World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:1, No:1, 2007

Reducing Power in Error Correcting Code using Genetic Algorithm

Heesung Lee, Joonkyung Sung, and Euntai Kim

***Abstract*—**This paper proposes a method which reduces power consumption in single-error correcting, double error-detecting checker circuits that perform memory error correction code. Power is minimized with little or no impact on area and delay, using the degrees of freedom in selecting the parity check matrix of the error correcting codes. The genetic algorithm is employed to solve the non linear power optimization problem. The method is applied to two commonly used SEC-DED codes: standard Hamming and odd column weight Hsiao codes. Experiments were performed to show the performance of the proposed method.

***Keywords*—**Error correcting codes, genetic algorithm, non-linear power optimization, Hamming code, Hsiao code.

I. Introduction

AS technology continues to scale with smaller features sizes, lower power supply voltages, and higher operating frequencies, the soft error rate in logic circuits is rapidly increasing. Concurrent error detection using error correcting codes (ECCs) at the outputs of a circuit provides means to detect soft errors quickly before they have a chance to propagate and compromise the data integrity of a system. Error correcting codes are commonly used to protect against soft errors and thereby enhance system reliability and data integrity [1]. Single error correcting and double error detecting (SEC-DED) codes are generally used for this purpose. These codes are able to correct single-bit errors and detect double-bit errors in a codeword.

A design criterion that has become very important in recent times is power reduction. With increasing miniaturization of devices, power has become a first-order design consideration motivating researchers to look at techniques of reducing power consumption in all components of system design. For memory ECC, power reduction is also an important consideration, since the ECC checker circuit is activated during reading and writing access to the memory. As power has become an important consideration, researchers have begun looking at methods to reduce power consumption in error detection circuitry [2-4]. Most of the works, however, have been developed for general circuits can be applied to the design of error detection circuitry in a straightforward manner [5].

In this paper, we focus on reducing power in specific memory ECC checkers. Such circuits are widely used in industry in all types of memories including caches and embedded memories. There are many ways to construct SEC-DED codes and implement the corresponding ECC circuitry. The proposed approach selects a parity check matrix, **H**, that minimizes power using genetic algorithms (GAs). GAs provide an adaptive and robust computational procedure modeled on the mechanics of natural genetic systems [6]. GAs can be considered as a good solution for this problem, because the selection of the parity check matrix are complicated combinational problem in a large search space. Once the **H**-matrix has been selected, the corresponding ECC circuitry for implementing the code can be synthesized.

The paper is organized as follows: Section 2 gives backgrounds about SEC-DED and genetic algorithms. Section 3 gives the proposed implementing way to the memory ECC checkers consuming less power. In Section 4, the experimental results and discussion are presented. We give our conclusions in Section 5.

II. Background

A. Error correcting code

The goal of this paper is to reduce the switching activity in the part of the ECC circuitry, namely the parity generator block which is used on every memory access (both read and write). Therefore, we obtained reducing power consumption in memory ECC. In ECC circuitry, Hamming and Hsiao codes are commonly used. To construct a Hamming and Hsiao codes, the minimum weight requirement is 4, which implies that three or fewer columns of the **H**-matrix are linearly independent. One way to satisfy this condition is to have the columns of the **H**-matrix meet the following constraints:

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

A distance-4 Hsiao code is an odd-weight-column SEC-DED code, invented by M.Y**.** Hsiao [7]**.** Compared with Hamming codes, it provides improvements in speed, cost and reliability in the decoding logic**.** The following is an example of matrix of the *(8, 4)* code.

Manuscript received November 28, 2006. This work was supported by the Ministry of Commerce, Industry and Energy of Korea.

Heesung Lee is with the School of Electrical and Electronic Engineering, Yonsei University, Korea (e-mail: 4u2u@yonsei.ac.kr).

Joonkyung Sung is with the School of Electrical and Electronic Engineering, Yonsei University, Korea (e-mail: saturn2019@hanmail.net).

Euntai Kim is with the School of Electrical and Electronic Engineering, Yonsei University, Korea (corresponding author to provide phone: 82-2-2123-2863; fax: 82-2-313-2879; e-mail: etkim@yonsei.ac.kr).

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

For a *(n, k)* Hsiao code, n is the code word length and k is number of data bits. The number of parity check bits r is equal to n-k**.** The **H**-matrix is

$$\mathbf{H} = [\mathbf{A}^T \mid \mathbf{I}_{n-k}] \quad (2)$$

Where **A** is a *k*-by-*(n-k)* parity check generator matrix and $\mathbf{I}_{n-k}$ is an *(n-k)*-by-*(n-k)* identity matrix. The code generator matrix denoted as **G** is the defined as

$$\mathbf{G} = [\mathbf{I}_k \mid \mathbf{A}] \quad (3)$$

If **u** is a 1-by-k data bit vector, then his corresponding *n* bit codeword vector **x** is formed as $\mathbf{x} = \mathbf{uG}$ [8].

