Soft-Error-Resilient FPGAs Using Built-In 2-D Hamming Product Code

Sang Phill Park, *Student Member, IEEE,* Dongsoo Lee, and Kaushik Roy, *Fellow, IEEE*

***Abstract*—Radiation-induced soft error rate (SER) degrades the reliability of static random access memory (SRAM)-based field programmable gate arrays (FPGAs). This paper presents a new built-in 2-D Hamming product code (2–D HPC) scheme to provide reliable operation of SRAM-based FPGAs in hostile operating environments such as space. Multibit error correction capability of our built-in 2-D HPC can improve the reliability, and hence, system availability, by orders of magnitude. Simulation results show that the large number of error correction capability of 2-D HPC can recover configuration bits without depending on an external memory preserving a golden copy of the configuration bits. To provide efficient 2-D HPC in a built-in logic, we also propose a new 2-D SRAM buffer. Using the proposed multibit error correction scheme, system availability of an SRAM-based FPGA can be more than 99.9999999% with SRAM cell failures in 1 billion h of operation of 7.**

***Index Terms*—Error correction coding, field programmable gate arrays, product codes, soft errors.**

I. Introduction

RECONFIGURABLE static random access memory (SRAM)-based field programmable gate arrays (FPGAs) are widely adopted in various application domains due to their fast time-to-market, high density and programmability, and cost-effectiveness. Despite aforementioned advantages of SRAM-based FPGAs, its widespread use in mission-critical space applications is limited due to its susceptibility to single-event upsets (SEUs). Impact of highly energized particles (e.g., protons, neutrons, and alpha particles) on sensitive locations of circuits results in SEUs. In SRAM-based FPGAs, the functionality is specified by the contents of the configuration memories. However, SEU can alter a configuration bit in an FPGA which may result in a permanent malfunction of the mapped program. Researchers have experimentally investigated the impact of radiation effects in FPGAs on the surface of earth and space. Rapid elevation of soft error rate (SER) has been observed in high-altitude experiments [1]–[4]. For example, experiments in 1000-km orbit has reported multiple failures of FPGA systems in a day [2].

Aggressive device scaling also increases the vulnerability to soft errors. Shrinking feature sizes and supply voltage scaling leads to a reduction in stored charge per device. This increases the likelihood of lower energy particles causing a soft error. Besides, increased circuit density may also lead to higher system SER [1], [5]. Hence, it is clear that increased SER threatens the stable operation of a system, especially for space applications, when no mitigation techniques are employed.

Most of the memory bit cells in SRAM-based FPGA are configuration bits, occupying more than 98% of memory [6], [7]. Thus, the probability of SEU-induced errors in configuration bits is much higher compared with that in user data [block random access memories (RAMs)]. To mitigate susceptibility to SEUs in such memory cells, redundant design techniques such as dual-module redundancy (DMR) and triple-module redundancy (TMR) can be used for mission-critical applications [2], [6]–[9]. In a TMR-based system, there are three copies of same module, and the final output is determined by a majority voting system. However, a TMR-based system can accumulate SEU-induced errors and produce erroneous outputs when two copies of any module in a system fail. Therefore, mitigation techniques are still required to ensure correct and continuous operation of the system. Configuration scrubbing is one such technique which performs periodical rewriting of the valid bits over configuration memories [7]–[9]. This scheme can handle multiple upsets but requires continuous access to an external storage device which contains the original (golden) program. An alternative technique known as configuration read-back enables detecting erroneous bits using cyclic redundancy check (CRC). Upon detection of an error, FPGAs have to be reprogrammed. Some of high-end commercial FPGAs such as Xilinx Virtex 5 and 6 series have dedicated built-in logics to support faster detection and correction of bit upsets [8]. These techniques rely on external nonvolatile memory devices which also must be protected against SEU. Such radiation-hardened memories can be expensive. Additionally, excessive access delay for the external memory can degrade the performance of the system [6]–[9].

In this paper, we propose an efficient built-in 2-D Hamming product code (2-D HPC) scheme which provides large performance improvement in multibit error detection and correction. Based on the investigation of existing soft error mitigation techniques of SRAM arrays and their limitations in the state-of-the-art FPGAs, we address the need of a new 2-D error correction (EC) scheme for FPGA configuration data. Hardware implementation of the proposed scheme includes a novel SRAM architecture which provides a single-cycle bitwise column read of the contents of memory array. This unique feature enables simplification of computation and hardware costs in 2-D HPC. In comparison to other multidimensional product codes used in applications for communication and massive storage, we show that the proposed 2-D HPC is very efficient in hardware costs while providing enough EC performance. The logic size of 2-D

Manuscript received June 09, 2010; revised October 02, 2010; accepted November 08, 2010. Date of publication January 10, 2011; date of current version January 18, 2012. This research was supported in part by NASA.

The authors are with Purdue University, West Lafayette, IN 47907 USA (e-mail: sppark@purdue.edu; dslee@purdue.edu; kaushik@purdue.edu).

Digital Object Identifier 10.1109/TVLSI.2010.2095435

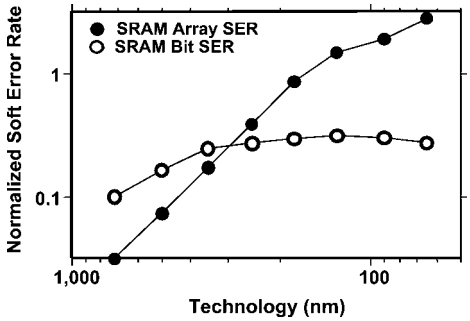

Fig. 1. Normalized SRAM bit and array SERs as a function of technology scaling [5].

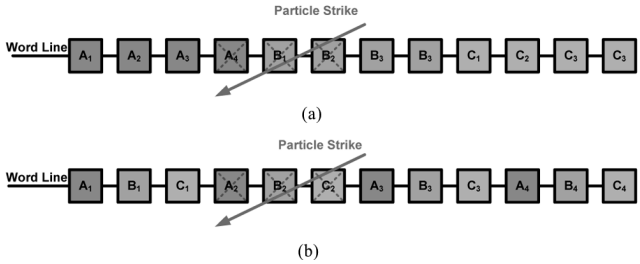

Fig. 2. Simplified example of bit-interleaving in the case of multibit upsets in conventional memory array. Cells with cross marks represent cells with bit upsets and ECC is applied to a word size. (a) Three words stored in a row without interleaving. (b) Three words stored in a row with interleaving.

