VLSI Architecture Development of A Memory Module using SEC-DED Codes and TDES

Shehina S P
PG Scholar, Dept. of ECE
Muslim Association College of Engineering, Trivandrum, India

Mrs. Sajina S
Assistant professor , Dept. of ECE
Muslim Association College of Engineering, Trivandrum, India

***Abstract*—Hackers are everywhere in the world. Preventing interception is impossible; instead data can be transformed into an unreadable form during transmission and storage. Stuck at defects is the fault where the memory cell permanently store the same value regardless what is supposed to be saved. Continuous down scaling of CMOS technology lead to device shrinking. As a result soft error tolerance of VLSI circuit reduces. Different techniques have been used to deal with defects and soft errors. Repair techniques are commonly used for defects, while error correction codes are used for soft errors. Recently, some proposals have been made to use error correction codes to deal with defects. So secure and error protected memory is required to save highly confidential, sensitive and safety critical data. It can be achieved by saving cipher in error protected memory. For that plain text is encrypted using TDES algorithm, and then saved in the memory. SEC-DED along with interleaving soft error correction algorithm is applied to memory for error correction. Hamming codes are used for error detection and correction of 64 bit cipher word. These encrypted data can be decrypted by user having the key when it is required.**

Index Terms —Defects, error correcting codes, soft errors, secure communication, triple data encryption standard.

I. INTRODUCTION

Secrecy is certainly important to the security or integrity of information storage and transmission. Preventing interception is impossible; instead, the data must be made unreadable (encrypted) during transmission and storage, with a way for the intended recipient to be able to transform the received information back to its readable form (decryption process). Encryption is a mechanism by which a message is transformed or stored so that only the sender and recipient can see. When a message is encrypted, that means that it is transformed into a form when the data is passed through some substitute technique, shifting technique, table references or mathematical operations. All those processes generate a different form of that data and that is not readable; the encrypted form often looks like random characters or gibberish. When a message is decrypted, it is returned to its original readable form. Encryption can provide strong security for data to give sensitive data the highest level of security. As a general term, cryptography is used in order to keep crucial or secret information from unauthorized access. Encryption, a cryptographic implementation, is the conversion of data into a seemingly incomprehensible mixture of characters that, when viewed, cannot be read as simple text. Simple text is defined as standard written text, such as this document. The algorithm used to encrypt data is called a cipher, or cipher text which is representation of the original data in a difference form, while unencrypted data is called plaintext. Decryption is the process of converting encrypted data (cipher text) back into its original form (plaintext), so it can be understood.

Device scaling trends dramatically increase the susceptibility of various devices to soft errors. Soft errors, which are also called transient faults or single-event upsets (SEUs). These are errors in processor execution that are due to electrical noise or external radiation rather than design or manufacturing defects. In particular, we study soft errors caused by high energy neutrons resulting from cosmic rays colliding with particles in the atmosphere. Hence soft error correcting circuits became a necessity to integrate with medical instrument, defense equipments, measuring instrument, aviation industry and nuclear plant etc for to meet both reliability and the safety concern. Technology scaling has made today`s designs much more susceptible to soft errors. In telecommunication, Hamming codes are a family of linear error correcting codes. Hamming codes can detect upto two and correct upto one bit errors. By contrast the simple parity code cannot correct errors, and can detect only odd number of errors. Hamming code is special in that they are perfect codes, which are they achieve the highest possible rate for codes with their block length and minimum distance. Because of the simplicity of hamming codes, they are widely used in computer memory.

A secure and reliable memory is necessary to save highly confidential, safety critical and sensitive data. Such a memory can be designed by combining cryptography along with a suitable error correcting algorithm. When we compare DES and Triple Des algorithms, TDES provide better security against brute force attack. 3DES which is secure enough for most purposes today. 3DES is a construction of applying DES three times in sequence. 3DES with three different keys (K1, K2 and K3) has

effective key length is 168 bits. The use of three distinct key is recommended of 3DES. By comparing with AES, RSA algorithms TDES is a simple algorithm. So it is suitable for memory applications. In the case of error correction a neat example of a block code is the Hamming code. This is an error detecting and error-correcting binary code, for example which transmits N=7 bits for every K=4 source bits. This kind of codes can detect and correct single bit errors or detects double bit errors.

