JKAU: Eng. Sci., **Vol. 18** No. 2, pp: 43- 72 (2007 A.D./1428 A.H.)

On the Linear Analysis of Synchronous Switching Networks

Ali Muhammad Ali Rushdi and Sultan Obayyed Saad Al-Otaibi

Department of Electrical and Computer Engineering,
King Abdulaziz University
P.O. Box 80204, Jeddah 21589, Saudi Arabia
arushdi@kau.edu.sa & *sootaibi@hotmail.com*

Abstract. An exposition of an earlier seminal paper on the linear analysis of synchronous switching networks is presented. This analysis, based on the use of the finite or Galois field GF(2), resembles the linear analysis of continuous systems and has important applications in genetics and biochemistry. A synchronous switching network is represented by a function matrix or by a transition matrix, which are related by a similarity transformation in terms of a state matrix. Our use of a novel recursive ordering for the keys or indices of these matrices reveals several new and interesting features and properties. The state matrix is observed to depend not on the particular network but merely on its number of nodes, and is further given a novel interpretation via the modern concept of subsumption of a logical product by another. This reveals a recursive structure of the state matrix and leads to a proof that it is involutory (self-inverse). The autonomous behavior of synchronous switching networks is studied via the characteristic equations and eigenvectors of the aforementioned matrices. In general, the classical ideas are enriched with modern concepts and terminology, supported with correct proofs, and clarified with detailed tutorial examples.

1. Introduction

Two of the most complex biological systems of current interest are the cellular genetic system and the intracellular biochemical signal transduction networks of the central nervous system. These systems are too complex to allow exact or nearly exact analysis. Kauffman [1] and McCulloch and Pitts [2] proposed studying these respective systems using a synchronous switching network as an approximating model. In, fact a synchronous switching network is the simplest possible conceptual model that mimics or captures the essential features of the original systems.

A synchronous switching network is composed of a set of elements, corresponding to either genes or neurons, which have two states, on (1) and off (0). Time is assumed to be synchronized, *i.e.,* quantized in terms of a basic delay, corresponding to either the transport delay or a synaptic delay. At each instant of time, each element is in one of its two states, and the state of the element in the next instant of time is a function of the present states of the elements and the states of the set of external inputs.

In a seminal paper, Cull [3], presented a study of synchronous switching networks using the mathematical concept of the finite or Galois field GF(2) [4,5]. He preferred the finite-field approach to the switching-algebra or Boolean-algebra approach [6,7], so as to emphasize the strong analogy between the linear analysis of (discrete) synchronous switching networks and that of continuous systems [8]. Unfortunately, this seminal work of Cull [3] has gone almost unnoticed in the literature, and has received only few citations by authors who apparently were unable to make any real use of it [9,10]. This is due in part to the fact that it was written in a very obscure and outdated language, and in part to the fact that it appeared in a little known periodical that publishes essentially in German and not in English. With a revival of work on synchronous Boolean networks and their applications to biological and biochemical systems [9-11], the need for accessing, comprehending, updating, and extending Cull's work is much felt. Therefore, we find it worthwhile to present this paper in which we make a detailed exposition of the ideas of Cull [3], extend them with modern ideas and terminology, support them with correct proofs, supplement them with clarifying specific examples, and, wherever possible, relate them to the concepts of switching or Boolean algebra. As an offshoot of this endeavor, we propose herein a new ordering of the pertinent variables and their products, which serve as keys or indices of the system matrices. This new ordering is superior to the one adopted by Cull [3], as it enjoys an elegant recursive property and leads to several useful interpretations and features.

The rest of this paper is organized as follows. Section 2 reviews the properties of functions over the simplest finite or Galois field GF(2), while Section 3 introduces the concepts of function matrix and transition matrix. These two matrices are related by a similarity transformation, and each of them is a sufficient representation of the synchronous switching network. Our new ordering for the keys or indices of these matrices reveals several novel and interesting features and properties. We observe that the state or modal matrix, used in the transformation between the network-dependent function and transition matrices, is network-independent and depends only on the number of nodes. Hence, we present a novel interpretation of the state matrix via the modern concept of subsumption of a logical product by another. This reveals a recursive structure of the state matrix and leads to a proof that it is involutory (self-inverse). The state matrix is also shown to serve as the transformation matrix between two distinct vector representations of the network instantaneous state. The autonomous behavior of synchronous switching networks is obtained from a consideration of the characteristic equation (discussed in Sec. 4) and eigenvectors (discussed in Sec. 5) of either of the two matrices. Section 6 is devoted to some concluding remarks. This organization of our current paper parallels that of the original paper of Cull [3]. However, we give due credit to Cull by citing his work repeatedly, and we take care to distinguish our contributions from his. A part of the work in Ref. [3] that is not covered herein deals with a criterion for invertibility of systems of switching equations. More general methods for solving switching and Boolean equations are given in Ref. [12-14].

