*Turkish Journal of Computer and Mathematics Education Vol.12 No. 7 (2021), 1266-1273*

*Research Article*

# MGDI Based Reliable Low Power Memory Design With Clock Splitting MBIST

**[1]S. Sambasiva Rao Dannina , [2]Dr. Sunil Kumar**

[1]Research Scholar, Dept. of ECE, KALINGA UNIVERSITY, Naya Raipur, Chhattisgarh

[2]Professor, Dept. of ECE, KALINGA UNIVERSITY, Naya Raipur, Chhattisgarh

**Article History**: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 16 April 2021

**Abstract :** In order to minimize ATE (Automatic Test Equipment) time and expense, deep submicron systems contain a large number of memories that demand lower area and quick access time, so an automated test strategy for such designs is needed. When it is upgraded, the memory arrangement becomes complicated. Due to higher rate of memory size incorporation, the device's production expense is declining, and the cost of testing is rising. Memory BIST (Built-in Self-test) is a promising response to this predicament, incorporating test and fix circuitry to the memory itself and offering a reasonable yield. A novel SRAM cell is suggested in this concept and the cell with Fredkin and Feynman gates was planned. For parameter optimizations, the GDI-based reversible gate architecture is implemented. This upgrade requires improved memory construction at the most powerful high throughput and low latency-based density. The updated Low Transition Linear Feedback Shift Register (LFSR) dependent clock splitting technique is built to produce addresses for this SRAM to reach both rows and columns. In addition to implementing the principle of BIST and Decimal Matrix codes, error detection including correction is also incorporated to eliminate each small memory cell for enhanced memory design.

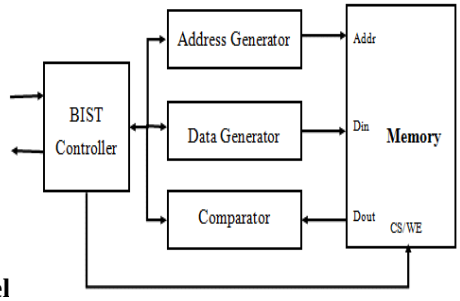
***Keywords***— Built in Self-Test, Linear feedback Shift Register, Clock gating, Gate Diffusion Input, Static Random-Access Memory, Decimal Matrix Code, Automatic Test Equipment, Feynman Gate, Fredkin Gate.

## I. INTRODUCTION

Testing plays a crucial role in any device in detecting faults that deteriorate the efficiency of the system or even contribute to system failure. A significant number of memory cores are now assembled on a single chip with the advancement of deep sub-micron processing technologies and system-on chip (SoC) architecture methodology. External testing of embedded memory cores is a challenging job because the numbers of I/O pins are small. BIST structures are used extensively to address this problem [1, 4]. As the time and cost criteria are minimal, BIST circuitry analysis of circuits in the chip becomes an easy job. Checking the NOC infrastructure elements includes evaluating routers and inter-router communications. Routers, which are primarily filled by FIFO buffers and routing logic, are occupied by a large amount of region of the NOC data transport medium. Consequently, relative to the other elements of the NOC, the probabilities of run-time errors or flaws arising in buffers and logic was substantially larger. Therefore, the NOC infrastructure testing phase must continue with the buffer test and routing logic of the routers. Furthermore, to ensure that no error is accrued, the examination must be done annually. One of the main problems during testing of profoundly scaled CMOS-based memories has been the rare run-time practical faults. These faults are attributed to physical causes, such as sensitivity to the setting, ageing and low supply voltage, and are thus transient in nature (non-permanent suggesting damage or breakdown of the device). These sporadic faults, though, typically show a reasonably high rate of incidence and appear to become irreversible gradually. In comparison, memory wear-out also allows occasional errors to become regular enough to be categorized as irreversible. Therefore, online testing strategies are required to identify run-time errors that are transient in nature, but eventually become persistent over time.

## II.BUILT IN SELF-TEST

It has the ability to be not only quick and effective, but also hierarchical when testing is integrated into the hardware. In other terms, the same hardware will measure processors, boards, and systems in a well-designed test technique. At the device stage, the cost advantages, which might not sound large at the chip level, are immense.




