ERROR DETECTION IN DIGITAL DEVICES AND COMPUTER PROGRAMS WITH THE AID OF LINEAR RECURRENT EQUATIONS OVER FINITE COMMUTATIVE GROUPS

M. G. Karpovsky


Copyright © 1977 by The Institute of Electrical and Electronics Engineers, Inc.
Printed in U.S.A. Annals No. 703CX003
Error Detection in Digital Devices and Computer Programs with the Aid of Linear Recurrent Equations Over Finite Commutative Groups

MARK G. KARPOVSKY

Abstract—A method is proposed for error detection in digital devices and computer programs calculating the values of functions \( f(x) \), where \( x \in G \) and \( G \) is a finite commutative group. For the case of network implementation of the method, "errors" are catastrophic structural failures; for the case of program implementation, they are errors in the text of the program.

The method is based on finding, given the function, a linear "recurrent equation over \( G \)" with coefficients 0 or 1, of which \( f \) is a solution. The verification whether this is indeed the case constitutes an error detection method.

Implementation of the method requires only the operations of summation, delay, and the group operation in \( G \).

The equation whose solution is the given function \( f \) will be sought using methods of abstract harmonic analysis on the group \( G \).

Index Terms—Error detection for digital devices and computer programs, error detection tests for digital devices, characters of commutative groups, Fourier transform over finite commutative groups, spectral and autocorrelation functions over finite commutative groups, fast Hadamard-Walsh transform.

I. INTRODUCTION

SUPPOSE given a digital device or program for calculation of a function \( f: G \to C \), where \( G \) is a finite commutative group and \( C \) is the field of complex numbers. (Examples of such devices are the blocks of the arithmetic unit of a computer, networks whose operation is described by two- or many-valued logical functions, devices operating in systems of residue classes, etc.) To detect errors, one can construct another device or program calculating the same \( f(x) \); errors are detected by comparing the results of the calculations ("system redundancy method"). This method is widely used but is, generally speaking, highly uneconomical.

In this paper we propose another method of error detection, according to which, given the function \( f \), one determines a linear recurrent equation over \( G \):

\[
\sum_{q \in G} a(q)f(x^{*q^{-1}}) = \Phi(x)
\]

(1)

(* denotes the group operation in \( G \), \( q^{-1} \in G \) is the inverse of \( q \) and checks the validity of (1) for given \( x \).

Note that the meaning of the term "errors" depends on the context; errors in a digital device are catastrophic stable structural failures, and in programs they are errors in the texts of the programs. To simplify the error detection process, we shall consider the case \( a(q) \in \{0,1\} \).

The proposed error detection method will not depend on the specific features of implementation of the device or program for calculating the function \( f \); moreover, it will be universal in the sense that for any \( f: G \to C \) there exists an equation (1) with \( a(q) \in \{0,1\} \) and a "fairly simple" function \( \Phi(x) \) (for example, \( \Phi(x) = \text{const} \); see Section II, Theorem 1). It will also be seen that the above-mentioned "system redundancy method" is a special case of our method.

To search for the optimal checking equation (1) and to estimate the complexity of the equation we shall use methods of abstract harmonic analysis on \( G \). The advantages and the limitations of this technique will be discussed in Section V.

Related questions, concerning the analysis, synthesis, and optimization of digital devices by methods of abstract harmonic analysis and generalized Fourier transforms, were dealt with in [1]-[5].

II. ERROR DETECTION BY LINEAR HOMOGENEOUS EQUATIONS OVER A GROUP

A. Let \( f: G \to C \), where \( G \) is a finite commutative group and \( C \) the field of complex numbers; we denote the elements of \( G \) by \( 0, \cdots, q - 1 \) (\( q \) is the order of \( G \)); 0 is the identity element of \( G \); 

\[
\Phi(x) = f(x) + f(x^{*q_1^{-1}}) + \cdots + f(x^{*q_{L-1}^{-1}}) = d,
\]

\( q_1, \cdots, q_{L-1} \in G \), \( d \in C \).

(2)

Equation (2) generates a simple method for detection of errors in the calculation of \( f(x) \). A network interpretation of this method is illustrated in Fig. 1.

In the network of Fig. 1, signals corresponding to \( x, x^{*q_1^{-1}}, \cdots, x^{*q_{L-1}^{-1}} \) are applied at constant intervals of time to the input of the network calculating \( f(x) \); a nonzero signal at the output of the adder-accumulator with initial state \( -d \) after \( L(f) \) elementary additions may be used as an error signal.

The complexity of the network of Fig. 1 and the time required for error detection for fixed \( x \) depend only on the
number \( L(f) \) in the checking equation (2) \((1 \leq L(f) \leq g - 1)\).

B. We now consider the problem of finding a linear equation (2), given \( f \).

For the solution of this problem and the other problems considered below, we use methods of abstract harmonic analysis. We recall the main definitions.

A character of a finite commutative group \( G \) is defined as a homomorphism of \( G \) into the multiplicative group of complex numbers. The set of characters of \( G \) is a multiplicative group isomorphic to \( G \) [6, p. 367]. The character mapped onto an element \( \omega \in G \) under this isomorphism will be denoted by \( \chi_\omega(x)(x \in G) \).

Express \( G \) as a direct product of cyclic subgroups \( G = \prod_{\nu=0}^{g-1} G_\nu \). Let \( \xi \in G_\nu \) denote a generator of \( G_\nu \); \( \nu \); the order of \( G_\nu \) where \( p_\nu \) is power of prime. Then \[6, p. 367]\]

\[
\chi_\omega(x) = \exp \left( 2\pi i \sum_{\nu=0}^{p_\nu-1} \frac{\chi_\omega(x)}{p_\nu} \right)
\]

where
\[
x = \prod_{\nu=0}^{p_\nu-1} \xi \nu x_\nu, \quad \omega = \prod_{\nu=0}^{p_\nu-1} \xi \nu \omega_\nu,
\]
\[
x_\nu, \omega_\nu \in \{0, 1, \ldots, p_\nu - 1\}, \quad i = \sqrt{-1},
\]
\[
\xi \nu \cdot \xi ^* = \xi \nu ^* \cdot \xi ^* = \xi \nu ^* \xi ^* = 1.
\]

Due to the orthogonality and completeness of the set of characters \( \{ \chi_\omega \} \), one can use this set as a complete orthogonal basis in the space of functions mapping \( G \) into the field \( C \) of complex numbers. Thus if \( f: G \to C \), then

\[
f(x) = \sum_{\omega \in G} S_f(\omega) \chi_\omega(x)
\]

and

\[
S_f(\omega) = \frac{1}{g} \sum_{x \in G} f(x) \chi_\omega(x)
\]

where

\[
S_f(\omega) = \frac{1}{g} \sum_{x \in G} f(x) \chi_\omega(x)
\]

and \( \bar{\chi}_\omega \) is the character complex-conjugate to \( \chi_\omega \).

