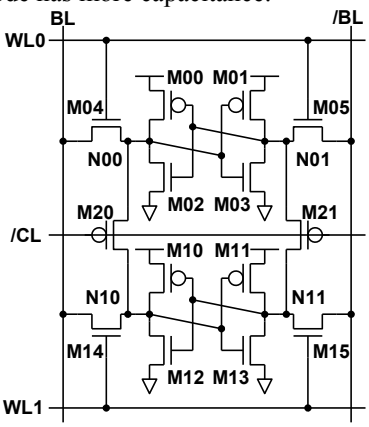

Figure 8. 7T/14T memory cell pair.

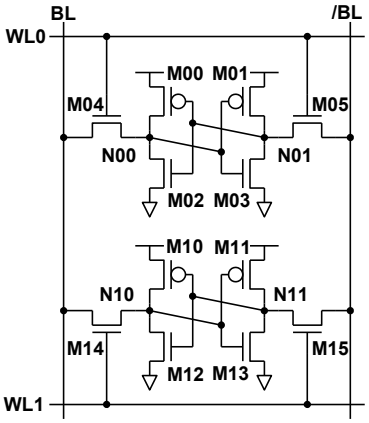

Figure 9. Conventional 6T memory cell.

TABLE I. Two modes in 7T/14T memory cell.

	# of memory cells comprising 1 bit	# of WL drives	CL
Normal	1 (7T/bit)	1	Off ("H")
Dependable (read)	2 (14T/bit)	1	On ("L")
Dependable (write)	2 (14T/bit)	2	On ("L")

B. Bit Error Rate(BER)

Fig. 10(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 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. 10(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.

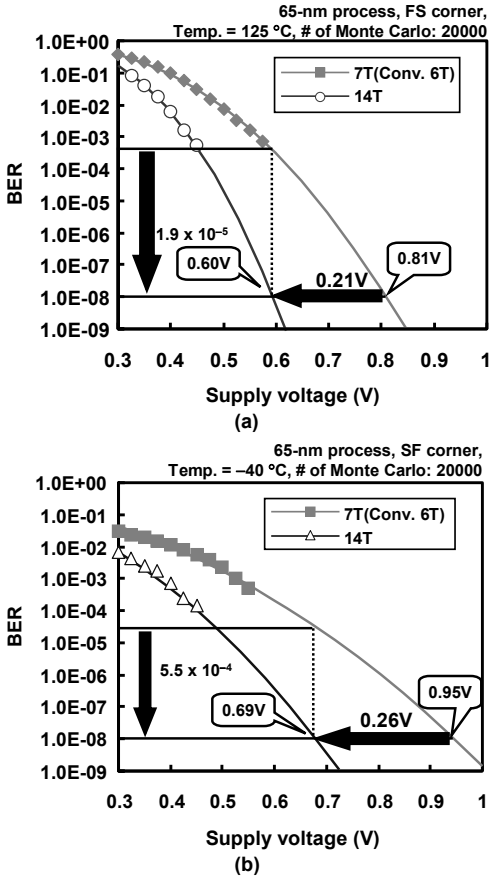

Figure 10. Bit error rates (BERs): (a) read operation and (b) write operation. The respective "6T" and "14T" signify the conventional 6T memory cell and 14T dependable mode in the 7T/14T memory cells. Note that the performance of 7T is the same as 6T.

V. 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. 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. Note that the dependable mode is used in the evaluation using the 7T/14T SRAM.

A. 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. Note 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 V.B. The process corner is a TT corner. For the degree of aging of the transistor, we assumed a degradation of PMOS threshold voltage as –24mV assuming a 10-year aging by NBTI [10].

Table II 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 1050 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 II.

TABLE II. 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
Soft error rate	300 FIT

B. Evaluation Result

Fig. 11 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. 11(a) and (c) and 11(b) and (d), respectively. Figs. 11(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 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.

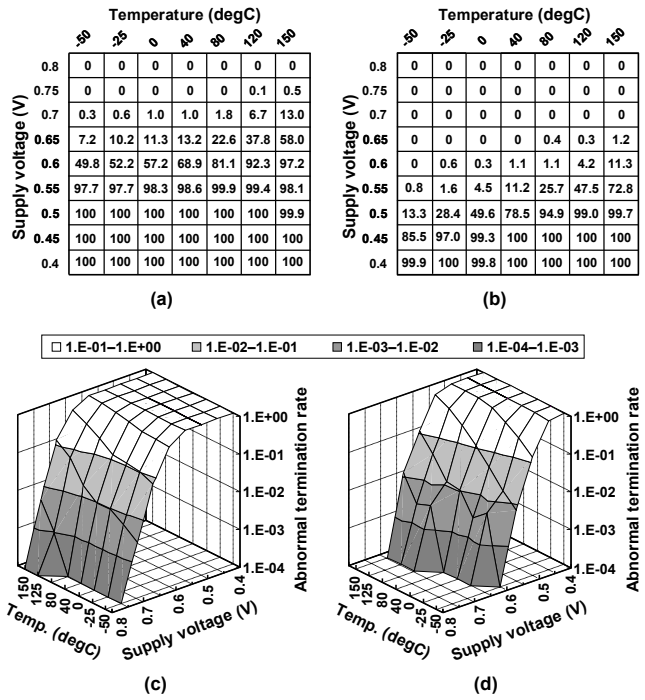

Figure 11. Abnormal terminating rates (%) of engine control ECU system: (a) and (c) 6T SRAM, (b) and (d) 14T dependable mode of 7T/14T SRAM. (c) and (d) are plotted on logarithmic graphs.

VI. CONCLUSION

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 easily interpreted 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.

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 easily performed.

REFERENCES

[1] K. Itoh, R. Takemura, "Low-Voltage Limitations and Challenges of Memory-Rich Nano-Scale CMOS LSIs," *14th IEEE Int. Conference on Electronics, Circuits and Systems (ICECS)*, pp.739-742, 2007

[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 5.3GHz 8T-SRAM with Operation Downto 0.41V in 65nm CMOS," *Symposium on VLSI Circuits*, pp. 252-253, 2007.

[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 Journal of Solid-State Circuits*, vol. 41. no. 3, pp. 705-711, 2006.

[4] V.K. Reddv, A.S. Al-Zawawi, and E. Rotenberg, "Assertion-Based Microarchitecture Design for Improved Fault Tolerance," *Int. Conference on Computer Design (ICCD)*, pp.362-369, 2006.

[5] B. Eklow, A. Hosseini, C. Khuong, S. Pullela, T. Vo, and H. Chau, "Simulation Based System Level Fault Insertion Using Co-verification Tools," *Int. Test Conference (ITC)*, pp.704-710, 2004.

[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," *IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN)*, pp.425-430, 2010.

[7] E. Seevinck, F.J. List, J. Lohstroh, "Static-noise margin analysis of MOS SRAM cells," *IEEE Journal of Solid-State Circuits*, vol. 22, no. 5, pp. 748-754, 1987.

[8] R. Heald, and P. Wang, "Variability in sub-100 nm SRAM designs," *IEEE/ACM Int. Conf. on Computer Aided Design (ICCAD)*, pp. 347-352, 2004.

[9] 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," *9th Int. Symposium on Quality Electronic Design (ISQED)*, pp. 98-102, 2008.

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