A Novel Fault Detection and Correction using Xilinx ISE

CH.Manjusha
Associate Professor, Department of ECE, Narayana Engineering College, Gudur, AP, 524101

C.Chaithanya, K.Jhansi, K. V. Sai Krishna, M.Yamini Indu Priya
UG Student, Department of ECE, Narayana Engineering College, Gudur, AP, 524101

Abstract: Due to scaling of technology and increased density of integration there may be parameter variations and levels of noise which might cause more error rates at different computational levels. With regards to memory applications soft errors and single event errors might always be problematic. The main objective of this paper is to design an efficient Majority Logic Detector/Decoder (MLDD) for detecting as well as correcting faults in memory applications by maintaining sufficient reduction in detection time. One method is using one step majority logic decoding technique which is implemented for Euclidean Geometry Low Density Parity Check Codes (EG-LDPC). Majority decodable codes are capable of correcting errors to a greater extent; however, they require more decoding time for error detection and the majority logic decoding method might take same time for code words with and without errors. This might cause delay in the performance of memory. A new fault detection method is introduced here which can detect faults in minimal decoding cycles. If the data can be read without error, memory access time may be reduced and power consumption may be less even for large size code words.

Keywords: One Step Majority Logic Decoding; Error Detection & Correction Codes; Euclidean Geometry Low-Density Parity Check (EG-LDPC); Memory; Single event errors.

I.INTRODUCTION

Memories are very important components in logic circuits today. For a very long period, many steps have been taken to protect memory cells from soft errors. Embedded memory, such as ROM, SRAM, DRAM, flash memory etc is present in majority of system chips. Memory failure rates are increasing now a days, because of impact of technology scaling-smaller dimensions, high density of integration, lesser operating voltages etc. The key to high reliability and availability of applications is how quickly it can determine that a bit has flipped. Some of the frequently used error detecting methods are Triple Modular Redundancy (TMR) and Error Correction Codes (ECCs).

The disadvantage with Triple Modular Redundancy (TMR) method is large area and overhead complexity. Therefore, Error Correction Codes (ECCs) was found to be the best way to protect memories from soft errors. Common ECC codes are Single Error Correction (SEC) codes which can correct one bit error, Double Error Correction (DEC) codes, Multi error correction codes, such as Reed– Solomon (RS) or Bose Chaudhuri–Hocquenghem (BCH). However, these are not feasible due to complex decoding algorithms.

Another type of ECC codes having high error correction capability and less decoding complexity are the Cyclic block codes which have Majority Logic (ML) decodable properties. The advantages with ML decoding are simple implementation and less complexity. However, it has disadvantages of requiring as many cycles as number of bits in the incoming signal, requirement of same decoding time for both erroneous and error free code words. An alternative for this is to first detecting if there are errors in the word and only when there are errors to perform the rest of the decoding process which can greatly reduce the average power consumption if majority code words have no errors.

This project mainly focuses on using the Majority Logic Decoder circuitry itself as an error detection module therefore requiring no additional hardware for data read.

II.LITERATURE SURVEY

"DEC ECC design to improve memory reliability in sub-100nm technologies", R. Naseer and J. Draper[1], Exacerbated SRAM reliability issues, due to soft errors and increased process variations in sub-100 nm technologies, limit the efficacy of conventionally used error correcting codes (ECC). The double error correcting (DEC) BCH codes have not found favorable application in SRAMs due to non-alignment of their block sizes to typical memory word widths and particularly due to the large multi-cycle latency of traditional iterative decoding algorithms. This work presents DEC code design that is aligned to typical memory word widths and a parallel decoding implementation approach that operates on complete memory words in a single cycle.

"Efficient majority fault detection with difference set codes for memory applications", Shih-Fu Liu, Pedro Revingo, and Juan Antonio Meastro[2], Nowadays, single event upsets (SEUs) altering digital circuits are becoming a bigger concern for memory applications. This paper presents an error-detection method for difference-set cyclic codes with majority logic decoding. Majority logic decodable codes are suitable for memory applications due to their capability to correct a large number of errors. However, they require a large decoding time that impacts memory performance. The proposed fault-detection method significantly reduces memory access time when there is no error in the data read.

