Error detection for polynomial computations

M. Karpovsky

Indexing terms: *Computational complexity, Error correction codes, Error detection, Fourier transforms, Polynomials, Walsh functions*

Abstract: We consider the problem of error detection in a process of computation of a polynomial over the field of complex numbers or over $GF(p)$. By errors we mean errors in the text of a program or 'stuck-at' errors in a device computing a polynomial. For error detection we use linear checks constructed by the technique of Fourier transformation over the group of binary vectors. Complexity estimations, optimal checks and estimations of the error-correcting capability of these checks are obtained.

1 Introduction

We consider the problem of error detection in the process of computing polynomials

$$f(x_1, \ldots, x_m) = \sum_{i_1=0}^{s_1} \cdots \sum_{i_m=0}^{s_m} a(i_1, \ldots, i_m) x_1^{i_1}, \ldots, x_m^{i_m} \quad (1)$$

where $x_t \in \{0, \ldots, 2^{n_t} - 1\}$ $(t = 1, \ldots, m)$, and all the computations are in the field C of complex numbers or in a finite field $GF(p)$ ($p > 2$ is a prime). The set of all polynomials of this type we denote $K_{s_1,\ldots,s_m}[x_1, \ldots, x_m]$.

The errors to be detected are errors in the texts of the programs, in the case where f is calculated by a computer program, and they are catastrophic stable structural failures in the case where computations are carried out by a specialised digital device. For practical reasons, we suppose that the argument x_t is represented in binary form, $x_t = (x_t^{(0)}, \ldots, x_t^{(n_t-1)})$. [The generalisation of the results of this paper to the case when x_t is represented by q_t-ary vectors $(t = 1, \ldots, m)$ may be done without any difficulties].

Denote by C_2^r the group of binary vectors with r components $(r = 1, \ldots, \sum_{t=1}^{m} n_t)$ with respect to componentwise addition mod 2. For error detection, we shall use linear checks over $GF(2)$:

$$\sum_{\tau=(\tau_1,\ldots,\tau_m)\in T} f(x_1 \oplus \tau_1, \ldots, x_m \oplus \tau_m) = d_f$$

$$\text{for every } x_1, \ldots, x_m \quad (2)$$

when T ('check set for f') is a subgroup of C_2^n, $n = \sum_{t=1}^{m} n_t$, $\tau_t \in C_2^{n_t}$, d_f — some constant, the symbol $\oplus$ stands for componentwise addition mod 2 of binary vectors, and summation is carried out in the same field as in eqn. 1.

The verification of whether condition eqn. 2 is satisfied for the given $x_1, \ldots, x_m$ constitutes an error-detection method. The method may be effectively used for the testing of manufacturing acceptance of the program or of the device computing the given polynomial. In the case of network implementation (see Section 4), the method may be used for testing during installation and maintenance of the corresponding devices. As examples of these devices, we note such standard computer blocks as counters, adders, subtractors and multipliers.

We note, also, that, although all the examples given below deal with computations in the field of real numbers, the same error-detecting technique may be used for the computations of polynomials with complex coefficients.

We shall denote by $T(f)$ a check set T with minimal cardinality. We shall use the cardinality $|T(f)|$ as a criterion for check complexity of the polynomial f.

Our first problem is to find for the given polynomial $f \in K_{s_1,\ldots,s_m}[x_1, \ldots, x_m]$ a minimal check set $T(f)$. The implementation of the check eqn. 2 and error-detecting capability of these checks will be considered in Section 4.

For the search of $T(f)$ for the given f we shall use the method proposed in Reference 1. [The generalisation of this method for the functions defined over an arbitrary (possibly even non-Abelian) finite group and for the case of error detection and/or error correction is given in Reference 2]. This method is based on the technique of Fourier transforms over the group C_2^n. The advantages and limitations of this technique were discussed in Reference 1. A similar technique was used in References 3 and 4 for the problems of logical design. We note also that there is another simple method of error detection for polynomial computations by linear checks based on the finite differences of the orders $s_1, \ldots, s_m$ for polynomial equation 1. In this case, we have the following check:

$$\sum_{\tau_1=0}^{s_1} \cdots \sum_{\tau_m=0}^{s_m} (-1)^{\tau_1+\ldots+\tau_m} \binom{s_1}{\tau_1} \cdots \binom{s_m}{\tau_m}$$

$$f(x_1 + s_1 - \tau_1, \ldots, x_m + s_m - \tau_m)$$

$$= a(s_1, \ldots, s_m) \prod_{t=1}^{m} s_t! \quad (3)$$

The number of values of f involved in check eqn. 3 is $\prod_{t=1}^{m} (s_t + 1)$, and this number, generally speaking, is less than the corresponding number $|T(f)|$ for check eqn. 2; (Estimations for $|T(f)|$ will be given in Section 3), but there are at least three disadvantages to check eqn. 3. These are as follows:

(*a*) For implementation of eqn. 3 we need 'arithmetical shifts' of arguments x_t, whereas in eqn. 2 we need only componentwise additions mod 2

(*b*) For check eqn. 3 we need additional multiplications by the constants

$$(-1)^{\tau_1+\ldots+\tau_m} \prod_{t=1}^{m} \binom{s_t}{\tau_t}$$

Paper T304 C, first received 21st June and in revised form 20th November 1978

Dr. Karpovsky is with the Computer Science Department, School of Advanced Technology, State University of New York at Binghamton, Binghamton, New York 13901, USA

(c) Check eqn. 3 cannot detect errors in coefficients $a(i_1, \ldots, i_m)$ resulting in the replacement of the given polynomial f by another polynomial $\varphi \in K_{s_1,\ldots,s_m}[x_1, \ldots, x_m]$ with the same $a(s_1, \ldots, s_m)$. We shall see in Section 4 that eqn. 2 detects almost all errors of this type.

2 Complexities of linear checks over GF(2)

Before going to the minimal checks for polynomials, we shall consider some general properties of check complexities $|T(f)|$. Denote

$$(f_1 \oplus f_2)(z) = \sum_{x \in C_2^n} f_1(x) f_2(x \oplus z)$$

Theorem 1

(i) Linear transform of arguments: Let σ be an $(n \times n)$ binary nonsingular over $GF(2)$ matrix, $y \in C_2^n$ and

$$\varphi(x) = f(\sigma x \oplus y) \tag{4}$$

for every $x \in C_2^n$.
Then, $|T(\varphi)| = |T(f)|$.

