A New Efficient Self-Checking Hsiao SEC-DED Memory Error Correcting Code

Fradi Aymen
Electronic & Microelectronic
Laboratory.
Monastir Tunisia
Email: fradi.aymen@gmail.com

Hamdi Belgacem
ISSAT, Sousse
Electronic & Microelectronic
Laboratory.
Monastir Tunisia

Khedhiri Chiraz
Electronic & Microelectronic
Laboratory.
Monastir Tunisia

***Abstract*— Interest in on-line error detection continues to grow as VLSI circuits increase in complexity. Concurrent checking is increasingly becoming a desirable characteristic thanks to its ability to detect transient faults that may occur in a circuit during normal operation. Accordingly, Concurrent Error Detection (CED) techniques allow the detection of transient faults, which probably not be detected in off-line testing, since they may not occur in test mode. Actually, memories occupy 90% of the SOC area. As they are a fault sensitive, adding error correcting codes (ECC) structures is becoming conventional to enhance the reliability. Hence, designing the ECC logic must ensure not only responding to nanotechnology requirements as high density, reduced power consumption and faster calculation delays but also to be a fail-safe.**

In this paper, a new self-checking error correcting SEC-DED code architecture is presented. We added the self-checking capability to the SEC-DED circuit using a self-checking differential XOR implemented in Complementary Pass Transistor Logic (CPL). We have selected the Hsiao code to design an efficient SEC-DED.

***Index Terms*— Error correcting codes, self-checking circuits, differential XOR, Complementary-Pass-Transistor Logic.**

I. Introduction

Nowadays, memory manufacturers are facing many problems of reliability and yield. Initially, the chips are extensively tested using A.T.E and embedded Built In Self Test (BIST) to be validated as fault free. Recently, very high density memories in deep submicron technology are sensitive to failures. Then many parts are rejected after test causing great losses at the end of manufacturing. To increase the yield, hardware redundancy techniques (BISR: Built In Self Repair) are used by substituting faulty bits with spare columns and/or rows. These repair solutions cannot be used for errors that can occur in the operational mode. To cope with this problem, fault tolerant information redundancy techniques using Error Correcting Codes (ECC) were developed. In reality, these two redundancy techniques can be used all together to solve the memory yield and reliability problems.

In this paper, we discuss only the Error Correcting codes structures. They are simple codes using added parity bits to data memory. They can detect and correct a little number of errors and they are suitable for integration in most of industrial memory products [1].

However, due to radiation, power disturbance and noise not only memories are more prone to errors but also the ECC logic. Indeed, the environment and the manufacturing process can produce degradations in the parity bits generator and syndrome generator logics which can generate faults. To cope with this problem, online detection techniques must be designed assuring a continuous operation of the ECC. Also to be efficient in submicron scale, some parameters as power consumption, delay and noise immunity must be considered mainly in the transistor level.

In this paper, a new self-checking error correcting SEC-DED codes architecture is presented. We added the self-checking capability to the SEC-DED circuit using a self-checking differential XOR implemented in Complementary Pass Transistor Logic (CPL).

In section II we describe the background of the Hsiao SEC-DED code and of the self-checking techniques. In section III, we discuss an overview of the complementary pass transistor logic. In section IV, the proposed self-checking SEC-DED is presented. In section V, we finish with conclusions.

II. Background

A. *SEC-DED overview*

In the deep submicron, the use of Error correcting Codes for RAM is becoming conventional. The well known binary linear codes is SEC-DED Hamming and Hsiao codes [1][2]. They can correct single error and detect double errors as well as transient (soft) or hard errors. Also, as the construction of an ECC code depends on its parity matrix H, we have selected Hsiao codes implementation using minimum odd weight columns to build an efficient ECC circuitry. We describe below how to build Hsiao SEC-DED code.

