S. Winograd

Coding for Logical Operations

Abstract: The behavior of a computation system consisting of encoders, an unreliable logical operator and a decoder is investigated. It is shown that for almost all Boolean functions, coding each block of k input bits into a block of n bits such that all sets of s or less errors will be corrected requires that $n \geqq (2s + 1)k$. This result suggests that the capacity (in the information theoretical sense) of such a computation system is zero.

Introduction

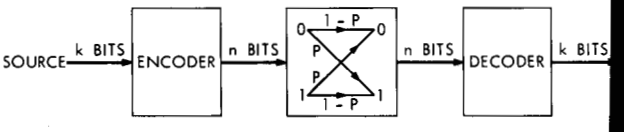
This paper investigates one aspect of the problem of performing reliable computation using less reliable components. This problem is of interest both to the neurophysiologist and to the computer engineer. The nervous system is capable of functioning reliably, yet there is evidence that the function performed by a single neuron is probabilistic. To use the language of a communication engineer, the function performed by a single neuron is perturbed by noise. The computer engineer is faced with the problem of a growing demand, both military and commercial, for computers designed to function reliably in spite of thermal and electrical noise present in the system. The problem facing the computer engineer may become even more acute if the techniques which are used for producing active elements in very large quantities do not result in the production of highly reliable individual elements.

Von Neumann,[1] Moore and Shannon,[2] Muroga,[3] and McCulloch, Cowan et al.[4] applied themselves to the problem of designing a reliable automaton using less reliable components. Each of them investigated the problem using different elements; and in each case the results of the investigation showed that by making the automaton redundant, i.e., by using more components than is absolutely necessary for the design of the automaton, it was possible to make the probability of malfunction arbitrarily small. The exact amount of redundancy required to achieve a certain level of reliability differed in each case; yet all the methods had one feature in common: In order to obtain arbitrarily high reliability, almost all the elements in the redundant automaton were needed to combat the noise, and only a very small fraction (which approached zero with increase in the reliability) were needed to perform the desired function.

Elias[5] noted that this feature of the results is "unsatisfying, or at least disappointing, from a theoretical point of view in the context set by information theory. In fact, these results have the character of the pre-information theory results on the reliable transmission of information over unreliable (noisy) channels." Consider the transmission system shown in Fig. 1. The source sends ZEROS and ONES, the encoder converts a block of k bits sent by the source into a block of n bits. With probability p, each of the n bits entering the memoryless binary symmetric channel comes out incorrectly. The decoder accepts the block of n bits perturbed by the noise and decodes it into a block of k bits, usually the same block of k bits sent by the source. Mistakes are still made by the system, namely the block of k bits coming out of the decoder is not the same block sent by the source; but the fundamental theorem for a discrete memoryless channel with noise[6] guarantees that if the rate of transmission, $R = k/n$ bits per symbol, is less than the channel capacity, $C = 1 + p \log_2 p + (1 - p)\log_2(1 - p)$, then keeping R constant and letting k and n increase can make the probability of error arbitrarily small.

The question then arises: Can we have a computation system, as shown in Fig. 2, similar to the transmission system, which will exhibit similar capabilities

Figure 1 **Transmission system for binary symmetric channel.**



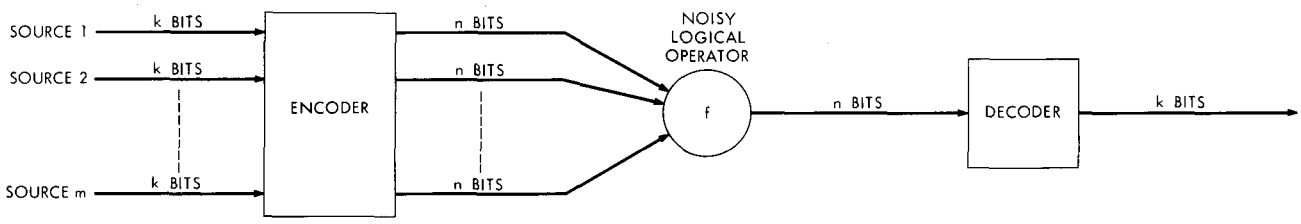


Figure 2 **Computation system for noisy logical operator.**

for combating the noise? In other words, what is the capacity of a noisy logical operator?

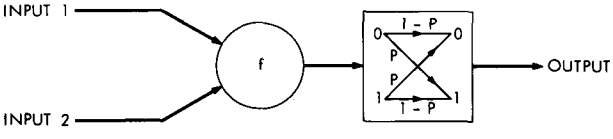
One major difference between the communication system shown in Fig. 1 and the computation system shown in Fig. 2 is the requirement on the block of k bits coming out of the decoder. In the communication system we want the block of k bits coming out of the decoder to be the same as the block of k bits sent by the source; while in the case of the computation system we want the k bits coming out of the decoder to be the same as the block of k bits which would have resulted had a noiseless logical operator f operated on the m blocks of k bits sent by the m sources.

