Development of Configurable Static Random Access Memory (SRAM) for Space Applications

2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT) | 978-1-6654-1480-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICECCT52121.2021.9616927

Padmapriya K.
U R Rao Satellite Centre
Indian Space Research Organization
Bangalore, India

Shivani Prasad Bondapalli
Department of Computer Science
University Visvesvaraya College of Engineering
Bangalore, India

B. K. S. V. L. Varaprasad
U R Rao Satellite Centre
Indian Space Research Organization
Bangalore, India
bksvlvp@ursc.gov.in

Rahul Anilkumar
U R Rao Satellite Centre
Indian Space Research Organization
Bangalore, India
rahulak@ursc.gov.in

Debjyoti Mallik
Semi-Conductor Laboratory
Department of Space
Chandigarh, India
mallik@scl.gov.in

Sajjade Faisal Mustafa
U R Rao Satellite Centre
Indian Space Research Organization
Bangalore, India
sajjade@ursc.gov.in

***Abstract*— Developing configurable electronic devices aimed for use in multiple spacecrafts is economical considering the huge costs incurred when developing reliable, radiation hardened devices for space applications. In this context, developing an SRAM device with configurable size and Single-Event Upsets (SEU) mitigation technique(s) is a good choice for critical and semi-critical on-board electronic systems of a spacecraft. This article presents the design of an SRAM device with dynamically configurable size and SEU mitigation techniques, and fabrication using 180 nm Complementary Metal Oxide Semiconductor (CMOS) non-radiation hardened process technology. Implementation of the SRAM design with memory macros/cuts, and the techniques followed for logic synthesis, place and route design activities for achieving the timing closure and best Quality of Results (QoR) are illustrated. Simulation studies, reliability analysis, and memory refresh time give confidence for using this device in spacecrafts with short and long mission life requirements.**

Keywords*— *Static RAM; TMR; EDAC; SEU; Multiple Bit Upsets; Bit Error Rate.

I. INTRODUCTION

Development of an Application Specific Integrated Circuit (ASIC) or any device for use in a spacecraft is very expensive due to the requirement of devices per spacecraft being limited in number and the need for the device to be fault tolerant, and have high-reliability to operate in the radiation environment of space. Hence, it is desirable to combine several functions and develop a configurable multifunctional device for use in multiple spacecraft projects to make it cost effective. An SRAM device, being widely used in control/guidance, data processing and data storage electronic systems of a spacecraft, is a good choice for development as a configurable device.

SRAM devices which are capable of withstanding space radiation have been developed using various radiation hardening techniques at process level and cell design level by using SEU protection circuits, and other techniques as detailed in [1-3]. When an SRAM device is manufactured using non-radiation hardened (non-rad-hard) CMOS advanced technology nodes, it will suffer from space radiation that leads to SEUs which can cause failures in spacecraft electronic systems. The SEU rate of an SRAM device highly depends on process technology, feature size, orbit geometry of the spacecraft and space weather. Hence, protection of the SRAM device using suitable SEU mitigation techniques like Triple Modular Redundancy (TMR), Error Detection and Correction (EDAC), scrubbing and so on is necessary for reliable operation of a spacecraft.

For memories, the most common approach to maintain reliability and stability is to use Error Correcting Codes (ECCs) [4]-[6]. Most frequently adopted EDAC codes including the parity code, the Hamming code and the matrix code have a limitation in the number of bits detected and corrected. The presence of Multiple Bit Upsets (MBUs) is becoming more frequent with advanced technology nodes and these single-bit error-correction codes cannot guarantee adequate protection against MBUs [7]-[9]. Hence, a suitable design technique is required to manage the occurrence of more than two-bit upset errors.

Triple Modular Redundancy (TMR) is also a commonly employed strategy to mitigate the SEU effects on SRAMs [10]. In TMR, the circuit is replicated three times and Majority Voter Logics (MVL) are inserted on the outputs so that only the majority logic value of the three redundant output values is generated as output from the device. The effectiveness and limitations of TMR have been well explored in literature. The effectiveness of TMR is limited due to the presence of Common Mode Failures (CMF) in the TMR implementation. For example, two or more bit upsets at the same logical position will cause circuit failures, despite adding TMR to the entire storage area of SRAM [11]-[14].

In this paper, we propose the use of two fault mitigation techniques - TMR followed by EDAC, to overcome multiple bit upset errors of an SRAM.

