SEL-63-118

The Synthesis of Nonlinear Feedback Shift Registers

by
K. B. Magleby

October 1963

Technical Report No. 6207-1

Prepared under
Office of Naval Research Contract
Nonr-225(24), NR 373 360
Jointly supported by the U.S. Army Signal Corps, the
U.S. Air Force, and the U.S. Navy
(Office of Naval Research)

SYSTEMS THEORY LABORATORY

STANFORD ELECTRONICS LABORATORIES

STANFORD UNIVERSITY · STANFORD, CALIFORNIA

SEL-63-118

THE SYNTHESIS OF NONLINEAR FEEDBACK SHIFT REGISTERS

by

K. B. Magleby

October 1963

Technical Report No. 6207-1

Prepared under
Office of Naval Research Contract
Nonr-225(24), NR 373 360
Jointly supported by the U.S. Army Signal Corps,
the U.S. Air Force, and the U.S. Navy
(Office of Naval Research)

Systems Theory Laboratory
Stanford Electronics Laboratories
Stanford University Stanford, California

ABSTRACT

The nonlinear feedback shift register is a very useful digital sequence generator. Even though the design of this network has been unnecessarily difficult and unsystematic, the network is often used for code generation, sequence generation, and counting. The development of an efficient synthesis procedure in this report allows the potential of this important class of digital networks to be more fully utilized.

Two domains that describe the behavior of a feedback shift register have been developed. These are the sequence and polynomial domains and they are analogous to the frequency and time domains in the description of continuous systems. The domains are related by an expansion of orthogonal functions.

The synthesis procedure developed in the polynomial domain consists of four steps: (1) Constructing a finite field with the necessary properties; (2) finding the polynomials that correspond to the desired output sequences; (3) obtaining the polynomial that describes the shift register as a product of the polynomials that represent the desired output sequence; and (4) obtaining the feedback network from the polynomial that describes the shift register. In the procedure, the output sequences are mapped to the roots of irreducible polynomials, thereby providing an algebraic description of the register's behavior.

To synthesize the shift register in the sequence domain, several properties of the output sequences are needed. The class of sequences and state graphs corresponding to shift-register behavior is established. The cycles and output sequences of a simple, circulating shift register are used to synthesize an arbitrary feedback shift register. The procedure has two steps. First, the specification is expressed in terms of the joining and removing of cycles of a circulating register. Second, the feedback network and output sequences are found from a knowledge of these operations.

CONTENTS

ILLUSTRATIONS

ACKNOWLEDGMENT

The author wishes to thank Professor Donald L. Epley for his assistance and direction of the work described in this report. Valuable discussions with Drs. B. Elspas, W. Kautz, and R. Edwards are greatly appreciated. The author also wishes to thank Professor R. L. Mattson for his assistance in the organization of this report.

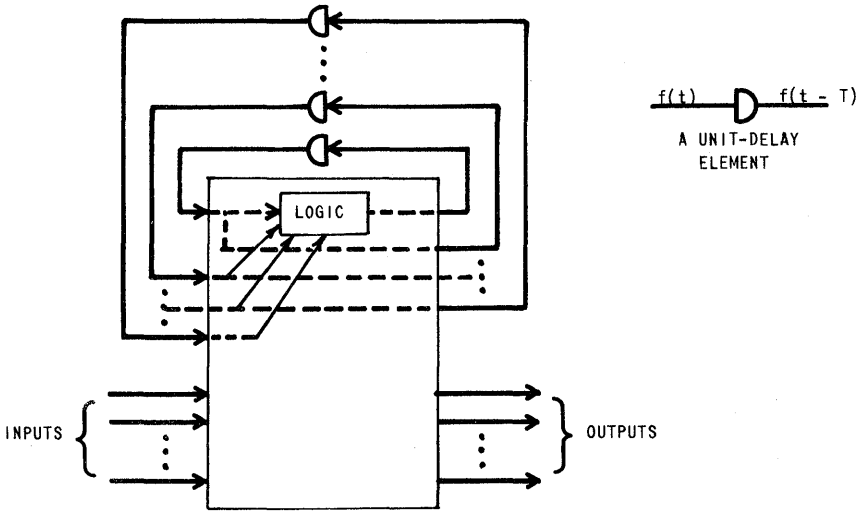
I. INTRODUCTION

A. MOTIVATION

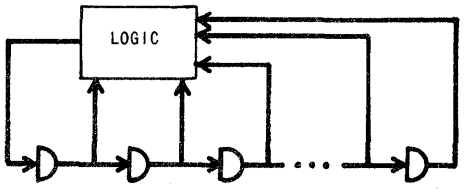
The feedback shift register is a type of digital oscillator. In recent years it has found wide usage in sequence or code generation, counting, and sequence recognition or decoding [Refs. 1, 2, 3]. Even though it is the simplest type of digital sequential network, no general systematic design procedure exists for the nonlinear register. A great deal has been done, however, to give aids to design [Ref. 4], and design procedures have been given for the linear register [Ref. 5]. The properties of the sequences produced by the nonlinear shift register have been studied, and methods of modifying a linear register to produce several interesting nonlinear shift-register sequences have been developed [Ref. 6].

The topology of the general sequential network is shown in Fig. 1a. The unit delays can be physical delay elements or clocked storage elements. When the system is clocked, a signal can propagate through the delays only at the clocking intervals. The network delays are assumed to be of sufficient duration to prevent loop propagation during the clock pulse. In unclocked networks the operation may be a function of the order of switching of the various elements. When this possibility exists, the network is said to have a race condition. The networks studied here are clocked, so race conditions do not exist.

The shift register is obtained from the general sequential network by making the logic for n - 1 of the state variable a straight-through connection, as indicated in Fig. 1a. The network is redrawn in Fig. 1b in the form of a feedback shift register. The networks considered here will have no inputs or outputs. It is assumed that the state variables themselves are the outputs or that the output is obtained from the state variables by combinatorial logic. For a large number of applications the loss of inputs is a serious one. However, the author hopes that the theory presented here can be extended to include the case with inputs in a manner similar to that done by Srinivasan for the linear networks [Ref. 7].



a. A general sequential network. (Dotted lines indicate shift-register connections)



b. A feedback shift register.

FIG. 1. SOME SEQUENTIAL NETWORKS.

B. SOME LINEAR RELATIONSHIPS

The use of the term "nonlinear" in digital networks is a little different than in analog networks. A linear digital network is one which consists entirely of modular adders, the modulus being 2 for the binary case. If the logic in Fig. 1a consists entirely of modulo 2 adders, the outputs can be written as a linear function of the inputs.

$$
\begin{aligned}
x_1' &= a_{11}x_{12} + a_{12}x_2 + \cdots a_{1n}x_n \\
x_2' &= a_{21}x_1 + a_{22}x_2 + \cdots a_{2n}x_n \\
&\vdots \\
x_n' &= a_{n1}x_1 + a_{n2}x_2 + \cdots a_{nn}x_n
\end{aligned}
$$

The operation + is defined as follows:

+	0	1
0	0	1
1	1	0

or, in matrix notation, $X' = TX$, where

$$
T = \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & & & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nn}
\end{bmatrix}
$$

The characteristic polynomial of the T matrix is defined as the determinant of the matrix obtained by subtracting x from the diagonal elements of T.

$$\phi(x) = |T - xI|$$

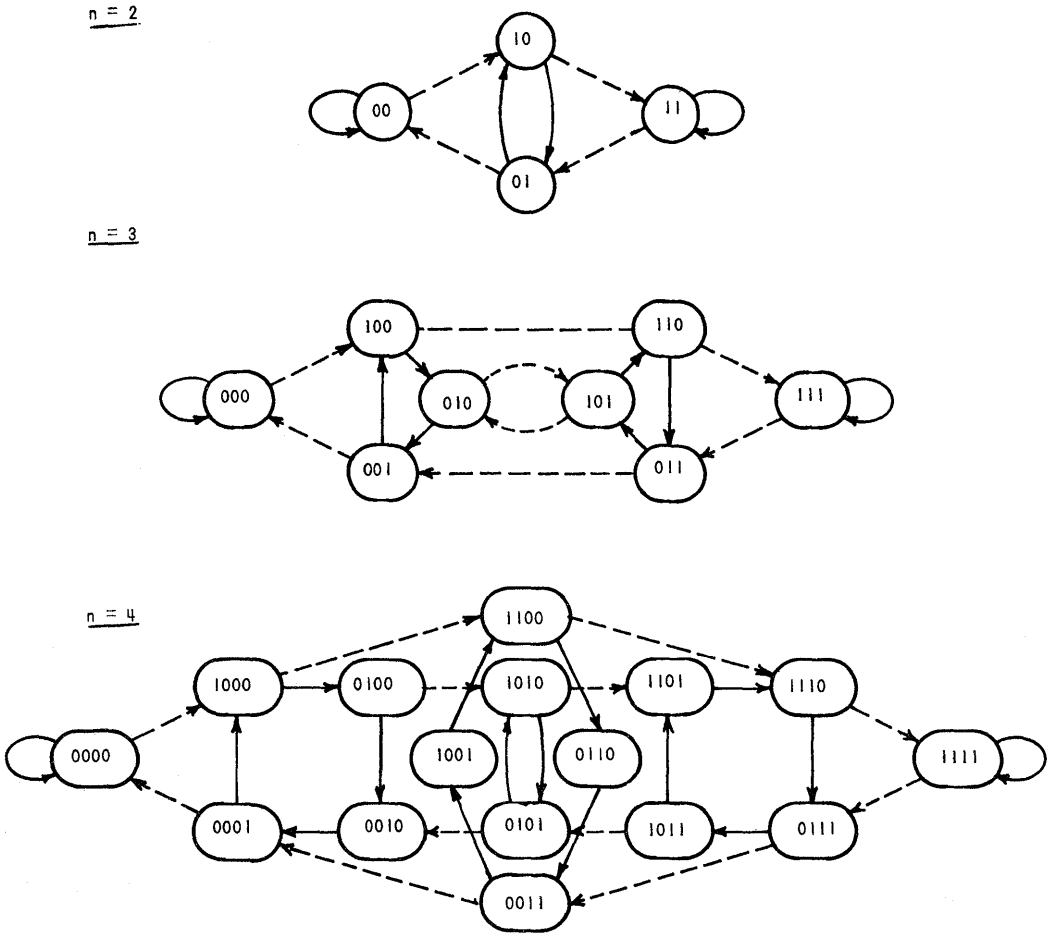
Since the networks being considered have no inputs, a description of the cyclic behavior is sufficient to characterize the network. Zierler [Ref. 8] and Elspas [Ref. 5] have developed a method to determine the cycle lengths of the register in terms of the properties of its characteristic polynomial. Elspas also develops a synthesis procedure in terms of the characteristic polynomial. Thus, for the linear register, both the analysis and synthesis depend on properties of the characteristic polynomial. The polynomial also provides an algebraic description of the register. In Chapter II, the author develops an algebraic description and synthesis in terms of a different polynomial.

C. NONLINEAR RESULTS

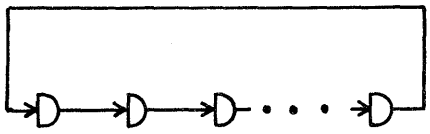
When the logic consists of arbitrary elements, the linear theory fails since the equations describing the next state in terms of the present state are not linear. Attempts have been made to extend the linear theory to include the nonlinear behavior by increasing the size of the matrix [Ref. 9] or increasing the degree of the characteristic polynomial [Refs. 10, 11]. The polynomial approach will be described in more detail in Chapter II.

A great deal of work has been done to provide tools which are useful in the design of the nonlinear register. Several relations between the state-graph structure and the feedback logic have been given [Ref. 11]. An exhaustive analysis has been carried out for short registers and a generalized state graph drawn for each length [Ref. 12]. For a small number of variables, the cycles can be found by inspecting this graph, called Good's diagram [Ref. 13]. Good's diagrams for $n = 2$, 3, and 4 are given in Fig. 2. The solid lines in these diagrams indicate the cycles of a circulating shift register. A circulating register is a feedback shift register whose feedback network consists of a connection from the output of the last delay element to the input of the first. The states on a cycle of a circulating register are obtained from a representative state by cyclically permuting its digits.

The main difference between the research presented in this report and the work described in the above references is that the nonlinear



a. Good's diagrams for n = 2, 3, 4.



b. Circulating shift register.

FIG. 2. AN EXHAUSTIVE ANALYSIS OF SOME SHIFT REGISTERS.

registers are studied directly rather than modifying linear registers to make them nonlinear. This approach is very satisfying since the linear behavior is realized as a special case of the nonlinear behavior. Even though it is more general, the approach presented here is much simpler than the linear theory in many cases.

D. ORGANIZATION

The nonlinear theory is presented in two different domains and a correspondence between the two domains is given. These are called the sequence domain and the polynomial domain and are analogous to the time and frequency domains in continuous systems.

In Chapter II the correspondence between the two domains is described and the polynomial-domain approach is presented. In Chapter III the sequence domain is introduced and several properties of shift-register sequences are developed. Chapter IV attacks the problem of decomposing the prescribed cycle set in terms of operations on the cycles of a circulating register. This chapter also shows the correspondence between the operations on circulating-register cycles and the feedback-network specification, as well as a design example. Chapter V gives some concluding remarks and recommendations for future study.

Chapters II - IV provide a general solution to the characterization and synthesis of the nonlinear shift register. An algebraic description of the register and its output sequences is given in the polynomial domain. A synthesis procedure is given in both domains, and a connection between the two domains is developed. The polynomial gives a concise method of describing the network behavior and is conceptually simpler. As a practical design tool, the sequence-domain synthesis procedure is simpler and does not depend upon the existence of tables of polynomials. While the procedure itself is simple, the development of the procedure is not.

II. NONLINEAR SHIFT-REGISTER SYNTHESIS BASED ON IRREDUCIBLE POLYNOMIALS

A. BACKGROUND

There is an interesting analogy between digital systems and continuous systems. The autonomous feedback shift register in digital systems corresponds to an oscillator in continuous systems. In the digital oscillator several different cycles may be generated depending on its starting state; in the analog oscillator several different frequencies may be generated depending on the starting conditions.

In describing the behavior of both classes of systems, there are two corresponding domains that can be used. For analog oscillators, one may use either the frequency or time domain. Some operations are more convenient in the frequency domain, while others are easier in the time domain. In digital systems, there are also two domains for describing the behavior. These are the sequence domain and the polynomial domain. The analogy between the sequence domain in digital systems and the time domain in analog systems is a very natural one, since both describe the output as a function of time. Perhaps the correspondence between the polynomial domain and the frequency domain may seem a little strange, but it will be shown to be very natural also. As will be seen, there is a similar connection between the two domains for both the digital and analog oscillators in the form of an expansion of orthogonal functions.

Kautz, Elspas, and Stone [Refs. 10, 11] have observed some interesting properties of what they call the "associated polynomial." This is the polynomial obtained from the normal characteristic polynomial of the linear network by replacing x^i by x^{2^i}.

If $\phi(X)$ = characteristic polynomial of the network described in Chapter I

$$= \sum_{i=0}^{n} a_i X^i$$

then $A(X)$ = associated polynomial

$$= \sum_{i=0}^{n} a_i X^{2^i} \quad \text{where the } a_i \text{ are the same as above.}$$

The associated polynomial can be modified to describe the behavior of a nonlinear register since some terms of the general polynomial of degree 2^n are missing in the description of a linear register. This will be shown in the following sections. Those terms not in a linear register $\left(X^{2^i}\right)$ will be called nonlinear terms, and it is to be expected that they will play a key role in describing the behavior of a nonlinear shift register. Since the subject of this chapter is the design of nonlinear registers, the polynomial for an n-stage register will have degree 2^n and will be expected to degenerate to the associated polynomial when the register is linear.

B. PROPERTIES OF ASSOCIATED POLYNOMIALS

The properties of the associated polynomial established from linear theory in Refs. 10 and 11 are summarized below;

1. For an n-stage register, $A(X)$ has degree 2^n.
2. All terms are of the form X^i, $i \leq n$.
3. The term $a_o = 0$ or 1, depending on whether the register is singular or nonsingular.
4. The degree of each of the irreducible factors of $A(X)$ is equal to a cycle length in the cycle set. There is a one-to-one correspondence between cycles and irreducible factors of $A(X)$.
5. There exists a linear mapping of the roots of $A(X)$ to the states of the register. Each state is a binary n-tuple which is an element of an n-dimensional vector space. Successive states on a cycle are mapped to a root and its square.

Each digit of the binary sequence representing the states is either zero or one. These scalars, together with the following operations, define a two-element field, F.

$F = \{0,1\}$

+	0	1
0	0	1
1	1	0

·	0	1
0	0	0
1	0	1

A linear vector space over the field F can be obtained by considering the space of all binary n-tuples $X = x_1 x_2 \ldots x_n$ of elements x_i in F. The vector addition and scalar multiplication are defined as follows:

$$F^n = \{X = x_1 \ldots x_n | x_i \text{ in } F\} \quad \text{for } X, Y \text{ in } F^n$$

$$X = x_1, x_2, \ldots, x_n$$

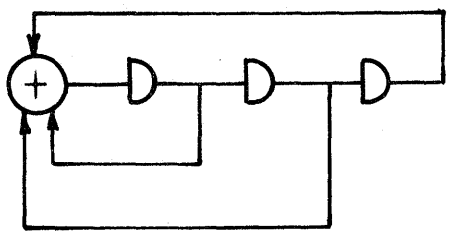
$$Y = y_1, y_2, \ldots, y_n$$

$$X + Y = x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n$$

For c in F,

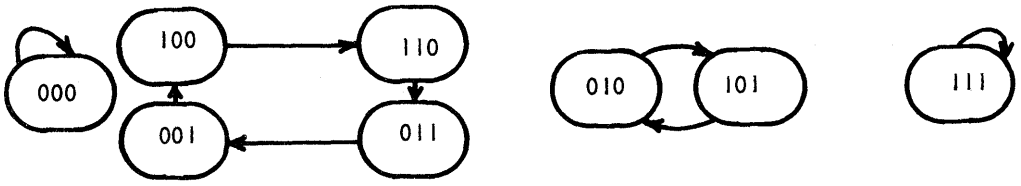
$$cX = cx_1, cx_2, \ldots cx_n$$

The unit vector is the all-one vector, 111 ... 1. As an example of applying the associated-polynomial theory to the linear register, consider the following network:



$$\emptyset = x^3 + x^2 + x + 1$$
$$A(x) = x^8 + x^4 + x^2 + x$$
$$= x(x + 1)(x^2 + x + 1)(x^4 + x + 1)$$

The characteristic polynomial $\phi(x)$ is found using the method described in Chapter I, and from it the associated polynomial $A(x)$ is found. The state graph, obtained by analysis of the register, is seen to be:



Using the relation that successive states on a cycle are mapped to a root and its square if $A = 100$, then $A^2 = 110$, $A^4 = 011$, $A^8 = 001$; if $B = 010$, $B^2 = 101$.

In the equation $x^4 + x + 1$, let $x \Rightarrow A = 100$:

$$A^4 + A + 111 = 011 + 100 + 111 = 0$$

Also in $x^2 + x + 1$, let $x \Rightarrow B = 010$:

$$B^2 + B + 111 = 101 + 010 + 111 = 0$$

The states of the 4 cycle are seen to be roots of $x^4 + x + 1$, those of the 2 cycle are roots of $x^2 + x + 1$, while the two states on the 1 cycles are roots of x and $x + 1$ respectively.

The above example is unusual in that all the factors of the associated polynomial contain only linear terms. This fact allows the determination of the roots without defining a multiplication operation since only the squaring and addition operations are used.

There are three irreducible polynomials of degree four. The correspondence between one polynomial and one output sequence is established. The remaining two irreducible polynomials of degree four and output sequences with period four are:

$$\left.\begin{array}{l} x^4 + x^3 + 1 \\ x^4 + x^3 + x^2 + x + 1 \end{array}\right\} \quad \left\{\begin{array}{l} 00010001 \ldots \\ 11101110 \ldots \end{array}\right.$$

From linear considerations, there is no way to decide which sequence should correspond to each polynomial. Since one sequence never occurs without the other in a linear network, this presents no difficulty for the linear theory; however, this ambiguity must be resolved to extend the theory to include the nonlinear register. The ambiguity is recognized in Refs. 10 and 11, but no solution is given. The solution obtained by the author is given in the following section.

C. PROPERTIES OF POLYNOMIALS REQUIRED TO DESCRIBE NONLINEAR BEHAVIOR

To extend the theory to include the nonlinear behavior it is first necessary to define a multiplication operation to produce the nonlinear terms of the polynomial. Having done this, the ambiguity mentioned above regarding the sequence-polynomial correspondence will be resolved. The network required to realize the multiplication operation will then be determined. Finally, a test of realizability will be developed to determine which polynomials can be realized with a register of a given length.

The multiplication operation must be defined such that the properties of the polynomial will be similar to those for the associated polynomial for the linear register. The polynomial which describes the feedback rule for the nonlinear register will be called the describing polynomial, denoted by $D(x)$, and will have the following properties:

1. For an n-stage register, $D(x)$ has degree 2^n.
2. The degree of the irreducible factors of $D(x)$ is equal to the cycle lengths in the cycle set. There is a one-to-one correspondence between cycles of the register and irreducible factors of $D(x)$.
3. There is a mapping of the roots of $D(x)$ to the output sequences of the register. The output sequence for each cycle is mapped to a root of the factor of $D(x)$ corresponding to that cycle. The various roots of the irreducible factor of $D(x)$ are the various phases of the output sequence. The sequences are elements of a k-dimensional vector space (F^k), where k is the least common multiple of the cycle lengths. The operations of scalar addition, scalar multiplication, vector addition, and vector multiplication by a scalar are the same as for F^k described above.

These properties are very similar to the corresponding properties of the associated polynomial. The primary difference is in the correspondence between the roots of the polynomial and the behavior of the register. The roots of the associated polynomial are mapped to the states of the register, while the roots of the describing polynomial are mapped to the output sequences of the register.

The sequence that represents each cycle of a shift register can be represented as a binary L-tuple, where L is the length of the cycle: $X = x_1 x_2 \ldots x_L$; x_L is the output after the first unit in time; x_{L-1} is the output after the second; etc. The states of the register which are placed on a particular cycle are obtained by considering the first n digits of the above sequence, where n is the number of delay elements in the register. As the digits of the output sequence are cyclically permuted, the first n digits of the resulting sequences give the binary representation of the states on that cycle. For example in a three-stage register, states 000, 100, 010, 001 are on the cycle represented by the sequence 0001.