Equations (4) and (5) define the generalized Fourier transform \( F_G \) and the inverse generalized Fourier transform \( F_G^{-1} \) on \( G \); each function \( f \) is associated with its spectrum \( S_f \).

**Theorem 1:** For any \( f: G \to C \) there exist \( a: G \to [0,1] \) \((a \neq 0)\) and \( d \in C \) such that for every \( x \in G \)

\[
\sum_{\omega \in G} a(\omega)f(x \omega^{-1}) = d
\]

and

\[
\sum_{\omega \in G} a(\omega) = g/\bar{g}_a
\]

where \( g \) is the order of an arbitrary subgroup \( G_a \) of \( G \) such that if \( \omega \in G_a \), and \( \omega \neq 0 \) then \( S_f(\omega) = 0 \).

**Proof:** The left-hand side of (6) is the convolution over \( G \) of the functions \( a \) and \( f \). Spectrum of the convolution of two functions is equal to product of the spectra of the functions multiplied by the order of the group \( g \) ("convolution theorem"). Hence the required function \( a \) must be such that

\[
S_a(\omega) \cdot S_f(\omega) = \begin{cases} d/\bar{g}_a, & \text{if } \omega = 0 \text{ (where } S_a(\omega) \text{ spectrum)} \\ 0, & \text{if } \omega \neq 0 \end{cases}
\]

Let

\[
d = \frac{\sum_{x \in G} f(x)}{g},
\]

and

\[
S_a(\omega) = \begin{cases} 1/\bar{g}_a, & \text{if } \omega \in G_a, \\ 0, & \text{if } \omega \notin G_a. \end{cases}
\]

Then equality (8) (and hence also (6)) will follow from (9), (10), and by (3), (4), (10), for every \( q \in G \),

\[
a(q) = \sum_{\omega \in G} S_a(\omega) \chi_\omega(q) = 1/\bar{g}_a \sum_{\omega \in G_a} \chi_\omega(q). \quad (11)
\]

The function \( \chi_\omega(q), \omega \in G_a \) is a character of the subgroup \( G_a \). Hence, using (3) and (11), we obtain
\[ \sum_{\omega \in G_a} x_\omega(\omega) \in \{0, g_a\}; \quad a(0) = 1, a(q) \in \{0, 1\}. \]  

Thus if \( a(q) \) is defined by (11) and \( d \) is defined by (9), then \( a, f, \) and \( d \) satisfy (6) and \( a(q) \in \{0, 1\} \). We now prove (7). Consider a homomorphism of the group of characters of \( G \) onto the group of characters of \( G_a \). The kernel \( G_a \perp \) of this homomorphism is defined by the condition:

\[ q \in G_a \perp \text{iff } x_\omega(\omega) = 1 \text{ for all } \omega \in G_a. \]

or

\[ q \in G_a \perp \text{iff } a(q) = 1. \]

Since \( G_a \perp \) is a subgroup of \( G \) (this subgroup is isomorphic to quotient group \( G/G_a \)), and the order of \( G_a \perp \) is \( g/g_a \), it follows that

\[ \sum_{q \in G} a(q) = g/g_a. \]

C. The proof of Theorem 1 generates a simple method for constructing a checking equation (2) or (6) for the given function \( f \). This method reduces to the following operations:

1. Compute the generalized Fourier spectrum \( S_f \) of \( f \) by (3), (5).

2. Construct a subgroup \( G_a \) of \( G \) such that if \( \omega \in G_a \) and \( \omega \neq 0 \), then \( S_f(\omega) = 0 \).

3. Construct \( a: G \rightarrow \{0, 1\} \) by (11), and \( d \) by (9).

The spectrum \( S_f \) may be calculated by using the highly effective algorithm of the fast Fourier transform on the group \( G \) [7].

We now illustrate the error detecting technique described above for binary adders and multipliers.

**Example 1:** Consider error detection in an \( n \)-bit binary adder using Theorem 1. Let

\[ X = \sum_{p=0}^{n-1} x_p 2^{n-1-p} \quad Y = \sum_{p=0}^{n-1} y_p 2^{n-1-p}, \]

and \( f(x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1}) = X + Y \). Then \( (x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1}) \in G_{2^{2n}} \) where \( G_{2^{2n}} \) is the group of binary \( 2n \)-vectors with respect to componentwise addition mod 2.

According to (3), the characters of \( G_{2^{2n}} \) are the Walsh functions:

\[
W_\omega(x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1}) = (-1)^{\sum_{p=0}^{n-1} \omega_p x_p + \omega_p y_p},
\]

\((\omega_p \in \{0, 1\}; r = 0, \ldots, 2n - 1). \quad (13)\)

Letting \( \omega = (0, \ldots, 0, 1, 0, \ldots, 0) (r = 0, \ldots, 2n - 1) \), we put

\[ R_{p+1} = W_\omega(x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1}) \quad (14)\]

(These functions are known as the Rademacher functions [2], [5].)

Then by (13), (14),

\[ x_p = 0.5(1 - R_{p+1}) \quad y_p = 0.5(1 - R_{n+p+1}), \quad p = 0, \ldots, n - 1 \]

and

\[ X + Y = 2^n - 1 - \sum_{p=0}^{n-1} (R_{p+1} + R_{n+p+1}) 2^{n-2} - p. \quad (16)\]

Consequently, in view of (4), (14),

\[ S_f(\omega) = S_{X+Y}(\omega) = \begin{cases} 2^n - 1, & \text{if } \omega = (0, \ldots, 0); \\ -2^{n-2} - p, & \text{if } \omega = (0, \ldots, 0, 1, 0, \ldots, 0), \\ \frac{2^n}{2n} - p - 1, & \omega = (0, \ldots, 0, 1, 0, \ldots, 0); (p = \leq -1); \\ 0, & \text{otherwise} \end{cases} \]

In accordance with (17), we get

\[ G_a = \left\{ \omega \left| \sum_{r=0}^{2n-1} \omega_p = 2k \quad (k = 0, 1, \ldots, n) \right. \right\}. \]

Then

\[ G_a = 2^{2n-1}, G_a \perp = \{0, \ldots, 0\}, (1, \ldots, 1) \]

\[
\frac{2n}{2n} \quad \frac{2n}{2n}
\]

Since

\[ \sum_{X,Y} f(x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1}) = 2^n (2^n - 1) \]

it follows from (9) that \( d = 2(2^n - 1) \), and we have the following checking equation for binary adders:

\[ f(x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1}) + f(x_0 \oplus 1, \ldots, x_{n-1} \oplus 1, y_0 \oplus 1, \ldots, y_{n-1} \oplus 1) = 2(2^n - 1) (\text{mod } 2). \quad (18)\]

(Henceforth, the symbol \( \oplus \) with "(mod 2)" to the right of the equation stands for addition mod 2.)

**Example 2:** We now consider error detection in a binary multiplier. Let

\[ X = \sum_{p=0}^{n-1} x_p 2^{n-1-p} \quad Y = \sum_{p=0}^{n-1} y_p 2^{n-1-p}, \]

and \( f(x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1}) \in G_{2^{2n}} \) where \( G_{2^{2n}} \) is the group of binary \( 2n \)-vectors with respect to componentwise addition mod 2.