"Models and algorithmic limits for an ECC- based approach to hardening sub-100-nm SRAMs", M.A. Bajura et al[3] , A mathematical bit error rate (BER) model for upsets in memories protected by error-correcting codes (ECCs) and scrubbing is derived. This model is compared with expected upset rates for sub-100-nm SRAM memories in space environments. Because sub-100-nm SRAM memory cells can be upset by a critical charge (Q_{crit}) of 1.1 fC or less, they may exhibit significantly higher upset rates than those reported in earlier technologies. Because of this, single-bit-correcting ECCs may become impractical due to memory scrubbing rate limitations.

"Radiation-induced soft errors in advanced semiconductor technologies", R. C. Baumann[4], The once-ephemeral radiation-induced soft error has become a key threat to advanced commercial electronic components and systems. Left unchallenged, soft errors have the potential for inducing the highest failure rate of all other reliability mechanisms combined. This article briefly reviews the types of failure modes for soft errors, the three dominant radiation mechanisms responsible for creating soft errors in terrestrial applications, and how these soft errors are generated by the collection of radiation-induced charge.

"Fault secure encoder and decoder for Nano Memory applications," H. Naeimi and A. DeHon[5], Memory cells have been protected from soft errors for more than a decade; due to the increase in soft error rate in logic circuits, the encoder and decoder circuitry around the memory blocks have become susceptible to soft errors as well and must also be protected. We introduce a new approach to design fault-secure encoder and decoder circuitry for memory designs. The key novel contribution of this paper is identifying and defining a new class of error-correcting codes whose redundancy makes the design of fault-secure detectors (FSD) particularly simple. We further quantify the importance of protecting encoder and decoder circuitry against transient errors, illustrating a scenario where the system failure rate (FIT) is dominated by the failure rate of the encoder and decoder.

"Low-density parity check codes for error correction in nano scale memory", S. Ghosh and P. D. Lincoln[6], The continued scaling of photolithographic fabrication techniques down to 32 nano meters and beyond faces enormous technology and economic barriers. Self- assembled devices such as silicon nano wires or carbon nano tubes show promise to not only achieve aggressive dimensions, but to help address power and other gating issues in system architecture, while potentially helping contain rampant increases in fabrication capital costs. However, assembling high-quality, large- scale nano electronic circuits (e.g., with Langmuir-Blodgett or related methods) has proven challenging. Among the major challenges are extremely high defect and fault rates in assembled devices. Apart from fabrication errors, nano scale devices are also more prone to soft errors than microscale devices. Current-day micro scale devices (e.g., gates, PLAs, memories) constructed using top-down lithographic techniques have error rates of less than 1% (10).

"Construction of cyclic codes suitable for iterative decoding via generating idempotent," T. Shibuya and K. Sakaniwa[7] , A parity check matrix for a binary linear code defines a bipartite graph (Tanner graph) which is isomorphic to a sub graph of a factor graph which explains a mechanism of the iterative decoding based on the sum-product algorithm. It is known that this decoding algorithm well approximates MAP decoding, but degradation of the approximation becomes serious when there exist cycles of short length, especially length 4, in Tanner graph. In this paper, based on the generating idempotents, we propose some methods to design parity check matrices for cyclic codes which define Tanner graphs with no cycles of length 4. We also show numerically error performance of cyclic codes by the iterative decoding implemented on factor graphs derived from the proposed parity check matrices.

"Cache and memory error detection, correction, and reduction techniques for terrestrial servers and workstations" ,C. W. Slayman[8], As the size of the SRAM cache and DRAM memory grows in servers and workstations, cosmic-ray errors are becoming a major concern for systems designers and end users. Several techniques exist to detect and mitigate the occurrence of cosmic-ray upset, such as error detection, error correction, cache scrubbing, and array interleaving. In most system applications, a combination of several techniques is required to meet the necessary reliability and data-integrity targets.