Define δ as a mapping from F^n onto F^n such that for any X in F^n:

$$X = x_1 x_2 \ldots x_n$$

$$\delta X = x_n x_1 x_2 \ldots x_{n-1} = X^2$$

$$\delta(\delta X) = \delta^2 X = x_{n-1} x_n x_1 x_2 \ldots x_{n-2} = X^4$$

Let ω be the vector in F^k defined as follows:

$$\omega = \underbrace{100 \ldots 0}_{k \text{ digits}}$$

then

$$\delta\omega = 0100 \ldots 0 = \omega^2$$

$$\delta^2\omega = 0010 \ldots 0 = \omega^4$$

$$\delta^{k-1}\omega = 00 \ldots 01 = \omega^{2^{k-1}}$$

$$\delta^k\omega = 100 \ldots 0 = \omega$$

The n vectors $\delta^i\omega$, where $i = 0, 1, 2, \ldots, k-1$, form a basis for the vector space F^k defined above. Any vector (X) in F^k can be written as:

$$X = \sum_{i=0}^{k-1} a_i \delta^i \omega$$

The application of the mapping δ to a sequence is equivalent to observing the output sequence of the register one clock time later. Thus if the content of the register is state $S_i = s_1 s_2 \ldots s_n$, the i^{th} state on a cycle generated by the sequence $X = x_1 x_2 \ldots x_n \ldots x_k$, then the first n digits of $\delta X = x_k x_1 x_2 \ldots x_{n-1} x_n \ldots x_{k-1}$ will be the $(i+1)^{th}$ state on the cycle, $S_{i+1} = x_k s_1 s_2 \ldots s_{n-1}$.

Another interpretation of the mapping δ is the relationship between the output and input of a delay element. If $X = x_1 x_2 \ldots x_k$ is the sequence at the output of a delay element, then $\delta X = x_k x_1 x_2 \ldots x_{k-1}$ is the sequence at the input of that delay element.

Edwards uses a finite field with a normal basis in the algebraic synthesis of some switching circuits [Ref. 14]. This method of field construction is also useful in the design of shift registers. The finite field is constructed with elements of F^n. The operation of vector addition is the addition operation of the field. The multiplication operation is obtained by considering the basis elements of F^k, $(\delta^i\omega)$, as roots of an irreducible polynomial of degree k. These roots are linearly independent, so the polynomial chosen to correspond with them

must have linearly independent roots. Every element in the field must be a linear combination of the basis elements and hence equivalent to the basis element raised to some power. Thus, the roots of the polynomial chosen are to be primitive elements in the field. A polynomial whose roots are primitive is called a primitive polynomial. A field formed by taking polynomials over a field F modulo an irreducible polynomial of degree n is called an extension field of degree n over F. The polynomial whose roots are the basis elements of the space F^n is then primitive, with linearly independent roots. The method of constructing the field is most easily shown by giving several examples which will be useful when the synthesis procedure is given.

Example 1: Construct the Galois field of 2^2 elements, $GF(2^2)$, formed as a field of polynomials over F modulo $x^2 + x + 1$. Let ω be a root of $x^2 + x + 1$.

$$\delta^0\omega = \omega = 10$$
$$\delta\omega = \omega^2 = 01$$
$$\omega^2 + \omega + 1 = 0$$
$$\omega^3 = \omega^2 + \omega = 11$$

then

X = 00 is a root of x

X = 01, 10 are roots of $x^2 + x + 1$

X = 11 is a root of $x + 1$

Example 2: Construct the Galois field of 2^3 elements formed as a field of polynomials over F modulo $x^3 + x^2 + 1$. Let ω be a root of $x^3 + x^2 + 1$.

$\omega = 100$

$\omega^2 = 010$

$\omega^3 + \omega^2 + 1 = 0$

$\omega^3 = \omega^4 + \omega$

$\omega^3 = 101$ $\qquad$ $\omega^5 = \omega^4 + \omega^2$

$\omega^4 = 001$ $\qquad$ $\omega^6 = \omega^2 + \omega$

$\omega^5 = 011$ $\qquad$ $\omega^7 = \omega^4 + \omega^2 + \omega + 1$

$\omega^6 = 110$ $\qquad$ $\omega^8 = \omega$

$\omega^7 = 111$

With this algebra, the binary sequence corresponding to roots of the third-degree irreducible polynomials are:

$x^3 + x^2 + 1$	001, 010, 100
$x^3 + x + 1$	110, 011, 101

Example 3: Construct $GF(2^6)$ where $\omega = 100000$ is a root of $x^6 + x^5 + 1$. A few of the elements of $GF(2^6)$ are given below. (These will be used in a later example.) Rather than write the symbol ω in the equations, for convenience only the exponents will be used. For example: $\omega^5 = \omega^4 + \omega^{16} + \omega^{32}$ will be written $5 = 4 + 16 + 32$.

$5 = 4 + 16 + 32$	001011
$9 = 1 + 4 + 8 + 32$	101101
$10 = 1 + 8 + 32$	100101
$18 = 1 + 2 + 8 + 16$	110110
$20 = 1 + 2 + 16$	110010
$21 = 2 + 8 + 32$	010101
$25 = 1 + 2 + 8 + 32$	110101
$29 = 1 + 2 + 4 + 8 + 16$	111110
$30 = 1 + 2 + 16 + 32$	110011
$31 = 8 + 16$	000110
$33 = 2 + 16 + 32$	010011
$45 = 1 + 8$	100100
$48 = 1 + 8 + 16$	100110
$54 = 4 + 32$	001001

$57 = 1 + 2 + 4 + 8$	111100
$60 = 1 + 2 + 4 + 32$	111001
$62 = 16 + 32$	000011

With this algebra, the roots of some of the sixth-degree irreducible polynomials are

$x^6 + x + 1$	$\omega^{31} = 000110$
$x^6 + x^5 + x^4 + x^2 + 1$	$\omega^{5} = 001011$
$x^6 + x^5 + x^4 + x^2 + 1$	$\omega^{24} = 001101$

Note:

$\omega^{45} = 100100$ is a root of $x^3 + x^2 + 1$

$\omega^{18} = 110110$ is a root of $x^3 + x + 1$

$\omega^{21} = 010101$ is a root of $x^2 + x + 1$

which agrees with the above.

The mapping between the elements of the finite field (ω^i) and the output sequence can be interpreted as an expansion of the sequence in terms of orthogonal functions. The method of obtaining this expansion is identical to that normally used in expanding a continuous function in terms of orthogonal functions. This is outlined below.

An inner or dot product of two vectors, X and Y, denoted $\Pi(X)(Y)$, is:

$$X = x_1 x_2 \cdots x_n$$

$$Y = y_1 y_2 \cdots y_n$$

$$\Pi(X)(Y) = x_1 y_1 + x_2 y_2 \cdots + x_n \cdot y_n$$

The basis vectors are orthogonal with respect to Π:

$$\Pi(\delta^i\omega)(\delta^j\omega) = 1 \quad \text{if } i = j$$
$$= 0 \quad \text{otherwise}$$

Also, the following properties of the inner product will be used:

$$\Pi\ (X)\ (cY) = c\Pi\ (X)\ (Y)$$

$$\Pi\ (X)\ (Y + Z) = \Pi\ (X)\ (Y) + \Pi\ (X)\ (Z)$$

Any sequence of length p, where p divides n, can be written as a linear sum of the basis elements. The property of orthogonality with respect to Π is used to obtain the expansion of the sequence in terms of the basis elements.

Let X be any vector in F^n. Then

$$X = \sum_{i=0}^{2^{n-1}} a_i(\delta^i\omega)$$

$$\Pi(\delta^j\omega)(X) = \sum_{i=0}^{2^{n-1}} a_i\Pi(\delta^j\omega)(\delta^i\omega)$$

$$= 0 \text{ when } i \neq j$$

$$= a_i \text{ when } i = j$$

$$a_i = \Pi(\delta^i\omega)(X)$$

This method of constructing the finite field, together with the following theorem from finite-field theory, allows the construction of the mapping of the roots of the polynomials to the sequences with the desired properties.

Theorem: If A is a root of an irreducible polynomial $f(x)$ of degree k in an extension field over F_2, then the other roots are A^{2^2}, A^{2^3}, ..., A^{2^k}.

The proof of this theorem is given in Ref. 3, Theorem 6.26. The following corollaries are a direct consequence of this theorem:

Corollary 1: If A and B are roots of the same irreducible polynomial, then A and B are mapped to sequences S_a and S_b which generate the same cycle.

Proof: Since A and B are roots of the same irreducible polynomial, $A = B^{2^i}$ for some i, then by definition of δ, $S_a = \delta^i S_b$ and S_a is the same output sequence as S_b shifted i units in time. Thus S_a and S_b generate the same cycle.

Corollary 2: If k is the length of the cycle which the sequence S generates, then S is mapped to the root (A) of an irreducible polynomial of degree k.

Proof: The sequences $S, \delta S, \delta^2 S \ldots \delta^K S$ are mapped to distinct elements $(A, A^2, \ldots, A^{2^k})$ which are roots of some polynomial of the field; $\delta^{K+1} S$ is not a new sequence and hence must be mapped to one of the previous roots,

$$\delta^{K+1} S \Rightarrow A^{2^i} \quad \text{for } i < K$$

If $i \neq 0$ we have $A^{2^{K+1-i}} = A$, which cannot be true since all the roots are distinct. Hence $i = 0$ and the polynomial has degree K.

Corollary 3: The product of the irreducible polynomial whose roots describe the cycles of a nonsingular register of length n is of degree 2^n. This polynomial is the describing polynomial for the register $(D(x))$.

Proof: There are 2^n states in the register, all of which are the first n digits of a sequence which is mapped to some root of a factor of $D(x)$. Thus, the sum of the degrees of the factors of $D(x)$ is 2^n so the degree of $D(x)$ is 2^n.

These corollaries establish the three properties of $D(x)$ required for the synthesis procedure described below.

D. THE SYNTHESIS PROCEDURE

The design of a nonlinear shift register based on irreducible polynomials is summarized in the following steps:

1. Select an irreducible polynomial of degree such that it is divisible by all the desired cycle lengths and which is primitive with linearly independent roots. (If the specification is in terms of polynomials, choose the degree such that it is divisible by all the degrees of the specified polynomials.) See Ref. 3 for a table of irreducible polynomials.
2. Construct a finite field based on this irreducible polynomial. This can be done once for each degree and tabulated for future reference.
3. Choose an irreducible polynomial for each desired cycle with degree equal to the cycle length. The polynomial chosen will determine the sequence which is produced, and in some cases the number of delay elements required in the register. (This step is omitted if the specification is in terms of polynomials.)
4. Multiply the polynomials together to obtain the polynomial to associate with the register.
5. Determine the logic required for the nonlinear terms in the polynomial.
6. Obtain the feedback network by connecting all the terms of the polynomial as inputs to a modulo 2 adder.

E. GENERATION OF NONLINEAR TERMS

The terms that correspond to X^i, where $i \neq 2^j$ for j an integer, are called the nonlinear terms of the polynomial since they are not present in the polynomials that describe a linear register.

A multiplication is defined by the finite field and this is used to obtain the nonlinear terms of the polynomial. This multiplication cannot be performed on the serial sequences one bit at a time as the operation of squaring can, and hence in general some storage is required. In some cases, the proper sequence can be obtained without storage, as will be seen in the following section. These are the sequences which can be generated by a register of minimum length. A diagram of a multiplier is shown in Fig. 3.

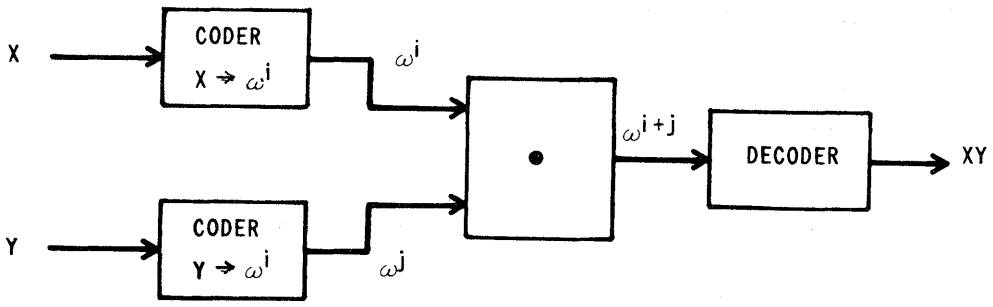


FIG. 3. A MULTIPLIER.

The following example will illustrate the procedure:

Example: The desired cycle set is $(1_2, 6)$.

Since all cycle lengths divide 6, a sixth-degree polynomial will be used to define the multiplicative group in the Galois field. Choose $\omega = 100000$ as a root of $x^6 + x^5 + 1$, which is primitive and has linearly independent roots. Example 3 in Sec. C above gives the algebra based on this polynomial.

The polynomials corresponding to the two 1 cycles are x and $x + 1$. We see that there are several sixth-degree polynomials that are irreducible. Suppose we choose $x^6 + x^5 + x^4 + x + 1$ to obtain the 6 cycle. The polynomial of the register $(\emptyset)$ is:

$$\emptyset = (x)\,(x + 1)\,(x^6 + x^5 + x^4 + x + 1)$$

$$= x^8 + x^5 + x^3 + x$$

We see that $b = \omega^4 + \omega^{16} + \omega^{32}$ is a root of $x^6 + x^5 + x^4 + x + 1$. From example 3,

$$b = \omega^5$$

$$b^3 = \omega^{15} = \omega + \omega^8 + \omega^{16} + \omega^{32}$$

$$b^5 = \omega^{25} = \omega + \omega^2 + \omega^8 + \omega^{32}$$

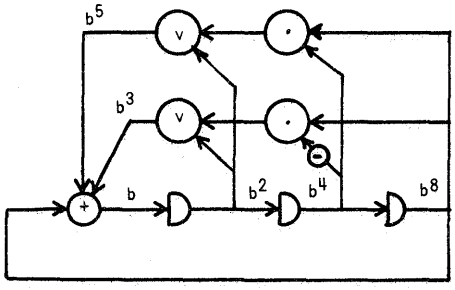
The logic to obtain b^3 and b^5 is given in Fig. 4.

y_1	y_2	y_3	b^3	b^5
1	1	0	1	1
0	1	1	0	1
0	0	1	0	0
1	0	0	1	1
0	1	0	1	0
1	0	1	1	1
0	0	0	0	0
1	1	1	1	1

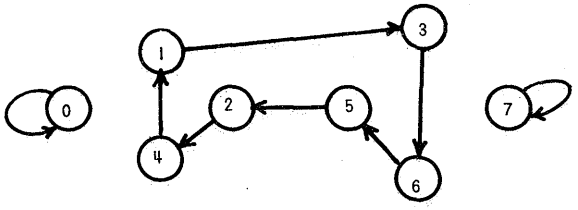
$b^3 = y_1 \vee y_2\bar{y}_3$

$b^5 = y_1 \vee y_2y_3$

a. Truth Table



b. Register



c. State Graph

FIG. 4. A THREE-STAGE NONLINEAR SHIFT REGISTER.

F. A TEST FOR REALIZABILITY

It was pointed out earlier that not every irreducible polynomial of degree k will be realizable by an n-stage register ($k \leq 2^n$) without requiring storage in the feedback network. A simple test to determine which polynomials are realizable can be given in terms of the roots of the polynomial. The particular polynomial chosen to construct the multiplicative group of the finite field will determine the roots of the remaining polynomials, so the realizability of a given polynomial depends upon the algebra chosen.

Theorem 1: Let S_a be the binary sequence that represents a root of $P(x)$ an irreducible polynomial of degree k. Then S_a is realizable as an output sequence of an n-stage register if and only if the sequence S'_a obtained from S_a defined below contains each n-digit word only once.

$$S_a = x_1 x_2 \cdots x_k$$

$$S'_a = x_1 x_2 \cdots x_k x_1 x_2 \cdots x_{n-1}$$

Proof: Each n-digit word represents a state of the shift register that is on the cycle which produces the output sequence S_a. Suppose a given n-digit word exists twice on the sequence. Then the cycle length is less than k, which contradicts corollary 2 above.

If each n-digit word in S'_a occurs only once, then there are k distinct states on the cycle. There exists a combinatorial network which can produce an arbitrary output for each distinct input; so a shift register exists which produces the sequence S_a as an output sequence.

Corollary: If the sequence S'_a contains each n-digit work only once, then the vectors A^i for i any integer less than 2^n can be formed as a function of the n digits in a network without storage.

Proof: Since each n-digit word occurs only once as the cycle of length k is traversed, there exists a combinatorial network which can produce any k-digit sequence and hence the sequence representing A^i.

G. ALTERNATE NETWORK REALIZATION METHODS

Although the above procedure gives a method of realizing the prescribed behavior, the realization of the feedback is unnecessarily complex. A more straightforward procedure is given in Chapter IV in terms of the sequences which are the roots of the polynomials. The usefulness of the polynomial approach is in obtaining an algebraic description (the polynomial) for the shift register with nonlinear logic, and an algebraic description for the sequences which it produces. Each sequence which is produced by the register is a root of the polynomial describing the register. Each irreducible factor of the describing polynomial is associated with one and only one output sequence.

III. SOME PROPERTIES OF SHIFT-REGISTER SEQUENCES

A. SEQUENCE PROPERTIES

This chapter establishes some useful properties of shift-register output sequences and state graphs. The relationship between the state graph and the output sequence is pointed out. Chapter IV develops a design method based on these properties.

Several obvious properties of the state graph can be deduced from Good's diagrams given in Chapter I. See Fig. 2 for the following properties:

1. Each state has two possible predecessors and two possible successors.
2. There are only two 1 cycles. These give the output sequences 000...0 and 111...1.
3. There is only one 2 cycle which has the output sequence 0101...01.
4. The graph has a total of 2^n states.

Before proceeding to study shift-register sequences, several definitions will be given. Following the definitions one comprehensive example is given to illustrate their meaning.

1. A sequence (s) is said to be <u>recursive of degree n</u> if any n successive digits of the sequence occur at most once.*
2. A sequence ($S = s_1 s_2 \ldots s_L$) is <u>cyclic with length L and degree n</u> if the generator sequence S' of S of length $L + n - 1$ defined by $S' = s_1 s_2 \ldots s_L s_1 \ldots s_{n-1}$ is recursive of degree n.
3. A <u>state of degree n</u> on a sequence is any successive n digits on the sequence. For convenience, the octal equivalent to the binary number represented by the n digits with the <u>least significant binary digit on the left</u> will be used to refer to the state. When speaking of the sequences generated by n-element shift registers,

*The degree of the sequence is the number of delay elements required in a shift register to generate the sequence. This corresponds to the degree of the polynomial associated with the sequence. Some authors have used the term "order" to mean the number of delay elements in a sequential network.

the degree of recursiveness will be the same as the number of shift-register elements unless otherwise stated.

4. The cycle of degree n generated by a cyclic sequence S is the connected graph of states of degree n contained on the generator sequence of S. The states on the graph occur in the order that they appear on the generator sequence from right to left.
5. The span from state a to state b on sequence S is the number of digits on S from the beginning of a to the beginning of b.
6. State a is called adjacent to state b if a has a span of one to b on a sequence.
7. The (n - 1)-digit word that is common to two adjacent states is called an adjacency sequence. The adjacency sequence will be given in the octal representation of the sequence, with the least significant binary digit on the left.
8. The states spanned from a to b are the states contained on the sequence connecting a to b.
9. A nonsingular state graph contains cycles only. If the state graph has one or more states which are not on a cycle, it is called singular.
10. A cycle is said to have an appendage of length L if L states are contained on a single sequence of states leading into the cycle.
11. The measure of a state is equal to the number of ones in the binary representation of the state.

Example:

The sequence 00010111 is recursive of degree 3. The sequence is cyclic with length 8 and degree 3. The states of degree 4 contained on the sequence are 10, 4, 12, 15, and 16. The span from 16 to 4 is 3, and states 12 and 15 are spanned from 16 to 4; state 12 is adjacent to state 15. The cycles of degree 3 and 4 generated by the sequence are given in Fig. 5.

S = 00010111

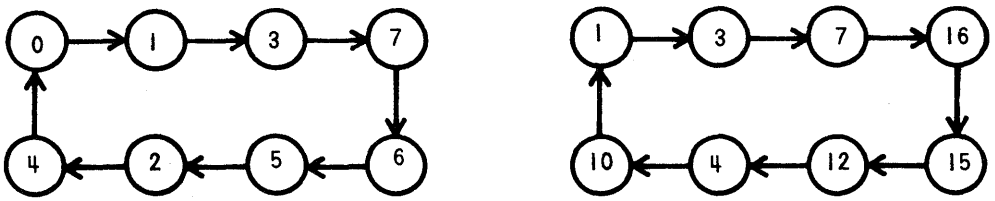


FIG. 5. EXAMPLE OF CYCLES GENERATED BY A SEQUENCE.

Theorem 2: Two states of a feedback shift register with one common successor have both successors in common.

Proof: Let S_i have two successor states, S_a and S_b; and Y_i have two successor states, Y_a and Y_b.

$$S_i = s_1 s_2 \cdots s_n, \qquad Y_i = y_1 y_2 \cdots y_n$$

If S_i and Y_i have one common successor, say $S_a = Y_a$, then:

$$S_a = s_a s_1 s_2 \cdots s_{n-1} = y_a y_1 y_2 \cdots y_{n-1} = Y_a$$

Now

$$S_b = s_b s_1 s_2 \cdots s_{n-1}$$

$$Y_b = y_b y_1 y_2 \cdots y_{n-1}$$

But

$$s_b = \bar{s}_a \qquad y_b = \bar{y}_a \qquad \text{since } S_a \neq S_b \text{ and } Y_a \neq Y_b$$

Since $s_a = y_a$, $s_b = y_b = \bar{s}_a$; hence, $S_b = Y_b$. Note that the pattern shown in Fig. 6 must always exist.

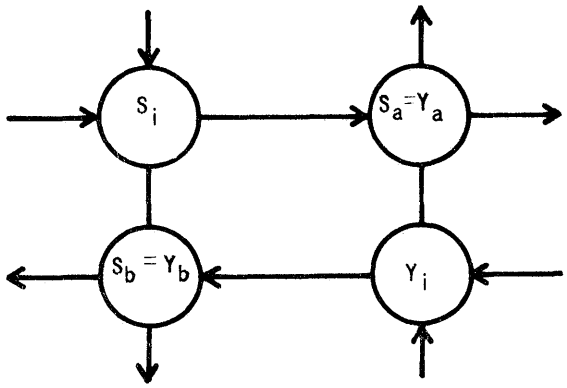


FIG. 6. AN ADJACENCY PATTERN.

