Article Title: AN Codes for Arbitrarily Large Distances

Author's Names: P. M. Monteiro, R. W. Newcomb, and T. R. N. Rao

Affiliation: Electrical Engineering Department, University of Maryland, College Park, Maryland 20742

Name and Address for Correspondence:

Professor R. W. Newcomb

Electrical Engineering Department

University of Maryland

College Park, Maryland 20742

List of Symbols (in order of appearance)

AN

$\prod_{j=1}^{r}(2^{m_j}-1)$

d_{min}

$>$

$<$

s

r

Z

Z^+

Z^-

N

ϵ

a_i

b_i

NAF

$W(\cdot)$

Z_M

M

$\equiv$

$\overline{N}$

$\bmod M$

$\cup$

$\{\cdot\}$

$AW(\cdot)$

$D_M(\cdot,\cdot)$

$C(M,A)$

x_i

N_R

$|\cdot|_A$

E

$S(\cdot)$

$\underline{b}$

$\forall$

$G(M,\underline{b})$

$\oplus$

c

X

Y

y_i

$\mathcal{S}(\cdot)$

(m_i, m_j)

n_o

I_i

L

ℓ_i

A'

N'_R

d'_{min}

d

t

M_R

m

IR

IR'

IR_R

$\log_2$

k

R_i

$\hat{S}(\cdot)$

$\underline{0}$

Number of pages: 21

Number of tables: one

Number of figures: one

Running head:

Large Distance AN Codes

AN Codes for Arbitrarily Large Distances[+]

P. M. Monteiro*, R. W. Newcomb*, and T. R. N. Rao*

Abstract:

A systematic method of obtaining an AN code with minimum distance within a specified range is presented. It is shown that when A is of the form $A = \prod_{j=1}^{r} (2^{m_j}-1)$ a minimum distance d_{min} in the range $2^s < d_{min} < 2^{s+1}$, for any $s<r$, can be obtained by suitably choosing the m_j ($j=1, 2, \ldots, r$). Sufficient conditions for obtaining the minimum distance are established and information rates for various choices of m_j ($j=1, 2, \ldots, r$) and s are tabulated. It is seen that for $s<1$ these codes are non-cyclic. A significant advantage, however, is that the form of A allows for ease of syndrome generation by using residue generators modulo $2^{m_j}-1$. Error correction properties are discussed and implementation considered for either direct coding or for conversion to multi-residue codes with check bases of the convenient from $2^{m_j}-1$.

- - - - - - - - - - - - -

*Electrical Engineering Department, University of Maryland, College Park, Maryland 20742.

+This work was supported in part by the Air Force Office of Scientific Research under Grant AFOSR 70-1910.

"I am tempted to write a poem" Neto (1968)

I. Introduction

In the design of digital computers with internal error correction it is desirable to have available arithmetic codes which are a) capable of multiple error correction and b) relatively easily implemented. Here we discuss a systematic method of obtaining such AN codes of arbitrarily large distances suitable for implementation through the use of simple residue generators, perhaps in terms of their multiresidue equivalents.

Previous studies on AN codes have led to the discovery of certain classes of multiple error correcting cyclic AN codes, such as codes of Barrows (1966) and Mandelbaum (1967), and those of Chien and Hong and Preparata (1971). Though exact expressions for the minimum distance have been obtained for the former class of codes, Chang and Tsao-Wu (1968), no systematic method is available for the calculation of minimum distance of the latter class. Besides, there still remains a large gap between the theory and implementation of large distance codes of these cyclic kinds.

One of the main problems in the implementation of AN codes arises because of the complicated encoding and syndrome generation procedures. Encoding of an AN code involves multiplying the information integer N by the fixed integer A, whereas syndrome generation consists of obtaining the residue of a result modulo A (which, in general, is not easy for arbitrary A). Here we will primarily consider those A whose factors allow ease of implementation.

A desirable feature of a code is the separability of information and check digits, so that simple encoding and decoding circuits can be designed. But, in general, AN codes yield a non-separate form of coding such that there is no way to distinguish between information and check digits. In contrast, multiresidue codes, first introduced by Dadayev (1965) and later studied by Rao (1970) and Rao and Garcia (1971) are separate codes, which can be directly derived from AN codes, and are implemented by means of a processor working in parallel with an appropriate number of checkers. If the check bases are of suitable form, the implementation is fairly straightforward. However, the use of multiresidue codes necessitates the consideration of a new factor, viz. the reliability of individual checkers, and this raises some difficult questions, as we shall see later.

The codes presented in this paper can be shown to have arbitrarily large distances for suitable choices of the factors of A, and, in addition, have the advantage that they have relatively straightforward procedures for encoding and syndrome generation.

The next section will be devoted to a brief review and outline of the background material necessary for the discussion to follow. Section III gives a discussion of the necessary conditions for an AN code with A of the form $\prod_{j=1}^{r}(2^{m_j}-1)$ to have a minimum distance $d_{min}>2^s$ for some $s<r$. Section IV investigates the sufficiency of the above necessary conditions with no additional constraint on the various parameters. Section V contains the main contribution of this paper, which is a theorem establishing sufficient conditions for an AN code with $A=\prod_{j=1}^{r}(2^{m_j}-1)$ to have a distance greater than 2^s for any $s<r$; the resulting codes are noncyclic except when $s=1$. Section VI discusses the relationship of such codes to repetition codes and their information rates. The error correction properties of the constructed codes are investigated in Section VII while finally Section VIII presents a discussion on implementation of these codes.

"A poem that would not be letters
But blood alive
In the pulsating arteries of the mathematical universe." Neto (1968)

II. Preliminaries

We take as our starting point material covered in Peterson & Weldon (1972), [pp. 451-465] and/or Massey and Garcia (1971), these treatments primarily being for cyclic codes. However, the codes discussed here are generally noncyclic for which our review in this section contains appropriate modifications.

