Generalized Parity-Check Matrices for SEC-DED Codes with Fixed Parity

Valentin Gherman, Samuel Evain, Nathaniel Seymour, Yannick Bonhomme
CEA, LIST, Embedded Systems Reliability Laboratory,
Point Courrier 94, 91191 Gif-sur-Yvette CEDEX, FRANCE
firstname.lastname@cea.fr

***Abstract*—Hsiao and extended Hamming parity-check matrices can be used to define systematic linear block codes for Single Error Correction-Double Error Detection (SEC-DED). Their fixed code word parity enables the construction of low density parity-check matrices and fast hardware implementations. Fixed code word parity is enabled by an all-one row in extended Hamming parity-check matrices or by the constraint that the modulo-2 sum of all rows is equal to the all-zero vector in Hsiao parity-check matrices. In this paper, we show that these two constraints are particular instantiations of a more general constraint which involves an arbitrary number of rows in the parity-check matrix. As a consequence, sparser parity-check matrices with faster hardware implementations can be found. Moreover, special instantiations of these matrices enable the detection of all triple-bit and quadruple-bit burst errors.**

Keywords—SEC-DED code; parity-check matrix

I. INTRODUCTION

Single Error Correction-Double Error Detection (SEC-DED) codes provide an effective way to increase the reliability of semiconductor memory subsystems [11]. For example, any type of one chip failure can be masked and two chip failures can be detected in SEC-DED-protected memory subsystems if each bit of any code word is stored in a different memory chip [3].

SEC-DED codes also have a relatively low performance overhead which makes them an ideal candidate for the protection of main and cache memories [4][5]. Especially, SEC-DED codes with fixed code word parity enable efficient hardware implementations since they facilitate the double error detection. Another factor that influences the latency of these codes is the density of the parity-check matrix, also called *H*-matrix, which is defined as the percentage of 1-elements with respect to the total number of matrix elements.

Nowadays, only extended Hamming and Hsiao *H*-matrices are available to implement SEC-DED codes with fixed code word parity [6][7]. In an extended Hamming *H*-matrix, the fixed code word parity is encoded with the help of an all-one row, while in a Hsiao *H*-matrix, this is ensured by the restriction to have only columns with an odd number of 1-elements. In general, the Hsiao *H*-matrices enable faster implementations due to a lower density and to a more uniform distribution of the 1-elements over the matrix rows.

Here, we propose a way to reduce the *H*-matrix density for SEC-DED codes with fixed code word parity based on a generalization of the restrictions used until now. We prove that the fixed code word parity is ensured if a subset of *H*-matrix rows can be found such that it intersects each *H*-matrix column in an odd number of 1-elements. Hsiao and extended Hamming *H*-matrices correspond to the extreme cases when this subset contains only one or all matrix rows, respectively. The reported experimental results prove that the generalized *H*-matrices enable faster implementations. Such generalizations can be applied to reduce the *H*-matrix density of any linear block code with fixed parity [1][2][8][12].

A special flavor of the new *H*-matrices is obtained if the constraint of having an odd number of 1-elements on a certain segment of each matrix column is reinforced to have a single 1-element. These special matrices enable the detection of all triple-bit and quadruple-bit burst errors. Consequently, the SEC-DED codes defined by such *H*-matrices may be helpful to protect fast on-chip inter-connections affected by cross-talk induced signal degradation [10] or fast embedded memories for which bit-interleaving [3] reveals as too expensive.

After a brief review of the linear block SEC-DED codes in Section II, the generalized *H*-matrices are introduced in Section III. The paper achievements are summarized in Section IV.

II. SYSTEMATIC LINEAR BLOCK SEC-DED CODES WITH FIXED PARITY

Binary linear block codes are described by an $r \times n$ *H*-matrix such that each code word *V* fulfills the relation below [2][9]:

$$\bigoplus_{i=0}^{n-1} \left(H_j^{\ i} \wedge V_i \right) = 0, \quad 0 \le j < r, \tag{1}$$

where *r* and *n-r* are the numbers of check and data-bits and the symbols '⊕' and '∧' denote the logic exclusive disjunction and conjunction operators, respectively. Each *H*-matrix column corresponds to a particular bit position and each *H*-matrix row to a particular check-bit position in the code words. In case of systematic codes, the *H*-matrix columns that correspond to the check-bit positions are linearly independent and usually form an identity or a triangular matrix. Consequently, these

978-1-4577-1056-8/11/$26.00 ©2011 IEEE

2011 IEEE 17th International On-Line Testing Symposium

H-matrix columns build an $r \times r$ triangular matrix in the case of extended Hamming *H*-matrices [6] or an $r \times r$ identity matrix in the case of Hsiao *H*-matrices [7] as shown in Fig. 1 and Fig. 2, respectively.