"Characterization of multi-bit soft error events in advanced SRAMs ", J. Maiz[9], Error correction code schemes are being implemented in memories and microprocessor caches in response to SER increases which result from increasing bit counts and technology scaling. These methods can be rendered ineffective by multi-bit error events. An exhaustive characterization of multi-bit errors in 90/130 nm SRAMs is presented to support bit interleaving rules that make the impact of multi-bit errors negligible. "Multi-bit error correction methods for latency-constrained flash memory systems", P. Ankolekar , S. Rosner , R. Isaac, and J. Bredow[10], As architectures advance to accommodate more bits/cell and geometries decrease to structures that are smaller than 32 nm, single-bit error-correction codes (ECCs) are unable to compensate for the increasing array bit error rates, making it imperative to use 2-b ECC.

III. PROPOSED WORK

Majority Logic Decoder & Detector:

The modified MLDD algorithm performs error detection similar to existing plain ML decoder, however with some modifications. This algorithm requires additional logic compared to existing algorithm. Corrections are performed during the first n iterations. Using a counter it is checked if there are more than t errors in those n iterations and depending on the result the corrected register is sent to the output. In case the majority gate detects any error in code word, number of iterations depends on the code

word length. Modified MLDD can detect more than five bit-flips and is efficient compared to existing majority logic decoder.

The figure below shows the ML decoder with a 15-tap shift register, an XOR array to calculate the orthogonal parity check sums and a majority logic circuit to decide if the current decoding bit is erroneous and the need for its inversion.

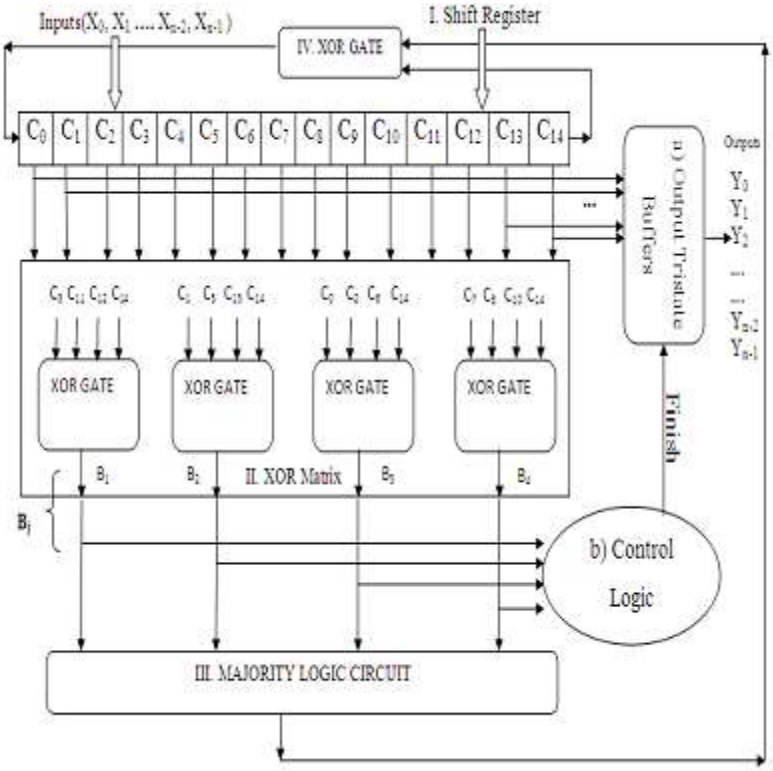

Figure 1: Schematic of the proposed MLDD for 15 bit code word

The circuit mainly consists of,

- Control logic unit and
- Output tristate buffers.