The computation system shown in Fig. 2 should not be viewed as a scheme for designing a reliable automaton using less reliable components. In the computation system, as well as in the communication system, the encoder and the decoder are assumed to operate with complete reliability. It is therefore unrealistic to assume that parts of the automaton, namely the logical operator f, are affected by noise, while other parts, namely the encoder and decoder, are not affected. The computation system of Fig. 2 was devised for the sole purpose of studying the relation of information theory of reliable automata.

Background

In 1958 P. Elias presented a paper entitled "Computation in the Presence of Noise,"[5] in which he investigated the possibilities of block coding the inputs of a noisy logical operator and then decoding the output block. In particular, Elias investigated all Boolean functions of two variables. He assumed that the nature of the noise is such that with probability p the output of the noisy logical operator is incorrect. Thus the logical operators he investigated can be represented schematically as shown in Fig. 3. The box f is assumed to be noiseless, and the entire effect of the noise is represented by a memoryless binary symmetric channel.

Figure 3 **Noisy logical operator.**



Elias divided the 16 Boolean functions of two variables into two classes:

$$C_1 = \{0, 1, a, a', b, b', a \oplus b, a' \oplus b\}$$

$$C_2 = \{a \cdot b, a' \cdot b, a \cdot b', a' \cdot b', a + b, a' + b, a + b', a' + b'\}.$$

(We use the notation $a \oplus b \equiv ab' + a'b$.)

No coding is necessary for the first two functions of C_1, namely $f = 0$ and $f = 1$, because the output does not convey any information about the inputs. As for the other six functions in C_1, group codes[7] may be used to code their inputs, and as long as the rate of flow of information at the output is less than $1 + p \log_2 p + (1 - p)\log_2(1 - p)$ bits per symbol, which is the capacity of the binary symmetric channel, an arbitrarily small probability of error may be achieved.

In order to investigate the possibilities of efficient coding for the functions of C_2, Elias considered the system shown in Fig. 4. He proved that in the system of Fig. 4, if f is in C_2, then in order to correct all possible sets of s or less errors, n must be at least $(2s + 1)k$. This is in marked contrast to the results obtained for group codes.[7,8,9]

On the basis of his investigation, Elias conjectured that even in the more general case, shown in Fig. 5, if $f \in C_2$ then $n \geqq (2s + 1)k$ is a necessary condition for correction of all possible sets of s or less errors.

Consider the computation system shown in Fig. 5. Source 1 sends a block of k bits $\mathbf{X}_1 = (x_{11}, x_{12} \cdots x_{1k})$, which are encoded by Encoder 1 into a block of n bits $\mathbf{Y}_1 = E_1(\mathbf{X}_1) = (y_{11}, y_{12} \cdots y_{1n})$. Similarly, the second source sends a block of k bits $\mathbf{X}_2 = (x_{21}, x_{22} \cdots x_{2k})$ which are encoded by Encoder 2 into a block of n bits $\mathbf{Y}_2 = E_2(\mathbf{X}_2) = (y_{21}, y_{22} \cdots y_{2n})$. The function F operates on the two vectors $\mathbf{Y}_1$ and $\mathbf{Y}_2$ bit by bit, and in the absence of noise the result of the computation would have been $\mathbf{Z} = (z_1, z_2 \cdots z_n)$. Because of the noise, which is represented by the binary symmetric channel, the output of the noisy operator, F, is the block $\mathbf{Z}^* = (z_1^*, z_2^* \cdots z_n^*)$, which is the block $\mathbf{Z}$ distorted by the noise. The decoder accepts $\mathbf{Z}^*$ as its input and performs the function D on it to obtain $\mathbf{U} = D(\mathbf{Z}^*) = (u_1, u_2 \cdots u_k)$. For a reliable computation system we expect $\mathbf{U}$ to be $f(\mathbf{X}_1, \mathbf{X}_2)$ most of the time, and consider the system to be in error when

$\mathbf{U}$ is not $f(\mathbf{X}_1, \mathbf{X}_2)$. Note that in this general scheme, the function performed by the logical operator, F, is not necessarily the function, f, which we want the whole system to perform reliably.

A comparison between the computation system of Fig. 2 and that of Fig. 5 shows a restriction which Elias imposed on the system. The blocks of k bits sent from each source are to be encoded independently of the blocks sent from the other source. This is shown schematically by the two separate encoders; Encoder 1, which operates on $\mathbf{X}_1$ above, and Encoder 2, which operates on $\mathbf{X}_2$ above. The reason for this restriction is to guarantee that $\mathbf{Y}_1$ and $\mathbf{Y}_2$ carry information only about $\mathbf{X}_1$ and $\mathbf{X}_2$ respectively and not about any logical combination of the two blocks. This means that none of the desired computation, $f(\mathbf{X}_1, \mathbf{X}_2)$, is carried out in the encoder, which is assumed to be noiseless. Elias imposed another restriction on the system, to ensure that none of the desired computation is performed by the decoder. He required that in the absence of noise the function performed by the decoder will be one-to-one. This means that there is a one-to-one correspondence between $\mathbf{U}$ and $\mathbf{Z}$, and that $\mathbf{Z} = D^{-1}(\mathbf{U})$. This restriction means that whatever information is present in $\mathbf{Z}^*$ about $\mathbf{X}_1$ and $\mathbf{X}_2$ concerns only the logical combination of $\mathbf{X}_1$ and $\mathbf{X}_2$. For further discussion on these two restrictions, the reader is referred to Elias' paper.[5]