1) *SEC DED Hsiao code construction*

For memories with k data bits, to construct a (n,k) Hsiao SEC-DED, we must define the parity matrix H. This matrix is

978-1-4577-2209-7/11/$26.00 ©2011 IEEE

of the form H =[C, I_r], where r is the number of parity bits, I_r the identity matrix and C is a k×r binary matrix. The k data bits are encoded by H in r parity bits. The data and the parity bits form an n bit codeword i.e. n=k+r. For SEC-DED, the length of parity can be determined by the rule $2^{r-1} \geq k+r$. A Hamming distance equal to 4 is needed for double error detection.

In [2], to provide SEC-DED capability an *H*-matrix must satisfy the following three constraints:

1. There are no all-0 columns.
2. Every column is distinct.
3. Every column contains an odd number of 1's.

To check if there are error bits in the data read from memory, the syndrome S is calculated from the data signal d and the r-by-n parity matrix H. The non-zero S implies the location of an error. The C-matrix columns are used to find the error location by comparing it to the non-zero syndrome vector. Fig.1 illustrates an (13,8) Hsiao code H-matrix.

$$H = \begin{matrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{matrix}$$

Fig. 1. (13,8) Hsiao code H-matrix.

The minimum the number of 1's in the H-matrix by using minimum odd weight columns make less hardware area in the ECC SEC-DED circuit since each 1 represent an XOR unit.

Also, using tow more constraints Hsiao code hardware and delay can be minimized [3]:

1. The total number of 1's in each column should be a minimum.
2. The number of 1's in each row should be made as balanced as possible.

To reduce power with little to no impact on area and delay there are two degrees of freedom in selecting the *H*-matrix [4]. The first degree of freedom is simply permuting the columns. This has no impact on area or delay as it does not change either the total number of 1's in the *H*-matrix or the balancing of 1's among the rows. The second degree of freedom is in selecting the odd-weight-columns that are included in the matrix. To minimize area and delay, the smallest odd weight columns should be used first.

To create an efficient implementation of the SEC-DED Hsiao code, we will modify the construction of the Parity Checker, the Overall Parity Bit Generator and the Syndrome Generator blocs in fig. 2.

In fact, these components are using XOR trees as shown in Fig. 3.

In the write mode, parity bits (said also check bits) and overall parity bit are generated with XOR trees corresponding, respectively to Parity Checker and Overall Parity Bit Generator units, and they are added to the data bits to be stored in the memory array.

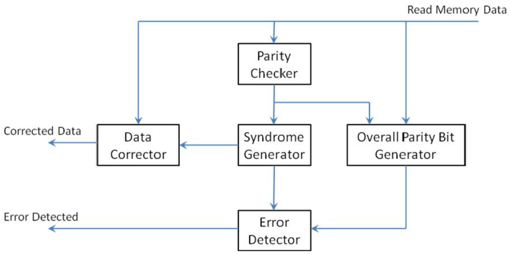

Fig. 2. SEC DED in Read mode.

In the read mode (fig. 2), the stored codeword is retrieved from memory array. Using the same XOR trees new check bits and overall parity bit are created from stored data bits and check bits. In the syndrome generator, stored and new check bits are compared. The conditions for error detection and correction are not discussed in this paper.

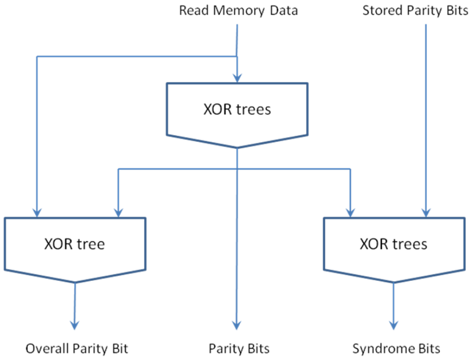

Fig. 3. Syndrome and parity bits generation using differential XOR trees.

The number of the used XOR trees is increasing with the data memory bits count and the error correction detection capability, what is influent on overhead, timing and parametric characteristics. While the Hsiao in this case of study, is good in CMOS technology, there are some needs for incoming nanotechnology memory features. The trade-offs are the reduction of power consumption and minimization of area overhead with a good fault tolerant capability. We will focus in this paper on the design of the ECC modules by using differential XOR build using Complementary Pass Transistor Logic.