2. Functions Over the Finite or Galois Field GF(2)

Let F be a set of objects on which two operations: addition (+) and multiplication (represented by (*) or juxtapositioning) are defined. F is said to be a field if and only if

1. F forms a commutative group under addition. The additive identity element is labeled "0".

2. F-{0} (the set F with the additive identity removed) forms a commutative group under multiplication. The multiplicative identity element is labeled "1".

3. Multiplication is distributed by addition: a * (b + c) = (a * b) + (a * c).

A field can also be defined as a commutative ring with identity in which every element has a multiplicative inverse [15]. Well known examples of finite fields include the real field (field of real numbers) and the complex field (field of complex numbers).

If a field has a finite order (number of elements) q, then its order completely specifies it, and it is called the Galois field GF(q). Thus two finite fields of the same size are isomorphic, *i.e.*, they are always identical up to the labeling of their elements, regardless of how the fields are constructed. In GF(q), there are q^{q^n} functions or polynomials of n variables. A finite field GF(q) exists iff $q = p^r$, where p is a prime integer and r is a positive integer. Finite fields are of central importance since any function in n variables over $GF(p^r)$ can be represented by r functions in nr variables over GF(p). Cull [3] restricted his work to GF(2), but he claimed that his results can carry over to any GF(p) and thus to any $GF(p^r)$.

In direct analogy with the use of the real or complex fields in describing and analyzing continuous systems, Cull [3] uses the simplest finite field GF(2) in describing and analyzing systems made up of switching elements. The structure of GF(2), also called the binary field or the mod 2 field, is remarkably simple. The field has only two elements: the additive identity (0) and the multiplicative identity (1). The field operations are defined by the following axioms:

$$0 + 0 = 1 + 1 = 0$$

$$1 + 0 = 0 + 1 = 1$$

$$0*0 = 0*1 = 1*0 = 0$$

$$1*1 = 1$$

Note that the addition (+) operation is a modulo 2 operation that resembles the exclusive-OR operation ($\oplus$) in switching algebra. Any function of n variables over GF(2) is a polynomial of 2^n terms (called a Taylor or a Reed-Muller polynomial [4,5,16]), and hence can be represented by a vector of length 2^n, whose elements are the binary coefficients of the 2^n terms in the polynomial. Note that the 2^n terms are the constant term (1), n terms representing each of the n variables, $\binom{n}{2}$ terms representing products of variables taken 2 at a time, $\binom{n}{3}$ terms representing products of variables taken 3 at a time, and so on, ending with a single term for the product of all variables. Therefore, the total number of terms is expressed by the binomial identity

$$2^{n}=\sum_{k=0}^{n}\binom{n}{k} \tag{1}$$

A vector of these 2^n terms, arranged in some agreed-upon order, constitutes a basis for all functions of n variables over GF(2), in the sense that any of these functions equals the (scalar) dot product of the vector of binary coefficients representing it with the aforementioned basis vector. Cull [3] proposed the use of a lexicographic order for the basis vector of the form

$[1\ \ x_1\ \ x_2\ \ x_3\ \ldots\ x_n\ \ x_1x_2\ \ x_1x_3\ \ldots\ x_1x_n\ \ x_2x_3\ \ldots\ x_{n-1}x_n\ \ x_1x_2x_3\ \ldots\ x_{n-2}x_{n-1}x_n\ \ldots\ x_1x_2x_3\ldots x_n]$.

In fact, he replaced the leading 1 with a 0; an oversight since 0 is not a product at all. We propose, however an order of the form

$$\mathbf{B}_n = [\ 1\ \ x_1\ \ x_2\ \ x_1\,x_2\ \ x_3\ \ x_1x_3\ \ x_2x_3\ \ x_1x_2x_3\ \ldots\ x_1x_2x_3\ \ldots\ x_n]^T \tag{2}$$

We call the above order of the of the basis vector a recursive order, since our column vector $\mathbf{B}_n$ can be defined by the recursive relation

$$\mathbf{B}_n=\begin{bmatrix} \mathbf{B}_{n-1} \\ \mathbf{B}_{n-1}\quad x_n \end{bmatrix} \tag{3}$$

together with the boundary condition

$$\mathbf{B}_0= [1] \tag{4}$$

A restatement of the recursive relation (3) is that none of the logical products (that are elements of $\mathbf{B}_n$) is succeeded by products that are subsumed by it. We recall that a product subsumes another if the set of literals of the former is a superset of that of the latter [6].