Let Z, Z^+ and Z^- represent the ring of integers, the positive integers, and the nonpositive integers, respectively. Then given an integer N it has a unique representation in radix-2 form

$$N=\sum_{i=0} a_i 2^i \quad \text{with} \begin{cases} a_i=0,1 \text{ for } N\in Z^+ \\ a_i=0,-1 \text{ for } N\in Z^- \end{cases} \tag{2.1}$$

If one allows a_i of both signs simultaneously, this representation becomes nonunique since

$$2^j = 2^{j+1}-2^j \tag{2.2}$$

But calculation errors can readily introduce coefficients of both signs, so that of most use in arithmetic coding is the unique representation in nonadjacent form (NAF), Reitweisner (1960)

$$N = \sum_{i=0} b_i 2^i \quad , \quad b_i b_{i+1} = 0 \quad \text{with } b_i = 0, \pm 1 \text{ for } N \in Z \tag{2.3}$$

The NAF can be found by repeated application of (2.2) or from the radix-2 form of 3N and N using $N = (3N-N)/2$, Tsao-Wu and Chang (1969). Hence, given an integer $N \in Z$ we can now define the weight $W(N)$ as the number of nonzero terms, b_i, in the NAF of N. Note that $W(N) = W(-N)$, $W(N_1+N_2) \leq W(N_1)+W(N_2)$ and $W(N) = 0$ implies $N = 0$. Consequently, $W(N_1-N_2)$, known as the arithmetic distance, serves to define a metric, on Z making it into a metric space, Kelley, Namioka, et al (1963) [p. 29].

Most physical implementations work, however, over a finite ring Z_M of integers modulo a positive integer M. In such cases there are two possible interpretations of $N \in Z_M$ when considered as a number $\dot{N} \in Z$; these are N itself and the negative of the complement of N, the complement $\overline{N}$ of N being defined by

$$-N \equiv \overline{N} = (M-N) \bmod M \tag{2.4}$$

where $\overline{N} \in Z^+ \cup \{0\}$, if $N \in Z$. Since it may occur that $W(N) \neq W(-\overline{N}) = W(\overline{N})$ and since we often wish to consider N and $-\overline{N}$ equivalently for error correction we are led to define the modular arithmetic weight, Rao and Garcia (1971) [p. 89],

$$AW(N) = \min \{ W(N), W(\overline{N}) \} \qquad N \in Z_M \tag{2.5}$$

Note that $AW(\cdot)$ depends upon M, since the complement does, though we omit this from the notation. Using $AW(\cdot)$ we next define the modular arithmetic distance

$$D_M(N_1, N_2) = AW(N_1-N_2) \quad , \quad N_1, N_2 \in Z_M \tag{2.6}$$

The modular arithmetic distance is useful in characterizing error correction properties, but only in special cases is it known to define a metric on Z_M, Massey and Garcia (1971) [p. 288].

The discussion in the remaining sections centers on the distance of AN codes of a given length, so we proceed with definitions and discussions on these terms.

By definition, an AN code in Z_M is the set

$$C(M,A) = \{ x \mid x \equiv AN \bmod M \text{ for all } N \in Z \} \tag{2.7}$$

generated by A of modulus $M = AN_R$, for fixed $A, N_R \in Z^+$. If we support $C(M,A)$ with the algebraic operations of Z_M, as shall be assumed, then the AN code becomes an ideal in Z_M. The AN code's information range is the integer $N_R = M/A$ while the code-words are the integers AN for $0 \leqslant N < N_R$. In some situations it is convenient to extend the definition of an AN code to include the empty code, in which $N_R = 0$ is allowed; such a code is called imaginary. The (minimum) distance d_{min} of an AN code is the minimum of the modular arithmetic distances between all distinct code-words, that is

$$d_{min} = \min D_M(x_i, x_j) \text{ for all } x_i \neq x_j \in C(M,A) \tag{2.8a}$$

$$= \min AW(x_i) \text{ for all nonzero } x_i \in C(M,A) \tag{2.8b}$$

where the latter follows since the difference of two code-words is a code-word, $x_j - x_k = x_i \in C(M,A)$, and vice versa. If the code is imaginary, that is $C(M,A) = \varphi$, the empty set, then we take $d_{min} = 0$. By the code-length n of an AN code is meant the minimum integer n for which $M = AN_R < 2^n$.

The whole purpose of coding is to detect and/or correct errors, where for an AN code an error is an element of the set $Z_M - C(M,A)$. Thus, given an element $x \in Z_M$ an error is detected if $|x|_A \neq 0$ where we introduce the notation $|\ |_A$ for mod$A$. If error correction is desired by means of an AN code an element $E \equiv 0 \bmod A$ is decoded into a code-word in $C(M,A)$ with the help of its syndrome $S(E)$, where for any $X \in Z$ we define $S(X) = |X|_A$. Associated with a nonzero syndrome is a rule which relates $S(E)$ to the error value giving a correction which may then be subtracted from E. Error correction of all errors having an upper bound on their modular arithmetic weights is possible if each such error

has a distinct syndrome with respect to A so that there is no ambiguity in obtaining the error value. For example, when $AW(\cdot)$ serves to define a metric on Z_M, if $d_{min} \geq 2t+1$ then syndromes can be assigned such that the AN code-decodes $x \in Z_M$ to AN_x which minimizes $D_M(x, AN_x) \leq t$, if such exists; otherwise it flags that an error has been detected. In this case all errors E are corrected having $AW(E) \leq t$, Massey and Garcia (1971) [p. 291].

Since AN codes are at times practically implemented as multiresidue codes we end this section with a discussion of the latter. By definition a <u>k-residue code</u> $G(M, \underline{b})$, $\underline{b} = (b_1, b_2, \ldots, b_k)$, is the set of $(k+1)$-tuples ($\forall$ denotes "for all")

$$G(M, \underline{b}) = \{(x, x_1, x_2, \ldots, x_k) \mid x_i = |x|_{b_i} \text{ for } i = 1, \ldots, k, \forall x \in Z_M\} \tag{2.9}$$

The b_i $(i = 1, 2, \ldots, k)$ are called the <u>check bases</u>, M is the code modulus and the $(k+1)$-vector $(x, \underline{x})$, $\underline{x} = (x_1, x_2, \ldots, x_k)$, is called the multiresidue <u>code-word</u>. As an example,

$$G(7, (3, 5)) = \{(0, 0, 0), (1, 1, 1), (2, 2, 2), (3, 0, 3), (4, 1, 4), (5, 2, 0), (6, 0, 1)\} \tag{2.10}$$