The two adjacencies in this pattern have the same adjacency sequence. Further, these four states are the only states in the state graph for the register that contain this adjacency sequence. The term "adjacency quadruple" will refer to the four states containing an adjacency sequence.

B. STATE-GRAPH RELATIONSHIPS

Using Theorem 2, several properties of the state graph of a feedback shift register can be obtained. These properties are useful in determining the type of sequences which can be generated by a shift register. Only state graphs of feedback shift registers will be considered.

Property 1: A nonsingular state graph with only two cycles (C_1 and C_2) can always be modified by changing the successors of two states to produce a maximal cycle. A state on C_1, S_a, is found that is adjacent to some state on C_2, Y_i (see Fig. 7). By choosing the alternate successors for S_i (the state preceding S_a) and for Y_i, the maximal cycle is produced.

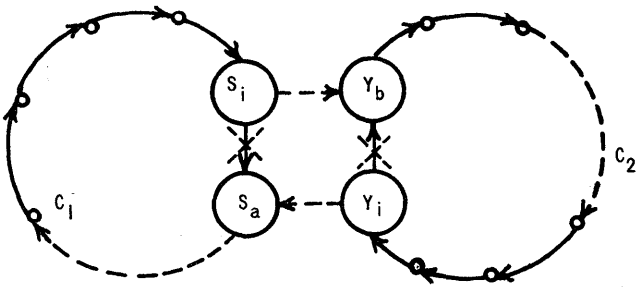


FIG. 7. A STATE GRAPH CONTAINING ONLY TWO CYCLES.

Some state S_a on C_1 must be a possible successor of a state Y_i on C_2, since if this is not so, there can be no maximal cycles. DeBruijn [Ref. 15] and Golomb and Welch [Ref. 12] have shown that there are $2^{2^{n-1}-n}$ maximal cycles possible in a shift register of length n. The state preceding S_a on C_1, S_i has as the other successor Y_b on C_2, which is the successor of Y_i. Then by choosing the appropriate successors for Y_i and S_i the maximal cycle is obtained.

Property 2: A nonsingular state graph containing only two cycles C_1 and C_2 of length L_1 and L_2 can be modified (Fig. 8) to give a single cycle of length L_1 with an appendage of length L_2.

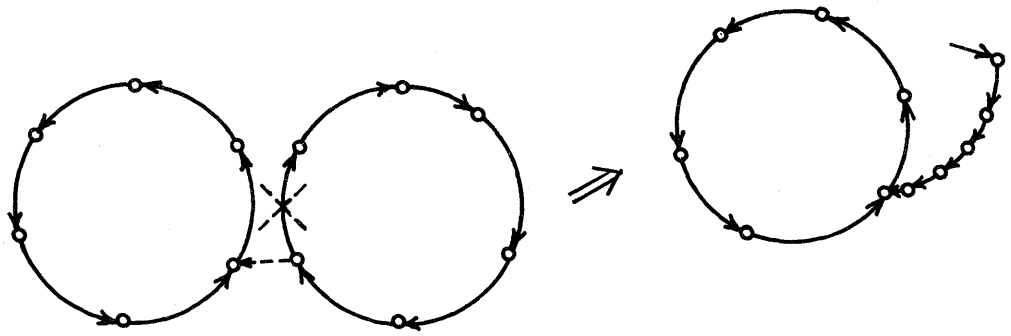


FIG. 8. A MODIFICATION TO PRODUCE AN APPENDAGE.

The same argument as used above can be applied with the choice of the alternate successor for Y_i only.

Property 3: A singular state graph with only one appendage and one cycle can be modified by changing the successor of one state to give a maximal cycle (see Fig. 9).

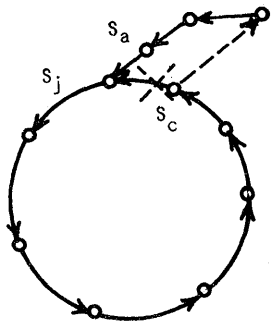
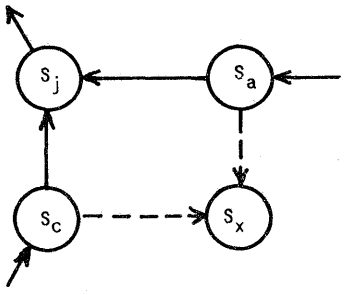


FIG. 9. PRODUCING A MAXIMAL CYCLE FROM A CYCLE WITH AN APPENDAGE.

Consider the state S_j which is the junction of the cycle and the appendage. Its two predecessors are specified, one on the appendage (S_a) and one on the cycle (S_c). Since S_c and S_a have one possible successor in common, by Theorem 2 they must have another. Let S_x denote the second common sucessor of S_a and S_c.



Every state, except the last state on the appendage, has one of its predecessors specified and therefore these states cannot be S_x. The only possible state that can have as its predecessor both S_a and S_c is the last state on the appendage. Hence S_x is the end of the appendage and the modification is possible.

Property 4: A singular state graph with only one cycle and t appendages can be modified to give a maximal cycle by changing the successors of t states (see Fig. 10).

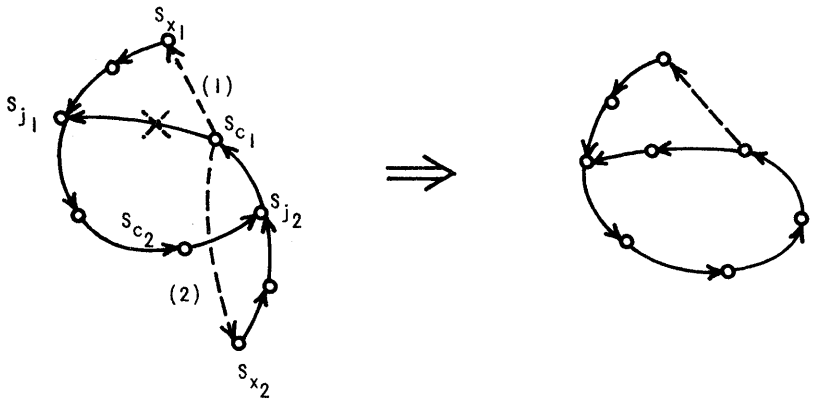


FIG. 10. PRODUCING A MAXIMAL CYCLE FROM A CYCLE WITH TWO APPENDAGES.

By the preceding argument, S_{c_1} must have as one of its successors either S_{x_1} or S_{x_2}. If S_{x_1} is the possible successor of S_{c_1}, modification (1) in Fig. 10 is made, leaving a single cycle and a single appendage. If S_{x_2} is the possible successor of S_{c_1}, modification (2) is made which again leaves a single cycle and a single appendage. Next, modification (3) is made to produce the maximal cycle.

Property 5: An arbitrary graph can be modified to give a maximal cycle, as shown in Fig. 11.

For each appendage there is at most one modification required, and for each pair of cycles joined there is one modification required. The maximum number of modifications needed to produce a maximal cycle is:

$$M \leq t + C - 1$$

where

C = number of cycles in the graph

t = number of appendages in the graph

M = number of modifications required to obtain a maximal cycle

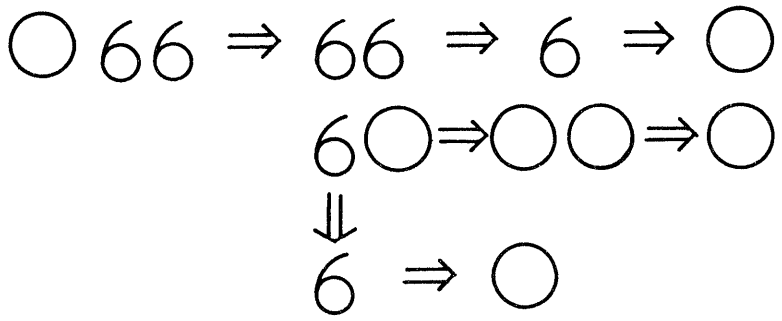


FIG. 11. PRODUCING A MAXIMAL CYCLE FROM AN ARBITRARY STATE GRAPH.

It is interesting to compare the number of maximal linear cycles with the number of maximal nonlinear cycles. There are $\phi(2^n - 1)/n$ maximal linear cycles possible (where $\phi(x)$ is the Euler ϕ function, which is the number of integers less than x that are relatively prime to x). The number of maximal cycles in a nonlinear shift register is $2^{2^{n-1}-n}$ [Refs. 5 and 12]. Table 1 gives the value of these functions for shift registers of length 10 or less.

The cycle set of an n-stage circulating register (see Fig. 2c) will be denoted as follows:

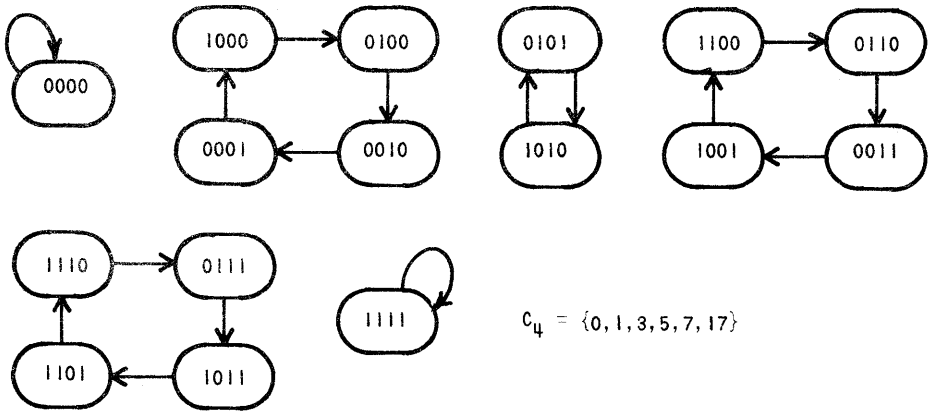
$$C_n = \{C_1, C_2, \ldots, C_k\}$$

The C_i are representative states of each cycle of the register. The binary sequence which represents the given state generates the cycle containing that state. The representative states will be given as octal numbers with the least significant digit on the left. For convenience, the representative state will be chosen to be that state which has the smallest number.

TABLE 1. NUMBER OF MAXIMAL CYCLES POSSIBLE WITH A FEEDBACK SHIFT REGISTER

n	$2^{2^{n-1}-n}$	$\frac{\phi(2^n-1)}{n}$
2	1	1
3	2	2
4	16	2
5	2048	6
6	25, 165, 951	6
7	2×10^{16}	18
8	19.4×10^{34}	16
9	4×10^{73}	48
10	32×10^{149}	60

Example: The cycles of a four-stage circulating register are:



$C_4 = \{0, 1, 3, 5, 7, 17\}$

All states on the same cycle of C_n have the same measure since all the states are obtained from a representative state by permuting its digits.

C. SEQUENCE THEOREMS

Theorem 3: Two cyclic sequences that are identical for at least $L_1 + L_2$ digits (where L_1 and L_2 are the lengths of the respective cycles) are identical for all time.

Proof: Let the sequences be:

$$S_1 = x_1 x_2 \cdots x_{L_1} x_{L_1+1} \cdots x_{2L_1} \cdots x_{L_1+L_2} x'_k$$

$$S_2 = x_1 x_2 \cdots x_{L_2} x_{L_2+1} \cdots x_{L_1+L_2} x_k$$

where L_2 is the longest cycle. By hypothesis the first $L_1 + L_2$ digits are identical. Let x'_k be the first digit of S_1 that can be different from the corresponding digit in S_2. Since S_1 is cyclic with length L_1:

$$x'_k = x_{k-L_1} = x_{L_1+L_2+1-L_1} = x_{L_2+1}$$

The first $L_1 + L_2$ digits of S_1 are cyclic with length L_2 so:

$$x_{L_2+1} = x_1$$

$$x'_k = x_1$$

By an identical argument, $x_k = x_1$. The same argument is repeated for the $(k + 1, k + 2, \ldots)$ digits to show that the sequences are identical for all time.

Theorem 4: Two cyclic sequences C_1 and C_2 representing different cycles of C_n whose lengths are less than n cannot have adjacent states.

Proof: Since the cycle length of each sequence is less than n (n is the degree of the states on the cycle), it must be a divisor of n. The longest such sequence has length $L = n/2$. To have adjacent states the sequences must be identical for $n - 1$ digits. If $L_1 \neq L_2$ by Theorem 3, the sequences are identical for all time and cannot represent different cycles of C_n. If $L_1 = L_2$, the sequences have the following form:

$$C_1 = x_1 x_2 \cdots x_L x_1 x_2 \cdots x_{L-1} \cdots x_L$$

$$C_2 = x_1 x_2 \cdots x'_L x_1 x_2 \cdots x_{L-1} \cdots x'_L$$

To be states on different cycles of C_n, $x_L \neq x'_L$. Then the first $n - 1$ digits of C_1 are not the same as the corresponding digits of C_2, so the cycles have no adjacent states.

In the next chapter a method of obtaining the output sequences producible with a feedback shift register is given. Two operations are defined in terms of the adjacencies between the cycles of a particular shift register, which describe the modifications necessary to produce the output sequences for an arbitrary specification.

Definition: Two cyclic sequences are said to be joined if a sequence is obtained which is cyclic and contains all states that were contained on the original two sequences and no others. The length of the joined sequence must be the sum of the lengths of the two sequences to be joined.

D. METHODS OF REPRESENTING ADJACENCIES

In the next chapter it is shown that a necessary and sufficient condition for joining two cycles together is that they possess adjacent states. To accomplish the cycle joining, the adjacent states need to be

known. A knowledge of the adjacencies between cycles and their location is sufficient to describe the entire cyclic behavior possible in the register. The cycle structure for a very simple configuration--the circulating register--and the adjacencies between cycles will be described. A similar description can be given for any nonsingular register and its cycles used to generate an arbitrarily specified behavior. The circulating register is chosen because of its simple logic and well understood cycle structure.

Once the adjacencies between cycles have been found, they can be recorded for future reference. A convenient method for recording the adjacencies is necessary. Lists or tables can be used for such purposes, but at least for $n \leq 8$, a diagram which shows the cycles and their adjacencies is more useful. Several properties of the adjacencies between cycles of the circulating register are deduced by considerations based on these diagrams. As n becomes large, the value of the diagrams decreases and tables of adjacencies are more desirable. Many of the properties of the adjacencies deduced from the diagram will be useful for constructing tables for larger values of n.

E. CONSTRUCTION CONVENTIONS FOR ADJACENCY DIAGRAMS

Some restrictions on the cycles that can have adjacencies will aid in constructing adjacency diagrams. Whenever a circulating-register cycle is not followed, the measure of the adjacent states must differ by one. Thus, one need only look for adjacencies between cycles whose measure differs by one. Theorem 4 establishes that two cycles of C_n whose length is less than n do not have adjacencies. Theorem 2 indicates that the adjacencies occur in pairs, so that any two cycles with adjacency connection between two corresponding states also have the reverse adjacency connection. As indicated in the proof of Theorem 2, both adjacencies and all four adjacent states are defined by a common $(n - 1)$-digit sequence. Thus, in the construction of the diagram, it is necessary to include the common $(n - 1)$-digit sequence only once to

define the adjacency quadruple. Each different (n - 1)-digit word defines a different adjacency quadruple. The convention used to indicate the adjacency quadruple is shown in Fig. 12.

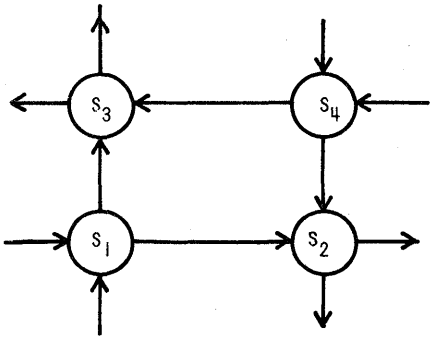


FIG. 12. ADJACENT STATES DEFINED BY AN ADJACENCY SEQUENCE.

$$S_1 = x_1x_2x_3 \cdots x_n$$
$$S_2 = \bar{x}_n x_1 x_2 \cdots x_{n-1}$$
$$S_3 = x_n x_1 x_2 \cdots x_{n-1}$$
$$S_4 = x_1 x_2 \cdots x_{n-1}\bar{x}_n$$
$$A = \text{adjacency sequence}$$
$$= x_1 x_2 \cdots x_{n-1}$$

The adjacency diagram is constructed by ordering the cycles of C_n according to their measure. Each node in the diagram contains the representative state of the cycle and the length of the cycle. Each branch is labeled with the numbers giving the octal representation of the (n - 1)-digit words describing the adjacencies between the cycles represented by the nodes at the ends of the branch. Adjacency diagrams for $n \leq 8$ are given in Fig. 13 on pages 38 and 39.

F. PROPERTIES OF ADJACENCY DIAGRAMS

Before observing several interesting properties of these diagrams, the following definitions will be useful:

1. A cycle is said to be contained between two cycles (C_1 and C_2) if it has fewer states than the sum of the number of states on C_1 and C_2 and contains only states that are on C_1 and C_2.
2. A cycle has an n intersection with another cycle if it has n states in common with the other cycle.
3. An adjacency diagram for an n-stage circulating shift register (A_n) is a diagram showing all the adjacencies between cycles representing transitivity sets. Since the adjacencies occur in pairs (Theorem 2) only one member from each pair is shown on the diagram.
4. An adjacency sequence is an (n - 1)-digit sequence which defines an adjacency quadruple between two cycles.
5. The inverse of a sequence is the sequence which is obtained from the given sequence by inverting the order of occurrence of the digits. For example:

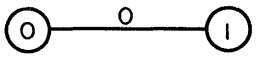
$$S = x_1 x_2 \cdots x_n$$

$$S^{-1} = x_n \cdots x_2 x_1 = \text{inverse of } S$$

A study of the adjacency diagrams for $n \leq 8$ given in Fig. 13 reveals the following facts:

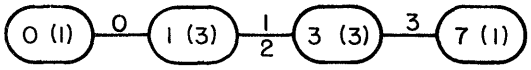
1. There are at most two adjacencies between any two cycles of C_n.
2. If there are two adjacencies, their adjacency sequences are related by inversion. (Note: This is not true in general for $n > 8$.)
3. The number of adjacencies between two levels of A_n is equal to the number of (n - 1)-digit words whose measure is equal to that of the lowest measure level.
4. If there are two adjacencies between two cycles, there is a cycle of length less than n contained between the two cycles of C_n.
5. Adjacencies occur only between cycles whose measure differs by one.
6. There are no adjacencies between two cycles of C_n whose length is less than n.

A_1

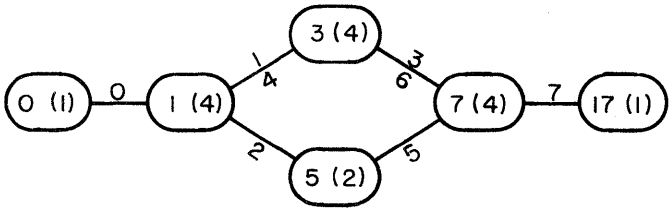


A_2

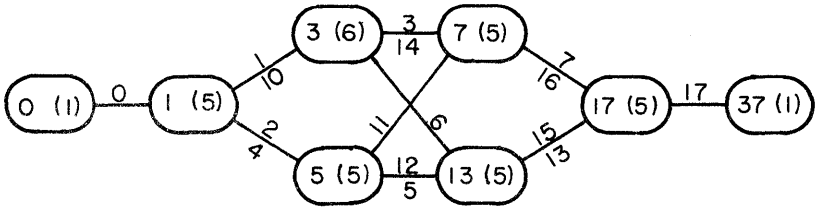
A_3



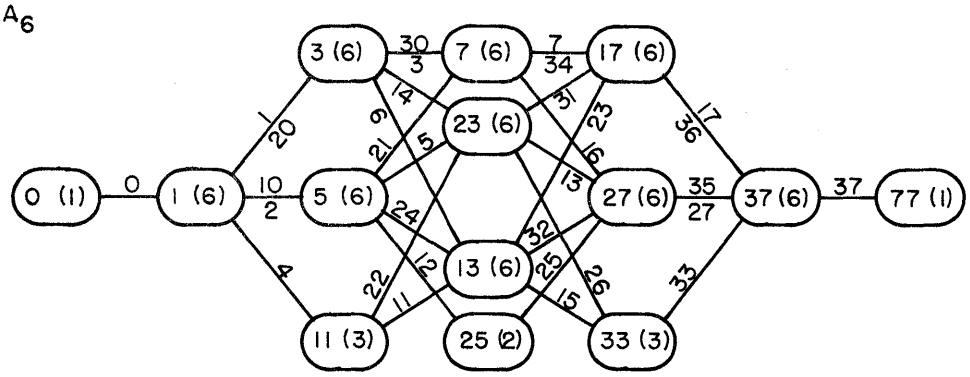
A_4



A_5



A_6



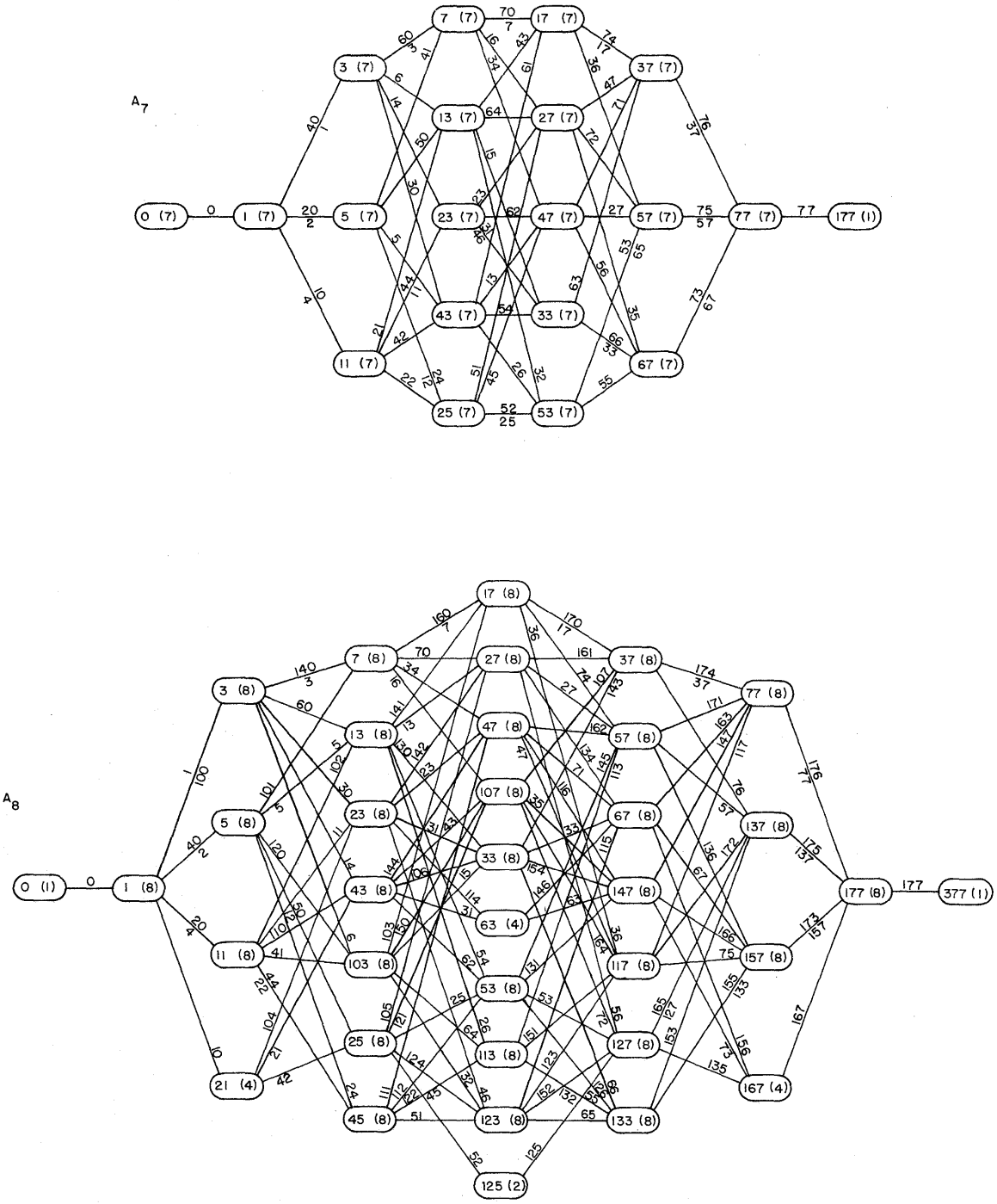


FIG. 13. ADJACENCY DIAGRAMS.