HPC is comparable to that of SECDEC for 1024-bits frame size while providing correction of more than ten errors in the success rate of 0.99. Due to its low hardware cost and superior performance in multibit EC, it is highly suitable for *in situ* EC to provide enhanced reliability of FPGAs. In addition, elimination of excessive external memory access improves the availability of the system.

The remainder of this paper is organized as follows. In Section II, we discuss SEUs in scaled technologies. We also discuss the necessity of built-in multibit EC in SRAM-based FPGAs. In Section III, we describe our proposed 2-D HPC with detailed hardware implementation. Simulation results on reliability and availability improvement are given in Section IV. Finally, Section V concludes the paper.

II. Soft Errors in Scaled Technologies

In the past decades, scaling of devices has been driven by demands for higher functionality, higher density, lower cost and lower power. Aggressive feature size and supply voltage scaling has resulted in reduction of critical charge (Qcrit) in memory cells [5]. Qcrit is defined as the minimum charge capable of flipping the stored bit in a memory cell. Intuitively, smaller Qcrit would result in higher SER. However, it has been observed that SRAM bit SER has started to saturate and is expected to decrease in-deep submicrometer regimes (Fig. 1). Saturation in the supply voltage scaling and decrease in junction collection efficiency has resulted in the saturation of SRAM bit SER. Notwithstanding the saturation in SRAM bit SER, the SRAM system SER has increased dramatically with each technology generation. The increase in system SER can be attributed to the exponential growth in SRAM integration density with device scaling and has become a great concern for future technology nodes [5].

In addition to the increase in SER due to aggressive technology scaling, cosmic radiation is another significant factor in accelerating SER in high-altitude and space applications. While SER induced by alpha particles can be suppressed by purification of packaging materials, it is challenging to address SER due to cosmic neutrons. The intensity of cosmic ray flux increases by an order of magnitude every 10 000 ft [1]–[4]. The same trend has been experimentally shown in the Rosetta Experiment using SRAM-based FPGAs [1]. Also, a TMR experiment for low-earth orbital path have shown upsets at a rate of one per hour, spread across three Xilinx V1000 devices, with each device having about 6.5-Mbits configuration memory [7].

A. Multibit Upsets in SRAM-Based FPGAs

As scaling of devices enables higher density, higher performance, low cost, and low power in ASICs, the same trend has been observed in FPGAs. Moreover, higher integration density is desirable in FPGAs due to the presence of excessively large programmable circuitries (e.g., configuration logic blocks (CLBs) and routing logic). Since the contents of configuration memory define the functionality, susceptibility to soft error is even more important issue in high-density SRAM-based FPGAs. For instance, a single-bit soft error in configuration memory can result in permanent malfunction of a system. While prevention of SEUs in configuration bits of FPGAs is of paramount importance, the SEU mitigation techniques commonly used in conventional SRAM arrays are not easily applicable to SRAM-based FPGAs .

The most commonly used technique for soft error protection in conventional SRAM arrays is use of Hamming code, known as single double error detection (SECDED), with bit interleaving. While SECDED alone cannot handle multiple upsets, its effectiveness on multibit upsets is greatly improved by bit interleaving [10]–[12]. Fig. 2(a) depicts a simplified example when a single row stores three 4-bit words in a conventional SRAM array. In the case of a particle strike occurrence, bit upsets may occur in a group of bits which are neighboring. Since SECDED only can correct a single bit in a particular word which ECC is applied to, word B cannot be corrected using SECDED technique alone. Bit-interleaving technique shown in Fig. 2(b) can be used to mitigate such multiple bit upsets. Even if all three bits in A2, B2, and C2 are flipped, SECDED can be applied individually to word A, B, and C, and corrected.

However, the bit-interleaving technique with SECDED is not suited for FPGAs mainly due to difference in physical location and access scheme of data. For example, the unit size of configuration bit stream in Xilinx architecture is called a frame, and a frame consists of more than a thousand bits. These bits are programmed through serialized interface, and neighboring bits in a frame are placed physically nearby in a CLB (Fig. 3). Therefore, an interleaving of configuration data cannot be easily applied and its effectiveness is not as efficient as that of a conventional memory array.

Data stored in the SRAM cells in FPGA consist of user data as well as configuration bits. User data is as important as the configuration bits to ensure the correct functionality of a system. Note that the number of configuration bits is more than 98%

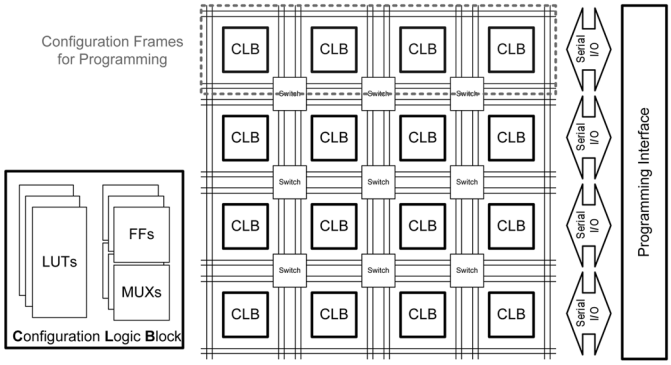

Fig. 3. Simplified diagram of a FPGAs programming interface. SRAM cells are physically placed in CLBs. Each CLB consists of multiple programmable components, and CLBs are programmed by a dedicated interface [13].

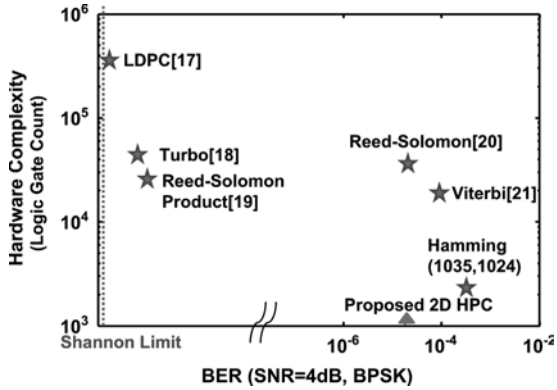

Fig. 4. Hardware complexity (logic gate count) versus BER. BER is estimated using SNR = 4 dB and binary phase shift keying (BPSK) [17]–[21].

of all memory elements in an FPGA [6], [7]. Since the portion of user data is very small compared with that of configuration data and several effective protection techniques are available for user data [6]–[9], this work mainly focuses on protection of configuration memory.