For example, the function of two variables $x_1+x_1x_2$ is represented by the vector [0 1 0 1]. Since the function has no constant term, a zero appears as the first component of this vector. The second component of the vector is 1 since the coefficient of x_1 is 1. The third component is 0 since the coefficient of x_2 is 0, and the fourth component is 1 since the coefficient of x_1x_2 is 1. The four possible values of the two variables x_1 and x_2 can be represented by the four vectors representing the pertinent instances of $\mathbf{B}_2 = [1\ \ x_1\ \ x_2\ \ x_1x_2]$: [1 0 0 0], [1 1 0 0], [1 0 1 0], and [1 1 1 1]. The first vector represents the case $x_1=x_2=0$; the second vector represents $x_1=1$, $x_2=0$; the third vector represents $x_1 = 0$, $x_2 = 1$; and the fourth vector represents $x_1 = x_2 = 1$. Taking the scalar product of each of these vectors with the vector representing the function, we see that the function only equals 1 for the second vector; that is, when x_1=1 and x_2=0.

3. Transition Matrix and Function Matrix

A synchronous switching network is a set of elements, each of which can (at a given instant of time) be in one of two states, 0 or 1. Time is quantized and the state of an element in the next instant of time is a binary function of the present states of the elements. It will be convenient to call the set of elements that are in the 1 state the state of the network, and represent this state by a product of subscripted x's where the subscripts are the numbers of the elements that are in the 1 state. If there are n elements in the network, there will be 2^n states of the network. For example, if n = 5, there are 2^5 = 32 states. The product 1 indicates the states in which all 5 elements are in state 0, while the product $x_1x_3x_5$ depicts the state in which elements 1, 3, and 5 are in state 1 while elements 2 and 4 are in state 0. The general correspondence between network state labels and element states for n = 3 is shown in Table 1.

Table 1. Correspondence between Network State Labels and Element States for n=3.

State label	**1**	x_1	x_2	x_1x_2	x_3	x_1x_3	x_2x_3	$x_1x_2\,x_3$
Value of the three tuple $x_3\,x_2\,x_1$	000	001	010	011	100	101	110	111

Note that in Table 1, we deliberately reversed the order of the three tuple (x_1, x_2, x_3). This way, our proposed ordering of the products corresponds exactly to a binary ordering of the integer $(x_3\ x_2\ x_1)_{10}$. No similar interpretation could be given to the original product ordering used by Cull [3].

The network state at time t can therefore be represented by a vector $\mathbf{Y}_t$, which is a binary column vector whose elements are all 0 except one 1 element. This particular one 1 element corresponds to the position where the product depicting the network state occurs in the basis vector $\mathbf{B}_n$. An alternative vector representation of the network state is vector $\mathbf{X}_t$, which equals the basis vector $\mathbf{B}_n$ evaluated at the network state at instant t. Hence, $\mathbf{X}_t$ is the vector of length 2^n that has l's in the positions representing any of the products of the elements of the synchronous switching network that are in state 1. In the above example of a network of 5 elements with elements 1, 3, and 5 in state 1 and elements 2 and 4 in state 0 at time t, the vector $\mathbf{Y}_t$ has a single 1 at the position of product $x_1x_3x_5$, while the vector $\mathbf{X}_t$ has 1's at the positions of all the products subsumed by that product, namely products 1, x_1, x_3, x_1x_3, x_5, x_1x_5, x_3x_5 and $x_1x_3x_5$.

Since the network is in only one state at a time and proceeds to its next state in the next instant of time, the behavior of the network is usually represented by a state transition diagram [6,16]. Cull [3] proposed the use of a transition matrix **T** such that:

$$\mathbf{Y}_{t+1}=\mathbf{T}\mathbf{Y}_t \tag{5}$$

The transition matrix **T** is a $2^n\times 2^n$ matrix having exactly one 1 in each column and 0 elements otherwise. Its name is reminiscent of matrices used in the analysis of Markov chains [17].

Cull [3] also defined a function matrix **A** as the $2^n\times 2^n$ matrix that has as its rows the vector representations of the 2^n products of the n functions computed by the elements of a synchronous switching network. Hence, the function matrix **A** relates two consecutive instances of the vector **X** as:

$$\mathbf{X}_{t+1}=\mathbf{A}\mathbf{X}_t \tag{6}$$

To see the connection between the transition matrix **T** and the function matrix **A**, Cull [3] introduced a state matrix **P** as a $2^n\times 2^n$ upper triangular matrix, the columns of which are the **X** vectors representing all the 2^n possible states of the network. The i^{th} row of **P** has a 1 in each column that represents a state in which the i^{th} product of the variables has the value 1. Considering the i^{th} row of **P** as the vector representation of a binary function, we see that it represents a function in which the i^{th} product of the variables may be factored out, leaving the sum of all the possible products of the remaining variables (see Example 1 below). The sum of all the possible products of a set of variables equals 1 if all the variables equal 0, and equals 0 otherwise. For example, in the case of $n = 2$, the sum $(1+x_1+x_2+x_1x_2)$ equals 1 when $x_1 = x_2 = 0$ and is 0 otherwise. Thus the i^{th} row of **P** represents the function that equals 1 if and only if the i^{th} product of the variables equals 1 and the remaining variables equal 0.

