DESIGN AND IMPLEMENTATION OF DOUBLE ERROR CORRECTION CODES FOR PROTECTING DATA

[1]K. Bhagya sree, [2]A.Mallaiah

[1]M.Tech, [2]Associate Professor

[1]Dept of Electronics and Communication Engineering,

[1]Gudlavalleru Engineering College, Andhra Pradesh, India

Abstract: The information signal is first encrypted and then transmitted using some transmission media which may be corrupted in the process. These errors can be corrected and detected by employing error-detecting and error-correcting codes in RAMs. The most common error detection scheme is parity bit. An error correcting code uses multiple parity check bits that are stored with the data word in memory. It is essential to have SEC codes that protect both the data and the associated control bits. It is possible for these codes to provide fast decoding of the control bits, as these are used to determine the processing of the data and are commonly on the critical timing path. In this brief, a method to extend SEC codes to support a few additional control bits is presented. The derived codes support fast decoding of the additional control bits and are therefore suitable for networking applications.

Index Terms: Error correction codes, high-speed networking, memory, single error correction (SEC). -

I. INTRODUCTION

In transmitting information from one place to another digital machine uses codes that uses simply sets of symbols to which values and codes are attached. Data can be corrected during transmission. For reliable communication, errors must be corrected and detected. Error correction and detection are implemented either at data link or transport layer of OSI model. There are two types of binary codes they are linear block codes and convolution codes.

Instead of repeating the entire data stream, a shorter group of bits may be appended to the end of the each unit. This technique is called redundancy because the extra bit are redundant to the information. They are discarded once the accuracy of the transmission is determined .The aim of the extra digits, called redundant or parity digits, is to detect and hopefully correct any errors that occurred in transmission. Single error correction (SEC) codes are the ones most commonly used to protect standard memories and circuits.

The main reason for this is that SEC codes can be encoded or decoded with simple circuitry and require a low number of redundant bits. For memory protection, SEC codes are commonly extended to also detect double errors. A classical type of SEC codes is hamming codes that can be constructed in a simple way. The use of an ECC (Error Correction Code) impacts the circuit design in terms of both delay and area. Consistency is a key requirement for networking equipment such as core routers. To protect memories, error correction codes (ECCs) are widely used.

The delay is added as data has to be encoded when writing into the memory and decoded when reading from it. Single error correction (SEC) codes are the ones most commonly used to protect standard memories and circuits. Dependable memory systems can be designed either by using highly reliable but expensive components or by employing inexpensive protective redundancy in terms of a single error correcting code that uses redundant check bits. To correct a single-bit error, information in the form of check bits is required to address the bit in error. As check bits are equally prone to failure, they are required to address themselves also in case of error. A SEC code is generated by appending certain parity check bits to the data bits. These check bits are generated with the help of the parity check matrix of the SEC code. The Hamming weight of a vector , denoted by w(u), is the number of nonzero elements of u. Error correcting codes is a technique whereby more than the minimum number of binary digits are used to represent the messages. The aim of the extra digits, called redundant or parity digits is to detect and hopefully correct any errors that occurred in transmission.

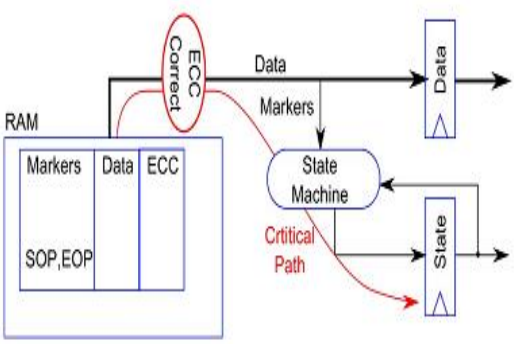


Fig.1 Typical packet data storage in a networking

2. BACKGROUND CODING THEORY:

Background coding theory more detailed accounts of error-correcting codes can be found in Hill, Pless, Mac Williams and Sloane, van Lint, and Assumes and Key. See also Peterson for an early article written from the engineers' point of view. Proofs of all the results quoted here can be found in any of these texts; our summary here follows. The usual pictorial representation of the use of error-correcting codes to send messages over noisy channels is shown in the schematic diagram.