This paper is organized as follows: Specifications and functional design of the memory device is

presented in Section II. In Section III, ASIC design activities like Logic synthesis, verification, and physical design are illustrated using Electronic Design Automation (EDA) tools. Device testing is illustrated in section IV. Refresh rate estimation using reliability analysis is presented in section V. Finally, conclusions are made in section VI.

II. Specification and Functional Design

The objective of this development is to realize a monolithic high density (15 Mb) memory device using 180 nm CMOS technology for space applications. The device should be configurable such that it can be used as a rad-hard memory in critical systems and a rad-tolerant memory in non-critical systems. Also, in both modes of operation, error flags are generated when the SEU happens to indicate the occurrence of an error. In addition to error flag generation, in-built error detection and correction logic are helpful for non-critical mode of operation.

A. Features/Configurations

The features/configurations of the memory device are

- Mode-1: 384K x 40-bit organization (without TMR and EDAC) for usage in ground applications.
- Mode-2: 384K x 32-bit organization (without TMR and with 8bit EDAC) for usage in payload applications.
- Mode-3: 128K x 40-bit organization (with TMR and without EDAC) for usage in critical sub-systems in spacecrafts using external EDAC function.
- Mode-4: 128K x 32-bit organization (with TMR and EDAC) for usage in critical sub-systems in spacecrafts with long mission life.
- Dynamically configurable in Mode-1, Mode-2, Mode-3 or Mode-4 depending on space orbit and space weather.
- Error flag generation for identification of SEUs on the fly that helps to prevent the device from accumulating errors through scrubbing as soon as an error is identified.
- Testing: Bank – A, B, C memories individually, ensuring that the device is free from manufacturing defects.

B. Functional Design

Realization of a 15 Mb memory is fulfilled using 8Kx8 memory macros of 180 nm CMOS technology. Initially, an 8Kx40 memory block is designed using five 8Kx8 memory macros.

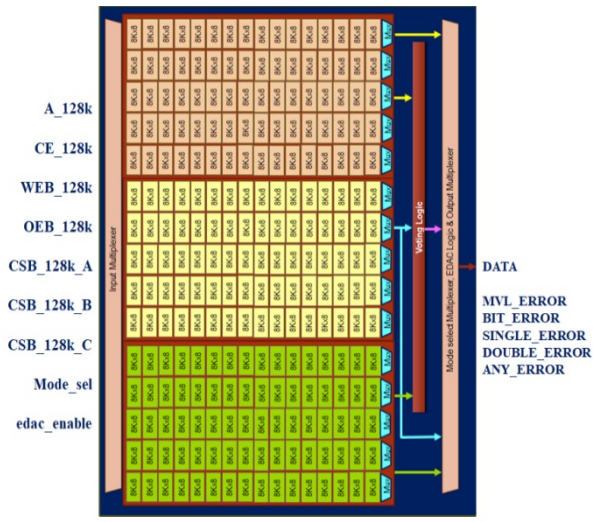

Fig. 1 Block Diagram of SRAM with memory macros

Later, a 128Kx40 capacity memory block is built using 16 units of 8Kx40 capacity memory blocks. To meet the radiation hardened requirement, the 128Kx40 is triplicated (TMR) and the data outputs of the 3 blocks are fed to a Majority Voting Logic (MVL) as shown in Fig. 1. An address decoder is designed to generate the required chip select signals to drive the respective banks. The maximum device capacity of 384Kx40 is achieved using a multiplexer that is controlled using the most significant address bits to select the output data bus of each block in a sequential manner.

A built-in EDAC logic is included in the design for single error correction and double error detection as illustrated in Fig. 2. The EDAC logic is used to internally generate and save check-bits for 32-bit data during a write operation, and to check data from memory for correctness during a read operation in both critical and non-critical modes of operation.

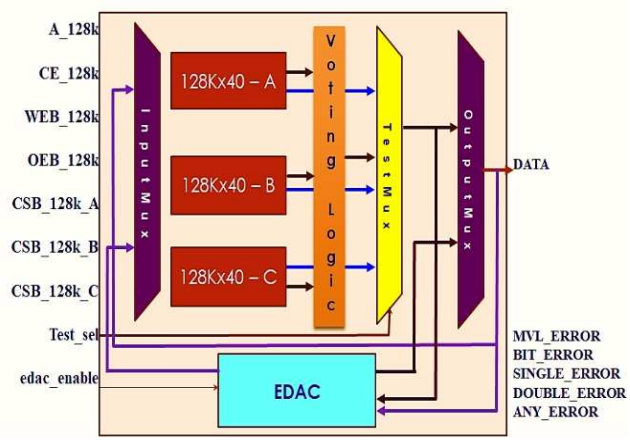

Fig. 2 Functional Block Diagram of SRAM

During a data write, the check-bits can either be written from input pins or can be generated using the inbuilt EDAC logic, and then written along with 32-bit data bits into the memory.

