Soft Error and Its Countermeasures in Terrestrial Environment

Masanori Hashimoto

Dept. Information Systems Eng., Osaka University
e-mail: hasimoto@ist.osaka-u.ac.jp

Wang Liao

School of System Eng., Kochi University of Technology
e-mail liao.wang@kochi-tech.ac.jp

Abstract— This paper discusses soft errors in digital chips consisting of SRAM, flip-flops, and combinational logic in the terrestrial environment. We review the effectiveness of error-correction coding (ECC) in processor systems and point out the importance of radiation-hardened flip-flops for further error mitigation. The discussion includes the difference between planar and FD-SOI transistors, and the type of secondary cosmic rays including neutron and muon, using irradiation test results. Also, the difficulty in characterizing SER of a commercial GPU chip is exemplified.

I. INTRODUCTION

People and society have been more and more dependent on the services provided by the information systems. For example, autonomous driving is intensively studied, and experiments with prototyped cars are carried out all over the world. The autonomous driving is an intelligible and highly probable near-future situation that we are entrusting our lives to VLSI-centric information systems. Another technology movement is Internet of Things (IoT), and a report predicts that more than 30 billion devices are connected to Internet in 2018 [1]. We are expecting IoT would enable a more comfortable, more efficient, safer and securer society. Thus, these technology trends make it a social requirement to guarantee the reliability of the VLSIs in the information systems.

The problem of soft errors occurring inside the VLSIs in the terrestrial radiation environment has been recognized as a major threat to electronics at ground level [2]. Radiation-induced soft error is represented as a transient malfunction in VLSIs due to single event upset (SEU), which is caused by a transient signal induced by energetic ionizing radiation and destroy the information stored in memory elements.

This paper aims to share the mechanism of soft error occurrence, the impact of soft error on digital systems, and countermeasures for developing cost-effective countermeasures. We first explain the soft error occurring in the terrestrial environment in Section II. Section III discusses the soft error rate (SER) of SRAM, which is the most sensitive component in VLSIs [3] with irradiation experiments. The effectiveness of error correction code (ECC) is also investigated. The presented results cover conventional bulk transistor, which continues to be used for cost-effective IoT applications, and FD-SOI transistors. Furthermore, muon-induced soft error is also discussed as a future reliability concern. We next discuss the impact of soft error on digital chips in Section IV. First, the contributions of SRAM, FF, and combinational circuit to the chip-level SER are estimated. We also point out the difficulty in SER characterization of commercial chips taking an irradiation test for GPU as an example. Finally, Section V introduces high-
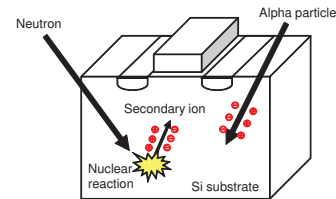

Fig. 1. Soft error mechanism due to neutron and alpha.

level countermeasures for soft error mitigation, and Section VI gives concluding remarks.

II. SOFT ERROR IN TERRESTRIAL ENVIRONMENT

In the terrestrial environment, soft errors are induced by alpha particles emitted from package material and neutrons originating from cosmic ray. Alpha particles are ionized radiation particles, and hence they can directly generate electron-hole pairs, as illustrated in Fig. 1. Neutrons, on the other hand, indirectly induce soft error through reaction with atomic nucleus of transistor materials, which is also shown in Fig. 1. The nuclear reaction generates charged secondary particles like protons, alpha particles, and heavy ions. An example of such nuclear reactions and charge deposition due to the generated secondary particles is found in [4].

The charged particle generates electron-hole pairs on the particle track and deposits charge. The generated charge is collected to the drain by drift and diffusion and causes soft error [2]. The collected charge finally induces a temporal glitch at the drain node of the transistor. Temporal errors caused by this glitch are called soft errors. More specifically, a glitch occurring in a combinational circuit is called single event transient (SET), and a glitch that occurs in a memory element and upsets the memory information is called SEU. The critical region inside the silicon substrate where an ionized particle hit causes a soft error is called a sensitive node or sensitive volume. Alpha particle induced soft error can be mitigated by using a low alpha emission package. On the other hand, although building can somewhat reduce neutron flux [5], neutrons are difficult to be eliminated in general, and then neutron is the major source of soft error in terrestrial environment [6].

