

FAULT TOLERANT SRAM MEMORY USING 2-D SYMBOLIC HAMMING MATRIX CODE

Divya.C, *Second year M.E, Communication System,*
Rohini College of Engineering and Technology, Paulkulam.

Abstract

Error Correction Codes (ECCs) are commonly used to protect memories from soft errors. As technology scales, Multiple Cell Upsets (MCUs) become more common and affect a larger number of cells. To prevent the occurrence of MCUs several error correction codes (ECCs) are used, but the main problem is that they require complex encoder and decoder architecture and higher delay overheads. The 2-D Symbolic Hamming Matrix Code (2-D SHMC) minimizes the area and delay overheads compared to the existing codes such as hamming, RS codes and also improves the memory reliability by enhancing the error correction capability. In this paper, 2-D Symbolic Hamming Matrix Code (2-D SHMC) based on divide-symbol is proposed to enhance memory reliability with lower delay overhead. The proposed 2-D SHMC utilizes decimal algorithm to obtain the maximum error detection capability. Moreover, the encoder-reuse technique (ERT) is proposed to minimize the area overhead of extra circuits without disturbing the whole encoding and decoding processes. ERT uses DMC encoder itself to be part of the decoder.

1. Introduction

An exponential growth of soft error rate (SER) in SRAM is one of the technological challenges faced in today's scenario. The major factors for this growth of SER include forceful transistor downscaling, ever-increasing transistor count per chip, and reduced supply voltage [1,2]. In addition to that, in mission-critical space applications the usage of SRAM based FPGAs is limited due to the vulnerability to single event upsets (SEUs) [1,2]. More over the functionality of the FPGA is maintained by the contents of the SRAM based configuration memories (CMs). The contents of the configuration memories are altered by SEU and produces malfunction of the user program on FPGA. To mitigate the soft error issue, the recent SRAM based FPGAs employ error-correcting codes (ECCs) to shield the configuration bit and improve the reliability [3,4].

To mitigate vulnerability to soft errors in such configuration memory, redundant intend methods such as duplication with comparison (DWC) and triple-module redundancy (TMR) can be used for mission-critical space applications [5]. Built-in TMR mechanism can accumulate soft errors and produce faulty results when two copies of any module in FPGA fail. An alternate method of cyclic redundancy check (CRC) based configuration read back is used to detect the soft error in recent 7series Xilinx devices (Kintex-7 and Zynq-7000) [7-10]. The above method relies on external nonvolatile radiation hardened memory which stores the golden copy of configuration data. Accessing configuration data from external memory incur an extra delay and degrades the performance.

MBU detection is possible by reading back the configuration data from FPGA via internal configuration access port (ICAP) interface, and

comparing it with a golden configuration data stored in an external non-volatile memory. For this solution, Xilinx-EDA tools create readback data file (RDF) and mask data file (MDF) for each bit stream file. The MDF is used to mask the readback bits that may change during operation. The word-by-word masking and comparison is a slow procedure and requires accessing the complete configuration golden memory for comparison. In case of mismatch, an error detection ensign must be raised to timely alleviation actions. Both storage overhead and access time for these two files has to be considered at system level, which may not be reasonable for the complex FPGAs. The read back process involves unnecessary delay in accessing external nonvolatile memory to pick up the original configuration data through joint text access group (JTAG) or SELECTMAP interface. Compared to golden reference and comparison technique, performing a CRC check on the readback data, is a faster technique which needs reference CRC codes instead of reference configuration data [13].

Therefore, improvement techniques are still mandatory to ensure continuous and correct operation of the system. The dedicated built-in ECC and CRC hardware blocks are available to mitigate soft errors in recent FPGAs [5,6]. The built-in ECC performs single error correction double error detection (SEDED), and it cannot mitigate multiple bit errors in a single word of configuration frame. More over CRC is also performed concurrently during read-back process to detect possible multiple bit errors. For the detection of MBU, the reconfiguration process involves unnecessary delay in accessing external nonvolatile memory to pick up the original configuration data through restricted I/Os in FPGAs [6]. In this work the proposed method is 5 times faster than existing CRC based inbuilt error mitigation solution.