II. RELATED WORK

The use of redundant rows and columns has been widely used in memory design to cope with this problem. One-dimensional. (1-D) redundancy is the simplest variation in which only redundant rows (or columns) are included in the memory array and used to replace the defective rows (or columns) detected during test. The main advantage of this approach is that its implementation does not require any complex allocation algorithms. Unfortunately, its repair efficiency can be low because a defective column (row) containing multiple defective cells cannot be replaced by a single redundant row (column).

In two-dimensional (2-D) redundancy approach which improves the efficiency of the 1-D approach. This approach adds both redundant rows and columns to the memory array to provide more efficient repair when multiple defective cells exist in the same row or column of the array. When multiple faulty cells are detected, the choice between the use of a redundant row or a redundant column to replace them is made based on the maximum repair capability of each alternative. The main drawback of this approach is that the optimal redundancy allocation becomes a problem. Although many heuristic algorithms have been proposed to solve this problem, it is still difficult to develop built-in repair implementations using them.

For both redundancy approaches, when the number of defective cells in the array exceeds the repair capability through the use of redundant elements, the last alternative before discarding the defective chip is to try to use it as a downgraded version of memory. For example, when all remaining defective cells are located in one half of the array, the other half can still be used as a memory with reduced capacity. This reduction is done by permanently setting the most significant bit of the addresses either to 0 or 1, depending on which part of the memory is to be used. However, in most cases, the remaining defective cells are evenly distributed across the whole array, and not clustered in one half of the array, making this technique useless.

DES encrypts data in 64 bit block size and uses effectively a 56 bit key. 56 bit key space amounts to approximately 72 quadrillion possibilities. Even though it seems large but according to today's computing power it is not sufficient and vulnerable to brute force attack. Therefore, DES could not keep up with advancement in technology and it is no longer appropriate for security.

III. PROPOSED TECHNIQUE

A secure and reliable memory is necessary to save highly confidential, safety critical and sensitive data. Such a memory can be designed by combining cryptography along with a suitable error correcting algorithm. TDES provide better security against brute force attack and which is secure enough for most purposes today. Data encryption (cryptography) is utilized in various applications and environments. The specific utilization of encryption and the implementation of the DES and TDEA will be based on many factors particular to the computer system and its associated components. In general, cryptography is used to protect data while it is being communicated between two points or while it is stored in a medium vulnerable to physical theft. Communication security provides protection to data by enciphering it at the transmitting point and deciphering it at the receiving point.

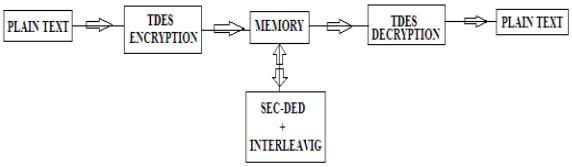

Fig. 1. Block Diagram of Proposed Memory Module.

DES is a block cipher that takes a plain text string as input and creates a cipher text string of the same length. It uses a symmetric key, which means that the same key is used to convert cipher text back into plain text. The DES block size is 64 bits. The key size is also 64 bits, although 8 bits of the key are used for parity (error detection), which makes the effective DES key size 56 bits. A 56-bit key length is now considered weak due to advances in computer processing power. DES is a block cipher, which means that during the encryption process, the plaintext is broken into fixed length blocks and each block is encrypted at the same time. Basically it takes a 64 bit input plain text and a key of 64-bits (only 56 bits are used for conversion purpose and rest bits are used for parity checking) and produces a 64 bit cipher text by encryption and which can be decrypted again to get the message using the same key. Additionally, we must highlight that there are four standardized modes of operation of DES: ECB (**E**lectronic **C**ode**b**ook mode), CBC (**C**ipher **B**lock **C**haining mode), CFB (**C**ipher **F**eed**b**ack mode) and OFB (**O**utput **F**eed**b**ack mode).