During a read operation, the EDAC logic can generate an indicator flag and correct single bit errors, and also generate an indicator flag for double bit errors without output data correction.

When upsets in memory are not corrected by either overwriting them with new values or scrubbing them, eventually enough SEUs will accumulate to cause the TMR and ECC methods to fail. Hence, error flags help to identify the SEUs generated to refresh the memory contents at the appropriate time.

III. ASIC Implementation and Verification

The ASIC implementation translates the behavioral Register Transfer Logic (RTL) design to actual hardware using standard cells and memory macros from the technology library. This involves two major steps viz., Logic Design and Physical Design. The design flow used for this design is shown in Fig.3.

A. Logic Design

Fig. 3 ASIC Design Flow for Configurable SRAM

The memory design consists of 95% memory macro blocks and 5% logic for address decoding, EDAC and selection logics. As the memory is directly accessible from input-output ports and the remaining logic cells are exercised during read and write operations of memory, additional test structures are not required for defect coverage of the device. Hence, for this device, logic synthesis alone is carried out after lint analysis of the RTL code and functional verification. The functionality of the design is verified through formal equivalence check and functional verification after logic synthesis.

B. Physical Design

Physical design of a device starts with floor planning activity followed by placement of logic blocks and cells, clock tree synthesis, high fan-out net synthesis and signal routing. Initially floor planning is done, with the objectives of minimizing the chip area and signal delays between blocks of the design, by estimating the sizes of various blocks in the design and assigning initial relative locations for these blocks in the die. An efficient floor plan helps in completing the physical design early (i.e., timing closure) by placing functional modules in such a way that makes the critical path lengths shorter and also minimizes the routing congestion.

The first placement strategy used in this memory design is focused on radiation Single Event Effects (SEE) tolerance by placing the memory blocks of 128K X 40: Bank A, Bank B and Bank C adjacent to each other as shown in Fig. 4. To form a memory block of 128K X 40, all the required memory cuts of 8K X 8 are grouped together and placed in the die. This arrangement makes a separation of minimum 39 bit positions for every bit in a bank with its corresponding bit in other banks. In this arrangement, SEU occurrence of same bit in more than one memory block is least probable for a given address, and hence TMR logic can correct the SEU, if it occurs in any one of the memory banks. However, this placement of keeping memory cuts at positions (0,0), (0,1) to (0,15) in a one-dimensional manner adjacent to each other for making a memory block of 128K X 8, results in timing violations. These timing violations are due to signals reaching the multiplexer logic from the output of memory cuts placed at extreme positions (0,0) and (0,15) which fall under the category of longest timing critical paths as shown in Fig. 5. Also routing of the longest paths which create routing congestion is another issue in this placement strategy.