Also, recent literature [7]–[10] points out that muons are a potential source of soft error in the terrestrial environment. The muon is an elementary particle similar to the electron, but the mass of a muon is 207 times larger than the mass of an electron. There are two types of muons, namely, the negative muon (μ^-) and the positive muon (μ^+). Fig. 2 shows the energy spectra of major secondary cosmic ray particles. A substantial component of secondary cosmic rays at ground level is known to be muons [7], and its fraction is about three-quarters of the total cosmic ray flux. Fig. 3 illustrates the
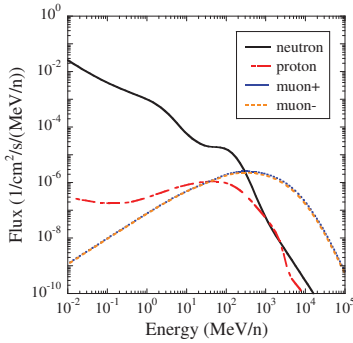
Fig. 2. Energy spectra of major secondary cosmic ray particles at NYC obtained with EXPACS [11], [12].
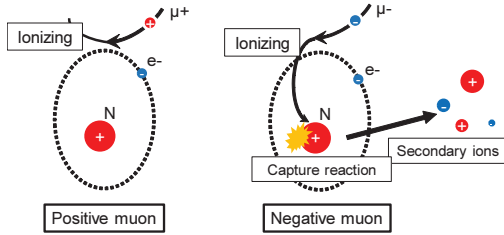

Fig. 3. Charge deposition mechanisms of positive and negative muons.

difference in charge deposition mechanism between negative and positive muons. Both positive and negative muons can deposit charge due to direct ionization. Also, a capture reaction of low-energy negative muon generates secondary ions, and they deposit larger charge than direct ionization.

III. SOFT ERROR IN SRAM

A. MCU mechanism in bulk and FD-SOI SRAM

Among SEUs in SRAM, multiple cell upsets (MCUs) induced by a single neutron are becoming a serious concern [13]. MCUs can be mostly mitigated by popular countermeasures of interleaving and ECC. On the other hand, as the number of upsets for an event becomes larger, MCU patterns which cannot be eliminated by interleaving and ECC are more likely to arise. Such critical MCUs are called MBU (multiple bit upsets), and they prevent massively-parallel high-performance computing systems and highly reliability-demanding applications from being implemented and operated. Besides, there are four possible mechanisms of MCU: (1) successive hits of an ion, (2) simultaneous multiple hits by multiple ions, (3) charge drift/diffusion (charge sharing), and (4) parasitic bipolar action.

In bulk SRAM, (3) charge sharing and (4) parasitic bipolar action are major mechanisms at the nominal supply voltage. Charge sharing causes MCU due to charge diffusion to multiple cells. Parasitic bipolar action triggered by well potential fluctuation flips multiple cells in a well. Fig. 4 illustrates the parasitic bipolar action in bulk SRAM. Holes generated by a neutron-induced nuclear reaction elevate the voltage of the P-well, which is equivalent to the base-emitter voltages of the parasitic bipolar transistors, due to well resistance. Consequently, the collector-emitter current of the parasitic bipolar transistors increases, which causes MCUs. Regarding supply voltage, these two mechanisms have opposite tendencies. As the supply voltage becomes lower, the critical charge, namely the threshold for the collected charge inducing soft errors,
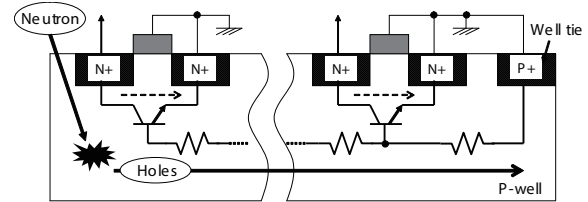

Fig. 4. Cross section of bulk NMOSs in memory cells. Parasitic bipolar transistors cause multiple upsets due to an increase in potential of P-well.
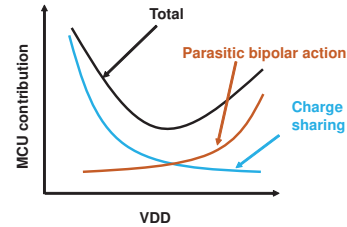

Fig. 5. Qualitative explanation of contributions of parasitic bipolar action and charge sharing to MCU in bulk SRAM.

becomes smaller, which makes charge sharing-induced MCUs occur easily. On the other hand, the parasitic bipolar action becomes less active due to lower collector-emitter voltage, and consequently, MCUs due to the parasitic bipolar action are less likely to arise. These two tendencies determine the dependency of MCU on voltage, which is illustrated in Fig. 5.

In FD-SOI SRAM, on the other hand, MCUs due to (3) charge sharing and (4) parasitic bipolar action do not occur since FD-SOI transistors do not share a well. Therefore, the remaining (1) successive hits of an ion and (2) multiple hits by multiple ions cause MCUs in FD-SOI SRAM. Consequently, the MCU rate is lower, and large-bit MCUs are less probable to occur in FD-SOI SRAM.