The general depiction of DES encryption algorithm which consists of initial permutation of the 64 bit plain text and then goes through 16 rounds, where each round consists permutation and substitution of the text bit and the inputted key bit, and at last goes through a inverse initial permutation to get the 64 bit cipher text. The key-dependent computation can be simply defined in terms of a function "f", called the cipher function, and a function

"KS", called the key schedule. The cipher function "f" is defined in terms of primitive functions which are called the selection functions "Si" and the permutation function "P".

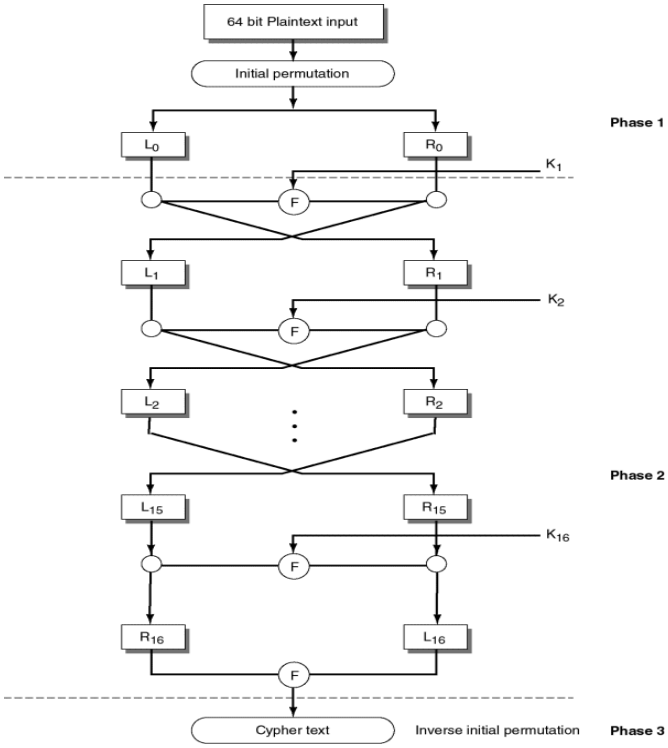

Fig. 2. Enciphering Computation.

Double DES is not used due to a meet-in-the-middle attack, which makes the effective key size 57 bits (it is essentially twice as hard to crack as DES, not exponentially harder). TDES uses three rounds of DES encryption and has a key length of 168 bits (56 * 3). Brute force attacks against TDES are currently not practical. TDES is a construction of applying DES three times in sequence. TDES with three different keys (K1, K2 and K3) has effective key length is 168 bits. The use of three distinct key is recommended of TDES. By comparing with AES, RSA algorithms TDES is a simple algorithm. So it is suitable for memory applications. In the case of error correction a neat example of a block code is the Hamming code. This is an error detecting and error-correcting binary code, for example which transmits N=7 bits for every K=4 source bits. This kind of codes can detect and correct single bit errors or detects double bit errors.