(ii) Linear transform of functions: Let $\psi = \sum_{i=1}^{r} C_i f_i$, where $C_1, \ldots, C_r$ are some constants. Then

$$|T(\psi)| \leqslant \left| \bigoplus_{i=1}^{r} T(f_i) \right|$$

where

$$\bigoplus_{i=1}^{r} T(f_i) = \{\tau_1 \oplus \ldots \oplus \tau_r | \tau_1 \in T(f_1), \ldots, \tau_r \in T(f_r)\} \tag{5}$$

(iii) Convolution of functions over C_2^n: Let $\xi = f_1 \oplus \ldots \oplus f_r$, then

$$|T(\xi)| \leqslant \min_i |T(f_i)| \tag{6}$$

(iv) Necessary condition for nontrivial checks: If $f: C_2^n \to \{0, \pm 1, \ldots\}$ and $|T(f)| < 2^n$, then there exist $d \in \{0, \pm 1, \ldots\}$ and $i \in \{1, \ldots, n-1\}$ such that

$$\sum_{x \in C_i^n} f(x) = d 2^i \tag{7}$$

(v) Lower bound for check complexity: If $f: C_2^n \to \{0, 1, \ldots\}$ and $f \not\equiv 0$ then

$$|T(f)| \geqslant 2^n \left(\sum_{x \in C_2^n} f(x) \right)^{-1} \min_{\{x | f(x) \neq 0\}} f(x) \tag{8}$$

and there exists $f: C_2^n \to \{0, 1, \ldots\}$ such that the bound expression 8 is reached.

Proof

(i) By definition of $T(f)$, there exists d_f such that $\sum_{\tau \in T(f)} f(x \oplus \tau) = d_f$ for every $x \in C_2^n$. Then, we have from eqn. 4

$$\sum_{\tau \in \sigma^{-1} T(f)} \varphi(x \oplus \tau) = \sum_{\tau \in T(f)} \varphi(x \oplus \sigma^{-1} \tau)$$

$$= \sum_{\tau \in T(f)} f(\sigma x \oplus y \oplus \tau) = d_f$$

(σ^{-1} is the inverse of σ over $GF(2)$, and $\sigma^{-1} T(f) = \{\sigma^{-1} \tau | t \in T(f)\}$) and $\sigma^{-1} T(f)$ is a check set for $\varphi(x)$; $\sigma^{-1} T(f)$ is a minimal check set because $|\sigma^{-1} T(f)| = |T(f)|$.

(ii) Let $T = \bigoplus_{i=1}^{r} T(f_i)$ and T_i be a subgroup isomorphic to the factor group $T/T(f_i)$. Then

$$\sum_{\tau \in T} \Psi(x \oplus \tau) = \sum_{\tau \in T_i} \sum_{\tau_i \in T(f_i)} \Psi(x \oplus \tau_i \oplus \tau)$$
$$(i = 1, \ldots, r)$$

and, because

$$\sum_{\tau_i \in T(f_i)} f_i(x \oplus \tau_i) = d_{f_i} \quad (i = 1, \ldots, r)$$

for every $x \in C_2^n$ we have

$$\sum_{\tau \in T} \Psi(x \oplus \tau) = \sum_{i=1}^{r} C_i \sum_{\tau \in T_i} \sum_{\tau_i \in T(f_i)} f_i(x \oplus \tau \oplus \tau_i)$$
$$\sum_{i=1}^{r} C_i d_{f_i} |T_i|$$

and

$$T = \bigoplus_{i=1}^{r} T(f_i)$$

is a check set for Ψ.

(iii) Let $\delta_i(x)$ be a characteristic function for $T(f_i)$:

$$\delta_i(x) = \begin{cases} 1, x \in T(f_i) \\ 0, x \notin T(f_i) \end{cases}$$

Then, by definition of $T(f_i)$, $(f_i \oplus \delta_i)(x) = d_{f_i}$ for all $x \in C_2^n$, and there exists the constant d_ξ such that for every $i = 1, \ldots, r$

$$\xi \oplus \delta_i = \bigoplus_{s \neq i} f_s \oplus f_i \oplus \delta_i = \bigoplus_{s \neq i} f_s \oplus d_{f_i} = d_\xi$$

and $T(f_i)$ is a check set for ξ.

(iv) If H is isomorphic to $C_2^n/T(f_i)$, then

$$\sum_{x \in C_i^n} f(x) = \sum_{x \in H} \sum_{y \in T(f)} f(x \oplus y) = d|H| = d 2^i$$

(v) Formula 8 follows from eqn. 7 because $|H| = 2^n |T(f)|^{-1}$ and $d \geqslant \min_{f(x) \neq 0} f(x)$. The bound inequality 8 is reached, for example, for

$$f(x) = \begin{cases} 1, 0 \leqslant x < 2^{n-1} \\ 0, 2^{n-1} \leqslant x < 2^n \end{cases}$$

In this case

$$\sum_{x \in C_2^n} f(x) = 2^{n-1}, d = 1$$

and $f(x) + f\{x \oplus (1, 0, \ldots, 0)\} = 1$.

Thus: $\sigma^{-1} T(f)$ is a minimal check set for $\varphi(x) = f(\sigma x \oplus \bigoplus_{i=1}^{r} y)$; $\bigoplus_{i=1}^{r} T(f_i)$ is a check set for $\Psi = \sum_{i=1}^{r} C_i f_i$; for every $i = 1, \ldots, r$, $T(f_i)$ is a check set for $\xi = \bigoplus f_i$; and for linear

systems over C_2^n with input f_2, impulse-response function f_1 and output $e = f_1 \ominus f_2$ (these systems have been considered[5-7]) the same check may be used for error detection in input and output signals.

We note also that, because it follows from step (iv), only a very small part of functions defined over C_2^n have nontrivial checks over $GF(2)$, but it will be shown in the following Section that these nontrivial checks always exist in the case of polynomial computations.

3 Linear checks for polynomial computations

Let, for $f \in K_{s_1,\ldots,s_m}[x_1,\ldots,x_m]$,

$$f(x_1,\ldots,x_m) = \sum_{i_1=0}^{s_1} \cdots \sum_{i_m=0}^{s_m} a(i_1,\ldots,i_m) \times x_1^{i_1} \ldots x_m^{i_m}; x_t \in \{0,\ldots,2^{n_t}-1\}$$

$$x_t = (x_t^{(0)},\ldots,x_t^{(n_t-1)}) \in C_2^{n_t}, (x_t^{(j)} \in \{0,1\});$$

$$s_t < n_t (t = 1,\ldots,m)$$

For $f(x_1,\ldots,x_m) = f(x^{(0)},\ldots,x^{(n-1)})$ $(x^{(j)} \in \{0,1\}; j = 0,\ldots,n-1; n = \sum_{t=1}^{m} n_t)$ we define Fourier (Walsh) transform $f \to \hat{f}$ and inverse Fourier transform $\hat{f} \to f$ over C_2^n by the formulas

$$\hat{f}(\omega) = \hat{f}(\omega^{(0)},\ldots,\omega^{(n-1)})$$

$$= 2^{-n} \sum_{x^{(0)},\ldots,x^{(n-1)} \in \{0,1\}} f(x^{(0)},\ldots,x^{(n-1)}) W_\omega(x)$$

$$f(x) = f(x^{(0)},\ldots,x^{(n-1)})$$

$$= \sum_{\omega^{(0)},\ldots,\omega^{(n-1)} \in \{0,1\}} \hat{f}(\omega^{(0)},\ldots,\omega^{(n-1)}) W_\omega(x)$$

where

$$W_\omega(x) = (-1)^{\sum_{i=0}^{n-1} x^{(i)}\omega^{(i)}}$$

is the ωth character of C_2^n or so-called Walsh function.[8] For Fourier transforms (eqn. 9) the main properties of classical Fourier transform remain valid (Reference 8, Section 1.4). We note here the convolution property of this transform

$$\Psi = f \circledast \varphi \quad \text{iff} \quad \hat{\Psi} = 2^n \hat{f}\hat{\varphi} \tag{10}$$