Fig. 6 shows the measurement results of the accelerated neutron test with voltage scaling in 65-nm bulk and silicon-on-thin-box (SOTB) SRAMs [14], where SOTB is an FD-SOI device with thinner buried oxide and SOI layers [15]. The number of measured SEUs on the SOTB SRAM at 0.4 V supply voltage was 4.4 times larger than that at 1.0 V supply voltage, while the number of SEUs on the SOTB SRAM at 0.4 V was 0.08 times smaller than that on the bulk SRAM at 0.4 V supply voltage. The number of SEUs on the SOTB SRAM at 0.4 V operation was roughly equivalent to that on the bulk device at 1.0 V. On the other hand, the numbers of measured MCUs on the SOTB SRAM at 0.4 V and 1.0 V were 0.007x and 0.003x smaller than those on the bulk SRAM, respectively. Therefore, SOTB SRAM can achieve more than two orders of magnitude lower SER with ECC.

Fig. 7 shows the MCU rates for each number of simultaneous bit flips in the SOTB and bulk SRAMs at the incident angle of $0°$. As the number of bit flips increases, the number of measured MCUs quickly decreases in the SOTB SRAM, while it slowly decreases in the bulk SRAM. Even the MCU rate of simultaneous 10-bit flips in the bulk SRAM is higher than the MCU rate of 2-bit flips in the SOTB SRAM. In terms of MCU, SOTB is superior to bulk since MOS transistors are isolated by the BOX layer in SOTB, and the charge sharing and parasitic bipolar action do not occur.

This MCU mechanism due to successive multiple hits of sensitive volumes makes upset classification difficult in
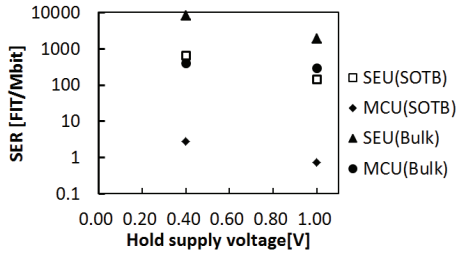
Fig. 6. Measured neutron-induced SEU and MCU vs. supply voltage (0°). Each error bar indicates the standard deviation of the obtained upsets.
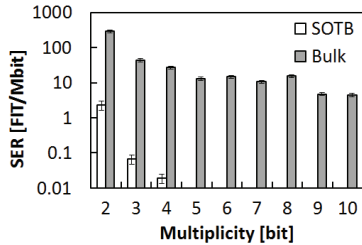

Fig. 7. Measured neutron-induced MCU rate as a function of the number of bit flips in the SOTB and bulk SRAMs at 0.4 V and 0 degree.
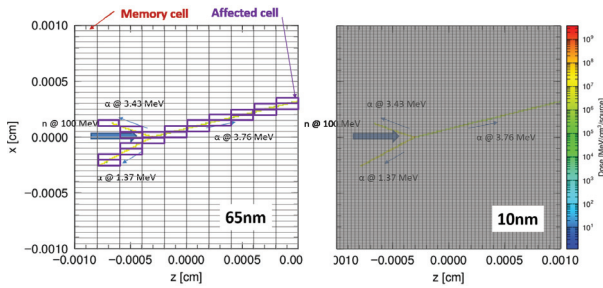

Fig. 8. Number of cells affected by ions increases significantly for the same nuclear reaction from 65 nm to 10nm SRAMs.

TABLE I
MBU RATE IN SRAM CHIPS [FIT/MBIT].

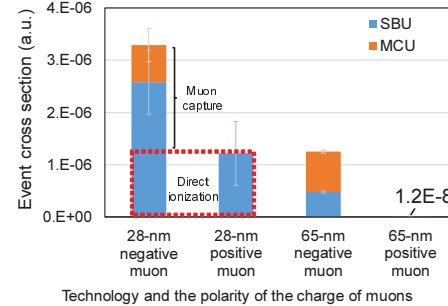|	Bulk		SOTB	
	0.4 V	1.0 V	0.4 V	1.0 V
w/o ECC 1-col	737.33	611.30	0.62	0.09
w/ ECC 1-col	573.92	503.13	0.19	0.05
w/ ECC 2-col	3.88	0.56	0.02	–
w/ ECC 4-col	0.05	–	–	–

Fig. 9. Comparison of SEU event cross sections in 28-nm and 65-nm SRAMs. The error bar stands for one standard deviation.

2-col MUX interleaving achieves around 200 x and 1,000 x reduction in bulk device at 0.4 V and 1.0 V, respectively.