B. Self-checking background

1) Self-checking (SC) circuits: Self-checking circuits are increasingly becoming a suitable approach to the design of complex VLSI ICs, to cope with the growing difficulty of on-line and off-line testing [5]. Self-checking circuits are class of circuits in which occurrence of fault can be determined by observation of the outputs of the circuits. An important subclass of these self-checking circuits is known as totally self-checking (*TSC*) circuits.

2) Totally self-checking (TSC) circuits: TSC circuits are used to detect errors concurrently with normal operation. These circuits operate on encoded inputs to produce encoded outputs. TSC checkers are used to monitor the outputs to indicate error when a non-code word is detected [6]. The concept of TSC circuits was first proposed in [7], and then generalized in [8], as follows:

Definition 1: A circuit is *fault-secure* for a set of faults, F, if for any valid input code word; any single fault, the circuit either produces an invalid code word on the output, or doesn't produce the error on the output.

Definition 2: A circuit is *self-testing* for a set of faults F, if for every fault in F, the circuit produces a non-code output for at least one code input (i.e. any single fault is detectable by some valid input code word).

Definition 3: A circuit is *totally self-checking* if it is fault-secure and self-testing.

Definition 4: A circuit is *code disjoint* if it always maps code word inputs into code word outputs and non-code word inputs into non-code word outputs.

Definition 5: A circuit is a *totally self-checking checker* if it is self-testing and code-disjoint.

III. Complementary pass transistor logic

The fundamental units in an ECC syndrome generator are the Exclusive-OR (XOR) and the Exclusive-NOR (XNOR). To meet the nanotechnology requirements new implementations use CPL (Complemntary Pass transistor Logic) technology for its high density, high speed and low power characteristics.

A 10 transistor XOR-XNOR [9] is shown in fig. 4. It uses two transmission gates and three inverters to give good output signals. It has no complementary inputs. It is power consuming design and the main drawbacks are the slow delays and the big number of transistors.

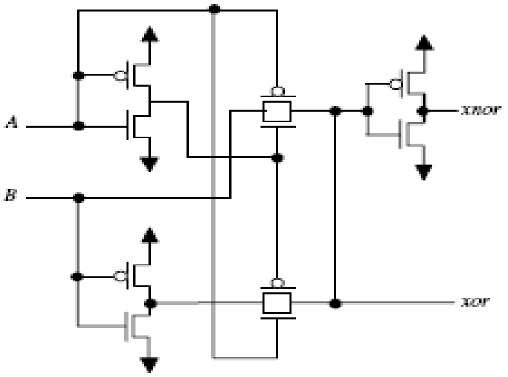

Fig. 4. 10 transistor circuit for XOR/XNOR.[9]

An XOR-XNOR architecture shown in fig. 5 is presented in [10]. It is constituted with 8 transistors with a combination of a feedback loop giving noise tolerant and reduced power consumption circuit. It has also non-complementary inputs.

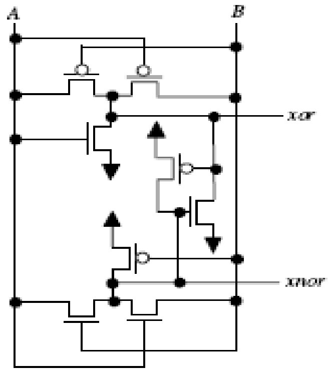

Fig. 5. An 8 transistors XOR/XNOR circuit.[10]

A combination of a power less (P-) XOR and Groundless (G-) XOR using single and double feedbacks is shown in fig. 6. They are constituted respectively of 10 and 12 transistors. They have two complementary inputs. They are also proved to be noise immune structure and have a good output levels [11] [12]. It can operate at low power voltages. The power consumption is minimized.