Denote by $V(n_t, s_t+1)$ a maximal linear code in $C_2^{n_t}$ with Hamming distance $s_t + 1$[9]

$$V^{\perp}(n_t, s_t+1) = \{(\tau_t^{(0)},\ldots,\tau_t^{(n_t-1)}) \overset{n_t-1}{\underset{j=0}{\oplus}} \tau_t^{(j)} y_t^{(j)} = 0$$

for every $(y_t^{(0)},\ldots,y_t^{(n_t-1)}) \in V(n_t, s_t+1)\}$ and

$$V^{\perp} = \prod_{t=1}^{m} V^{\perp}(n_t, s_t+1)$$

$$= \{\tau = (\tau_1,\ldots,\tau_m) | \tau_1 \in V^{\perp}(n_1, s_1+1), \ldots, \tau_m \in V^{\perp}(n_m, s_m+1)\}$$

Theorem 2

(i) For every $f \in K_{s_1,\ldots,s_m}[x_1,\ldots,x_m]$

$$\sum_{\tau=(\tau_1,\ldots,\tau_m) \in V^{\perp}} f(x_1 \oplus \tau_1,\ldots,x_m \oplus \tau_m) = d_f \tag{11}$$

where

$$d_f = \prod_{t=1}^{m} |V(n_t, s_t+1)|^{-1} \sum_{x_1,\ldots,x_m} f(x_1,\ldots,x_m) \tag{12}$$

(ii) For every $s = (s_1,\ldots,s_m)$, $(n_1,\ldots,n_m)$ $(s_t < n_t)$ there exists $f \in K_{s_1,\ldots,s_m}[x_1,\ldots,x_m]$ such that $T(f) = V^{\perp}$.

Proof

(i) Let us consider a Fourier image (see eqn. 9) $\hat{f}$ of f over C_2^n:

$$n = \sum_{t=1}^{m} n_t.$$

If there exists t such that

$$\|\omega_t\| = \sum_{j=0}^{n_t-1} \omega_t^{(j)} \geqslant s_t + 1,$$

then

$$\hat{f}(\omega) = 0 (\omega = (\omega_1,\ldots,\omega_n), \omega_t \in C_2^{n_t}) \tag{13}$$

because, for every $j < n_t$,

$$\hat{x}_t^j(\omega t) = 0$$

if $\|\omega_t\| \geqslant j+1$ (Reference 8, Section 3.5). Denote

$$\delta_v(\omega) = \begin{cases} \sum_{t=1}^{m} |V(n_t, s_t+1)|^{-1} & \text{if} \quad \omega \in V \\ 0 & \text{if} \quad \omega \notin V \end{cases}$$

$$\left(V = \prod_{t=1}^{m} V(n_t, s_t+1)\right) \tag{14}$$

then

$$\hat{f}(\omega)\hat{\delta}_v(\omega) = \delta_{\omega;0} 2^{-n} \sum_{t=1}^{m} |V(n_t, s_t+1)|^{-1} \sum_{x_1,\ldots,x_m} f(x_1,\ldots,x_m) \tag{15}$$

($\delta_{\omega;0}$ = Kronecker-delta symbol).
From eqns. 12 and 15 we have

$$(f \circledast \delta_v) = d_f \tag{16}$$

for every $x \in C_2^n$.
For $\delta_v(x)$, we have, from eqn. 14, the definition of $V^{\perp}(n_t, S_t+1)$ and (9)

$$\delta_v(x) = \begin{cases} 1, x \in V^{\perp} \\ 0, x \notin V^{\perp} \end{cases} \tag{17}$$

and eqn. 11 follows from eqns 16 and 17.
(ii) Let

$$f(x_1,\ldots,x_m) = \sum_{t=1}^{m} c(t) f_t(x_t), c(t) > 0,$$

and

$$f_t(x_t) = a(t,0) + a(t,1)\bar{x}_t + \ldots + a(t,s_t)\bar{x}_t^{s_t}$$

$$(t = 1,\ldots,m)$$

where

$$\bar{x}_t = x_t - \tfrac{1}{2}(2^{n_t}-1),\ a(t,j)>0 \quad \text{if} \quad j = 2k$$

and $\quad a(t,j)<0 \quad \text{if} \quad j = 2k+1.$

Then, from Reference 8, Section 3.5,

$$\bar{\bar{x}}_t = \sum_{l=0}^{n_t-1} (-2^{n_t-2-l}) W_{l_t}(x_t)$$

$$l_t = (\underbrace{0,\ldots,0}_{t},1,0,\ldots,0),\ W_{l_t}(x_t) = (-1)^{x_t^{(l)}}$$

and

$$\bar{x}_t^j = \sum_{\omega \in C_2^n} b(t,\omega) W_\omega(x_t)$$

where

$$b(t,\omega)>0, \quad \text{if} \quad j = 2k \quad \text{and}$$

$$\|\omega\| = \sum_{l=0}^{n_t-1} \omega^{(l)} \leqslant j;\ b(t,\omega)<0, \quad \text{if} \quad j = 2k+1$$

and $\quad \|\omega\| \leqslant j;\ b(t,\omega) = 0 \quad \text{if} \quad \|\omega\| \geqslant j+1$

for all $\quad t = 1,\ldots,m;\ \omega \in C_2^n.$

Hence, by definition of f, we have

$$\hat{f}(\omega_1,\ldots,\omega_m)>0, \|\omega_t\| \leqslant s_t \quad (t = 1,\ldots,m) \quad (18)$$

and

$$\hat{f}(\omega_1,\ldots,\omega_m) = 0, \quad \text{if there exists } t \text{ such that}$$

$$\|\omega_t\| \geqslant s_t + 1$$

Now, let δ_T be the characteristic function of the subgroup T of C_2^n:

$$\delta_T(x) = \begin{cases} 1, & x \in T \\ 0, & x \notin T \end{cases}$$

Then, from eqn. 9,

$$\hat{\delta}_T(\omega) = \begin{cases} |T|^{-1} = |T|2^{-n}, & \omega \in T^\perp \\ 0, & \omega \notin T^\perp \end{cases} \quad (19)$$

where

$$T^\perp = \{\omega = (\omega_1,\ldots,\omega_m) \Big| \bigoplus_{t=1}^{m} \bigoplus_{l=0} \omega_l^{(l)} x_l^{(l)} = 0$$

for all $\quad x = (x_1,\ldots,x_m) \in T\}.$

If T is a check set for f, then there exists the constant d such that $(f \oplus \delta_T)(x) = d$ for every x or, by eqn. 10,

$$\hat{f}(\omega)\hat{\delta}_T(\omega) = \delta_{\omega_1;0} \ldots \delta_{\omega_1;0} 2^{-n} d \quad (20)$$