Cull [3] claimed, without proof, that the matrix P is its own inverse (or in modern terminology an involutory matrix [18]), *i.e.*,

$$\mathbf{P}^{-1}=\mathbf{P} \tag{7a}$$

$$\mathbf{P}^{2}=\mathbf{P}\mathbf{P}=\mathbf{I} \tag{7b}$$

where **I** is the identity matrix of 2^n elements. We will provide a formal proof of (7) shortly. The matrix **AP** formed by pre-multiplying the state matrix **P** by the function matrix **A** is made up of the columns of **P**, some of which may be repeated, the i^{th} column representing the state to which the i^{th} state goes. The matrix **PT** formed by post-multiplying the state matrix **P** by the transition matrix **T** is also composed of the columns of **P** with possible repetitions so that the ith column represents the state to which the ith states goes. Thus:

$$\mathbf{A\,P = P\,T} \tag{8}$$

We will present a formal proof of (8) later. We pre-multiply (8) by $\mathbf{P}^{-1}$ to obtain

$$\mathbf{P^{-1}A\,P = P\,A\,P = T} \tag{9}$$

and post-multiply (8) by $\mathbf{P}^{-1}$ to obtain

$$\mathbf{A = P\,T\,P^{-1} = P\,T\,P} \tag{10}$$

This means that we may obtain the functional representation from the transitional representation and vice- versa. In linear algebra terminology, we say that the function matrix **A** and transition matrix **T** are similar [14], with the similarity transformation between them being achieved by a self-inverse modal matrix **P**.

We contribute a novel observation that the state or modal matrix **P** is independent of the particular network used. It is unique for a given number of nodes n in the network, and therefore we will label it $\mathbf{P}_n$ when the need arises. We propose a straightforward technique for constructing **P** using the modern terminology of switching theory as presented in the standard text of Muroga [6]. Let T_j be the product labeling column j of **P** and T_i be the product labeling row i of **P**. Then we construct **P** by using the rule:

$$\mathbf{P}_{ij} = \begin{cases} 1, \ \text{if } T_j \text{ subsumes } T_i & \text{(11a)} \\ 0, \ \text{otherwise} & \text{(11b)} \end{cases}$$

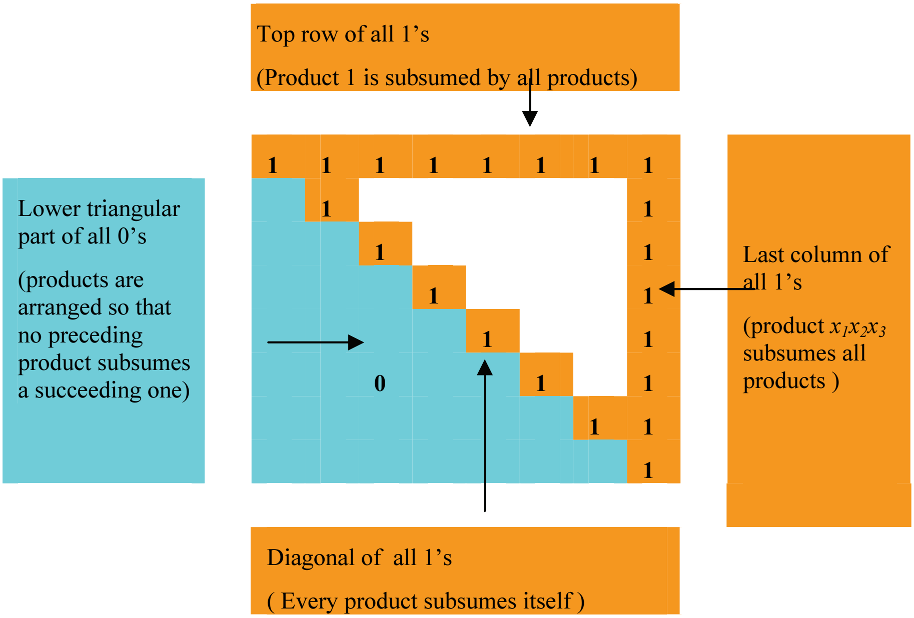
Note that a product T_j subsumes another T_i if the set of literals S_j of T_j contains (is a superset of) the set of the literals S_i of T_i. Eq.s (11) state that $\mathbf{P}_n$ records whether or not the elements of its column basis $\mathbf{B}_n$ subsume those of its row basis, which is also $\mathbf{B}_n$. The general nature of the state matrix **P** is shown in Table 2.

Furthermore, we deduce that the matrix **P** has the following properties:

- All columns (except the first) have even numbers of 1's.

This is because the number of subsets of a set S is $2^{|S|}$, where |s| is the cardinality of S.

Table 2. On the General Nature of the State Matrix P.


Top row of all 1's
(Product 1 is subsumed by all products)
Lower triangular part of all 0's
(products are arranged so that no preceding product subsumes a succeeding one)
Last column of all 1's
(product $x_1x_2x_3$ subsumes all products)
Diagonal of all 1's
(Every product subsumes itself)
1 1 1 1 1 1 1 1
1 1
1 1
1 1
1 1
0 1 1
1 1
1

For |S| > 0 , $2^{|S|}$ is even, and for |S| = 0 (S=Φ) , $2^{|S|}$ =1=odd. Since the set of literals of the product 1 is Φ, it is the only set that has an odd number of subsets (namely 1).