Addition, $\oplus$, of two multiresidue code-words $X = (x, \underline{x})$ and $Y = (y, \underline{y})$ is defined in $G(M, \underline{b})$ by

$$X \oplus Y = (|x+y|_M, |x_1+y_1 - cM|_{b_1}, \ldots, |x_k+y_k - cM|_{b_k}) \tag{2.11a}$$

where

$$c = \begin{cases} 0 & \text{if } x+y < M \\ 1 & \text{if } x+y \geq M \end{cases} \tag{2.11b}$$

As a consequence, the multiresidue code is closed under addition. Further, if b_i divides M for all i then,

$$X \oplus Y = (|x+y|_M, |x_1+y_1|_{b_1}, \ldots, |x_k+y_k|_{b_k}) \tag{2.11c}$$

Practically, a multiresidue code is implemented by a processor working in parallel with k checkers. Operations in the processor are carried out modulo M and those in checker i modulo b_i for $i = 1, \ldots, k$. Thus, it is evident why multiresidue codes are called <u>separate</u> codes. The <u>syndrome</u> of a separately coded word $X = (x, \underline{x})$ is defined as the k-vector

$$\mathcal{S}(X) = \underline{s} = (s_1, s_2, \ldots, s_k) \tag{2.12a}$$

where

$$s_i = |x - x_i|_{b_i} \qquad i = 1, 2, \ldots, k \tag{2.12b}$$

For example, for the code of (2.10) we have

$$\mathcal{S}((6, 0, 1)) = (|6 - 0|_3 , |6 - 1|_5) = (0, 0) \tag{2.13}$$

It is easily seen that the syndrome of a multiresidue code-word is the 0 vector. However, if an error e occurs in the processor it yields the syndrome

$$\mathcal{S}(X \oplus (e, \underline{0})) = (|e|_{b_1}, \ldots, |e|_{b_k}) , \quad X \in G(M, \underline{b}) , \; e \in Z_M \tag{2.14}$$

where X is the correct multiresidue code-word.

It is shown in Rao and Garcia (1971) that for every AN code $C(M, A)$ there exists a multiresidue code $G(M, \underline{b})$ with $A = \text{LCM}\{b_1, \ldots, b_k\}$ such that every error that is correctable by $C(M, A)$ is correctable by $G(M, \underline{b})$, that is, if unique syndromes exist for such errors in the former case the same is true for the latter case. This will be used further in Section VIII but we do mention here that the information range of the multiresidue code is $0 \leqslant N < AN_R$ in contrast to $0 \leqslant N < N_R$ for the corresponding AN code.

"And would be stars scintillating
For calm nights" Neto (1968)

III. Necessary Conditions for $d_{min} > 2^s$

Necessary conditions for achieving large d_{min} by general AN codes have been stated by Kondrat'yev & Trofimov (1969) [p. 86], though they scarcely sketched a proof. We comment that what is of interest, once an A is given, is the choice of code modulus M, or equivalently the code length n.

We use the standard notation $(m_i, m_j) = 1$ to mean that m_i and m_j are relatively prime.

THEOREM 1:

Let an AN code be formed with

$$A = \prod_{j=1}^{r} (2^{m_j} - 1) \tag{3.1a}$$

having the m_j pair-wise relatively prime (and ordered), that is,

$$(m_i, m_j) = 1,\ m_i > m_j > 1 \text{ for all } i > j \text{ for all } j \in \{1, \ldots, r-1\} \tag{3.1b}$$

Then <u>necessary</u> conditions for $d_{min} > 2^s$ are that $r > s$ and the code length satisfies

$$n \leq n_o = \text{minimum of } \left(\prod_{j \in I_1} m_j + \prod_{j \in I_2} m_j + \ldots + \prod_{j \in I_s} m_j \right) \tag{3.2}$$

over all nonempty disjoint partitions of $\{1, \ldots, r\} = \bigcup_{i=1}^{s} I_i$

Proof: We first observe that A is a code-word and on multiplying out the terms of (3.1a) that there are 2^r terms, or perhaps fewer in its NAF, in which case $2^s < d_{min} \leq AW(A) \leq 2^r$, by (2.8b); hence, necessarily $s < r$.

Consider next the integer, for some partition of $\{1, \ldots r\}$,

$$L = \prod_{i=1}^{s} (2^{\ell_i} - 1) \quad \text{where} \quad \ell_i = \prod_{j \in I_i} m_j \tag{3.3}$$

where the nonempty, nonintersecting sets of integers I_i contain all integers $1, \ldots r$, that is, $\bigcup_{i=1}^{s} I_i = \{1, \ldots, r\}$. We observe that L is divisible by A and hence a candidate for a code-word and/or modulus. However, $AW(L) \leq 2^s$, since L has at most 2^s terms in its NAF expansion, in which case we cannot have L as a code-word. Consequently, since L is divisible by A, $AN_r \leq L$ while $AN_r < 2^n$ by definition of n . Partitioning the integers such that the highest exponent of L , $\prod_{j \in I_1} m_j + \prod_{j \in I_2} m_j + \ldots + \prod_{j \in I_s} m_j$, is minimized gives the minimum such L; this exponent is n_o of (3.2) . Since the next highest power of 2 in L is subtractive we see that the greatest lower bound on n is n_o , i.e. $AN_r \leq \min L < 2^{n_o}$ from which (3.2) necessarily follows. Q.E.D.

It is worth commenting that (3.1b) is merely for convenience. Indeed if the m_j are not relatively prime then the same theorem holds except that πm_j is replaced by $LCM\{m_j\}$ in (3.2). However, the use of m_j which are not relatively prime leads practically to inefficient coding so is scarcely considered in the following. We illustrate the theorem numerically for a simple but interesting case.