To ensure the integrity of configuration memory data in FPGAs, some recent FPGAs provide background read-back function that performs continuous read of the memory contents without disturbing the function of FPGAs. Dedicated built-in EC and CRC hardware blocks further optimize read-back process [7], [8]. The built-in EC circuit performs SECDED. However, due to the lack of capability to handle multibit errors in a single frame using SECDED, CRC has to be performed simultaneously during read-back process to detect possible multibit errors. In the case of multibit error detection, the reconfiguration process involves excessive delay in accessing external storage device to recover the original configuration data through limited I/Os in FPGAs. For instance, the internal FPGA system clock runs at several hundred megahertz while I/O clock is slower than one tenth of the system clock [6]–[8].

In order to mitigate the above-mentioned drawback of excessive delay in accessing external memory and low performance in ECs, we propose a built-in 2-D HPC scheme. The proposed scheme has the capability of correcting a large number of multibit errors (e.g., at least ten simultaneous errors in 32×32 bit array can be corrected with probability of 99%). Hence, we believe that the proposed 2-D HPC can provide self-recovery of FPGA configuration data without relying on expensive external storage device. Moreover, on-chip EC circuit can eliminate the need for data transfer through the limited FPGAs' I/Os, thereby improving detection and reconfiguration delay.

B. Comparison to Existing Multibit EC Schemes

There exists several other multibit correction algorithms used in communication and massive storage applications such as low-density party check (LDPC), turbo, Viterbi, and Reed–Solomon (RS) codes [15]–[21]. Fig. 4 illustrates hardware complexity and performance in terms of logic counts and bit error rate (BER) of [17]–[21]. These BER results are approximated performance from several implemented designs in the literatures when the input bit stream has SNR equal to 4 dB. The BER provided by Hamming (SECDED) (1035,1024) in this figure shows the performance of conventional built-in ECC in FPGAs. Although performance and size of the hardware can vary depending on its application, most of the multibit EC algorithms are not easily applicable to FPGAs since implementation of such algorithms requires highly complex hardware. In addition, a built-in ECC in FPGAs does not require near-Shannon-limit performance which LDPC and Turbo coding can achieve. As will be evident later in the paper, the enhanced performance of 2-D HPC is sufficiently high to achieve reliable operation of FPGAs since expected SER of SRAMs in FPGAs is a lot lower than error rates in communication application domain.

RS product code and HPC have long latency as shown in Table I. This is mainly because product codes require column directional data accesses to memory array (buffer), and such memory operation is only supported by multiple cycles to perform column direction ECC. Noting that the symbol size of RS code is multibit (usually a byte) and that of HPC is a bit, single-cycle bitwise-column access provided by our proposed 2-D SRAM buffer can greatly improve the performance of 2-D HPC. The detailed discussions of 2-D HPC hardware implementation with 2-D SRAM will be discussed in Section III.

III. Proposed 2-D HPC Algorithm and Architecture

Here, we first analyze the EC efficiency of our proposed 2-D HPC scheme and show reliability improvement in FPGAs.

A simplified example in Fig. 5(a) illustrates how 2-D HPC can achieve multibit correction in a 7×7 bit array using a conventional single-bit EC scheme. Note that we define EC windows as the size of the data [D in Fig. 5(a)], which is protected by parity bits in the following analysis. Even with the presence of three bit errors in a row in the window, all of three errors can be corrected by three SECDED operations applied in column directions. These corrections can be done by using a smart decision-making algorithm to determine where to start/proceed after collecting information about all of the errors in the array. It may lead to optimized throughput of the 2-D HPC scheme. However, hardware implementation of such decision logic can result in additional overhead for EC. Instead, the same error correction performance can be obtained by multiple iterations (a 2-D HPC iteration is a set of row and column directional EC trials) of 2-D HPC. For example, Fig. 5(b), (c), and (e) illustrate how iterative SECDED can potentially correct multibit errors in an array.

TABLE I
COMPARISON BETWEEN THE PROPOSED AND EXISTING MULTIBIT EC SCHEMES [17]–[21]

	Soft-In/Soft-Out(SISO) coding		Product coding		Convolutional coding	Block coding	
	LDPC	Turbo	Reed-Solomon Product	Proposed 2D HPC	Viterbi	Reed-Solomon	Hamming (SECDED)
Hardware Complexity	High	High	Moderate	Low	Moderate	Moderate	Low
ECC Performance	Near-Shannon Limit	Near-Shannon Limit	Very Strong	Strong	Strong	Moderate	Low
Latency	Moderate	Long	Very Long	Moderate	Moderate	Low	Very low
Note	Parallel processing	Exponential increase in complexity	Low throughput	Difficult to implement using conventional Memory	Exponential increase in complexity	Burst error correction	Single bit error correction

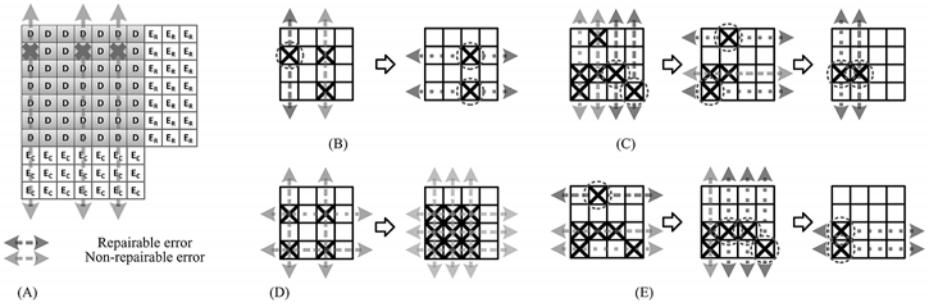

Fig. 5. Examples of 2-D HPC scheme. D, E_R, E_C, and E_E are Configuration bit, Row EC parity bit, and Column EC parity bit, respectively. Blue and red colored trials are successful and unsuccessful trials, respectively. X represents a bit error. (a) 2-D EC with 7×7 window. (b) Successful 2-D EC. (c) Successful 2-D EC in 1.5 iterations. (d) Nonreparable and nondetectable case. (e) Equivalent correction to (c).