Then, in view of (14), (15),

\[ X \cdot Y = 0.25(2^n - 1)^2 - 0.5(2^n - 1) \]

\[ - \sum_{p=0}^{n-1} (R_{p+1} + R_{n+p+1}) 2^{n-2} - p \]

\[ + \sum_{p_1 
eq p_2} R_{p_1+1} \cdot R_{n+p_2+1} \cdot 2^{n-4-p_1-p_2}. \quad (19)\]

Now it follows from (13), (14) that

\[ R_{p+1} \cdot R_{n+p_2+1} = W_\omega(x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1}) \]

where
and so, in view of (4), (19), we have

\[
S_f(\omega) = S_X \cdot r(\omega) = \begin{cases} 
0.25 (2^n - 1)^2, & \text{if } \omega = (0, \ldots, 0); \\
0.5 (2^n - 1) 2^{n-k}, & \text{if } \omega = (0, \ldots, 0, 1, 0, \ldots, 0, 0), \\
\frac{2^{n-k} + p_1 p_2}{\frac{n}{p} + 1}, & \text{if } \omega = (0, \ldots, 0, 1, 0, \ldots, 0, 1, 0), \\
0, & \text{otherwise}, \quad (p_1, p_2, p_3 = n - 1). 
\end{cases}
\]

In accordance with (20), we set

\[
G_a = \left\{ \omega \mid \sum_{p=0}^{n-1} \omega_p = 2K_1, \quad \sum_{p=0}^{n-1} \omega_n + \rho = 2K_2, \quad (k_1, k_2 = 0, \ldots, n) \right\}. 
\]

Then

\[
G_a = 2^{n-2}, \quad G_a^{-1} = \left\{ (0, \ldots, 0), (0, \ldots, 0, 1, \ldots, 1), (1, \ldots, 1, 0, \ldots, 0), (1, \ldots, 1) \right\}. 
\]

Since in this case

\[
\sum_{x \in X \cdot Y} f(x_0, \ldots, x_{n-1} y_0, \ldots, y_{n-1}) = 2^{n-2}(2^n - 1)^2 
\]

it follows from (9) that \( d = (2^n - 1)^2 \) and we have the following checking equation for binary multipliers:

\[
f(x_0, \ldots, x_{n-1} y_0, \ldots, y_{n-1}) = \\
f(x_0, \ldots, x_{n-1} y_0 \oplus 1, \ldots, y_{n-1} \oplus 1) + \\
f(x_0 \oplus 1, \ldots, x_{n-1} \oplus 1, y_0, \ldots, y_{n-1}) + \\
f(x_0 \oplus 1, \ldots, x_{n-1} \oplus 1, y_0 \oplus 1, \ldots, y_{n-1} \oplus 1) = (2^n - 1)^2 \text{mod } 2. 
\]

To end this section, we consider the special case of Theorem 1 for \( d = 0. \)

**Corollary 1:** For any \( f: G \to C \) such that \( \sum_{x \in G} f(x) = 0, \) there exists \( \alpha: G \to \{0, 1\} (\alpha \neq 0) \) with the property: for every \( x \in G, \)

\[
\sum_{q \in G} a(q)f(x \cdot q^{-1}) = 0 
\]

and

\[
\sum_{q \in G} a(q) = g_a 
\]

where \( g_a \) is the order of an arbitrary subgroup \( G_a \) of \( G \) such that if \( \omega \in G_a \) then \( S_f(\omega) = 0. \)

**III. ERROR DETECTION BY LINEAR NONHOMOGENEOUS EQUATIONS OVER A GROUP**

A. Suppose that besides the device or program for calculation of the function \( f: G \to C \) we have another (independent) device or program for calculation of some function \( \Phi: G \to C, \) such that the calculation of \( \Phi(x) \) for all \( x \in G \) is "simple" in the sense that the probability of errors in \( \Phi(x) \) is small. (An example is \( \Phi(x) = x_r (r = 0, 1, \ldots, m - 1), \) where \( x_r \) is the \( r \)th component of the vector \( x. \))

We shall use \( \Phi(x) \) to detect errors in the calculation of \( f(x). \)

Let there exist \( q_1, \ldots, q_{L(f)} S_1, \ldots, S_{L(\Phi)} \subset G \) and \( d \subset C \) such that, for given \( f \) and \( \Phi, \) and for every \( x \in G, \)

\[
f(x) + f(x \cdot q_1^{-1}) + \ldots + f(x \cdot q_{L(f)}^{-1}) = \Phi(x) + \Phi(x \cdot S_1^{-1}) + \ldots + \Phi(x \cdot S_{L(\Phi)}^{-1}) + d. 
\]

Equation (22) is the general form of a linear nonhomogeneous equation with coefficients 0 or 1 over the group \( G; \) it generates another method of error detection, illustrated (for the case \( L(\Phi) < L(f) \)) in Fig. 2. In this case detection of an error for given \( x \) requires max \( L(f), L(\Phi) \) elementary additions (errors in the calculation of \( \Phi(x) \) may also be detected by the block diagram of Fig. 2 if there are no errors in the calculation of \( f(x). \)) The use of the block diagram of Fig. 2 instead of that of Fig. 1 may result in a significant decrease in \( L(f), \) so that nonhomogeneous checking equations may be more effective for error detection (see Example 3).

B. Using the notation of (22), we denote

\[
a(q) = \begin{cases} 
1, & \text{if } q \in \{0, q_1, \ldots, q_{L(f)}\}; \\
0, & \text{otherwise}; 
\end{cases} 
\]

\[
b(q) = \begin{cases} 
1, & \text{if } q \in \{0, S_1, \ldots, S_{L(\Phi)}\}; \\
0, & \text{otherwise}. 
\end{cases} 
\]

Then by (12) and (23)

\[
\sum_{q \in G} a(q)f(x \cdot q^{-1}) = \sum_{q \in G} b(q) \Phi(x \cdot q^{-1}) + d. 
\]

Set

\[
\Omega_f = \{ \omega | S_f(\omega) = 0 \}, \quad \Omega_\Phi = \{ \omega | S_\Phi(\omega) = 0 \}, 
\]

\[
\Omega_{f, \Phi}(\alpha) = \{ \omega | S_\Phi(\omega) = \alpha \}. 
\]