These observations lead to the following theorems:

Theorem 5: When one cycle (C_1) is contained between two cycles (C_a and C_b), the remaining states are on another cycle contained between C_a and C_b (see Fig. 14).

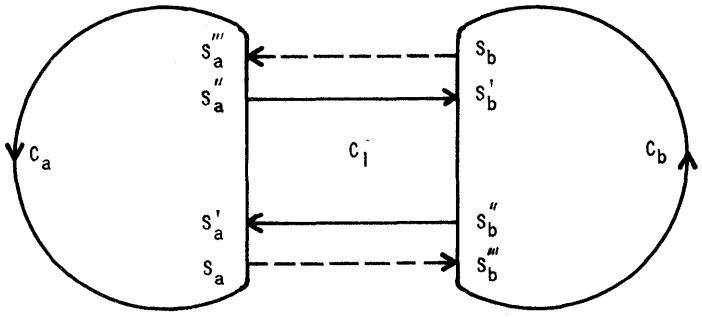
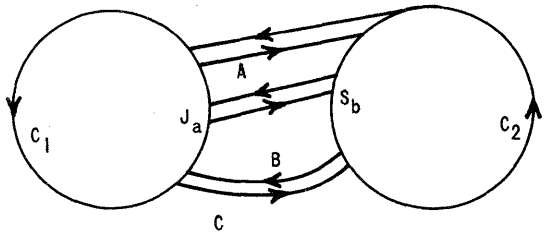


FIG. 14. A CONTAINED CYCLE.

Proof: This follows directly from Theorem 2. If there is a cycle contained between two cycles, then there are two different adjacency sequences. The dotted connections from S_b to S_a''' and from S_a to S_b''' result from application of Theorem 2. Thus the states of C_a and C_b that are not on C_1 are on an additional cycle which is contained between C_a and C_b.

Theorem 6: Only one cycle with length less than n is contained between two cycles of C_n.



Proof: If only two adjacencies exist between the two cycles of C_n, then there are two cycles contained between the cycles of C_n. Only one of them can have length less than n since the sum of the two must have length $2n$.

If three or more adjacencies exist, then there may be three or more cycles contained between the cycles of C_n. Two of these cycles must have length less than n while only one cycle can have length longer than n. Call the two short cycles A and B and let C be the cycle whose length can be longer than n. If there are more than three cycles, let B be the shortest cycle that is adjacent to A. (Note: If A is shorter than n, B must be also since there are $2n$ states to be divided into three or more cycles.) It will be shown that if both A and B are shorter than n, C_1 cannot be a cycle of C_n and hence there cannot be two or more cycles shorter than n contained between two cycles of C_n.

Let S_b be the first state on B which is on C_2, and let S_a be the first state on A which is on C_1. Let S_B be the sequence which generates the cycle B, and S_A the sequence that generates the cycle A. The sequences are written with S_a and S_b as the first n digits of their corresponding sequence.

There are three different cases to consider.

Case I: $L < \frac{n}{2}$

$P < \frac{n}{2}$

where L = length of A and P = length of B. The general form of the sequences is:

$$S_A = x_1 x_2 \cdots x_L x_1 x_2 \cdots x_L x_1 \cdots x_L \cdots x_n$$

$$S_B = x_1 x_2 \cdots x_P x_1 \cdots x_P \cdots x_n$$

Case II: $L > \frac{n}{2}$

$P > \frac{n}{2}$

The general form of the sequences is:

$$S_A = x_1 x_2 \cdots x_L x_1 \cdots x_n \cdots x_L$$

$$S_B = x_1 x_2 \cdots x_P x_1 \cdots x_n \cdots x_P$$

Case III: $L > \frac{n}{2}$

$P > \frac{n}{2}$

The general form of the sequences is:

$$S_A = x_1 x_2 \cdots x_L x_1 x_2 \cdots x_L \cdots x_n$$

$$S_B = x_1 x_2 \cdots x_P x_1 x_2 \cdots x_n \cdots x_P$$

For all cases, the first state on each sequence (S_a and S_b) has an (n - 1)-digit terminal word in common. This occurs because S_b is obtained from S_a by reversing one state on C_1 and then proceeding along the adjacency connecting C_2 and C_1 by shifting the digits and inserting $\bar{x}_1$ as the first digit.

$$S_a = x_1 x_2 \cdots x_n$$

$$S_b = \bar{x}_1 x_2 \cdots x_n$$

A proof will be given for each case.

$$\underline{\text{Case I}}: \quad S_A = \overbrace{x_1 x_2 \cdots x_n \cdots x_L x_1 x_2 \cdots x_n \cdots x_L}^{S_a} \cdots x_L$$

$$S_B = \underbrace{\bar{x}_1 x_2 \cdots x_n \cdots x_L \cdots x_P \bar{x}_1 \cdots x_L \cdots x_P \cdots x_n}_{S_b} \cdots x_p$$

The two sequences are identical for a time that is equal to or longer than twice their period ($x_2 \ldots x_n$); hence they must be identical for all time and therefore x_1 on S_A cannot equal $\bar{x}_1$ on S_B (Theorem 4).

Case II: Suppose the lengths of A and B are equal, then S_A and S_B will have the following form:

$$S_A = \overbrace{x_1 x_2 \cdots x_n \cdots x_L x_1 \cdots x_n}^{S_1} \cdots x_L$$

$$S_B = \underbrace{\bar{x}_1 x_2 \cdots x_n \cdots x_L x_1 \cdots x_n}_{S_1'} \cdots x_L$$

Then S_1 is not the same as S_1', so the two sequences do not contain adjacent states.

Now consider the case where the length of A is not equal to that of B. Let A be the shorter cycle. Then the form of the sequences representing the cycles will be determined as follows.

First, since the sequences are cyclic with lengths L and P, they must have the following general form:

$$S_A = \underbrace{x_1 x_2 \cdots x_L x_1 x_2 \cdots x_n x_i}_{S_a} \cdots x_L$$

and

$$S_B = \underbrace{\bar{x}_1 x_2 \cdots x_P \bar{x}_1 x_2 \cdots x_n \bar{x}_i}_{S_b} \cdots x_P$$

The digit x_i in S_A following x_n must be different from the corresponding digit in S_B or the two cycles will contain a common state.

Next, since the two states S_a and S_b have a common $(n - 1)$-digit terminal word, additional structure of the two sequences is specified:

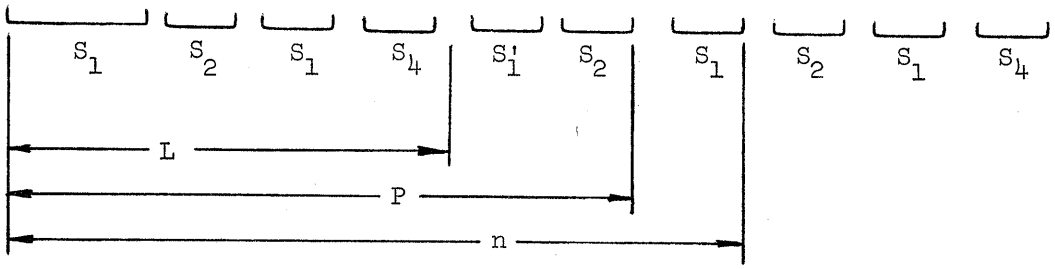
$$S_A = x_1 x_2 \cdots x_L x_1 \cdots x_P \bar{x}_1 \cdots x_n x_i \cdots x_L$$

$$S_B = \bar{x}_1 x_2 \cdots x_L x_1 \cdots x_P \bar{x}_1 \cdots x_n \bar{x}_i \cdots x_P$$

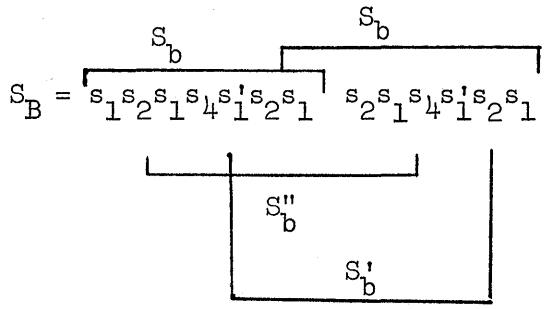
The first occurrence of the sequence $x_1 x_2 \cdots x_L$ in S_A is different from the second. Since S_A is cyclic with length L, the sequence $x_1 \cdots x_P \bar{x}_1 \cdots x_n x_i$ must appear in the first L digits as well as in the second L digits. Also, S_B must be cyclic with length P so the first and second P-digit sequences must be identical. Imposing these conditions, the structure of the sequence becomes as follows:

$$S_A = x_1 x_2 \cdots x_n \bar{x}_i \cdots x_P \bar{x}_1 \cdots x_n x_i \cdots x_L x_1 \cdots x_n \bar{x}_i \cdots x_P \bar{x}_1 \cdots x_n x_i \cdots x_L x_1 x_2 \cdots x_n \bar{x}_i \cdots$$

$$S_B = \bar{x}_1 x_2 \cdots x_n \bar{x}_i \cdots x_P \bar{x}_1 \cdots x_n x_i \cdots x_L x_1 \cdots x_n \bar{x}_i \cdots x_P \bar{x}_1 \cdots x_n \bar{x}_i \cdots x_P x_i \cdots x_n x_i \cdots x_L \cdots$$



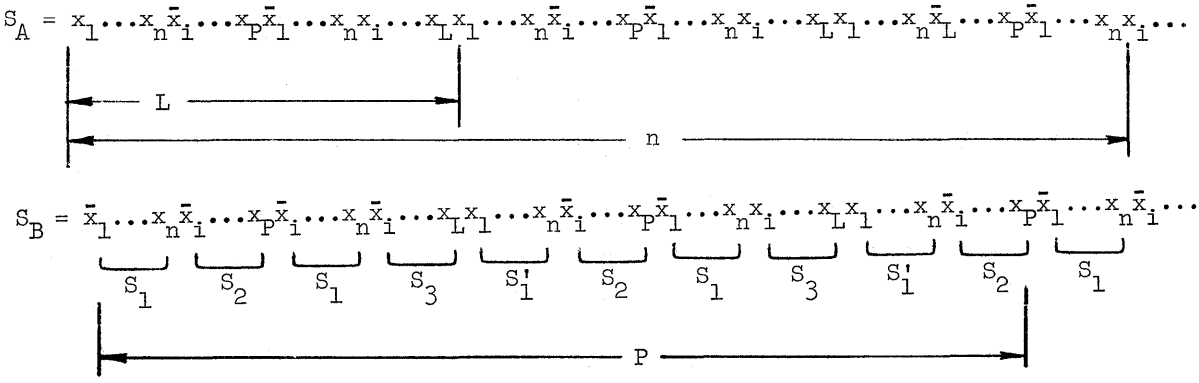
Represent the sequence S_B as a series of subsequences as defined above:

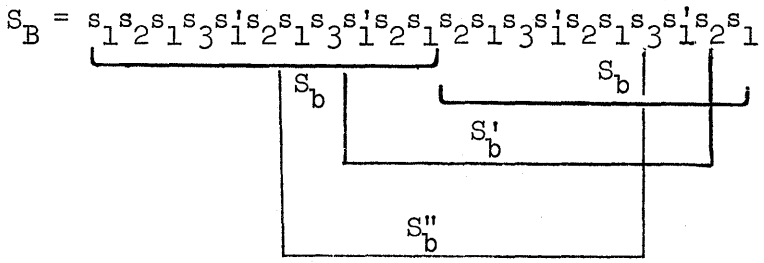


The measure of S_b and S_b'' is the same since they contain the same subsequences. Since S_b is the first state on B that is on C_2 and since S_b'' is also on C_2, all states between S_b and S_b'' are also on C_2. The measure of S_b' is different from the measure of S_b since it contains the same subsequences except for the replacement of S_1 with S_1' whose measure differs from that of S_1. Since S_b' and S_b are both on the cycle C_2, C_2 cannot be a cycle of C_n.

Case III: The sequences will have a slightly different structure when $P > 2L$ than when $P < 2L$. These two possibilities will be considered separately. The sequences have the form:

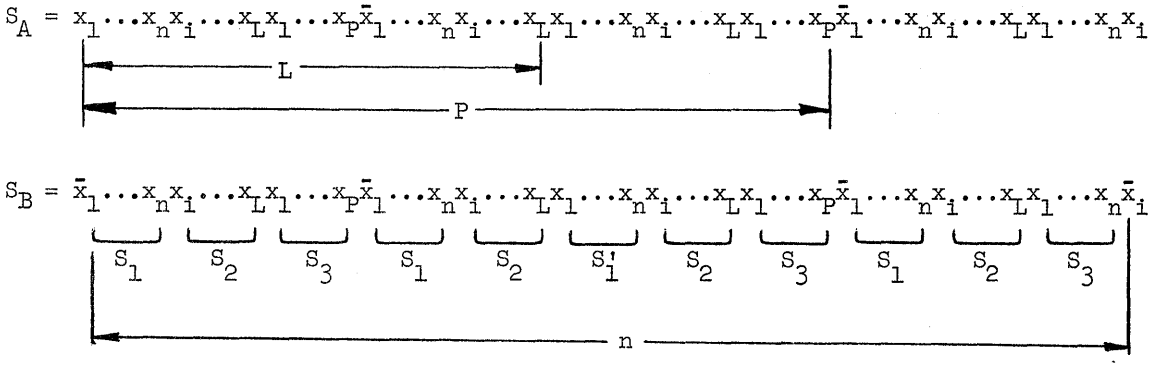
(a) $P > 2L$

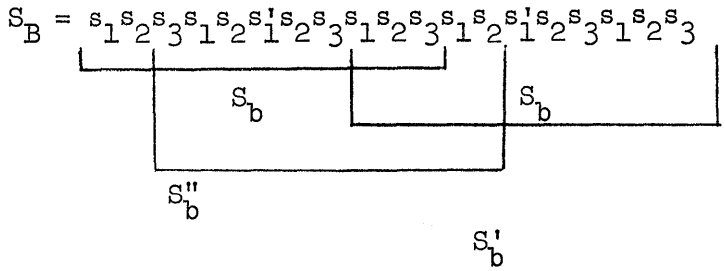




The state S_b' has measure of one different from S_b since it contains the same subsequences as S_b except S_1 is replaced by S_1'. Since all states of C_2 are on the same cycle of C_n, they have the same measure; so S_b' cannot be on C_2 and thus must be on C_1. The state S_b is the first state on C_2 which is on B, so all the states following S_b' but preceding S_b must be on C_1. Now it is seen that S_b'' which is on C_1 has measure different from S_b', so C_1 cannot be a cycle of C_n.

(b) $P < 2L$

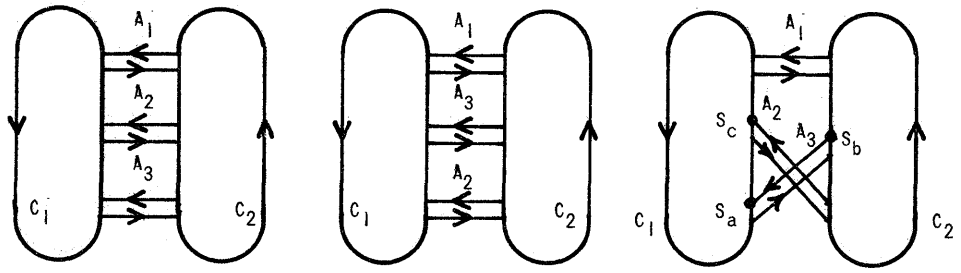




The state S_b' has measure different than S_b and as before must be on C_1, as are all states following S_b' and preceding S_b. If C_1 contains a state of measure different than S_b', it cannot be a cycle of C_n. Such a state is S_b'' whose measure is different than S_b'; so C_1 cannot represent a cycle of C_n.

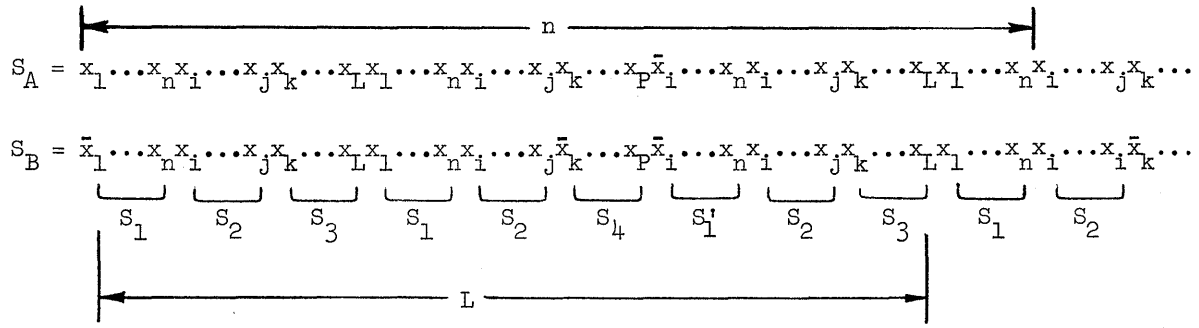
Theorem 7: There are at most two adjacency pairs between two different cycles of C_n.

Proof: Let there be two adjacency pairs (A_1 and A_2) between two cycles (C_1 and C_2) of C_n; there are three ways by which it might be possible to draw a third (A_3):



The first two possibilities are not really different except for a relabeling. They both violate the previous theorem and hence cannot exist. The third possibility is different from the first two since if all three adjacency pairs are used, the two cycles are joined to form a single cycle. This third possibility is similar to the first two in that three cycles are formed, but two of them have δ overlap, where δ is the number of states on the sequence from S_c to S_b.

Let S_c be the first state on C_1 preceding A_1 that is adjacent to some state on C_2. Let S_a be the first state on C_1 preceding S_c that is adjacent to some state on C_2. S_b is the state following the state on C_2 which is adjacent to the state preceding S_a.



Two slightly different cycle structures are possible:

1. A and B are equal in length.

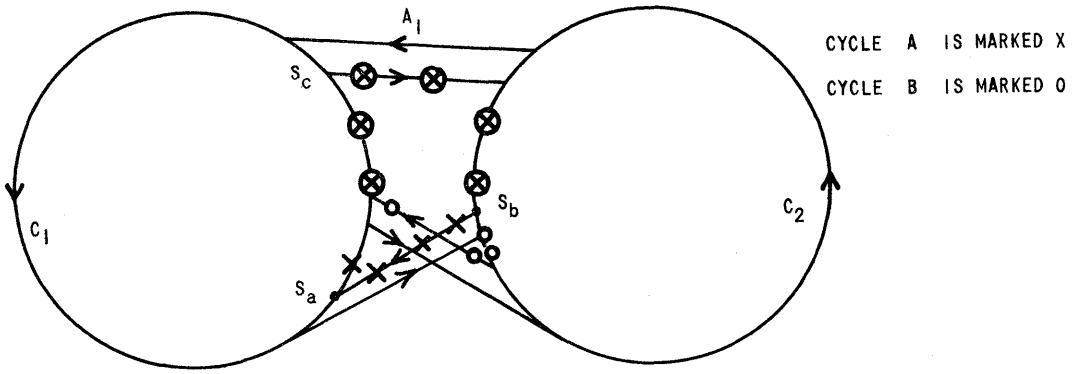
$$S_A = x_1 x_2 \cdots x_L x_1 \cdots x_n \cdots x_i x_n \cdots x_L$$

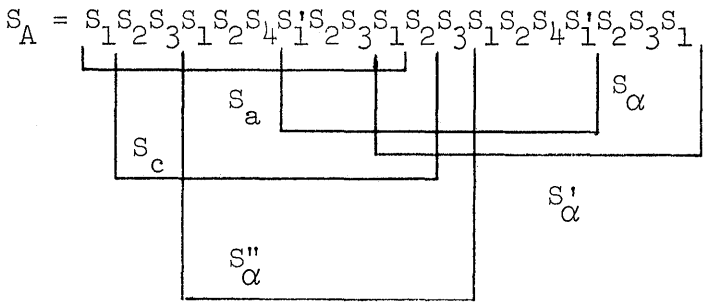
$$S_B = \bar{x}_1 x_2 \cdots x_L \bar{x}_1 \cdots x_n \cdots x_j \bar{x}_k \cdots x_L$$

But $x_2 \cdots x_j$ in $S_A = x_2 \cdots x_j$ in S_B

Now s_1 in A is not equal to $\bar{x}_1$ in B, and x_k in A is not equal to $\bar{x}_k$ in B, so the sequence cannot represent different cycles.