Detection process is managed by the control unit which uses a counter that counts up to three, distinguishing the first three iterations of the ML decoding. Data bits stay in memory for few cycles and during this time, memory bits can be flipped by a transient fault with certain probability. The process will be stopped after three iterations if the codeword length is less than ten and six iterations for codeword length less than twenty, if the codeword length is greater than 20 then nine iterations are performed. In order to avoid accumulation of many errors in any memory word, the system performs memory scrubbing which is the process of reading memory words periodically from the memory, correcting potential errors if any and writing them back into the memory.

Control unit triggers a finish flag when there are no errors detected in data read. The output tristate buffers will be in high impedance state until the control unit sends the finish signal so that the present values are transferred to the output y through the shift register.

Control Unit:

It is the most important unit of MLDD system that controls and manages the entire detection process. It uses a counter which counts up to three iterations. In these three iterations, the control unit evaluates by combining them with the XOR1 function. This value is then fed into a three-stage shift register. In the third cycle, the XOR2 gate evaluates detection register contents. When the result is "0," finish signal is sent indicating that the processed word has no error. Otherwise, if the result is "1," the ML decoding process runs till the end. This provides a performance improvement with respect to the traditional method.

Most of the words would usually take three cycles. The majority logic gate is implemented using the conventional majority logic decoding mechanism. If during the memory read access an error is detected, the XOR gate will correct it, by inverting the current bit under decoding.

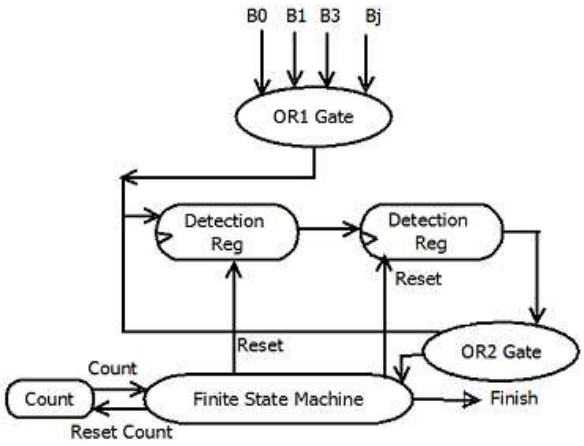

Figure 2: Schematic of the control unit

The flow chart for MLDD method as shown in Fig: 2. Iteration from 1 to 3 are performed for detecting errors in the codeword. If the codeword has no error, then output can be obtained in three iterations. Otherwise, iterations are continued to detect and correct the errors in the codeword. Iterations from 4 to N will be preformed to correct the error in the codeword. If an error is detected, it will be corrected, otherwise, retransmission will be processed.

Properties of MLDD:

Properties of the Majority Logic Decoder and Detector (MLDD) are:

- Ability to correct large number of errors.
- Sparse encoding, decoding and checking circuits synthesizable into simple hardware
- Modular encoder and decoder blocks that allow an efficient hardware implementation
- Systematic code structure for clean partition of information and code bits in the memory.

IV. SIMULATION RESULTS

In this section the simulation results of the proposed Majority Logic Decoder/Detector (MLDD) and the existing Plain MLD are presented. The front end design of the architecture, its simulation, synthesis and comparison are done and the tools used are Modelsim 10.1c for simulation, XILINX ISE Design Suite 10.1 for synthesis and comparison. The target device is Spartan 3-XC3S400. The designs are coded in verilog language. A codeword of size 15 and information bits of 7 is chosen here for designing.