**MBIST Model**

**Fig.1: General MBIST architecture**

The method of checking the verification of the produced chip configuration on automatic test equipment requires the use of external test patterns as a trigger. The response of the computer on the tester is evaluated, contrasting it with the golden response that is retained as part of the data of the test pattern. Through putting all of these functions inside a test circuit surrounding the memory on the chip itself, MBIST makes this possible.

## III PROPOSED M BIST ARCHITECTURE

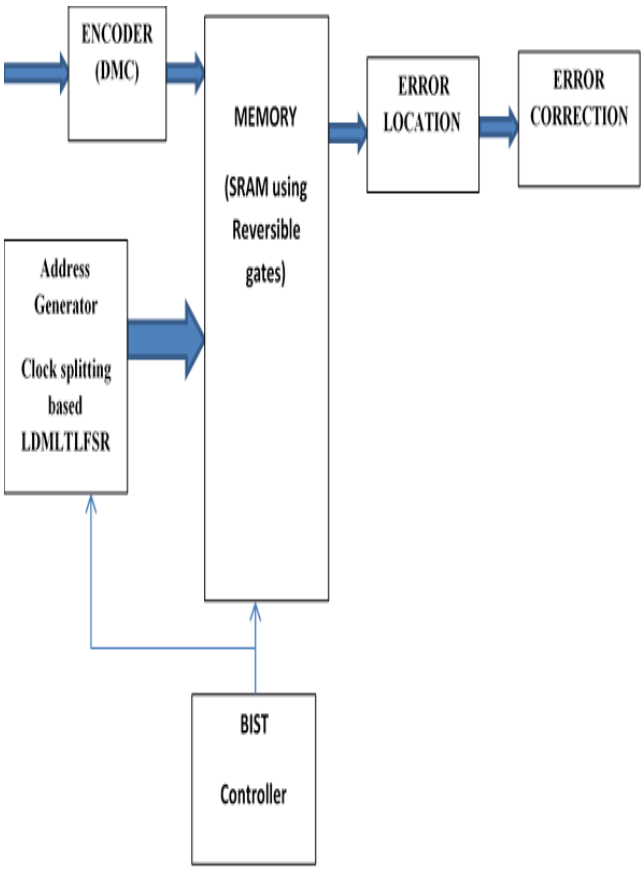



**Fig.2: Proposed MBIST architecture**

The updated Automatic Test Pattern Generator (ATPG) with Low Power Dissipation is the first to be pleased with this whole definition. One typical approach adopted to minimise power consumption is the construction of low transition test pattern generators. In this method, the updated Low Transition Linear Feedback Shift Register (LFSR) dependent clock splitting technique is designed to produce addresses. This paper's key objective is reversible memory with low density and low strength. The cell was constructed with Fredkin and Feynman gates and a new SRAM cell is suggested. The SRAM cell architecture uses one Fredkin gate and one Feynman gate. The overall quantum expense of the proposed SRAM cell is, thus, more easily

minimized. Here, for parameter optimization, the GDI-based reversible gates architecture is implemented in this process. This upgrade requires improved memory construction at the most powerful high throughput and low latency-based density. The design of a new error position and correction system focused on data similarities calculated by the internal product is another enhancement of this principle. The corrupted word can be found by integrating the row similarity matrix and the column similarity matrix, and the error deviation of the corrupted word can be conveniently determined based on the search results. Owing to the features of block-level encoding, this approach produces a substantial reduction in redundancy and time. Again, for more accurate error correction, this approach is combined with Decimal Matrix code. By combining the above strategies, pacing complexities in the above techniques and error correcting rates may be increased. Effective memory the key slogan of this final design is based on self-test with an error prone mechanism and complete fault coverage. Automatic low-power test pattern generator, optimized hardware memory design, and SRAM combinley error tolerant types Effective MBIST architecture fault tolerant.

### 3.1 Address Generator

Linear feedback shift register is used to generate all row addresses for designed SRAM.