Because

$$V = \prod_{t=1}^{m} V(n_t, S_t + 1)$$

a maximal subgroup in the set $\{(\omega_1,\ldots,\omega_m) | \exists\, t\, \|\omega_t\| \geqslant \omega_t + 1\}$, it follows from eqns. 18–20 that $V = T^\perp(f)$, $T(f) = V^\perp$, and this completes the proof of theorem 2.

Theorem 2 generates a simple method of constructing a check eqn. 2 for the given polynomial $f \in K_{s_1,\ldots,s_m}[x_1,\ldots,x_m]$. This method reduces to the following operations:

(*a*) Choose any error correcting codes $V(n_t, s_t+1)$ with the distance s_t+1 in n_t-dimensional space of binary vectors ($t = 1,\ldots,m$).

(*b*) Construct the corresponding orthogonal codes $V^\perp(n_t, s_t+1)$ $(t = 1,\ldots,m)$. Methods for choosing $V(n_t, s_t+1)$ and for constructing $V^\perp(n_t, s_t+1)$ may be found in Reference 9.

(*c*) Construct the direct product

$$T_f = V^\perp = \prod_{t=1}^{m} V^\perp(n_t, s_t+1).$$

(*d*) Compute by eqn. 12 the right-hand constand d.

We note that the check set $V^\perp$ for $f \in K_{s_1,\ldots,s_m}[x_1,\ldots,x_m]$ depends only on degrees $s_1,\ldots,s_m$ of f, but right-hand constant d_f of the check depends on coefficients of f.

For the estimation of check complexity of polynomials from $K_{s_1,\ldots,s_m}[x_1,\ldots,x_m]$ one may use the following corollary from theorem 2:

Denote

$$\langle a \rangle = 2^l \quad \text{iff} \quad 2^{l-1} \leqslant a < 2^l \quad (21)$$

Corollary 1

For every $f \in K_{s_1,\ldots,s_m}[x_1,\ldots,x_m]$

$$\log_2 |T(f)| \leqslant \sum_{t=1}^{m} \log_2 \left\langle \sum_{j=0}^{s_t-1} \binom{n_t-1}{j} \right\rangle \quad (22)$$

Proof

From theorem 2 we have

$$|T(f)| \leqslant |V^\perp| = 2^n \prod_{t=1}^{m} |V(n_t, s_t+1)|^{-1}$$

and eqn. 22 follows now from the Varshamov–Gilbert bound for $|V(n_t, s_t+1)$.[9]

We note that lower bounds for

$$\max_{f \in K_{s_1,\ldots,s_m}[x_1,\ldots,x_m]} \log_2 |T(f)|$$

may be also obtained from theorem 2 by the Hamming–Rao bound or Plotkin bound for $|V(n_t, s_t+1)|$[9].

It follows from theorem 2 and corollary 1 that it is expedient to use linear checks over $GF(2)$ in the case of relatively small number m of arguments. We note also that $V^\perp$ is a check set for every $f \in K_{s_1,\ldots,s_m}[x_1,\ldots,x_m]$, but, for some special f, checks may be considerably simplified. For example, it is easy to show that, if f depends only on even (or only on odd) degrees of $\bar{x}_t = x_t - \frac{1}{2}(2^{n_t}-1)$, then

$$f(x_1,\ldots,x_m) + (-1)^{1+\sum_{t=1}^{m} s_t} f(x_1 \oplus \mathbf{1},\ldots,x_m \oplus \mathbf{1}) = 0 \quad (23)$$

where $\mathbf{1} = (1,\ldots,1)$.

We shall now consider the case, important from the practical point of view, of small-degree polynomials ($s_t = 1, 2, 3; t = 1,\ldots,m$). Let

$$f(x_1,\ldots,x_m) =$$

$$\sum_{i_1,\ldots,i_m \in \{0,\ldots,s\}} a(i_1,\ldots,i_m) x_1^{i_1} \ldots x_m^{i_m} \quad (0 \leqslant x_t \leqslant 2^{n_t}-1)$$

We shall say that f depends on x_t^j, if there exist

$$i_1, \ldots, i_{t-1}, i_{t+1}, \ldots, i_m \in \{0, \ldots, s\}$$

such that

$$a(i_1, \ldots, i_{t-1}, j, i_{t+1}, \ldots, i_m) \neq 0.$$

The set of all polynomials which depend on all x_t^j ($t = 1, \ldots, m; j = 1, \ldots, s$) we denote by $K_s[x_1, \ldots, x_m]$.

Corollary 2
(i) If $f \in K_1[x_1, \ldots, x_m]$, then

$$\log_2 |T(f)| = m \tag{24}$$

(ii) If $f \in K_2[x_1, \ldots, x_m]$, then

$$\log_2 |T(f)| = \sum_{t=1}^{m} \log_2 \langle n_t \rangle \tag{25}$$

(iii) If $f \in K_3[x_1, \ldots, x_m]$, then

$$\log_2 |T(f)| = m + \sum_{t=1}^{m} \log_2 \langle n-1 \rangle \tag{26}$$

Proof
Note that, if $f \in K_s[x_1, \ldots, x_m]$, then

$$\hat{f}(\omega_1, \ldots, \omega_m) \neq 0 \quad \text{iff} \quad \|\omega_t\| \leqslant s \; (t = 1, \ldots, m)$$

Because

$$|T(f)| = |V^{\perp}| = 2^n \sum_{t=1}^{m} |V(n_t, s+1)|^{-1} \tag{27}$$

$$\left(n = \sum_{t=1}^{m} n_t\right)$$

and[9]

$$\log_2 |V(n_t, 2)| = n_t - 1 \tag{28}$$

$$\log_2 |V(n_t, 3)| = n_t - \log_2 \langle n_t \rangle \tag{29}$$

$$\log_2 |V(n_t, 4)| = n_t - \log_2 \langle n_t - 1 \rangle - 1 \tag{30}$$

we have eqns. 24–26 from eqns. 27–30.