This paper presents a new built-in reconfigurable matrix code (RMC) and symbolic hamming matrix code (SHMC) schemes to afford

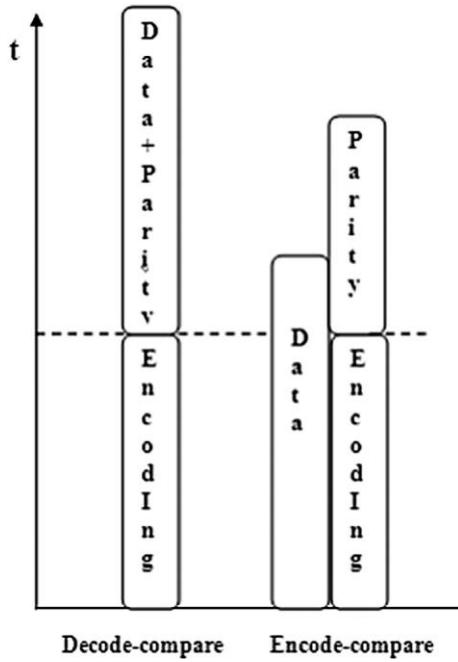


Fig. 1. Delay overhead diagram.

the trustworthy operation of SRAM-based FPGAs in mission-critical space applications. Finally the comparison of existing built-in ECCs and proposed 2-D ECCs have been evaluated in Xilinx-7 series FPGAs in terms of error correction time and hardware complexity.

Contributions: The following are the key objectives and contributions of this work:

- ❖ Encode-and-Compare scheme reduces the error correction process delay
- ❖ Fine grain ECC optimizes the overall delay and the reliability
- ❖ Novel built-in ECC solution avoids the accumulation of soft error in CM
- ❖ Integrated novel built-in ECC for MBU tolerant FPGAs.

2. Encode-and-compare scheme

The most complicated task in built-in error correction is decoding. Decoding process includes syndrome computation for error detection and error correction. Normally decoding consumes more time compared to encoding. Instead of using decode-and-compare mechanism, encode-and-compare technique can be used to reduce the overall delay of proposed built-in error correction codes. Encoding of original configuration data from configuration frame and comparison between the contents i.e. parity bits of redundant memory and current encoded parity bits can be done in encode-and-compare mechanism. This will lead to quick error detection and it reduces the complexity as well. In the traditional decode-compare scheme shown in Fig. 1 the decoder, decodes the parity and data from the codeword and compares with the stored parity.

In the encode-compare scheme shown in Fig. 1 the data is encoded at the encoder and the resultant parity is compared with the stored parity. In this work encode-and-compare scheme is incorporated with the novel built-in 2-D ECCs for reducing the overall delay of error correction.

3. Proposed ECCs: 2-D RMC and 2-D SHMC

In the proposed method, the configuration word is arranged in the matrix format with divide by symbol approach called fine grain mechanism. The fine grain mechanism improves the reliability and speed of the operation. N-bit word is divided into K symbols and m bits/symbol. The symbol K is computed by $K = R \times C$ where R and C represents the number of rows and columns respectively and makes the matrix format. The complete N-bit word is organized as $N = R \times C \times m$. In 2-D RMC, N/2 horizontal and vertical redundant bits are computed to correct the soft errors for an N-bit word. In 2-D SHMC, $\log(N + 1)$ redundant bits are computed for each symbol of configuration word.

Algorithm 1 shows the procedure of MBU detection and correction using 2-D RMC method to be applied on a configuration word B, where H and V are the horizontal check bits and the vertical parity bits that are generated using the saved configuration data bits in the memory. These bits are compared with the saved redundant memory contents to generate the syndrome bits. Similarly algorithm 2 shows the procedure of MBU