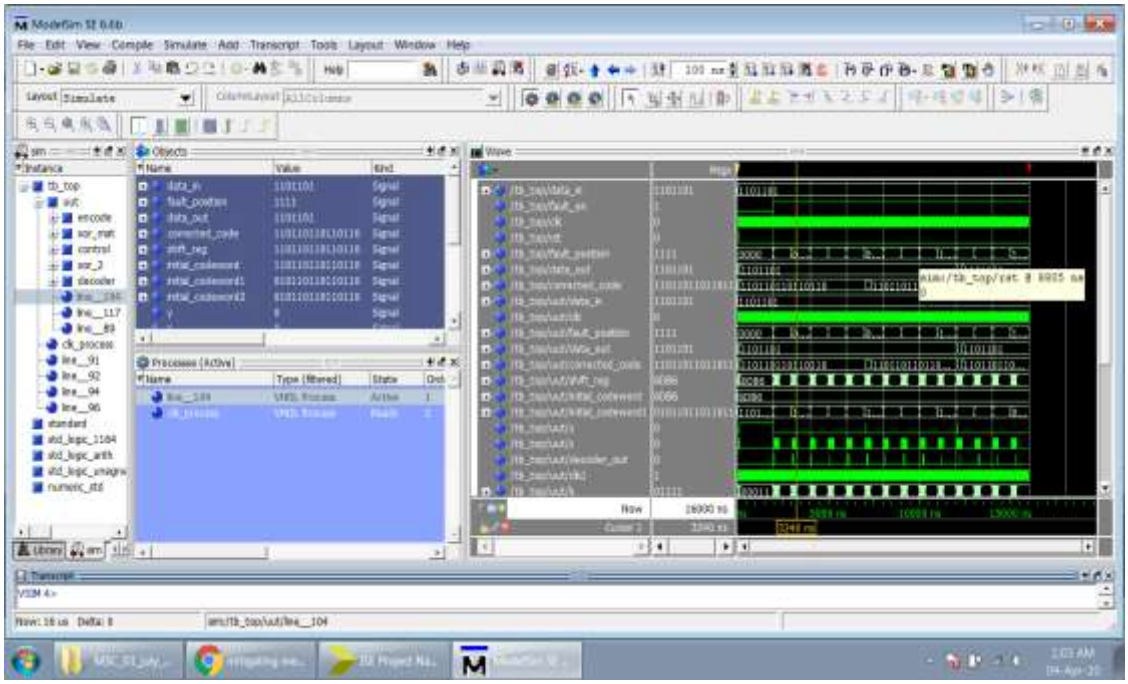

Figure 3: Existing technique MLD without fault in memory access

In the plain MLD, delay in memory read access is directly dependent on the code size, i.e., for a code 15 length needs 15 cycles etc. Then, for I/O two extra cycles needed. For the detection of errors, MLD always needs N+2 cycles in all cases. The above simulation result Fig.7.1 shows the existing technique MLD without fault in memory access, it has taken N+2 cycles to detect and correct the error, in the above examples MLD used 17 cycles of size (15, 7, 5).

novel_fault_detection_top Project Status			
Project File:	mic_01_mple.xise	**Parser Errors:**	No Errors
Module Name:	novel_fault_detection_top	**Implementation State:**	Programming File Generated
Target Device:	xc3s500e-4fg320	**•Errors:**	No Errors
Product Version:	ISE 13.2	**•Warnings:**	10 Warnings (0 new)
Design Goal:	Balanced	**•Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	**•Timing Constraints:**	All Constraints Met
Environment:	System Settings	**•Final Timing Score:**	0 (Timing Report)

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Total Number Slice Registers	55	9,312	1%	
Number used as Flip Flops	40			
Number used as Latches	15			
Number of 4 input LUTs	142	9,312	1%	
Number of occupied Slices	106	4,656	2%	
Number of Slices containing only related logic	106	106	100%	
Number of Slices containing unrelated logic	0	106	0%	
Total Number of 4 input LUTs	144	9,312	1%	
Number used as logic	142			
Number used as a route-thru	2			
Number of bonded IOBs	36	232	15%	
Number of BUFGMUXs	1	24	4%	
Average Fanout of Non-Clock Nets	2.93			

Figure 4: Design summary for the proposed MLDD without fault in memory access

A schematic representation of our synthesized source file after the HDL synthesis phase of the synthesis process is shown. The below figures shows the representations of the pre-optimized design in terms of generic symbols, such as adders, multipliers, counters, AND gates, and OR gates, that are independent of the targeted Xilinx device. This schematic may help to discover design issues early in the design process. RTL schematic representation of the pre-optimized design shown at the Register Transfer Level (RTL).