EXAMPLE 1:

Let it be desired to create an AN code with $d_{min} > 2^2 = 4$ using the smallest possible A of the form of (3.1). Then $r = 3$ and we wish, in the first instance to consider $m_1 = 2$, $m_2 = 3$, $m_3 = 5$ or

$$A = (2^2 - 1)(2^3 - 1)(2^5 - 1) = 651 = 2^9 + 2^7 + 2^4 - 2^2 - 1$$

where the right side is the NAF, showing $W(A) = 5$. The set of L's,

(3.3), are

$$L_1 = (2^{2\cdot 3}-1)(2^5-1) = 2^{2\cdot 3+5} - 2^5 - 2^{2\cdot 3} + 1 = 2^{11} - 2^7 + 2^5 + 1 = 1953 = 3\cdot A$$

$$L_2 = (2^{2\cdot 5}-1)(2^3-1) = 2^{2\cdot 5+3} - 2^3 - 2^{2\cdot 5} + 1 = 2^{13} - 2^{10} - 2^3 + 1 = 7161 = 11\cdot A$$

$$L_3 = (2^{3\cdot 5}-1)(2^2-1) = 2^{3\cdot 5+2} - 2^2 - 2^{3\cdot 5} + 1 = 2^{17} - 2^{15} - 2^2 + 1 = 98301 = 151\cdot A$$

Thus the minimum L is L_1 from which we see that

$$n_o = 11 = 2\cdot 3 + 5 = \min \begin{cases} m_1 m_2 + m_3; I_1 = \{1, 2\}, I_2 = \{3\} \\ m_1 m_3 + m_2; I_1 = \{1, 3\}, I_2 = \{2\} \\ m_2 m_3 + m_1; I_1 = \{2, 3\}, I_2 = \{1\} \end{cases}$$

Consequently, we require $M = AN_R = 651N_R < 2048 = 2^{11}$. The maximum N_R satisfying this inequality is $N_R = 3$. Choosing this N_R gives $M = 3A$ and the AN code is $C(3A, A) = \{0, A, 2A\}$ in the ring $Z_{3A} = \{0, 1, \ldots, 3A-1\}$. In this ring $\bar{A} = 2A$ and always $W(A) = W(2A)$. Hence, $d_{min} = 5 > 2^2$ for the largest possible n, $n = n_o = 11$, and thus, the conditions of Theorem 1 are seen to also be sufficient in this case.

"Of winters rainy and cold" Neto (1968)

IV. Insufficiency of the Necessary Conditions

As will be used extensively later, in the special case of $s = 2$ the necessary conditions of Theorem 1 are known from Kondrat'yev and Trofimov (1969), [p. 90], to be sufficient to guarantee $d_{min} > 4 = 2^s$. Consequently, the code of Example 1 does have $d_{min} > 4$, as has already been seen by inspection. However, we show here by (counter-) example that for $s > 2$ the conditions of Theorem 1 need not be sufficient for any $n > 0$.

EXAMPLE 2:

Consider an AN code with $m_1 = 3$, $m_2 = 4$, $m_3 = 5$, $m_4 = 7$, that is,

$$A = (2^3-1)(2^4-1)(2^5-1)(2^7-1) = 2^{19} - 2^{17} + 2^{14} + 2^{12} - 2^8 - 2^6 + 2^3 + 2^0$$

where the NAF on the right shows that $W(A) = 8$. Consequently, any AN code generated by A must have $d_{min} \leq 8$. However, if the conditions of Theorem 1 were sufficient they would allow $d_{min} > 8 = 2^3$, since $s = 3 < 4 = r$ would be possible for some $n \leq n_o = 3 \cdot 4 + 5 + 7 = 24$ calculated according to (3.2).

We, therefore, conclude that in general additional constraints on the m_j are necessary to obtain $d_{min} > 2^s$ for some n satisfying (3.2). Indeed since we would desire at least one nonzero code-word, meaningful n are $n > \sum_{j=1}^{r} m_j$, the highest exponent in A.

"And would be light to greet the gazelles
That graze insecure
In the fields that host immense life" .Neto (1968)

V. Sufficiency Results

Our main result, Theorem 2 of this Section, will give a constructive method for designing a code with $d_{min} > 2^s$ for any s. Because of their multiresidue implementation we are most interested in those A having the form given in (3.1a). However, the primary result of Theorem 2 is more general and hence given for arbitrary A.

THEOREM 2:

Given an AN code with minimum distance d_{min}, length n and information range N_R, the A'N code formed with

$$A' = A \cdot (2^m - 1) \tag{5.1a}$$

has minimum destance $d'_{min} = 2\, d_{min}$, with $N'_R = N_R$ for a code length $m+n$ if

$$m \geq n+1 \tag{5.1b}$$

Proof: Let $d = d_{min}$ and $A \cdot N = 2^{j_1} \pm 2^{j_2} \pm \ldots 2^{j_{d+t}}$ be any codeword in NAF for the code generated by A, where $j_1 \leq n$ and $j_{d+t} = 0$, $t \geq 0$. Then

$$A' \cdot N = A \cdot N \cdot (2^m - 1) = 2^{j_1 + m} \pm 2^{j_2 + m} \ldots \pm 2^{j_{d+t} + m} \mp 2^{j_1} \mp 2^{j_2} \ldots \mp 2^{j_{d+t}} .$$

Now, since $m \geq n+1$ and $j_1 \leq n$, $j_{d+t} + m - j_1 \geq 1$. This all the $2(d+t)$ terms of $A' \cdot N$ are nonadjacent, since $A \cdot N$ was originally in NAF, and $A' \cdot N$ consists of $A \cdot N$ shifted by m places added to $-A \cdot N$ (which has the same weight as $A \cdot N$). Thus, $AW(A' \cdot N) = 2(d+t)$.

Thus, given any codeword $A \cdot N$ in the original code, we can show that its weight is doubled when multiplied by $2^m - 1$. We also observe that the information range of the new A'N code is the same as that of the original AN code, from which we conclude that every A'N codeword can be written as $A \cdot N \cdot (2^m - 1)$. Since every codeword is the original AN code has a weight at least d, the minimum distance of the A'N code is seen to be at least $2d$. Q.E.D.

To achieve a construction of large distance codes having A of the desired form of (3.1a) we iteratively apply (5.1) to an initial A formed according to the known, but little recognized, results for $d_{min} > 2^2$ of Kondrat'yev and Trofimov (1969). We thus state the latter here for reference but without proof. We do point out that the code-length definition we use is numerically one greater than that used by Kondrat'yev and Trofimov in their proof.

