Detection and location of errors by linear inequality checks

M. Karpovsky, Mem. I.E.E.E.

Indexing terms: *Decoding, Error analysis*

Abstract: Problems of error detection and error location in programs or specialised devices computing values of real functions are considered. Systems of linear inequality checks are used for error detection and error location. Theorems are given for solving the problem of the error-detecting and locating capabilities of memoryless and memory-aided decoding procedures based on linear-inequality checks.

1 Introduction

We shall consider the problems of error detection and error location in programs or devices for computing real functions. By errors, we mean errors in the text of the programs or catastrophic structural failures in digital devices. As in References 1–5, we shall use the additive way of describing the influence of errors; namely, by *the error* e in a program or a device computing $f(x)$, we mean the function $e(x)$ such that, as a result of the error, our program or device computes $f(x) + e(x)$.

By the *multiplicity* $\|e\|$ of the error e, we mean the number of nonzero values of the function $e(x)$. (This definition of the multiplicity of errors is natural if errors for different xs are independent. This may be the case, for example, when $f(x)$ is the word stored in a memory cell whose address is x. In this case, the multiplicity $\|e\|$ of an error is the number of faulty cells in our memory.)

For error detection and/or location we shall use an approach based on the analysis of the results of some linear checks. This approach was developed in References 1–5 and it was based on the very powerful techniques of Fourier transforms over finite groups, corresponding fast Fourier transforms, least-absolute-error polynomial approximations and linear error-correcting codes.

We note also that techniques based on Fourier (Walsh) transforms over finite groups were widely used for the problems of logic design [5–11], for the design of linear systems over the groups [12, 13], and in fault-tolerant computing [16].

Let $x = (x_1, \ldots, x_n) \epsilon G$, $x_i \epsilon \{0, 1\}$, so that G is the group of binary n-vectors with respect to the operation $\oplus$ of componentwise addition mod 2.

In References [1–4], the methods of error detection in a device or program computing $f(x)$ based on *linear equality checks*, namely

$$\sum_{\tau \in T} f(x \oplus \tau) - C = 0 \tag{1}$$

were investigated. In eqn. 1, T is a 'check' subgroup of G, and C is a constant. In a program or device computing $f(x)$, the problem of error correction by a system of linear equality checks was considered in References 2 and 3. It was shown in References 1–4 that linear equality checks have very good error-detecting and/or error-correcting capabilities and may be easily implemented. Very simple equality checks were constructed in Reference 2 for many standard computer blocks and in Reference 4 for programs which evaluate polynomials.

It was shown [1–4] that equality checks may be effectively used in the case where $f(x)$ is an integer for every x, and very few noninteger functions have nontrivial equality checks.

The generalisation of linear check methods to the case of noninteger computations was given in Reference 5. It was proposed that *linear inequality checks*

$$\sum_{\tau \in T} f(x \oplus \tau) - C \leqslant \epsilon \tag{2}$$

(where ϵ is a given small constant) should be used for error detection in numerical computations (checks (eqn. 1) are a special case of (eqn. 2) with $\epsilon = 0$). The method of constructing optimal inequality so as to minimise the cardinality $|T|$ of a check set T (and therefore the testing time) was proposed, and optimal checks for such important noninteger computations as exponential, logarithmic and trigonometric computations were given.

We shall describe, in this paper, methods or *error detection* and/or *error location* by systems of linear inequality checks. We shall introduce two methods of error detection and location by the analysis of results of the checks, namely memoryless and memory-aided decoding. Finally, we shall describe the error-detecting and error-locating capabilities of both methods of decoding.

The detection and/or location of errors by systems of linear inequality checks may be effectively used for acceptance-testing of programs in the course of development, or of devices in the course of manufacture. For example, this approach can be used for testing a read only memory (ROM) containing the value $f(x)$ in a cell whose address is x*. In the case of testing a random access memory (RAM) we first have to choose $f(x)$, and then write in every cell the value $f(x)$ corresponding to its address x. After this we scan out the memory, verify the checks (eqn. 2) and by analysing the results of these checks we can detect or locate errors.

This approach is applicable to stuck-at faults in cells of a ROM or RAM, stuck-at faults at the outputs of an address decoder, bridging faults between output lines of the decoder and faults that affect power supply or read/write circuits. The testing time in this case is less than for such widely used procedures as SHIFTED DIAGONAL, GALLOPING COLUMNS, WALKPAT, GALWERC and GALPAT [19–21].

We note also that the well known syndrome testing technique [18] is a special case of the inequality checks (eqn. 2) with $\epsilon = 0$ and $T = G$.