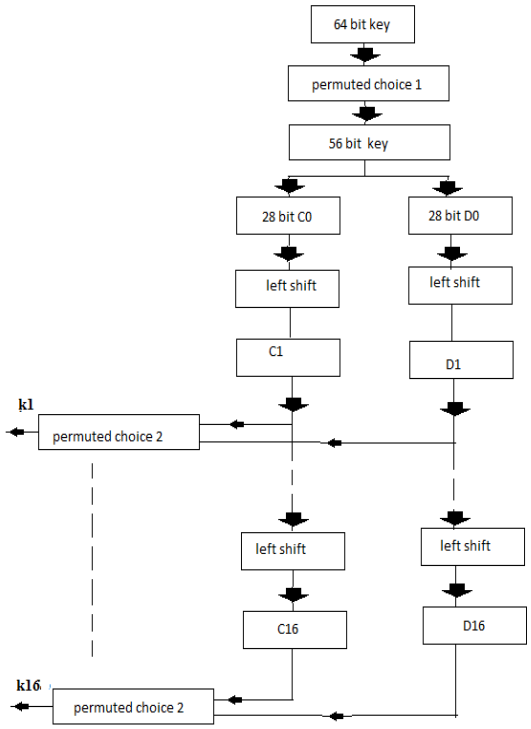

Fig. 3. Flowchart For Key Schedule Calculation.

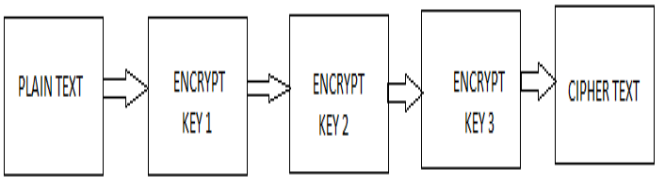

Fig. 4. Block Diagram of TDES.

The main problem when using SEC-DED correction to deal with manufacturing defects is that words in which a cell has a defect are left unprotected, as a single soft error will cause a failure. This outcome can significantly reduce memory reliability. However, for defects that manifest as isolated stuck-at failures such that a cell that is read always gives the same value, an alternative correction scheme can be used to improve reliability. The proposed technique is as follows. When a word is read and an error is detected, if it is classified as a single error, then it is corrected as in a normal SEC-DED memory. However, if an uncorrectable error is detected, a procedure is triggered to detect if the word contains defects. This procedure stores the contents of the word in a register, and then writes all-zeros into the word and reads it back to check that there are no errors. The same operation is then done for the all-ones pattern. If there is a stuck-at defect on that word, the procedure will detect the defect and locate it. If there is no defect, a failure

is triggered as the error is in fact uncorrectable. However, if there is a defect, the corresponding bit in the register is inverted, and the modified word is decoded again. This technique can effectively correct words that contain either a soft error, or a stuck-at defect, or both simultaneously.

When two bits are affected by errors: one by a soft error, and another by a defect. When the word is read, two errors are detected, thus uncorrectable. Following the proposed technique, a defect will be detected, and that defective bit will be inverted. In this case, the word will be correctly decoded as there is only a single error (the soft error). Therefore, when the proposed technique is used, single bit errors will not cause failures, even if they affect a word that contains a stuck-at defect. This approach would greatly increase the reliability when ECC is used to correct defects.

The following algorithm generates a single-error correcting (SEC) code 64 bits.

1. Number the bits starting from 1: bit 1, 2, 3, 4, 5, 6 etc.
2. Write the bit numbers in binary. 1, 10, 11, 100, 101, 110 etc.
3. All bit positions that are powers of two (have only one 1 bit in the binary form of their position) are parity bits.
4. All other bit positions, with two or more 1 bits in the binary form of their position, are data bits.
5. Each data bit is included in a unique set of 2 or more parity bits, as determined by the binary form of its bit position.

a). Parity bit 1 covers all bit positions which have the least significant bit set: bit 1 (the parity bit itself), 3, 5, 7, 9, etc.

b). Parity bit 2 covers all bit positions which have the second least significant bit set: bit 2 (the parity bit itself), 3, 6, 7, 10, 11, etc.

c). Parity bit 4 covers all bit positions which have the third least significant bit set: bits 4–7, 12–15, 20–23, etc.

d). Parity bit 8 covers all bit positions which have the fourth least significant bit set: bits 8–15, 24–31, 40–47, etc.

e). In general each parity bit covers all bits where the binary AND of the parity position and the bit position is non-zero.