THEOREM 3:

Let an AN code be formed with

$$A = \prod_{j=1}^{r} (2^{m_j} - 1) \tag{3.1a}$$

having

$$(m_i, m_j) = 1 \quad , \quad m_i > m_j > 1 \quad \forall i > j \quad \forall j \in \{1, , , r-1\} \tag{3.1b}$$

Then $d_{min} > 4$ for all $r \geq 3$ and all positive n satisfying

$$n \leq n_o = \text{minimum of} \left(\prod_{j \in I_1} m_j + \prod_{j \in I_2} m_j \right) \tag{3.2'}$$

over all nonempty disjoint partitions of $\{1, \ldots, r\} = I_1 \cup I_2$

In other words $n \leq n_0$ is a necessary and sufficient condition for $d_{min} > 2^s$ for $2 = s < r$. Using any A and n satisfying Theorem 3, Theorem 2 can be iteratively applied to obtain AN codes with A of the form of (3.1a) and having $d_{min} > 2^s$ for any integer $s \geq 2$.

Theorems 2 and 3 give a lower bound on d_{min} for the chosen code. In some cases it may be of interest to also fix an upper bound. For this we can fix the code modulus $M = AN_R$ such that a code-word of weight 2^{s+1} occurs within the code for $d_{min} > 2^s$; such a code-word is

$$AN = \prod_{i=1}^{s+1} (2^{\prod_{j \in I_i} m_j} - 1) \tag{5.2}$$

where the sets I_i are those for which the minimum of $\sum_{i=1}^{s+1} (\prod_{j \in I_i} m_j)$ occurs when taken over all $s+1$ nonempty disjoint partitions of $\{1, 2, \ldots, r\} = I_1 \cup \ldots \cup I_{s+1}$. Consequently, we can obtain $2^{s+1} \geq d_{min} > 2^s$.

The $s = 1$ situation, where $n_0 = \prod_{j=1}^{r} m_j$, is worth a comment. In this case the necessary conditions of Theorem 1 are seen to be sufficient also. Likewise, in this $s = 1$ case, $d_{min} = 4$ is known, Kondrat'yev and Trofimov (1969) [p. 86] (though some $m_j > 3$ must be held, Monteiro (1972)). The $s = 1$ code is of some interest since it is the only cyclic code of this class.

Some example codes are given in Table 1, along with their information rates, as defined in the next section.

"A motor that impels the impossible
Toward the reality of the hours;
A harmonious chant to the magnificance of man." Neto (1968)

VI. Relation to Repetition Codes - Information Rate

The codes treated here can be looked upon somewhat as repetition codes, Massey and Garcia (1971) [p. 305], with, however, a change in sign in the repetition. To see this we define an A'N code as one formed via Theorem 2; we then note that every A'N codeword is of the form

$$A' \cdot N = A \cdot N \cdot (2^m - 1) = A \cdot N \cdot 2^m - A \cdot N \tag{6.1a}$$

In the right hand expression there is, in view of the constraint $m \geq n+1$, no overlapping of the two terms; that is, the term $AN\,2^m$ is a repetition of -AN with a sign change. In contrast a true repetition code, called here an $A_R N$ code, would have

$$A_R \cdot N = A \cdot N \cdot (2^m + 1) \tag{6.1b}$$

With the choice $m = n$ the latter yields cyclic $A_R N$ codes when the original AN code is cyclic, since $M_R = A_R \cdot N_R = A \cdot N_R (2^m + 1) = (2^n - 1)(2^n + 1) = 2^{2n} - 1$. This is in contrast to the situation with the A'N codes which are noncyclic.

From the repetitive nature, it is clear, on observing (6.1) that the distances of the $A_R N$ and A'N codes are twice those of the original AN code when subject to $m \geq n+1$ (the $+1$ in $n+1$ being to preserve nonadjacency).

In terms of information rate, for the same choice of m the A'N codes are just slightly better than the $A_R N$ codes. Thus, on defiing the information rate by, Massey and Garcia (1971) [p. 296],

$$IR = \frac{\log_2 N_R}{\log_2 M} \tag{6.2}$$

we have for the codes of Theorem 2

$$IR' = \frac{\log_2 N_R}{\log_2 A + \log_2 N_R + \log_2 (2^m - 1)} \tag{6.3a}$$

while for the corresponding repetition code

$$IR_R = \frac{\log_2 N_R}{\log_2 A + \log_2 N_R + \log_2 (2^m + 1)} \tag{6.3b}$$

Thus, for the same value of m always $IR' > IR_R$ though the difference for large code lengths is negligible.

"A poem closed within itself
To be understood
From the brightness of the sky
And from the upright character of man" Neto (1968)

VII. Error Correcting Properties

The relation between the minimum distance of an AN code in the infinite ring, Z, and the number of errors it can correct (or detect) has been given by Massey (1964), [p. 7]. In his proof of the result Massey has used the fact that the integer ring Z is a metric space with the arithmetic distance as the metric. More recently, Massey and Garcia (1971), [p. 290], have shown that for AN codes in certain finite rings, the modular arithmetic distance is still a metric function if M, the code modulus (or ring modulus) is of the form 2^k or $2^k \pm 1$. Hence, all such codes have the same relation between the minimum distance and error correcting properties as those in infinite rings. However, the general question, as to what happens if the modular arithmetic distance is not a metric, is still not completely answered. Moreover the relationship between faults in equipment versus errors as mathematically defined here needs investigation, particularly when there are more than one end-around carries. We will, however, now establish the relation between minimum distance and error correction capability when M is of the form

$$M = (2^{\pi_{j\in I_1} m_j} - 1)(2^{\pi_{j\in I_2} m_j} - 1) \cdots (2^{\pi_{j\in I_s} m_j} - 1) \qquad (7.1)$$

with the m_j partitioned as for n_o of (3.2) in Theorem 1.

We will first show, by means of an example, that when M is of the form (7.1), the modular arithmetic distance is not necessarily a metric.

EXAMPLE 3:

Let $M = (2^5-1)(2^6-1) = 1953$ and consider $N_1 = N_2 = 929$ and $N_3 = N_1 + N_2 = 1858$. Then $AW(N_1) = AW(N_2) = \min\{W(929), W(1953-929)\} = 1$ and $AW(N_3) = \min\{W(1858), W(95)\} = 3$ hence

$$3 = AW(N_1 + N_2) > AW(N_1) + AW(N_2) = 2$$

Consequently, the modular arithmetic weight on Z_{1953}, or equivalently the arithmetic distance, does not satisfy the triangle inequality which is an essential property of a metric.