The inequality-check approach is a high-level functional testing technique which does not depend on the implementation of a program or device computing the given function $f(x)$. The constructed checks for many practical cases are very simple (see Reference 5 and Karpovsky (op. cit.,

Paper 1699E, first received 26th January and in revised form 16th September 1981

The author is with the Computer Science Department, State University of New York, Binghamton, New York 13901, USA

*KARPOVSKY, M.G.: 'Memory testing by linear checks' (IEEE, under consideration)

Table 1: Upper bounds for check complexity $\log_2|T|$ for functions $f(y)$ with $\max_{y\in[0,1)} |f^{(s)}(y)| \leqslant 1$ for every $s \leqslant n$, $y = 2^{-n}Z$, $Z \in \{0, 1, \ldots, 2^n-1\}$

$-\log_{10}\epsilon$						
n	1	2	3	4	5	6
3	2	3	3	3	3	3
4	3	3	4	4	4	4
5	3	4	4	5	5	5
6	3	4	5	5	6	6
7	3	4	6	6	6	7
8	4	4	7	7	7	8
9	5	7	7	8	8	9
10	5	7	8	9	9	9
11	5	7	8	9	10	10
12	5	8	8	10	10	11
13	5	9	10	11	12	12
14	5	9	10	11	12	13
15	5	9	10	11	13	13
16	5	9	11	11	14	14
17	6	9	11	12	14	15
18	6	9	11	12	15	15
19	6	10	11	12	16	16
20	6	10	11	12	16	16
21	6	10	11	12	16	16
22	6	10	11	12	17	18
23	6	10	11	12	17	18
24	6	10	12	12	18	19
25	6	11	13	13	19	19
26	6	11	14	18	19	20
27	6	11	14	18	19	20
28	6	11	14	18	19	20
29	6	11	15	18	19	20
30	6	11	15	19	19	20
31	6	11	15	20	20	20

$\|e\| = 1$), we have, from eqn. 10:

$$\left|\sum_{\tau\in T_i} e(Z\oplus\tau)\right| = |e(t)| > 2\epsilon \qquad (11)$$

Thus for a single error $e(x) = \delta_{x,t}e(t)$ $(t \in Z\oplus T_i)$, we have, from eqns. 9 and 11, $S_i^{(e)}(Z) = 1$, and this error is detected.

We shall consider two methods of error detection and/or location by the previously computed binary syndrome vector $S^{(e)}(x) = (S_1^{(e)}(x), \ldots, S_m^{(e)}(x))$ (see eqn. 9), namely memoryless and memory-aided decoding.

In the case of memoryless decoding for every given x, we first compute $S^{(e)}(x)$, and then by the analysis of $S^{(e)}(x)$ in the case of error detection we decide whether there exists $\tau \in \bigcup_{i=1}^{m} T_i$, such that $e(x\oplus\tau) \neq 0$ and in the case of error location we decide whether $e(x) \neq 0$.

In the case of memory-aided decoding, for every given x we first compute $S_x^{(e)} = \left\{S^{(e)}(x\oplus\tau) \,\middle|\, \tau \in \bigoplus_{i=1}^{m} T_i\right\} \left(\bigoplus_{i=1}^{m} T_i = \{t_1\oplus\ldots\oplus t_m \quad t_j \in T_j, i = 1, \ldots, m\}\right)$, and then, by the analysis of the set $S_x^{(e)}$ of syndromes, in the case of error detection we decide whether there exists $\tau \in \bigoplus_{i=1}^{m} T_i$, such that $e(x\oplus\tau) \neq 0$, and in the case of error location we compute the error locator

$$l(t) = 1 - \delta_{0,e(t)} \qquad \left\{\sum_{t\in G} l(t) = \|e\|\right\} \qquad (12)$$

for all $t \in x \oplus \bigoplus_{i=1}^{m} T_i$.

We note that these definitions of error detection and error location by memoryless and memory-aided decoding are very similar to the corresponding definitions of error detection and error correction by systems of orthogonal equality checks [2, 3]. The main difference is that, in the case of inequality checks for every x syndrome, $S^{(e)}(x)$ is a *binary* vector, and instead of computing the error $e(x)$ we compute the error locator $l(x)$ (see eqn. 12).

The following results have been proven [3] for m orthogonal equality checks.

For memoryless decoding:

(*a*) All errors with multiplicity at most m are detected, and there exist errors with multiplicity $m + 1$ which cannot be detected.

(*b*) All errors with multiplicity at most $[m/2]$ ($[a]$ is the greatest integer less or equal a) are corrected, and there exist errors with multiplicity $[m/2] + 1$ which cannot be corrected.