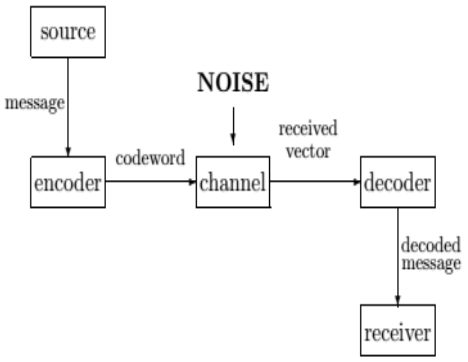


Fig 2: A noisy communication channel

Here a message is first given by the source to the encoder that turns the message into a code word, i.e. a string of letters from some alphabet, chosen according to the code used.

2.1 Hamming codes

The most common types of error-correcting codes used in RAM are based on the codes devised by R. W. Hamming. In Hamming code, k parity bits are added to an n-bit data word, forming a new word of n + k bits. The bit positions are numbered in sequence from 1 to n-k. Those positions numbered with powers of two are reserved for the parity bits. The remaining bits are the data bits. The code can be used with words of any length. Before giving the general characteristics of the Hamming code, let us demonstrate the operation with a data word of eight bits. For example, 8-bit data word 11000100. We include four parity bits with this word and arrange the 12 bits as follows.

Bit position	1	2	3	4	5	6	7	8	9	10	11	12
	P1	P2	1	P4	1	0	0	P8	0	1	0	0

The 4 parity bits P1 through P8 are in positions 1, 2, 4 and 8, respectively. The 8 bits of the data word are in the remaining positions. Each parity bit is calculated as

P1=XOR of bits (3, 5, 7, 9, 11) =0
P2=XOR of bits (3, 5, 7, 10, 11) =0
P4=XOR of bits (5, 6, 7, 12) =1
P8=XOR of bits (9, 10, 11, 12) =1

Exclusive-OR operation performs the odd function. It is equal to 1 for an odd number of 1's among the variables and to 0 for an even number of 1's. Thus, each parity bit is set so that the total number of 1's in the checked positions, including the parity bit, is always even.

2.2 Data Protection in Networking Applications

Modern data rates that range from 10 to 400 Gbit/s, and terabit rates are expected in future. To find high data rates, on-chip packet data buses are wide, with typical widths between 64 and 2048 bits.

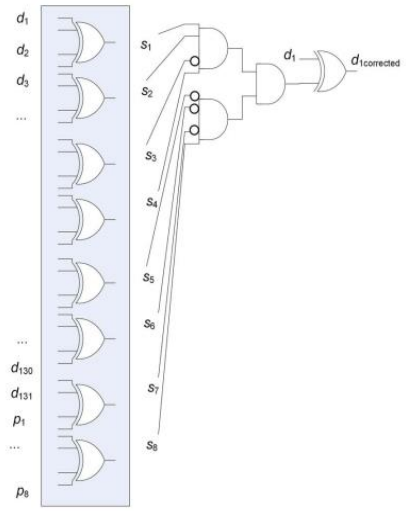


Fig 3 (a): Syndrome Computation

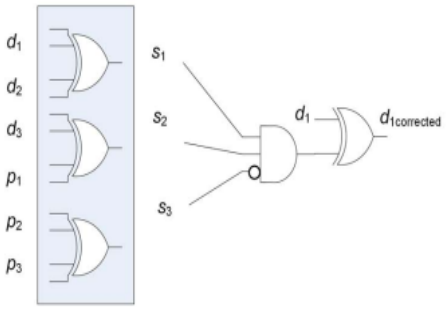


Fig.3(b): Syndrome Computation

Fig.3. Decoding of a control bit for single and independent SEC codes for data and control. (a) SEC code for both data and control bits. (b) Independent SEC codes for data and control bits

Data packets must normally be stored in RAMs that means in FIFOs for adapting processing rates. When storing packet data, it is necessary to delineate the packet boundaries. In the entire simplest case, each segment on the bus can be defined with a single EOP marker. The next valid segment is then assumed to be the start of the following packet. Where a packet is in error and it must be dropped. To spot such disintegrate packets, an additional control signal (ERR) may be required.

3. PROPOSED SYSTEM

As mentioned in the introduction, our goal is to design SEC codes that can protect a data block plus a few control bits such that the control bits can be decoded with low delay. As discussed, the data blocks to be protected have a size that is generally a power of two, e.g., 64 or 128 bits. To protect a 64-bit data block with a SEC code, 7 parity check bits are needed, while 8 bits are enough to protect 128 bits. In the first case, there are 2^7=128 probable syndromes, and therefore, the SEC code can be prolonged to cover a few additional control bits. The same is true for 128 bits and, for a SEC code that protects a data block that is a power of two. This implies that the control bits can also be protected with no additional parity check bits. This is more efficient than using two separate SEC codes (one for the data bits and the other for the control bits) as this involves additional parity check bits.