B. Genetic Algorithms (GAs)

Genetic algorithms are numerical optimization algorithms and they are inspired by natural selection and natural genetics. The methods have been applied to a wide range of problems [6]. The GAs typically maintain a population of individuals which represents the set of solution candidates for the optimization problem to be solved. The goodness of each candidate solution is evaluated based on its fitness value. The population of the GAs evolves by a set of genetic operators. The basic genetic operators are selection, crossover and mutation. In the selection process, some individuals are selected to be copied into a tentative next population. Individual with higher fitness value is more likely to be selected. The selected individuals are altered by the mutation and crossover and form a new population of solutions. The GAs are simple yet provide an adaptive and robust optimization methodology [9]. Given below is the basic configuration of the GAs.

```
procedure Genetic Algorithms
begin
   initialize P(t);
   while termination-condition not satisfied do
   begin
      evaluate P(t);
      select P(t+1) from P(t);
      crossover P(t+1);
      mutation P(t+1);
      t=t+1;
   end
end
```

III. Reducing Power in Error Correcting Code

As mentioned above, the focus of this paper is reducing power consumption in Hamming and Hsiao codes that provides SEC-DED. Such circuits are widely used in industry in all types of memories including caches and embedded memories. The key design issue is selecting the code that is used. A *(n, k)* linear SEC-DED block code has *n* bits in each codeword consisting of *k* data bits and *n-k* check bits. The code can be represented by a parity-check matrix, **H**, having *n-k* row, one for each check bit, and *n* columns, one for each bit in the codeword. In order for the code to be SEC-DED, the **H**-matrix must be formed in a way that the minimum distance between any codes words is 4. Two well-known methods for constructing a SEC-DED **H**-matrix were described by Hamming and Hsiao. Different **H**-matrices result in different power.

A. Encoding

Fig. 1 shows the structure of the chromosome used in our GA for Hamming code. The chromosome corresponds to a particular input permutation and is encoded as a string of the mapping for the input memory bits position. For example, for *n=64*, one possible permutation could be represented by the string "2, 3, 10, 4, 5, ... 63, 64", representing the permutation where the 1st memory bit position is mapped to the 2nd input in the checker circuit and the 2nd bit is mapped to the 3rd input, and the other memory bits are mapped to their corresponding circuit inputs.

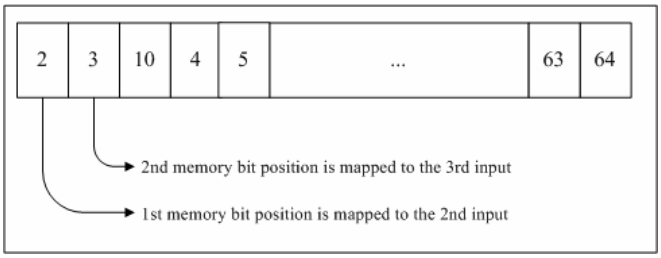


Fig. 1 The chromosome for Hamming code

Unlike Hamming code, each chromosome for Hsiao code contains an additional component representing the design of the **H**-matrix. Because we use 64-bit architecture, we index the 56 possible weight-5 columns in increasing order of their binary representation. The chromosome is represented by both the permutation of the 64 with all possible weight-1 and weight-3 columns and the indices of the 8 weight-5 columns out of the 56 possible ones that are selected to fill up the last 8 positions of the **H**-matrix. Fig. 2 shows the structure of the chromosome for Hsiao code. So, a possible chromosome would be "1, 2, …, 52", representing the indices of the particular weight-5 columns selected while creating the **H**-matrix. In the general case, for architectures of other sizes, the chromosome would have a representation of a similar design choice of selecting some columns form a total set of possible odd weight columns.

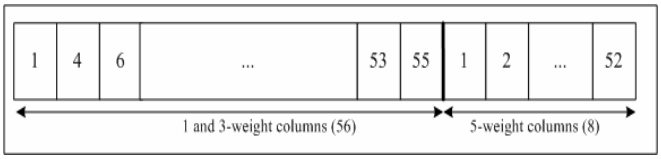


Fig. 2 The chromosome for Hsiao code

B. Genetic Operator

The operators are used to create new solutions based on existing solutions in the population. There are two basic types of operators: crossover and mutation. Crossover takes two chromosomes (parents) and produces two new chromosomes (children). For Hamming code, we use cycle crossover [9]. Because the cycle crossover preserves the absolute position of the elements in the parent chromosome, we obtain new chromosomes which have a good inherited character of the parents. We explain the cycle crossover using the following example. Suppose that two parents

$$P_1 = (1,2,3,4,5,6,7,8,9)$$
$$P_2 = (4,1,2,8,7,6,9,3,5) \quad (4)$$

are given and they produce the first chromosome by taking the first gene from the first parent.