Even though modular arithmetic distance does not define a metric on Z_M the error correcting properties are essentially those of a metric space under a slight modification of the conditions of Theorem 3. These modifications and the relationship of the error correcting properties to the minimum distance are now presented.

THEOREM 4:

Let an AN code have $d_{min} > 2^s$ with

$$A = \prod_{j=1}^{r} (2^{m_j} - 1) \tag{3.1a}$$

and having $r > s > 1$ with

$$(m_i, m_j) = 1, \quad m_i > m_j > 1 \quad \forall i > j \quad \forall j \in \{1, 2, \ldots, r-1\} \tag{3.1b}$$

be formed according to Theorems 2 and 3 using $m \geq n+2$ at each step of application of Theorem 2, then the code is capable of correcting all errors of weight 2^{s-1} or less if M, the modulus of the code, satisfies (7.1).

Outline of Proof: We will only outline the proof here, as details which are lengthy, are provided in Monteiro (1972). Since distinct syndromes lead to distinct errors of weight 2^{s-1} or less, the proof consists in showing that any two errors having weights 2^{s-1} or less have distinct (modulo A) syndromes. The condition for this results by showing that all multiples of A in the range $M < AN < 2M$ have an arithmetic weight greater than 2^s (those in $0 < AN < M$ do by assumption); the upper bound $2M$ results by considering the addition of two numbers less than M. This condition is first shown to be true for $s = 2$ and all r. Finally, by induction this condition is shown to be satisfied for all s, completing the proof.

By means of Theorem 4 we have established that the relation between the minimum distance and error correction capabilities of such codes is the same as that for AN codes in infinite rings, hence, the modulus M can be used in the implementation, which we shall now discuss.

"To the beauty of virgin forests
And the precision of gears, of existence
Over the barbaric rattling of machines
And the aspiration of man" Neto (1968)

VIII. Code Implementation

The implementation of AN codes involves three main aspects: (1) the encoding of an AN code consists of multiplying the information integer N by the code generator A. (2) Syndrome generation consists of obtaining the residue of the result modulo A. (3) Syndrome decoding and error correction involves determining the error magnitude and polarity from the form of the syndrome. The error is then subtracted from the result. Syndrome decoding is by far the most complicated of the three steps involved in the implementation of AN codes.

In the following treatment we shall briefly point out the advantage of using A of the form (3.1a), that is, $A=\prod_{j=1}^{r}(2^{m_j}-1)$, in the processes of encoding and syndrome decoding. Finally, we will discuss the implementation of these codes as multiresidue codes and the limitations of such an implementation.

Encoding. Generally, for arbitrary A, a multiplier circuit is needed for forming the product $A \cdot N$. However, in the case where A is of the form in (3.1a) we can perform the operation by means of the schematic presented in Fig. 1. Using Fig. 1 the operation of multiplying N by A takes r cycles where r is the number of factors of A. In each cycle the number R_i is shifted by the variable shifter to form $2^{m_i}R_i$. Then both R_i and its shifted value are fed into the subtractor, to form $R_{i+1}=(2^{m_i}-1)R_i$. The gating is accomplished by using a D-flip-flop array which on the arrival of clock pulse p_i holds R_i at its output until the arrival of p_{i+1} at which time R_{i+1} is transferred through. The heavy lines in the figure indicate multiple bit signals to represent the numbers being handled.

Syndrome Generation. We recall from Section II that the AN code syndrome of a number X is defined as $S(X) = |X|_A$. A nonzero syndrome indicates an error in the result X . It is important to note with Sitnichenko (1970) and Peterson and Weldon (1972) that practically obtaining the residue of a number X modulo A can be cumbersome if A is not of either of the forms 2^k or 2^k-1 (which occurs with $r>1$ for our A). This being the case, we proceed by first finding the residues of X modulo the factors of A to form a vector $\hat{S}(X)$ to which the Chinese Remainder Theorem, LeVeque (1958), can be applied to yield the actual syndrome $S(X)$.

For $A = \prod_{j=1}^{r} (2^{m_j}-1)$ we define the r-vector $\hat{S}(X)$ by

$$\hat{S}(X) = (|X|_{b_1}, |X|_{b_2}, \ldots, |X|_{b_r}) \quad , \quad b_j = 2^{m_j}-1 \tag{8.1}$$

Because of the form of A , $\hat{S}(X)$ is easier to generate than $S(X)$. Indeed each of the components $|X|_{b_j}$, $j=1, \ldots, r$, is readily generated at relatively low cost using the well-known "tree" method, Sellers, Hsiao and Bearnson (1968) [p. 79], for finding the residue of a number modulo 2^k-1 for some k . The Chinese Remainder Theorem assures us that a given $\hat{S}(X)$ yields a unique $|X|_A = S(X)$ insuring that syndrome decoding for $\hat{S}(X)$ is equivalent to that for $S(X)$. Indeed $\hat{S}(X) = \underline{0}$ if and only if $S(X) = 0$ so that any error detected by $S(X)$ is detected by $\hat{S}(X)$ and similarly for error correction. Consequently, we need not really calculate $S(X)$ if $\hat{S}(X)$ is known.

Syndrome Decoding. This is a complicated problem for errors of large multiplicity. Tsao-Wu (1968) has suggested a method for the decoding of syndromes of cyclic arithmetic codes. However, his method cannot be applied as such here, as our codes are non-cyclic. The form of $\hat{S}(X)$ does enable us though to use a method similar to that described in Monteiro and Rao (1972) for multi-residue syndromes of single and double errors. In essence the method, which is detailed in Monteiro (1972) cycles a syndrome to a canonical double error form, the canonical forms being relatively few in number. The canonical double errors are stored in a read-only memory which is accessed when a canonical syndrome is reached by cycling. The actual, double or single, error is found by a reverse shifting of a canonical error. For errors of greater

multiplicity we propose to stop at detection, as no simple method is known for decoding syndromes of errors of large multiplicity, and our limited present knowledge requires checking circuits that could be more complex than those being checked.