Major difficulty in using an extended SEC code is that the decoding of the control bits is more multifarious. To explain this issue, let us consider a 128-bit data block and 3 control bits. The primary SEC code for the 128-bit data block has the parity check matrix. Decoding of a bit in each case is shown in Fig. 4, and the difference in complexity is obvious. As discussed before, our goal is to simplify the decoding of the control bits while using a single SEC code for both data and control bits. For that, the first step is to note that, in some cases, SEC decoding can be streamlined to check only some of the syndrome bits.

One illustration is the decoding of constant-weight SEC codes proposed. Now in this case, only the syndrome bits that have a 1 in the column of the parity check matrix need to be tested

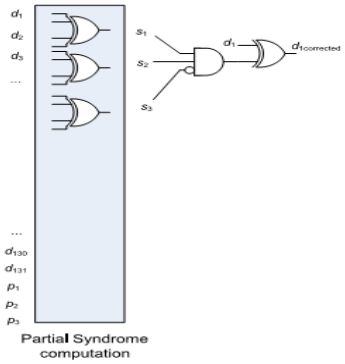


Fig 4: Bit decoding of a control bit in the Proposed SEC code

Table 1: Minimum number of *pcd* bits for 128 and 256 data bits

Control bits	128 data bits	256 data bits
3	3	3
4	4	4
5	4	4
6	4	4
7	4	4
8	5	5

This shortens the decoding for all bits but, in many cases, requires additional parity check bits. In this case, the major focus is to simplify the decoding of the control bits as those are commonly on the critical path. For that, the parity check bits can be separated in two groups: a first group that is shared by both data and control bits and another that is used only for the data bits. Therefore the decoding of the control bits only requires the recomputation of the first group of parity check bits. Now let us

consider a 128-bit data block and 3 control bits protected with 8 parity check bits. These 8 bits are separated in a group of 3 shared between data and control bits and a second group of 5 that is used only for the data bits. To guard the control bits, the first three parity check bits can be allotted different values for each control bit, and the enduring parity check bits are not used to protect the control bits.

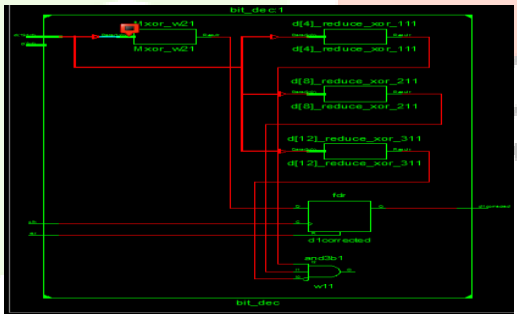
The remaining values are used to protect the data bits, and for each value, different values of the remaining five parity check bits can be used. Here in this case, the first group has 3 bits that can take 8 values, and three of them are used for the columns that relate to the control bits. This leaves 5 values that is used to protect the data bits. The another group of parity check bits has 5 bits that can be used to code 32 values for each of the 5 values on the first group. Thus extreme of 5×32 = 160data bits can be protected. In detail, the number is lower as the zero value on the first group cannot be combined with a zero or a single one on the second group as the corresponding column would have weight of zero or one. In any case, 128 data bits can be simply protected. Example of the parity check matrix of a SEC code resulting using this method. The three first columns correspond to the added control bits. The two groups of parity check bits are also separated, and the first three rows are shared for data and control bits, whereas the last five only protect the data bits. It is observed that the control bits can be decoded by merely re computing the first three parity check bits. In adding, the zero value on these three bits is also used for some data bits. This means that those bits are not needed to re compute the first three parity check bits. The decoding of one of the control bits is shown in Fig. 3.It can be observed that the circuitry is considerably simpler than that of a traditional SEC code.