TABLE II
AVERAGE REPARABLE ERRORS IN A 1-Mb ARRAY ($P_{success} = 99\%$)

Error Correction Scheme	2D HPC			SECDED
ECC unit Size	32x32b	128x128b	256x256b	1024b
# of Repairable Errors	4981.8	2050.2	255.8	3.7

Also note there are cases where 2-D HPC fails to recover the data, as shown in Fig. 5(d). When there are multibit errors in both directions, 2-D HPC may not be able to recover the data. The probability of such occurrences depends on windows size of 2-D HPC and the expected number of errors in an array. There are cases that nonrepairable errors with 64×64 window can be corrected by 32×32 window. For instance, a 2-D HPC of 64×64 window cannot correct cases of 4-bit errors shown in Fig. 5(d). However, it is possible that the four errors are distributed into four 32×32 array portions in a 64×64 window. Then, all four errors can be corrected by four iterations of 2-D HPC with 32×32 window.

Table II shows the EC performance of 2-D HPC with varying window sizes in a 1-Mb array. In the simulations, errors are uniformly distributed in the array. The result of conventional SECDED with word size of 1 Kb is also provided as a reference, since the frame size of commercial FPGAs is about 1 Kb. Conventional SECDED shows low EC performance mainly due to incapability of multibit EC in a single frame. On the other hand,

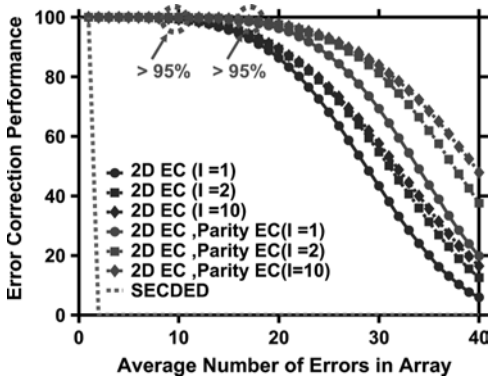

Fig. 6. EC Performance in 32×32 b memory unit. I value in parentheses represents number of 2-D HPC iterations. The results with Parity EC are obtained with additional EC applied to parity bits.

2-D HPC with 32×32 window size can correct ten errors in a frame with 99% of success rate using a perfect decision-making algorithm (or large number of iterations). The simulation result in Fig. 6 shows that one iteration of 2-D HPC can correct ten errors with slightly degraded success rate of 95%. The results have been obtained from one million random samples with consideration of failures in parity data. The errors in parity data may not directly result in a system failure. However, it can increase chances of nonreparable errors. Hence, we assume same importance for all errors. Table II shows that a 1-Mb array with

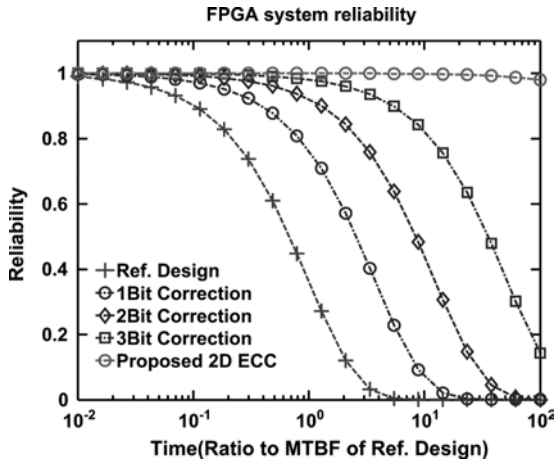

Fig. 7. Reliability improvement using multibit EC.

32×32 2-D HPC can tolerate approximately five errors in a frame to maintain 99% of success rate. It is because the success rate in a frame has to be much higher to guarantee 99% of success rate from an array.

The size of EC and the number of iterations are the design parameters which determine error correction performance of 2-D HPC. As discussed in the previous sections, SER can be exceptionally high for space applications. The results in Table II show that 2-D HPC can achieve robust operation in such hostile environments. To see the effectiveness of 2-D HPC in reliability improvement, simulation results on ISCAS'89 circuit, s38584, are presented in Fig. 7. The reference design is an unprotected circuit (s38584 in this case). We used Xilinx ISE9.0 to obtain mapped design sizes in this simulation [22]. The simulator injects random errors to an array with the same size of configuration memory, and continuously performs error corrections. As the ultimate goal of our proposed scheme is to realize self-recovery without an external golden copy, we only consider the errors in configuration memories including parity memory. Most of protection techniques for SRAM-based FPGAs use continuous monitoring of configuration bit errors using an external device [2], [6]–[8]. The system has to be stopped and reconfigured upon an error occurrence. In this simulation, error correction is performed only using various built-in EC schemes. The FPGAs fail to operate when any nonrepairable error exists in frame data. For comparison, we assume reliability of a design with an external memory is 1. The proposed 2-D HPC based scheme shows almost no degradation (Fig. 7) in reliability. The improvement is achieved due to less likelihood of nonrepairable errors using 2-D HPC scheme. From Fig. 7, we note that the mean time between failure (MTBF) of 2-D HPC protected design is enhanced more than five orders of magnitude compared with that of the reference design.

A. Hardware Implementation of 2-D HPC

Here, we will discuss the considerations for hardware implementation of 2-D HPC with minimal modification of conventional FPGA architecture.

1) Additional Parity Bits: The hardware implementation of 2-D HPC requires additional parity bits for each row and column as shown in Fig. 5(a). According to the theory of Hamming code, SECDED requires $\log N + 1$ additional parity bits [15], [16], where N is the number of bits to protect. Table III shows the memory overhead due to additional parity bits. The required memory overhead decreases with increased size of 2-D HPC. This is due to the logarithmic relationship of the number of parity bits to the protected data as shown above. Note that the parity bits can be precalculated using FPGA design software [8], [14], [22]. The additional parity bits become a part of FPGA configuration data and can be programmed when FPGA is powered on. Therefore, generation and programming of parity bits are not a bottleneck. However, the memory overhead is one of the concerns in hardware implementation as shown in Table III.

TABLE III
PARITY MEMORY SIZE FOR 2-D HPC SCHEME

2D HPC Window	Parity Bits per Row or Column	Total Memory Overhead
32x32b (1 frame)	6	37.50%
64x64b (4 frames)	7	21.88%
128x128b (16frames)	8	12.50%
256x256b (64frames)	9	7.03%

*SECDED requires 11 parity bits per 1Kb data (1% overhead).