**Linear Feed Back Shifts Register**

Linear feedback shift registers (LFSR's) are an efficient way of describing and generating certain sequences in hardware implementations. A linear feedback shift register is composed of a shift register R which contains a sequence of bits and a feedback function f which is the bit sum (XOR) of a subset of the entries of the shift register. The shift register contains n memory cells, or stages, labeled $R_{n-1}$,. . . , $R_1$, $R_0$, each holding one bit. Each time a bit is needed the entry in stage R0 is output while the entry in cell $R_i$ is passed to cell $R_{i-1}$ and the top stage $R_{n-1}$ is updated with the value f(R). The following is a schematic of a linear feedback shift register:

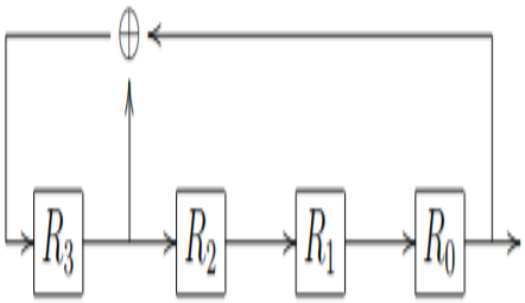



**Fig.3: General Linear feedback shift register**

## IV.LOW DENSITY MODIFIED LOW TRANSITION LP TPG WITH CLOCK SPLITTING LOGIC

The key value of our proposed approach is that it may operate with both combinational and sequential circuits, and the consistency of the models' randomness does not degenerate. There are several suggested random pattern generator methods that only decrease transformations either between the shapes or through an n-bit LFSR between the patterns.

**Clock Splitting**

The technique of clock splitting is used to monitor the clock propagation movement from source to computer. In this principle, the clock is broken into two pieces. The first is accountable for the first half of the digital portion, and the second is responsible for the second half of the digital section. For the clock splitting method, the clock divider dependent diagram is used. When the second clock is disabled, certain parts of the first clock are triggered. Similarly, when the first clock is disabled, certain parts of the second clock are triggered.

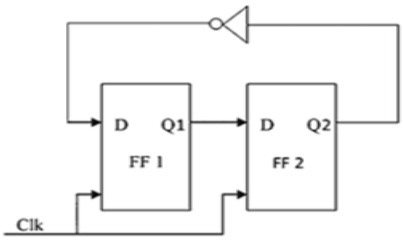



**Fig.4: Clock Splitting logic**

Q1 and Q2 outputs can be used as clocks and those are shown in below figure.