- All rows (except the last) also have even numbers of 1's.

This is because every set within the given universe has an even number of supersets, except the set corresponding to the product $x_1x_2x_3$ (the universe in this case) which is a superset of only itself.

Table 3 shows the gradual evolution of matrix $\mathbf{P}_n$ for n = 0, 1, 2, and 3 (2^n =1, 2, 4, and 8). Note that every matrix contains all its predecessors as submatrices, thanks to our new proposed ordering of products. In fact, we can now give the following recursive definition of **P**:

$$\mathbf{P}_0 = [1] \tag{12a}$$

$$\mathbf{P}_n = \begin{bmatrix} \mathbf{P}_{n-1} & \mathbf{P}_{n-1} \\ \mathbf{0}_{n-1} & \mathbf{P}_{n-1} \end{bmatrix} \tag{12b}$$

where $\mathbf{0}_{n-1}$ is the zero matrix of dimensions $2^{n-1}\times 2^{n-1}$. Note that $\mathbf{P}_{n-1}$ appears three times in the partitioning of $\mathbf{P}_n$ in (12b). Its first appearance in the top left partition is where it signifies the subsumption of its row basis $\mathbf{B}_{n-1}$ by its column basis $\mathbf{B}_{n-1}$. In the top right and bottom right partitions, we need to represent the subsumptions of row bases $\mathbf{B}_{n-1}$ and

$\mathbf{B}_{n-1}x_n$ by column basis $\mathbf{B}_{n-1}x_n$. Both subsumptions are again represented by $\mathbf{P}_{n-1}$. We can now give a formal proof via mathematical induction of the fact that $\mathbf{P}_n$ is involutory (self-inverse). We prove

(a) the non-recursive (boundary or limiting) case

$$\mathbf{P}_0^2 = [1]\,[1] = [1] = \mathbf{I}_0 \tag{13}$$

(b) $\{ \mathbf{P}_{n-1}^2 = \mathbf{I}_{n-1} \}$ implies $\{ \mathbf{P}_n^2 = \mathbf{I}_n \}$ since

$$\begin{aligned}
\mathbf{P}_n^2 = \mathbf{P}_n\,\mathbf{P}_n &= \begin{bmatrix} \mathbf{P}_{n-1} & \mathbf{P}_{n-1} \\ \mathbf{0}_{n-1} & \mathbf{P}_{n-1} \end{bmatrix} \begin{bmatrix} \mathbf{P}_{n-1} & \mathbf{P}_{n-1} \\ \mathbf{0}_{n-1} & \mathbf{P}_{n-1} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{P}_{n-1}^2 + \mathbf{0}_{n-1} & \mathbf{P}_{n-1}^2 + \mathbf{P}_{n-1}^2 \\ \mathbf{0}_{n-1} + \mathbf{0}_{n-1} & \mathbf{0}_{n-1} + \mathbf{P}_{n-1}^2 \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{I}_{n-1} + \mathbf{0}_{n-1} & \mathbf{0}_{n-1} \\ \mathbf{0}_{n-1} & \mathbf{0}_{n-1} + \mathbf{I}_{n-1} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{n-1} & \mathbf{0}_{n-1} \\ \mathbf{0}_{n-1} & \mathbf{I}_{n-1} \end{bmatrix} \\
&= \mathbf{I}_n
\end{aligned} \tag{14}$$

where $\mathbf{I}_n$ is the identity matrix of dimensions $2^n \times 2^n$. The abstract concepts of this section are now clarified with a few detailed tutorial examples.

Example 1

This example was discussed earlier by Cull [3]. Let us consider a network of 3 elements whose next states are given by:

$$f_1 = 1 + x_2 + x_1x_2 + x_1x_3 + x_1x_2x_3 \tag{15a}$$

$$f_2 = 1 + x_2 + x_3 + x_1x_2 + x_1x_3 + x_1x_2x_3 \tag{15b}$$

$$f_3 = 1 + x_1 + x_1x_2 + x_1x_3 + x_2x_3 + x_1x_2x_3 \tag{15c}$$

Here, the current state of element i is called x_i while its next state is called f_i. An alternative terminology is to call the current and next states x_{i_t} and $x_{i_{t+1}}$ [9,10]. This terminology will allow us to speak of later instances of the element state such as $x_{i_{t+2}}$, $x_{i_{t+3}}$, …, *etc*. The above equations can be conveniently expressed in map form. Figure 1 shows a Karnauagh-map representation (actually three maps combined together) for the 3-tuple of next state functions $f_1 f_2 f_3$. For convenience, the current value of each map cell is shown in a small corner box within the cell.