W. W. Peterson and M. O. Rabin, in their paper, "On Codes for Checking Logical Operations,"[10] proved Elias' conjecture, with a very mild restriction. They let $F = (F_1, F_2 \cdots F_n)$, which means that $z_i = F_i(y_{1i}, y_{2i})$, and showed that if we require that $E_1(\mathbf{0}^k) = E_2(\mathbf{0}^k) = D^{-1}(\mathbf{0}^k) = \mathbf{0}^n$, ($\mathbf{0}^k$ is the block of k ZEROS and $\mathbf{0}^n$ is the block of n ZEROS), then Elias' conjecture holds for any $f \in C_2$. This means that under these conditions, in order to correct all s or less errors, $n \geqq (2s + 1)k$ has to hold.

The problem

Consider the general computation system of Fig. 6. This system operates under the same conditions which Elias imposed in the case of functions of two variables, namely:

1. The block of k bits sent by each source is encoded independently of the bits sent by any other source. This restriction is manifested by having m different encoders for the m different sources.

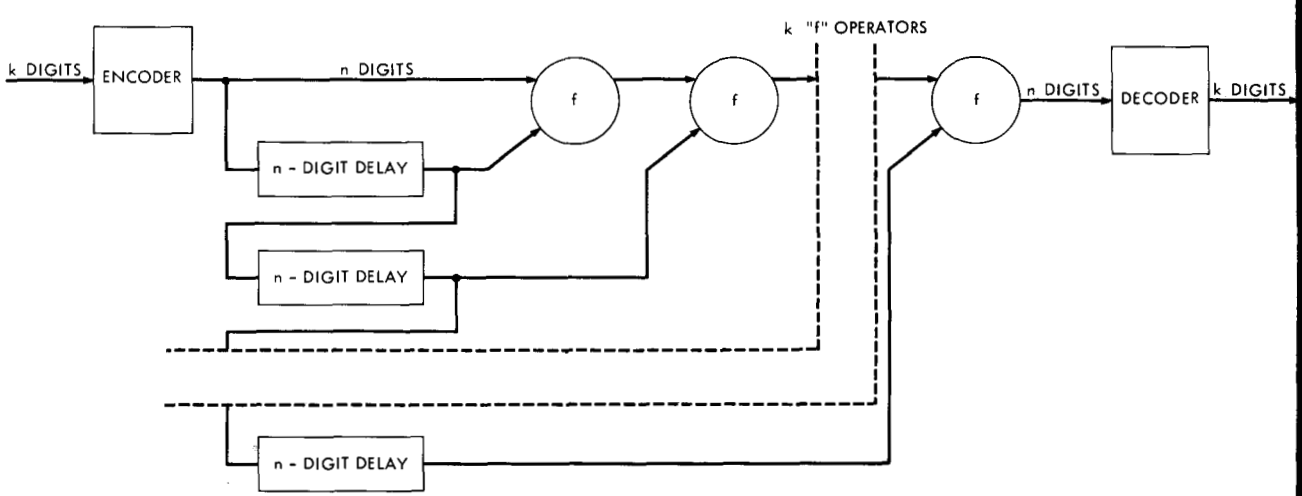
2. In the absence of noise, the function performed by the decoder is one-to-one. That means that $\mathbf{Z} = D^{-1}(\mathbf{U})$.

We shall now investigate the possibilities of coding for a pair of functions (F, f). In particular we would like to find those pairs (F, f) for which the ability to correct all sets of s or less errors requires that $n \geqq (2s + 1)k$. We say that *Elias' conjecture holds for a pair of functions (F, f) if the ability to correct all sets of s or less errors for this pair implies $n \geqq (2s + 1)k$.*

We will limit ourselves to memoryless functions, which means that the function F can be represented as $F = (F_1, F_2 \cdots F_n)$ and $z_i = F_i(y_{1i}, y_{2i} \cdots y_{mi})$; and the function f can be represented as $f = (f_1, f_2 \cdots f_k)$, which means that u_i is supposed to be $f_i(x_{1i}, x_{2i} \cdots x_{mi})$.

With no loss of generality we can assume that for each coordinate i, $(1 \leqq i \leqq n)$, there are two vectors $\mathbf{Z}_1 = D^{-1}(\mathbf{U}_1)$ and $\mathbf{Z}_2 = D^{-1}(\mathbf{U}_2)$ such that $(\mathbf{Z}_1)_i = 0$ and $(\mathbf{Z}_2)_i = 1$. In other words, $\sum_{\text{all } \mathbf{Z}} \mathbf{Z} = \mathbf{1}^n$ and $\prod_{\text{all } \mathbf{Z}} \mathbf{Z} = \mathbf{0}^n$, where $\prod$ indicates the logical operation AND performed coordinate-wise, and $\sum$ indicates the logical operation OR performed coordinate-wise. If this condition is not satisfied for some coordinate i of $\mathbf{Z}$ then this coordinate carries no information about the