**Theorem 2:** For any two functions \( f, \Phi: G \to C, \) there exist \( \alpha: G \to \{0, 1\}, \) \( b: G \to \{0, 1\} (\alpha \neq 0, b \neq 0), \) and \( d \subset C \) such that \( a, b, f, \Phi, d \) satisfy (24), and if for some \( \alpha \) the sets \( \Omega_f \cup \Omega_{f, \Phi}(\alpha) \cup \{0\} \) and \( \Omega_\Phi \cup \Omega_{f, \Phi}(\alpha) \cup \{0\} \) contain subgroups \( G_a(\alpha) \) and \( G_b(\alpha) \) of \( G, \) of orders \( g_a(\alpha) \) and \( g_b(\alpha) \) respectively, and \( G_a(\alpha) \cap \Omega_{f, \Phi}(\alpha) = G_b(\alpha) \cap \Omega_{f, \Phi}(\alpha), \) then
\[ \sum_{q \in G} a(q) + \sum_{q \in G} b(q) = \frac{g}{g_a(\alpha)} \left( 1 + \frac{1}{\alpha} \right). \]  

(25)

**Proof:** By (24) and the convolution theorem, we have

\[ S_a(\omega)S_f(\omega) = \begin{cases} 
S_b(0) \cdot S_F(0) + \frac{d}{g^t} & \text{if } \omega = 0; \\
S_b(\omega) \cdot S_F(\omega) & \text{if } \omega \neq 0.
\end{cases} \]  

(26)

\((S_a, S_b, S_f, S_F)\) denote the spectra of \(a, b, f, F.\)

Assume that for some \(\alpha\) all the conditions of Theorem 2 are satisfied. Let

\[ S_a(\omega) = \begin{cases} 
\frac{1}{g_a(\alpha)}, & \text{if } \omega \in G_a(\alpha); \\
0, & \text{if } \omega \notin G_a(\alpha);
\end{cases} \]  

(27)

\[ S_b(\omega) = \begin{cases} 
\frac{1}{g_b(\alpha)}, & \text{if } \omega \in G_b(\alpha); \\
0, & \text{if } \omega \notin G_b(\alpha);
\end{cases} \]  

and

\[ d = \frac{1}{g_a(\alpha)} \sum_{x \in G} f(x) - \frac{1}{g_b(\alpha)} \sum_{x \in G} \Phi(x). \]  

(28)

We now prove that (26) (and hence also (24)) will follow from (27) and (28). We consider all possible cases.

1) Let \(\omega = 0\); then by (27), (28)

\[ S_a(0) \cdot S_f(0) = \frac{1}{g_a(\alpha)} \cdot \frac{1}{g_b(\alpha)} \sum_{x \in G} f(x) \]

\[ = \frac{1}{g} \left( \frac{1}{g_b(\alpha)} \sum_{x \in G} \Phi(x) + d \right) = S_b(0) \cdot S_F(0) + g^{-1}d. \]

2) Let \(\omega \neq 0\) and \(\omega \in G_a(\alpha) \cap G_b(\alpha).\)

a) If \(\omega \in \Omega_f\), then \(\omega \in \Omega_\Phi,\) since for every \(\alpha, \Omega_f \cap \Omega_{\Phi,\Phi}(\alpha) = \Omega_\Phi \cap \Omega_{\Phi,\Phi}(\alpha) = \emptyset\) (the empty set), \(S_f(\omega) = S_\Phi(\omega) = 0\) and \(S_a(\omega) \cdot S_F(\omega) = S_b(\omega) \cdot S_\Phi(\omega) = 0.\)

b) If \(\omega \in \Omega_{\Phi,\Phi}(\alpha),\) then \(S_\Phi(\omega)/S_f(\omega) = \alpha\) and, since \(g_\Phi(\alpha) = \alpha \cdot g_a(\alpha), S_a(\omega) \cdot S_f(\omega) = (1/\alpha)S_f(\omega) = (1/\alpha)S_b(\omega)S_\Phi(\omega) = (1/\alpha)g_b(\alpha)S_b(\omega) \cdot S_\Phi(\omega).\)

3) Let \(\omega \in G_a(\alpha)\) and \(\omega \notin G_b(\alpha).\) Since \(G_a(\alpha) \cap \Omega_{\Phi,\Phi}(\alpha) = G_b(\alpha) \cap \Omega_{\Phi,\Phi}(\alpha),\) it follows that for every \(\omega \notin \Omega_{\Phi,\Phi}(\alpha),\) so that \(\omega \notin \Omega_f\) and \(S_f(\omega) = 0;\) but if \(\omega \notin G_b(\alpha),\) then \(S_b(\omega) = 0,\) and thus

\[ S_a(\omega) \cdot S_f(\omega) = S_b(\omega) \cdot S_\Phi(\omega) = 0. \]

(The case \(\omega \notin G_a(\alpha)\) and \(\omega \in G_b(\alpha)\) may be treated similarly.)

4) Let \(\omega \notin G_a(\alpha), \omega \notin G_b(\alpha).\) Then \(S_a(\omega) = S_b(\omega) = 0\) and \(S_a(\omega) \cdot S_f(\omega) = S_b(\omega) \cdot S_\Phi(\omega) = 0.\) Thus (26) follows from (27) and (28). Furthermore, if we consider the subgroups \(G_a(\alpha)\) and \(G_b(\alpha)\) as in the proof of Theorem 1 for \(G_a\), we see that \(a(q), b(q) \in [0,1]\) for every \(q \in G,\) and

\[ L(f) + 1 = \sum_{q \in G} a(q) = \frac{g}{g_a(\alpha)}, \]

\[ L(\Phi) + 1 = \sum_{q \in G} b(q) = \frac{g}{g_b(\alpha)} \]  

(29)

whence, using the fact that \(g(\alpha) = \alpha \cdot g_a(\alpha),\) we have (25).

**Example 3:** Let \(f\) be the function defined on the dyadic group \(G_2^3\) by Table I.

We take \(\Phi(x) = x_2\) (the spectra \(S_f, S_\Phi\) are shown in Table I). Then by Table I

\[ \Omega_{f,\Phi}(\alpha) = \begin{cases} 
\{(0,0,0),(0,0,1),\} & \text{if } \alpha = 2; \\
\emptyset & \text{if } \alpha \neq 2. 
\end{cases} \]

\[ \Omega_f = \{(0,1,0),(0,1,1),(1,0,0),(1,0,1),(1,1,0),(1,1,1)\}; \]

\[ G_a(2) = \Omega_f \cup \Omega_{f,\Phi}(2); \]

\[ G_b(2) = \Omega_\Phi \cup \Omega_{f,\Phi}(2) = G; \]

\[ G_a(2) \cap \Omega_{f,\Phi}(2) = G_b(2) \cap \Omega_{f,\Phi}(2) = \Omega_{f,\Phi}(2); \]

\[ g(2) = 4; g_b(2) = 8.\]  

The conditions of Theorem 2 are satisfied for \(\alpha = 2.\) The functions \(S_a, S_b, a, b\) are shown in Table I. Since it follows from (28) that \(d = 0,\) we have from Table I.
I the following nonhomogeneous checking equation:

\[ f(x_0, x_1, x_2) + f(x_0 \oplus 1, x_1, x_2) = x_2 \pmod{2} \]

(A block diagram of error detection for this example is shown in Fig. 3.) To find an optimal nonhomogeneous checking equation (24) for given \( f \) and \( \Phi \), i.e., an equation minimizing the complexity \( \Sigma_{q \in G} a(q) + \Sigma_{q \in G} b(q) \), one must consider all \( a \) such that \( a \) or \( a^{-1} \) is a divisor of the order \( g \) of the group \( G \).