Table 3. The Matrix P with Associated Keys or Bases for n = 0, 1, 2, and 3 (2^n=1, 2, 4, and 8).

$\mathbf{P}_0$ =

	1
1	1

$\mathbf{P}_1$=

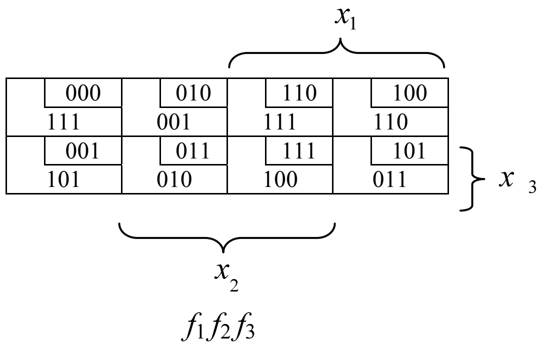
		1	x_1
1		1	1
x_1		0	1

$\mathbf{P}_2$=

	T$_i$	1	x_1	x_2	x_1x_2
T$_i$					
1		1	1	1	1
x_1		0	1	0	1
x_2		0	0	1	1
x_1x_2		0	0	0	1

$\mathbf{P}_3$=

		1	x_1	x_2	x_1x_2	x_3	x_1x_3	x_2x_3	$x_1x_2x_3$
1		1	1	1	1	1	1	1	1
x_1		0	1	0	1	0	1	0	1
x_2		0	0	1	1	0	0	1	1
x_1x_2		0	0	0	1	1	0	0	1
x_3		0	0	0	0	1	1	1	1
x_1x_3		0	0	0	0	0	1	0	1
x_2x_3		0	0	0	0	0	0	1	1
$x_1x_2x_3$		0	0	0	0	0	0	0	1



000 111	010 001	110 111	100 110
001 101	011 010	111 100	101 011

Fig. 1. A Karnaugh map representation for the next-state functions in (15).

From Fig. 1, it becomes apparent that the system above has the 2-cycle state diagram shown in Fig. 2, or in Fig. 3 (with an alternative state labeling).

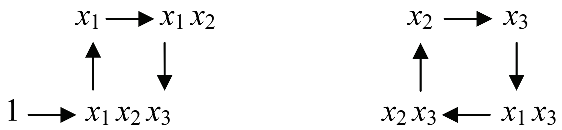


Fig. 2. State diagram for example 1 with states identified by the products representing them.

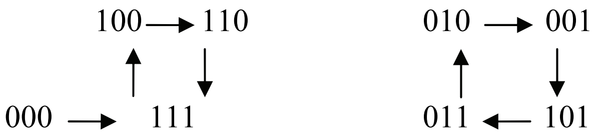


Fig. 3. The state diagram in Fig. 2 with states designated by the binary values of the 3-tuple $x_1x_2x_3$.

The transition matrix **T** is automatically derived from the state diagram and is shown in Table 4, wherein each state is labeled by its pertinent product. The first row in **T** has all 0's, since no state goes to state 1; the second row has a 1 in the last position, since state $x_1x_2x_3$ goes to state x_1, and so on. Every column in **T** must have one, and only one, 1, since in every current state there is a unique way for going to some next state.

Table 4. The Transition Matrix T for Example 1.

		Current state	1	x_1	x_2	x_1x_2	x_3	x_1x_3	x_2x_3	$x_1x_2x_3$
	Next state									
	1		**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	x_1		**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**
T =	x_2		**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**
	x_1x_2		**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**
	x_3		**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**
	x_1x_3		**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**
	x_2x_3		**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**
	$x_1x_2x_3$		**1**	**0**	**0**	**1**	**0**	**0**	**0**	**0**

In addition to the three original functions, we add the empty product (multiplication identity) f_0=1 and the products

$$f_1f_2 = 1 + x_2 + x_3 + x_1x_2 + x_2x_3 + x_1x_2x_3 \tag{15d}$$

$$f_1f_3 = 1 + x_1 + x_2 + x_1x_2x_3 \tag{15e}$$

$$f_2f_3 = 1 + x_1 + x_2 + x_3 + x_2x_3 + x_1x_2x_3 \tag{15f}$$

$$f_1f_2f_3 = 1 + x_1 + x_2 + x_3 + x_1x_3 + x_2x_3 \qquad (15g)$$

The function matrix **A** is simply the matrix that has the eight vectors [1 f_1 f_2 f_1f_2 f_3 f_1f_3 f_2f_3 $f_1f_2f_3$] as its rows. Thus we can write **A** as shown in Table 5.

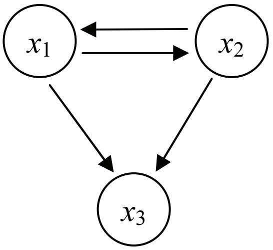
Table 5. The Function Matrix A for Example 1.