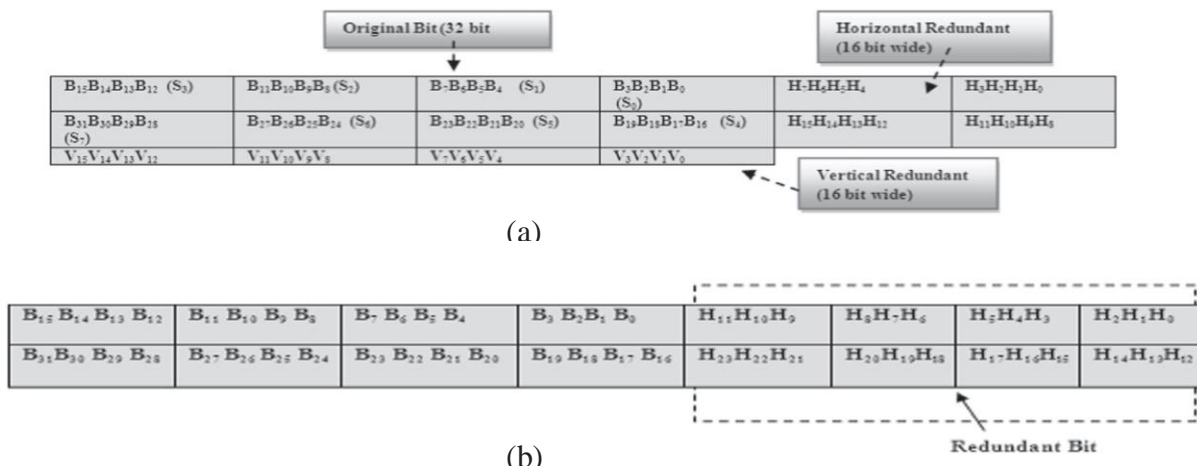


Fig. 2. (a) 2-D RMC redundant bits calculation (b) 2-D SHMC redundant bits calculation.

detection and correction in 2-D SHMC method. Upto 8 bits are corrected by the proposed ECCs for a single configuration word.

Algorithm 1. 2-D RMC MBU correction

1. Read the configuration data (B') bits from configuration frame
2. Arrange the B' into matrix format using divide by symbol approach (fine-grain)
3. Generate horizontal check bits ($H_{15}-H_0$)
4. Generate vertical parity bits ($V_{15}-V_0$)
5. Generate syndrome bits S_p' from stored parity data and encode-and-compare method
6. Correct the erroneous bits using the eqn $C = S_p' \oplus B'$
7. Output the corrected data

Algorithm 2. 2-D SHMC MBU correction

1. Read the configuration data (B') bits from configuration frame
2. Arrange the B' into matrix format using divide by symbol approach (fine-grain)
3. Compute the SECCED (7, 4) for each symbol
4. Generate syndrome bits S_p' from stored parity data and encode-and-compare method
5. Correct the erroneous bits using the Eq. $C = S_p' \oplus B'$
6. Output the corrected data

For example in [algorithm 1](#), the horizontal and vertical redundant bits calculation is as follows. From 32 bit configuration word, 8 different symbols are divided and named as Symbol-0, 1, 2, 3, 4, 5, 6, and 7 as illustrated in [Fig. 2](#). The horizontal redundant bits H_0 to H_{15} are generated for 32 bit configuration word using Eqs. (1)–(4). The symbol-0 ($B_0B_1B_2B_3$) and symbol-2 ($B_8B_9B_{10}B_{11}$) are applied to X-OR operation to generate the redundant bits ($H_3H_2H_1H_0$). Similarly symbol pair {1, 3}, {4, 6}, {5, 7} are applied to X-OR operation to generate their redundant bits ($H_7H_6H_5H_4$), ($H_{11}H_{10}H_9H_8$), ($H_{15}H_{14}H_{13}H_{12}$) respectively. The vertical redundant bit can be computed using Eq. (5).

$$B_{11}B_{10}B_9B_8 \oplus B_3B_2B_1B_0 \quad \delta_{1p} \quad B_{31}B_{30}B_{29}B_{28} \oplus B_{23}B_{22}B_{21}B_{20} \quad \delta_{4p}$$

$$B_{15}B_{14}B_{13}B_{12} \oplus B_7B_6B_5B_4 \quad \delta_{2p} \quad V_n \quad \delta_{5p}$$

$$B_{27}B_{26}B_{25}B_{24} \oplus B_{19}B_{18}B_{17}B_{16} \quad \delta_{3p} \quad B_1 \oplus B_2 \oplus B_3 \quad \delta_{6p}$$