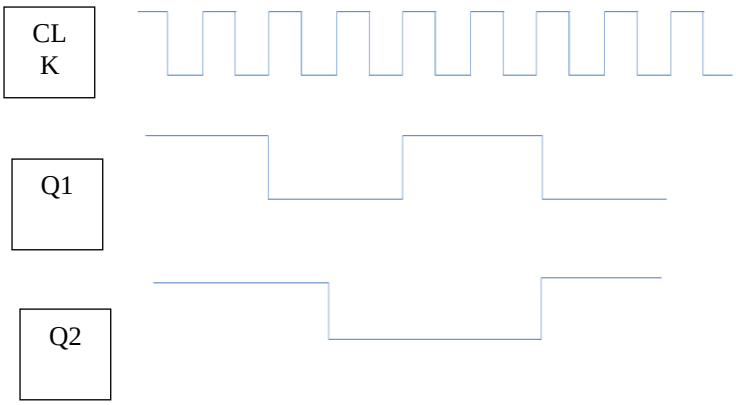



**Fig.5: Clock splitting using single click**

**Clock Splitting With LDMLTLFSR**

More electricity is used by electronic devices such as cell phones, iPods and tablets, which can drain the battery charge within a limited period of time. Most of the dissipation of power is of the dynamic form, which includes a decrease in the dissipation of switching power. There is an option to switch a portion of the circuit that is not in operation for a specific duration in handheld devices. This contributes to the reduction of the device's complex power dissipation by reducing the cumulative power there. Clock splitting is the technique by which part of the architecture may be gated, which ensures that clock signals are not provided to registers that do not alter their state. The power usage of holding the same bit in the flip-flop memory is minimized by this strategy. There are numerous ways in which the division of the clock may be extended to a design. Machine phase, splitting of combinational clocks and splitting of sequential clocks. Splitting a module in a specification may be gated at the device level clock while not in service. The smartphone turns off the light while a mobile is left idle and certain other characteristics that are not used often result in substantial power reduction. During that point, sequential clock splitting switches off the clock, added to the flip-flops in a pipelined configuration, are not in operation. It is difficult to achieve sequential clock splitting and tools are not equipped with the potential to enforce this function, so we have to forecast and check the outcome that is quite difficult to achieve. RTL clock splitting is a procedure in which if the registers meet the requirement, the architecture is evaluated for any condition, then the registers can be clock gated. During code injection, the gating may be performed in the architecture or components for clock splitting are added during the design synthesis. The criteria for a design's clock splitting are that it should have a register feedback mechanism, the MUX's allow signal activation logic should be defined, and the logic conditions that provide the performance should be understood. The gating elements are incorporated according to the conditions that it is necessary to insert clock splitting elements, resulting in a significant reduction in strength. In the clock delivery network, the clock tree layout obtained along with the clock synthesis report offers crucial knowledge about the skew and slack. There is an issue of strong fan outs and flip-flop clock gating by people gating cases, resulting in problem of slack and skew. The dissipation of power may be minimized by utilizing the splitting and combining approaches properly. Split is the mechanism by which the instances of clock gating are split from a set number of registers. Merge is the mechanism by which flip flops in the gates are concurrently combined under a particular instance of clock gating.

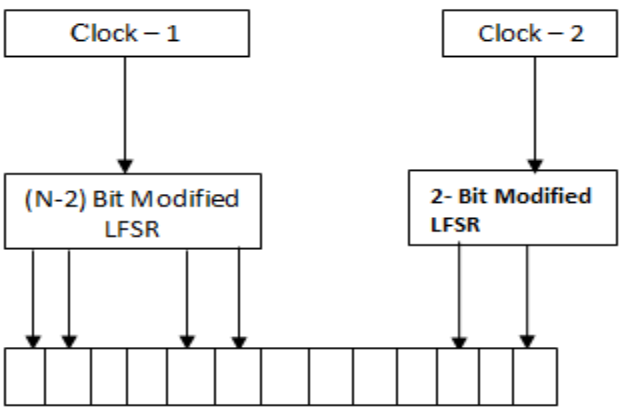



**Fig.6: Proposed clock splitting based LDMLTLFSR**

We are separating the LFSR into (N-2), 2 bit. Two different clock pulses are applied to the 2 bit LFSR and n-2 bit LFSR. When the initial stage of first Flip flop equal to 1 then the next flip flop also getting 1 as a output.
The final output for the first cycle clock was $Q_1$=0 then the second cycle output was $Q_2$=1. Then these $Q_1$ and $Q_2$ are taken as the MSB of the final LFSR generated address and the second operation we needs to divide the clock for second stage LFSR and the resultant of the overall LSB and MSB of the LFSR generated outputs are considered as the final Address of testing of the memory processor.
Here in this we use LT LOW POWER LFSR, in the LFSR Blocks. In order to reduce the number of components and hence the area, in LT LOW POWER LFSR, we replace the MUX with Ex-OR gates. The proposed LDMLTLFSR circuit is given below.

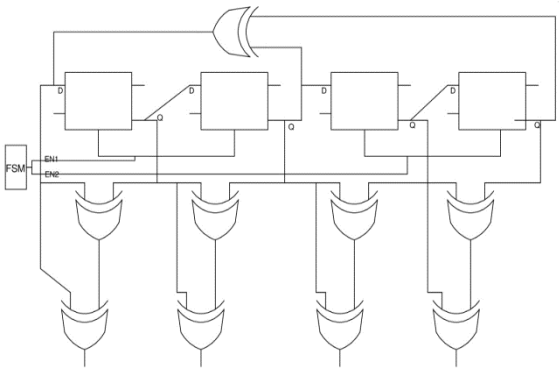



**Fig.7: Proposed LFSR used in clock splitting based LDMLTLFSR**

There is an FSM feature, an LFSR and a data selector unit in the proposed MLT LFSR. The FSM can produce EN1 and EN2 control signals. 1+x+xN is the LFSR polynomial used. MLT LFSR operation for 4 bit circuit is given above.
The usage of basic multiplexers eliminates LT LFSR multiplexers. Instead of using 4 gates for multiplexers, a single XOR gate is used for low density and expense here in this proposed LFSR.

