Testing for Multiple-Valued Computations

M. Karpovsky, Member IEEE

Computer Science Department
State University of New York
Binghamton, New York 13901

* This work was partially supported by the Division of Mathematical and Computer Sciences of NSF under Grant No. MCS-8008339

Abstract

We consider methods for testing (error detection, correction, and location) in multiple-valued computations. These methods are based on systems of linear equality and inequality checks and analysis of the corresponding syndromes. The error detecting/correcting/locating capabilities of these checks are described.

I. Testing by Equality Checks

Suppose we are given a device or a program computing $f(z)$ where z and $f(z)$ are represented in the p-ary form.

($z = \sum_{i=0}^{m-1} z^{(i)} p^{m-1-i}$, $p \geq 2$, $z^{(i)} \in \{0,\ldots,p-1\}$, the case $p = 10$ is important from the practical point of view).

Let G be the group of all p-ary m-vectors with respect to componentwise addition mod p.

An error $e(z)$ is said to be present in a device or a program computing $f(z)$ if for the latter $f(z) + e(z)$ is computed instead of $f(z)$. As in [1-5], by the multiplicity of an error $e(z)$, we mean the numbers of non-zero values for the function $e(z)$.

To detect errors, one can compute $f(z)$ for all $z \in G = \{0,1,\ldots,p^m-1\}$, and then verify that $\sum_{z \in G} f(z)$ is equal to the precomputed constant C[6].

This method is very time-consuming in most cases. The generalization of this approach, which may result in considerable decrease in testing time, have been described in [1-5]. In this case linear equality checks

$$\sum_{\tau \in T_i} f(z \oplus \tau) = C_i \quad (i=1,\ldots N) \text{ for all } z \in G \tag{1}$$

have been used. In (1) T is a subgroup in G, $\oplus$ stands for the componentwise addition mod p of p-ary vectors z and $\tau \in G$, and C_i is a precomputed constant.

Methods for the constructing of optimal equality checks (1) have been described in [1,5]. These methods are based on Fourier transforms over G. (Chrestenson transforms [5,7]).

Example 1. Let $x = (x^{(0)},\ldots, x^{(n-1)})$, $y = (y^{(0)},\ldots,y^{(n-1)})$ $(m=2n)$

$x^{(i)},y^{(j)} \in \{0,\ldots,p-1\}$ and $f(x,y) = \sum_{i,j=0}^{n-1} a_{ij}\, x^{(i)} y^{(j)}$ $(a_{ij} \in \{0,\ldots,p-1\})$. (2)

Then [2], the following check can be used for the testing of a device or program computing the p-ary quadratic form (2)

$$\sum_{i,j=0}^{p-1} f\,(x \oplus (i,i,\ldots,i)),\ y \oplus (j,j,\ldots,j)) = \frac{1}{4}\, p^2 \cdot (p-1)^2 \cdot A \text{ for all } x,y \qquad (3)$$

where $A = \sum_{i,j=0}^{n-1} a_{i,j}$

In the case of polynomial computations [4,5] $f(z)=Q_d(z)=\sum_{i=0}^{d} a_i z^i$, $z\in\{0,\ldots,p^m-1\}$

linear checks (1) may be constructed by p-ary error correcting codes. For polynomial computations the optimal check set T in (1) with the minimal cardinality $|T|$ is a code $V^{\perp}(m,d+1)$ which is dual to the maximal p-ary error-correcting code with the length of codewords m and the Hamming distance d+1.

Error detection or error correction for linear equality checks (1) is implemented by the analysis of the syndrome

$S^{(e)}(z) = (S_1^{(e)}(z),\ldots, S_n^{(e)}(z))$ where

$$S_i^{(e)}(z) = \sum_{\tau\in T_i} (f(z \oplus \tau) + e\,(z \oplus \tau)) - C_i = \sum_{\tau\in T_i} e(z \oplus \tau). \qquad (4)$$

(By error correction we mean the computation of an error e using the previously computed syndrome $S^{(e)}$).

Two methods of error detection/correction by the analysis of the syndrome $S^{(e)}$ have been considered, namely memoryless and memory-aided decoding.

For memoryless decoding error detection/correction for any given $z \varepsilon G$ is implemented by $S^{(e)}(z)$; for memory-aided decoding we first compute $S^{(e)}(z)$ for all $z \varepsilon G$ and then detect or correct errors. The following results [2,5] describe an error-detecting and an error-correcting capability of systems of N orthogonal equality checks. (The checks (1) are orthogonal if $T_i \cap T_j = 0$ for $i \neq j$.)