TABLE IV
ESTIMATION OF PARITY MEMORY AREA IN XILINX XC5VLX50

2D HPC Window	Parity Memory Area in 65nm (1-port)	Parity Memory Area in Xilinx XC5VLX50 [26]
32x32b (1 frame)	1.944 mm^2	14.85%
64x64b (4 frames)	1.134 mm^2	8.66%
128x128b (16frames)	0.648 mm^2	4.95%
256x256b (64frames)	0.362 mm^2	2.78%

In order to minimize memory overhead, we propose to store the additional parity bits in a dedicated on-chip memory similar to block RAM (BRAM). In conventional FPGAs, BRAM is available for massive user data storage. It provides dual-port capability for high performance user applications [12], [23]. Since the dedicated parity memory does not require high-performance operation, it can be implemented using a single-port SRAM array to further optimize area of the parity memory. Approximated area of dual-port BRAM array is 15% of total device area in a Xilinx XC5VLX50 implemented in 65-nm technology [26]. Table IV shows the estimation of additional area for a dedicated parity memory in the same device. To estimate area of the parity memory, we have used memory cell sizes of normal single-port and dual-port memory in 65-nm technology reported in [24]. As discussed earlier in this section, the size of 2-D HPC window is a design parameter that determines EC performance of a built-in EC. Therefore, parity area overhead can be utilized for higher system reliability.

In addition, conventional built-in EC of the memory can protect parity bits. Although an error in the parity bits does not affect the function of FPGA, it increases chances of nonrepairable error in configuration data, in the case when there is additional

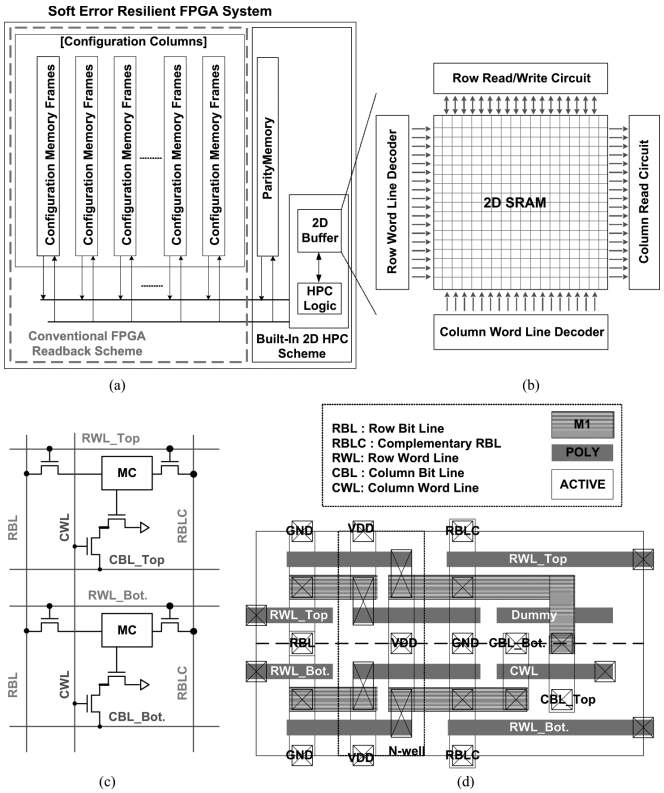

Fig. 8. Proposed FPGA architecture with built-in 2-D HPC and 2-D buffer design. 2-D SRAM buffer which enables bitwise column direction access is presented in detail. (a) FPGA architecture with built-in 2-D HPC scheme. (b) Periphery layout of 2-D SRAM array. (c) Schematics of two memory cells. (d) Layout of two memory cells equivalent to (c).

error in the same row or column to which the parity is applied. Therefore, additional protection on parity bits can further improve error correction performance as shown in Fig. 6.

The proposed scheme requires only few changes to the conventional FPGA architecture. In our hardware implementation, a buffer memory temporarily stores a copy of frame data and parity data to perform 2-D HPC. Configuration bits are loaded through conventional read-back bus, and parity bits are taken from a dedicated parity memory array (Fig. 8). Hence, it is expected that our scheme can easily substitute the SECDED/CRC logic in conventional FPGAs to provide enhanced EC.

2) Syndrome Generation: In hardware implementation of SECDED (Hamming code), syndrome generation is the major part of computation. This requires large number of XOR (exclusive OR) gates and the number varies depending on the code size and the topologies of the XOR tree. Noting that 2-D HPC performs iterative SECDED with narrowed bit width, it is expected that the number of XOR gates is much less in 2-D HPC circuit. To compare the computation effort of syndrome generation in 1-D and 2-D cases, Table V shows the number of XOR gates and operations in each 1-D and 2-D cases approximately. An (n, k) hamming code requires input XOR function for each syndrome generation and there are syndromes, where n is code bit width (data + parity), and k is data bit width [25]. Under the condition that only two-input XOR gate can be used in generation of syndrome bits, the total number of two-input XOR gate is shown in

TABLE V
COMPARISONS IN SYNDROME GENERATOR COMPLEXITY

	1D EC (conventional) (n,k) = (1035,1024)	2D HPC (n,k) = (38,32)
# of 2-input XOR gates to generate single syndrome bits	517	18
# of syndrome bits	11	6
# of 2-input XOR gates in syndrome generator hardware	**5687**	**108**
# of 2-input XOR operations to complete 1 frame size EC	5687	6912

Table V. The results show that 2-D HPC requires a very small number of XOR gates in comparison to conventional EC. This is because the syndrome generator logic is repeatedly used in the proposed 2-D HPC circuit and the detailed operation will be discussed in the following section. As a result, the total numbers of XOR operations to complete EC for a frame is similar in the both schemes. Results show that the logic size can significantly reduce and the logic energy consumption can increase by 21% in 2-D HPC scheme. However, the increase may not impact total power consumption since the programmable logic dominates area and power in FPGAs.

3) New Memory Buffer for Column Access: Fig. 8(a) shows the proposed FPGA architecture using built-in 2-D HPC logic. Since the proposed 2-D HPC requires row and column accesses to the frame data, a frame buffer can be used to provide optimized throughput during error correction operation. It is important to note that conventional memory can only provide one directional access of its data. As a result, column direction access requires multiple transpose of the EC buffer contents and could result in increased computation time of 2-D HPC. Thus, to optimize the 2-D HPC repair time further, a 2-D memory is proposed as shown in Fig. 8(b)-(d). The proposed memory can provide **bitwise-column** direction access to its contents. The cell structure is a modification of conventional two-port 1R/1W SRAM cell (8T) [12]. Since the two ports are independently used in conventional two-port memory, modifications can be made since one port is used for row read/write access and the other port is used for column read access as shown in Fig. 8(b) and (c). Thin cell layout of the proposed cell is presented in Fig. 8(d).