**RAM Using Modified Gate Diffusion Input**

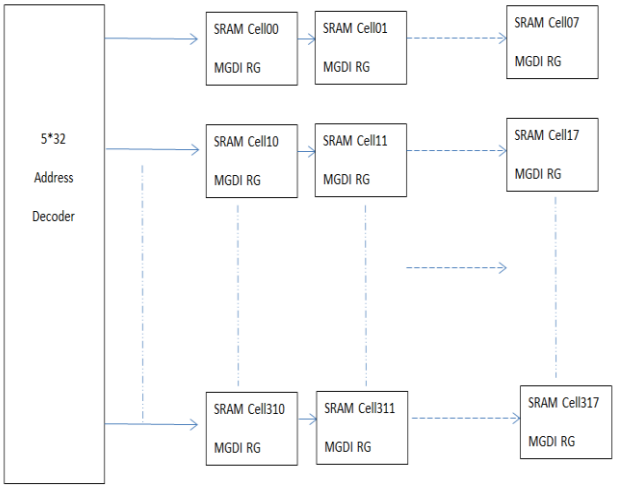



**Fig.8: Proposed GDI based SRAM design**

Above diagram shows the SRAM full architecture using modified GDI SRAM cells with 32 rows and 8 columns.

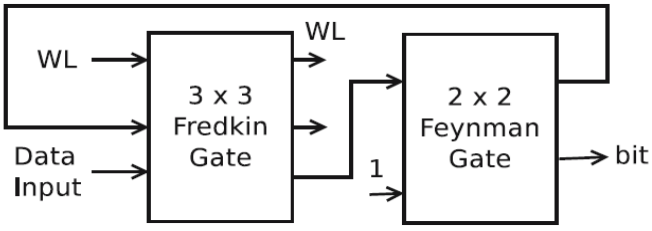



**Fig.9: Proposed Reversible SRAM cell architecture.**

Using two reversible logic gates, Figure 9 demonstrates the suggested completely reversible SRAM cell. The reversible gates of Fredkin and Feyman are used to design SRAM cells here. Word collection (row selection) may be rendered using a WL signal, the "Data Input" signal provides input data. In order to enforce the architecture element, the Fredkin gate requires "AND", "NOT", "OR" gates. Here, updated GDI transistors are used instead of traditional CMOS logic transistors to design logical gates to further save energy and density.

**Fault Tolerant Memory**

**DMC Encoder**

The contribution of this paper is a novel decimal matrix code (DMC) based on divide-symbol is implemented to provide enhanced memory reliability. The implemented DMC utilized decimal algorithm (decimal integer addition and decimal integer subtraction) to identify errors. By using decimal algorithm is that the error detection capability was maximized so that the reliability of memory was enhanced. Besides, the encoder-reuse technique (ERT) was implemented to minimize the area overhead of extra circuits (encoder and decoder) without disturbing the whole encoding and decoding processes, because ERT use DMC encoder itself to be part of the decoder.