For memory-aided decoding:

(*a*) All errors with multiplicity at most $2^m - 1$ are detected and there exist errors with multiplicity 2^m which cannot be corrected.

(*b*) All errors with multiplicity at most $2^{m-1} - 1$ can be corrected and there exist errors with multiplicity 2^{m-1} which cannot be corrected.

We shall see in Section 3 that, for memoryless decoding, the error-detecting and the error-locating capabilities of inequality checks are equal correspondingly to the error-detecting and the error-correcting capabilities of equality checks.

For memory-aided decoding, we shall see in Section 4 that the error-detecting capabilities of equality and inequality checks are equal, but the error-locating capability of inequality checks is less than the error-correcting capability of equality checks.

3 Memoryless decoding for a system of inequality checks

We shall consider in this Section the error-detecting and error-locating capabilities of a system of m orthogonal linear inequality checks (eqn. 7) for the case of memoryless decoding.

Theorem 1

For any system of m orthogonal inequality checks, we have for memoryless decoding:

(*a*) All errors with multiplicity at most m are detected.

(*b*) There exist errors with multiplicity $m + 1$ which are not detected.

Proof

(*a*) The error e is not detected under memoryless decoding, iff there exists x such that $e(x) \neq 0$ and

$$\left|\sum_{\tau\in T_i} \{f(x\oplus\tau) + e(x\oplus\tau)\} - C_i\right| \leqslant \epsilon \quad (i = 1, \ldots, m) \qquad (13)$$

Thus, we have from eqns. 7 and 13:

$$\left|\sum_{\tau\in T_i} e(x\oplus\tau)\right| = \left|e(x) + \sum_{\tau\in T_i-0^n} e(x\oplus\tau)\right| \leqslant \epsilon$$
$$(i = 1, \ldots, m) \qquad (14)$$

Since $|e(x)| > 2\epsilon$, it follows from eqn. 14 that, for every $i = 1, \ldots, m$, there exists $\tau_i \in T_i - 0^n$ such that $e(\tau_i) \neq 0$. For any x, T_i and $T_j (i \neq j)$, we have from the orthogonality relation (eqn. 8) $\{x\oplus T_i\} \cap \{x\oplus T_j\} = x$, hence $\tau_i \neq \tau_j$, $\|e\| \geqslant m + 1$, and any error with multiplicity not greater than m is detected by m orthogonal inequality checks.

(*b*) We define the error e_0 as follows:

$$e_0(x) = \begin{cases} 3\epsilon, x = 0^n \\ -3\epsilon, x \in \{\tau_1, \ldots, \tau_m\} \text{ where } \tau_i \in T_i - 0^n \\ 0, \text{otherwise} \end{cases} \qquad (15)$$

For any binary vector $\sigma = (\sigma_1, \ldots, \sigma_m) \in \{0, 1\}^m$, we denote

$$M(\sigma) = \bigoplus_{i=1}^{m} \sigma_i(T_i - 0^n) = \left\{ \bigoplus_{i=1}^{m} \sigma_i \tau_i \,|\, \tau_i \in T_i - 0^n; \right.$$
$$\left. \sigma_i \tau_i = \tau_i \qquad \text{if } \sigma_i = 1 \text{ and } \sigma_i \tau_i = 0^n \text{ if } \sigma_i = 0 \right\}$$

Utilising conditions similar to those described in References 2 and 3, we also require that, for any $\alpha, \beta \epsilon \{0, 1\}^m \quad (\alpha \neq \beta)$

$$M(\alpha) \cap M(\beta) = \emptyset \quad (\emptyset \text{ is the empty set}) \tag{23}$$

(Note that by setting $\alpha = (0^{i-1}\ 1\ 0^{m-i}), \beta = (0^{j-1}\ 1\ 0^{m-j})$ we have, by eqn. 23, $T_i \cap T_j = 0^n$.

Theorem 3

For any system of m inequality checks satisfying eqn. 23, we have for memory-aided decoding:

(*a*) All errors with multiplicity at most $2^m - 1$ are detected.

(*b*) There exist errors with multiplicity 2^m which are not detected.