This will be confirmed by the trial results presented. The process can also be used to protect more than three control bits. In a total, let us consider that we need to protect d data bits and c control bits using p parity check bits. Then, p is split in two groups p_{cd} and p_d. The first group is shared between control and data bits, and the second is used only for the data bits. The number of data bits that can be protected with this system can be considered as. The number of combinations of the first group available to be used to protect the data bits is$2^{Pcd}-c$. For each of those, up to 2^{Pd} values can be used, giving total of $(2^{Pcd}-c)\cdot 2^{Pd}$. But, for the zero value, the combinations of the second group with weight zero or one cannot be used, so p_d+1 should be subtracted. Correspondingly, for the p_{cd} values with weight one on the first group, the zero value on the second group cannot be used as the subsequent column would have weight one. Therefore, pcd should also be subtracted, giving a total of $(2^{Pcd}-c)\cdot 2^{Pd}-(p_d+1)-p_{cd}$. This is the number of data bits that can be protected in addition to the control bits. Equally as the number of control bits increases, pcd must also be increased to be able to protect the block of data bits with the same number of parity check bits. This is shown in Table I for 128 and 256 data bits. Increasing p_{cd} makes the decoding of control bits more complex. Thus, the minimum value should be used. And extension of the single error correction is we are implementing the double error correction. So that the consistency of the system increases.

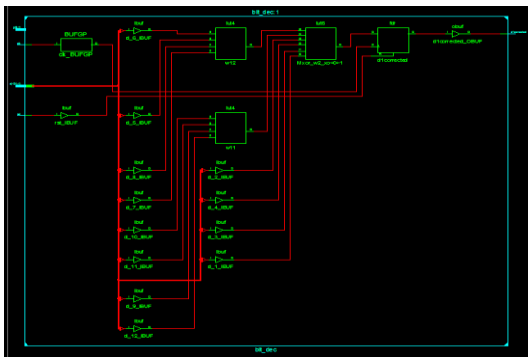
3.1 Synthesis results

The developed project is simulated and verified their functionality. Once the functional verification is done, the RTL model is taken to the synthesis process using the Xilinx ISE tool. In synthesis process, the RTL model will be converted to the gate level netlist mapped to a specific technology library.

Rtl schematic:



Technology schematic:

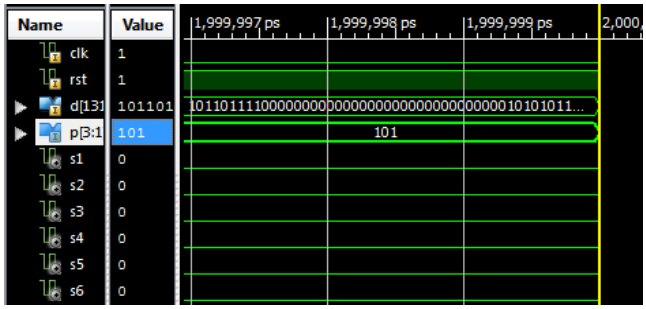


Design summary:

Device utilization summary (estimated value)			
Logic utilization	Used	Available	utilization
No of LUTS	3	21000	0%
No of IOBs	15	210	7%

No of fully used FF pairs	0	3	0%

Simulation:



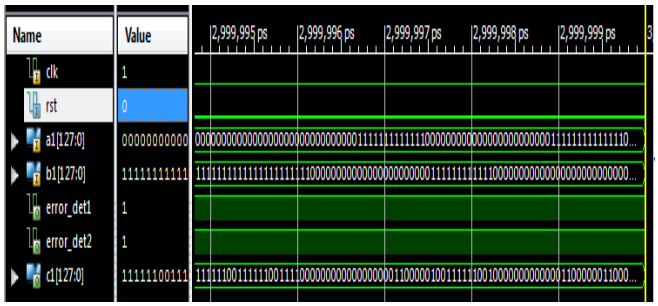


Table II: Comparison of proposed work with existing work

parameters	Existing 128-bit sec for data bits	Existing 128-bit sec for control bits	Proposed 128-bit single error	Proposed 128-bit double error
delay	4.760 ns	3.616 ns	3.556 ns	2.292ns
Registers	1	1	1	1
Xors	9	4	9	4

4. CONCLUSION

In this basically, a technique to construct SEC codes that can guard a block of data and some additional control bits has been offered. The derived codes are designed to enable fast decoding of the control bits. The derived codes have the same number of parity check bits as existing SEC codes and thus they do not require additional cost in terms of memory or registers. To estimate the benefits of the proposed scheme, several codes have been implemented and compared with minimum-weight SEC codes. The proposed codes are useful in applications, where a few control bits are added to each data block and the control bits have to be decoded with low delay. Example is arithmetic circuits where the critical path is commonly on the least significant bits. Therefore, reducing the delay on those bits can increase the overall circuit speed.