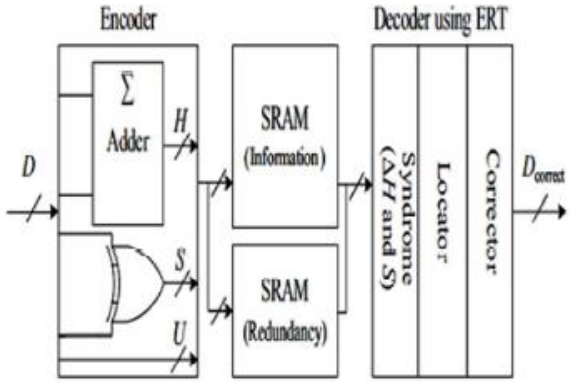



**Fig.10: Proposed fault tolerant unit**

The circuit region of the DMC is reduced in the suggested scheme by reusing its encoder. This is alluded to as ERT. Without disrupting the entire encoding and decoding operations, the ERT will decrease the region overhead of DMC. It can be found that the DMC encoder in the DMC decoder is often reused to retrieve the syndrome bits. As a consequence of utilising the same encoder circuits, the whole circuit region of the DMC will also be reduced. In addition, for determining if the encoder wants to be part of the decoder, this figure also shows the suggested decoder with an allow signal En. In other terms, to separate the encoder from the decoder, the En signal is used, and it is under the power of memory read and write signals. Therefore, the DMC encoder is just an encoder in the encoding (write) phase to perform the encoding operations. However, this encoder is used in the decoding (read) method for the computation of the syndrome bits in the decoder. This specifically illustrate how it is possible to significantly minimize the region overhead of extra circuits.

First, the divide-symbol and arrange-matrix principles are carried out in the proposed DMC, i.e. the N-bit term is broken into k symbols with m bits ($N = k \times m$), and these symbols are grouped in a $k1 \times k2$ 2-D matrix ($k = k1 \times k2$, where k1 and k2 values denote the number of rows and columns in the logical matrix respectively). Second, by performing decimal integer addition of selected symbols per row, the horizontal redundant bits H are created. Here, any symbol is known to be a decimal integer. Third, among the bits per column, the vertical redundant bits V are obtained through binary operation. It should be remembered that divide-symbol and arrange-matrix are both applied in conceptual rather than physical terms. The suggested DMC does not, however, involve a modification in the physical configuration of the memory.

We use a 32-bit word as an illustration, as seen in Fig.11, to illustrate the proposed DMC scheme. Info bits are the cells from D0 to D31. This 32-bit term has been broken up into eight 4-bit symbols. Simultaneously, k1 = 2 and k2 = 4 were picked. H0-H19 are horizontal search bits; the vertical check bits are V0 to V15. It can, however, be remembered that since the separate values for k and m are used, the overall adjustment capacity (i.e. the maximum size of MCUs to be corrected) and the amount of redundant bits are different. To optimise the correction capability and minimise the number of redundant parts, k and m should therefore be carefully adjusted.

In this case, for example, if $k = 2 \times 2$ and m = 8, only a 1-bit error can be resolved, and the number of redundant bits is 80. If $k = 4 \times 4$ and m = 2, it is possible to fix 3-bit errors and the number of redundant bits is decreased to 32. When $k = 2 \times 4$ and m = 4, however, the overall correction capacity is up to 5 bits and the number of redundant bits is 72, respectively. In this article, the error correcting capacity is first regarded to increase the

| PARAMETER | AREA (Gate Count) | TIME (ns) | POWER (mw) |
|---|---|---|---|
| EXISTING | 4320 | 16.079 | 500 |
| PROPOSED | 2274 | 12.532 | 779 |

efficiency of memory, so k = 2 × 8 and m = 4 are used to build DMC.