$$C_1 = (1, X, X, X, X, X, X, X, X) \quad (5)$$

The next gene to be considered is '4', as the gene from the second parent just below the selected gene of the first parent.

$$C_1 = (1, X, X, 4, X, X, X, X, X) \quad (6)$$

This implies '8', as the gene from the second parent just below the selected gene. Thus

$$C_1 = (1, X, X, 4, X, X, X, 8, X) \quad (7)$$

Following this rule, the next genes to be included are '3' and '2'.

$$C_1 = (1,2,3,4, X, X, X, 8, X) \quad (8)$$

The selection of '2', however, requires selection of '1', which is already on the list. Thus, the remaining genes are filled from the other parent:

$$C_1 = (1,2,3,4,7,6,9,8,5) \quad (9)$$

Similarly,

$$C_2 = (4,1,2,8,5,6,7,3,9) \quad (10)$$

In case of the Hsaio code, considering the additional component of the chromosome of the Hsaio code, we use two cycle and arithmetic crossovers as in Fig. 3.

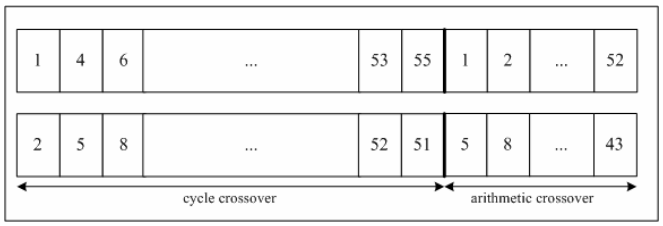


Fig. 3 Crossover operator for Hsiao code

The arithmetic crossover produces two complimentary linear combinations of the parents.

$$C_1 = rP_1 + (1-r)P_2$$
$$C_2 = (1-r)P_1 + rP_2 \quad (11)$$

Where $r = U(0,1)$.

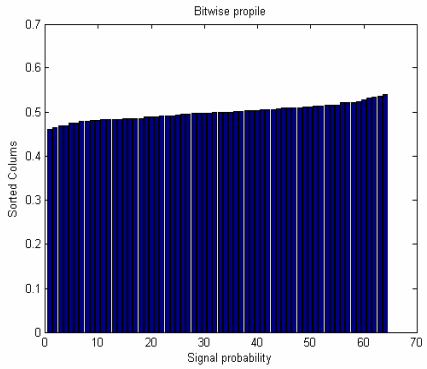
Mutation alters one chromosome to produce a single new solution. In case of both the Hamming and the Hsaio codes, new solution is created by choosing two input index mappings at random in the parent gene and swapping them.

C. Fitness of Chromosome

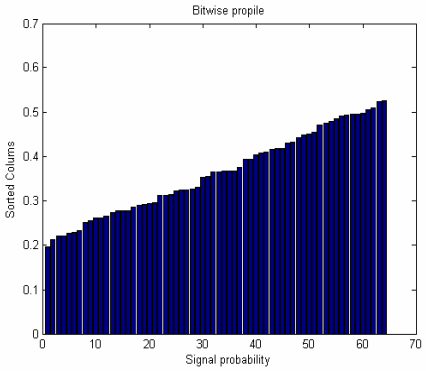
The fitness function of a chromosome is inversely proportional to the power dissipated during ECC checking where we use 2-input XOR gates and the power dissipation occur when gates switch the output. We do not have to consider delay or area because permuting the columns has no impact on area or delay.

IV. Experimental Result

We performed experiment on two synthetic memory data. Fig. 4 shows the characteristics of the two data. While all bits of the data1 are likely to have an even distribution between zero and one, the high order bits of the data2 are more likely to be a zero than one and the low order bits are more likely to have an even distribution between zero and one.



(a) data1



(b)data2

Fig. 4 The synthetic memory data

The evolution parameters used in this paper are given in Table I.

TABLE I
Evolution Parameters

Parameter	Value
Crossover rate	0.6
Mutation rate	0.05
Population size	300
generation	100

In Table Ⅱ and Ⅲ, the results of the proposed methods are compared with those of the previous method. The experimental results shown in Table Ⅱ and Ⅲ are mean and variance of 10 experiments.