2. A and B are unequal in length; let A be the longer cycle.





States S_a and S_c are both on C_1; so all states on S_A between them are also on C_1. State S_a' contains the same subsequences as does state S_a except with S_1 replaced with S_1' which has different measure. Thus the cycle C_1 has states with different measure and cannot be a cycle of C_n. This completes the proof of the theorem.

This theorem can be stated in several equivalent ways:

1. Two cycles of degree n generated by sequences representing cycles of C_{n+1} can have at most a double intersection in graph G_n. (G_n is the Good diagram for an n-digit register.)
2. Two cycles of C_n can have at most two common (n - 1)-digit words.

Conjecture: When there are two adjacencies between two cycles of C_n, the second adjacency sequence is related to the first by inversion when n and n - P (P equals the length of the shortest cycle contained between the two cycles) are relatively prime.

This conjecture has not as yet been proved, but in all examples studied for n as large as 30 it has been found to be true. It is known to be true for $n \leq 8$ by exhaustive analysis. The reasons for posing the conjecture follow from the process of removing a contained cycle, given in Chapter IV.

The use of the conjecture simplifies the construction of the adjacency diagram since specification of some adjacencies determines others. Also, the removal of contained cycles as indicated in Chapter IV, is an aid in locating the cycles of C_n that have two adjacencies. These aids make the construction of diagrams for small n very simple. For $n > 10$, an adjacency table rather than a diagram is more useful. The process of searching for adjacencies for longer registers subject to the above-mentioned restrictions should be accomplished easily with a computer.

With the aid of an adjacency diagram, it is easy to see which cycles of C_n can be joined and which adjacency sequence is required for this joining operation. Using the method developed, one can then determine the output sequence which corresponds to this joined pair of cycles. The network specification which realizes the joined cycles will be determined from the knowledge of the adjacency used.

IV. SHIFT-REGISTER SYNTHESIS IN THE SEQUENCE DOMAIN

A. SUMMARY OF GENERAL APPROACH

The previous chapter gives several useful properties of shift-register sequences. The purpose of this chapter is to use these properties to develop a synthesis procedure. First, the realizable output sequences are obtained by operating on the output sequences of a circulating shift register. Then the feedback network specification is determined from the output sequences. Finally, a method is given for finding the feedback network directly from a knowledge of the operations used on the cycles of C_n.

Two operations on the cycles of C_n are developed. A joining operation is used to find the output sequence generated by the cycle containing the states on two or more cycles of C_n. A removal operation is used to find the shorter output sequences contained between two cycles of C_n. A proof is given which establishes the generality of these operations; that is, any output sequence which is producible by a shift register can be expressed in terms of these operations on C_n. If the synthesis procedure fails at some point, the specification cannot be realized by a shift register. Thus, the synthesis procedure is a realizability test, although a rather awkward one.

When the specification of the register is in terms of a cycle set, it is first determined if the specification can be achieved by several isolated registers. This is done by factoring the cycle set in a manner similar to that of Elspas [Ref. 5]. This factoring is not essential in some cases, but it will simplify the remaining decomposition problem. In general, if the number of k cycles is equal to or less than $I(k)$ (the number of irreducible polynomials of degree k) for all k in the prescribed cycle set, a single register may possibly be obtained. If k is greater than $I(k)$, then factorization is necessary.

After the factoring has been completed, or if the original specification does not include more than $I(k)$ cycles for any k, the cycles are decomposed in terms of C_n. Let n be the smallest integer such that $2^n \geq \sum_{i=1}^{P} k_i$, where $k_1 k_2 \ldots k_P$ are the cycle lengths in the specification.

Then the register needs to contain at least n stages. In the synthesis procedure the prescribed cycles must be expressed in terms of C_n such that they are obtained by either joining two cycles or removing a cycle which is contained between two or more cycles. The cycles that can be expressed in terms of the joining operation can be decomposed as follows:

Let the prescribed cycle set be:

$$CS = \{k_1, k_2, \ldots, k_P\}$$

A cycle k_j is expressible in terms of C_n if:

$$k_j = a_0 n + a_1 m_1 + \ldots + a_L m_L$$

where the a_i are integers and the m_i divide n.

Any cycle in the prescribed set not expressible in the above form must be obtained by removing a cycle from between two or more cycles of C_n. It may be necessary to first join two pairs of cycles together and then remove the cycle from between the two longer cycles.

A generating function is used to express the cycle set in terms of joining and removing operations on C_n. A simple relation between the generating function and the feedback network allows the number of the feedback network to be written by inspection. The generating function determines the cyclic behavior of the register. Thus, for the nonsingular shift registers, the function completely determines the register's behavior. When the behavior of the register is not completely specified, the generating function will not completely determine the characteristic number of the feedback network. The unspecified entries of the characteristic number are treated as Don't Cares in the realization of the combinational network.

A convenient method of keeping track of the cycles of C_n which have been used in a particular realization is given in terms of a table similar to the prime implement table used in the minimization of switching functions. In general, there will be more than one generating function that will give a particular cyclic behavior. In such cases there will be more than one function that will cover the table of cycles

of C_n. The particular generating function that leads to the simplest combinational network must be determined by additional considerations. The function used depends on the network configuration and type of logic, such as AND-OR logic or threshold logic. This is a separate problem and is not considered here.

The joining operation is a little simpler to use and in some cases leads to a simpler network; thus, it is used whenever possible. This policy forces the coding of the cycles to be as close to that of C_n as possible. The method which is given for removing a cycle from between cycles of C_n is systematic and fairly easy to follow. It is used only when the prescribed behavior is not obtainable by simply joining cycles of C_n.

B. SEQUENCE JOINING

Two sequences, S_1 and S_2 cyclic with degree n of length L_1 and L_2 respectively, each representing a cycle of C_n, can be joined if the sequences have at least one pair of adjacent states. Let A be the adjacency sequence which defines the adjacency between a state on S_1 and a state on S_2. The joined sequence will be denoted $S_1 J_A S_2$. If the sequences have more than one pair of adjacent states, a different joined sequence can be obtained for each pair of adjacent states.

Joining Procedure: The process of joining two sequences produces a sequence representing the cycle containing all the states of the original sequences. This process is defined as follows: Let s_1 be the state on S_1 adjacent to state s_2 on S_2 (see Fig. 15). Let s_3 be obtained by reversing the cycle which S_2 generates (C_2) and proceeding backward $L_1 + L_2 - (n + 1)$ states on the cycle. State s_3 has an overlap of $2n - (L_1 + L_2)$ digits with s_1. The joined sequence is obtained by juxtaposing s_1 and s_3 with s_3 on the right, with overlapping digits removed.

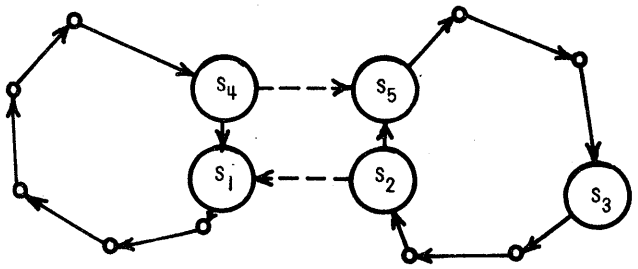


FIG. 15. JOINED CYCLES.

Proof of Joining Procedure: First, it is necessary to show that states s_3 and s_1 have an overlap of $2n - (L_1 + L_2)$ so that the joining process is possible. Since s_2 is adjacent to s_1, it has an $n - 1$ overlap with s_1.

$$s_1 = a_1 a_2 \cdots a_n \qquad s_2 = b_1 b_2 \cdots b_n$$

To be adjacent, $a_n = b_{n-2}$

$$a_{n-1} = b_{n-2}$$

$$\vdots$$

$$a_2 = b_1$$

Let r_{21} denote the overlap of state s_2 with state s_1. Proceeding on C_2 in the reversed direction from s_2, each succeeding state has an overlap with s_1 which is one less than its predecessor. Thus, after proceeding backward $L_1 + L_2 - (n + 1)$ states on the cycle, the overlap has been reduced to:

$$r_{s_3} = n - 1 - L_1 + L_2 - (n + 1)$$

$$= 2n - (L_1 + L_2)$$

The juxtaposed sequence has length $L_1 + L_2$, so if it is cyclic and contains all states of C_1 and C_2 it satisfies the definition of a joined sequence. According to Theorem 2, the state preceding s_1 (s_4) is adjacent to the state following s_2 (s_5). The operation of proceding in the reversed direction on S_2 to obtain s_3 and juxtaposing insures that the states spanned from s_3 to s_1 are on the cycle generated by the juxtaposed sequence.

The span from s_1 to s_3 is shown in Fig. 16.

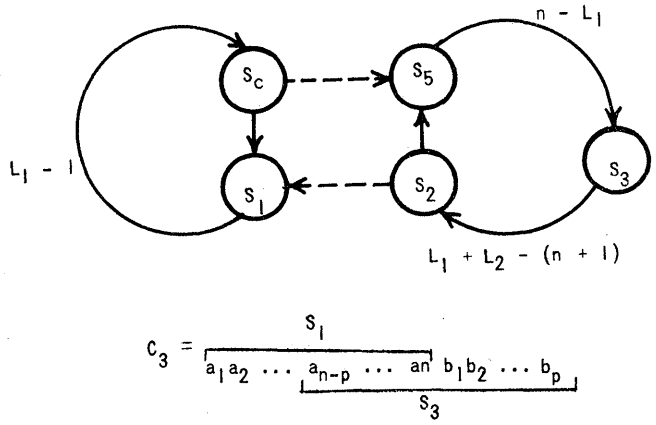


FIG. 16. THE SPAN OF SEQUENCES ON JOINED CYCLES.

Let C_3 be the cycle generated by the juxtaposed sequence:

$$C_3 = \overbrace{a_1 a_2 \cdots \underbrace{a_{n-L_1} \cdots a_n}_{s_2} b_1 b_2 \cdots b_P}^{s_1}$$

The length of $C_3 = L_3 = L_1 + L_2$.

Let $L_{1,3}$ = span from s_1 to s_3

$L_{3,2}$ = span from s_3 to s_2

$= L_1 + L_2 - (n + 1)$ (by construction)

$L_{2,1} = 1$

$L_{1,3} = L_3 - [L_{32} + L_{21}]$

$= L_1 + L_2 - [L_1 + L_2 - (n + 1) + 1]$

$= n$

To see which n states are spanned on the sequence, start with the two states known to have an overlap, s_4 and s_5.

Let $s_1 = a_1 a_2 \cdots a_n$ $\quad s_4 = a_2 a_3 \cdots a_n a_1$

$s_2 = a_2 \cdots a_n b_n$ $\quad s_5 = b_n a_2 \cdots a_n$

The states s_4 and s_5 have an $n - 1$ overlap. Each shift in the forward direction along S_2 to s_3 reduces this overlap by one, so state s_3 has an overlap of $n - 1 - (n - L_1) = L_1 - 1$. Proceeding in the reversed direction on C_1 from s_4, each shift again reduces the overlap by one so the first state on C_1 that has zero overlap with s_3 is s_1, which is $L - 1$ states from s_4 on the reversed cycle. State s_1 has an overlap of $n - L_1$ with s_4 and hence an overlap of $n - L_1 - 1$ with s_5.

The states on the cycle of degree n generated by C_3 are on the following sequence (C_3^1):

$$C_3^1 = \overbrace{a_1 a_2 \cdots \underbrace{a_{n-P} \cdots a_n}_{s_3}}^{s_1} b_1 \cdots \overbrace{b_P}^{y_1} a_1 a_2 \cdots a_{n-1}$$

Let x_1 be the state on C_1 following s_1

x_2 be the state on C_1 following x_1

x_k be the state on C_1 following x_{k-1}

and let y_1 be the state on C_3 following s_1

y_2 be the state on C_3 following y_1

y_k be the state on C_3 following y_{k-1}

then

$$x_1 = c_1 a_1 a_2 \cdots a_{n-1}$$

$$x_2 = c_2 c_1 a_1 a_2 \cdots a_{n-2}$$

$$\vdots$$

$$x_P = c_P \cdots c_2 c_1 a_1 \cdots a_{n-P}$$

Since x_1 has a one overlap with s_3, $c_1 = b_P$.

Since x_2 has a two overlap with s_3, $c_2 = b_{P-1}$.

Since x_P has a p overlap with s_3, $c_P = b_1$.

By inspecting C_3^1, y_1, y_2, ..., y_P are seen to be:

$$y_1 = b_P\, a_1 a_2 \cdots a_{n-1} = x_1$$

$$y_2 = b_{P-1}\, b_P\, a_1 \cdots a_{n-2} = x_2$$

$$\vdots$$

$$y_P = b_1 \cdots b_P\, a_1 \cdots a_{n-P} = x_P$$

So C_3^1 contains the states on C_1 and C_2.

Theorem 4 insures that two cycles of C_n whose lengths are less than n cannot have adjacent states. Thus the distance from state S_2 to S_3 in the above procedure $(L_1 + L_2 - n - 1)$ will always be a positive integer less than n. If, however, the cycles to be joined are not cycles of C_n, care must be taken to insure that their cycle lengths are not too short.

C. AN EXAMPLE

Consider for example the joining of two cycles of C_4: $C_1 = 0001$ and $C_5 = 0101$. First, notice that state 4 on C_1 is adjacent to state 12 on C_5. C_1 and C_5 are cyclic with lengths 4 and 2 respectively. State 5 is obtained from state 12 by proceeding $L_1 + L_2 - (n + 1) = 1$ state on the reversed cycle generated by C_5. Justaposing state 5 on the left of state 4 with the $2n - (L_1 + L_2) = 2$ overlapping digits removed produces 001010 as the joined sequence. The cycle generated by the joined sequence is illustrated in Fig. 17.

D. JOINING LONGER CYCLES

A joining procedure for longer cycles can be obtained by placing a restriction on the sequence length. As was seen in Theorem 4, two cycles of C_n with length less than n cannot have adjacent states. In order to join two cyclic sequences having adjacent states in the manner

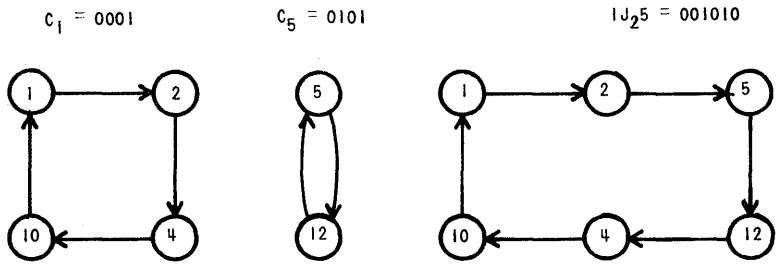


FIG. 17. AN EXAMPLE OF SEQUENCE JOINING.

described above, the sum of their cycle lengths must be equal to or greater than the degree of their states. Moreover, the rule for determining the overlap and the juxtaposing order must be modified to allow the sequence length to be longer than the degree of the states.

Let the sequences to be joined be S_a and S_b of length L_a and L_b respectively; and let s_a be the state on S_a that is adjacent to state s_b on S_b. The rule for determining the overlap and the justaposing order follows:

Let $P_a = L_a$ if $L_a < n$ n = degree of the states

$= n$ if $L_a \geq n$

$P_b = L_b$ if $L_b < n$

$= n$ if $L_b \geq n$

Let state s_c be obtained by proceeding backward $P_a + P_b - (n + 1)$ states on the cycle generated by S_b. Then s_c will have an overlap of $2n - (P_a + P_b)$ digits with s_a. The joined sequence is obtained by cyclically permuting the digits of S_a to obtain s_a on the right and cyclically permuting the digits of S_b to obtain s_c on the left and then juxtaposing the two sequences with S_b on the right with the overlapping digits removed from s_b. The proof of this procedure is identical to that of the joining procedure for cycles of C_n with P_a and

P_b replacing L_1 and L_2, and the restriction that $L_1 + L_2 > n$ replacing the use of Theorem 4 in the proof.

As an example of this procedure, consider the two sequences to be joined, $S_a = 0001$, $S_b = 0111$ with the degree of states equal to three. We notice that state 4 on S_a is adjacent to state 6 on S_b. State 3 is obtained from state 6 by reversing the cycle generated by S_b and proceeding $P_a + P_b - (n + 1) = 2$ states. The joined sequence is 00011101 and the cycle generated by this sequence is shown in Fig. 18.

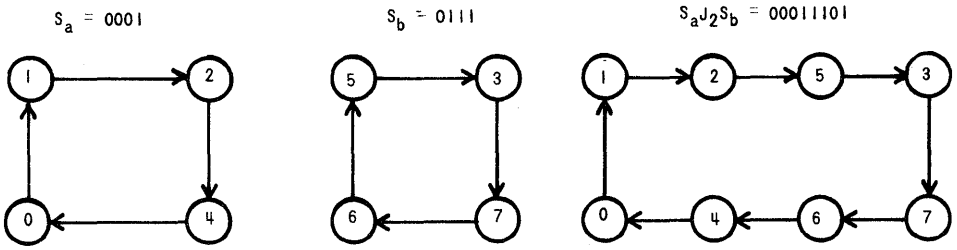
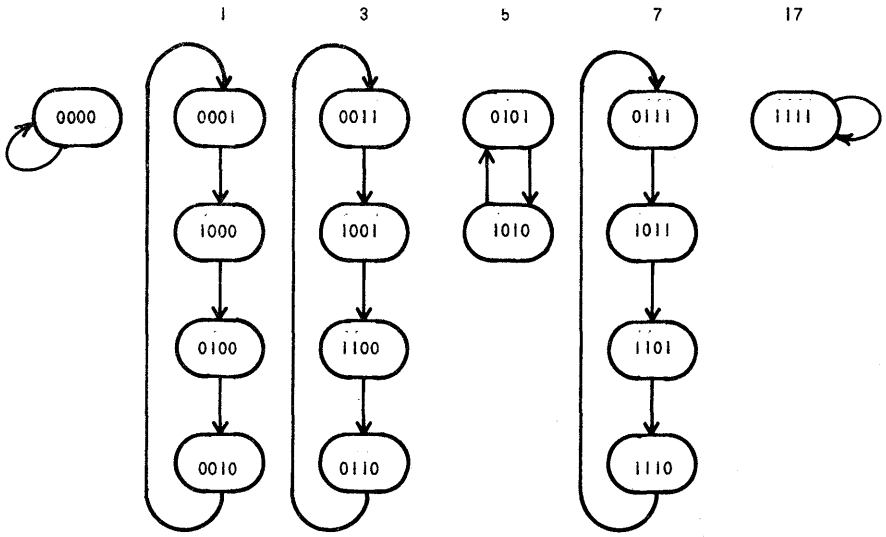


FIG. 18. A MAXIMAL CYCLE PRODUCED BY JOINING CYCLES OF C_3.

E. SOME MAXIMAL SEQUENCES

As another example of the application of the joining procedure, the sequences representing maximal cycles of degree four are obtained. The cycles of C_4 are:



Joining sequences 0 and 1 gives $0J_0 1 = 00001$. Notice that state 10 of the joined sequence is adjacent to state 14 on sequence 3 (see the adjacency diagram A_4 in Fig. 13). Proceeding n - 1 states on the reversed cycle generated by sequence 3 to obtain state 9 and justaposing, $0J_0 1J_4 3 = 000011001$ is obtained.

Next notice that state 14 on the joined sequence is adjacent to state 16 on sequence 7. Again proceeding three states on the reversed cycle generated by sequence 7 to obtain state 15 and juxtaposing to the joined sequence with state 14 on the right, $0J_0 1J_4 3J_6 7 = 001000011011$ is obtained. By using the adjacency of state 13 to state 5 to join sequence 5 to the above sequence and the adjacency of state 16 to 17 to join sequence 4 to the resulting sequence, $0J_0 1J_4 3J_6 7J_7 17 = 0101100100001111$ is obtained. The cycle generated by this sequence is shown in Fig. 19.

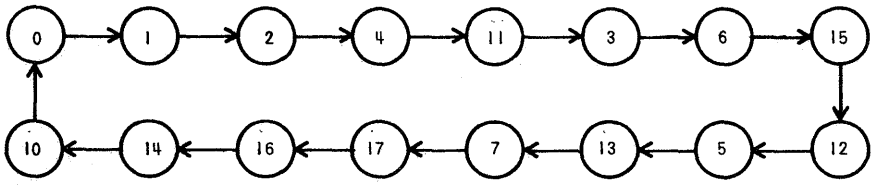


FIG. 19. A MAXIMAL CYCLE PRODUCED BY JOINING CYCLES OF C_4.

By using other adjacencies or joining the sequences in another order, different maximal sequences are produced. Table 2 gives the maximal cycles produced by the other choices of adjacencies and joining order.

TABLE 2. MAXIMAL CYCLES PRODUCED BY JOINING SEQUENCES REPRESENTING CYCLES OF C_4

Joining Operation	Sequence
$0J_0 1J_1 3J_3 7J_2 5J_7 15$	0000101001101111
$0J_0 1J_1 3J_3 7J_5 5J_7 15$	0000100110101111
$0J_0 1J_1 3J_6 7J_2 7J_7 15$	0000101001111011
$0J_0 1J_1 3J_6 7J_5 5J_7 15$	0000100111101011
$0J_0 1J_4 3J_3 7J_2 5J_7 15$	0000110111100101
$0J_0 1J_4 3J_3 7J_5 5J_7 15$	0000110101111001
$0J_0 1J_4 3J_6 7J_2 4J_7 15$	0000111101100101
$0J_0 1J_4 3J_6 7J_5 5J_7 15$	0000111101011001
$0J_0 1J_4 3J_2 5J_5 7J_7 15$	0000110010111101
$0J_0 1J_1 3J_2 5J_5 7J_7 15$	0000101111010011
$0J_0 1J_2 5J_5 7J_3 3J_7 15$	0000101111001101
$0J_0 1J_2 5J_5 7J_6 3J_7 15$	0000101100111101

Only 12 of the 16 possible maximal sequences are obtained in this manner. The remaining maximal sequences are:

0000110100101111
0000111101001011
0000101101001111
0000111100101101

These sequences cannot be expressed in terms of the joining cycles of C_n. They are obtained by joining cycles of C_n and removed cycles that are contained between cycles of C_n.

F. REMOVING A CONTAINED CYCLE

When two cycles (C_a and C_b in Fig. 20) contain a cycle (C_1) between them, they will have two adjacencies connecting them. Let A_1 and A_2 be the sequence defining the adjacencies. Call the first state common to C_a and C_1, s_a; and call the last state common to C_1 and C_a, s'_a. Let q be the number of states on C_1 that are on C_b and let p be the length of C_1. Call the last state that is common to C_b and C_1, s_b; and call the first state common to C_1 and C_b, $s_b^{-(q-1)}$.