For polynomials of one argument $f \in K_s[x]$ ($x \in \{0, \ldots, 2^n - 1\}$) using the construction of Bose–Chaudhuri codes[9] we have the following upper bound for the check complexity

$$\log_2 |T(f)| \leqslant \log_2 |V^{\perp}(n, s+1)| \leqslant \begin{cases} \alpha \log_2 \langle n \rangle, & \text{if} \quad s = 2\alpha \\ \alpha \log_2 \langle n-1 \rangle + 1, & \text{if} \quad s = 2\alpha + 1 \end{cases} \tag{31}$$

Check complexities $|T(f)|$ and right-hand constants d_f for polynomials $f \in K_s[x]$ of one argument are given in Table 1 (here, B_ν stands for Bernoulli numbers).

By the proof completely analogous to the proof of theorem 2 one may obtain the following result:

Theorem 3
Let

$$f(x^{(0)}, \ldots, x^{(n-1)}) = a_0 + \sum_{i_1=0}^{n-1} a(i_1) x^{(i_1)} + \sum_{0 \leqslant i_1 < i_2 \leqslant n-1} a(i_1, i_2) x^{(i_1)} x^{(i_2)} + \ldots + \sum_{0 \leqslant i_1 < \ldots < i_s \leqslant n-1} a(i_1, \ldots, i_s) x^{(i_1)} \ldots x^{(i_s)} \tag{32}$$

$(x^{(i)} \in \{0, 1\};\ i = 0, \ldots, n-1;\ s < n)$.

Then:
(i) for every $x = (x^{(0)}, \ldots, x^{(n-1)})$

$$\sum_{\tau \in V^{\perp}(n, s+1)} f(x \oplus \tau) = |V^{\perp}(n, s+1)| 2^{-s} \left\{ 2^s a_0 + 2^{s-1} \sum_{i_1=0}^{n-1} a(i_1) + \ldots + \sum_{0 \leqslant i_1 < \ldots < i_s \leqslant n-1} a(i_1, \ldots, i_s) \right\} \tag{33}$$

(ii) If $a_0 \neq 0$, $a(i_1) \neq 0, \ldots, a(i_1, \ldots, i_s) \neq 0$ for all $i_1, \ldots, i_s$ then the check set $V^{\perp}(n, s+1)$ is minimal, i.e. $T(f) = V^{\perp}(n, s+1)$.

We note that checks constructed by theorem 3 for polynomials of one argument coincide with checks constructed by theorem 2, but, for polynomials of a small degree and of a large number of arguments, checks constructed by theorem 3 may be simpler. This may be the case when we are dealing with matrix computations.

For example, let $F(X) = AX$ where A, X are matrices of dimensions $(k \times m)$ and $(m \times r)$, and $X_{ij} \in \{0, \ldots, 2^n - 1\}$. Then, every $(F(X))_{ij}$ may be considered as a linear function of mn binary arguments, and we have by theorem 3

$$F(X) + F(\bar{X}) = (2^n - 1)A \tag{34}$$

where $\bar{X}_{ij} = X_{ij} \oplus (1, 1, \ldots, 1)$.

By theorem 3, we may also construct checks for any polynomial $f(\|x\|)$, where

$$\|x\| = \sum_{i=0}^{n-1} x^{(i)}$$

Table 1 Parameters of the optimal checks for polynomials of one argument

Degree of a polynomial s	Check complexity $\|T(f)\|$	Right-hand constant d_f
1	2	$2a(0) + a(1)(2^n - 1)$
2	$\langle n \rangle$	$\langle n \rangle (a(0) + \frac{1}{2}a(1)(2^n - 1) + \frac{1}{6}a(2)(2^n - 1)(2^{n+1} - 1))$
③	$2\langle n-1 \rangle$	$2\langle n-1 \rangle (a(0) + \frac{1}{2}a(1)(2^n - 1) + \frac{1}{6}a(2)(2^n - 1)(2^{n+1} - 1) + \frac{1}{4}a(3)2^n(2^n - 1)^2)$
s	$2^n \|V(n, s+1)\|^{-1}$	$\|V(n, s+1)\|^{-1} \sum_{i=0}^{s} a(i) \frac{1}{i+1} \sum_{\nu=0}^{i} \binom{i+1}{\nu} 2^{(i+1-\nu)n} B_\nu$

is the Hamming weight of x.

We note that theorems 2 and 3 provide us with multiplicative checks for exponential functions.

Corollary 3
Let

$$f_i \in K_{s_1,\ldots,s_m}[x_1,\ldots,x_m] \quad (i = 1,\ldots,r)$$

and

$$\varphi(x_1,\ldots,x_m) = \sum_{i=1}^{r} a_f{}^{i(x_1,\ldots,x_m)}$$

$$(x_t \in \{0,\ldots,2^{n_t}-1) \qquad (35)$$

Then, for every $x_1,\ldots,x_m$,

$$\prod_{(\tau_1,\ldots,\tau_m)\in V^\perp} \varphi(x_1 \oplus \tau_1,\ldots,x_m \oplus \tau_m) = d_\varphi \qquad (36)$$

where

$$d_\varphi = \left\{ \prod_{x_1,\ldots,x_m} \varphi(x_1,\ldots,x_m) \right\}^{\lambda}$$

$$V^\perp \triangleq \prod_{t=1}^{m} V^\perp(n_t, s_t+1)$$

$$\lambda = \prod_{t=1}^{m} |V(n_t, s_t+1)|^{-1}$$

Proof
From eqn. 35 we have

$$\log \varphi(x_1,\ldots,x_m) = \sum_{i=1}^{r} f_i(x_1,\ldots,x_m) \times$$

$$\log a_i \in K_{s_1,\ldots,s_m}[x_1,\ldots,x_m] \qquad (37)$$

Then, by theorem 2,

$$\sum_{(\tau_1,\ldots,\tau_m)\in V^\perp} \log \varphi(x_1 \oplus \tau_1,\ldots,x_m \oplus \tau_m) =$$

$$\lambda \sum_{x_1,\ldots,x_m} \log \varphi(x_1,\ldots,x_m) \qquad (38)$$

and eqns. 36 and 37 follow from eqn. 38.

The network implementation of a multiplicative check (eqns. 36 and 37) may be obtained from the network implementations of an additive check (eqns. 11 and 12) (see Figs. 1, 2 in the following Section) by the replacement of adder accumulations by multiplier accumulators.