Let us take an example of 2-D HPC applied to one frame as shown in Table III. In commercial FPGAs, a single frame size is approximately 1 K bits, and the serialized read-back produces 32 bit of frame data every clock cycle [8], [14]. We reconstruct the frame data into a 32×32 bits array using the conventional read-back process. Noting that the proposed 2-D SRAM array has one write port, correction of an erroneous bit has to be through the write port of the memory. The operation of proposed 2-D HPC circuit can be described as follows.

Step 1) 32 bits of frame data are taken from the serial read-back bus, and EC parity bits are taken from parity memory.

Step 2) After SECDED is performed on 32-bit word, store the data into 2-D frame buffer.

Step 3) Repeat steps 1) and 2) until the buffer is full.

TABLE VI
COMPARISON IN OPERATION CYCLES REQUIRED TO COMPLETE

	1D EC (conventional)	2D HPC
# of cycles to load buffer	32	32 (row EC is performed while the buffer is loaded)
# of cycles to perform EC	1 (assuming all syndrome generation is done in a cycle)	32 (column EC is performed after the buffer is fully loaded)

Step 4) Perform column read and SECDED in column direction.
Step 5) If no error is detected, repeat step 4) until the buffer is fully scanned.
Step 6) Record the row address identified by SECDED and the column address of current operation.
Step 7) Perform correction of erroneous bit through row read/write port based on row and column address.
Step 8) Repeat step 4) until the buffer is fully scanned in column direction.
Step 9) In the case of additional iteration required, proceed to step 10). Otherwise, go to step 12).
Step 10) Perform row SECDED though read/write port of frame buffer.
Step 11) Continue row SECDED until the buffer is fully scanned and corrected, and then proceed to step 4).
Step 12) If any bit had been corrected, write back repaired frames in 2-D buffer through programming bus.
Step 13) Go to step 1) for next set of frames.

In comparison to the operation of conventional built-in 1-D SECDED circuit, the total number of cycles to complete 2-D HPC can be longer than that of 1-D SECDED. Table VI describes the expected cycles in a single frame operation for the example discussed above.

The proposed 2-D HPC technique offer several benefits such as faster reconfiguration, increased system throughput, and higher error tolerance as explained here.

- *Faster reconfiguration*: The time required to reprogram couple of frames is very short. For instance, partial reconfiguration using I/O blocks takes only few microseconds in commercial FPGAs [7], [14]. Hence, the proposed *in situ* EC circuit can provide faster reconfiguration using internal bus only.
- *Higher FPGA system throughput*: The read-back can be performed as a background operation. Hence, the 2-D HPC is performed without stalling the system function. Only the short reconfiguration of erroneous frames results in interruption of the system operation.
- *High error tolerance*: The physical dimension of the frame buffer is very small compared to the entire FPGA, and the time required to perform 2-D HPC can be few micro seconds with internal FPGA clock frequency of 200 MHz, for example. Hence, the probability of having soft errors in the buffer during the maintenance period is negligible.

In Section IV, we will explore the application of 2-D HPC technique in space application designs where an extremely high error protection scheme is an absolute necessity.

TABLE VII
AVERAGE NUMBER OF MODULE ERRORS THAT TMR DESIGNS CAN TOLERATE AT EACH CONFIDENCE LEVEL USING XILINX VIRTEX 5

Prob. of Operation	0.5Mb (s38584)	4.5Mb (MPEG4)	12.5Mb (*)
99%	5.8	17.1	27.4
95%	12.5	37.3	63.2
90%	17.4	54.1	90

(*) a design with 70% utilization of Xilinx Virtex 5

IV. CONTINUOUS FPGA OPERATIONS USING 2-D HPC IN SPACE APPLICATION DOMAIN

In a satellite system, FPGA performs programmed functions while it interacts with other system components. Due to the susceptibility of SRAM-based FPGA to soft errors, frequent repair processes have to be performed to ensure reliable operation. Thus, repairing FPGAs in such architecture involves interruptions of multiple system components. In that case, synchronization with other blocks, known as coherence problem, can result in further degradation of system performance. Therefore, the actual cost of correcting errors in FPGA and restarting the entire system is much higher than that of repairing FPGA itself. To prevent frequent system interruptions due to SEUs in FPGA, TMR is widely used in mission critical space application [2], [7]–[9]. TMR can ensure correct and continuous FPGA operation while tolerating both temporal transient errors and hard faults (configuration bit upsets), based on its majority voting mechanism. The error tolerance of TMR improves system reliability (R) which can be defined as follows [27]:

$$R(R_0, R_V, m) = (3R_V^2 R_0^{2/m} - 2R_V^3 R_0^{3/m})^m \qquad (1)$$

where R_0, R_V, and m are the reliability of nonredundant design, the reliability of voting system, and the resolution of a TMR design, respectively. With an assumption of perfect voting system $(R_V = 1)$, system reliability (R) increases with resolution (m) of TMR system. This assumption, $RV = 1$, can be valid since voting circuit in recent FPGAs can be implemented without dependency on SRAM configuration bits [7]. In addition, an increase in TMR granularity (m) can further increase the system reliability as given by (1). This means that the TMR system can tolerate multiple failures. In a TMR-based design, the smallest unit size in which a module can be implemented is known as a slice (one CLB consists of two slices in Xilinx architecture) [7], [23]. Since recent FPGAs can provide more than 100 K slices, a TMR with high-level resolution which can tolerate larger number of errors can be implemented. Table VII shows the average number of errors that a TMR implementation of several benchmark circuits can tolerate, while maintaining certain level of system reliability. As shown in Table VII, only a limited number of errors can be tolerated in TMR implementations. In addition, TMR design requires utilization of FPGA resources—at least $3\times$ higher than that of original design. This additional area can increase the chance of soft errors in the TMR based system. As can be seen in Fig. 9, the reliability of TMR degrades faster than the reference design, resulting in system failure. This implies that periodic preventive maintenance is required to guarantee continuous operation of system before the

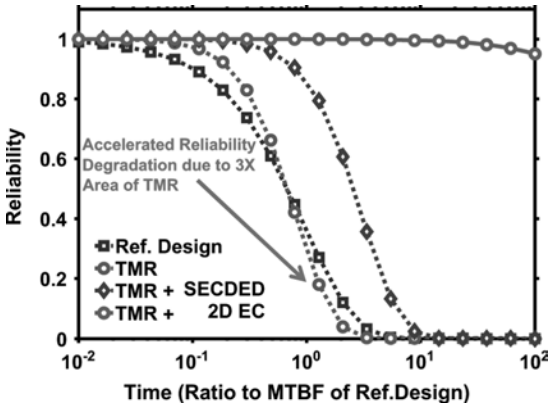

Fig. 9. TMR system reliability with built-in EC.