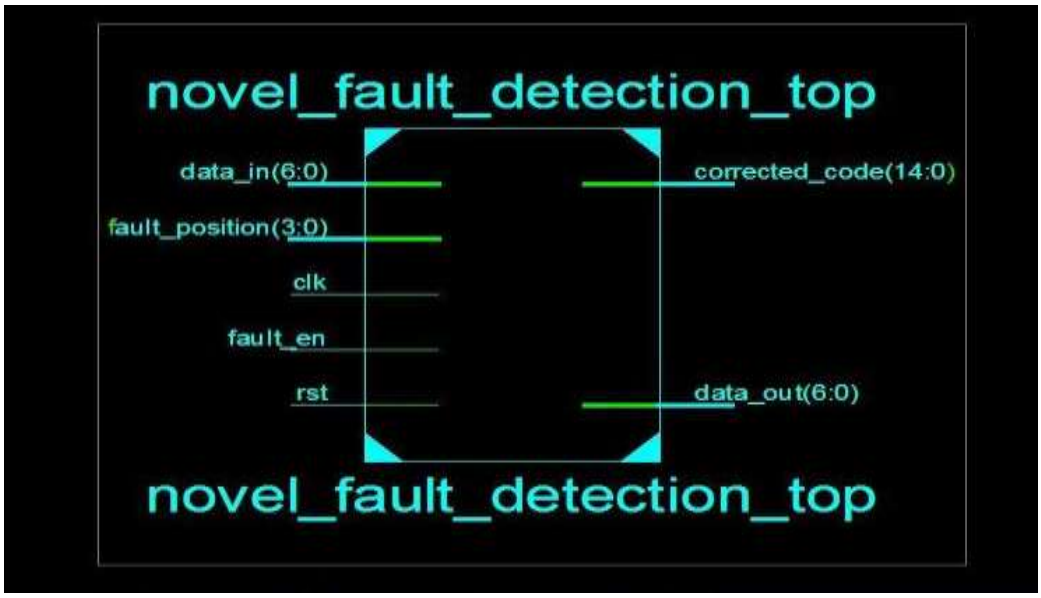

Figure 5: RTL view of the proposed technique MLDD with fault in memory access

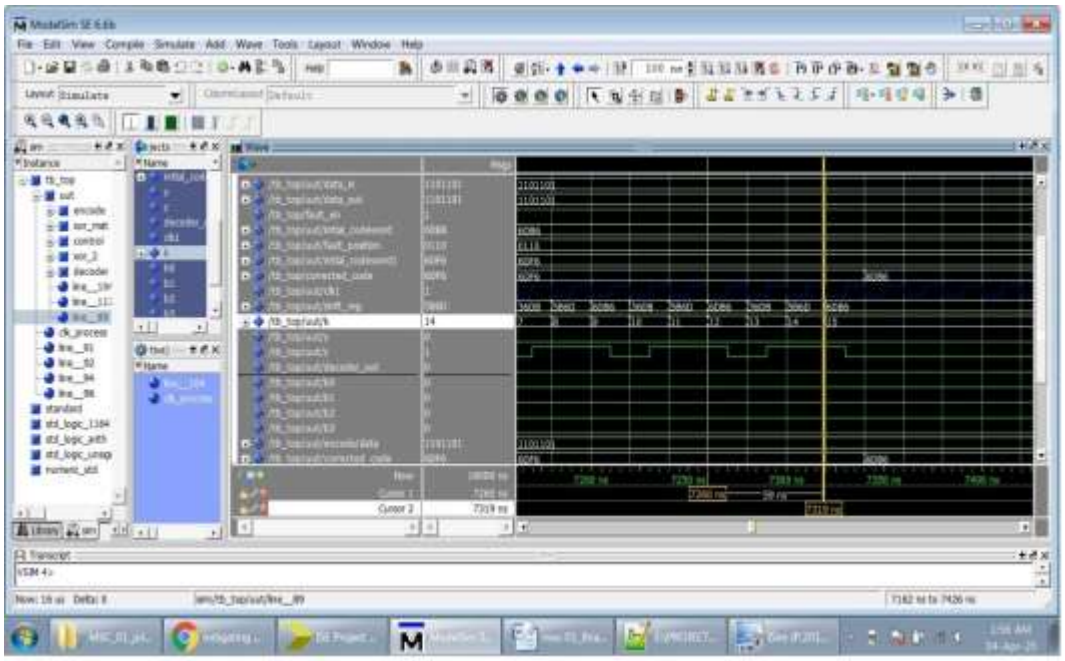

Figure 6: Proposed technique of MLDD waveform

The synthesized design can be viewed in a technology schematic viewer as a schematic. This schematic view displays Gates and elements as to how they will appear on the Xilinx device. After the optimization and technology targeting phase of the synthesis process, a schematic representation of synthesized source file can be displayed.