TABLE Ⅱ
A COMPARISON OF THE PERFORMANCE FOR HAMMING CODE

	Random	[5]	Proposed
Data1	40387.2	40007.6 (±57.1)	39996.4 (±61.7)
Data2	37809.8	36377.4 (±60.8)	36253.1 (±63.5)

TABLE Ⅲ
A COMPARISON OF THE PERFORMANCE FOR HSAIO CODE

	Random	[5]	Proposed
Data1	32405.8	32038.9 (±65.6)	32026.3 (±71.0)
Data2	30242.8	29579.5 (±53.1)	29522.5 (±45.3)

As can be seen in the Tables, the proposed method shows the better performance than the existing method. The reason for the better performance might be that proposed method preserves the absolute position of the elements in the parent chromosome. Therefore, we obtain new chromosomes which have a good inherited character of the parents.

V. CONCLUSION

In this paper, we optimize the **H**-matrix of the memory ECC checker using GA with reducing the power consumption. Once the **H**-matrix has been selected, the corresponding ECC circuitry for implementing the code can be synthesized. A design criterion that has become very important in recent times is power reduction. With increasing miniaturization of devices, power has become a first-order design consideration motivating researchers to look at techniques of reducing power consumption in all components of system design. Because the proposed method preserves the absolute position of the elements in the parent chromosome, It obtained the better performance that the existing method. The experimental results show that the efficiency of the proposed method.

ACKNOWLEDGMENT

This work was supported by the Ministry of Commerce, Industry and Energy of Korea.

REFERENCES

[1] C. L. Chen and M. Y. Hsiao, "Error-Correcting Codes for Semiconductor memory applications: A State-of-the-Art review, *IBM J. Res. Develop.*, vol. 28, pp. 124-134, July 1984.
[2] K. Favalli and C. Metra, "Design of Low-Power CMOS Two-Rail Checkers", *Joumal of Microelectronics Systems Integration,* vol. 5**,** no. 2, pp. 101-1 IO, 1997.
[3] K. Mohanram and N. A. Touba, "Input ordering in concurrent Checkers to Reduce Power Consumption," *Proc. of IEEE Symposium on Defecated Fault Tolerance*, pp. 87-95,2002.
[4] D. Rossi, V. Dijk, R. Kleihorst, A. K. Nieuwland, and C. Metra, "Power Consumption of Fault Tolerant Codes: the Active Elements,'' *Proc. Of Intentional On-Line Testing Symposium*, pp. 61-67, 2003.
[5] S Ghosh, S Basu, NA Touba, "Reducing Power Consumption in Memory ECC Checkers," *International Test Conference,* pp. 1322-1331, 2004.
[6] L. Davis, *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, 1991.
[7] M. Y. Hsiao, "A class of optimal minimum odd-weight-column SECDED codes," *IBM J. Res. Develop.*, vol. 14, pp. 395-401, July 1970.
[8] W. Gao and S. Simmons, " A study on the VLSI implementation of ECC for embedded DRAM," *Electrical and Computer Engineering, 2003. IEEE CCECE 2003. Canadian Conf.,* vol. 1, pp. 203-206, May 2003.
[9] D. Coley, *An Introduction to Genetic Algorithms for Scientists and Engineers*, World Scientific, 1999.
[10] I. M. Oliver, D. J. Smith, and J. Holland, " A study of permutation crossover operators on the traveling salesman problem," *Proc. Of the Second Int. Conf. Genetic Algorithms, Lawrence Erlbaum Associates, Hillsdale, NJ*, 1987

Heesung Lee was born in Seoul, Korea, in 1979. He received the B.S. and the M.S. degrees in Electrical and Electronic engineering, all from Yonsei University, Seoul, Korea, in 2003 and 2005, respectively. He is currently a Ph. D. candidate of Dept. of Electrical and Electronic engineering in Yonsei University. His current research interests include computational intelligence, pattern recognition, biometrics, and robot vision.

Joonkyung Sung was born in Seoul, Korea, in 1981. He is currently a B.S. candidate of Dept. of Electrical and Electronic engineering in Yonsei University. His current research interests include computational intelligence.

Euntai Kim (S'96–M'00) was born in Seoul, Korea, in 1970. He received the B.S. (*summa cum laude*) and the M.S. and the Ph.D. degrees in electronic engineering, all from Yonsei University, Seoul, Korea, in 1992, 1994, and 1999, respectively. From 1999 to 2002, he was a full-time Lecturer in the Department of Control and Instrumentation Engineering at Hankyong National University,
Kyonggi-do, Korea. Since 2002, he has joined the faculties of the School of Electrical and Electronic Engineering at Yonsei University, where he is currently an Associate Professor. His current research interests include intelligent control and modeling, intelligent service robot, robot vision, and intelligent sensor network.
Dr. Kim is an Associate Editor of the *International Journal of Control and Systems*.