During the error checking/correction operation of a code word *V'*, syndrome bits S_j are calculated according to the following expression:

$$S_j = \bigoplus_{i=0}^{n-1} \left(H_j^{\ i} \wedge V'_i \right), \quad 0 \le j < r. \tag{2}$$

If *S* is an all-zero vector, the code word *V'* is assumed to be error-free. Otherwise, the syndrome *S* is used to correct or detect the occurred errors. A single-bit error generates a syndrome identical to the *H*-matrix column that corresponds to the corrupted bit position while a multi-bit error produces a syndrome equal to the modulo-2 sum of the *H*-matrix columns associated to the affected bit positions. Consequently, in order to ensure the SEC property, the columns of the *H*-matrix must be different from each other and from the all-zero vector. DED capacity is enabled if each column is also different from the modulo-2 sum of any other two columns.

SEC-DED codes with fixed parity are used to facilitate the double-bit error detection. The *H*-matrix in Fig. 1 enables fixed code word parity due to the all-one row that corresponds to an overall parity bit. The same property is ensured by the odd number of 1-elements on each *H*-matrix column in Fig. 2. These two *H*-matrices become identical if the last *H*-matrix row in Fig. 1 is replaced by the modulo-2 sum of all initial rows. As (1) is still satisfied if one *H*-matrix row is replaced by any linear combination of the initial rows, the *H*-matrices in Fig. 1 and Fig. 2 define the same code words. Consequently, it makes more sense to use the notion of Hsiao or extended Hamming *H*-matrices instead of Hsiao or extended Hamming codes.

If the SEC-DED code words have even parity, a double-bit error can be indicated if the relation below becomes true:

$$\overline{\bigoplus_{i=0}^{n-1} V'_i} \wedge \bigvee_{j=0}^{r-1} S_j = 1, \tag{3}$$

data-bit positions								check-bit positions				
V0	*V1*	*V2*	*V3*	*V4*	*V5*	V6	*V7*	*V8*	V9	V10	*V11*	V12
0	0	**1**	0	**1**	**1**	0	**1**	**1**	0	0	0	0
0	**1**	0	**1**	0	**1**	**1**	0	0	**1**	0	0	0
1	0	0	**1**	**1**	0	**1**	**1**	0	0	**1**	0	0
1	**1**	**1**	0	0	0	**1**	**1**	0	0	0	**1**	0
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

Figure 1. 5×13 extended Hamming *H*-matrix. (Only the last row is used to enable fixed code word parity.)

data-bit positions								check-bit positions				
V0	*V1*	*V2*	*V3*	*V4*	*V5*	V6	*V7*	*V8*	V9	V10	*V11*	V12
0	0	**1**	0	**1**	**1**	0	**1**	**1**	0	0	0	0
0	**1**	0	**1**	0	**1**	**1**	0	0	**1**	0	0	0
1	0	0	**1**	**1**	0	**1**	**1**	0	0	**1**	0	0
1	**1**	**1**	0	0	0	**1**	**1**	0	0	0	**1**	0
1	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	**1**

Figure 2. 5×13 Hsiao *H*-matrix. (All rows are used to enable fixed code word parity.)

where the first factor corresponds to the negated code word parity and the symbol '∨' stands for the logic disjunction operator. Relation (3) can also be used to detect any error that affects an even number of bits and generates a non-zero syndrome.

In case of extended Hamming *H*-matrices, the first factor in (3) can be obtained by negating the syndrome bit that corresponds to the all-one row. The parity of a Hsiao code word can also be calculated as the overall parity of the syndrome bits as follows:

$$\bigoplus_{i=0}^{n-1} V'_i = \bigoplus_{j=0}^{r-1} S_j. \tag{4}$$

An odd parity code can be obtained from an even parity code by changing the polarity of an odd number of check-bits. As the same *H*-matrices can be used to define an even or an odd SEC-DED code, only SEC-DED codes with even parity will be considered in the following.