		Current state	1	x_1	x_2	x_1x_2	x_3	x_1x_3	x_2x_3	$x_1x_2\,x_3$
	Next state									
	f_0		**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	f_1		**1**	**0**	**1**	**1**	**0**	**1**	**0**	**1**
A =	f_2		**1**	**0**	**1**	**1**	**1**	**1**	**0**	**1**
	f_1f_2		**1**	**0**	**1**	**1**	**1**	**0**	**1**	**1**
	f_3		**1**	**1**	**0**	**1**	**0**	**1**	**1**	**1**
	f_1f_3		**1**	**1**	**1**	**0**	**0**	**0**	**0**	**1**
	f_2f_3		**1**	**1**	**1**	**0**	**1**	**0**	**1**	**1**
	$f_1f_2f_3$		**1**	**1**	**1**	**0**	**1**	**1**	**1**	**0**

It is simple to calculate **AP** and **PT** and verify that they are equal as expected.

Example 2

This example is taken from Farrow *et al.* [10] and deals with the network shown in Fig. 4. Here the next states are given by



x_1	f_2
0	1
1	0

OFF

x_2	f_3
0	0
1	1

ON

x_2	x_3	f_1
0	0	0
0	1	0
1	0	0
1	1	1

AND

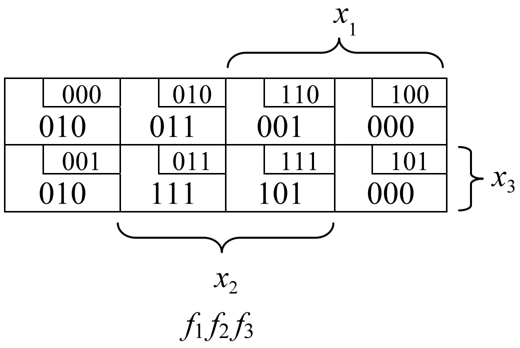
Fig. 4. The Boolean network of Example 2 and the truth tables of its excitations.

$$f_1 = x_2x_3 \qquad (16a)$$

$$f_2 = \overline{x_1} \qquad (16b)$$

$$f_3 = x_2 \tag{16c}$$

Figure 5 shows Karnaugh-map representations for the 3-tuple of next state functions $f_1f_2f_3$. Consequently, the system has the state diagram shown in Fig. 6.



000 010	010 011	110 001	100 000
001 010	011 111	111 101	101 000

Fig. 5. A Karnaugh map representation for the next-state functions in (16).

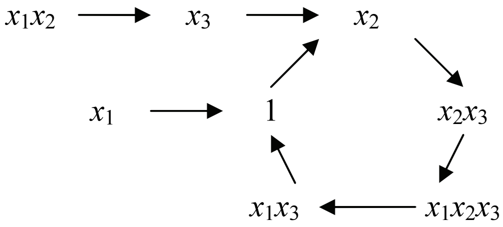


Fig. 6. The state diagram for example 2.

The transition matrix **T** is automatically derived from the state diagram and is shown in Table 6, wherein each state is labeled by its pertinent product. For convenience, we will henceforth adopt the convention of not explicitly writing the 0 entries in binary matrices such as **T**, **A**, and **P**. Instead, we will use blanks to express 0 entries.

In addition to the three original functions, we add f_0=1 and the products

$$f_1f_2 = x_2x_3 + x_1x_2x_3 \tag{16d}$$

$$f_1f_3 = x_2x_3 \tag{16e}$$

$$f_2f_3 = x_2 + x_1x_2 \tag{16f}$$

$$f_1f_2f_3 = x_2x_3 + x_1x_2x_3 \tag{16g}$$

Table 6. The Transition Matrix T for Example 2.

		Current state	1	x_1	x_2	x_1x_2	x_3	x_1x_3	x_2x_3	$x_1x_2\,x_3$
	Next state									
	1			**1**				**1**		
	x_1									
T =	x_2		**1**				**1**			
	x_1x_2									
	x_3					**1**				
	x_1x_3									**1**
	x_2x_3				**1**					
	$x_1x_2\,x_3$								**1**	

Table 7 shows the function matrix **A** as the matrix that has the eight vectors [f_0 f_1 f_2 f_1f_2 f_3 f_1f_3 f_2f_3 $f_1f_2\,f_3$] as its rows. Again, it is straightforward to verify the equality of **AP** and **PT** as required by (8).

Table 7. The Function Matrix A for Example 2.