REFERENCES:

[1] P. Bosshart et al., "Forwarding metamorphosis: Fast programmable match-action processing in hardware for SDN," inProc. SIGCOMM,
2013, pp. 99–110.
[2] J. W. Lockwood et al., "NetFPGA—An open platform for gigabit-ratenetwork switching and routing," inProc. IEEE Int. Conf. Microelectron.Syst. Educ., Jun. 2007, pp. 160–161.
[3] A. L. Silburt, A. Evans, I. Perryman, S.-J. Wen, and D. Alexandrescu,"Design for soft error resiliency in Internet core routers,"IEEE Trans.Nucl. Sci., vol. 56, no. 6, pp. 3551–3555, Dec. 2009.
[4] E. Fujiwara,Code Design for Dependable Systems: Theory and PracticalApplication. Hoboken, NJ, USA: Wiley, 2006.
[5] C. L. Chen and M. Y. Hsiao, "Error-correcting codes for semiconductormemory applications: A state-of-the-art review,"IBM J. Res. Develop.,vol. 28, no. 2, pp. 124–134, Mar. 1984.
[6] V. Gherman, S. Evain, N. Seymour, and Y. Bonhomme, "Generalizedparity-check matrices for SEC-DED codes with fixed parity," inProc.IEEE On-Line Test. Symp., 2011, pp. 198–20.
[7] P. Zabinski, B. Gilbert, and E. Daniel, "Coming challenges with terabitper-second data communication," IEEE Circuits Syst. Mag., vol. 13, no. 3, pp. 10–20, 3rd Quart. 2013.
[8] UltraScale Architecture Integrated Block for 100 G Ethernet v.14. LigCOREIP Product Guide. PG165, Xilinx, San Jose, CA, USA. Jan. 22, 2015.
[9] OpenSilicon Interlaken ASIC IP Core. [Online]. Available: www.opensilicon.com/open-silicon-ips/interlaken-controller-ip/
[10] P. Reviriego, S. Pontarelli, J. A. Maestro, and M. Ottavi, "A method to construct low delay single error correction (SEC) codes for protecting data bits only," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 32, no. 3, pp. 479–483, Mar. 2013.
[11] V. Gherman, S. Evain, N. Seymour, and Y. Bonhomme, "Generalized parity-check matrices for SEC-DED codes with fixed parity," in Proc. IEEE On-Line Test. Symp., 2011, pp. 198–20

[12] J. E. Stine et al. "FreePDK: An open-source variation-aware design kit," in Proc. IEEE Int. Conf. Microelectron. Syst. Educ., Jun. 2007, pp. 173–174.
[13] M. Y. Hsiao "A class of optimal minimum odd-weight column SECDED codes," *IBM J. Res. Develop.*, vol. 14, pp. 395–301, Jul. 1970.
[14] R. W. Hamming, "Error detecting and error correcting codes," *Bell Syst. Tech. J.*, vol. 29, pp. 147–160, Apr. 1950.
[15] V. Gherman, S. Evain, N. Seymour, and Y. Bonhomme, "Generalized
[16] parity-check matrices for SEC-DED codes with fixed parity," in *Proc. IEEE On-Line Testing Symp.*, Jul. 2011, pp. 198–20.
[17] M. Richter, K. Oberlaender, and M. Goessel, "New linear SEC-DED
[18] codes with reduced triple bit error miscorrection probability," in *Proc. IEEE On-Line Testing Symp.*, Jul. 2008, pp. 37–42.
[19] A. M. Saleh, J. J. Serrano, and J. H. Patel, "Reliability of scrubbing
[20] recovery-techniques for memory systems," *IEEE Trans. Reliab.*, vol. 39, no. 1, pp. 114–122, Apr. 1990.
[21] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2004.
[22] M. Y. Hsiao, D. C. Bossen, and R. T. Chien, "Orthogonal latin square
[23] codes," *IBM J. Res. Develop.*, vol. 14, no. 4, pp. 390–394, Jul. 1970.
[24] sG. Li, I. J. Fair, and W. A. Krzymien, "Low-density parity-check codes for space-time wireless transmission," *IEEE Trans. Wirel. Commun.*, vol. 5, no. 2, pp. 312–322, Feb. 2006.