So far we have supposed that every argument x_t $(t = 1,\ldots,m)$ of the polynomial is represented in the binary form

$$x_t = (x_t^{(0)},\ldots,x_t^{(n_t-1)}) \; (x_t^{(l)} \in \{0,1\}).$$

In the case where x_t is represented in the q_t-ary form $(q_t \geqslant 2, t = 1,\ldots,m)$ all the previous results remain valid, but the check set

$$\sum_{t=1}^{m} V^\perp(n_t, s_t+1)$$

must be replaced by the set

$$\prod_{t=1}^{m} V_{q_t}^\perp(n_t, s_t+1),$$

where $V_{q_t}(n_t, s_t+1)$ is the maximal linear code in n_t-dimensional space of q_t-ary vectors with Hamming distance s_t+1 [9] and

$$V_{q_t}^\perp(n_t, s_t+1) = \left\{ (y^{(0)},\ldots,y^{(n_t-1)}) \,\middle|\, \bigoplus_{i=0}^{n_t-1} x^{(i)}y^{(i)} = 0 \right.$$
$$\left. (x^{(0)},\ldots,x^{(n_t-1)}) \in V_{q_t}(n_t, s_t+1) \right\}$$

where $x^{(i)}, y^{(i)} \in \{0,\ldots,q_t-1\}$, and the symbol $\oplus$ stands for mod q_t addition. (For the computations in the finite field $GF(p)$ we need the additional requirement that the least common multiple of $q_1,\ldots,q_m$ will be a divisor of $p-1$, because only in this case do we have convolution theorem eqn. 10 for the corresponding Fourier transform[2]).

4 Network implementation of checks for polynomial computations

Error-detecting capability
We shall consider now the problem of the network implementation of checks constructed in Section 3.

Let us begin with the example of the cubic polynomial of one argument $f(x) = x^3 - 95x^2 - 52x - 60$, when $x \in \{0,\ldots,127\}$, x is represented in the binary form $(n = 7, s = 3)$, and all the computations are carried out in the field of real numbers. Then $V(7,4)$ and $V^\perp(7,4)$ may be chosen as linear spaces over $GF(2)$ with the bases

$$\begin{pmatrix} 100 & 0111 \\ 010 & 1011 \\ 001 & 1101 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 011 & 1000 \\ 101 & 0100 \\ 110 & 0010 \\ 111 & 0001 \end{pmatrix}$$

respectively, and

$$(\tau^{(0)},\ldots,\tau^{(6)}) \in V^\perp(7,4) \quad \text{iff} \quad \tau^{(0)} = \tau^{(4)} \oplus \tau^{(5)} \oplus \tau^{(6)}$$
$$\tau^{(1)} = \tau^{(3)} \oplus \tau^{(5)} \oplus \tau^{(6)} \quad \text{and} \quad \tau^{(2)} = \tau^{(3)} \oplus \tau^{(4)} \oplus \tau^{(6)}.$$

Thus, by theorem 1 and Table 1, we have the following check for $f(x) = x^3 - 95x^2 - 52x - 60$:

$$\sum_{\tau\in V^\perp(7,4)} f(x \oplus \tau) = 56$$

for every $x \in \{0,\ldots,127\}$. The network implementation of this check is given in Fig. 1. There are three blocks in the check network Fig. 1: a counter, generating the information bits $\tau^{(3)}$, $\tau^{(4)}$, $\tau^{(5)}$ and $\tau^{(6)}$ of the code, $V^\perp(7,4)$, a coding network, generating the check bits $\tau^{(0)}$, $\tau^{(1)}$ and $\tau^{(2)}$ and a network for mod 2 summation of x and $\tau \in V^\perp(7,4)$.

Let us denote by $L(f)$ a minimal number of 2-input gates in a check network for f. The asymptotic behaviour of $L(f)$ may be described by the following theorem.

Theorem 4
If $n_t \to \infty$, $s_t < n_t$ $(t = 1,\ldots,m)$, then for every $f \in K_{s_1,\ldots,s_m}[x_1,\ldots,x_m]$

$$L(f) < \sum_{t=1}^{m} \left(\left[\frac{s_t}{2} \right] + 1 \right) n_t \qquad (39)$$

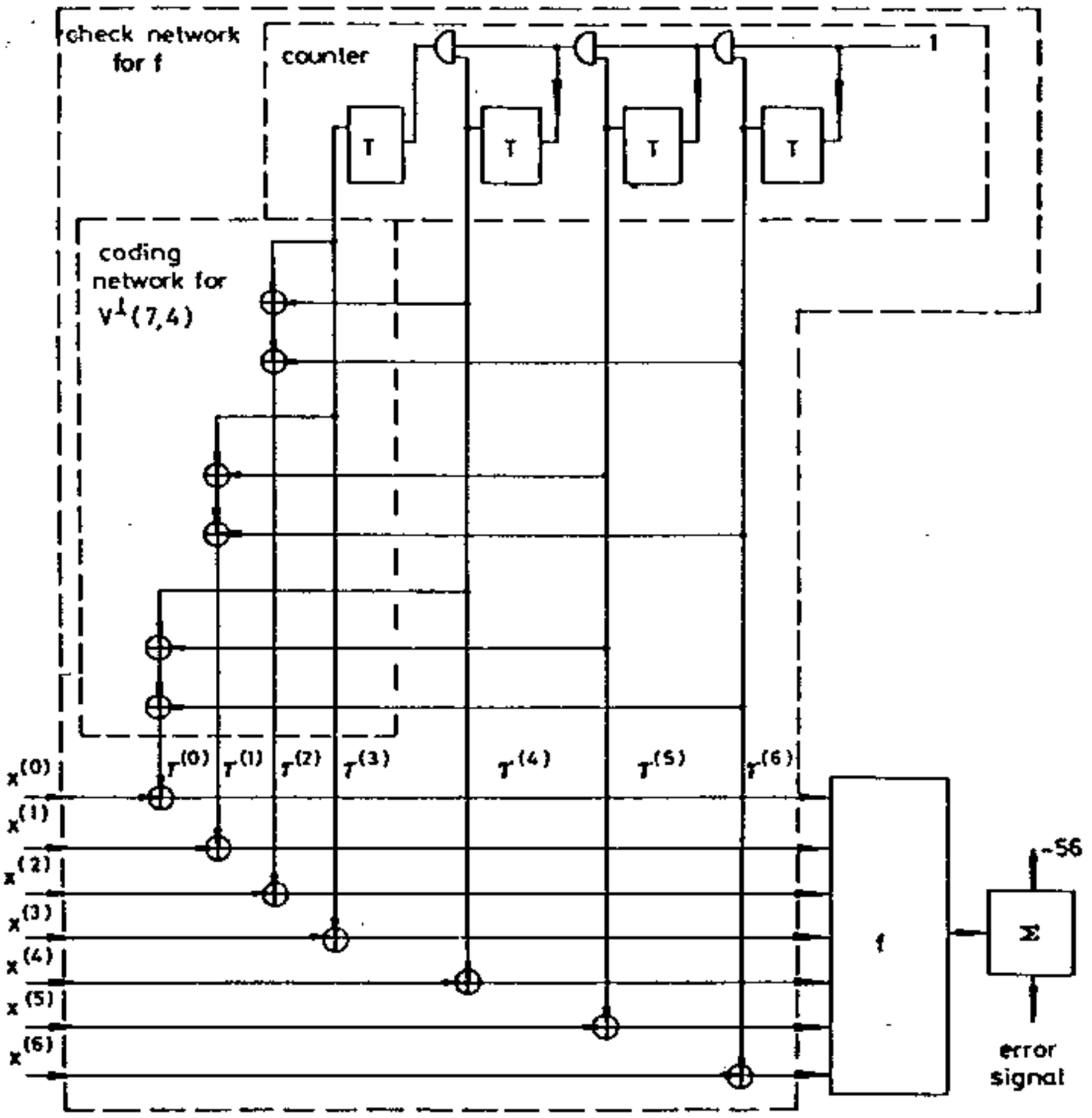

Fig. 1 *Network implementation of the check for polynomial* $f(x) = x^3 - 95x^2 - 52x - 60$