III. Generalized *H*-Matrices for Systematic Linear Block SEC-DED Codes

We start from the observation already mentioned in the previous section: in any linear code, a code word *V* multiplied to any linear combination of the *H*-matrix rows must be zero. Based on (1), this can be formalised by the following expression:

$$\bigoplus_{j \in L} \left[\bigoplus_{i=0}^{n-1} \left(H_j^{\ i} \wedge V_i \right) \right] = 0, \tag{5}$$

where *L* contains the indices of a subset of *H*-matrix rows. If these rows intersect each *H*-matrix column in an odd number of 1-elements, then the following relations become true:

$$\bigoplus_{j \in L} H_j^{\ i} = 1, \quad 0 \le i < n, \tag{6}$$

and (5) can be transformed as shown below which indicates a fixed code word parity:

$$\bigoplus_{j \in L} \left[\bigoplus_{i=0}^{n-1} \left(H_j^{\ i} \wedge V_i \right) \right] = \bigoplus_{i=0}^{n-1} V_i = 0. \tag{7}$$

L contains only the index of one matrix row in case of an extended Hamming *H*-matrix or the indices of all rows in case of a Hsiao *H*-matrix.

Fig. 3 and Fig. 4 illustrate two generalized *H*-matrices where *L* contains the indices of 4 and 3 rows, respectively. The advantage of such generalized *H*-matrices comes from their lower density. The number of 1-elements is 28 in Fig. 3 and 27 in Fig. 4 while this number is 35 in Fig. 1 and 29 in Fig. 2.

The only restriction that must be fulfilled by the elements of the *H*-matrix rows with indices outside *L* comes from the single-bit error correction capability which requires that any *H*-matrix column is different from the all-zero vector and from each other.

Another advantage of the generalized *H*-matrices with respect to the Hsiao *H*-matrices comes from the lower cost of computing the overall parity of a code word *V'* during the decoding process. For an arbitrary code word *V'* only the first equality of (7) is true and

data-bit positions								check-bit positions				
V0	*V1*	*V2*	*V3*	*V4*	*V5*	V6	*V7*	*V8*	V9	V10	*V11*	V12
0	0	0	0	1	1	1	1	1	0	0	0	0
0	1	1	1	0	1	0	0	0	1	0	0	0
1	1	1	0	1	0	1	0	0	0	1	0	0
1	1	0	1	1	0	0	1	0	0	0	1	0
1	0	1	1	1	0	0	0	1	0	0	0	1

Figure 3. 5×13 SEC-DED *H*-matrix with 4 rows used to enable fixed code word parity.

data-bit positions								check-bit positions				
V0	*V1*	*V2*	*V3*	*V4*	*V5*	V6	*V7*	*V8*	V9	V10	*V11*	V12
0	1	1	1	1	0	1	0	1	0	0	0	0
0	1	1	1	0	1	0	1	0	1	0	0	0
1	1	0	0	1	0	0	1	0	0	1	0	0
1	0	1	0	0	1	0	0	1	0	0	1	0
1	0	0	1	0	0	1	0	0	1	0	0	1

Figure 4. 5×13 SEC-DED *H*-matrix with 3 rows used to enable fixed code word parity.

based on (2) it comes out that the overall parity of *V'* is equal to the parity of the syndrome bits with indices in the subset *L* as illustrated below.

$$\bigoplus_{i=0}^{n-1} V'_i = \bigoplus_{j\in L}\left[\bigoplus_{i=0}^{n-1}\left(H_j{}^i \wedge V'_i\right)\right] = \bigoplus_{j\in L} S_j. \quad (8)$$

As long as the cardinality of *L* is smaller than the number of rows *r* in the *H*-matrix, the modulo-2 sum in the right-hand side of (8) requires a smaller hardware overhead as compared to (4).

An existing *H*-matrix can be optimized based on the property expressed in (5): a code word that fulfils (1) will also fulfill it if the *H*-matrix is replaced by an *H'*-matrix whose rows are linear combinations of the initial *H*-matrix rows. For example, the code words defined by the *H*-matrix in Fig. 1 can also be defined by the *H'*-matrix in Fig. 5 obtained by replacing the last matrix row in Fig. 1 with the modulo-2 sum of the last two rows. In this way, the number of 1-elements in the *H*-matrix has been reduced from 35 to 29. Such transformations can be performed on the *H*-matrices of any linear block code. For example, this can be applied to double error correction-triple error detection BCH codes [1][2][8][12].