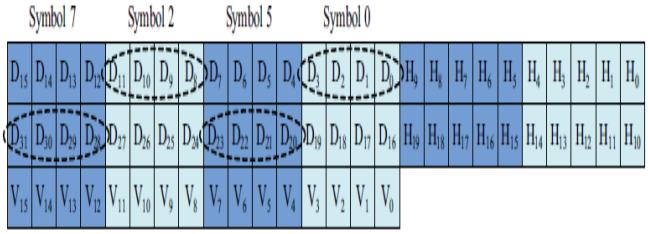



**Fig.11:32 bit word can be divided as 8 symbols with k=2x4 and m=4.**

## V.SIMULATION RESULTS

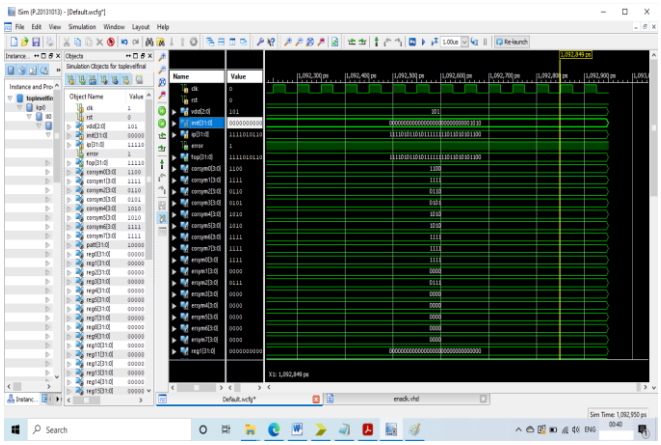


**Fig.12: memory testing with fault**

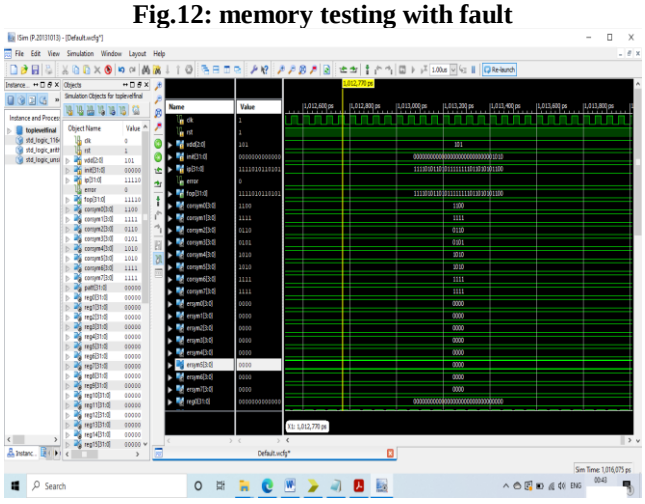


**Fig.13: memory testing without fault**

Above simulation results carried out in XILINXISE14.7 with input data “F5AFF6AC” using 32-bit pattern generator, 32*32 SRAM.

## VI.CONCLUSION

Finally, to ensure the durability of successful memory, a novel per-word DMC was suggested. Decimal algorithms were used to identify errors in the proposed security code, so further errors were found and fixed.

The findings obtained revealed that the proposed device has a superior degree of security against broad in-memory MCUs. In addition, the suggested strategy of detecting decimal errors is an appealing view for the identification of MCUs in reversible SRAM since it is paired with low-power BIST to have a sufficient degree of immunity.