Multiresidure Implementation. For multiresidue codes the syndrome is generated in a manner similar to that for $\hat{S}(X)$, as shown by (2.12). Consequently, given an AN code with $A = \prod_{j=1}^{r} (2^{m_j}-1)$ the equivalent multiresidue code, which has $b_j = 2^{m_j}-1$ as the moduli for its checkers, has the advantages of syndrome generation just mentioned for $\hat{S}(X)$. Likewise, the expansion of the code range from N_R to AN_R, as mentioned at the end of Section II, occurs. Encoding for these multiresidue codes is also relatively straightforward consisting of forming residues, also of moduli $b_j = 2^{m_j}-1$. However, when r is very large the question of failure of the checkers becomes as important, and as difficult, as that of the main processor. In fact, if we try to make what seem as reasonable assumptions regarding checker reliability and the number of units that can fail at any given instant, the information range drops sharply when $r > 3$, Monteiro (1972). Consequently, at this point in the development of the theory implementation of an AN code for $r > 3$ in its non-separate form is recommended, instead of the mutiresidue implementation.

However, for the case when $r = 3$, multiresidue implementation of an AN code does seem warranted. Such has been carried out in Monteiro and Rao (1972) for A of the form of (3.1a) thus enabling double error correction and triple error detection through use of the syndrome vector of (2.12) using check bases $b_i = 2^{m_i}-1$, $i = 1, 2, 3$.

"A poem traced over strength
Sculptured in Love
A poem solution
Resolving the interrogative curve of an image
In a straight line of affirmation." Neto (1968)

IX. Conclusions

The codes of Kondrat'yev and Trofimov (1969), as summarized in Theorem 3 are used as a basis for iteratively deriving, through Theorem 2, large distance

AN codes having A of the specialized form $A = \prod_{j=1}^{r} (2^{m_j}-1)$. However, Theorem 2 allows the consideration of other than the Kondrat'yev and Trofimov codes for initial choices in iterations. Thus, should more efficient base codes become available Theorem 2 can be used for increasing their distance to arbitrarily large values. As seen in section VI the codes presented are closely related to repetition codes, though as discussed for multiresidue implementation are more convenient for some purposes.

Because of the form of A the codes discussed are easily implemented as multiresidue codes where only residues modulo $2^{m_j}-1$, rather than modulo A, are evaluated for the checkers. Such an implementation seems quite practical for $r = 3$, as discussed in Section VIII, but, for larger r, problems in checker reliability seem to indicate that alternate means of implementation would be more profitable. Consequently, we propose at this point to encode these large distance AN codes by using Fig. 1 to obtain the multiplication, based upon r multiplications using the $2^{m_j}-1$, and then decoding through the syndrome equivalent vector $\hat{S}(X)$ of (8.1). Since syndrome decoding eventually comes down to "table look up" perhaps after shifting to canonical form, Monteiro (1972), it does presently seem impractical to correct more than two errors. Consequently, these large distance codes seem best adapted to the correction of at most two errors allowing then the detection of at least 2^s-4 other errors. In the end, however, it must be admitted that the problem of determining the actual error value, once an error is indicated, is a complicated problem, regardless of the class of codes used, when multiple errors are considered.

"A poem closed" Neto (1968)

References

Barrows, J.T., Jr. (1966), "A New Method for Constructing Multiple Error Correcting Linear Residue Codes," Report R-277, Coordinated Science Laboratory, University of Illinois, Urbana, January.

Berlekamp, E. R. (1968), "Algebraic Coding Theory," McGraw-Hill, New York.

Birkhoff, G., and MacLane, S. (1965), "A Survey of Modern Algebra," Third Edition, MacMillan, New York.

Chang, S-H., and Tsao-Wu, N. (1968), "Discussion on Arithmetic Codes with Large Distance," IEEE Transactions on Information Theory, Vol. IT-14, No. 1, pp. 174-176, January.

Chien, R. T., Hong, S. J., and Preparata, F. P. (1971) "Some Results in the Theory of Arithmetic Codes," Information and Control, Vol. 19, No. 3., pp. 246-264, October.

Dadayev, Y. G. (1965), "Arithmetic Divisible Codes with Correction for Independent Errors," Engineering Cybernetics, November, pp. 79-88.

Kelley, J. L., Namioka, I. et al. (1963), "Linear Topological Spaces," D. Van Nostrand, Princeton (New Jersey).

Kondrat'yev, V. N., and Trofimov, N. N. (1969), "Error-Correcting Codes with a Peterson Distance Not Less Than Five," Engineering Cybernetics, No. 3, pp. 85-91, May-June.

LeVeque, W. (1958), "Topics in Number Theory," Vol. 1, Addison-Wesley Publishing Co., Reading (Mass.), pp. 31-35.

Mandelbaum, D. (1967), "Arithmetic Codes with Large Distance," IEEE Transactions on Information Theory, Vol. IT-13, No. 2, pp. 237-242, April.

Massey, J. L. (1964), "Survey of Residue Coding for Arithmetic Errors," ICC Bulletin, Vol. 3, No. 4, October, pp. 1-17.

Massey, J. L., and Garcia, O. N. (1971), "Error Correcting Codes in Computer Arithmetic," Chapter 5 in Advances in Information Systems Science, (J. L. Tow, Editor), Vol. 4, Plenum Press, New York, pp. 273-326.

Monteiro, P., and Rao, T. R. N. (1972), "Multiresidue Codes for Double Error Correction," IEEE-TCCA Symposium on Computer Arithmetic, May.

Neto, A. (1968), "Poema," in Очиј Без Сузе , Kultura, Beograd, pp. 84-93 (in Portuguese and Russian).

Peterson, W. W., and Weldon, E. J., Jr. (1972), "Error-Correcting Codes," Second Edition, The MIT Press, Cambridge (Mass.).

Rao, T. R. N. (1970), "Biresidue Error Correcting Codes for Computer Arithmetic," IEEE Transactions on Computers, Vol. C-19, May, pp. 398-402.

Rao, T. R. N., and Garcia, O. N. (1971), "Cyclic and Multiresidue Codes for Arithmetic Operations," IEEE Transactions on Information Theory, Vol. IT-17, No. 1, pp. 85-91, January.

Reitweisner, G. H. (1960), "Binary Arithmetic," in Advances in Computers, Vol 1, (F. L. Alt, Editor), Academic Press, New York, pp. 232-308.

Sellers, F. F. Jr., Hsiao, M.-Y., and Bearnson, L. W. (1968), "Error Detecting Logic for Digital Computers," McGraw-Hill Book Co., New York.