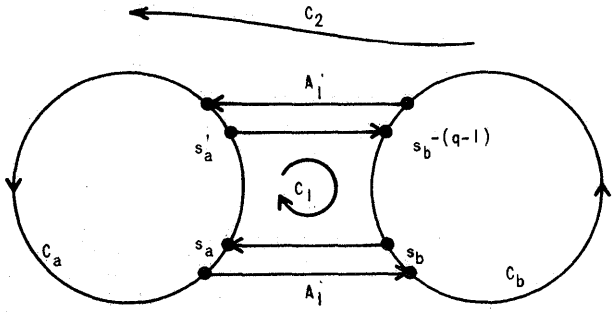


FIG. 20. A REMOVED CYCLE.

Let $s_a = x_1 x_2 \cdots x_n$.

Let C_1 be the cycle whose length is less than n, while C_2 is the cycle whose length is greater than n.

By definition, s_a is periodic with period p, $p < n$,

$$s_a = x_1 x_2 \cdots x_p x_1 \cdots x_{n-p}$$

A different representation of s_a is obtained by proceeding backwards around C_1. First, s_b is obtained from s_a by cyclically permuting the digits of s_a and complementing x_1. Secondly, $s_b^{-(q-1)}$ is obtained from s_b by permuting the digits of s_b cyclically $q - 1$ times.

$$S_b = x_2 \cdots x_p x_2 \cdots x_{n-p}\bar{x}_1$$

$$S_b^{-(q-1)} = x_{q+1} \cdots x_p x_1 \cdots x_{n-p}\bar{x}_1 x_2 \cdots x_q$$

Next, s'_a is obtained by permuting the digits of $s_b^{-(q-1)}$ while complementing x_{q+1}. Finally, s_a is obtained by cyclically permuting the digits of s'_a p-q-2 times.

$$S'_a = x_{q+2} \cdots x_p x_1 \cdots x_{n-p}\bar{x}_1 x_2 \cdots x_q \bar{x}_{q+1}$$

$$S_a = x_1 \cdots x_{n-p}\bar{x}_1 x_2 \cdots x_q \bar{x}_{q+1} \cdots x_p$$

The two adjacency sequences are obtained by observing the (n - 1)-digit words common to S_a and S_b, and to S'_a and $S_b^{-(q-1)}$.

$$A_1 = x_2 \cdots x_p x_1 \cdots x_{n-p}$$

$$A_2 = x_{q+2} \cdots x_p x_1 \cdots x_{n-p}\bar{x}_1 \cdots x_q$$

Since the two representations of S_a must describe the same state, their digits are equated. The first n - p equations give no information, but the last p equations determine the restrictions for S_a to be the first state on the cycle C_1.

$$x_{n-p+1} = \bar{x}_1$$

$$x_{n-p+2} = x_2$$

$$x_{n-p+3} = x_3$$

$$\vdots \qquad \vdots$$

$$x_p = x_{2p-n}$$

$$x_1 = x_{2p-n+1}$$

$$\vdots \quad \vdots$$

$$x_{n-(p+q)+1} = x_{q+1}$$

$$\vdots \quad \vdots$$

$$x_{n-p} = x_p$$

Then,

$x_i = x_j$ if $i = j$ modulo p or if $i = j$ modulo $n - p$ and $j \neq 1$ or $q + 1$

$x_i = \bar{x}_j$ if $j = 1$ or $q + 1$ modulo p or $n - p$

The following examples illustrate the removal process:

Example 1: Let $n = 13$, $p = 9$, and $q = 6$.

$$S_a = x_1x_2x_3x_4x_5x_6x_7x_8x_9x_1x_2x_3x_4$$

$$S_b = x_2x_3x_4x_5x_6x_7x_8x_9x_1x_2x_3x_4\bar{x}_1$$

$$S_b^{-5} = x_7x_8x_9x_1x_2x_3x_4\bar{x}_1x_2x_3x_4x_5x_6$$

$$S'_a = x_8x_9x_1x_2x_3x_4\bar{x}_1x_2x_3x_9x_5x_6x_7$$

$$S_a = x_1x_2x_3x_4\bar{x}_1x_2x_3x_4x_5x_6\bar{x}_7x_8x_9$$

$x_5 = \bar{x}_1$ $\qquad$ $x_1 = x_6$

$x_6 = x_2$ $\qquad$ $x_2 = \bar{x}_7$

$x_7 = x_3$ $\qquad$ $x_3 = x_8$

$x_8 = \bar{x}_4$ $\qquad$ $x_4 = x_9$

$x_9 = x_5$

$x_1 = x_2 = \bar{x}_3 = \bar{x}_4 = \bar{x}_5 = x_6 = \bar{x}_7 = \bar{x}_8 = \bar{x}_9$

$A_1 = x_2x_3x_4x_5x_6x_7x_8x_9x_1x_2x_3x_4$

$A_2 = x_8x_9x_1x_2x_3x_4\bar{x}_1x_2x_3x_4x_5x_6$

If $x_1 = 1$,

$S_a = 1100010001100$

$S_b = 1000100011001$

$A_1 = 100010001100$

$A_2 = 001100010001$

If $x_1 = 0$,

$$S_a = 0011101110011$$

$$S_b = 0111011100110$$

$$A_1 = 011101110011$$

$$A_2 = 1100111011110$$

Example 2: Let $n = 9$, $p = 6$, $q = 3$.

$$S_a = x_1x_2x_3x_4x_5x_6x_1x_2x_3$$

$$S_b = x_2x_3x_4x_5x_6x_1x_2x_3\bar{x}_1$$

$$S_b^{-2} = x_4x_5x_6x_1x_2x_3\bar{x}_1x_2x_3$$

$$S_a^{-8} = x_5x_6x_1x_2x_3\bar{x}_1x_2x_3\bar{x}_4$$

$$S_a = x_1x_2x_3\bar{x}_1x_2x_3\bar{x}_4x_5x_6$$

$$A_1 = x_2x_3x_4x_5x_6x_1x_2x_3$$

$$A_2 = x_5x_6x_1x_2x_3\bar{x}_1x_2x_3$$

$$x_4 = \bar{x}_1 \qquad x_1 = \bar{x}_4$$

$$x_5 = x_2 \qquad x_2 = x_5$$

$$x_6 = x_3 \qquad x_3 = x_6$$

There are three independent choices: x_1, x_2, and x_3; a few are given below:

Let $x_1 = x_2 = x_3 = 0$

$S_a = 00100000$ $\qquad$ $A_1 = 01101001$

$S_b = 00100001$ $\qquad$ $A_2 = 01001101$

Let $x_1 = x_2 = 0,\ x_3 = 1$

$S_a = 001101001$ $\qquad$ $A_1 = 01101001$

$S_b = 011010011$ $\qquad$ $A_2 = 01001101$

Let $x_1 = x_3 = 0,\ x_2 = 1$

$S_a = 010110001$ $\qquad$ $A_1 = 10110001$

$S_b = 101100011$ $\qquad$ $A_2 = 10010110$

Let $x_1 = 1,\ x_2 = 0,\ x_3 = 1$

$S_a = 101001101$ $\qquad$ $A_1 = 01001101$

$S_b = 010011011$ $\qquad$ $A_2 = 01101001$

The above examples illustrate several interesting points. The adjacency sequence A_2 in example 1 can be obtained from A_1 by inverting the order of the digits. This can be done for all the cycles contained between the cycles of C_n for $n \leq 8$, as can be seen by looking at the adjacency diagrams (Fig. 13). The inversion relationship holds whenever $n - p$ and p are relatively prime in the examples with larger n that have been investigated, but a general proof has not been

obtained. The adjacency sequences of the second example are not related in this manner since 3 and 6 are not relatively prime. Notice that all the different combinations of $x_1x_2x_3$ in example 2 do not give different cycles since for $x_1 = x_2 = 0$, $x_3 = 1$ and for $x_1 = x_3 = 1$, $x_2 = 0$, the same adjacencies are used.

When $n - p$ and p are not relatively prime, q divides $n - p$. If q does not divide $n - p$ in example 2, say $q = 4$, then the system of equations becomes:

$$x_4 = x_1 \tag{1}$$

$$x_5 = x_2 \tag{2}$$

$$x_6 = x_3 \tag{3}$$

$$x_1 = x_4 \tag{4}$$

$$x_2 = \bar{x}_5 \tag{5}$$

$$x_3 = x_6 \tag{6}$$

There is a contradiction between Eqs. (1) and (4); and Eqs. (2) and (5). This leads to the following theorem.

Theorem 8: If p and $n - p$ are not relatively prime, then $q = 0$ modulo the greatest common divisor of p and $n - p$ ($p_2n - p$).

Proof: Consider the digit subscripts in the system of equations obtained by the removal process. Two digits are equal if their subscripts are equivalent modulo p or modulo $n - p$. First reduce the integers modulo p and then reduce this set of integers modulo $n - p$.

Let r be the greatest common divisor of p and $n - p$

$$i = j + kp \tag{7}$$

i, j, k are integers

$$i = j + kp + L(n - p) \tag{8}$$

Let $mr = p$ in Eq. (7) and $sr = m - p$ in Eq. (8).

$$i = j + krm + Lrs$$

$$= j + r(km + Ls)$$

then $i = j$ modulo r.

Two digits are in the same equivalence class if their subscripts are equivalent modulo r. Since x_1 and x_{q+1} are the only digits which appear complemented in the system of equations, they must be in the same class or a contradictory set of equations will result. Thus, $1 = q + 1$ modulo r, so $q = 0$ modulo r.

Theorem 8 simplifies the search for all cycles of a given length contained between the cycles of C_n. If $n - p$ and p are relatively prime, q can have any value from 2 to $n - 2$. When $n - p$ and p are relatively prime, $q = 0$ modulo r and therefore $q = kr$, where k is any integer such that $2 \leq kr \leq p - 2$.

G. GENERALITY OF DECOMPOSITION PROCESS

The question now arises of how many of the realizable cycles can be expressed as a function of cycles of C_n in terms of joining removal operations. The following two theorems show that any realizable cycle set can be decomposed in terms of these two operations. If the decomposition process fails at some point, the prescribed cycle set cannot be realized by a minimal length shift register. Thus, the decomposition is a test for realizability, but it may be quite laborious.

Theorem 9: The operations of J and R allow the adjacencies in the adjacency diagram to be used independently.*

*When a cycle is used with two different removal operations, care must be taken to insure that the resulting cycles do not intersect. When a removed cycle is joined to another cycle, the cycles must contain an adjacent state.

Proof: For two cycles connected by only one adjacency, the two cycles can be joined or not, independently of the remaining adjacencies. If the two cycles are connected by two adjacencies, the adjacencies can be used one at a time by joint operations or together by a removal operation. If three cycles are connected by adjacencies as shown in Fig. 21b, they are considered two at a time and the argument for two cycles is used to show that the adjacencies can be used independently. The cycles C_2 and C_3 cannot have adjacent states since they must either have the same measure or their measure must differ by two.

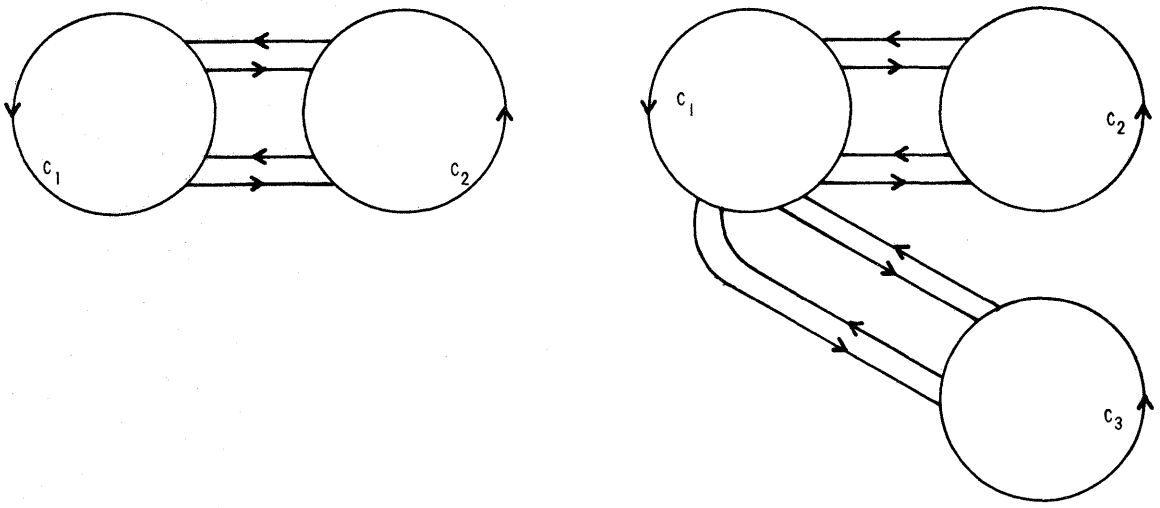


FIG. 21. ADJACENCIES BETWEEN CYCLES OF C_n.

The same argument is applied for an arbitrary number of cycles; first the cycles are considered in pairs and the argument for two cycles is used. If the number of cycles is odd, the remaining three cycles are considered using the argument given above for three cycles. The cycles obtained by operating on pairs of cycles are again considered in pairs and the process repeated until only two cycles remain. Here again the argument for two cycles is applied. Thus for an arbitrary number of cycles the adjacencies may be chosen independently.

Theorem 10: Any cycle set that is realizable in a nonsingular feedback shift register can be expressed in terms of the cycles of C_n and the operations of J and R.

Proof: Theorem 9 shows that all the adjacencies in C_n can be expressed independently as a function of J and R. There is an adjacency for each $(n - 1)$-digit word, hence there are 2^{n-1} different adjacencies. Since these adjacencies can be chosen independently, there are $2^{2^{n-1}}$ different nonsingular shift registers that can be obtained by the different choices of the adjacencies. Golomb and Welsh [Ref. 12] have shown that there are exactly $2^{2^{n-1}}$ different nonsingular shift registers, and hence all the possible different shift registers can be expressed as a function of C_n in terms of J and R.

While Theorem 10 above insures that any realizable cycle set can be decomposed in terms of J and R, it does not specify a procedure. The problem that remains is the development of a systematic method of exhausting the cycles of C_n while obtaining the specified cyclic behavior. This problem is met by constructing a series of decomposition tables.

H. DECOMPOSITION TABLES

Since the cycles to be removed are contained between particular cycles of C_n, they will be removed first and the discarded cycles added to the table of available cycles before the joined cycles are formed. Only the cycles with length $< n$ will be removed first, since the longer cycles are more numerous and may be the discarded cycles after the removal of a shorter cycle. A table of removal possibilities is constructed as follows:

FIRST REMOVAL TABLE

S_1	S_2	S_3	$\ldots S_k \ldots$	S_n	Prescribed Cycle
A_1, A_2		A_1, A_2			C_1
	A_3, A_4			$A_3 A_4$	C_2
A_5, A_6	A_5, A_6				C_3
A_7, A_3		A_7, A_3			
	$A_i\ A_j$		$A_i\ A_j$		C_j
A_1, A_2		A_1, A_2			C_m

1. The S_i are cycles of C_n.
2. The C_i are the prescribed cycle lengths requiring a removal operation.
3. The pairs $(A_i A_j)$ give the adjacencies needed to realize C_j.

Using the table, a nonintersecting set of operations is chosen. The table is first inspected for rows with a single entry. If two or more of these entries are in the same column, then only one of the cycles can be removed from C_n. The remaining cycles are set aside and will enter the table that is constructed for C_{2n}. For the cycles that are removed, the row corresponding to those cycles and the two columns corresponding to the members of C_n used are deleted from the table. The operation which is used is listed for inclusion later in the generating function. This process is repeated until all the columns which have entries are deleted. The discarded cycles are then added to the list of cycles of C_n, with those cycles used in the first removal operation deleted.

SECOND REMOVAL TABLE

S_1	S_2	$\ldots S_k \ldots$	S_n	$D_j \ldots D_k$	D_m	Prescribed Cycle
$A_1 A_2$					$A_1 A_2$	C_1
	$A_3 A_4$			$A_3 A_4$		C_2
$A_i A_j$					$A_i A_j$	C_k
	$A_s A_t$	$A_s A_t$				C_p

If the second removal table, when used in the same manner as the first table, allows a complete realization of the cycles requiring removal, then the removal process terminates, and the remaining behavior is realized by the joining operation. If not, the third removal table is constructed which considers removal from C_{2n}; the fourth table from $C_{4n} \ldots C_{in}$; and the i^{th} table from C_{in}, where i is an integer such that $i \cdot n < 2^n$.

JOINING TABLES

S_1	S_2	S_3	--	D_i	D_u	Prescribed Cycle
A_1		A_1				C_1
	A_2		A_2			C_2
						. . .
A_j				A_j		C_j
						. . .
Ak					Ak	C_n

The remaining cycles are obtained by joining the cycles of C_n that were not used in the removal process and those cycles discarded from the removal process. If the process fails, then the prescribed behavior is

not realizable by a single minimal-length feedback shift register, since it was shown that any cycle in such a register can be expressed in terms of J and R.

I. NETWORK RELATIONS

The cycle set is expressed by the generating function denoted by $G(C_1C_2...C_k)$, where the C_i are the cycles in the prescribed cycle set. The generating function is simply a list of the operations performed in the decomposition process.

$$G(C_1C_2 \ldots C_k) = C_aR_{i,j}C_n; \quad D_dR_{e,m}C_e; \ \ldots \ ; \ C_pJ_kC_q$$

The operations on the cycles of C_n allow the determination of the output sequences which correspond to cycles with the specified length. Having obtained the operations required, the feedback network for the register must be obtained. Two methods are presented; method one uses the output sequences and method two uses the generating functions.

A convenient method for specifying the switching function of a combination network is to specify its characteristic number. The characteristic number $t = t_1t_2...t_{2^n}$ is a sequence of zeros and ones that describe the truth table of the network. Each entry in the characteristic number, t_i, corresponds to the output of the network for the combination of inputs that form the binary number, i. By arranging the input variables in the truth table in the same order as the outputs in the shift register, the subscripts of the characteristic number are made to correspond to the octal representation of the states. Thus the value of t_i is the output desired from the combination network when the i^{th} state is in the register.

a. Method One

The output sequence corresponding to each cycle is found by performing the operations indicated by the generating function. The first n digits of the output sequence determine a state on the cycle, while the last digit is the output of the combinatorial network for that

state. The L digits in the sequence determine L of the 2^n components of the characteristic number of the feedback network. The characteristic number is obtained by letting the first n digits determine the subscript of each component, while the last digit determines the value of that component. By cyclically permuting the digits in the sequence, the L components of the characteristic number are in turn determined.

As an example of this process, the characteristic number τ for the network will be found which produces the maximal cycle generated by the sequence 0101100100001111.

0101100100001111	$t_{12} = 1$
1010110010000111	$t_5 = 1$
1101011001000011	$t_{13} = 1$
1110101100100001	$t_7 = 1$
1111010110010000	$t_{17} = 0$
0111101011001000	$t_{16} = 0$
0011110101100100	$t_{14} = 0$
0001111010110010	$t_{10} = 0$
0000111101011001	$t_0 = 1$
1000011110101100	$t_1 = 0$
0100001111010110	$t_2 = 0$
0010000111101011	$t_4 = 1$
1001000011110101	$t_{11} = 1$
1100100001111010	$t_3 = 0$
0110010000111101	$t_6 = 1$
1011001000011110	$t_{15} = 0$

τ = 1000111101110000

To determine the feedback network, a map of this function is drawn:

x_1x_2 \ x_3x_4	00	10	11	01
00	1	1	0	0
10	0	1	0	1
11	0	1	0	1
01	0	1	0	1

$$f(x_1x_2x_3x_4) = \bar{x}_1\bar{x}_2\bar{x}_4 \vee x_2\bar{x}_3x_4 \vee x_3\bar{x}_4 \vee x_1\bar{x}_3x_4$$

The network is shown in Fig. 22.

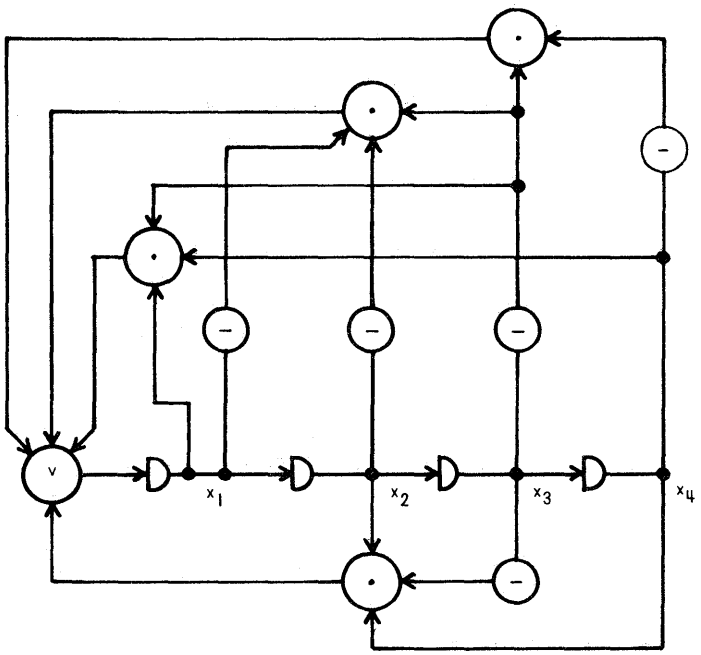


FIG. 22. A SHIFT REGISTER WHICH REALIZES A MAXIMAL CYCLE.

While the above procedure gives a method of obtaining the characteristic number for the feedback network from the sequences themselves, in many cases it will be easier to obtain the feedback network from a knowledge of the sequences that have been joined and the adjacencies that have been used. When the sequences are obtained as roots of polynomials, the joining operations that produce these sequences will not be known,

and the above procedure will be the most convenient method of determining the feedback network.

b. Method Two

The generating function completely describes the behavior of the shift register and contains all the information required to determine the feedback network directly. The generating function of $C_n\ G(C_n) = 0$, indicates that none of the adjacencies are being used. In this case the feedback formula is $f(x_1x_2\ldots x_n) = x_n$. For each adjacency that is used, the feedback formula is modified whenever the $(n - 1)$-digit word that describes the adjacency occurs independently of the n^{th} digit. Thus the feedback formula is

$$f(x_1x_2\ldots x_n) = x_n = x_n + A_1' + A_2' + \ldots + A_k',$$

where A_i' is the Boolean function described by the adjacency sequence A_i. Since the A_i' are independent of x_n the feedback formula can be written

$$f(x_1x_2\ldots x_n) = x_n + g(x_1x_2\ldots x_{n-1})$$

where $g(x_1x_2\ldots x_{n-1}) = A_1' + A_2' + \ldots + A_k'$. This describes a network of the form shown in Fig. 23. Then A_i determines the i^{th} digit in the characteristic number representing $g(x_1x_2\ldots x_{n-1})$. Let $\tau = t_1t_2\ldots t_{2^{n-1}}$ be the characteristic number of $g(x_1x_2\ldots x_{n-1})$; then $t_i = 0$ if A_i is not used in the generating function, and $t_i = 1$ if A_i appears as a subscript of some operation in the generating function. Thus τ can be written by inspection of $G(cs)$.