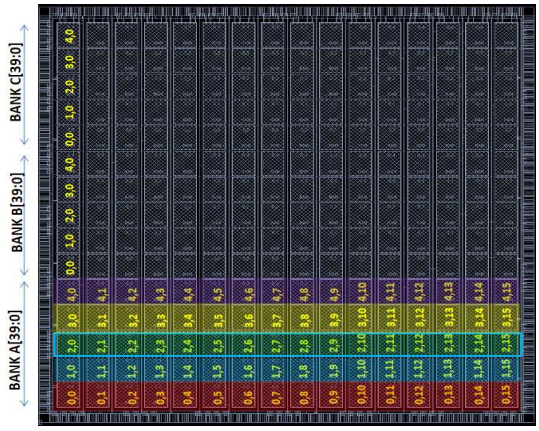

Fig. 4 Placement Strategy 1

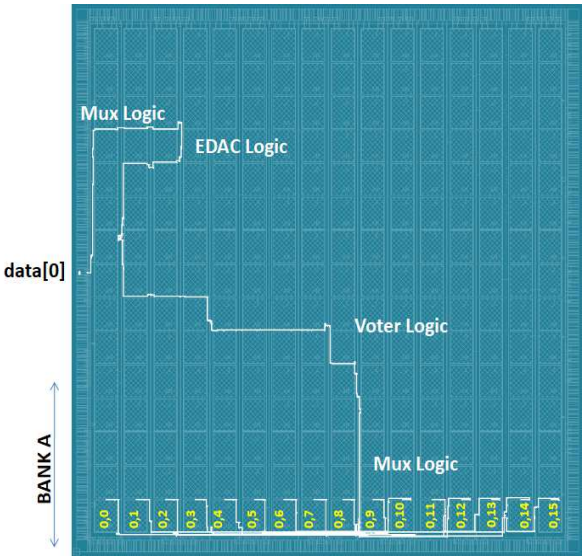

Fig. 5 Longest Critical Path in Strategy 1

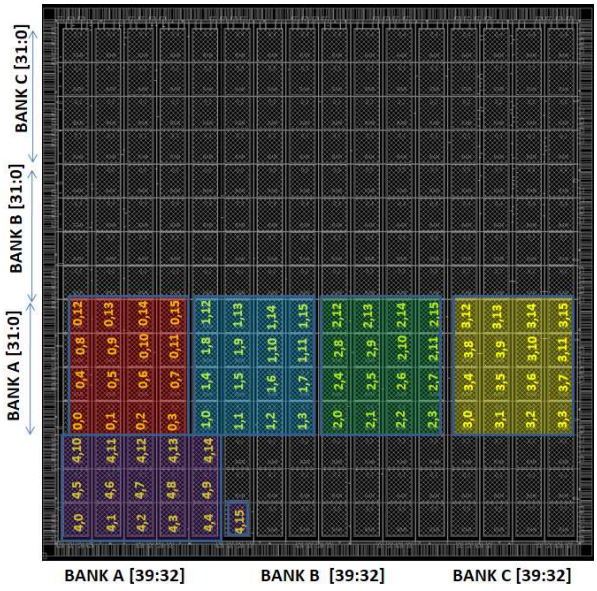

Fig. 6 Placement Strategy 2

In the second placement strategy, to make a memory block of 128K X 8, all the requisite 16 memory cuts are grouped and placed in a two-dimensional manner to reduce the memory cuts output signals length to reach the multiplexer logic, as shown in Fig. 6. This arrangement makes a separation of minimum 23 bit positions for every bit in a memory bank (128 K X 40) with its corresponding bit in other banks. This placement also ensured timing closure (i.e., no timing violations) as well as less routing congestions as shown in Fig. 7.

The QoR report obtained for both placement strategies is presented in Table I. It is clearly seen that the routing congestion, indicated by total number of Global Routing Cells (GRCs), for strategy 2 is much lower than strategy 1. Further, the critical path timing violations (i.e., negative slack for path timing) in strategy 1 are overcome in strategy 2 and the paths have better timing margins, which indicate the improved performance of the device.

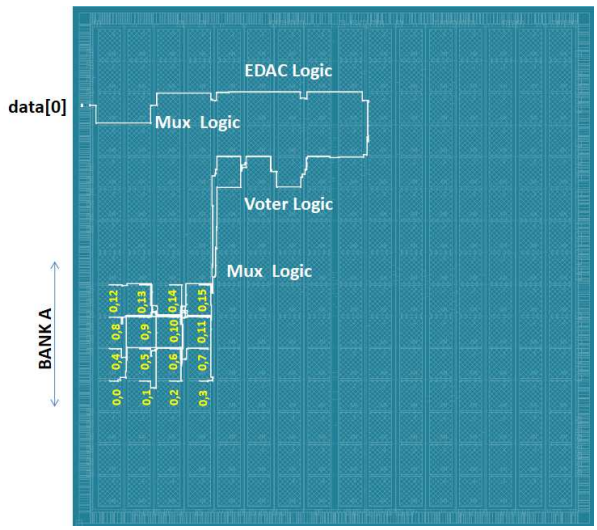

Fig. 7 Longest Critical Path in Strategy 2

TABLE I: TIMING CLOSURE OF STRATEGY 1 AND STRATEGY 2

	Strategy 1			Strategy 2		
	On Placement	Post CTS	Post Route	On Placement	Post CTS	Post Route
Congestion (No. of GRCs)	0	13	1275	0	0	488
Timing Path Groups/Worst critical path timing in the respective path groups						
Combinational	2.28ns	3.24 ns	3.85 ns	2.38 ns	3.33 ns	4.5 ns
Inputs	5 ns	5.01 ns	4.9 ns	5.01 ns	5 ns	4.89 ns
Outputs	-0.96 ns	-3.55 ns	-2.73 ns	1.09 ns	0.16ns	0.18 ns