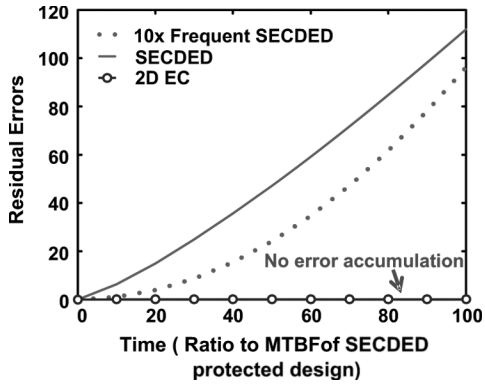

Fig. 10. Accumulation of residual errors in 1 K frames (1 Mb). one frame is 1 Kb. 2-D HPC was applied on 4×4 frames.

error accumulation exceeds the limits. Note that Xilinx Virtex5 and ISE9.0 [22] have been used to determine the design size of benchmark circuits in this analysis.

Simulation results in Fig. 10 show accumulation of residual errors in a system with SECDED increases with time. Even with $10\times$ shorter maintenance interval using conventional SECDED, the residual error accumulation occurs. On the other hand, the system with 2-D HPC repair scheme shows almost no residual errors. This is mainly due to high performance of multibit corrections in 2-D HPC. Therefore, TMR design with the proposed technique can perform continuous operation. Simulation result of reliability degradation with time is shown in Fig. 9. The reliability of protected TMR design using the proposed scheme results in long time survival of operation.

Availability of a system is the metric to measure continuous operation of a system. It is defined by the probability of continuous system operation when it is requested to operate and can be expressed as follows [6]:

$$\text{Availability}_{\text{FPGA}} = \frac{\text{MTTD}}{\text{MTTD} + \text{MTTR}} \tag{2}$$

where MTTD is the mean time to detect error with given maintenance interval, and MTTR is the mean time to recover. Note that availability becomes zero when there is an error which cannot be corrected. This is equivalent to infinite MTTR in (2). Therefore, availability of a system using conventional SECDED is zero due to residual errors exceeding the error tolerance of TMR based design. We applied 2-D HPC to TMR based designs. We note that accumulation of residual errors occurs with very low probability. In Table II, we have shown that 2-D HPC can correct more than hundreds of errors in 1-Mb array. Therefore, residual error accumulation which exceeds error tolerance is extremely unlikely.

Los Alamos National Laboratory reported that an application using Xilinx XQVR300 in a 1000-km circular orbit experienced SER of 7.2/day in the worst case. This can be translated into approximately 7 Failures in 1 billion hours of operation (FITs) per bit. The availability of the FPGA in the experiment (using TMR) is 99.998%. To calculate the availability in their work, they assumed actual repair time to be ten times the recovery time [2]. This is because the entire system has to stop to perform long detection and repair using slow I/O of FPGAs. However, our built-in 2-D HPC performs detection and correction using an internal bus. In addition, only a single frame can be reconfigured since the operation is independently performed on every configuration frame. With a pessimistic assumption of 3 μs reconfiguration time, the availability of the FPGA using the proposed scheme can be easily more than 99.9999999% with the 7 FITs per bit. (Reconfiguration time of a single frame can be 3 μs using I/Os in FPGAs [7]. Note, our scheme can provide faster reconfiguration using internal bus only.)

Continuous stable operation of a system is the most important goal in mission critical applications. We have shown that the proposed built-in 2-D HPC can provide highly reliable operation of FPGAs in space where extremely high SER is experienced.

V. Conclusion

In this work, a simple built-in 2-D HPC architecture that provides long survival of FPGAs in mission-critical space applications is presented. The simulation results show that 2-D single bit corrections can provide very high multibit ECs. This work also includes an optimized hardware implementation of our proposed scheme using a new 2-D SRAM array. Furthermore, due to the extremely large multibit error-handling capability of the proposed design, self-healing of SRAM-based FPGA can be achieved even in harsh radiation environments.

References

[1] A. Lesea *et al.*, "The Rosetta experiment: Atmosperic soft error rate testing in differing technology FPGAs," *IEEE Trans. Device Mater. Rel.*, vol. 5, no. 3, pp. 317–328, Sep. 2005.

[2] E. Fuller *et al.*, "Radiation testing update, SEU mitigation, and availability analysis of the Virtex FPGA for space reconfigurable computing," in *Proc. MAPLD*, 2000.

[3] J. F. Ziegler, "Terrestrial cosmic ray intensities," *IBM J. Res. Develop.*, vol. 42, no. 1, pp. 117–139, 1998.

[4] E. Normand, "Single event effects in avionics," *IEEE Trans. Nucl. Sci.*, vol. 43, no. 2, pp. 461–474, Apr. 1996.

[5] R. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *IEEE Trans. Device Mater. Rel.*., vol. 5, no. 3, pp. 305–316, Sep. 2005.

[6] H. Asadi *et al.*, "Analytical techniques for soft error rate modeling and mitigation of FPGA-based designs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 16, no. 12, pp. 1320–1331, Dec. 2007.

[7] C. Carmichael *et al.*, "SEU mitigation techniques for Virtex FPGAs in space applications," in *Proc. MAPLD*, 1999.

[8] K. Chapman *et al.*, "SEU strategies for Virtex-5 devices," Xilinx Application Note 864, 2009.

[9] M. Garvie *et al.*, "Scrubbing away transients and jiggling around the permanent: Long survival of FPGA systems through evolutionary self-repair," *Proc. 10th IOLTS*, pp. 155–155, 2004.