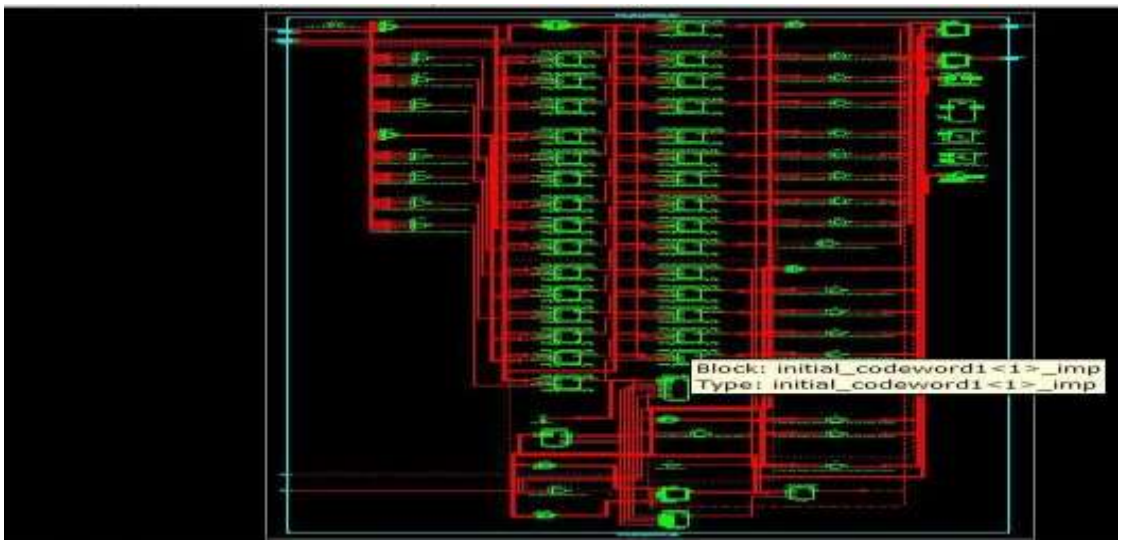

Figure 7: RTL schematic for the proposed technique MLDD with fault in memory access

This schematic technology shows a representation of the design in terms of its logic elements optimized to the target Xilinx device or "technology," for example, in terms of of LUTs, carry logic, I/O buffers, and other technology-specific components. This schematic view allows to see a technology-level representation of HDL optimized for a specific Xilinx architecture, which may help discovering design issues early in the design process.