Figure 4 **Multiple-*f* computer.**



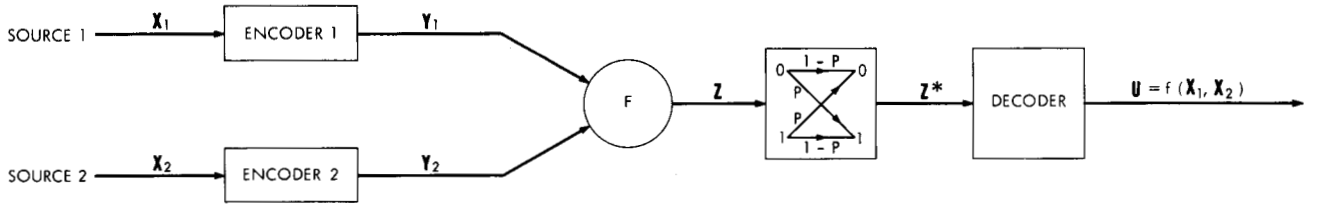


Figure 5 **General computation system for functions of two variables.**

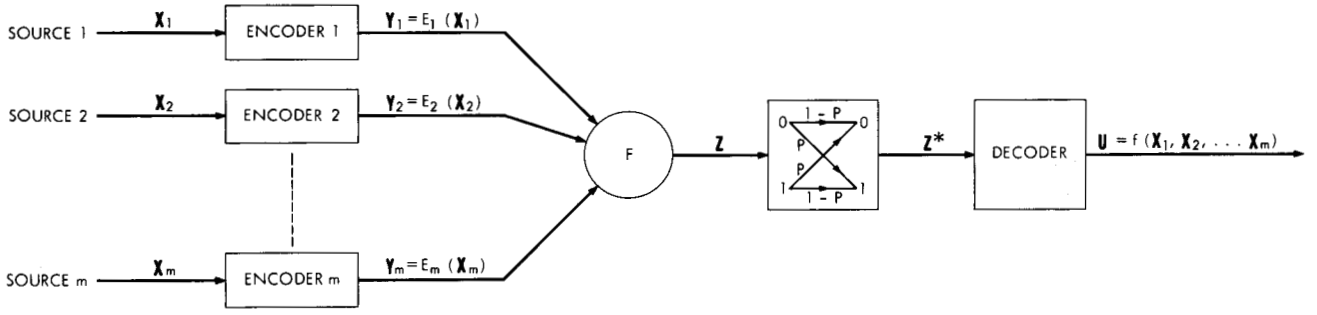


Figure 6 **General computation system for functions of *m* variables.**

desired output. Let $n'(n' \leqq n)$ be the number of coordinates which do satisfy this condition then proving that $n' \geqq (2s+1)k$ proves also that $n \geqq (2s+1)k$.

We will first consider functions of two variables ($m = 2$), and then we will generalize the results obtained, to functions of m variables.

Functions of two variables

We will now demonstrate that Elias' conjecture holds for all the functions of two variables in C_2 even without the restriction of Peterson and Rabin.

Theorem 1

Let $F = (F_1, F_2 \cdots F_n)$ and $f = (f_1, f_2 \cdots f_k)$. If $F_i(a, b) = f_j(a, b) = a \cdot b$ for all $1 \leqq i \leqq n$ and all $1 \leqq j \leqq k$ then Elias' conjecture holds.

To prove the theorem we will need the following lemma:

Lemma 1: Let $D^{-1}(\mathbf{X}_1 \cdot \mathbf{X}_2) = E_1(\mathbf{X}_1) \cdot E_2(\mathbf{X}_2)$ and let $\sum_{\text{all } \mathbf{X}} D^{-1}(\mathbf{X}) = \mathbf{1}^n$ then:

1. For all $\mathbf{X}$, $D^{-1}(\mathbf{X}) = E_1(\mathbf{X}) = E_2(\mathbf{X})$
2. $\mathbf{X}_1 \geqq \mathbf{X}_2 \Rightarrow D^{-1}(\mathbf{X}_1) \geqq D^{-1}(\mathbf{X}_2)$

Proof: For all $\mathbf{X}$,

$$D^{-1}(\mathbf{X}) = D^{-1}(\mathbf{X} \cdot \mathbf{1}^k) = E_1(\mathbf{X}) \cdot E_2(\mathbf{1}^k);$$

this means that $D^{-1}(\mathbf{X}) \leqq E_2(\mathbf{1}^k)$. But since $D^{-1}(\mathbf{X}) \leqq E_2(\mathbf{1}^k)$ for all $\mathbf{X}$ then $\mathbf{1}^n = \sum_{\text{all } \mathbf{X}} D^{-1}(\mathbf{X}) \leqq E_2(\mathbf{1}^k)$ and therefore $E_2(\mathbf{1}^k) = \mathbf{1}^n$. Thus, for all $\mathbf{X}$,

$$D^{-1}(\mathbf{X}) = E_1(\mathbf{X}) \cdot E_2(\mathbf{1}^k) = E_1(\mathbf{X}) \cdot \mathbf{1}^n = E_1(\mathbf{X}).$$

Similarly $D^{-1}(\mathbf{X}) = E_2(\mathbf{X})$ for all $\mathbf{X}$. Let $\mathbf{X}_1 \geqq \mathbf{X}_2$ then:

$$D^{-1}(\mathbf{X}_2) = D^{-1}(\mathbf{X}_1 \cdot \mathbf{X}_2) = E_1(\mathbf{X}_1) \cdot E_2(\mathbf{X}_2) = D^{-1}(\mathbf{X}_1) \cdot D^{-1}(\mathbf{X}_2)$$

which means that $D^{-1}(\mathbf{X}_1) \geqq D^{-1}(\mathbf{X}_2)$.