***References***


1. F. Najm, "Transition Density, A New Measure of Activity in Digital Circuits", in IEEE Transaction on Computer Aided Design no.2, vol.12, pp.310-323, February 1993.
2. A. Chandrasekaran, T. Sheng and R.Brodersen, "Low-Power CMOS Digital Design", in IEEE Journal of Solid State Circuits,1992.
3. Mohammad Tehranipoor, MehrdadNourani, Nisar Ahmed, "Low Transition LFSR for BIST Based Applications. January 2006,Proceedings of the Asian Test Symposium.
4. FulvioCorno, Paolo Pinetum, MatteoSonza Ford "Testability Analysis and ATPG on Behavioral RT- Level VHDL", IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 5, May 2015
5. F.Corno, M.Rebaudengo, M.SonzaReorda, "A Test Pattern Generation Methodology for Low Power Consumption", pp.1-5, 2008.
6. Dr. K. Gunavathi, Mr. K. Paramasiva, Ms.P. SubashiniLavanya, M.Umamageswaram, "A Novel BIST TPG for Testing of VLSI Circuits", International journal of computer trends and technology-volume2Isssue-2011.
7. Balwinder Singh, ArunKhosla, SukhleenBindra, "Power optimization of LFSR for Low Power BIST", IEEE International Conference 2009, pp. 311-31.
8. S. W. Golomb, Shift Register Sequences. Laguna Hills, CA: Aegean Park, 1982.
9. T.-C. Huang and K.-J. Lee, "A token scan architecture for low power testing," in Proc. IEEE Int. Test Conf., Baltimore, MD, 2001, pp. 660–669.
10. E. J. McCluskey, "Verification testing—A pseudoexhaustive test techniques," IEEE Trans. Comput., vol. C-33, no. 6, pp. 541–546, Jun. 1984.
11. R. Sankaralingam, B. Pouya, and N. A. Touba, "Reducing power dissipation during test using scan chain disable," in Proc. VLSI Testing Symp., Marina Del Rey, CA, 2001, pp. 319–324.
12. J. Saxena, K. M. Butler, and L. Whetsel, "An analysis of power reduction techniques in scan testing," in Proc. IEEE Int. Test Conf., Baltimore, MD, 2001, pp. 670–677.
13. S. Wang, "Minimizing heat dissipation during test application," Ph.D. dissertation, EE-Systems, Univ. Southern California, Los Angeles, 1998.
14. "Generation of low power dissipation and high fault coverage patterns for scan-based BIST," in Proc. IEEE Int. Test Conf., Baltimore, MD, 2002, pp. 834–843.
15. Ha z Md. Hasan and A.R. Chowdhury, \Design of Reversible Binary Coded decimal Adder by using Reversible 4 bit Parallel Adder, IEEE Trans. Very Large Scale Integr. (VLSI) Jan. 2005.
16. B.Raghu kanth, B.Murali Krishna, M. Sridhar, \A DISTINGUISH BE-TWEEN REVERSIBLE AND CONVENTIONAL LOGIC GATES, In-ternational Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 2,Mar-Apr 2012, pp.148-151.
17. Abu Sadat Md. Sayem, Masashi Ueda, \Optimization of reversible se-quential circuits Journal of Computing Volume 2, Issue 6, June 2010, ISSN 2151-9617.
18. Design of Basic Sequential Circuits Using Reversible Logic By Rohini H Dr. Rajashekar S Dr. Priyatamkumar 2016 IEEE.
19. A Novel Design of Reversible Universal Shift Register with Reduced Delay and Quantum Cost D. Krishnaveni, M. Geetha Priya JOURNAL OF COMPUTING, VOLUME 4, ISSUE 2, FEBRUARY 2012, ISSN 2151-9617.
20. Design of Static and Dynamic RAM Arrays using a Novel Reversible Logic Gate and Decoder by Matthew Morrison, Matthew Lewandowski, Richard Meana and Nagarajan Ranganathan 2011 11th IEEE International Conference on Nanotechnology.
21. A Novel SRAM Cell Design Using Reversible Logic by S Dinesh Kumar and Noor Mahammad Sk 2014 IEEE.
22. E. Ibe, H. Taniguchi, Y. Yahagi, K. Shimbo, and T. Toba, "Impact of scaling on neutron induced soft error in SRAMs from an 250 nm to a 22 nm design rule," IEEE Trans. Electron Devices, vol. 57, no. 7,pp. 1527–1538, Jul. 2010.
23. C. Argyrides and D. K. Pradhan, "Improved decoding algorithm for high reliable reed muller coding," in Proc. IEEE Int. Syst. On Chip Conf., Sep. 2007, pp. 95–98.
24. A. Sanchez-Macian, P. Reviriego, and J. A. Maestro, "Hamming SEC-DAED and extended hamming SEC-DED-TAED codes through selective shortening and bit placement," IEEE Trans. Device Mater. Rel., to be published.
25. S. Liu, P. Reviriego, and J. A. Maestro, "Efficient majority logic fault detection with difference-set codes for memory applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 1, pp. 148–156, Jan. 2012.
26. M. Zhu, L. Y. Xiao, L. L. Song, Y. J. Zhang, and H. W. Luo, "New mix codes for multiple bit upsets mitigation in fault-secure memories," Microelectron. J., vol. 42, no. 3, pp. 553–561, Mar. 2011.
27. R. Naseer and J. Draper, "Parallel double error correcting code design to mitigate multi-bit upsets in SRAMs," in Proc. 34th Eur. Solid-StateCircuits, Sep. 2008, pp. 222–225.
28. G. Neuberger, D. L. Kastensmidt, and R. Reis, "An automatic technique for optimizing Reed-Solomon codes to improve fault tolerance in memories," IEEE Design Test Comput., vol. 22, no. 1, pp. 50–58, Jan.–Feb. 2005.