C. We now consider the special case of the checking equation (24) in which \( a(q) = b(q) \) for all \( q \in G \). The block diagram Fig. 2 is now considerably simplified, and \( \Phi \) may be chosen as a function realized by some subnetwork implementing \( f \) (see Fig. 4).

Corollary 2: For any two functions \( f, \Phi : G \to C \), there exist \( a : G \to \{0, 1\} \) (\( a \neq 0 \)) and \( d \in C \) such that \( a, f, \Phi, d \) satisfy (24) with \( a(q) = b(q) \) for all \( q \in G \), and

\[ \sum_{q \in G} a(q) = \frac{\delta}{g_a(1)} \tag{30} \]

where \( g_a(1) \) is the order of an arbitrary subgroup \( G_a(1) \) contained in \( \Omega_f \cap \Omega_b \) \( \cup \Omega_f, a(1) \cup [0] \).

Proof: Corollary 2 follows from Theorem 2 with \( \alpha = 1 \) and \( G_a(1) = G_b(1) \). In this case \( g_a(1) = g_b(1) \) and, by (27), \( S_a = S_b, a = b \) and (30) follows from (29).

Note that the “system redundancy method” mentioned above is a special case of error detection method generated by Corollary 2. In this case \( f(x) = \Phi(x) \) for all \( x \in G \), \( \Omega_f \cap \Omega_b \cup \Omega_f, a(1) = G, d = 0, g_a(1) = g \) and \( \Sigma_{q \in G} a(q) = 1 \).

D. We now consider a usage of error detection methods generated by Theorems 1, 2 and Corollaries 1, 2 for finite automata. Let \( M_f = (X, Q, Y, q_0, \delta, \lambda) \) be the given finite automaton \( X = \{0, 1, \ldots, n_x - 1\} \) be the input alphabet; \( Q \) the set of internal states; \( Y \subseteq C \) the output alphabet; \( q_0 \in Q \) the initial state; \( \delta : X \times Q \to Q \) the transition function; \( \lambda : Q \to Y \) the output function); and \( x = (x_0, x_1, \ldots, x_{m-1}) \in X^m \). Set

\[ f(x) = \lambda(\delta(x_{m-1}, \ldots, \delta(x_1, \delta(x_0, q_0)))) \ldots) \tag{31} \]

The function \( f \) now may be considered as a function defined on the group \( G \) of the all vectors \((x_0, x_1, \ldots, x_{m-1}) \in X^m \) with the respect to componentwise modulo \( n_x \) addition and all the error detection methods of Theorems 1, 2 and Corollaries 1, 2 may be used for \( f \).
Example 4: Let the automaton $M_f$ be defined by the transition diagram Fig. 5(a) and suppose that a network implementing $M_f$ contains a subnetwork implementing automaton $M_+ \Phi$ defined by the transition diagram Fig. 5(b). Let us try to use $M_+ \Phi$ to detect errors in $M_f$ for $m = 3$.

The functions $f$ and $\Phi$ constructed for $M_f$ and $M_+ \Phi$ by (31) and the spectra $S_f$ and $S_\Phi$ of $f$ and $\Phi$ are shown in Table II. Then $\Omega_f = \emptyset$, $\Omega_\Phi = \emptyset$, $\Omega_{f+\Phi} = \{0,1,1\}, (1,0,0),(1,1,1)$. Let $G_a(1) = G_b(1) = (0,0,0),(0,1,1), (1,0,0),(1,1,1)$. Let $g_a(1) = g_b(1) = 4$. (The functions $S_a$, $S_b$, $a$, and $b$ are also shown in Table II.)

By (28), $d = 1/4 \left( \sum_{x \in G} f(x) - \sum_{x \in G} \Phi(x) \right) = 2$; thus, by Table II, and Corollary 2 we have the checking equation:

$$f(x_0,x_1,x_2) + f(x_0,x_1 \oplus 1,x_2 \oplus 1)$$
$$= \Phi(x_0,x_1,x_2) + \Phi(x_0,x_1 \oplus 1,x_2 \oplus 1) + 2 \pmod{2}.$$ 

IV. TESTS FOR ERROR DETECTION NETWORKS—RECOGNITION OF THE “SIMPLEST” FUNCTIONS

A. It follows from Theorem 1.2 and Corollaries 1.2 that the complexity of the error detection block diagrams (Figs. 1, 2, and 4) depends on the orders $g_a,g_a(a),g_a(1)$ of the selected subgroups $G_a,G_a(a),G_a(1)$. To minimize the complexity, therefore, one should choose subgroups of the maximal possible order. The measures $\sum_{x \in G} a(x)$ and $\sum_{x \in G} b(x)$ of complexity are always divisors of the order $g$ of the original group $G$.

For every block diagram of Figs. 1, 2, and 4 we can organize the filter test detection by applying some signals $x$ to the input of the block diagram. Any such signal will be called a filter test for the corresponding block diagram. We now consider a problem of finding the minimal set of these tests.

It follows from the results of the previous sections that, for a given checking equation, the set $G_a^{--}$ defined by the condition $a(q) = 1$ is a subgroup of $G$. Hence, for every $x \in G$, the set $\{x,x^q,\ldots,x^{q^{q-1}}\}$ generated by the test $x$ is the coset of $G_a^{--}$ in $G$. Thus any system of distinct coset representatives of $G_a^{--}$ in $G$ (and, in particular, the subgroup $G_a$) is a minimal set of tests.

For instance, for the block diagram of Fig. 3 (Example 3) $G_a^{--} = \{(0,0,0),(0,1,0),(1,0,0),(1,1,0)\}$, and the minimal set of tests may be chosen as $\{(0,0,0),(0,0,1),(0,1,1),(1,1,0)\}$.

The minimal number of tests is $g^2 \sum_{q \in G} a(q)$, and for a given $f$ the product of the complexity $\sum_{q \in G} a(q)$ of the block diagram and the minimal number of tests is always equal to the order $g$ of the group $G$ (irrespective of the choice of the subgroup $G_a$).

B. We now consider the class of “simplest” functions $f$ which satisfy the general equation (24) when $\Phi(x) = 0$ for all $x \in G$, $d = 0$, the number of nonzero terms on the left of (24) is two and $f: G \to R$ (where $R$ is the set of real numbers). Then for all $x \in G$

$$f(x) + a(q)(x^q - 1) = 0 \quad (32)$$

but in this case we shall assume that $a(q) \in \{-1,1\}$. The case $a(q) = -1$ is convenient if $f(x) = 0$ (or $f(x) < 0$) for all $x \in G$. (All functions of two and many-valued logic, for example, satisfy this condition.)

Our problem is, given $f$, to find (if possible) $q \in G (q \neq 0)$, and $a(q) \in \{-1,1\}$, such that $f(x,a(q))$ satisfy (32).