Q.E.D.

Proof of Theorem 1: To be able to correct all possible sets of s or less errors, the Hamming distance[8] $d(\mathbf{Z}_1, \mathbf{Z}_2)$ between any two distinct vectors $\mathbf{Z}_1$ and $\mathbf{Z}_2$ has to satisfy $d(\mathbf{Z}_1, \mathbf{Z}_2) = W(\mathbf{Z}_1) + W(\mathbf{Z}_2) - 2W(\mathbf{Z}_1 \cdot \mathbf{Z}_2) \geqq 2s + 1$ where $W(\mathbf{Z})$ is the number of 1's in $\mathbf{Z}$. In particular if $\mathbf{Z}_1 \geqq \mathbf{Z}_2$ we obtain $W(\mathbf{Z}_1) = W(\mathbf{Z}_2) + d(\mathbf{Z}_1, \mathbf{Z}_2)$. Let $(\mathbf{1}^i\mathbf{0}^{k-i})$ represent the vector whose first i coordinates are 1 and the remaining $k - i$ are 0. Since we have

$$(\mathbf{1}^k) \geqq (\mathbf{1}^{k-1}\mathbf{0}) \geqq (\mathbf{1}^{k-2}\mathbf{0}^2) \cdots \geqq (\mathbf{0}^k)$$

then by Lemma 1,

$$D^{-1}(\mathbf{1}^k) \geqq D^{-1}(\mathbf{1}^{k-1}\mathbf{0}) \cdots \geqq D^{-1}(\mathbf{0}^k).$$

Thus:

$$n \geqq W[D^{-1}(\mathbf{1}^k)] = d[D^{-1}(\mathbf{1}^k), D^{-1}(\mathbf{1}^{k-1}\mathbf{0})] + W[D^{-1}(\mathbf{1}^{k-1}\mathbf{0})]$$

$$= \sum_{i=0}^{k-1} d[D^{-1}(\mathbf{1}^{k-i}\mathbf{0}^i), D^{-1}(\mathbf{1}^{k-i-1}\mathbf{0}^{i+1})] + W[D^{-1}(\mathbf{0}^k)]$$

$$\geqq k(2s+1) + W[D^{-1}(\mathbf{0}^k)] \geqq k(2s+1)$$

Q.E.D.

It is clear that a similar proof can be carried out

when the logical function to be performed is OR rather than AND, and that therefore Elias' conjecture holds also in the case that $F_i(a, b) = f_j(a, b) = a + b$ for all $1 \leqq i \leqq n$ and all $1 \leqq j \leqq k$. A similar argument can be carried out for the remaining six functions of C_2. To prove this result in a formal way, we note that any function $g \in C_2$ can be written as $g(a, b) = (a \oplus t_1) \cdot (b \oplus t_2) \oplus t_3$ for some t_1, t_2, t_3 ($t_i = 0$ or 1). We will show that if a function F^* is obtained from a function F by complementing some of the variables of F or possibly complementing F, and a function f^* is obtained from a function f in a similar manner, then the pair (F^*, f^*) is equivalent to the pair (F, f) as far as Elias' conjecture is concerned; i.e., Elias' conjecture holds for the pair (F^*, f^*) if and only if it holds for the pair (F, f). Since any function $g(a, b) \in C_2$ can be obtained from the function $g(a, b) = a \cdot b$ in such a way, this will prove that Elias' conjecture holds for all the functions of C_2.

Let X be a v-dimensional Boolean space ($X = \{0, 1\}^v$), and let T be any v-dimensional Boolean vector. By $T(X)$ we will denote the function which maps any vector $x \in X$ into the vector $x \oplus T \in X$, where $x \oplus T$ means modulus 2 addition, coordinate by coordinate.

• *Theorem 2*

Let $F = (F_1, F_2, \cdots, F_n)$ and $f = (f_1, \cdots, f_k)$, if $F_i \in C_2$ and $f_j \in C_2$ for all $1 \leqq i \leqq n$ and $1 \leqq j \leqq k$, then Elias' conjecture holds for the pair (F, f).