Let us touch on the overhead of the 2/4-col MUX interleaving. References [17]–[19] report an increase in power consumption with the increase in interval distance. This is mainly due to 'half selected' transistors of different words in the same word line [18]. Reference [19] shows an increase in access power of 30 % and 110 % from 1-col to 2-col and 4-col MUX interleaving, respectively, for a 4-MB 8-bank cache.

C. Muon-induced Soft Error in SRAM

As mentioned in Section II, muon-induced soft error could be significant in the figure. Fig. 9 shows our measurement results [8], where the cross section is proportional to the probability per bit. We observe that negative muon-induced SEU event cross section is 2.3 X larger at 0.6 V in 28-nm node and 104.3 X larger at 0.9 V in 65-nm node than positive muon-induced ones. It is confirmed that negative muons have higher error-inducing ability than positive muons at both 28-nm and 65-nm nodes. As for technology comparison, Fig. 9 shows that the negative muon-induced SEU event cross section increases 2.8 X from 65-nm to 28-nm. Meanwhile, the increase in positive muon-induced cross section would be 101.5 X, and the total increase including negative and positive muon-induced SEU event cross sections would be 3.6 X from 65-nm to 28-nm technology nodes. The event cross section difference of negative and positive muon is supposed to be due to capture reaction since positive muons are thought to be able to induce SEUs only by direct ionization. An important message is that the muon-induced SER per bit is increasing while neutron-induced SER per bit is decreasing [20], which indicates the contribution of muon could increase in the future.

simulation in two ways. First, as the technology becomes finer, each ion hits more sensitive volumes, as illustrated in Fig. 8. Second, the secondary ions traveling parallel to the chip surface pass through not only off transistors but also on transistors. Ref. [16] points out that the upset classification based on the charge deposited to on transistors in addition to off transistors improves the accuracy. Machine learning-based classifier construction proposed in [16] could be effective for fast and accurate MCU estimation.

B. MBU Rate with/without ECC and Interleaving

Next, we exemplify the MBU rate with and without the application of ECC and interleaving. For ECC application, we consider the most popular code of single bit correction double error detection (SECDED). SECDED means a 1-bit error along the WL is assumed to be mitigated whatever the number of upsets in at the BL. In this case, single bit upsets (SBUs) will be mitigated as well as MCUs only in BLs. For interleaving, 1-col (no interleaving), 2-col (2-bit interval), and 4-col (4-bit interval) MUX are considered. By applying interleaving w/ ECC, even if a successive 2-bit upset along the WL, they are corrected separately due to being in different words.

Table I lists the MBU rate showing that with ECC and interleaving, the MBU rate decreases compared to that without ECC. Another observation is that sole application of ECC is effective (2 x decrease), but a combination of ECC and small

IV. ERROR SOURCE ANALYSIS IN DIGITAL CHIP

A. Estimation in Processor Chips

In this section, we estimate chip-level SER using the measured error rates [3]. Fig. 10 shows the structure of a high-performance processor with large amount of cache memory,

TABLE II
SER DATA FOR SOTB CIRCUITS.

Voltage [V]	0.4	0.5	0.6	0.8	1.0	1.2
SET [FIT/Mbit]		(0.06)			(0.06)	
SBU [FIT/Mbit]	375	(314)			128	
MBU [FIT/Mbit]	0.6	(0.45)			0.1	
FF [FIT/Mbit]	29.5	(26.8)	26.2	16.2	(14.0)	11.0

TABLE III
SER DATA FOR BULK CIRCUITS.

Voltage [V]	0.4	0.5	0.6	0.8	1.0	1.2
SET [FIT/Mbit]		7.02			(7.02)	
SBU [FIT/Mbit]	1817	(1464)			498	
MBU [FIT/Mbit]	737	(715)			611	
FF [FIT/Mbit]		(1400)	1150	650	620	360
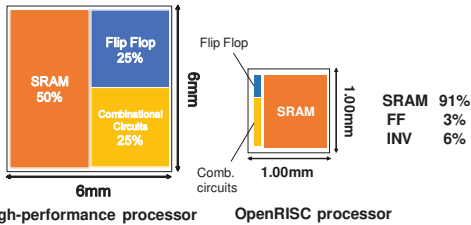
Fig. 10. Structure of high-performance and OpenRISC processor.

(a) w/o ECC   (b) w/ ECC
Fig. 11. Chip-level SER w/o and w/ ECC [FIT/Chip].