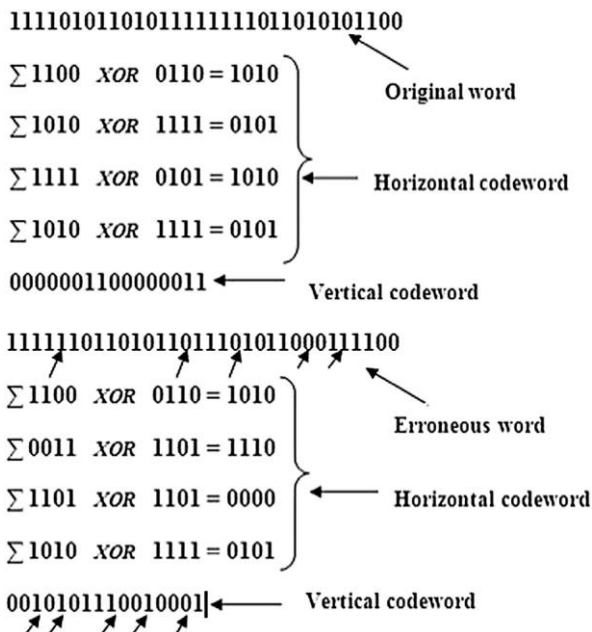


Fig. 3. Illustration of 2-D RMC for MBU detection.

MBU TOLERANT FPGA SYSTEM

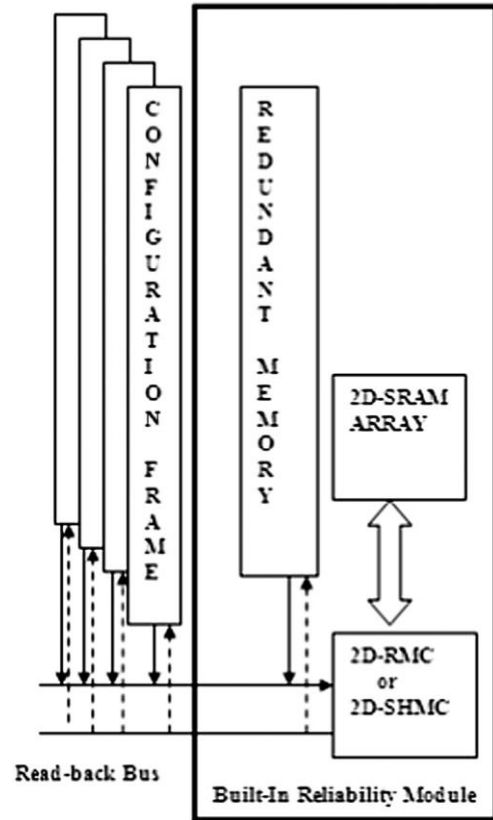


Fig. 4. Proposed MBU-tolerant FPGA architecture.

$$B_{31}B_{30}B_{29}B_{28} \oplus B_{23}B_{22}B_{21}B_{20} \quad \delta_{4p}$$

$$V_n \quad \delta_{5p}$$

Similarly in [algorithm 2](#), for 32-bit configuration word, 24 numbers of redundant bits are computed to correct the MBU errors. [Algorithm 2](#) uses SECCED (7, 4) computation for each symbol to generate the parity redundant bits. For example in [Fig. 2b](#), the redundant bits for symbol-0 ($B_0B_1B_2B_3$) can be computed using Eqs. (6)–(8). Similarly the redundant bits for all symbols are applied to X-OR operation to generate their redundant bits.

$$B_0 \oplus B_1 \oplus B_3 \quad \delta_{6p}$$

$$B_0 \oplus B_2 \oplus B_3 \quad \delta_{7p}$$

$$B_1 \oplus B_2 \oplus B_3 \quad \delta_{8p}$$

Finally the difference between stored redundant bits and the recalculated redundant bits gives the syndrome value (S_p'). Horizontal code word detects the error symbols and vertical code word identifies the location of error. The erroneous word can be detected from the

Table 1
Number of correctable soft errors by using built-in ECCs.

	2-D SHMC (n,k) = (56,32)	2-D RMC (n,k) = (64,32)	1-D SECCED (n,k) = (1035,1024)
*	8	8	1
**	256	256	1

(*) no. of correctable error for one configuration word (**) no. of correctable error for a one configuration frame.