To prove the theorem we need the following lemma:

Lemma 2: Let $T_0, T_1, \cdots, T_m$ be any $m + 1$ Boolean vectors of n coordinates. Let $T_0', T_1', \cdots, T_m'$ be any $m + 1$ vectors of k coordinates. Then: Elias' conjecture holds for the pair (F, f) if and only if it holds for the pair (F^*, f^*), where

$$F^*(\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_m) = T_0\{F[T_1(\mathbf{Y}_1), T_2(\mathbf{Y}_2), \cdots, T_m(\mathbf{Y}_m)]\}$$

and

$$f^*(\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_m) = T_0'\{f[T_1'(\mathbf{X}_1), T_2'(\mathbf{X}_2), \cdots, T_m'(\mathbf{X}_m)]\}.$$

Note that in the lemma we let F and f (and therefore F^* and f^*) be functions of m variables where m is not necessarily equal to 2.

Proof: We will first prove that if Elias' conjecture does not hold for (F, f) then it does not hold for (F^*, f^*) either. Let $E_1, E_2, \cdots, E_m$ be the encoders and D be the decoder for the pair (F, f) such that $n < (2s + 1)k$ and yet for any two distinct vectors $\mathbf{U}_1$ and $\mathbf{U}_2$, $d[D^{-1}(\mathbf{U}_1), D^{-1}(\mathbf{U}_2)] \geqq 2s + 1$. Define $E_1^*, E_2^*, \cdots, E_m^*$ to be $E_i^*(\mathbf{X}_i) = T_i E_i T_i'(\mathbf{X}_i)$ for all $1 \leqq i \leqq m$ and define D^* to be $D^*(\mathbf{Z}^*) = T_0' D T_0(\mathbf{Z}^*)$. Using $E_1^*, E_2^*, \cdots, E_m^*$ as encoders and D^* as a decoder for the logical operator F^* will make the whole system compute f^* because:

$$\mathbf{U}^* = D^*(\mathbf{Z}) = D^*F^*(\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_m) = D^*F^*[E_1^*(\mathbf{X}_1), E_2^*(\mathbf{X}_2), \cdots, E_m^*(\mathbf{X}_m)].$$

Substituting the values of D^*, F^*, and E_i^* in terms of D, F, E_i and using the fact that for all T_i, $T_i^2(\mathbf{Y}) = \mathbf{Y}$:

$$\mathbf{U}^* = T_0' D F\{E_1[T_1'(\mathbf{X}_1)], E_2[T_2'(Y_2)], \cdots, E_m[T_m'(X_m)]\}.$$

But since D and the E_i's are the decoder and encoder for the pair (F, f), and therefore $f = DF[E_1(\mathbf{X}_1), \cdots, E_m(\mathbf{X}_m)]$, we obtain:

$$\mathbf{U}^* = T_0' f[T_1'(\mathbf{X}_1), T_2'(\mathbf{X}_2), \cdots, T_m'(\mathbf{X}_m)] = f^*.$$

Thus for every vector $\mathbf{U}^*$, we obtain:

$$(D^*)^{-1}(\mathbf{U}^*) = T_0 D^{-1} T_0'(\mathbf{U}^*) = T_0 \oplus D^{-1}(\mathbf{U}),$$

where

$$\mathbf{U} = f[T_1'(\mathbf{X}_1), T_2'(\mathbf{X}_2), \cdots, T_m'(\mathbf{X}_m)].$$

Therefore for any two vectors $\mathbf{U}_1^*$ and $\mathbf{U}_2^*$ we obtain $d[(D^*)^{-1}(\mathbf{U}_1^*), (D^*)^{-1}(\mathbf{U}_2^*)] = d[T_0 \oplus D^{-1}(\mathbf{U}_1), T_0 \oplus D^{-1}(\mathbf{U}_2)] = d[D^{-1}(\mathbf{U}_1), D^{-1}(\mathbf{U}_2)] > 2s + 1$. Thus Elias' conjecture does not hold for (F^*, f^*) if it does not hold for (F, f). To prove that Elias' conjecture does not hold for (F, f) if it does not hold for (F^*, f^*), note that $(F, f) = [(F^*)^*, (f^*)^*]$.

Q.E.D.

Proof of Theorem 2: Since Elias' conjecture holds for the case $F_i(a, b) = f_j(a, b) = a \cdot b$ for all $1 \leqq i \leqq n$ and $1 \leqq j \leqq k$, it holds by Lemma 2 for the case that $F_i(a, b) = (a \oplus t_{i_1}) \cdot (b \oplus t_{i_2}) \oplus t_{i_3}$ and $f_j(a, b) = (a \oplus t_{j_1}) \cdot (b \oplus t_{j_2}) \oplus t_{j_3}$, which proves the theorem.

We will now demonstrate that the requirements imposed on the computation system imply that if $f \in C_2$ then $F_i \in C_2$ for all $1 \leqq i \leqq n$, and therefore Elias' conjecture will hold for every system which computes a function $f \in C_2$ using a bit-by-bit process, independently of F.

Lemma 3: Let $F = (F_1, F_2 \cdots F_n)$ and $f = (f_1, f_2 \cdots f_k)$, if $f_i(a, b) = a \cdot b$ for all $1 \leqq i \leqq k$ and $\sum_{\text{all } \mathbf{U}} D^{-1}(\mathbf{U}) = \mathbf{1}^n$ and $\prod_{\text{all } \mathbf{U}} D^{-1}(\mathbf{U}) = \mathbf{0}^n$, then $F_i \in C_2$ for all $1 \leqq i \leqq n$.