(a) w/o ECC

(b) w/ ECC
Fig. 12. Contributions of SRAM, FF, and combinational circuit to chip-level SER w/o and w/ ECC.
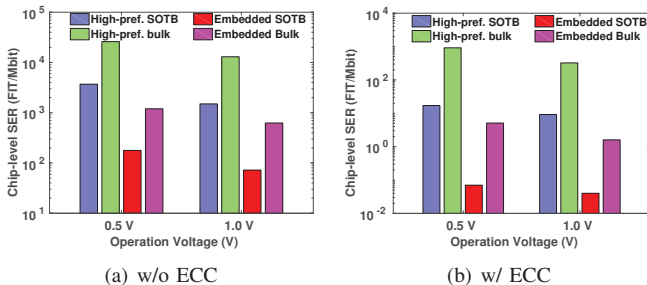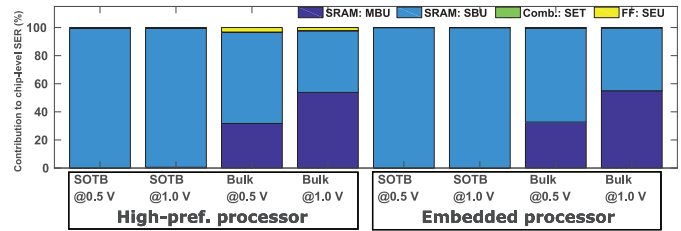
larger register files and deeper pipelines, where 50 % core area is occupied by SRAM, 25 % is occupied by FF and the remaining 25 % is occupied by combinational circuit. The core area is 36 mm². The capacity of SRAM is 11.79 Mbit, and the numbers of FFs and inverters are 0.61 M and 5.02 M, respectively. As for OpenRISC 1200, the SRAM size for the cache is 0.56 Mbit. To estimate the number of cells, we synthesized the RTL files with a standard cell library. The number of FFs is 24 k, and the number of combinational cells is 1.10 M. The chip size and area portions of OpenRISC 1200 are also shown in Fig. 10. The data used in the chip-level SER estimation is listed in Tables II and III. The numbers in the parenthesis are obtained by fitting, and the others are measured values.

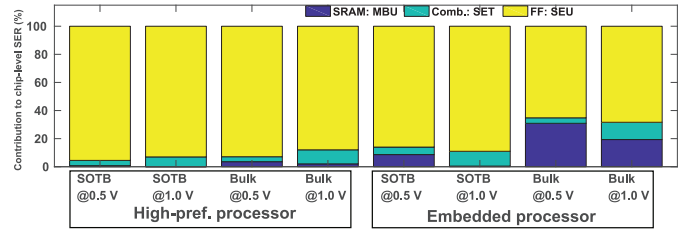Chip-level SER SER_{chip} is calculated as

$$SER_{chip} = (SER_{SBU} + SER_{MBU}) \times N_{SRAM} + SER_{SEU} \times N_{FF} + SER_{SET} \times N_{INV} \quad (1)$$

where N_{SRAM} is the number of SRAM bits in a chip, and N_{FF} and N_{INV} are the number of FFs and inverters in a chip. To estimate the maximum contribution of SET to chip-level SER, logical, temporal and electrical masking are not considered in this calculation

The calculated chip-level SER without ECC is shown in Fig. 11(a). The overall SERs of SOTB chip are 6.0 x and 7.7

x lower than those of bulk chip at 0.5 V and 1.0 V, respectively. Fig. 12(a) also shows a common tendency for both processors that soft error in SRAM dominates in the total chip-level SER. In SOTB chip, more than 99 % errors occur in SRAM, and other FF SEU and SET are negligible. Similarly, more than 95 % errors occur in SRAM in bulk chip.

Next, we apply ECC to SRAM. In this case, the MBU rates with ECC and 2-col interleaving calculated in the previous section are used. The chip-level SER with ECC is shown in Fig. 11(b). In the high-performance processor of SOTB, the chip-level SER is reduced by two orders magnitude while it is reduced by one order of magnitude in bulk chip because the MBU rate is much lower than the SBU rate in SOTB chip. Consequently, the SERs of SOTB chip are 53 x and 34 x lower than those of bulk chip at 0.5 V and 1.0 V, respectively. ECC provides more powerful mitigation to embedded processors. Fig. 12(b) shows the decomposition of chip-level SER. With ECC, the proportion of FF becomes dominant in all the embedded and high-performance processors of SOTB and bulk, while MBU in SRAMs still takes around 20 % in bulk embedded processors. In high-performance processor of SOTB, the contribution of FF reaches 95 % at 0.5 V while that in embedded processors also contributes to 60 % at least. Therefore, radiation-hard FF is helpful to further reduce chip-level SER since ECC is not applicable to FFs in general, where various types of radiation-hard FF are found in [4]. On the other hand, the SET contribution ranges from 4.55 % to 16.67 %. Considering that temporal and logical masking are ignored for upper bound estimation, we can conclude that SET cannot be a primary concern for both the high-performance and embedded processors. This result is consistent with the simulation-based result showing a maximum ratio of 2 % in [21].