C. Functional Verification

The functional correctness of the High-Density Memory design is ensured at various phases of development, viz. Behavioral (RTL) level, Post Synthesis (Netlist) level, Post Layout (Netlist) level, and Device (Fabricated chip) level. All modes of operation (TMR without EDAC, TMR with EDAC, non-TMR with EDAC, non-TMR without EDAC) are exercised and verified. Keeping the novelty of design and faster verification closure in mind, the verification targets established for the high density memory design are two-fold:

1. Verify TMR and EDAC based corrections in addition to normal read/write operations
2. Achieve > 95% code coverage

These targets are achieved with a set of directed testcases verifying the correctness of MVL operation (by corrupting data in a single memory bank to see if it gets corrected) and EDAC operation (checking for single bit error detection and correction and double bit error detection) for all memory locations. The verification of TMR proved to be a challenge, given that errors happening in one memory bank might get masked since the error would be auto-corrected by the MVL. Hence, in addition to testing the memory output from MVL logic, scenarios are added to test each memory bank individually for problems.

The testcases to verify the various features of the memory design are listed in Table II.

TABLE II: VERIFICATION SCENARIOS

Test ID	Memory Configuration	Verification Criteria
T1	▪ TMR Off ▪ EDAC Off ▪ Size: 384K x 40 data bits	▪ Write/Read operation to all locations ▪ Read data should not mismatch from write data
T2	▪ TMR On ▪ EDAC Off ▪ Size: 128K x 40 data bits	▪ Write/Read operation to all locations ▪ Read data should not mismatch from write data
T3	▪ TMR Off ▪ EDAC On ▪ Size: 384K x (32 data bits + 8 check bits)	▪ Write/Read data integrity check ▪ Check for EDAC bit generation and storage to memory ▪ Comparison of generated EDAC bits with reference EDAC bit values
T4	▪ TMR On ▪ EDAC On ▪ Size: 128K x (32 data bits + 8 check bits)	▪ Write/Read data integrity check ▪ Check for EDAC bit generation and storage to memory ▪ Comparison of generated EDAC bits with reference EDAC bit values
T5	▪ TMR On ▪ EDAC Off Size: 128K x 40 data bits	▪ Test for MVL logic - error(s) injected to memory location in a single bank ▪ MVL should correct the data bit(s) ▪ MVL error(s) should be raised corresponding to incorrect data bit(s)
T6	▪ TMR Off ▪ EDAC On ▪ Size: 384K x (32 data bits + 8 check bits)	▪ Test for EDAC correction when TMR is off and data in memory is corrupted ▪ Output should have corrected data for single bit error - EDAC single bit error should be high if error is in single bit ▪ EDAC multiple bit error should be high if there are errors in multiple bits
T7	▪ TMR On ▪ EDAC On ▪ Size: 128K x (32 data bits + 8 check bits)	▪ Test for correction when TMR is enabled and data/EDAC check-bit is corrupted ▪ Output should have correct data only ▪ MVL error(s) should be raised corresponding to incorrect data bit(s)
T8	▪ All Configurations	▪ Memory access with chip select disabled ▪ Write data should not get updated (should read back previously stored data) ▪ Data bus should remain in high impedance state when reading with chip select disabled
T9	▪ All Configurations	▪ Memory access with write enable set to 0 (write) and output enable set to 0 (read) ▪ Data in memory is expected to be updated to the write data provided with the wen = 0 and oen = 0 transfer

Tests T1-T4 list the testcases to verify the functionality of the design, by checking the memory in all possible configurations. Tests T5-T7 are error injection tests, to verify the correctness of the TMR and EDAC logics by simulating a corruption of the memory through forces in simulation. Tests T8-T9 are the testcases to verify the design functionality under negative scenarios, like an incorrect combination of control signals. These testcases to verify the functionality of the RTL design, are converted to device test vectors to validate the fabricated memory device.

The design is verified against all test scenarios through simulations, using Synopsys VCS-MX simulator, at the RTL, netlist and device levels.

In addition to directed test cases, the proposed fault-mitigation techniques are verified through fault injection experiments, which are conducted by executing a TCL script within the Synopsys VCS simulation tool. The simulation script is designed to inject a fault to each and every internal signal of SRAM one-at-a-time through a set of simulations, with faults injected at different simulation times.

IV. MEMORY TESTING

SRAM devices become more susceptible to faults when the density of these memories increases as the process technology shrinks. To cover multiple fault types using a minimum number of test vectors, a review of memory faults and standard algorithms was done [15]-[16]. The commonly occurring faults in memories are [15]:

1. Address Faults
2. Coupling Faults: write operation in one cell affecting the value in another cell. Coupling faults may be inversion, idempotent, bridge or state coupling faults.
3. Data Retention Faults
4. Stuck-at Faults: control signals or memory cells stuck at a particular value
5. Transition Faults: when one of the cells or control signals cannot make the transition from 0→1 or 1→0.