In Table I, we present the 1-element ratios between generalized, Hsiao and extended Hamming *H*-matrices for some of the most commonly used SEC-DED codes. Only generalized *H*-matrices that are different from extended Hamming and Hsiao *H*-matrices are considered. As compared to Hsiao matrices, the generalized *H*-matrices have a lower density except for the case when the number of data bits is equal to 32 and 64. The most important density reduction, 7.4%, was obtained for codes with 8 data-bits. Non-negligible reductions were also achieved for SEC-DED codes with wide code words which are interesting due to their reduced check-bit overhead.

The densities of extended Hamming *H*-matrices are more important due to the presence of the all-one row. As it will be seen later, this does not necessarily result in a slower implementation since the all-one row takes into

data-bit positions								check-bit positions				
V0	*V1*	*V2*	*V3*	*V4*	*V5*	V6	*V7*	*V8*	V9	V10	*V11*	V12
0	0	1	0	1	1	0	1	1	0	0	0	0
0	1	0	1	0	1	1	0	0	1	0	0	0
1	0	0	1	1	0	1	1	0	0	1	0	0
1	1	1	0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	1	1	0	1

Figure 5. 5×13 SEC-DED *H*-matrix with 2 complementary rows used to enable fixed code word parity.

TABLE I. NUMBER OF 1-ELEMENTS IN DIFFERENT TYPES OF SEC-DED *H*-MATRICES

Data bits (n-r)	Extended Hamming	Hsiao	Gene-ra-lized	Generalized w.r.t. Extended Hamming	Generalized w.r.t. Hsiao
8	35	29	27	17.1%	7.4 %
16	65	54	51	21.5%	5.6 %
32	126	103	104	18.3%	-1.0%
64	258	216	216	16.3%	0.0%
128	545	481	461	15.4%	4.2%
256	1153	1050	1010	12.4%	3.8%
512	2508	2241	2182	13.0%	2.6%

account all operations required for the calculation of the code word parity, which is not the case with the other *H*-matrices. Moreover, as long as only the correction of data-bit errors is required, only *r-1* *H*-matrix rows need to be considered to perform single error correction and the only row that can be neglected in extended Hamming *H*-matrices is the all-one row.

We implemented SEC-DED decoders for several Hsiao, extended Hamming and generalized *H*-matrices with the goal to get the minimum possible latency under the constraint that single error correction and double error detection can be performed in one clock cycle. Relation (3) was used to indicate the presence of double-bit errors. The overall code word parities were calculated with the first factor in (3) for all *H*-matrices, alternatively with (4) for all Hsiao *H*-matrices and with (8) in the case of generalized SEC-DED *H*-matrices. Synthesis results obtained with Synopsys Design Compiler and a 45nm standard cell library (TSMC N40LP CMOS) are reported in Table II and Table III. These results correspond to the case when only the data bits were made available at the decoder output.

The generalized *H*-matrices provided the smallest latencies despite the fact that hand-optimized versions of Hsiao *H*-matrices from [7] were used. Not only the lower density of the generalized *H*-matrices contributed to these improvements but also to the usage of (8) that simplifies the computation of the overall parity and implicitly the detection of double-bit errors. The only two cases reported in Table II for which no latency reduction could be obtained correspond to the two cases in Table I for which no improvements of the Hsiao *H*-matrix density could be achieved.

Since the main synthesis target was the improvement

of the decoder speed, the best latency results were accompanied by a certain logic overhead as reported in Table III. The negative figures represent a reduction of the decoder area. The largest overhead was slightly above 10% which can be considered as acceptable due to the very small size of these decoders.

Special versions of the generalized *H*-matrices enable fast detection of certain triple-bit and quadruple-bit errors. In these *H*-matrices, the rows with indices in the subset *L* are complementary. This means that all these rows define a sub-matrix with one and only one 1-element in each column.

An example of an *H*-matrix with 3 complementary rows for a SEC-DED code with 16 data-bits is shown in Fig. 6. In case of a single-bit error, one and only one of the 3 syndrome bits which corresponds to complementary *H*-matrix rows becomes equal to 1. However, in case of burst errors that affect 3 or 4 adjacent bit positions at least 2 of these 3 syndrome bits become 1. (The *H*-matrix in Fig. 6 is sparser than any 6×22 Hsiao *H*-matrix.)