In contrast to the previous treatment, the method proposed for solution of this problem will involve not spectra but correlation functions on the group $G$.

We first construct the following system of characteristic functions $f_t: G \to \{-1,0,1\}$ ($t > 0$):

$$f_t(x) = \begin{cases} (-1)^{[\text{sign}(x) + 1]} & \text{if } |f(x)| = t; \\ 0, & \text{if } |f(x)| \neq t; \end{cases}$$

$$\text{sign } f(x) = \begin{cases} 1, & f(x) \geq 0; \\ 0, & f(x) < 0 \end{cases} \quad (33)$$

the corresponding system of autocorrelation functions $B_{f_t}$ on $G$:

$$B_{f_t}(r) = \sum_{x \in G} f_t(x)f_t(\text{x}^r) \quad (34)$$

and total autocorrelation function $B_{\Sigma}$:

$$B_{\Sigma}(r) = \sum_{t > 0} B_{f_t}(r). \quad (35)$$

(The properties and applications of these autocorrelation functions to the analysis and synthesis of digital devices were studied in [2], [4], [5].)

Theorem 5: A function $f: G \to R$ satisfies (32) for given $q \in G (q \neq 0)$, $a(q) \in \{-1,1\}$ iff

$$(-1)^{[\text{sign}(q) + 1]} B_{f_t}(q) = B_{\Sigma}(0). \quad (36)$$

Proof: Since for all $x \in G$, $f_t(x) \in \{-1,0,1\}$, we have for every $q,r \in G$, $a(q) \in \{-1,1\}$

$$(-1)^{[\text{sign}(q) + 1]} f_t(x)f_t(\text{x}^r) \leq f_t(x). \quad (37)$$

It follows now from (34), (35), and (37), in a view of $B_{\Sigma}(0) = 0$ that (36) is satisfied iff

$$f_t(x) = (-1)^{[\text{sign}(q) + 1]} f_t(x)(x^q - 1) \quad (38)$$

but in a view of (33) and $(-1)^{[\text{sign}(q) + 1]} = -a(q)$, the last condition is satisfied iff the condition (32) holds.

Thus the “simplest” functions, for which there exist $q \in G (q \neq 0)$ and $a(q) \in \{-1,1\}$ satisfy (32), may be recognized by using autocorrelation functions on the group $G$.

C. Note that for given $f$ there may exist several $q$ and $a(q)$ satisfying (32); all of them may be found simultaneously by calculating the autocorrelation functions $B_{f_t}$.

The set of all $q$ satisfying (32) for a given $f$ when $a(q) = -1$ is a subgroup of $G$ ("inertia group" for $f$), so that the number of $q$ satisfying (36) for $a(q) = -1$ is always a divisor of the order $g$ of $G$ (this is also true for $a(q) = 1$), and this fact may be used to detect errors in the calculation of the autocorrelation function $B_{\Sigma}$. 
To calculate the autocorrelation functions $B_f$, one can use (34) and the "eveness relation": $B_f(t) = B_f(-t)$ for all $t \in G$. However, when the number of different values of $f$ is small, it is more convenient to use the formula [2]:

$$B_f = g \cdot F_G^{-1}(S_f \bar{S}_f)$$  

(39)

(where $\bar{S}_f$ is the complex conjugate of the spectrum $S_f$ of $f$, $F_G^{-1}$ the inverse Fourier transform on $G$, $S_f \cdot S_f(\omega) = S_f(\omega) \bar{S}_f(\omega)$). To calculate the spectrum and inverse Fourier transform $F_G^{-1}$ one can use Fast Fourier transform on the group $G$ [7].

D. Theorem 3 may be simplified when $f$ is a switching function. In this case $G$ is the dyadic group $G = 2^n, f(x) \in \{0,1\}$ for all $x \in G, f(x) = f_B = B_{11} = B_{11} = B_{-1} = B_{10} = B_{10} = B_{-1} = B_{10} = B_{11}$ and condition (36) may be replaced by

$$B_f(q) = B_f(q) = \sum_{x \in G} f(x).$$  

(40)

Since characters of the dyadic group are Walsh functions one can calculate $B_f$ by (39) and fast Walsh transform [8]. Tables of spectra and autocorrelation functions for a large number of classes of switching functions may be found in [5].

Example 5: Let $f(x_0, x_1, x_2)$ \((m = 3)\) be the switching function defined by Table III. Autocorrelation function $B_f$ is also shown in Table III. Since $B_f(1,1,0) = \Sigma_{x \in G} f(x) = 6$, formulas (40), (32) imply the following checking equation:

$$f(x_0, x_1, x_2) - f(x_0 \oplus 1, x_1 \oplus 1, x_2) = 0 \pmod{2}.$$

V. DESCRIPTION OF THE CLASS OF ERRORS DETECTED BY LINEAR EQUATIONS OVER A GROUP

A. Let us consider the class of errors detected by the general nonhomogeneous linear equation (24), if $a: G \to \{0,1\}$, $b: G \to \{0,1\}$, and $d \in C$ are chosen as described in the proof of Theorem 2.

We shall assume that the result of an error $e: G \to C$ is to replace the given function $f$ by the function $f + e$.

Set $\Omega_e = \{\omega | S_e(\omega) = 0\}$ (where $S_e(\omega)$ is the spectrum of the error function $e$).

Theorem 4: An error $e: G \to C$ is not detected by a nonhomogeneous equation (24) iff

$$G_e(\alpha) \subseteq \Omega_e$$  

(41)

where $\omega \in G_e(\alpha)$ iff $S_e(\omega) = 0$.

Proof: The error $e: G \to C$ is not detected by (24) iff, for every $x \in G$,

$$\Sigma \varphi(q) f(x^q^{-1}) + e(x^q^{-1}) = 0 \pmod{d} \Rightarrow \sum_{q \in d} b(q) \varphi(x^q^{-1}) + d = 0 \pmod{d}$$  

(42)

It follows from (42) by (24) that
\[ \sum_{q \in G} a(q) e(x^*q^{-1}) = 0 \text{ for every } x \in G \] (43)

and from (43) by the convolution theorem, that \( S_a(\omega) \cdot S_e(\omega) = 0 \) for all \( \omega \in G \), whence follows (41).

It follows from Theorem 4 that with increase in the order \( g_0(\alpha) \) of the subgroup \( G_0(\alpha) \) (and hence with decrease in the complexity \( \Sigma_{q \in G} a(q) \) of the checking equation (24); see, for example, (25)) the number of errors detected by (24) decreases. For an equation (6), if the set \( \Omega_f \cup \{0\}(\Omega_f = \{\omega \mid S_f(\omega) = 0\} \) is a subgroup of \( G \), it is convenient to take \( G_0 = \Omega_f \); then, by Theorem 4, an error \( e : G \to C \) is not detected iff the domain of nonzero values of the spectrum \( S_e \) of the error is a subset of the domain of nonzero values of the spectrum \( S_f \) of function \( f \).