March algorithms are the most popular tests used to bring out memory faults. It involves "marching" a pattern up and down the memory by reading and writing values to memory locations. Standard march2 (which is a type of March algorithm) tests cover Address faults, Stuck at faults, Transition faults, inversion and idempotent Coupling faults. To address other fault types like bridging faults, the data pattern written during each operation is modified.

Detection of faults in the SRAM is a time-consuming process. Hence, a customized optimal test using a March algorithm combined with a checker board pattern is designed, with the knowledge of layout of the storage cells, to test the memory devices and detect the manufacturing defects.

A custom MarchC+ (March2) algorithm with a checkerboard style data background is used to cover the maximum number of faults with 7n complexity (i.e., the memory is fully traversed 7 times for either read or write). The algorithm is described below using van de Goor notation [17][18], where Op_{pat} means an operation Op (which may be W for write or R for read) is performed on current address location with data pattern pat. Symbol $\downarrow_{0}^{M-1}$ indicates that the operation proceeds from the largest address M-1 down to address 0, and $\uparrow_{0}^{M-1}$ means the opposite direction. Using this notation, the custom March algorithm may be represented as,

I. $\uparrow_{0}^{M-1} W_p$

II. $\uparrow_{0}^{M-1} R_p ; W_q ; R_q$

III. $\downarrow_{0}^{M-1} R_q ; W_p ; R_p$

where p = 0101..0101 and q = 1010..1010

The memory locations are addressed to their full width of 40-bits by the patterns used. The structure of the memory (128Kx40 memory being constructed by 8Kx8 memory cuts) also limits the faults in an address location to within 7 neighboring bits, and among address locations to the nearest 8192 locations. How the algorithm addresses the different faults is explained as follows:

1. Address faults: The algorithm follows a pattern of $\uparrow_{0}^{M-1} R_p, W_q$ and $\downarrow_{0}^{M-1} R_q, W_p$ where q is an inversion of p [3], which can detect address decoder related faults as a read mismatch.
2. Coupling faults: Steps II and III will be able to detect coupling faults since a coupling fault would result in an incorrect read in the read section of either step.
3. Data Retention Faults: The patterns are read back after waiting for a *delay time* [3] allowing for any data retention faults to show up. A read mismatch here would be able to catch data retention faults.
4. Stuck-at-Faults: The algorithms write each cell with bit values 1 and 0, and expects to reach back the written value. Hence stuck-at faults would be identifiable with a read mismatch.
5. Transition Faults: Again, the ability of a cell to transition both ways is tested in steps II and III of the algorithm, and a read mismatch should be able to identify a transition fault.

Due to the use of a custom pattern of 0x55 and 0xAA - similar to a checkerboard, the custom March algorithm covers bridging faults among adjacent address locations, in addition to the faults covered by March algorithm. The custom March algorithm is run with both TMR on and then TMR off to test each bank of the memory independently.

In addition to memory operation scenarios, to test TMR operation, MVL is tested by writing different data combinations in the three memory blocks to create error scenarios. For EDAC testing, to generate Single Error Flag (SEF), a single bit error is introduced in each location when stepping through the entire 15Mb memory. To generate Double Error Flag (DEF), in addition to the previous error

bit, one more bit error is introduced by stepping through the memory again.

V. Reliability and SEU Rate Prediction

The reliability of the SRAM to SEUs on the storage area and the TMR, EDAC logics have been studied based on a test chip with unhardened memory cuts subjected to radiation experiments. The SEU rate for the test chip is found to be 1E-5 upset/ bit-day using the method given in [19]. Assuming that the upset rate follows exponential distribution, the reliability of the unhardened chip with respect to soft errors is computed using the reliability model given in (1)

$$R(t) = e^{-\lambda t} \quad .. (1)$$

where t is the life time of data in a memory location which is assumed to be a few seconds [20]. In a field application, the new data will get updated into the memory with a particular update rate.

The reliability of TMR protected memory can be estimated using the binomial distribution given in (2)

$$R_{TMR}(t) = 3e^{-2\lambda t} - 2e^{-3\lambda t} \quad .. (2)$$

The purpose of these reliability equations is to propose a methodology to compute the requirement of refresh or update rate for the unhardened and TMR protected memory chips. Generally, the update rate varies from tens of milliseconds to seconds and the choice is based on the reliability requirement for a particular application. We have used equations (1) and (2) to estimate the refresh time-based reliability of unhardened and TMR protected chips.