Proof

(*a*) Let $e(x) \neq 0$ for some x. If $\|e\| \leqslant 2^m - 1$, then there exist $\alpha, \beta \in \{0, 1\}^m$ and $i \in \{1, \ldots, m\}$ such that

$$M(\alpha) = M(\beta) \oplus \{T_i - 0^n\} \tag{24}$$

$$e(x \oplus \tau) = 0 \text{ for all } \tau \in M(\alpha) \tag{25}$$

$$e(x \oplus \tau_\beta) \neq 0 \text{ for some } \tau_\beta \in M(\beta) \tag{26}$$

It follows from eqn. 24 that $\tau_\beta \oplus \tau \in M(\alpha)$ for all $\tau \in T_i - 0^n$. Thus, we have from eqns. 24, 25 and 26:

$$\left|\sum_{\tau \in T_i} e(x \oplus \tau_\beta \oplus \tau)\right| = \left| e(x \oplus \tau_\beta) + \sum_{\tau \in T_i - 0^n} e(x \oplus \tau_\beta \oplus \tau)\right|$$
$$= \left| e(x \oplus \tau_\beta) + \sum_{\tau \in M(\alpha)} e(x \oplus \tau) \right| = |e(x \oplus \tau_\beta)| > 2\epsilon \tag{27}$$

Hence, from eqns. 7, 9 and 27, we have $S_i^{(e)}(x \oplus \tau_\beta) = 1$, and all errors e with $0 < \|e\| < 2^m$ are detected.

(*b*) We now construct the undetectable error e_0 with multiplicity 2^m. Let us fix $\tau_i \in T_i - 0^n$ $(i = 1, \ldots, m)$ and set

$$e_0(x) = \begin{cases} (-1)^{\|\sigma\|} 3\epsilon \text{ if there exist } \sigma = (\sigma_1, \ldots, \sigma_m) \\ \qquad \in \{0, 1\}^m \text{ such that } x = \bigoplus_{i=1}^{m} \sigma_i \tau_i \\ 0, \text{ otherwise} \end{cases} \tag{28}$$

$(\|e_0\| = 2^m)$

If for some x and $i \in \{1, \ldots, m\} e_0(x \oplus \tau) = 0$ for all $\tau \in T_i$, then $\sum_{\tau \in T_i} e_0(x \oplus \tau) = 0$ and $S_i^{(e_0)}(x) = 0$.

If there exists $t \in T_i$, such that $e_0(x \oplus t) \neq 0$, for some x and i, then by eqn. 28 there exists $I \subset \{1, \ldots, m\}$, such that $x \oplus t = \bigoplus_{j \in I} \tau_j$. In this case, $e_0(x \oplus t) = 3\epsilon(-1)^{|I|}$, and by eqn. 28,

$$\sum_{\tau \in T_i} e_0(x \oplus \tau) = e_0(x \oplus t) + \sum_{\tau \in T_i - t} e_0(x \oplus \tau)$$
$$= 3\epsilon(-1)^{|I|} + \sum_{\tau \in T_i - t} e_0\left(\bigoplus_{j \in I} \tau_j \oplus \tau\right)$$
$$= 3\epsilon(-1)^{|I|} + e_0\left(\bigoplus_{j \in I} \tau_j \oplus \tau_i\right) \tag{29}$$

Since, from eqn. 28

$$e_0\left(\bigoplus_{j \in I} \tau_j \oplus \tau_i\right) = \begin{cases} 3\epsilon(-1)^{|I|-1}, & \text{if } i \in I \\ 3\epsilon(-1)^{|I|+1}, & \text{if } i \notin I \end{cases} \tag{30}$$

we finally have, from eqns. 29 and 30, for this case:

$$\sum_{x \in T_{i(e_0)}} e_0(x \oplus \tau) = 0$$

Thus, $S_i(x) = 0$ for all x and i, and e_0 cannot be detected by memory-aided decoding. Suppose $G \Big/ \bigoplus_{i=1}^{m} T_i$ is the set of coset representatives of subgroup $\bigoplus_{i=1}^{m} T_i$ in G. $\left(\left|G \Big/ \bigoplus_{i=1}^{m} T_i\right| = 2^n \prod_{i=1}^{m} \left|T_i\right|^{-1}\right)$. To detect errors by memory-aided decoding for every $x_j \in G \Big/ \bigoplus_{i=1}^{m} T_i$, we have to compute $S^{(e)}(x_j \oplus \tau)$ for all $\tau \in \bigoplus_{i=1}^{m} T_i$. Then we conclude that $e(x_j \oplus \tau) = 0$ for all $\tau \in \bigoplus_{i=1}^{m} T_i$ iff $S^{(e)}(x_j \oplus \tau) = 0^m$ for all $\tau \in \bigoplus_{i=1}^{m} T_i$.

Let us now consider the problem of error location by memory-aided decoding.

Theorem 4

For any system of m inequality checks satisfying eqn. 23, we have for memory-aided decoding:

(*a*) All errors with multiplicity at most m are located.

(*b*) There exist errors with multiplicity $m + 1$ which are not located.