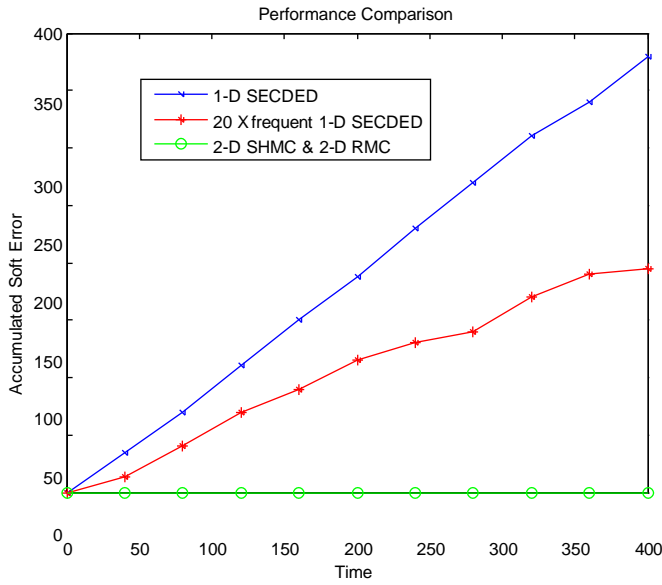


Fig. 5. Accumulated soft error vs time.

illustration of 2-D RMC shown in Fig. 3. The original bits and its horizontal and vertical code words are generated in the encoder. In later stage, horizontal and vertical code words are calculated for the erroneous word in Fig. 3. Finally horizontal syndrome and vertical syndrome are calculated to detect and locate errors respectively. In Fig. 3, the difference between the respective horizontal codeword, detects the error from the horizontal symbols and difference between the respective vertical codeword, locates the position of the error accurately.

4. Proposed MBU tolerant FPGA architecture

The proposed efficient 2-D RMC and 2-D SHMC are integrated on the conventional FPGA architecture and show the reliability improvement in MBU tolerant FPGAs. The proposed architecture requires only little logical modification to the conventional FPGA structure as shown in Fig. 4. Configuration frame is the smallest unit size of the configuration bit stream in the FPGA architecture. Normally a unit frame consists of 41 words, and each word consists of 32 bits. Configuration data is loaded from configuration frame through internal configuration access port (ICAP) read back bus [7]. The 2D-SRAM array stores the temporal copy of configuration frame data and parity data to execute 2-D ECC. Pre-computed parity data is retrieved from the dedicated on-chip redundant memory. This on-chip memory access can speed up the execution of the proposed ECC. The 2-D SRAM is used in the proposed MBU tolerant FPGA system for the row and column access and it increases the execution speed of the ECC. The corrected erroneous bits write back to the 2-D SRAM array through the write port.

The complete process of proposed MBU tolerant 2-D ECC system is as follows.

Step 1) 32 bits of configuration word from the frame data are read through the ICAP read back bus, while stored parity bits are read from redundant memory.

Table 2
Overall delay by using built-in ECCs.

	2-D SHMC (n,k) = (56,32)	2-D RMC (n,k) = (64,32)	1-D SECCED (n,k) = (1035,1024)
*	0.792 ns	1.032 ns	75.136 ns
**	25.344 ns	33.024 ns	75.136 ns

Table 3
Number of XOR gates in the built-in ECCs.

	2-D SHMC (n,k) = (56,32)	2-D RMC (n,k) = (64,32)	1-D SECCED (n,k) = (1035,1024)
*	24	32	11
**	24	96	5687
***	6912	9216	5687

(*) syndrome bits (**) no. of XOR gates (***) no. of XOR gates required for one configuration frame.

- Step 2) Following 2-D ECC is performed on 32-bit word, accumulate the data into 2-D SRAM array.
- Step 3) Replicate steps 1) and 2) until the 2-D SRAM array is full.
- Step 4) Execute column read and 2-D ECC in column direction.
- Step 5) Trace the row address recognized by 2-D ECC and the column address of contemporary operation.
- Step 6) Do correction of erroneous bit through row read and write port based on row and column address.
- Step 7) The corrected bits are written back to the erroneous frames in 2-D SRAM array through programming bus.
- Step 8) Continue the process from step 1) to step 8) for subsequent set.