B. The class of the most probable errors depends on the implementation of the device or program calculating the given function \( f \). We will consider two important examples to illustrate the good error detecting capability of the proposed methods.

We now illustrate the error detection capability of the above methods for the binary adder of Example 1. Block diagram of an \( n \)-bit adder is shown in Fig. 6.

We shall consider four classes of error for the adder of Fig. 6: input errors \( e_{\text{inp}}(X,Y) \), output errors \( e_{\text{out}}(X,Y) \), carry errors \( e_c(X,Y) \), and shift errors \( e_{sh}(X,Y) \).

An \( l \)-fold input (output) error, \( 0 < l \leq 2n \) (\( 0 < l \leq n + 1 \)), is said to occur if \( l \) binary components of \( x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1} \) (of \( g(x,Y), \ldots, f_s(x,Y) \)) are replaced by arbitrary binary constants (see Fig. 6).

An \( l \)-fold carry error, \( 0 < l \leq n \), occurs if \( l \) components of the vector \( (C_0, \ldots, C_n) \) (see Fig. 6) are replaced by arbitrary binary constants.

An \( l \)-fold shift error is a shift by \( l \) positions to the right or left in a vector recorded in any of the three registers \( X, Y, X + Y \) in Fig. 6. For a right (left) shift by \( l \) positions, the vector \((x_0, \ldots, x_{k-1})\) is replaced by \((0, \ldots, 0, x_{k-l}, \ldots, x_{k-1})\).

Shift errors are probable when the information is transferred to the \( X \) and \( Y \) registers and from the \( X + Y \) register in serial form.

We denote the relative frequency of \( l \)-fold errors which are detected of the above four classes by \( \eta_{\text{inp}}(n,l), \eta_{\text{out}}(n,l), \eta_c(n,l), \eta_{sh}(n,l) \), respectively.

**Corollary 3:** For an \( n \)-bit binary adder:

\[ \eta_{\text{inp}}(n,l) = \begin{cases} 1 - \left(\binom{n}{l}\right) 2^{-2s} & \text{if } l = 2s; \\ 1 & \text{if } l = 2s - 1; \end{cases} \] (44)

\[ \eta_{\text{out}}(n,l) = 1 - \delta_{l,n+1} 2^{-n-1} \] (45)

\[ \eta_c(n,l) = \eta_{sh}(n,l) = 1, \] for all \( n,l \). (46)

**Proof:** Any \( l \)-fold input error may be expressed as

\[ e_{\text{inp}}(X,Y) = \sum_{i=1}^{S_1} (\alpha_{pi} - x_{pi}^*) 2^{n-1-i} + \sum_{i=1}^{S_2} (\beta_{ri} - y_{ri}^*) 2^{n-1-i} \] (47)

where

\[ S_1 + S_2 = l; 0 \leq p_1 < \cdots < p_{s_1} \leq n - 1; \quad 0 \leq r_1 < \cdots < r_{s_2} \leq n - 1; \quad \alpha_{pi}, \beta_{ri} \in \{0,1\}. \]

Since

\[ \sum_{X,Y} (\alpha_{pi} - x_{pi}^*) = (-1)^{s_1+1} 2^{2n-1}, \quad i = 1, \ldots, S_1 \]
\[ \sum_{X,Y} (\beta_{ri} - y_{ri}^*) = (-1)^{s_2+1} 2^{2n-1}, \quad i = 1, \ldots, S_2 \] (48)

it follows that

\[ \sum_{X,Y} e_{\text{inp}}(X,Y) = 2^{2n-1} \left( \sum_{i=1}^{S_1} (-1)^{s_1+1} 2^{n-1-i} + \sum_{i=1}^{S_2} (-1)^{s_2+1} 2^{n-1-i} \right) \] (49)

Set \( E_{\text{inp}} = \{ e_{\text{inp}} \mid S_1 = S_2 = 0, \alpha_{pi} = 1 = \beta_{ri}, i = 1, \ldots, s_1 \} \). Then, if \( e_{\text{inp}} \notin E_{\text{inp}} \), we see from (49) that \( \Sigma_{X,Y} e_{\text{inp}}(X,Y) \neq 0 \) and \( e_{\text{inp}} \notin \Omega_{e_{\text{inp}}} \). Since \( 0 \in G_a(\alpha) \), it follows by Theorem 4 that \( e_{\text{inp}} \) is detected. If \( e_{\text{inp}} \in E_{\text{inp}} \), then by (49) \( \Sigma_{X,Y} e_{\text{inp}}(X,Y) = 0 \) and \( \Sigma_{X,Y} e_{\text{inp}}(X,Y) = 0 \) if \( \Sigma_{i=0}^{2n-1} e_{\text{inp}} = 0 \). Since (see Example 1)

\[ G_a(\alpha) = G_a = \{ \omega \mid \sum_{i=0}^{n-1} \omega_i = 2k (k = 0, \ldots, n) \} \]

it follows that in this case \( G_a \subseteq \Omega_{e_{\text{inp}}} \), so that by Theorem 4, \( e_{\text{inp}} \) is not detected.

Hence, in view of the fact that the total number of \( 2n \)-fold errors is equal to \( 2^{2n} - 2^{2n} \), and the number of \( 2n \)-fold errors \( e_{\text{inp}} \in E_{\text{inp}} \) is \( 2^n \cdot 2^n \), we obtain (44).

Any \( l \)-fold output error may be expressed as

\[ e_{\text{out}}(X,Y) = \sum_{i=1}^{l} (\alpha_{pi} - f_{pi}(X,Y)) 2^{n-p_i} \]

\[ \cdot (\alpha_{pi} \in [0,1], 0 \leq p_1 < \cdots < p_l \leq n). \] (50)

Since \( X + Y = \Sigma_{p=0}^{2n} f_{pi}(X,Y) 2^{n-p} (f_{pi}(X,Y) \in [0,1]) \), it is readily seen that for any \( p_i \in [0,\ldots,n] \) and \( \alpha_{pi} \in [0,1] \)

\[ \Sigma_{X,Y} f_{pi}(X,Y) = 2^{2n-1} - \delta_{p_i,0} 2^{n-1} \] and

\[ \sum_{X,Y} (\alpha_{pi} - f_{pi}(X,Y)) = (-1)^{n+1} 2^{2n-1} \]

\[ + (-1)^{n+1} \delta_{p_i,0} 2^{n-1}. \] (51)

Since \( 0 \leq p_1 < \cdots < p_l \leq n \), it follows from (50), (51) that

\[ \sum_{X,Y} e_{\text{out}}(X,Y) = 2^{2n-1} \left( \sum_{i=1}^{l} (-1)^{n+1} 2^{n-p_i} + (-1)^{n+1} \delta_{p_i,0} \right). \] (52)
Denote
\[ E_{out}(X, Y) = -f_0(X, Y)2^n + \sum_{p=1}^{n} (1 - f_p(X, Y))2^{n-p}. \]

Then if \( e_{out} \neq E_{out} \), it follows from (52) and (53) that \( \Sigma_{X,Y}e_{out}(X, Y) = 0 \) for some \( (0, \omega) \) not in \( \Omega_{e_{out}} \), and by Theorem 4, \( e_{out} \) is detected.