The decomposition process described above is primarily useful as a design tool, but in some cases it can be useful in the analysis of a register. Given τ, to find the behavior of the register one can write the C matrix and then draw the state graph. In some cases it is easier to analyze the register by observing the generating function that uses the adjacencies specified by τ. This allows the cycle length to be determined without actually drawing the state graph.

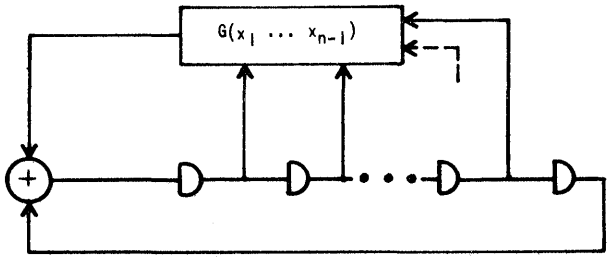


FIG. 23. A NONSINGULAR FEEDBACK SHIFT REGISTER.

The decomposition process and network correspondence are illustrated by the following example:

Let $n = 5$, then there are $2^5 = 32$ states.
Let the cycle set be $10_2\ 4_3$.

$$S_a = x_1x_2x_3x_4x_1$$

$$S_b = x_2x_3x_4x_1\bar{x}_1$$

$$S'_a = x_1\bar{x}_1x_2x_3x_4$$

$q = 1$	$q = 2$	$q = 3$
$x_2 = \bar{x}_1$	$x_2 = \bar{x}_1 = x_3$	$x_2 = x_1 = x_3 = x_4$
$x_3 = \bar{x}_2 = x_4$	$x_3 = x_2$	$x_3 = x_2$
$x_4 = x_3$	$x_4 = x_3$	$x_4 = x_3$
$x_1 = x_4 = x_3$	$x_1 = \bar{x}_4$	$x_1 = \bar{x}_4$

	$x_1 = 0$	$x_1 = 0$	$x_1 = 0$
C_1	$S_a = 01000$	C_3 $S_a = 01100$	$S_a = 01110$
	$S_b = 10001$	$S_b = 11001$	$S_b = 11101$
	$A_1 = 1000$	$A_1 = 1100$	$A_1 = 1110$
	$A_2 = 0001$	$A_2 = 0011$	$A_2 = 0111$

	$x_1 = 1$	$x_1 = 1$	$x_1 = 1$
C_2	$S_a = 10111$	$S_a = 10011$	$S_a = 10001$
	$S_b = 01110$	$S_b = 00110$	$S_b = 00010$
	$A_1 = 0111$	$A_1 = 0011$	$A_1 = 0001$
	$A_2 = 1110$	$A_2 = 1100$	$A_2 = 1000$

FIRST REMOVAL TABLE

1(5)	3(5)	7(5)	15(5)	Prescribed Cycle
(1,8)	(1,8) (3,12)	(7,14) (3,12)	(7,14)	4
(1,8)	(1,8) (3,12)	(7,14) (3,12)	(7,14)	4
(1,8)	(1,8) (3,12)	(7,14) (3,12)	(7,14) (7,14)	4

1) $1R_{(1,8)}{}^{3}$

2) $3R_{(3,12)}{}^{7}$

3) $7R_{(7,14)}{}^{15}$

Note: Since 3 and 7 are used with two different removal operations a check was made to see that the removed cycles do not intersect.

JOINING TABLE

0(1)	5(5)	11(5)	31(1)	D_{135}	Prescribed Cycle
(0,15)	10	10	(0,15)	(0,15)	10
(0,15)	10	10	(0,15)	(0,15)	10

The operation $0J_0D_{135}J_{15}31$, gives one 10 cycle and $5J_{10}11$ gives the other. The generating function is:

$$G(10_2 4_3) = 1R_{(8,1)}3;\quad 3R_{(3,12)}7;\quad 7R_{(7,14)}15;\quad D_{135}J_0 0J_{15}31;\quad 5J_{10}11$$

The feedback function is:

$$g = 1101000110101011$$

The shift register is shown in Fig. 24.

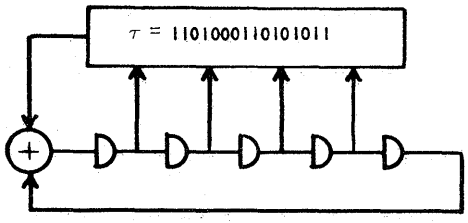


FIG. 24. A FIVE-STAGE SHIFT REGISTER.

As a check of the above design the register is analyzed using the C matrix [Ref. 4] to determine the state graph. The analysis can also be done by inspecting the generating function. The cycles of C_n and the operation used are indicated in the state diagram of Fig. 25.

J. STANDARD REMOVAL TABLES

Once the operation of removing a contained cycle of a given length from the cycles of C_n is completed, the results can be tabulated for future reference. Standard removal tables can be constructed and used directly to find the removal operations required to realize a specified

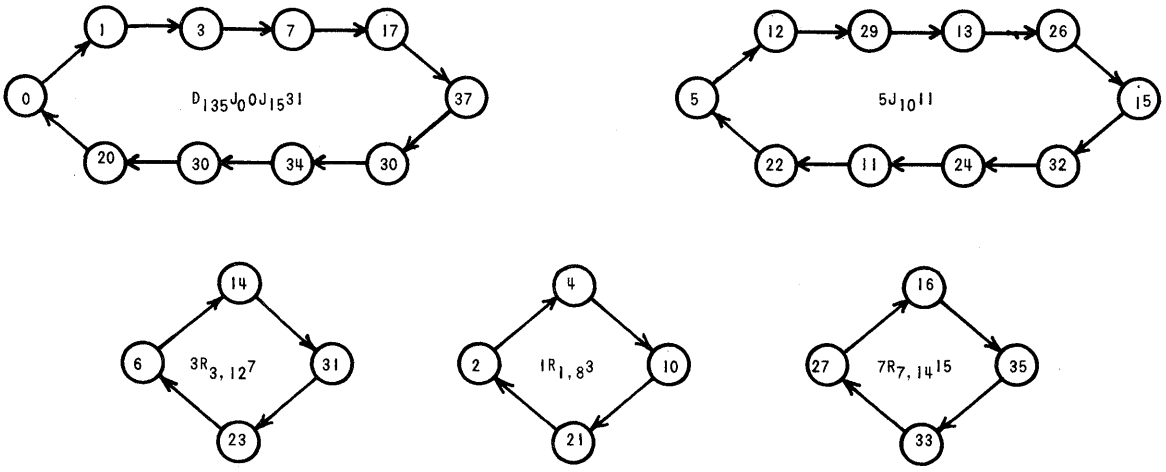


FIG. 25. STATE DIAGRAM FOR EXAMPLE OF DECOMPOSITION PROCESS.

cyclic behavior. For values of $n \leq 8$ the removal tables for C_n are given. These tables allow at least one cycle of arbitrary length to be realized since the operations required to obtain cycles of all lengths less than n are given. One of these short cycles can then be joined to the desired number of n cycles to obtain the prescribed cycle length.

Notice that each contained cycle can be located on the adjacency diagram by observing which cycles of C_n are connected by two adjacency sequences. The ordering of the nodes of any given column in the diagram was done to roughly minimize the average length of the connecting lines. It is interesting to note the effect this policy had on the location of the contained cycles of a given length. Figure 26 gives a skeleton adjacency diagram for $n = 7$, with the length of the contained cycles indicated on the connecting branches.

The location of the longer contained cycles near the top of the diagram occurs in all the other adjacency diagrams given. This fact

allows the operations required to produce these cycles to be determined by inspection of the adjacency diagram.

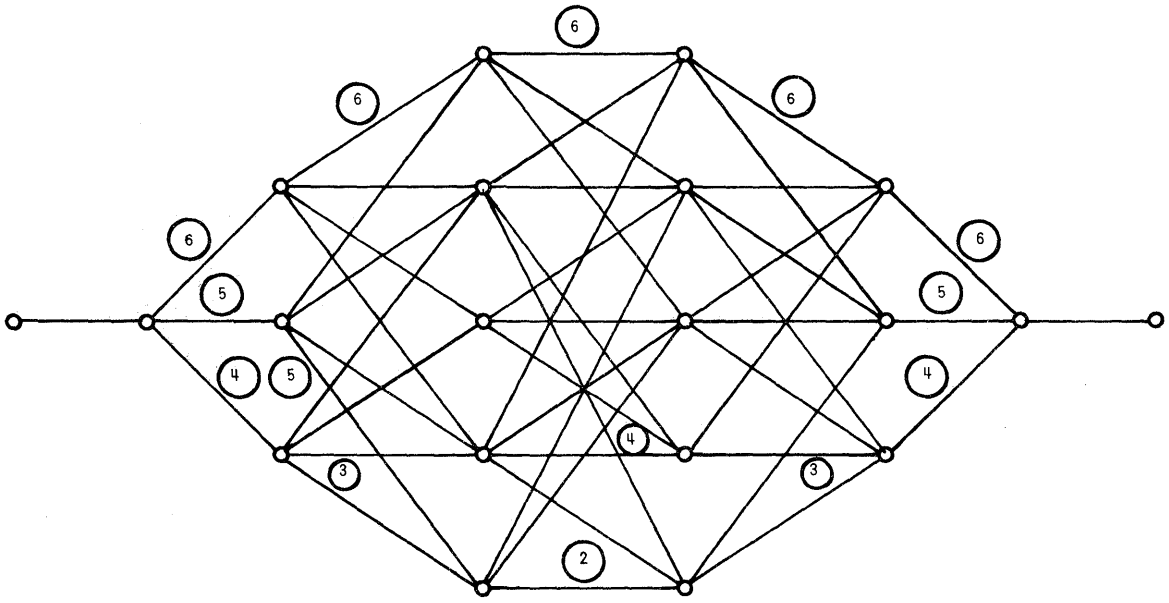


FIG. 26. LOCATION OF CONTAINED CYCLES.

REMOVAL TABLES

n = 3

1	3	GENERATING SEQUENCE DISCARDED CYCLE	CYCLE LENGTH	DISCARDED CYCLE LENGTH
1.2	1.2	3	2	4

n = 4

1	3	7	GENERATING SEQUENCE DISCARDED CYCLE	CYCLE LENGTH	DISCARDED CYCLE LENGTH
1, 4	1, 4		3	3	5
	3, 6	3, 6	7		

n = 5

1	3	5	17	13	17	GENERATING SEQUENCE DISCARDED CYCLE	CYCLE LENGTH	DISCARDED CYCLE LENGTH
		12, 5		12, 5		113	2	8
2, 4*		2, 4				5	3	7
				13, 15	13, 15*	57		
1, 10*	1, 10					3	4	6
			7, 16		7, 16*	27		
	3, 14		3, 14			7		

n = 6

1	3	5	11	7	17	27	37	GENERATING SEQUENCE DISCARDED CYCLE	CYCLE LENGTH	DISCARDED CYCLE LENGTH
10, 2*		10, 2						5	4	8
						27, 35	27, 35*	137		
1, 20*	1, 20							3		
					17, 36		17, 36*	37	5	7
	3, 30			3, 30				7		
				7, 34	7, 34			17		

n = 7

1	3	5	11	7	17	25	37	53	57	67	77	23	33	GENERATING SEQUENCE DISCARDED CYCLE	CYCLE LENGTH	DISCARDED CYCLE LENGTH
						25, 52		25, 52						2453	2	12
			11, 44									11, 44		423	3	11
										66, 33			66, 33	1467		
4, 10*			4, 10											11		
										67, 73	67, 73*			677	4	10
												31, 46	31, 46	233		
								53, 65	53, 65					253		
		12, 24				12, 24								25	5	9
2, 20*		2, 20												5		
									57, 75		57, 75*			277		
40, 1*	40, 1													101		
							37, 76				37, 76*			77		
	3, 60			3, 60										7	6	8
				7, 70	7, 70									17		
					17, 74		17, 74							37		

n = 8

1	3	5	11	7	25	45	17	37	127	133	113	77	137	157	167	177	GENERATING SEQUENCE DISCARDED CYCLE	CYCLE LENGTH	DISCARDED CYCLE LENGTH
										133, 155				133, 155			6557	3	13
			22, 44			22, 44											2045		
						45, 122					45, 122						513	5	11
										55, 132	55, 132						1133		
4, 20*			4, 20														11		
														157, 173		157, 173*	1577		
2, 40*		2, 40															5	6	10
									127, 167				127, 167				537		
		12, 50			12, 50												25		
													137, 175			137,175*	1177		
1, 100*	1, 100																3	7	9
												77, 176				77, 176*	177		
	3, 140			3, 140													7		
								37, 174				37, 174					77		
				7, 160			7, 160										17		
							17, 176	17, 176									37		

* THE CYCLES OBTAINED WITH THAT OPERATION INTERSECT.

V. CONCLUSIONS

A. GENERAL COMMENTS

The usefulness of the nonlinear shift register has been limited because its design was unnecessarily laborious and unsystematic. Even so, it has found wide usage in recent years. The design procedure presented in this report provides an easier and more systematic method of design for this important type of sequential network.

The design of an arbitrary sequential network with inputs and outputs is at present very unsystematic. The linear sequential networks are well understood, but nonlinear networks are not. The author hopes that this work will provide a starting point for the development of a theory that can be applied to a more general class of networks. The behavior of the network without inputs is, in a sense, the natural behavior of the network. As in analog networks, the natural behavior is expected to play an important role in the characterization of the complete behavior.

An analogy with analog networks has been useful in the development of the linear series expansion of a sequence in the polynomial domain. The close resemblance between much of the digital and analog theory has been pointed out in the hope that by continuing the analogy much of the work in analog theory can serve as a guide to develop additional results in digital theory.

The correspondence between the polynomial and sequence domains provides the link necessary to apply the results of the sequence domain work to the polynomial domain. This correspondence allows one to work in the domain which is most convenient and then to translate the results to the domain that is desired.

B. SUMMARY OF RESULTS

The graph shown in Fig. 27 summarizes the results described in this report. The specification of the register can be in terms of the sequences to be produced, the cycle set, or the polynomial that describes

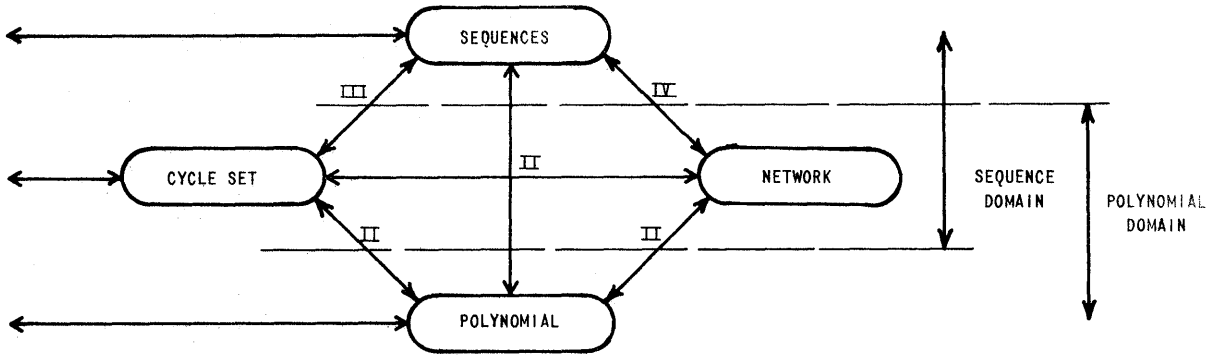


FIG. 27. SUMMARIZING GRAPH.

the sequences to be produced. This is indicated by inputs at these nodes of the graph. Methods of conversion from any one node to another are given in this report. The branches are labeled by the numbers of the chapters containing the corresponding conversion method.

The upper half of the diagram illustrates the sequence-domain synthesis. The study of shift-register sequences provides a method for selecting the specification when there is some freedom. The correspondence between these sequences and the cyclic behavior aids in the specification of the cycle set to be realized. The synthesis procedure allows the determination of the network directly from either specification.

The lower half of the diagram illustrates the polynomial-domain synthesis. The synthesis procedure allows the feedback network to be determined directly from a specification of the polynomials describing the output sequences. In addition, the polynomial gives an algebraic description of the network and its output sequences.

A more systematic method for describing the behavior of a sequential network is needed. Such a method will allow system design to be broken down in smaller blocks and the behavior of the individual blocks specified. The networks considered in this report have no inputs, so the behavior is conveniently specified in terms of a cycle set. The cycle-set specification, however, does not give any information regarding the coding that has been realized. This information is conveniently expressed by the

generating function of the cycle set. An algebraic description is given by the defining polynomial of the register. Each additional class of networks whose behavior can be conveniently specified leads one step closer to the general solution of the specification problem.

C. RECOMMENDATIONS FOR FUTURE STUDY

Some of the results which were obtained in the sequence domain may have important algebraic significance when translated to the polynomial domain. The fact that two cycles of C_n have no more than two adjacencies implies relationships among the roots of their corresponding polynomials. The operation of joining two cycles together implies a corresponding operation with two irreducible polynomials of degree n to produce an irreducible polynomial of degree $2n$. Continued investigation of the correspondence between the two domains may lead to many useful polynomial relationships.

REFERENCES

1. B. Elspas, "A Radar System Based on Statistical Estimation and Resolution Considerations," TR No. 361-1, Stanford Electronics Laboratories, Stanford, Calif., 1 Aug 1955.

2. P. R. Bryant, et al, "Counting with Feedback Shift Registers by Means of a Jump Technique," IRE Trans. (Electronic Computers), EC-11, Apr 1962, pp. 285-286.

3. W. W. Peterson, "Error Correcting Codes," MIT Press and John Wiley and Sons, New York - London, 1961.

4. W. H. Kautz, "State Logic Relations in Autonomous Sequential Networks," Proc. Eastern Joint Comp. Conf., Philadelphia, Pa., Dec 1958.

5. B. Elspas, "The Theory of Autonomous Linear Sequential Networks," IRE Trans. (Circuit Theory), CT-6, 1, 1959, pp. 45-60.

6. S. W. Golomb, "Cycles from Non-Linear Shift Registers," Progress Report No. 20-389, Jet Propulsion Lab., California Institute of Technology, Pasadena, Calif., 31 Aug 1959.

7. C. V. Srinivasan, "State Diagrams of Linear Sequential Machines," J. Franklin Institute, May 1962, pp. 383-418.

8. N. Zierler, "Several Binary Sequence Generators," Tech. Rep. No. 95, Lincoln Lab., Massachusetts Institute of Technology, Lexington, Mass., 12 Sep 1955.

9. P. R. Bryant and R. D. Killick, "Nonlinear Feedback Shift Registers," IRE Trans. (Electronic Computers), EC-11, Jun 1962, pp. 410-412.

10. W. H. Kautz, et al, "Advanced Computer Design Theory," Report No. 3131, Stanford Research Institute, Menlo Park, Calif., Jan 1961.

11. W. H. Kautz, et al, "Advanced Computer Design Theory," Report No. 3638, Stanford Research Institute, Menlo Park, Calif., Nov 1961.

12. S. W. Golomb, "Non-Linear Shift Register Sequences," Memorandum No. 20-149, Jet Propulsion Lab., California Institute of Technology, Pasadena, Calif., Oct 1957.

13. E. J. Good, "Normal Recurring Decimals," J. of the London Math. Soc., 21 (Part 3), 1946, pp. 167-169.

14. R. W. Edwards, "Algebraic Synthesis of Switching Networks," TR No. 2205-1 (SEL-63-029), Stanford Electronics Laboratories, Stanford, Calif., Apr 1963.

15. N. G. DeBruijn, "A Combinatorial Problem," Proc. of Koninkl. Ned. Akad. Wetenschap, 49, Sep 1946, pp. 758-764.

SYSTEMS THEORY
DISTRIBUTION LIST
September 1963

GOVERNMENT

USAELRDL
Ft. Monmouth, N.J.
1 Attn: Dr. H. Jacobs, SIGRA/SL-PF

Procurement Data Division
USAS Equipment Support Agency
Ft. Monmouth, N.J.
1 Attn: Mr. M. Rosenfeld

Commanding General, USAELRDL
Ft. Monmouth, N.J.
5 SIGRA/SL-SC, Bldg. 42
1 TDC, Evans Signal Lab Area

Commanding Officer, ERDL
Ft. Belvoir, Va.
1 Attn: Tech. Doc. Ctr.

Commanding Officer
Frankford Arsenal
Bridge and Tacony St.
Philadelphia 37, Pa.
1 Attn: Library Br., 0270, Bldg. 40

Ballistics Research Lab
Aberdeen Proving Ground, Md.
2 Attn: V. W. Richard, BML

Chief of Naval Research
Navy Dept.
Washington 25, D.C.
2 Attn: Code 427
1 Code 420
1 Code 463

Commanding Officer, USAERU
P.O. Box 205
1 Mt. View, Calif.

Commanding Officer
ONR Branch Office
1000 Geary St.
1 San Francisco 9, Calif.

Commanding Officer
ONR Branch Office
1030 E. Green St.
1 Pasadena, Calif.

Office of Naval Research
Branch Office Chicago
230 N. Michigan Ave.
1 Chicago 1, Ill.

Commanding Officer
ONR Branch Office
207 W. 24th St.
1 New York 11, N.Y.

Office of Naval Research
Navy 100, Box 39
Fleet Post Office
New York, N.Y.
1 Attn: Dr. I. Rowe

New York Naval Shipyard
Material Laboratory Library
Brooklyn, N.Y.
1 Attn: Code 911B, M. Rogofsky
Bldg. 291

Chief Bureau of Ships
Navy Dept.
Washington 25, D.C.
1 Attn: Code 691A1
1 Code 686
1 Code 607 NTDS
1 Code 687D
1 Code 732, A. E. Smith
1 Code 681A

Officer in Charge, ONR
Navy 100, Box 39, Fleet P.O.
16 New York, N.Y.