C. Results and Discussions

Plain text is converted to cipher text using triple DES algorithm. Three different keys are applied to same text to increase the security. 192 bit key is applied to 64 bit data and it produces 64 bit cipher. Before it is stored in the memory, it will be converted to a signal having 71 bits. (64, 71) bit hamming code is used to convert 64 bit cipher into 71 bit signal. 7 parity bits will be added to this encrypted data. This 71 bit cipher will save in the memory. During read operation syndrome decoder will generate syndrome of the signal. XORing of corresponding bits will generate parity bits. Reverse operation will performed for syndrome checking. If syndrome is zero data stored in the memory is

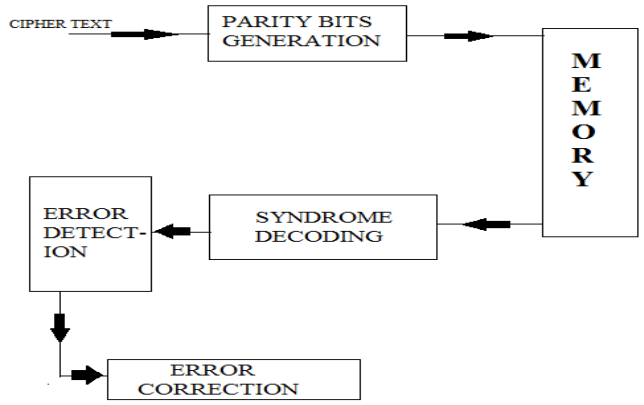

Fig. 5. Architectural Block Diagram Of Single Error Correction- Double Error Detection of 64 bit cipher.

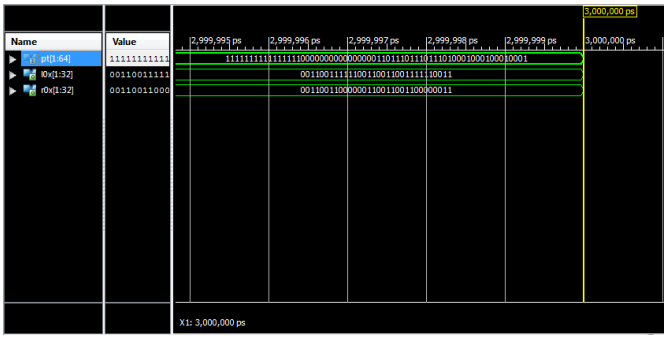

Fig. 6. Plain Text Is Converted To Cipher Text And Divided Into 2 Parts

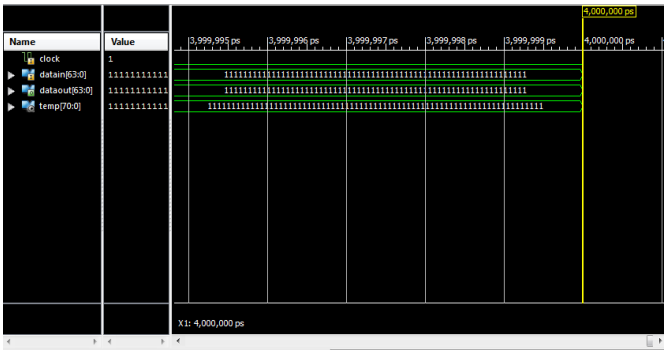

Fig. 7. Output Obtained When Signal Has Zero Error.

correct and error free. If there exist a non zero syndrome, then retrieved data is corrupted by error. This can be corrected by using single error correcting codes called hamming codes. End user can retrieve this data by using Tdes decryption.