B. Hardware Measurement in GPU

One of reliability demanding applications is autonomous driving, and it requires a huge amount of computation on GPU. Therefore, soft error rate of GPU, especially, application-level error rate is drawing a lot of attention, and several experiments are performed and reported [22], [23]. However, commercial
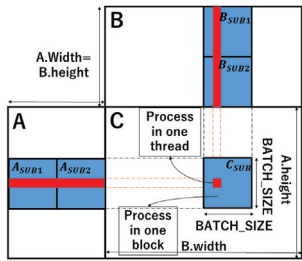
Fig. 13. Matrix multiplication.

TABLE IV
PROGRAMS FOR MATRIX MULTIPLICATION.

Prog.	A.height	B.width	BATCH_SIZE	Block size	Thread size	Run time [ms]
1	512	512	32	256	1,024	0.325
2	32	32	32	1	1,024	0.368
3	512	512	1	262,144	1	64.8

GPU has difficulty in error rate characterization and estimation since the circuit structure is only disclosed partially. For example, the numbers of cores and registers and the sizes of shared and cache memories are available since they are necessary for programmers to develop applications. Then, SEU rates of those visible register files and memories are characterized in [24]. On the other hand, pipeline registers, FFs, and registers in schedulers and dispatchers, etc. must exist in GPU, but their counts are unknown. For estimating application-level error rates, the contributions of those unknown components must be characterized, but such information is not reported so far.

For investigating the contribution from the unknown components, we carried out a preliminary irradiation test and measured the SEU rates of known memory components and the silent data corruption (SDC) error rates of three programs of matrix multiplication having different hardware resource utilization on Nvidia Tesla P4 GPU under neutron irradiation [25]. Then, we estimate the SDC rates assuming only the visible memory components contribute to the SDCs. The comparison between the measured and estimated SDC rates indicates that more than half SDCs come from unknown components.

Fig. 13 illustrates single-precision matrix multiplication achieving $C = A \times B$. The program is implemented, referring to a sample CUDA code provided by Nvidia. CUDA programming can specify the number of threads and blocks for the program execution. $Csub$ region in Fig. 13 is calculated within one block and one value in $Csub$ is calculated in one thread. The size of $Csub$ is defined by BATCH_SIZE × BATCH_SIZE, and consequently BATCH_SIZE × BATCH_SIZE threads will be executed in a block. When the program is executed on GPU, each block is mapped to an SM, and each thread in the block is mapped to a core inside the SM. Schedulers and dispatchers in the GPU chip are responsible for allocating SMs and cores. Depending on the matrix size and BATCH_SIZE, the hardware resource allocation varies.

Table IV lists the prepared programs having different parameter settings. Fig. 14 illustrates the hardware resource usage, where the black components are active and the white ones are idle. Program 1 fully utilizes hardware resources,
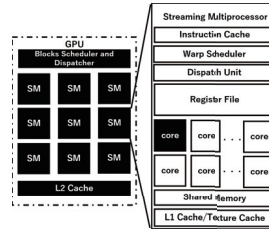
TABLE V
MEMORY USAGE.

Prog.	L2 cache [KB]	Shared memory [KB]	Registers [KB]	Total [KB]
1	2,048	640	8,320	11,008
2	0	640	208	848
3	2,048	0.01	8.125	2,056
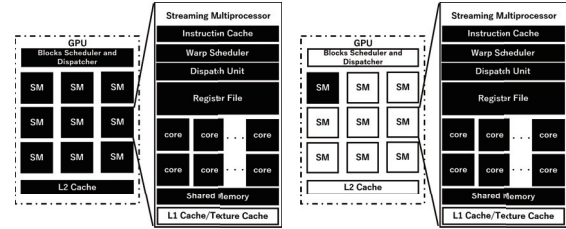
(a) Program 1   (b) Program 2

(c) Program 3

Fig. 14. Used hardware resources in GPU.

i.e., SMs and cores, since the numbers of blocks and threads are larger than the numbers of SMs and cores per SM, respectively. On the other hand, Program 2 has only one block, and hence only one SM is utilized. Program 3, conversely, has many blocks, while each block includes only one thread.

Fig. 15 shows the SDC cross sections for the three programs of matrix multiplication. We can see that the SDC cross section is different depending on the hardware resource usage, and that of Program 1 is the largest. As explained with Fig. 14, Program 1 fully utilizes the hardware resource, and hence the largest SDC cross section is reasonable. Unfortunately, due to the uncertainty, Program 2 and Program 3 cannot be compared.