Proof

For any two errors e_1 and e_2 with locators $l_1 \neq l_2$ (see eqn. 12), there exists $x \in G$ such that

$$e_1(x) = 0 \text{ and } |e_2(x)| > 2\epsilon \tag{31}$$

Denote

$$L_j(\sigma) = \begin{cases} 1, \text{ if there exists } \tau_\sigma \in x \oplus M(\sigma) \text{ such that} \\ \qquad |e_j(x \oplus \tau_\sigma)| > 2\epsilon \\ 0, \text{ otherwise } (j = 1, 2) \end{cases} \tag{32}$$

Since $\|e_1\|, \|e_2\| \leqslant m$, we have:

$$|\{\sigma | L_1(\sigma) = L_2(\sigma) = 1\}| < m \tag{33}$$

Thus, there exists $\alpha, \beta \in \{0, 1\}^m$ and $i \in \{1, \ldots, m\}$ such that

$$M(\alpha) = M(\beta) \oplus \{T_i - 0^n\} \tag{34}$$

$$L_1(\beta) \neq L_2(\beta) \tag{35}$$

$$L_1(\alpha) = L_2(\alpha) = 0 \tag{36}$$

Then, by eqns. 32 and 35 there exists $\tau_\beta \in x \oplus M(\beta)$, such that

$$l_1(x \oplus \tau_\beta) \neq l_2(x \oplus \tau_\beta) \tag{37}$$

Since, for any $\tau \in T_i - 0^n$, it follows from eqn. 36 that

$$e_1(x \oplus \tau_\beta \oplus \tau) = e_2(x \oplus \tau_\beta \oplus \tau) = 0 \tag{38}$$

we have, from eqns. 34 and 38:

$$\sum_{\tau \in T_i} e_j(x \oplus \tau_\beta \oplus \tau) = e_j(x \oplus \tau_\beta) \qquad (j = 1, 2) \tag{39}$$

and from eqns. 37 and 39

$$S_i^{(e_1)}(x \oplus \tau_\beta) \neq S_i^{(e_2)}(x \oplus \tau_\beta) \tag{40}$$

Consequently, all errors with multiplicity at most m are located.

(*b*) We now construct two errors e_1 and e_2 such that $\|e_1\| = m + 1, l_1 \neq l_2$ but $S^{(e_1)}(x) = S^{(e_2)}(x)$ for all x.

11 BESSLICH, P.W.: 'Determination of the irredundant forms of a Boolean function using Walsh-Hadamard analysis and dyadic groups', *IEE J. Compt. & Digital Tech.* 1978, **1**, (4), pp. 143–151
12 PEARL, T.: 'Optimal dyadic models of time-invariant systems', *IEEE Trans.*, 1975, **C-24**, pp. 598–603
13 KARPOVSKY, M.G., and TRACHTENBERG, E.A.: 'Some optimization problems for convolution systems over finite groups', *Inf. and Control*, 1977, **34**, pp. 227–247
14 REDINBO, G.R.: 'Noise analysis of soft errors in combinational digital circuits'. Proceedings of 1980 IEEE international symposium on electromagnetic compatibility, Baltimore, 1980, pp. 307–312
15 LYUSTERNIK, L.A., CHEZVONENKIZ, D.A., and YAMPOLSKII, A.R.: 'Handbook for computing elementary functions' (Pergamon Press, 1965)
16 PETERSON, W.W.: 'Error correcting codes' (MIT Press, Cambridge, MA, 1961)
17 KORN, G.A., and KORN, T.M.: 'Mathematical handbook for scientists and engineers' (McGraw-Hill, 1968)
18 SAVIR, J.: 'Syndrome-testable design of combinational circuits', *IEEE Trans.*, 1980, **C-29**, pp. 442–451
19 FRIEDMAN, A.D., and BREUER, M.A.: 'Diagnosis and reliable design of digital systems' (Computer Sci. Press, CA, 1976), Chap. 3
20 FEE, W.G.: 'Memory testing, tutorial on LSI testing', *IEEE Computer*, 1977, **7**, pp. 81–88
21 COCKING, J.: 'RAM test pattern and test strategy'. Proceedings of 1975 semiconductor test conference, pp. 1–8

Mark Karpovsky was born in Leningrad, USSR on 27th October, 1940. He received the B.S., M.S. and Ph.D. degrees in computing from the Leningrad Electrotechnical Institute, Leningrad, USSR in 1961, 1963 and 1967, respectively.

He is currently an associate professor in the Computer Science Department at the State University of New York at Binghamton. His interests include fault-tolerant computing, test generation, reliable software, error-correcting codes, logic and system design. He has published about 40 papers and two books in these areas.