TABLE II. IMPROVEMENTS OF MINIMAL DECODER LATENCY OBTAINED WITH THE GENERALIZED *H*-MATRICES

Data-bits (*n-r*)	Generalized w.r.t. extended Hamming	Generalized w.r.t. Hsiao
8	0.0%	3.1%
16	4.1%	5.3%
32	1.2%	0.0%
64	3.1%	0.0%
128	6.1%	2.7%
256	1.0%	5.4%
512	4.9%	2.8%

TABLE III. DECODER LOGIC OVERHEAD FOR MINIMAL DECODER LATENCY

Data-bits (*n-r*)	Generalized w.r.t. extended Hamming	Generalized w.r.t. Hsiao
8	-7.3%	3.0%
16	0.5%	10.9%
32	5.8%	-1.2%
64	4.6%	2.2%
128	3.6%	1.7%
256	-2.6%	-0.8%
512	3.5%	0.8%

IV. CONCLUSIONS

We proposed the concept of generalized *H*-matrices for SEC-DED codes according to which a variable number of *H*-matrix rows can be used to provide fixed code word parity. Hsiao and extended Hamming *H*-matrices become particular cases of the new concept and sparser *H*-matrices can be found for a wide range of code word sizes. Moreover, with the generalized *H*-matrices a lower number of syndrome bits can be used to compute the overall code word parity as compared to Hsiao *H*-matrices. Synthesis results proved the potential of the generalized *H*-matrices to increase the design space of SEC-DED codes with fixed code word parity and to enable more efficient hardware implementations.

REFERENCES

[1] E.R. Berlekamp "Algebraic coding theory," McGraw-Hill Book Co., Inc., New York, 1968.

[2] C.L. Chen, M.Y. Hsiao "Error-correcting codes for semiconductor memory applications: A State of the Art Review," Reliable Computer Systems - Design and Evaluation, Digital Press, 2nd edition, 1992.

[3] T.J. Dell "A white paper on the benefits of Chipkill-correct ECC for PC server main memory," IBM Microelectronics Division, 1997.

[4] J. Dorsey et al. "An Integrated Quad-Core Opteron Processor," Proc. Intl. Solid-State Circuits Conf., 2007.

[5] J. Friedrich et al. "Design of the Power6 Microprocessor," Intl. Solid-State Circuits Conf., 2007.

[6] R.W. Hamming "Error correcting and error detecting codes," Bell Sys. Tech. Journal, Vol. 29, April 1950, pp. 147-160.

[7] M.Y. Hsiao "A class of optimal minimum oddweight-column SEC-DED codes," IBM Journal of Research and Development, Vol. 14, 1970, pp. 395-401.

[8] P.K. Lala "An adaptive double error correction scheme for semiconductor memory systems," Digital Processes 4, 1978, pp. 237-243.

[9] F.J. MacWilliams, N.J.A. Sloane "The theory of error-correcting codes," NortH-Holland Mathematical Library, 2007.

[10] C. Metra, M. Favalli, B. Ricco "On-line detection of logic errors due to crosstalk, delay, and transient faults," IEEE International Test Conference, 1998, pp. 524-533.

[11] L. Spainhower, T.A. Gregg "IBM S/390 parallel enterprise server G5 fault tolerance: A historical perspective," IBM J. Research and Development, vol. 43, nos. 5/6, 1999, pp. 863-874.

[12] C.-E.W. Sundberg "Erasure and error decoding for semiconductor memories," IEEE Trans. Computers C-21, August 1978, pp. 696-705.

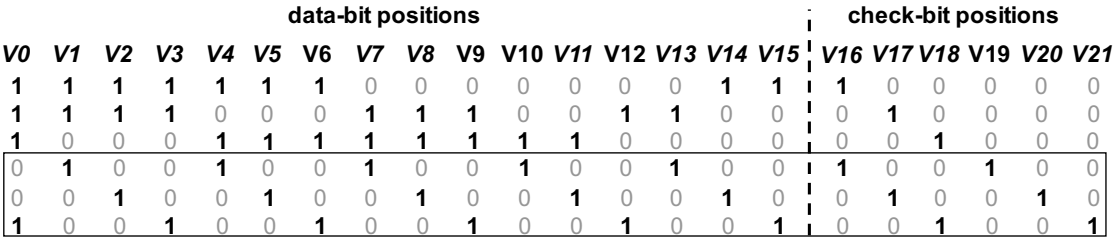

Figure 6. 6×22 *H*-matrix with 3 complementary rows.