(i) For memoryless decoding: all errors with multiplicity at most N are detected, and all those with multiplicity at most $[\frac{N}{2}]$ are corrected; there exist errors with multiplicity N+1 and $[\frac{N}{2}]+1$ which are not detected and not corrected respectively ($[\frac{N}{2}]$ is the greatest integer less or equal $\frac{N}{2}$).

(ii) For memory-aided decoding: all errors with multiplicity at most 2^N-1 are detected, and all those with multiplicity at most $2^{N-1}-1$ are corrected; there exist errors with multiplicity 2^N and 2^{N-1}, which are not detected or corrected respectively.

The previous results illustrate a good error detecting/correcting capability of equality checks.

Complexities and error detecting/correcting capabilities increase exponentially on transition from memoryless to memory-aided decoding.

II. Testing by Inequality Checks

Equality checks considered in Section I may be effectively used in the case where f(z) is an integer for every $z\varepsilon\{0,1,\ldots,p^m-1\}$ and very few noninteger functions have nontrivial checks of this type.

In this section, we shall generalize the linear checks methods to the case of noninteger p-ary computations. We shall use for error detection linear inequality checks

$$\left| \sum_{\tau\varepsilon T} f(z \oplus \tau) - C \right| \leq \varepsilon \quad \text{for all } z \varepsilon G, \tag{5}$$

where T is a subgroup in G and $\varepsilon \geq 0$ is a small constant (Checks (1) is a special case of (5) with $\varepsilon = 0$).

To construct an optimal inequality check (5) minimizing the testing time $|T|$ we shall use the techniques of least-absolute-error polynomial approximation for f(z) and our previous result on equality checks for polynomials (see Section I).

Let $Q_d(z) = \sum_{i=0}^{d} a_i z^i$ be a least-absolute-error approximation for f(z),

$f(z) = Q_d(z) + \Delta_d(z)$ and $|\Delta_d(z)| \leq \Delta_d$ for all $z\varepsilon G$. Using the equality check for $Q_d(z)$ with $T = V^{\perp}(m,d+1)$ (see Section I) we have

$$\left| \sum_{\tau\varepsilon V^{\perp}(m,d+1)} f(z \oplus \tau)-C \right| = \left| \left| \sum_{\tau\varepsilon V^{\perp}(m,d+1)} Q_d(z \oplus \tau) - C + \sum_{\tau\varepsilon V^{\perp}(m,d+1)} \Delta_d(z \oplus \tau) \right| \right| =$$

$$= \left| \sum_{\tau\varepsilon V^{\perp}(m,d+1)} \Delta_d(z \oplus \tau) \right| \leq \Delta_d \left| V^{\perp}(m,d+1) \right| . \tag{6}$$

Thus, we have from (5), (6) that the p-ary code $V^{\perp}$ (m,d+1), which is dual to the maximal error-correcting code with the Hamming distance d+1, is the check set for f(z) if

$$\Delta_d \mid V^{\perp}(m,d+1)\mid \leq \varepsilon. \tag{7}$$

We note, that for the great variety of analytical functions f(z) $\Delta_d \mid V^{\perp}(m,d+1)\mid$ decreases very rapidly with the increase of the degree d ($d \leq m$) of an approximating polynomial $Q_d(z)$ for all $p \geq 2$.

The values of $\mid V^{\perp}(m,d+1)\mid$ are well known in the coding theory [9]. If $Q_d(z)$ is an interpolating polynomial such that

$Q_d(p^{-n}\, i\, d^{-1}) = f(p^{-n}\, i\, d^{-1})$ $(i = 0,\ldots,d)$, then,

$$\Delta_d \leq ((d+1)!)^{-1} \prod_{i=0}^{d} \mid p^{-n} - id^{-1}\mid \max_{z\varepsilon G} \mid f^{(d+1)}(p^{-n}z)\mid , \tag{8}$$

where $f^{(d+1)}$ is d+1-th derivative of f.

Example 2 Suppose we have a ternary memory (p = 3) with m = 13 ternary address lines, where in a cell with an address z $(z \in \{0,1,\ldots,3^{13}-1\})$ the value f(z) is stored and

$$f(z) = (3^{-13}\, z)^{-0.5}\ \ \mathrm{Sin}\left(\frac{\pi}{2}(3^{-13}\, z)^{-0.5}\right). \tag{9}$$

Let us construct an optimal inequality check (5) for this memory with $\varepsilon = 5 \times 10^{-3}$.