Next, we estimate the SDC rates taking into account the error rates of used visible memories, i.e., L2 cache, shared memory, and register file. Here, the SEUs occurring in those memories do not necessarily contribute to the errors of matrix multiplication results due to various-level masking effects. On the other hand, here, we assume all the SEUs contribute to the SDC as the worst case, and we estimate the SDC rates of the three programs, which are also shown in Fig. 15. We can see that for all the programs, the measured SDC rate is higher than the estimated one even with the worst assumption, which reveals that the errors occurring in circuit components other than the visible memories, such as pipeline registers, schedulers, etc., contribute to more than half of the overall SDC rate.

V. HIGH-LEVEL COUNTERMEASURE EXAMPLES

Improving soft error immunity is studied at various levels, such as system, architecture, software, circuit, and device levels. At the system level, redundancy based fault tolerance, such as triple modular redundancy (TMR), is often adopted in
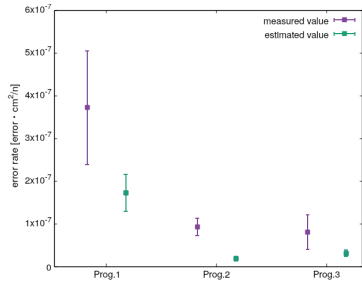
Fig. 15. Measured and estimated error rates of matrix multiplication.

mission-critical applications. At the architecture level, hardware instruction retry in a microprocessor [26], fine-grained TMR in reconfigurable architecture [27], etc. are studied. Ref. [28] demonstrates various levels of reliability can be attained by the reconfigurable architecture through an irradiation test. At the software level, fine-grained code duplication and check code insertion is a popular approach to detect soft errors [29]. More recently, the impact of Single Event Upset (SEU) induced parameter perturbation (SIPP) on neural networks is studied [30]. The impact of SIPP on different types of bits in a floating-point parameter, layer-wise robustness within the same network, and impact of network depth are experimentally evaluated. Based on the observation, two remedy solutions to protect DNNs from SIPPs are presented, which can mitigate accuracy degradation from 28% to 0.27% for ResNet with merely 0.24-bit SRAM area overhead per parameter.

VI. CONCLUSION

This paper discussed the mechanism of soft error occurrence, the impact of soft error on digital systems, and countermeasures aiming at developing cost-effective countermeasures for reliability-demanding applications. ECC is the most popular error mitigation technique, and its effectiveness is evaluated with interleaving in bulk and FD-SOI SRAMs. We also point out that muon-induced SER is increasing while neutron-induced SER is decreasing. We next estimated the contributions of SRAM, FF, and combinational circuit to the chip-level SER. When ECC and interleaving are applied to SRAM, radiation-hard FF is demanded for further reliability improvement. We also exemplified the difficulty in SER characterization of commercial chips using GPU example. Soft errors not only in datapath but also in undisclosed control units are playing an important role in actual VLSI systems. Recently reported countermeasures at architecture and application levels are introduced. Our future works include assessing the reliability of future devices and systems exploiting irradiation experiments, simulation, and emulation.