As shown in Table III, the reliability of bit storage in an unhardened memory, R_{UH}, drops exponentially when it is exposed to radiation environment. Even with a 100 microsecond refresh rate, data is not held in the memory location reliably.

On the other hand, TMR based bit protection has reliability, R_{TMR}, many folds higher than the reliability of unhardened memory as it holds bit information reliably beyond 100 seconds in order to achieve a similar degree of reliability. Therefore, with our approach, a system designer can easily assess the requirement of refreshing memory for memory-based systems. Further, the approach might provide sufficient relaxation to the designer to utilize the system resources in an effective and more efficient manner.

VI. Conclusions

Development of a configurable 15 Mb capacity SRAM device using 180 nm CMOS process technology for high reliability/aero-space applications is a challenging task due to timing closure issues. When the signal has to pass through error mitigation techniques, timing violations are inevitable. In this article it is demonstrated that using intelligent place and route techniques helps in: (i) minimizing the critical paths length so that timing closure happens without any timing violations, and (ii) separation of bit positions to the greatest possible extent so that impact of space radiation is minimized by reducing soft errors at device level. Refreshing or scrubbing the SRAM device is needed to correct the SEU effects. In this context, without proper estimation of refresh rate, either the system performance or reliability gets affected. We presented a method to compute the refresh rate of a memory device and hence keep the reliability at the required level without sacrificing the system performance. However, the effectiveness of the proposed reliability evaluation method is to be validated during operational use of the memory chip. Alternatively, to overcome the SEUs in actual use, programmable refresh rate feature may be included in system software and that shall be one to two orders higher than the computed refresh rate based on the method presented in this article.

TABLE III: Refresh Time Vs. Reliability of the Device

Data life (s)	Data life (h)	upset/day	upset/hour	R_{UH}	R_{TMR}
0.0001	0.0000000278	1.00E-05	4.17E-07	**0.999999999999988**	1.000000000000000
0.1	0.0000277778	1.00E-05	4.17E-07	0.999999999988426	1.000000000000000
1	0.0002777778	1.00E-05	4.17E-07	0.999999999884259	1.000000000000000
10	0.002777778	1.00E-05	4.17E-07	0.999999998842593	1.000000000000000
20	0.005555556	1.00E-05	4.17E-07	0.999999997685185	1.000000000000000
30	0.008333333	1.00E-05	4.17E-07	0.999999996527778	1.000000000000000
40	0.011111111	1.00E-05	4.17E-07	0.999999995370370	1.000000000000000
50	0.013888889	1.00E-05	4.17E-07	0.999999994212963	1.000000000000000
60	0.016666667	1.00E-05	4.17E-07	0.999999993055556	1.000000000000000
70	0.019444444	1.00E-05	4.17E-07	0.999999991898148	1.000000000000000
80	0.022222222	1.00E-05	4.17E-07	0.999999990740741	1.000000000000000
90	0.025	1.00E-05	4.17E-07	0.999999989583333	1.000000000000000
100	0.027777778	1.00E-05	4.17E-07	0.999999988425926	1.000000000000000
150	0.041666667	1.00E-05	4.17E-07	0.999999982638889	**0.999999999999999**
10000	2.777777778	1.00E-05	4.17E-07	0.999998842593262	0.999999999995981
11000	3.055555556	1.00E-05	4.17E-07	0.999998726852662	0.999999999995137
12000	3.333333333	1.00E-05	4.17E-07	0.999998611112076	0.999999999994213
13000	3.611111111	1.00E-05	4.17E-07	0.999998495371502	0.999999999993209
14000	3.888888889	1.00E-05	4.17E-07	0.999998379630942	0.999999999992124
15000	4.166666667	1.00E-05	4.17E-07	0.999998263890396	0.999999999990958
16000	4.444444444	1.00E-05	4.17E-07	0.999998148149863	0.999999999989712

REFERENCES

[1] Bernard Coloma, Patrick Delaunay, Olivier Husson, "High Speed 15ns 4Mbits SRAM for Space Application", Proceedings of the Eighth IEEE International On-Line Testing Workshop (IOLTW'02),2002.

[2] Ne Kyaw Zwa Lwin, H. Sivaramakrishnan, Kwen-Siong Chong, Tong Lin, Wei Shu and Joseph S. Chang, "Single-Event-Transient Resilient Memory for DSP in Space Applications", Proceedings of the 23rd IEEE International Conference on Digital Signal Processing (DSP), 2018.