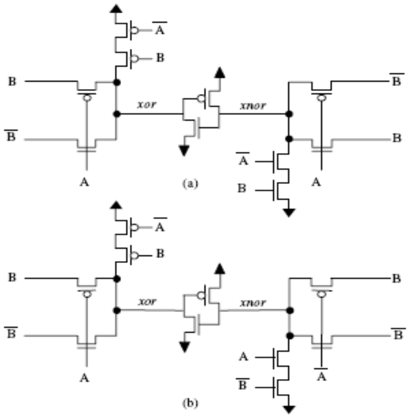

Fig. 6. P-/G- XOR/XNOR circuit using a single feedback [11] [12]

The design of Fig. 6 improves the speed of operation but the implementation of fig. 7 enhances the power-consumption statistics [11].

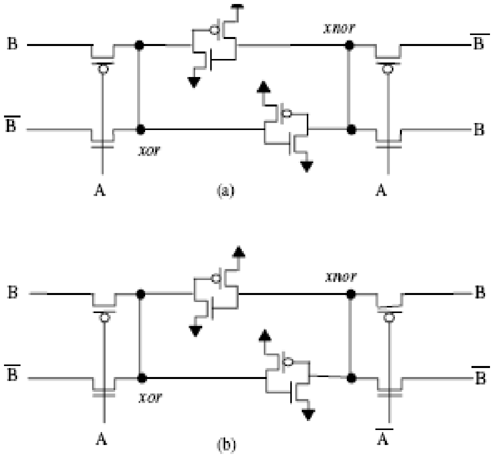

Fig. 7. P-/G- XOR/XNOR circuit using a dual feedback [11] [12]

In [13] Shiv et al has described two XOR-XNOR circuits.

They are constituted with an 8 transistors (fig. 8-a) and 10 transistors (fig. 8-b). These designs are supposed to be faster and suitable for arithmetic circuits with less power dissipation. They present a good output level. However, they give non full swing operations for some input patterns causing outputs to be degraded.

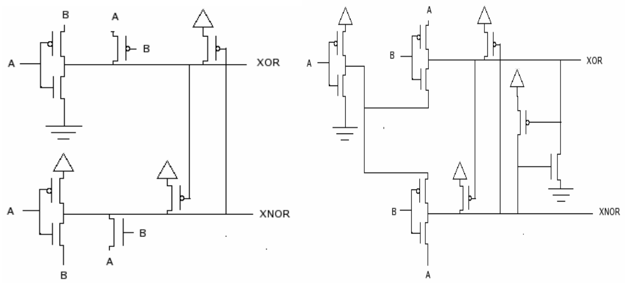

Fig. 8. (a) 8 transistors XOR/XNOR (b) 10 transistor XOR/XNOR [13].

A 6 transistors XOR-XNOR structure is discussed in [14] (fig. 9). It gives a good output signals with faster calculation delay. Also, it is intended to be energy efficient.

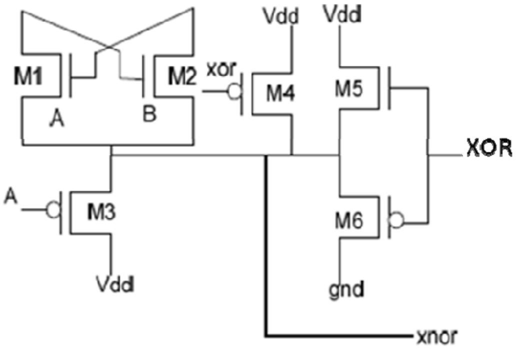

Fig. 9. 6 transistors XOR-XNOR implementation.[14]

Although the improvement in these XOR-XNOR structures, the number of transistor, the power consumption and delay are high. Also, they are giving a bad output signal swing for certain input patterns. While they are proposed to be noise tolerant, they are not reliable in very low supply voltage due to the threshold voltage loss problem commonly encountered in pass transistor logic design [14]. Also they have a drawback of loss of driving capability that is the main disadvantage of the CPL technology. To cope with this problem, we must then add inverters at the outputs of the XOR-XNOR components in the case of cascaded architecture.

In this paper, we have selected a new designed four transistors self-checking XOR-XNOR [15] (fig. 10) to meet our objectives in nanotechnology scale. It is constituted with a little area overhead inducing low power consumption and faster delays.