REFERENCES

[1] IHS Markit, "IoT trend watch 2018," https://cdn.ihs.com/www/pdf/IoT-Trend-Watch-eBook.pdf, 2018.
[2] E. Ibe, "Terrestrial Radiation Effects in ULSI Devices and Electronic Systems," *Wiley-IEEE Press*, 2015.
[3] W. Liao and M. Hashimoto, "Analyzing Impacts of SRAM, FF and Combinational Circuit on Chip-Level Neutron-Induced Soft Error Rate," *IEICE Trans. on Electronics*, E102-C(4), pp. 296–302, April 2019.
[4] M. Hashimoto et al., "Characterizing SRAM and FF Soft Error Rates with Measurement and Simulation," *Integration, the VLSI Journal*, 69, pp. 161–179, November 2019.
[5] S. Abe and T. Sato, "Shielding effect on secondary cosmic-ray neutron- and muon-induced soft errors," *Proceedings of European Conference on Radiation and Its Effects on Components and Systems (RADECS)*, 2016.
[6] R. Silberberg, C. H. Tsao, and J. R. Letaw, "Neutron Generated Single-Event Upsets in the Atmosphere," *IEEE Trans. Nucl. Sci.*, vol. 31, no. 6, pp. 1183–1185, 1984.
[7] B. D. Sierawski et al., "Muon-Induced Single Event Upsets in Deep-Submicron Technology," *IEEE Trans. Nucl. Sci.*, vol. 57, no. 6, pp. 3273–3278, Dec. 2010.
[8] W. Liao et al., "Negative and Positive Muon-Induced SEU Cross Sections in 28-nm and 65-nm Planar Bulk CMOS SRAMs," *Proc. IRPS*, 2019.
[9] S. Manabe et al., "Negative and Positive Muon-Induced Single Event Upsets in 65-nm UTBB SOI SRAMs," *IEEE Trans. Nuclear Science*, 65(8), pp. 1742–1749, Aug. 2018.
[10] W. Liao et al, "Measurement and Mechanism Investigation of Negative and Positive Muon-Induced Upsets in 65-nm Bulk SRAMs," *IEEE Trans. Nuclear Science*, 65(8), pp. 1734–1741, Aug. 2018.
[11] T. Sato, "Analytical Model for Estimating the Zenith Angle Dependence of Terrestrial Cosmic Ray Fluxes," *PLOS ONE*, vol. 11, no. 8, e0160390, 2016.
[12] T. Sato, "Analytical model for estimating terrestrial cosmic ray fluxes nearly anytime and anywhere in the world: Extension of PARMA/EXPACS," *PLOS ONE*, vol. 10, no. 12, e0144679, 2015.
[13] E. Ibe et al., "Spreading Diversity in Multi-cell Neutron-Induced Upsets with Device Scaling," in *Proc. CICC*, pp. 437–444, 2006.
[14] S. Hirokawa et al., "Multiple Sensitive Volume Based Soft Error Rate Estimation with Machine Learning," *Proc. RADECS*, 2016.
[15] Y. Yamamoto et al., "Ultralow-voltage operation of Silicon-on-Thin-BOX(SOTB) 2Mbit SRAM down to 0.37 V utilizing adaptive back bias," in Proc. *Symposium on VLSI Circuits*, pp. T212–T213, 2013.
[16] M. Hashimoto, W. Liao, and S. Hirokawa, "Soft Error Rate Estimation with TCAD and Machine Learning," *Proc. SISPAD*, pp.129–132, 2017.
[17] S. Baeg, S. Wen, and R. Wong, "SRAM Interleaving Distance Selection With a Soft Error Failure Model," *IEEE Trans. Nuclear Science*, vol. 56, no. 4, pp. 2111–2118, 2009.
[18] S. Kim and M. R. Guthaus, "Low-power multiple-bit upset tolerant memory optimization," *Proc. ICCAD*, pp. 577-581, 2011.
[19] J. Kim et al., "Multi-bit error tolerant caches using two-dimensional error coding," *Proc. MICRO*, pp. 197–209, 2007.
[20] N. Seifert et al., "Soft Error Rate Improvements in 14-nm Technology Featuring Second-Generation 3D Tri-Gate Transistors," *IEEE Trans. Nuclear Science*, vol. 62, no. 6, pp. 2570-2577, Dec. 2015.
[21] B. Gill, N. Seifert, and V. Zia, "Comparison of alpha-particle and neutron-induced combinational and sequential logic error rates at the 32nm technology node," *Proc. IRPS*, pp. 199–205, 2009.
[22] C. Lunardi et al., "On the Efficacy of ECC and the Benefits of FinFET Transistor Layout for GPU Reliability," *IEEE Trans. Nuclear Science*, vol. 65, no. 8, pp. 1843-1850, Aug. 2018.
[23] F. F. d. Santos et al., "Analyzing and Increasing the Reliability of Convolutional Neural Networks on GPUs," *IEEE Trans. Reliability*, in press.
[24] D. A. G. de Oliveira et al., "Evaluation and Mitigation of Radiation-Induced Soft Errors in Graphics Processing Units," *IEEE Trans. Computers*, vol. 65, no. 3, pp. 791-804, 1 March 2016.
[25] K. Ito et al., "Characterizing Neutron-Induced SDC Rate of Matrix Multiplication in Tesla P4 GPU," *Proc. RADECS*, 2019.
[26] R. Kan et al., "The 10th Generation 16-Core SPARC64 Processor for Mission Critical UNIX Server," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 1, pp. 32-40, Jan. 2014.
[27] D. Alnajjar et al., "Implementing Flexible Reliability in a Coarse Grained Reconfigurable Architecture," *IEEE Trans. VLSI Systems*, vol. 21, no. 12, pp. 2165 – 2178, Dec. 2013.
[28] H. Konoura et al., "Reliability-Configurable Mixed-Grained Reconfigurable Array Supporting C-Based Design and Its Irradiation Testing," *IEICE Trans. on Fundamentals*, E97-A(12), pp. 2518–2529, Dec. 2014.
[29] B. Nicolescu and R. Velazco, "Detecting soft errors by a purely software approach: method, tools and experimental results," *Proc. DATE* , pp. 57–62, 2003.
[30] Z. Yan et al.,"When Single Event Upset Meets Deep Neural Networks: Observations, Explorations, and Remedies," *Proc. ASP-DAC*, to appear.