Proof: For every i there are two vectors $\mathbf{X}^1$ and $\mathbf{X}^2$ such that $E_1(\mathbf{X}^1)_i = 0$ and $E_1(\mathbf{X}^2)_i = 1$, otherwise $F[E_1(\mathbf{X}), E_2(\mathbf{1}^k)] = D^{-1}(\mathbf{X})$ will have the same i^{th} coordinate for all $\mathbf{X}$, contrary to the assumption. Similarly we can find two vectors $\mathbf{X}^3$ and $\mathbf{X}^4$ (not necessarily distinct from $\mathbf{X}^1$ and $\mathbf{X}^2$) such that $E_2(\mathbf{X}^3)_i = 0$ and $E_2(\mathbf{X}^4)_i = 1$. But

$$F[E_1(\mathbf{X}^1), E_2(\mathbf{0}^k)] = F[E_1(\mathbf{X}^2), E_2(\mathbf{0}^k)] = F[E_1(\mathbf{0}^k), E_2(\mathbf{X}^3)] = F[E_1(\mathbf{0}^k), E_2(\mathbf{X}^4)] = D^{-1}(\mathbf{0}^k).$$

From this it follows that

$$F_i[0, E_2(\mathbf{0}^k)_i] = F_i[1, E_2(\mathbf{0}^k)_i] = F_i[E_1(\mathbf{0}^k)_i, 0]$$
$$= F_i[E_1(\mathbf{0}^k), 1] .$$

But the only Boolean functions g of two variables which are not constant and which can satisfy $g(0, b) = g(1, b) = g(a, 0) = g(a, 1)$ for some a, b are the functions of C_2.

Q.E.D.

• *Theorem 3*

Let $F = (F_1, F_2 \cdots F_n)$ and $f = (f_1, f_2 \cdots f_n)$, if $f_i \in C_2$ for all $1 \leqq i \leqq k$ then Elias' conjecture holds.

Proof: Because of Lemma 3 and Theorem 2, Elias' conjecture holds if $f_i(a, b) = a \cdot b$ for all $1 \leqq i \leqq k$. Apply Lemma 2 to this result.

Q.E.D.

Functions of many variables

As will be shown, results concerning functions of two variables may be used as a tool for investigating functions of many variables. We will limit the domain of a large class of functions of many variables, without affecting their range, and then apply the previous results to the functions with limited domain.

Before proceeding to investigate functions of many variables in general, we will study an example of a function of three variables and show that Elias' conjecture holds for this function. Then we will apply the same reasoning to find the class of functions of m variables for which Elias' conjecture holds. Consider the case when $F = (F_1, F_2, \cdots, F_n)$ and $f = (f_1, f_2, \cdots, f_k)$ where $f_i(a, b, c) = a \cdot b + c$ for all $1 \leqq i \leqq k$; that means that $f(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3) = \mathbf{X}_1 \cdot \mathbf{X}_2 + \mathbf{X}_3$ where the operations are performed coordinate by coordinate. Consider the set of all inputs $(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3)$ such that $\mathbf{X}_3 = (\mathbf{0}^n)$. In this case the function which is actually computed is $\mathbf{X}_1 \cdot \mathbf{X}_2$ and therefore (by Theorem 3) $n \geqq (2s + 1)k$. That means that Elias' conjecture holds for the case just studied.

We shall find it advantageous to represent the Boolean functions in terms of the binary operations " $\cdot$ " (multiplication) and " $\oplus$ " (modulus 2 addition). Thus every Boolean function $g(x_1, x_2, \cdots, x_m)$ can be represented as

$$g(x_1, x_2, \cdots, x_m) = \sum_{e_1 e_2 \cdots e_m} C_{e_1 e_2 \cdots e_m} \prod_{i=1}^{m} x_i^{e_i} ,$$

where e_i and $C_{e_1 e_2 \cdots e_m}$ take the values 0 or 1, and $x_i^0 = 1$ and $x_i^1 = x_i$, and the summation is performed modulus 2 over all the 2^m possible values of the m-tuples $(e_1 e_2 \cdots e_m)$. This means that each Boolean function can be viewed as a polynomial in the variables $x_1, x_2 \cdots x_m$. In particular we can define the linear Boolean functions, as those functions which do not include products of the variables (i.e., $C_{e_1 e_2 \cdots e_m} = 0$ for all m-tuples $e_1 e_2 \cdots e_m$ in which two or more e_i's take the value 1).

Lemma 4: Let $g(x_1, x_2, \cdots, x_m)$ be a nonlinear Boolean function, then there exist two variables x_i and x_j and $m - 2$ constants $c_r (r \neq i, j)$ such that

$$g(c_1, \cdots x_i, \cdots x_j, \cdots c_m) = h(x_i, x_j) \in C_2 .$$

Proof: Since $g(x_1, x_2 \cdots x_m)$ is nonlinear, there exists at least one product of some two variables. Let $x_i \cdot x_j \cdot x_{r_1} \cdot x_{r_2} \cdots x_{r_v}$ be the lowest product in which x_i and x_j appear. (The lowest product means the product with the smallest number of variables. If more than one lowest product exists we can take any one of them.) Let $c_r = 1$ if r is one of the r_i's which appear in the product, and $c_r = 0$ for all other $r \neq i, j$. Then: $g(c_1, c_2 \cdots x_i \cdots x_j \cdots c_m) = K_0 \oplus K_1 x_i \oplus K_2 x_j \oplus x_i x_j = h(x_i, x_j) \in C_2$.