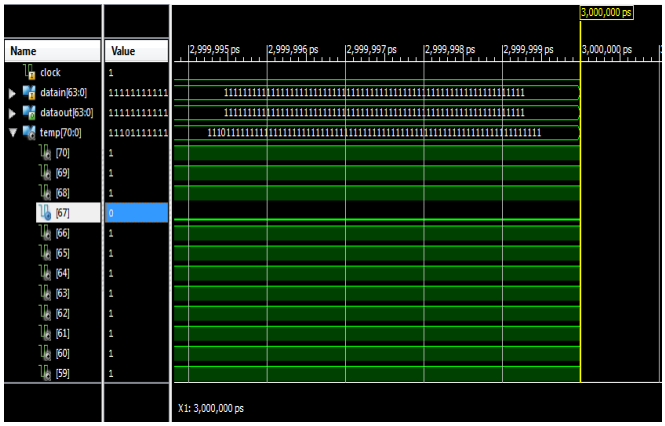

Fig. 8. Output Obtained When Signal Haveing A Single Bit Error.

IV. CONCLUSION

Secure and error free memory is required for some applications. Hackers are everywhere in the world. Stuck at defects is the fault where the memory cell permanently store the same value regardless what is supposed to be saved. Continuous down scaling of CMOS technology lead to device shrinking. As a result soft error tolerance of VLSI circuit reduces. In this context secure and reliable data storage will become an issue. But secure and reliable data storage required in some application, such as military purpose, business and banking sector, law and enforcement, space electronics and research etc. For example, suppose
a password to on the ignition system of a missile is saved in the memory. It is necessary to protect that data from eavesdroppers and it must be retrieved correctly. So a secure and reliable memory is required to store highly confidential and safety critical data. It can be achieved by saving cipher text in an error protected memory. For that plain text can be encrypted using TDES algorithm, then it can be stored in the memory. There SEC-DED algorithm is applied to correct soft errors. Here hamming codes are used for error correction and detection. This data can be retrieved by the user having the key, when is required. So such a memory can be developed by combining TDES algorithm and SEC-DED algorithm.

REFERENCES

[1] Costas Argyrides, *Member, IEEE*, Pedro Reviriego, *Member, IEEE*, and Juan Antonio Maestro, *Member, IEEE*, "Using single error correction codes to protect against isolated defects and soft errors," IEEE TRANSACTIONS ON RELIABILITY, VOL. 62, NO. 1, MARCH 2013
[2] D. Bhavsar, "An algorithm for row–column self-repair of RAMs and its implementation in the Alpha 21264," in *Proc. Int. Test Conf.*, 1999, pp. 311–318.
[3] Deepthy J R, Student Member, IEEE, Jismi K, and Anand P, "Luby Transform EncodedCommunication System With TDES Encryption for Erasure Channel,".
[4] Toby Schaffer, Alan Glaser, and Paul D. Franzon, "Chip-Package Co-Implementation of a Triple DES Processor", IEEE Transactions on Advanced Packaging, vol. 27, no. 1, Feb. 2004.
[5] Simon Haykin, Communication Systems, Fourth edition, 2001.
[6] Shah Kruti R & Bhavika Gambhava, " New Approach of Data Encryption Standard Algorithm", IJSCE, ISSN: 2231-2307, Volume-2, Issue-1, March 2012.
[7] Premkishore Shivakumar *et al.in*, "Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic", Appears in the Proceedings of the 2002 International Conference on Dependable Systems and Networks.
[8] Amandeep Singh & Manu Bansal, "FPGA Implementation of Optimized DES Encryption Algorithm on Spartan 3E", International Journal of Scientific & Engineering Research, ISSN 2229-5518, Volume 1, Issue 1, October-2013.
[9] Stephen M. Trimberger, Fellow IEEE, and Jason J. Moore, "FPGA Security : Motivations, Features, and Applications", Proceedings of the IEEE | Vol. 102, No. 8, August 2014.
[9] WilliamStallin, "Computer Organization And Architecture", 7th edition, chapter 5, 2006.
[10] Gregory Mitchell,"Investigation of Hamming, Reed-Solomon, and Turbo Forward Error Correcting Codes", Approved for public release; distribution unlimited,July 2009.

.