The function f(z) can be approximated by the polynomial Q_2 (z) of degree two [8]:

$$P_2 (y) = 0.07287\ y^2 - 0.64338\ y + 1.57064; \tag{10}$$

$$\max_y | \Delta_2 (y)| = \Delta_2 \leq 14 \times 10^{-5} \text{ where } y = 3^{-13} z \ .$$

Choose the perfect ternary (13,10) Hamming code with the distance 3 [9] as V (m,d+1) = V(13,3); then, $|V^{\perp}(13,3)| = 3^3$ and

$$\Delta_2 \ | \ V^{\perp} (13,3)| = 14 \times 10^{-5} \times 3^3 \leq \varepsilon = 5 \times 10^{-3}.$$

Thus, the dual code to the perfect ternary (13, 10) Hamming code is an optimal check set T for this memory.

For the constant C in (5) we have in this case

$$C = 3^{-10} \sum_{z=0}^{3^{13}-1} P_2 (3^{-13} z).$$

III. Error-Detecting and Error-Locating Capabilities of Inequality Checks

Using systems of inequality checks

$$\left|\sum_{\tau \in T_i} f(z \oplus \tau) - C_i \right| \leq \varepsilon \quad (i = 1,\ldots,N) \tag{11}$$

we cannot correct errors, but errors can be located in this case.

For an error e(z) by error location we mean the computation of the error locator $\ell(z)$

$$\ell(z) = \begin{cases} 1 & \text{if } e(z) \neq 0; \\ 0, & \text{if } e(z) = 0. \end{cases} \tag{12}$$

For error detection/location by a system of N orthogonal ($T_i \cap T_j = 0$ $i \neq j$) inequality checks (11) we shall use the binary syndrome

$S^{(e)}(z) = (S_1^{(e)}(z), \ldots, S_N^{(e)}(z))$ where

$$S_i^{(e)}(z) = \begin{cases} 0, & \text{if } \left| \sum_{\tau \in T_i} (f(z \oplus \tau) + e(z \oplus \tau)) - C_i \right| \leq \varepsilon; \\ 1, & \text{otherwise} \end{cases} \quad (i = 1, \ldots, N) \tag{13}$$

From now on we suppose that

$$\min_{\{z | e(z) \neq 0\}} |e(z)| > 2\varepsilon. \tag{14}$$

(The last condition may be used for the choice of ε for practical applications).

Using the technique similiar to the case of equality checks, we can prove the following results for N orthogonal inequality checks.

(i) For memoryless decoding:

all errors with multiplicity at most N are detected, and all those with multiplicity at most $[\frac{N}{2}]$ are located; there exist errors with multiplicity N+1 and $[\frac{N}{2}] + 1$ which are not detected and not located, respectively.

(ii) For memory-aided decoding:

all errors with multiplicity at most $2^N - 1$ are detected, and all those with multiplicity at most N are located; there exist errors with multiplicity 2^N and N+1 which are not detected and not located, respectively.

The error-detecting capability increases exponentially on transition from memoryless to memory-aided decoding, as in the case of equality checks (see Section I), whereas the error-locating capability of inequality checks increases only by a factor of

two with this transition.

For the important from the practical point of view case of two checks we have: (i) all single and double errors are detected and all single errors are located by memoryless decoding; (ii) all single, double, and triple errors are detected and all single and double errors are located by memory-aided decoding.

References

1. Karpovsky, M. G., "Error Detection" in Digital Devices and Computer Programs with the Aid of Linear Recurrent Equations over Finite Commutative Groups", IEEE Trans. on Computers, C-26, N3, 208-219,1977.

2. Karpovsky, M. G. and Trachtenberg, E. A., "Linear Checking Equations and Error-Correcting Capability for Computation Channels", Proc. 1977 IFIP Congress, North Holland 1977.

3. Karpovsky, M. G. and Trachtenberg, E. A., "Fourier Transforms over Finite Groups for Error Detection and Error Correction in Computation Channels", Information and Control, Vol. 40, N3, 1979.

4. Karpovsky, M. G., "Error Detection for Polynomial Computations", IEE J. on Computer and Digital Techniques, Vol. 2, No. 1, Feb. 1979.

5. Karpovsky, M. G., "Spectral Methods for Decomposition, Design and Testing of Multiple-valued Logical Networks", Proc. of 1981 Symposium on Multiple-valued Logic, Oklahoma, 1981.

6. Savir, J.,"Syndorme-Testable Design of Combinational Circuits", IEEE Trans on Computers, C-29, June 1980.

7. Karpovsky, M. G., "Finite Orthogonal Series in the Design of Digital Devices", Wiley New York, 1976.

8. Lyusternik, L. A., Chezvonenkis, D. A., and Yanpolskii, A. R., "Handbook for Computing Elementary Functions", Pergamon Press, 1965.

9. Peterson, W. W., and Weldon, E. J., "Error Correcting Codes", MIT Press, Cambridge, Mass., 1972, 2nd edn.