If \( e_{out} = E_{out} \), then \( \Sigma_{X,Y}e_{out}(X, Y) = 0 \), by (53) \( S_{e_{out}}(\omega) = -S_{X,Y}(\omega) \) for some \( \omega \) and, by Theorem 4, \( e_{out} \) is not detected.

Thus the only undetected output error is the \((n+1)\)-fold error \( E_{out} \) and this implies (45).

Now let \( e_{out}(X, Y) = e_{c}(x_0, \ldots, x_{n-1}, Y_0, \ldots, Y_{n-1}) \) be an \( l \)-fold carry error for which \( c_p = \alpha_{p_l} \) (see Fig. 6), where the \( \alpha_{p_l} \) are certain binary constants \((i = 1, \ldots, l; 0 \leq p_l < \cdots < p_1 \leq n-1)\). Then we have
\[ e_{c}(0, \ldots, 0) = \sum_{i=1}^{l} \alpha_{p_l}2^{2p_l}1, \quad e_{c}(1, \ldots, 1) = \sum_{i=1}^{l} (\alpha_{p_l} - 1)2^{2p_l+1} \]
and
\[ e_{c}(0, \ldots, 0) + e_{c}(1, \ldots, 1) = \sum_{i=1}^{l} (2\alpha_{p_l} - 1)2^{2p_l+1} \neq 0 \]
for any \( \alpha_{p_l} \in \{0, 1\} \). Now, for an adder (see (18)) we have
\[ e(q) = 1 \text{ iff } q = (0, \ldots, 0) \text{ or } q = (1, \ldots, 1) \]
and so it follows from (43) (54) that \( \pi_l(n, l) = 1 \) for all \( l \).

Similarly, for an arbitrary shift error \( e_{sh}(X, Y) = e_{sh}(x_0, \ldots, x_{n-1}, Y_0, \ldots, Y_{n-1}) \) we have
\[ e_{sh}(0, \ldots, 0) = 0, \quad e_{sh}(1, \ldots, 1) < 0 \]
and \( \nu_l(n, l) = 1 \) for all \( l \in \{1, \ldots, n\} \).

C. Another important example illustrating the good error detecting capability of our method is the transfer of information from the computer memory. Let the information be transferred from the memory by blocks consisting of \( P_0 \cdot P_1 \) words \((P_0, P_1 > 1)\) (e.g., we have \( P_0 \) independent memory devices and the information is transferred from every device by blocks consisting of \( P_1 \) words). We consider the every block as a \( P_0 \times P_1 \) matrix with the elements \( f(x_0, x_1) \), where \( f \) is the function defined on the group \( G \) of the all vectors \( x = (x_0, x_1) \) \((x_0 \in \{0, \ldots, P_0 - 1\}; x_1 \in \{0, \ldots, P_1 - 1\}) \) and \( G \) is a direct product of two cyclic groups of the orders \( P_0 \) and \( P_1 \).

The checking equation (2) for \( f \) (constructed by the method of Section II) generates the error detection method for errors arising in the process of storage and transfer of information from the computer memory. The method may be easily implemented by the simple program for checking (2) for some \( x \). (These \( x \) may be chosen as a test for block diagram of Fig. 1 for the function \( f \) (see Section IV.).

In this case it is natural to understand by an \( l \)-fold error \((l \in \{1, \ldots, P_0 \cdot P_1\})\) an error \( e \) disturbing \( l \) words in the block, i.e., the number of \( x \) such that \( e(x) \neq 0 \).

From Theorem 4 if \( \Sigma_{x \in G} e(x) \neq 0 \) then \( e \) is detected by (2). Hence, if each word is the \( n \)-bit binary number then we have the relative frequency \( \eta(n, l) \) of detected \( l \)-fold errors: \( \eta(n, 1) = 1, \eta(n, 2) \geq 1 - (2^n - 1)^{-1} \) and so on. For asymmetric errors (i.e., for errors such that \( e(x) \neq 0 \) or \( e(x) \leq 0 \) for all \( x \epsilon G \)) \( \eta(n, l) = 1 \) for all \( l \in \{1, \ldots, P_0 \cdot P_1\} \).
D. In order to look for solutions \( a(q) \) of the checking equation (1), we used methods of abstract harmonic analysis on finite commutative groups. When this is done, the complexity of Figs. 1, 2, and 4 depends on the orders of the selected subgroups \( G_\alpha, G_\alpha(\alpha), G_\alpha(1) \) of \( G \) contained in the sets \( \Omega_f \cup \{0\}, \Omega_f \cup \Omega_f, \phi \cup \{0\}, (\Omega_f \cap \Omega_f, \phi) \cup \Omega_f, \phi(1) \cup \{0\} \) (see Theorems 1, 2 and Corollary 2). Generally speaking, therefore, as far as the complexity of the checking block diagrams is concerned, the most suitable functions for this method are functions whose generalized Fourier spectrum contains sufficiently many zeros.

Another limitation on the use of our methods is implied by the fact that they yield only solutions \( a(q) \) of (1) for which \( \{q | a(q) = 1\} \) is a subgroup of the original group \( G \).

The main advantage of the methods proposed for solving (1) lies in their simplicity and convenience from the computational point of view. Thus if the initial functions \( f: G \to C \) are defined analytically, the solution \( a(q) \) may often be found analytically too (see Examples 1, 2). Note that in the binary case (when \( G \) is a dyadic group) the solution \( a(q) \) may be determined with the help of the tables in [5], which list the Walsh spectra and autocorrelation functions for a large number of important classes of Boolean functions.

If the function \( f: G \to C \) is specified in tabular form, the solution \( a(q) \) may be sought by employing the very effective fast Fourier transform algorithm on \( G \) [7], [8].

Another advantage of the methods is their weak dependence on the original group \( G \), so that one obtains a unified set of error detection methods for devices that operate in binary and \( q \)-ary \( (q > 2) \) systems, in systems using residue classes, and so on.

REFERENCES

[1] M. G. Karpovsky and E. S. Moskalev, "Realization of a system of logical functions by means of an expansion in orthogonal series,"


Mark G. Karpovsky was born in Leningrad, USSR, on October 27, 1940. He received the B.S. and M.S. degrees in computer science and the Ph.D. degree in theoretical cybernetics from the Leningrad Electrotechnical Institute, Leningrad, USSR, in 1961, 1963, and 1967, respectively.

Since March 1974 he has been Senior Lecturer at the Department of Mathematics, Tel-Aviv University, Tel-Aviv, Israel. His current research interests are in synthesis fault-tolerant networks, analysis and simulation biological systems, test diagnosis of logical devices, application of abstract harmonic analysis in analysis, and synthesis and optimization of logical networks.

Dr. Karpovsky is a member of the Israel Mathematical Society.