5. Evaluation

A traditional error correction technique for SRAM-based FPGAs relies on external golden copy and external devices to simultaneously correct the configuration bit errors. The read back process involves unnecessary delay in accessing external nonvolatile memory to pick up the original configuration data through joint text access group (JTAG) or SELECTMAP interface. The main objective of this work is to realize self-error correction without an external golden memory copy. The work has been evaluated in terms of behavioral simulation and physical implementation. Behavioral simulations have been carried out by using Xilinx ISE13.2 EDA tool. The proposed built-in error correction codes have been implemented in Verilog HDL and mapped on Xilinx's 7-series xc7k70t-2-fbg676 FPGA. The results based on LANSCEneutron test (2012) shows that 25% of 5-bit upsets or 16% of more than 6-bit upsets accumulation are occurring in the configuration frame [15]. The largest MBU size observed in the experiment has been 24 and the possible MBU patterns are studied [16]. Based on this observation to validate the proposed ECC, the multibit errors are injected in a random manner. The original configuration data and the fault configuration data can be specified in the text fixture, and fault injection can be implemented in a test-bench. Test-bench takes care of applying faulty stimulus to the design. This test-bench needs to be instantiated and the millions of random errors can be injected to an array of configuration memory, and simultaneously error correction has been done by the proposed 2-D ECCs.

Table 4
Error mitigation latency analysis for built-in ECCs.

	2-D SHMC (n,k) = (56,32)	2-DRMC (n,k) = (64,32)	Xilinx-CRC (n,k) = (1035,1024)
*	2.95 μ s	2.95 μ s	2.95 μ s
**	750 μ s	750 μ s	750 μ s
!!!	465 μ s	606 μ s	18,750 μ s
***	4.165 ms	4.306 ms	22.45 ms

Table 5
Dynamic power analysis for built-in ECCs.

	2-D SHMC (n,k) = (56,32)	2-D RMC (n,k) = (64,32)	Xilinx-CRC (n,k) = (1035,1024)
*	0.045w	0.045w	0.045w
**	0.092w	0.096w	–
***	0.138w	0.141w	–

(*) repair time for one configuration word (**) repair time for a one configuration frame of (xc7k70t-2-fbg676).

Traditional 1-D SECEDED has been implemented to assess the benefits of the proposed built-in 2-D ECCs. The reliability of proposed and existing ECCs is portrayed in terms of correctable errors in Table 1. Both 2-D SHMC and 2-D RMC corrects 8 bits for a single configuration word and corrects up to 256 for a single configuration frame. Frequent error correction is required for the inefficient, unprotected techniques and requires shorter correction interval. Fig. 5 shows the comparison of accumulated soft error correction capability. Due to the multibit correction performance of 2-D SHMC and 2-D RMC, almost no accumulation of soft errors present in the configuration memories. The accumulation of soft errors in a system increases with respect to correction time for 1-D SECEDED technique. The accumulation of soft errors occurs even when 1-D SECEDED is applied on FPGA with 20 times shorter correction interval.

The overhead results in terms of delay and hardware cost are illustrated in Tables 2 and 3 for the proposed and existing ECCs. The implementation of encode–compare scheme for 2-D ECCs reduces the delay with a trade off in hardware cost. The overall delay is reduced to 67% and 56% for 2-D SHMC and 2-D RMC respectively. The number of XOR gates is increased to 21% and 38% for 2-D SHMC and 2-D RMC respectively.

Error mitigation latency of whole configuration memory for the proposed method can be calculated using repair time for a single configuration frame as measured in Table 2. The readback latency and error classification latency for kintex-7 FPGA are 2.95 μ s and 750 μ s respectively in 28 nm transistor technology [14]. The error mitigation latency for 18,361 configuration frames of kintex-7 FPGAs depicts that the proposed method is 5 times faster than Xilinx CRC and is given in Table 4. Finally the area overhead for storing extra redundant bits for the proposed 2-D ECCs given in Table 6 has been compared with the existing Xilinx CRC solution. It is inferred that 2-D RMC and 2-D SHMC need 24 MB and 18 MB memory units respectively and Xilinx CRC solution need only 8-MB memory units. Finally the area overhead of the 2-D RMC is 3 \times and 2-D SHMC is 2.25 \times compared to Xilinx CRC. The power consumption for the proposed 2-D ECCs are given in Table 5, unfortunately the power consumption for Xilinx CRC is not available in literature.