trigger

'AND' gate

adder mod 2

block computing polynomial *f*

adder-accumulator with initial state −56

where $[c]$ is the greatest integer $\leqslant c$,

$$a(n) \lesssim b(n) \quad \text{iff} \quad \lim_{n \to \infty} a(n)b^{-1}(n) \leqslant 1,$$

$$a(n) \sim b(n) \quad \text{iff} \quad a(n) \lesssim b(n) \quad \text{and} \quad b(n) \lesssim a(n).$$

Proof

Denote $\log_2 |V(n_t, s_t + 1)| = K_t$. Then $V^{\perp}(n, s_t + 1)$ is an $(n_t, n_t - K_t)$-linear code and a check network for every $f \in K_{s_1,\ldots,s_m}[x_1, \ldots, x_m]$ may be implemented by the network of Fig. 2. Here, every coding network t is a network linear over $GF(2)$ with $n_t - K_t$ inputs and K_t outputs. If $s_t > 1$ then[9] $n_t - K_t \to \infty$, as $n_t \to \infty$, and the coding network t may be realised by the method described in Reference 8 Section 3.2 with the complexity

$$L_t \lesssim \frac{(n_t - K_t)K_t}{\log_2 K_t} \tag{40}$$

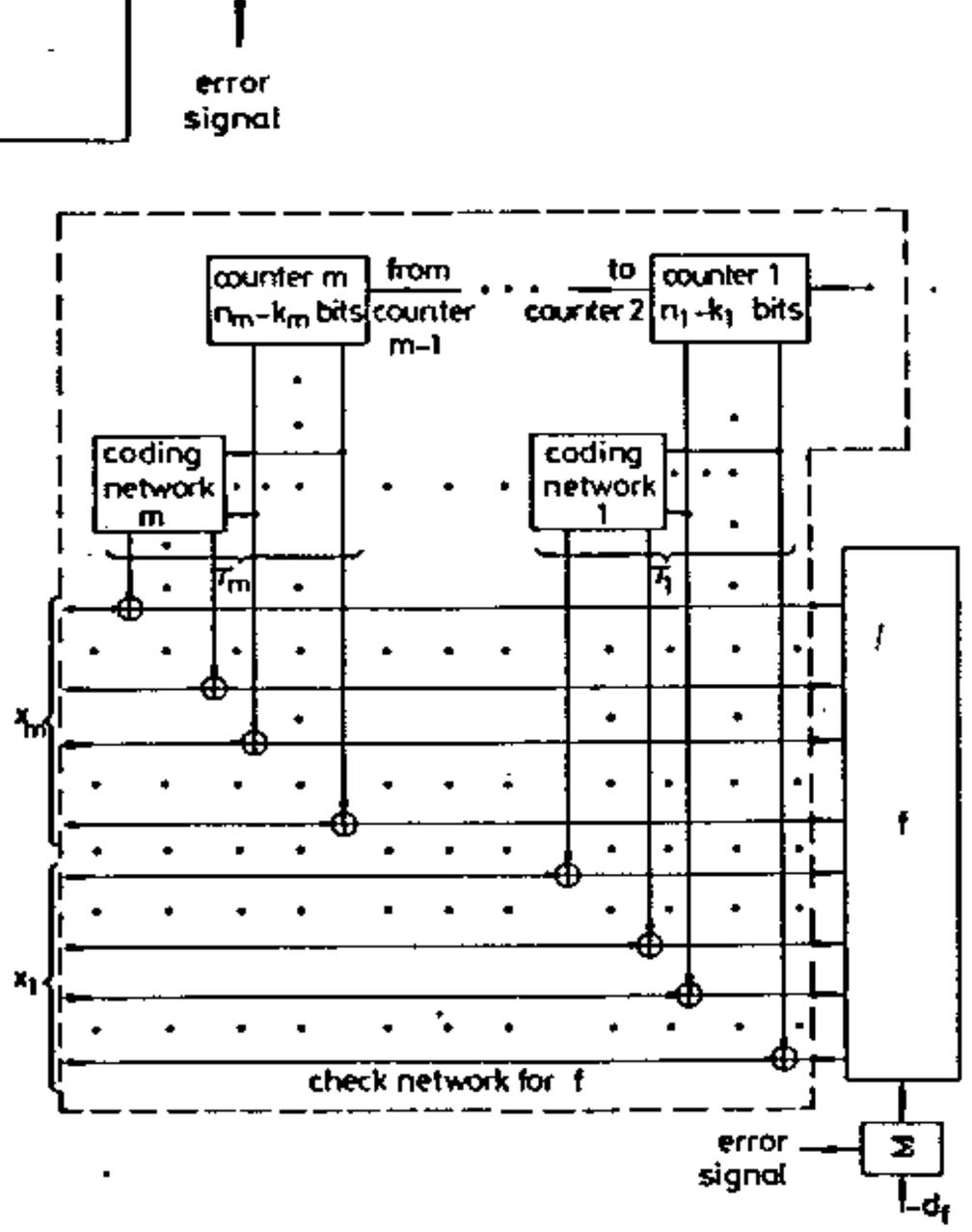

Fig. 2 *Network implementation of a check for polynomial f*

(If $s_t = 1$, then $n_t - K_t = 1$, $V^{\perp}(n_t, 2) = \{(0, \ldots, 0), (1, \ldots, 1)\}$ and $L_t = \text{constant}$). Because $n_t - K_t = \log_2 |V^{\perp}(n_t, s_t + 1)|$ and $n_t \sim K_t$ (Reference 9), we have from eqns. 31 and 40

$$L_t \leqslant \frac{\left[\frac{s_t}{2}\right] \log_2 n_t K_t}{\log_2 K_t} \sim \left[\frac{s_t}{2}\right] n_t$$

We note also that there exists C_t such that the complexity of the counter t with $n_t - K_t$ bits is no greater than

$$C_t(n_t - K_t) \lesssim C_t\left[\frac{s_t}{2}\right]\log_2 n_t.$$

Thus, we finally have

$$L(f) \lesssim \sum_{t=1}^{m}\left(\left[\frac{s_t}{2}\right]n_t + C_t\left[\frac{s_t}{2}\right]\log_2 n_t + n_t\right)$$

$$\sim \sum_{t=1}^{m}\left(\left[\frac{s_t}{2}\right]+1\right)n_t$$

Thus, for a polynomial f of degree S which is defined in 2^n points, the network complexity $L(f)$ of the optimal check increases at most linearly with increasing s or with increasing n.