Sitnichenko, S. I. (1970), "Decoding Arithmetic Error Correcting Codes," Engineering Cybernetics, No. 4, pp. 731-736.

Tsao-Wu, N. T. (1968), "Arithmetic Cyclic Codes," Department of Electrical Engineering, Northeastern University, Boston, Mass., Part 1 of Communication Theory Group Report No. 10, Final Report, AFCRL-68-0512 (June).

Tsao-Wu, N. T., and Chang, S-H. (1969), "On the Evaluation of Minimum Distance of Binary Arithmetic Cyclic Codes," IEEE Transactions on Information Theory, Vol. IT-15, No. 5, pp. 628-631, September.

Figure Title

1. Scheme for Obtaining the Product $A \cdot N$ where $A = \prod_{j=1}^{r} (2^{m_j} - 1)$; p_i are Clock Pulses at Instants $i = 0, 1, \ldots, r+1$; FF and D-FF Denote RS-Flip-Flops and D-Flip-Flops.

Table 1

Information rates for some AN codes with $A = \prod_{j=1}^{r} (2^{m_j}-1)$ and $d > 2^s$.

$A = \prod_{j=1}^{r} (2^{m_j}-1)$	s	length	rate
$(2^5-1)(2^6-1)(2^7-1)$	2	37	0.515
$(2^5-1)(2^6-1)(2^7-1)(2^{41}-1)$	3	78	0.244
$(2^7-1)(2^8-1)(2^9-1)(2^{67}-1)(2^{137}-1)$	4	269	0.152
$(2^7-1)(2^8-1)(2^9-1)(2^{11}-1)(2^{13}-1)(2^{653}-1)$	3	1300	0.46
$(2^7-1)(2^8-1)(2^9-1)(2^{11}-1)(2^{151}-1)(2^{305}-1)$	4	605	0.188

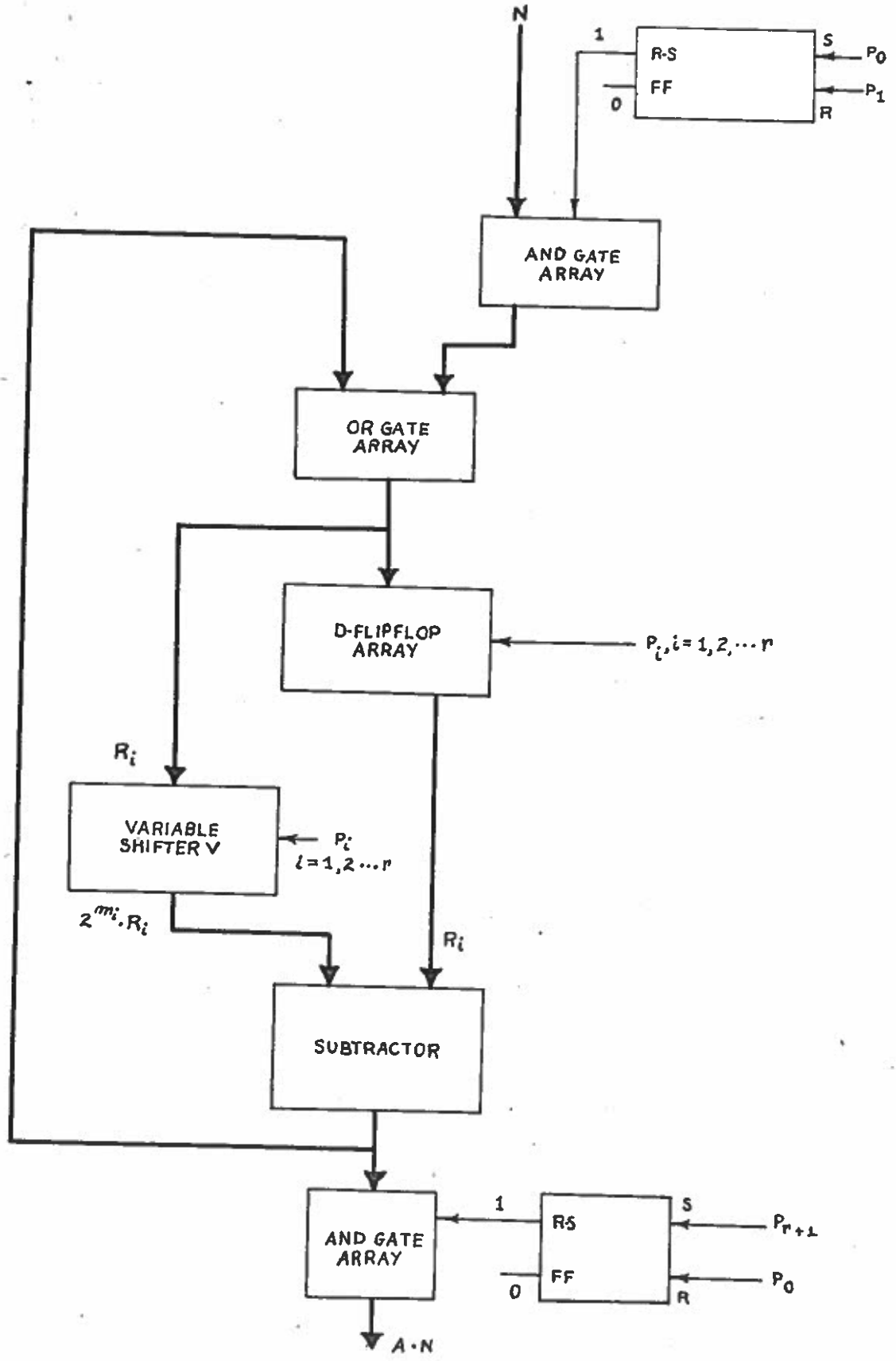
N
1
R-S
FF
0
S
R
P_0
P_1
AND GATE ARRAY
OR GATE ARRAY
D-FLIPFLOP ARRAY
$P_i, i = 1, 2, \cdots r$
R_i
VARIABLE SHIFTER V
P_i
$i = 1, 2 \cdots r$
$2^{m_i} \cdot R_i$
R_i
SUBTRACTOR
AND GATE ARRAY
1
RS
FF
0
S
R
P_{r+1}
P_0
A·N

Figure 1