U.S. Naval Research Lab
Washington 25, D.C.
6 Attn: Code 2000
1 5240
1 5430
1 5200
1 5300
1 5400
1 5266, G. Abraham
1 2027
1 5260
1 6430

Chief, Bureau of Naval Weapons
Navy Dept.
Washington 25, D.C.
1 Attn: RAAV-6
1 RUDC-1
2 RREN-3
1 RAAV-44

Chief of Naval Operations
Navy Dept.
Washington 25, D.C.
1 Attn: Code Op 945Y

Director, Naval Electronics Lab
1 San Diego 52, Calif.

USN Post Graduate School
1 Monterey, Calif.
1 Attn: Tech. Reports Librarian
1 Prof. Gray, Electronics Dept.
1 Dr. H. Titus

Weapons Systems Test Div.
Naval Air Test Center
Patuxent River, Md.
1 Attn: Library

U.S. Naval Weapons Lab
Dahlgren, Va.
1 Attn: Tech. Library

Naval Ordnance Lab
Corona, Calif.
1 Attn: Library
1 H. H. Wieder, 423

Commander, USN Air Dev. Ctr.
Johnsville, Pa.
1 Attn: NADC Library
1 AD-5

Commander
USN Missile Center
Pt. Mugu, Calif.
1 Attn: NO3022

Commanding Officer
U.S. Army Research Office
Box CM, Duke Station
Durham, N.C.
3 Attn: CRD-AA-IP

Commanding General
U.S. Army Materiel Command
Washington 25, D.C.
1 Attn: AMCRD-DE-E
1 AMCRD-RS-PE-E

Department of the Army
Office, Chief of Res. and Dev.
The Pentagon
Washington 25, D.C.
1 Attn: Research Support Div.,
Rm. 3D442

Office of the Chief of Engineers
Dept. of the Army
Washington 25, D.C.
1 Attn: Chief, Library Br.

Office of the Asst. Secy. of Defense
Washington 25, D.C.
1 (AE) Pentagon Bldg., Rm. 3D984

Hqs., USAF(AFRDR-NU.3)
The Pentagon, Washington 25, D.C.
1 Attn: Mr. H. Mulkey, Rm. 4D335

Chief of Staff, USAF
Washington 25, D.C.
2 Attn: AFDRT-ER

Hq., USAF
Dir. of Science and Technology
Electronics Div.
Washington 25, D.C.
1 Attn: AFRST-EL/CS, Maj. E. N. Myers

Aeronautical Systems Div.
Wright-Patterson AFB, Ohio
1 Attn: Lt. Col. L. M. Butsch, Jr.
ASRNE-2
1 ASRNE-2, D. R. Moore
1 ASRNR-32
1 ASRNE-1, Electronic Res. Br.
Elec. Tech. Lab
1 ASRNCF-2, Electromagnetic
and Comm. Lab
3 ASNXR
1 ASNXRR(Library)
6 ASRNE-32

Commandant
AF Institute of Technology
Wright-Patterson AFB, Ohio
1 Attn: AFIT (Library)

Executive Director
AF Office of Scientific Res.
Washington 25, D.C.
1 Attn: SRYA

AF Special Weapons Center
Kirtland AFB, N.M.
2 Attn: SWOI

Director
Air University Library
Maxwell AFB, Ala.
1 Attn: CR-4582

AF Missile Test Center
Patrick AFB, Fla.
1 Attn: AFMTC Tech. Library, MU-135

Commander, AF Cambridge Res. Labs
ARDC, L. G. Hanscom Field
Bedford, Mass.
1 Attn: CRTOTT-2, Electronics

Hqs., AF Systems Command
Andrews AFB
Washington 25, D.C.
1 Attn: SCTAE

Asst. Secy. of Defense (R and D)
R and D Board, Dept. of Defense
Washington 25, D.C.
1 Attn: Tech. Library

Office of Director of Defense
Dept. of Defense
Washington 25, D. C.
1 Attn. Research and Engineering

Institute for Defense Analyses
1666 Connecticut Ave.
Washington 9, D. C.
1 Attn: W. E. Bradley

Defense Communications Agency
Dept. of Defense
Washington 25, D. C.
1 Attn: Code 121A, Tech. Library

Advisory Group on Electron Devices
346 Broadway, 8th Floor East
New York 13, N. Y.
2 Attn: H. Sullivan

Advisory Group on Reliability of
Electronic Equipment
Office Asst. Secy. of Defense
The Pentagon
1 Washington 25, D.C.

Commanding Officer
Diamond Ordnance Fuze Labs
Washington 25, D. C.
2 Attn: ORDTL 930, Dr. R. T. Young

Diamond Ordnance Fuze Lab.
U.S. Ordnance Corps
Washington 25, D. C.
1 Attn: ORDTL-450-638,
Mr. R. H. Comyn

U.S. Dept. of Commerce
National Bureau of Standards
Boulder Labs
Central Radio Propagation Lab.
1 Boulder, Colorado
2 Attn: Miss J. V. Lincoln, Chief
RWSS

NSF, Engineering Section
1 Washington, D.C.

Information Retrieval Section
Federal Aviation Agency
Washington, D. C.
1 Attn: MS-112, Library Branch

DDC
Cameron Station
Alexandria 4, Va.
30 Attn: TISIA

U.S. Coast Guard
1300 E. Street, N.W.
Washington 25, D. C.
1 Attn: EEE Station 5-5

Office of Technical Services
Dept. of Commerce
1 Washington 25, D.C.

Director
National Security Agency
Fort George G. Meade, Md.
1 Attn: R42

NASA, Goddard Space Flight Center
Greenbelt, Md.
1 Attn: Code 611, Dr. G. H. Ludwig
1 Chief, Data Systems Divisions

Chief, U.S. Army Security Agency
Arlington Hall Station
2 Arlington 12, Virginia

*No AF or Classified Reports.

SCHOOLS

*U of Aberdeen
Dept. of Natural Philosophy
Marischal College
Aberdeen, Scotlant
1 Attn: Mr. R. V. Jones

U of Arizona
EE Dept.
Tucson, Ariz.
1 Attn: R. L. Walker
1 D. J. Hamilton

*U of British Columbia
Vancouver 8, Canada
1 Attn: Dr. A. C. Soudack

California Institute of Technology
Pasadena, Calif.
1 Attn: Prof. R. W. Gould
1 Prof. L. M. Field, EE Dept.
1 D. Braverman, EE Dept.

California Institute of Technology
4800 Oak Grove Drive
Pasadena 3, Calif.
1 Attn: Library, Jet Propulsion Lab.

U. of California
Berkeley 4, Calif.
1 Attn: Prof. R. M. Saunders, EE Dept.
1 Dr. R. K. Wakerling,
Radiation Lab. Info. Div.,
Bldg. 30, Rm. 101

U of California
Los Angeles 24, Calif.
1 Attn: C. T. Leondes, Prof. of
Engineering, Engineering
Department
1 R. S. Elliott,
Electromagnetics Div., College
of Engineering

U of California, San Diego
School of Science and Engineering
La Jolla, Calif.
1 Attn: Physics Dept.

Carnegie Institute of Technology
Schenley Park
Pittsburg 13, Pa.
1 Attn: Dr. E. M. Williams, EE Dept.

Case Institute of Technology
Engineering Design Center
Cleveland 6, Ohio
1 Attn: Dr. J. B. Neswick, Director

Cornell U
Cognitive Systems Research Program
Ithaca, N. Y.
1 Attn: F. Rosenblatt, Hollister Hall

Drexel Institute of Technology
Philadelphia 4, Pa.
1 Attn: F. B. Haynes, EE Dept.

U of Florida
Engineering Bldg., Rm. 336
Gainsville, Fla.
1 Attn: M. J. Wiggins, EE Dept.

Georgia Institute of Technology
Atlanta 13, Ga.
1 Attn: Mrs. J. H. Crosland
Librarian
1 F. Dixon, Engr. Experiment
Station

Harvard U
Pierce Hall
Cambridge 38, Mass.
1 Attn: Dean H. Brooks, Div of Engr.
and Applied Physics, Rm. 217
2 E. Farkas, Librarian, Rm.
303A, Tech. Reports
Collection

U of Hawaii
Honolulu 14, Hawaii
1 Attn: Asst. Prof. K. Najita,
EE Dept.

Illinois Institute of Technology
Technology Center
Chicago 16, Ill.
1 Attn: Dr. P. C. Yuen, EE Dept.

U of Illinois
Urbana, Ill.
1 Attn: P. D. Coleman, EE Res. Lab.
1 W. Perkins, EE Res. Lab.
1 A. Albert, Tech. Ed., EE
Res. Lab.
1 Library Serials Dept.
1 Prof. D. Alpert, Coordinated
Sci. Lab.

*Instituto de Pesquisas da Marinha
Ministerio da Marinha
Rio de Janeiro
Estado da Guanabara, Brazil
1 Attn: Roberto B. da Costa

Johns Hopkins U
Charles and 34th St.
Baltimore 18, Md.
1 Attn: Librarian, Carlyle Barton Lab.

Johns Hopkins U
8621 Georgia Ave.
Silver Spring, Md.
1 Attn: N. H. Choksy
1 Mr. A. W. Nagy, Applied
Physics Lab.

Linfield Research Institute
McMinnville, Ore.
1 Attn: G. N. Hickok, Director

Marquette University
College of Engineering
1515 W. Wisconsin Ave.
Milwaukee 3, Wis.
1 Attn: A. C. Moeller, EE Dept.

M I T
Cambridge 39, Mass.
1 Attn: Res. Lab. of Elec., Doc.
Rm. 26-327
1 Miss A. Sils, Libn. Rm 4-244,
LIR
1 Mr. J. E. Ward, Elec. Sys.
Lab.

M I T
Lincoln Laboratory
P.O. Box 73
1 Attn: Lexington 73, Mass.
1 Navy Representative
1 Dr. W. I. Wells

U of Michigan
Ann Arbor, Mich.
1 Attn: Dir., Cooley Elec. Labs.,
N. Campus
1 Dr. J. E. Rowe, Elec. Phys.
Lab.
1 Comm. Sci. Lab., 180 Frieze
Bldg.

U of Michigan
Institute of Science and Technology
P.O. Box 618
Ann Arbor, Mich.
1 Attn: Tech. Documents Service
1 W. Wolfe--IRIA--

U of Minnesota
Institute of Technology
Minneapolis 14, Minn.
1 Attn: Prof. A. Van der Ziel, EE Dept.

U. of Nevada
College of Engineering
Reno, Nev.
1 Attn: Dr. R. A. Manhart, EE Dept.

Northeastern U
The Dodge Library
Boston 15, Mass.
1 Attn: Joyce E. Lunde, Librarian

Northwestern U
2422 Oakton St.
Evanston, Ill.
1 Attn: W. S. Toth, Aerial Measurements Lab.

U of Notre Dame
South Bend, Ind.
1 Attn: E. Henry, EE Dept.

Ohio State U
2024 Niel Ave.
Columbus 10, Ohio
1 Attn: Prof. E. M. Boone, EE Dept.

Oregon State U
Corvallis, Ore.
1 Attn: H. J. Oorthuys, EE Dept.

Polytechnic Institute
333 Jay St.
Brooklyn, N. Y.
1 Attn: L. Shaw, EE Dept.

Polytechnic Institute of Brooklyn
Graduate Center, Route 110
Farmingdale, N. Y.
1 Attn: Librarian

Purdue U
Lafayette, Ind.
1 Attn: Library, EE Dept.

Rensselaer Polytechnic Institute
Troy, N. Y.
1 Attn: Library, Serials Dept.

*U of Saskatchewan
College of Engineering
Saskatoon, Canada
1 Attn: Prof. R. E. Ludwig

Stanford Research Institute
Menlo Park, Calif.
1 Attn: External Reports, G-037

Syracuse U
Syracuse 10, N. Y.
1 Attn: EE Dept.

*Uppsale U
Institute of Physics
Uppsala, Sweden
1 Attn: Dr. P. A. Tove

U of Utah
Salt Lake City, Utah
1 Attn: R. W. Grow, EE Dept.

U of Virginia
Charlottesville, Va.
1 Attn: J. C. Wyllie, Alderman Library

U of Washington
Seattle 5, Wash.
1 Attn: A. E. Harrison, EE Dept.

Worchester Polytechnic Inst.
Worchester, Mass.
1 Attn: Dr. H. H. Newell

Yale U
New Haven, Conn.
1 Attn: Sloane Physics Lab.
1 EE Dept.
1 Dunham Lab., Engr. Library

INDUSTRIES

Argonne National Lab.
9700 South Cass
Argonne, Ill.
1 Attn: Dr. O. C. Simpson

Admiral Corp.
3800 Cortland St.
Chicago 47, Ill.
1 Attn: E. N. Roberson, Librarian

Airborne Instruments Lab.
Comac Road
Deer Park, Long Island, N. Y.
1 Attn: J. Dyer, Vice-Pres. and Tech. Dir.

Amperex Corp.
230 Duffy Ave.
Hicksville, Long Island, N. Y.
1 Attn: Proj. Enginneer, S. Barbasso

Autonetics
Div. of North American Aviation, Inc
9150 E. Imperial Highway
Downey, Calif.
1 Attn: Tech. Library 3040-3

Bell Telephone Labs.
Murray Hill Lab.
Murray Hill, N. J.
1 Attn: Dr. J. R. Pierce
1 Dr. S. Darlington
1 Mr. A. J. Grossman

Bell Telephone Labs., Inc.
Technical Information Library
Whippany, N. J.
1 Attn: Tech. Repts. Librn., Whippany Lab.

*Central Electronics Engineering
Research Institute
Pilani, Rajasthan, India
1 Attn: Om P. Gandhi - Via: ONR/London

Columbia Radiation Lab.
538 West 120th St.
1 New York, New York

Convair - San Diego
Div. of General Dynamics Corp.
San Diego 12, Calif.
1 Attn: Engineering Library

Cook Research Labs.
6401 W. Oakton St.
1 Attn: Morton Grove, Ill.

Cornell Aeronautical Labs., Inc.
4455 Genesse St.
Buffalo 21, N. Y.
1 Attn: Library

Eitel-McCullough, Inc.
301 Industrial Way
San Carlos, Calif.
1 Attn: Research Librarian

Ewan Knight Corp.
East Natick, Mass.
1 Attn: Library

Fairchild Semiconductor Corp.
4001 Junipero Serra Blvd.
Palo Alto, Calif.
1 Attn: Dr. V. H. Grinich

General Electric Co.
Defense Electronics Div., LMED
Cornell University, Ithaca, N. Y.
1 Attn: Library - Via: Commander, ASD W-P AFB, Ohio, ASRNGW D.E. Lewis

General Electric TWT Products Sec.
601 California Ave.
Palo Alto, Calif.
1 Attn: Tech. Library, C. G. Lob

General Electric Co. Res. Lab
P.O. Box 1088
Schenectady, N.Y.
1 Attn: Dr. P. M. Lewis
1 R. L. Shuey, Mgr. Info. Studies Sec.

General Electric Co.
Electronics Park
Bldg. 3, Rm. 143-1
Syracuse, N.Y.
1 Attn: Doc. Library, Y. Burke

Gilfillan Brothers
1815 Venice Blvd.
Los Angeles, Calif.
1 Attn: Engr. Library

The Hallicrafters Co.
5th and Kostner Ave.
1 Attn: Chicago 24, Ill.

Hewlett-Packard Co.
1501 Page Mill Road
1 Attn: Palo Alto, Calif.

Hughes Aircraft
Malibu Beach, Calif.
1 Attn: Mr. Iams

Hughes Aircraft Ct.
Florence at Teale St.
Culver City, Calif.
1 Attn: Tech. Doc. Cen., Bldg. 6, Rm. C2048

Hughes Aircraft Co.
P.O. Box 278
Newport Beach, Calif.
1 Attn: Library, Semiconductor Div.

IBM, Box 390, Boardman Road
Poughkeepsie, N. Y.
1 Attn: J. C. Logue, Data Systems Div.

IBM, Poughkeepsie, N. Y.
1 Attn: Product Dev. Lab., E. M. Davis

*No AF or Classified Reports.

IBM ASD and Research Library
Monterey and Cottle Roads
San Jose, Calif.
1 Attn: Miss M. Griffin, Bldg. 025

ITT Federal Labs.
500 Washington Ave.
Nutley 10, N. J.
1 Attn: Mr. E. Mount, Librarian

Laboratory for Electronics, Inc.
1075 Commonwealth Ave.
Boston 15, Mass.
1 Attn: Library

LEL, Inc.
75 Akron St.
Copiague, Long Island, N. Y.
1 Attn: Mr. R. S. Mautner

Lenkurt Electric Co.
San Carlos, Calif.
1 Attn: M. L. Waller, Librarian

Librascope
Div. of General Precision, Inc.
808 Western Ave.
Glendale 1, Calif.
1 Attn: Engr. Library

Lockheed Missiles and Space Div.
P.O. Box 504, Bldg. 524
Sunnyvale, Calif.
1 Attn: Dr. W. M Harris, Dept. 67-30
1 G. W. Price, Dept. 67-33

Melpar, Inc.
3000 Arlington Blvd.
Falls Church, Va.
1 Attn: Librarian

Microwave Associates, Inc.
Northwest Industrial Park
Burlington, Mass.
1 Attn: K. Mortenson
1 Librarian

Microwave Electronics Corp.
4061 Transport St.
Palo Alto, Calif.
1 Attn: S. F. Kaisel
1 M. C. Long

Minneapolis-Honeywell Regulator Co.
1177 Blue Heron Blvd.
Riviera Beach, Fla.
1 Attn: Semiconductor Products Library

Monsanto Research Corp.
Station B, Box 8
Dayton 7, Ohio
1 Attn: Mrs. D Crabtree

Monsanto Chemical Co.
800 N. Linbergh Blvd.
St. Louis 66, Mo
1 Attn: Mr. E. Orban, Mgr. Inorganic Dev.

*Dir., National Physical Lab.
Hilside Road
New Delhi 12, India
1 Attn: S.C. Sharma - Via: ONR/London

*Northern Electric Co., Lmtd.
Research and Development Labs.
P.O. Box 3511, Station "C"
Ottawa, Ontario, Canada
1 Attn: J. F. Tatlock
Via: ASD, Foreign Release Office
W-P AFB, Ohio
Mr. J. Troyan (ASYF)

Nortronics
Palo Verdes Research Park
6101 Crest Road
Palos Verdes Estates, Calif.
1 Attn: Tech. Info. Center

Pacific Semiconductors, Inc.
14520 So. Aviation Blvd.
Lawndale, Calif.
1 Attn: H. Q. North

Philco Corp.
Tech. Rep. Division
P.O. Box 4730
Philadelphia 34, Pa.
1 Attn: F. R. Sherman, Mgr. Editor

Philco Corp.
Jolly and Union Meeting Roads
Blue Bell, Pa.
1 Attn: C. T. McCoy
1 Dr. J. R. Feldmeier

Polarad Electronics Corp.
43-20 Thirty-Fourth St.
Long Island City 1, N. Y.
1 Attn: A. H. Sonnenschein

Radio Corp. of America
RCA Labs., David Sarnoff Res. Cen.
Princeton, N. J.
2 Attn: Dr. J. Sklansky

RCA Labs., Princeton, N. J.
1 Attn: H. Johnson

RCA, Missile Elec. and Controls Dept.
Woburn, Mass.
1 Attn: Library

The Rand Corp.
1700 Main St.
Santa Monica, Calif.
1 Attn: Helen J. Waldron, Librarian

Raytheon Manufacturing Co.
Microwave and Power Tube Div.
Burlington, Mass.
1 Attn: Librarian, Spencer Lab.

Raytheon Manufacturing Co.
Res. Div., 28 Seyon St.
Waltham, Mass.
1 Attn: Dr. H. Statz
1 Mrs. M. Bennett, Librarian

Roger White Electron Devices, Inc.
Tall Oaks Road
1 Laurel Hedges, Stamford, Conn.

Sandia Corp.
Sandia Base, Albuquerque, N. M.
1 Attn: Mrs. B. R. Allen, Librarian

Sperry Rand Corp.
Sperry Electron Tube Div.
1 Gainesville, Fla.

Sperry Gyroscope Co.
Div. of Sperry Rand Corp.
Great Neck, N.Y.
1 Attn: L. Swern(MS3T105)

Sperry Gyroscope Co.
Engineering Library
Mail Station F-7
Great Neck, Long Island, N. Y.
1 Attn: K. Barney, Engr. Dept. Head

Sperry Microwave Electronics
Clearwater, Fla.
1 Attn: J. E. Pippin, Res. Sec. Head

Sylvania Electric Products, Inc.
208-20 Willets Point Blvd.
Bayside, Long Island, N. Y.
1 Attn: L. R. Bloom, Physics Lab.

Sylvania Electronics Systems
100 First Ave.
Waltham 54, Mass.
1 Attn: Librarian, Waltham Labs.
1 Mr. E. E. Hollis

Technical Research Group
1 Syosett, L.I., N.Y.

Texas Instruments, Inc.
Semiconductor-Components Div.
P.O. Box 205
Dallas 22, Tex.
1 Attn: Library
2 Dr. W. Adcock

Texas Instruments, Inc.
1300 N. Central Expressway
1 Dallas, Texas

Texas Instruments, Inc.
P.O. Box 6015
Dallas 22, Tex.
1 Attn: M. E. Caun, Apparatus Div.

Texas Instruments
6017 E. Calle Tuberia
Phoenix, Arizona
1 Attn: R. L. Pritchard

Texas Instruments, Inc.
Corporate Research and Engineering
Technical Reports Service
P.O. Box 5474
1 Attn: Dallas 22, Tex.

Textronix, Inc.
P.O. Box 500
Beaverton, Ore.
4 Attn: Dr. J. F. DeLord, Dir. of Research

Varian Associates
611 Hansen Way
Palo Alto, Calif.
1 Attn: Tech. Library

Weitermann Electronics
4549 North 38th St.
1 Milwaukee 9, Wisconsin

Westinghouse Electric Corp.
Friendship International Airport
Box 746, Baltimore 3, Md.
1 Attn: G. R. Kilgore, Mgr. Appl. Res. Dept. Baltimore Lab.

Westinghouse Electric Corp.
3 Gateway Center
Pittsburgh 22, Pa.
1 Attn: Dr. G. C. Sziklai

Westinghouse Electric Corp.
P.O. Box 284
Elmira, N. Y.
1 Attn: S. S. King

Zenith Radio Corp.
6001 Dickens Ave.
Chicago 39, Ill.
1 Attn: J. Markin

*No AF or Classified Reports.