We note also that upper bound eqn. 39 for complexity $L(f)$ may be decreased if $V(n_t, s_t + 1)$ and $V^{\perp}(n_t, s_t + 1)$ are cyclic codes, and coding networks in Fig. 2 are implemented by linear sequential networks,[9] but this results in an increase in the time required to implement the check.

We shall describe now an error-detecting capability of linear checks over $GP(2)$. We consider three classes of stuck-at errors: errors in coefficients; errors in arguments; and errors in values of the polynomial.

We assume also that every coefficient $a(i_1, \ldots, i_m)$ [argument x_t, value $f(x_1, \ldots, x_m)$ of the polynomial] is stored in a corresponding K-bit memory cell ($K = n_1 = \ldots = n_m$).

By an l-fold error in coefficients (arguments, values of the polynomial) we mean any error resulting in the replacement of l coefficients (arguments, values of the polynomial) by some constants $C_1, \ldots, C_l$ [binary representation of C_r $(r = 1, \ldots, l)$ contains K bits]. We denote the relative frequence of l-fold errors which cannot be detected by our check for the above three classes by $\eta_a(l)$, $\eta_x(l)$, $\eta_f(l)$, respectively.

The result of an error $e(x_1, \ldots, x_m)$ is to replace the given polynomial $f(x_1, \ldots, x_m)$ by $f(x_1, \ldots, x_m) + e(x_1, \ldots, x_m)$.

Corollary 4
For every

$$f(x_1, \ldots, x_m) = \sum_{t=1}^{m}\sum_{i_t=0}^{s_t} a(i_1, \ldots, i_m)\, x_1^{i_1} \ldots x_m^{i_m}$$

we have

$$\eta_a(l) \leqslant (1 - \delta_{l;1})(2^K - 1)^{-1}\left(0 < l \leqslant \prod_{t=1}^{m} s_t\right) \quad (41)$$

$$\eta_x(l) \leqslant 2^{-K}\max_r s_r \leqslant K2^{-K} \quad (0 < l \leqslant m) \quad (42)$$

$$\eta_f(l) \leqslant (1 - \delta_{l;1})(2^K - 1)^{-1} \quad (0 < l \leqslant 2^{Km}) \quad (43)$$

Proof
An l-fold error in coefficients $e_a(x_1, \ldots, x_m)$ [in arguments $e_x(x_1, \ldots, x_m)$, in values of f $e_f(x_1, \ldots, x_m)$] may be represented as

$$e_a(x_1, \ldots, x_m)$$

$$= \sum_{r=1}^{l}(C_r - a(i_{1,r}, \ldots, i_{m,r}))\, x_1^{i_{1,r}} \ldots x_m^{i_{m,r}}$$

$$(i_{t,r} \in \{0, \ldots, s_t\}, C_r \neq 0) \quad (44)$$

$$(e_x(x_1, \ldots, x_m) = f(x_{i_1} = C_1, \ldots, x_{i_l} = C_l) - f(x_1, \ldots, x_m)$$

$$(1 \leqslant i_1 < \ldots < i_l \leqslant m) \quad (45)$$

$f(x_{i_1} = C_1, \ldots, x_{i_l} = C_l)$ is the function of $m - l$ arguments obtained from $f(x_1, \ldots, x_m)$ by the replacement of x_{i_r} by C_r $(r = 1, \ldots, l)$

$$e_f(x_1, \ldots, x_m) = \sum_{r=1}^{l}(C_r - f(x_1, \ldots, x_m))$$

$$\delta_{x_1;i_{1,r}} \ldots \delta_{x_m;i_{m,r}}$$

$$(i_{t,r} \in \{0, \ldots, 2^K - 1\}; C_r \neq 0)) \quad (46)$$

If an error e is not detected, then by eqn. 11

$$\sum_{\tau \in V^{\perp}}(f(x \oplus \tau) + e(x \oplus \tau)) = d_f\,(x = (x_1, \ldots, x_m),$$

$$\tau = (\tau_1, \ldots, \tau_m))$$

and

$$\sum_{\tau \in V^{\perp}} e(x \oplus \tau) = \sum_{x_1, \ldots, x_m} e(x_1, \ldots, x_m) = 0 \quad (47)$$

We have from eqns. 44, 45 and 47, $\eta_a(1) = \eta_f(1) = 0$. Next, from eqns. 44–46, for every $C_1, \ldots, C_{r-1}, C_{r+1}, \ldots, C_l$ there exists at most one C_r such that eqn. 47 is satisfied for errors in coefficients a. ' values of the polynomial, and there exist at most s_r different values of C_r such that eqn. 47 is satisfied for errors in arguments. This completes the proof of corollary 4. Thus, for every polynomial $f(x_1, \ldots, x_m)$ which is defined in $2^n = 2^{Km}$ points, if K is big enough, then by a check network having a complexity of about $n([s/2] + 1)$ 2-input gates we may detect almost all stuck-at errors in coefficients, arguments and values of the polynomial.

5 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.*, 1977, C-26, pp. 208–218
2 KARPOVSKY, M.G., and TRACHTENBERG, E.A.: 'Fourier transform over finite groups for error detection and error correction in computation channels', *Inf. & Control* (to be published)
3 LECHNER, R.Y.: 'Harmonic analysis of switching functions', *in* MAKHOPADHYAY, A. (ed.): 'Recent developments in switching theory' (Academic Press, New York, 1971)
4 EDWARDS, C.R.: 'The application of the Rademacher-Walsh transform to Boolean function classification and threshold logic synthesis', *IEEE Trans.*, 1975, C-25, pp. 48–62
5 KARPOVSKY, M.G., and TRACHTENBERG, E.A.: 'Some optimization problems for convolution systems over finite groups, *Inf. & Control*, July 1977, 34, pp. 227–247
6 PICHLER, F.: 'On state space description of linear dyadic invariant systems'. Proceedings Symposium Walsh Functions, Washington, D.C. 1971, pp. 166–170
7 PEARL, J.: 'Optimal dyadic models of time-invariant systems', *IEEE Trans.*, 1975, C-24, pp. 598–603
8 KARPOVSKY, M.G.: 'Finite orthogonal series in the design of digital devices' (John Wiley, New York, 1976)
9 PETERSON, W.W.: 'Error-correcting codes' (Cambridge, Mass., MIT Press, 1961)