5.1. Correction time comparison with real time experiment

In practice, configuration memory scrubbing technique is used to perform the error tolerant execution of real-time applications in FPGA. Aeroflex a COBHAM company from United States of America (USA) introduces Leon-3 FPGA for commercial aerospace applications. The scrubbed time for the Leon-3 FPGA is 4.98 ms [11]. If the scrubbing method detects an error, the operation will be paused and the scrubbing controller will repair the error. After the error repair, the FPGA restores

the configuration data and resumes operation. Notice that it is possible for an error, which is detected in the current scrubbing cycle, had also been occurred in the previous scrubbing interval. In this manner, the errors are accumulating in the FPGA based real time embedded systems. The error mitigation latency of the proposed and existing ECCs is portrayed in Table 4. The error mitigation latency for 2-D SHMC and 2-D RMC is 4.165 ms and 4.306 ms respectively. According to the Leon-3 real time experiment, the error mitigation latency of the proposed work is sufficient for real time applications.

6. Conclusion and perspective

In this paper, an efficient implementation of MBU tolerant FPGA architecture has been presented and evaluated. The proposed implementation based on novel built-in error correction codes provides minimum correction delay with a tradeoff in the circuit area of an FPGA. In addition, its fine grain structure and encode-and-compare technique provides the opportunity for further reduction of repair time. The multibit correction performance of 2-D SHMC and 2-D RMC enables no accumulation of soft errors present in the configuration memories for a long time. The only drawback to the proposed MBU tolerant FPGA architecture is that it requires more redundant bits and slightly increased hardware cost for FPGA reliability. This work opens a door for 2-D ECC to be universally used in FPGAs for space applications.

References

- [1] A. Dixit, A. Wood, The impact of new technology on soft error rates, IRPS2011.
- [2] H. Kaul, Near-threshold voltage (NTV) design: opportunities and challenges, DAC2012.
- [3] R. Baumann, Radiation-induced soft errors in advanced semiconductor technologies, IEEE Trans Device Mater Reliab 5 (3) (2005) 305–316.
- [4] H. Asadi, Analytical techniques for soft error rate modeling and mitigation of FPGA-based designs, IEEE Trans (VLSI) Syst 16 (12) (2007) 1320–1331.
- [5] K. Chapman, SEU strategies for Virtex-5 devices, Xilinx Application Note8642009.
- [6] C. Carmichael, SEU mitigation techniques for Virtex FPGAs in space applications, Proc MAPLD, 1999.
- [7] U. Legat, SEU recovery mechanism for SRAM-based FPGAs, IEEE Trans Nucl Sci 59 (5) (2012) 2562–2571.
- [8] E. Mohsen, Balancing Performance, Power, and Cost with Kintex-7 FPGAs, White PaperWP432(v1.1)2013.
- [9] Device Reliability Report, 2014. (UG116, v10.1, Second Quarter).
- [10] Device Reliability Report, 2014. (UG116, v10.2.1, Second half).
- [11] Aitzan Sari, et al., Combining check pointing and scrubbing in FPGA based real-time systems, IEEE 31st VLSI Test Symposium (VTS), 2013.
- [13] I. Herrera-Alzu, M. López-Vallejo, Design techniques for Xilinx Virtex FPGA configuration memory scrubbers, IEEE Trans Nucl Sci Vol. 60 (No. 1) (February 2013).
- [14] 7-Series FPGAs Configuration User Guide UG470 (v1.9) November 14, 2014.
- [15] Michael Wirthlin, Using FPGAs in a radiation environment, ATLAS CMS Electronics Workshop for LHC Upgrades (ACES2014)18–20 March, 2014.
- [16] Parthasarathy M.B. Rao, Mojtaba Ebrahimi, Razi Seyyedi, Mehdi B. Tahoori, Protecting SRAM-based FPGAs against multiple bit upsets using erasure codes, ACM—Proceedings on DAC, 2014.

Table 6
Area overhead for storing redundant bits for built-in ECCs.

	2-D SHMC (n,k) = (56,32)	2-D RMC (n,k) = (64,32)	Xilinx-CRC (n,k) = (1035,1024)
*	18 MB	24 MB	8 MB

(*) on-chip redundant memory overhead for 18,361 frames.