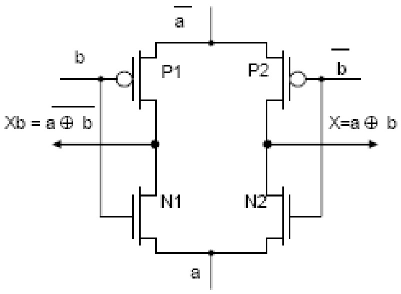

Fig. 10. Differential XOR [15]

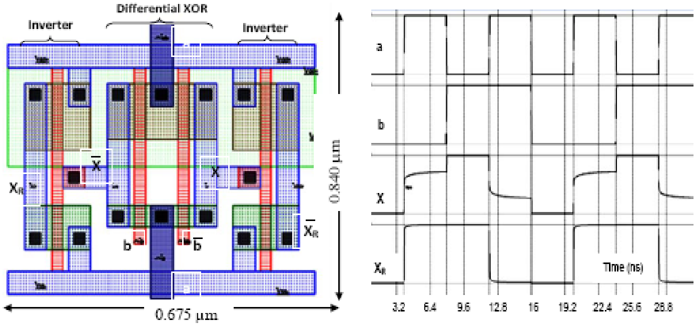

Fig. 11. Differential XOR Layout and Simulation [15]

This gate is *fault secure* and *self-testing* for all stuck-at, stuck-on and stuck-open faults. We made the proof in [15] that the proposed design is TSC for the entire set of faults.

Then, the layout of the proposed design had been implemented and simulated using 32nm CMOS technology. The proposed differential XOR can operate at low voltages, yet giving quite a good speed.

Actually, not only the memory matrix is affected by soft/transient errors but also the Error correcting code logics. To enhance the reliability of the ECC, we will give to it a self-checking capability using the four transistor self-checking differential XOR.

IV. PROPOSED SELF-CHECKING SEC-DED ARCHITECTURE

For our new architecture we modify the Parity Checker and the Syndrome Generator (fig. 12). For every memory data input (A_i) we must add an inverter to the Parity Checker to create the complementary input. The parity checker produce the Check parity bit C and its complement C' simultaneously using the differential XOR tree. We create also the complementary value of the stored check bit (Cs) using an inverter for each syndrome bit generation. The construction of the Overall parity bit generator bloc is the same of the Parity Checker. The Syndrome Generator use only one differential XOR for each syndrome bit.

The differential XOR is proved in [15] to be Totally Self-Checking (TSC). Therefore, using it in the implementation of the syndrome generator, of the parity Checker and of the Overall Parity bit Generator makes the ECC SEC-DED also Totally Self-Checking.

Hence a fault occurring in the stage 1 of the parity checker XOR tree, for example, will be detected by the next differential XOR of stage 2 and So on. The faults in memory data are not detected by the differential XOR.

To improve the driving capability of the ECC circuit, we can add inverters at the outputs of the differential XORs if necessary.

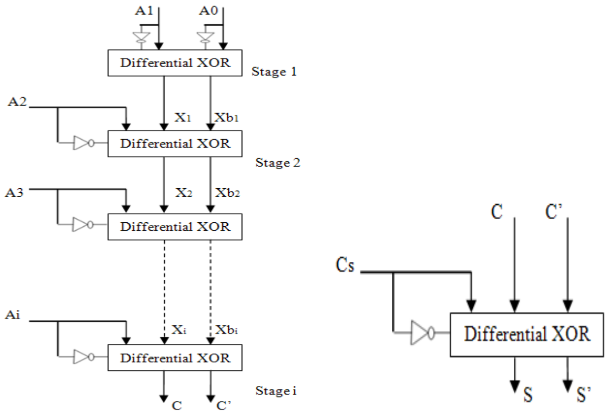

Fig. 12. Parity Checker (a) and Syndrome generator (b) using differential XOR.

To evaluate the number of transistors, we have implemented the parity checker, the Overall Parity bit generator and the syndrome generator for different CPL differential XORs.

By using the Differential XOR in [15] the SEC-DED structure use less to 50% of transistors than in the designs [11, 12] (Table I).