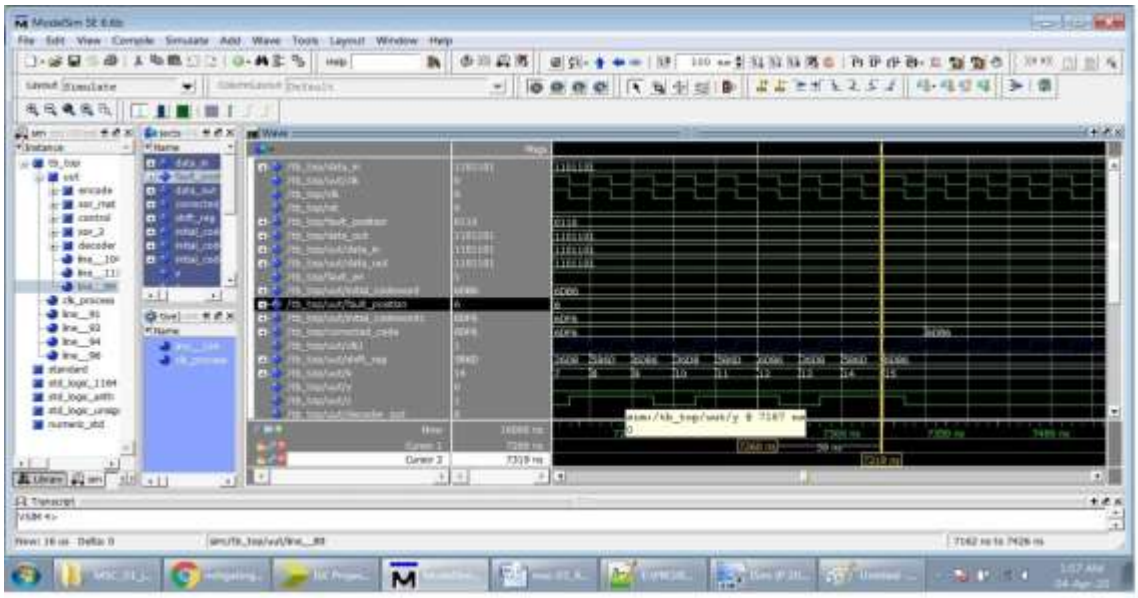

Figure 8: The inner modules (encoder, decoder, control logic , xor _matrix) waveform

V.CONCLUSION

In this paper a modified Majority Logic Decoder Detector (MLDD) has been implemented. The simulation results show that all combinations tested with errors affecting up to four bits are detected in the first three iterations. MLDD error detector is designed to be independent of the code word size and the area overhead of the MLDD decreases compared to the plain MLD technique. Therefore the proposed MLDD will be an efficient design for fault detection and correction.

VI.REFERENCES

1. R. C. Baumann, “Radiation-induced softerrors in advanced semiconductor technologies,” IEEE Trans. Device Mater. Reliab., vol. 5, no. 3,pp. 301–316, Sep. 2005.

   M. A. Bajura, Y. Boulghassoul, R. Naseer, S. DasGupta, A. F.Witulski,J. Sondeen, S. D. Stansberry, J. Draper, L. W. Massengill, and J. N.Damoulakis, “Models and algorithmic limits for an ECC-based approachto hardening sub-100-nm SRAMs,” IEEE Trans. Nucl. Sci., vol.54, no. 4, pp. 935–945, Aug. 2007.

2. R. Naseer and J. Draper, “DEC ECC design to improve memory reliabilityin sub-100 nm technologies,” Proc. IEEE ICECS, pp. 586–589,2008.

3. S. Ghosh and P. D. Lincoln, “Dynamic low-density parity check codesfor fault-tolerant nano-scale memory,” presented at the FoundationsNanosci. (FNANO), Snowbird, Utah, 2007.

4. S. Ghosh and P. D. Lincoln, “Low-density parity check codes for errorcorrection in nanoscale memory,” SRI Computer Science Lab., MenloPark, CA, Tech. Rep. CSL-0703, 2007.

5. H. Naeimi and A. DeHon, “Fault secure encoder and decoder formemory applications,” in Proc. IEEE Int. Symp. Defect Fault Toler.VLSI Syst., 2007, pp. 409–417.

6. B. Vasic and S. K. Chilappagari, "An information theoretical frameworkfor analysis and design of nanoscale fault-tolerant memoriesbased on low-density parity-check codes," IEEE Trans. Circuits Syst.I, Reg. Papers, vol. 54, no. 11, pp. 2438–2446, Nov. 2007.