[10] I. Chang *et al.*, "A 32 kb 10T subthreshold SRAM array with bit-interleaving and differential read scheme in 90 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 44, no. 2, pp. 650–658, Feb. 2009.
[11] B. Gill *et al.*, "Radiation induced single-word multiple-bit upsets correction in SRAM," in *Proc. 11th IEEE Int. LOTS*, Jul. 6-8, 2005, pp. 266–271.
[12] L. Chang *et al.*, "An 8T-SRAM for variability tolerance and low-voltage operation in high-performance caches," *IEEE J. Solid-State Circuits*, vol. 43, no. 4, pp. 956–963, Apr. 2008.
[13] V. George *et al.*, *Low-Energy FPGAs: Architecture and Design*. Boston, MA: Kluwer, 2001.
[14] *"Virtex Series Configuration Architecture User Guide, Xilinx Application Note 151,"* Xilinx Corp., San Jose, CA, 2004.
[15] S. Lin *et al.*, *Error Control Coding: Fundamentals and Applications*. New York: IEEE Press, 1994.
[16] M. B. Pursley, *Introduction to Digital Communications*. Upper Saddle River, NJ: Prentice-Hall, 2005.
[17] C. Liu *et al.*, "An LDPC decoder chip based on self-routing network for IEEE 802.16e applications," *IEEE J. Solid-State Circuits*, vol. 43, no. 3, pp. 684–694, Mar. 2008.
[18] C. Benkeser *et al.*, "Design and optimization of an HSDPA turbo decoder ASIC," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 98–106, Jan. 2009.
[19] H. Chang *et al.*, "A Reed–Solomon product-code (RS-PC) decoder chip for DVD applications," *IEEE J. Solid-State Circuits*, vol. 36, no. 2, pp. 229–238, Feb. 2001.
[20] L. Song *et al.*, "10- and 40-Gb/s forward error correction devices for optical communications," *IEEE J. Solid-State Circuits*, vol. 37, no. 11, pp. 1565–1573, Nov. 2002.
[21] P. J. Black *et al.*, "A 1-Gb/s, four-state, sliding block Viterbi decoder," *IEEE J. Solid-State Circuits*, vol. 32, no. 6, pp. 797–805, Jun. 1997.
[22] *"ISE Design Suite,"* Xilinx Corp., San Jose, CA [Online]. Available: http://www.xilinx.com
[23] *"Virtex-5 FPGA User Guide,"* Xilinx Corp., San Jose, CA [Online]. Available: http://www.xilinx.com, Guide
[24] K. Nii *et al.*, "Synchronous ultra-high-density 2 RW dual-port 8T-SRAM with circumvention of simultaneous common-row-access," *IEEE J. Solid-State Circuits*, vol. 44, no. 3, pp. 977–986, Mar. 2009.
[25] T. K. Moon, *Error Correction Coding: Mathematical Methods and Algorithms*. New York: Wiley-Interscience, 2005.
[26] H. Quinn, "Static proton and heavy ion testing of the Xilinx Virtex-5 device," in *Proc. IEEE Radiation Effects Data Workshop*, Jul. 23-27, 2007, pp. 177–184.
[27] R. E. Lyons *et al.*, "The use of triple-modular redundancy to improve computer reliability," *IBM J. Res. Develop.*, vol. 6, no. 2, pp. 200–209, Apr. 1962.

Sang Phill Park (S'07) received the B.S. degree in computer engineering from the University of Arizona, Tucson, in 2004. He is currently working toward the Ph.D. degree in electrical engineering at Purdue University, West Lafayette, IN.

In 2008, he was with Exploratory VLSI Design Group, IBM Austin Research Laboratory, as a Research Intern. His research interests include variation-resilient ultralow-power circuit design and failure/fault-tolerant design methodologies.

Dongsoo Lee received the B.S. and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2002 and 2004, respectively. He is currently working toward the Ph.D. degree in electrical engineering at Purdue University, West Lafayette, IN.

In 2004, he joined Samsung Electronics Co., Ltd., Suwon, Korea, where he worked on designing circuits for DTV one-chip-solutions. His research interests include digital circuits for signal processing, test data compression, and low-power design.

Kaushik Roy (SM'95–F'01) received the B.Tech. degree in electronics and electrical communications engineering from the Indian Institute of Technology, Kharagpur, India, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, Urbana, in 1990.

He was with the Semiconductor Process and Design Center, Texas Instruments, Dallas, where he worked on FPGA architecture development and low-power circuit design. He joined the Electrical and Computer Engineering Faculty with Purdue University, West Lafayette, IN, in 1993, where he is currently a Professor and holds the Roscoe H. George Chair of Electrical and Computer Engineering. He has authored or coauthored more than 500 papers in refereed journals and conferences, holds 15 patents, graduated 50 Ph.D. students, and is coauthor of two books, *Low-Power CMOS VLSI Circuit Design* (Wiley, 2000) and *Low-Voltage Low-Power VLSI Subsystems* (McGraw-Hill, 2004). He was guest editor for *IEE Proceedings—Computers and Digital Techniques* (July 2002). He is a Purdue University Faculty Scholar. He was a Research Visionary Board Member of Motorola Labs (2002) and held the M.K. Gandhi Distinguished Visiting faculty at Indian Institute of Technology (Bombay). His research interests include Spintronics, VLSI design/CAD for nano-scale silicon and nonsilicon technologies, low-power electronics for portable computing and wireless communications, VLSI testing and verification, and reconfigurable computing.

Dr. Roy was the recipient of the National Science Foundation Career Development Award in 1995, IBM faculty partnership award, ATT/Lucent Foundation award, 2005 SRC Technical Excellence Award, SRC Inventors Award, Purdue College of Engineering Research Excellence Award, Humboldt Research Award in 2010, and Best Paper Awards at the 1997 International Test Conference, the IEEE 2000 International Symposium on Quality of IC Design, the 2003 IEEE Latin American Test Workshop, 2003 IEEE Nano, the 2004 IEEE International Conference on Computer Design, the 2006 IEEE/ACM International Symposium on Low Power Electronics & Design, and the 2005 IEEE Circuits and System Society Outstanding Young Author Award (Chris Kim), the 2006 IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS Best Paper Award. He has been in the editorial board of *IEEE Design and Test*, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, and the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS. He was guest editor for a special issue on Low-Power VLSI in the *IEEE Design and Test* (1994) and the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS. (June 2000).