TABLE I
IMPLEMENTATIONS OF HSIAO SEC-DED WITH DIFFERENTIAL XOR

Data bits	Parity bits	XOR1[11] (#Transistor)	XOR2[12] (#Transistor)	XOR3[15] (#Transistor)
8	5	434	506	218
16	6	830	968	416
32	7	1610	1878	806
64	8	3350	3908	1676

V. CONCLUSION

We have realized a self-checking Hsiao SEC-DED error correcting using CPL differential XOR. The new architecture is more efficient as it has a little area overhead, a reduced consumption and fast delays. Nevertheless, some parameters must be well optimized to create a robust structure with better performances and good driving capability. The design is TSC for multiple types of faults by using the new four transistor self-checking differential XOR. We will focus, in the future, on integrating this new ECC circuit in large scale 3D memories.

REFERENCES

[1] C.L.Chen, M. Y. Hsiao, “Error-Correcting Codes for Semiconductor Memory Applications,” IBM J. Res Develop, vol. 28, no. 2, March 1984.

[2] M. Y. Hsiao, “A class of optimal minimum odd-weight-column sec-ded codes,” IBM J. Res Develop, vol. 14, no 4, July 1970.

[3] C.Y. Chen and C.W.Wu, “ An Adaptive Code Rate EDAC Scheme for Random Access Memory,” DATE10.

[4] S. Ghosh, S.Basu, and N. A. Touba, “ Selecting Error Correcting Codes to Minimize Power in Memory Checker Circuits,” J. Low Power Electronics, vol.1, no.1, 2005

[5] M.Nicolaidis, “On-line testing for VLSI: state of the art and trends, Integration,” the VLSI Journal, vol. 26, issues 1-2, 1 December 1998, pp. 197-209.

[6] D. K. Pradhan, J. J. Stiffler, “Error correcting codes and self-checking circuits in fault-tolerant computers,” IEEE Computer Mag. Vol. 13, March 1980, pp. 27-37.

[7] A. P. Chandrakasan, S. Sheng and R. W. Brodersen, “Low-Power CMOS Digital Design,” IEEE Journal of Solid State Circuits, vol. 27, vo. 4,April 1992, pp. 473–484.

[8] D. A. Anderson and G. Metze, “Design of totally self-checking check circuits for m-out-of-n codes,” IEEE Trans. on Computers, vol. 22, no. 3, March 1973, pp. 263-269.

[9] A.M. Shams, T.K. Darwish and M.A. Bayoumi, “Performance analysis of low-power 1-bit CMOS full adder cells,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 10, no. 1, 2002, pp 20-29.

[10] M.A. Elgamel, S. Goel and M.A. Bayoumi, “Noise Tolerant Low Voltage XOR-XNOR for Fast Arithmetic,” in Proceedings of the Great Lake Symposium on VLSI 2003, April 14-16, Washington, D.C.

[11] S.Goel, M.A. Elgamel and M.A. Bayoumi, “Novel design methodology for high-performance XOR.XNOR circuit design,” *Proc. 16th Symp. Integr. Circuits Syst. Design*, Brazil, Sep. 8-11 2003, pp. 71-76.

[12] S.Goel, M.A Elgamel and MA Bayoumi, “Design Methodologies for High-Performance Noise-Tolerant XOR-XNOR Circuits,” *IEEE Trans. Circuits Syst.I.*, vol. 53, no. 4, 2006.

[13] S. W. Shiv Shankar Mishra, R. K. Nagaria, and S.Tiwari, "New Design Methodologies for High Speed Low Power XOR-XNOR Circuits," *World Academy of Science, Engineering and Technology, 2009.*

[14] N.Ahmad, R.Hasan, “A New Design of XOR-XNOR gates for low power Application,” International Conference on Electronic Devices, Systems & Applications (ICEDSA2011).

[15] B.Hamdi, C.Khedhiri, A.Fradi, R. Tourki, “Four Transistors Self-checking Differential XOR,”10th International Symposium on Signals, Circuits and Systems (ISSCS), 2011.