[3] Duncan Yu Lam, James Lan, Larry McMurchie and Carl Sechen, "SEE-Hardened-by-Design Area-Efficient SRAMs", Proceedings of IEEE Aerospace Conference, 2005.

[4] Jin-Fu Li and Yu-Jane Huang, "An Error Detection and Correction Scheme for RAMs with Partial-Write Function", Proceedings of the IEEE International Workshop on Memory Technology, Design, and Testing (MTDT'05), 2005.

[5] C. Hafer, J. Mabra, D. Slocum, T. Farris, A. Jordan, "SEE and TID Results for a Radhard-by-Design 16Mbit SRAM with Embedded EDAC", Proceedings of the IEEE Radiation Effects Data Workshop, 2006.

[6] Craig Hafer, Jonathan Mabra, Duane Slocum "RadHard 16Mbit Monolithic SRAM for Space Applications", Proceedings of IEEE Aerospace Conference, 2007.

[7] Daniele Radaelli, Helmut Puchner, Skip Wong, and Sabbas Daniel, "Investigation of Multi-Bit Upsets in a 150nm Technology SRAM Device", IEEE Transactions On Nuclear Science, Vol. 52, No. 6, pp. 2433–2437, December 2005.

[8] D. G. Mavis, P. H. Eaton, M. D. Sibley, R. C. Lacoe, E. J. Smith, and K. A. Avery, "Multiple Bit Upsets and Error Mitigation in Ultra-Deep Submicron SRAMs", IEEE Transactions On Nuclear Science, Vol. 55, No. 6, pp. 3288 – 3294, December 2008.

[9] Fei Zhang, Jie Yan, Lixue Ma, Yan Li, and Wu Gao, "Multi-bit Upset Mitigation with Double Matrix Codes in Memories for Space Applications", Proceedings of IEEE International Conference on Unmanned Systems and Artificial Intelligence (ICUSAI), pp. 146 – 149, 2019.

[10] Andrey S. Prokhorov, Sergey F. Tyurin, Maksim A. Chudinov, "A Radiation Resistant Voting Circuit", Proceedings of IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), pp. 215 – 219, December 2018.

[11] Luca Sterpone, Luca Boragno, "Analysis of Radiation-induced Cross Domain Errors in TMR Architectures on SRAM-based FPGAs", IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS), pp. 174 – 179, 2017.

[12] Matthew Cannon, Andrew Keller and Michael Wirthlin, "Improving the Effectiveness of TMR Designs on FPGAs with SEU-Aware Incremental Placement", IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines, pp. 141 – 148, 2018.

[13] Matthew Cannon, Andrew Keller, Corbin A. Thurlow, Andrés Pérez-Celis, and Michael Wirthlin, "Improving the Reliability of TMR With Nontriplicated I/O on SRAM FPGAs", IEEE Transactions On Nuclear Science, Vol. 67, No. 1, pp. 312 – 320, January 2020.

[14] Matthew Cannon, Andrew Keller, Hayden C. Rowberry, Corbin A. Thurlow, Andrés Pérez-Celis, and Michael Wirthlin, "Strategies for Removing Common Mode Failures From TMR Designs Deployed on SRAM FPGAs", IEEE Transactions On Nuclear Science, Vol. 66, No. 1, pp. 207 – 215, January 2019.

[15] Ad J. Van De Goor, "Using March Tests to Test SRAM Devices", IEEE Design & Test Of Computers, pp.8-14, 1993.

[16] Qikai Chen, Hamid Mahmoodi, Swarup Bhuniaand, and Kaushik Roy, "Efficient Testing of SRAM With Optimized March Sequences and a Novel DFT Technique for Emerging Failures Due to Process Variations", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 13, No. 11, pp.1286-1295, November 2005.

[17] Hugo Cheung, Sandeep K. Gupta, "A BIST Methodology for Comprehensive Testing of RAM with Reduced Heat Dissipation", Proceedings of IEEE International Test Conference 1996, pp. 386-395, 1996.

[18] A. J. van de Goor, Testing Semiconductor Memories Theory and Practice, John Wiley & Sons Ltd., West Sussex, England, 1996.

[19] Test Procedures for the Measurement of Single-Event Effects in Semiconductor Devices from Heavy Ion Irradiation, JESD57A, 2017.

[20] G. C. Cardarilli et al., "Design of a fault tolerant solid state mass memory," IEEE Transactions on Reliability, vol. 52, no. 4, pp. 476-491, 2003.