Q.E.D.

Note that the range of $h(x_i, x_j)$ is the same as the range of $g(x_1 \cdots x_m)$.

• *Theorem 4*

Consider the general computation system of Fig. 6. Let $F = (F_1, F_2 \cdots F_n)$ and $f = (f_1, f_2 \cdots f_k)$ such that $f_1 = f_2 = \cdots = f_k$, then Elias' conjecture holds if and only if f_i is nonlinear.

Proof: If f_i is linear we can group-code the inputs $\mathbf{X}_1, \mathbf{X}_2 \cdots \mathbf{X}_m$ and use $F_i = f_i$ and obtain arbitrarily high reliability if $R = k/n < 1 + p \log_2 p + (1 - p) \log_2 (1 - p)$.

If f_i is nonlinear, we can (by Lemma 4) find two variables x_i and x_j and a set of constants $c_r (r \neq i, j)$ such that $f_i(c_1, c_2 \cdots x_i \cdots x_j \cdots c_m) = g_i(x_i, x_j) \in C_2$. Consider the case when $\mathbf{X}_t = (c_t, c_t \cdots c_t)$ for all $t \neq i, j$, i.e., $f(\mathbf{X}_1, \mathbf{X}_2 \cdots \mathbf{X}_m) = g(\mathbf{X}_i, \mathbf{X}_j) \in C_2$. Then by Theorem 3, Elias' conjecture holds.

Q.E.D.

Of all the 2^{2^m} Boolean functions of m variables only 2^{m+1} functions are linear, namely all functions which can be represented as $f(x_1, x_2 \cdots x_m) = K_0 \oplus \sum_{i=1}^{m} K_i x_i$ for some K_i's. Of those 2^{m+1} functions only 2 are explicit functions of all the m variables, namely $\sum_{i=1}^{m} x_i$ and $1 \oplus \sum_{i=1}^{m} x_i$. Thus we see that Elias' conjecture holds for almost all Boolean functions.

Discussion

The investigation of the reliability of codes for logical operations carried out in this paper dealt with their error-correcting capabilities. The underlying assumption was that as the reliability of a code increases, s, the number of errors which can always be corrected increases, and therefore $k/n \leqq (2s + 1)^{-1}$ decreases.

We saw that most Boolean functions are similar (in the sense of Lemmas 2 and 4) to the AND function, and therefore any results about coding for the logical operation AND will hold for all the nonlinear Boolean functions. It might be easier to investigate coding for the operation AND because Lemma 1 gives us some

information about the structure of the codes for this operation.

The results obtained in this paper do not rely on the assumption that the effect of the noise can be represented by a memoryless binary symmetric channel, but follow from the restrictions imposed on the computation system. These restrictions were:

1. The inputs $\mathbf{X}_1, \mathbf{X}_2 \cdots \mathbf{X}_m$ are to be encoded independently of each other.
2. In the absence of noise, the function performed by the decoder is one-to-one.
3. The logical operator F operates on the inputs bit by bit.

In a forthcoming paper by S. Winograd and J. D. Cowan, it is shown that relaxing any of these three assumptions can lead to more positive results concerning the possibility of performing reliable computation in the presence of noise, at nonzero rate of flow of information.

Acknowledgment

The author wishes to express his gratitude to Dr. C. C. Elgot for his many helpful discussions. Thanks are also due to J. D. Cowan who helped to clarify many points concerning coding for logical operations.

References

1. J. Von Neumann, "Probabilistic Logics," in *Automata Studies*, Ed. by C. E. Shannon and J. McCarthy, Princeton University Press, 1956, pp. 43-98.
2. E. F. Moore and C. E. Shannon, "Reliable Circuits Using Less Reliable Relays," *J. Franklin Inst.* **262**, 191 (1956).
3. S. Muroga, "Preliminary Study of the Probabilistic Behavior of a Digital Network with Majority Decision Elements," RADC-TN-60-146, August 1960.
4. Technical Session 1, *Bionics Symposium*, September 1960.
5. P. Elias, "Computation in the Presence of Noise," *IBM Journal* **2**, 346 (1958).
6. C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Tech. J.* **27**, 379 (1948).
7. D. Slepian, "A Class of Binary Signalling Alphabets," *Bell System Tech. J.* **35**, 203 (1956).
8. R. W. Hamming, "Error Detecting and Error Correcting Codes," *Bell System Tech. J.* **29**, 147 (1950).
9. R. C. Bose and D. K. Roy-Chandhuri, "On a Class of Error-Correcting Binary Group Codes," *Information and Control* **3**, 68 (1960).
10. W. W. Peterson and M. O. Rabin, "On Codes for Checking Logical Operations," *IBM Journal* **3**, 163 (1959).

Received November 13, 1961