			1	x_1	x_2	x_1x_2	x_3	x_1x_3	x_2x_3	$x_1x_2\,x_3$
	f_0		**1**							
	f_1								**1**	
A =	f_2		**1**	**1**						
	f_1f_2								**1**	**1**
	f_3				**1**					
	f_1f_3								**1**	
	f_2f_3				**1**	**1**				
	$f_1f_2\,f_3$								**1**	**1**

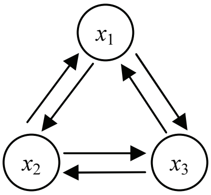
Example 3

This example is taken from Heidel *et al.* [9] and deals with the network shown in Fig. 7. Here the next states are given by:

$$f_1 = x_2 \vee x_3 \tag{17a}$$

$$f_2 = x_1 \vee x_3 \tag{17b}$$

$$f_3 = x_1 \vee x_2 \tag{17c}$$



x_2	x_3	f_1
0	0	0
0	1	1
1	0	1
1	1	1

OR

x_1	x_3	f_2
0	0	0
0	1	1
1	0	1
1	1	1

OR

x_1
0
0
1
1

Fig. 7. The Boolean network of Example 3 and the truth tables of its excitations.

Figure 8 shows Karnaugh-map representations for the 3-tuple of next state functions $f_1f_2f_3$. Consequently, the system has the state diagram shown in Fig. 9.

x_1

000 000	010 101	110 111	100 011
001 110	011 111	111 111	101 111

x_3

x_2

$f_1f_2f_3$

Fig. 8. A Karnaugh map representation for the next-state functions in (17).

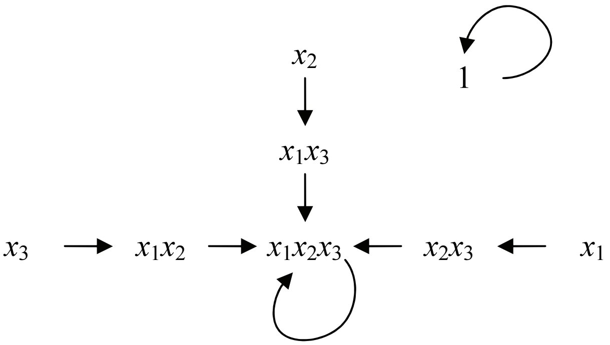


Fig. 9. The state diagram for example 3.

The transition matrix **T** is automatically derived from the state diagram and is shown in Table 8, wherein each state is labeled by its pertinent product.

Table 8. The Transition Matrix T for Example 3.

T =

	Current state	1	x_1	x_2	x_1x_2	x_3	x_1x_3	x_2x_3	$x_1x_2\,x_3$
Next state									
1		**1**							
x_1									
x_2									
x_1x_2						**1**			
x_3									
x_1x_3				**1**					
x_2x_3			**1**						
$x_1x_2\,x_3$					**1**		**1**	**1**	**1**

In addition to the three original functions, we add f_0=1 and the products

$$f_1f_2 \;= x_1\,x_2 + x_3 + x_1\,x_2\,x_3 \tag{17d}$$

$$f_1f_3 \;= x_1\,x_3 + x_2 + x_1\,x_2\,x_3 \tag{17e}$$

$$f_2f_3 \;= x_2\,x_3 + x_1 + x_1\,x_2\,x_3 \tag{17f}$$

$$f_1f_2f_3 = x_1\,x_2 + x_1\,x_3 + x_1\,x_2\,x_3 \tag{17g}$$

Table 9 shows the function matrix **A** as the matrix that has the eight vectors [f_0 f_1 f_2 f_1f_2 f_3 f_1f_3 f_2f_3 $f_1f_2f_3$] as its rows. Once more, direct multiplication verifies the validity of (8) in this case.

Table 9. The Function Matrix *A* for Example 3.

A =

		1	x_1	x_2	x_1x_2	x_3	x_1x_3	x_2x_3	$x_1x_2\,x_3$
f_0		**1**							
f_1				**1**		**1**		**1**	
f_2			**1**			**1**	**1**		
f_1f_2					**1**	**1**			**1**
f_3			**1**	**1**	**1**				
f_1f_3				**1**			**1**		**1**
f_2f_3			**1**					**1**	**1**
$f_1f_2\,f_3$					**1**		**1**	**1**	

Before closing this section, we must admit that we were intrigued by the fact that Cull [3] used two distict vectors $\mathbf{Y}_t$ and $\mathbf{X}_t$ to represent the same network state at instant t without giving a relation between them. Recall that $\mathbf{Y}_t$ has a single 1 element at the position where the state product occurs in the basis vector $\mathbf{B}_n$, while $\mathbf{X}_t$ has 1 elements at the positions of products subsumed by that state product. This means that:

$$\mathbf{X}_t = \mathbf{P}\ \mathbf{Y}_t \tag{18}$$

and hence

$$\mathbf{Y}_t = \mathbf{P}^{-1}\ \mathbf{X}_t = \mathbf{P}\ \mathbf{X}_t \tag{19}$$

If we substitute (18) in (6), we obtain

$$\mathbf{P}\ \mathbf{Y}_{t+1} = \mathbf{A}\ \mathbf{P}\ \mathbf{Y}_t \tag{20}$$

$$\mathbf{Y}_{t+1} = \mathbf{P}^{-1}\ \mathbf{A}\ \mathbf{P}\ \mathbf{Y}_t \tag{21}$$

Now, we compare (21) with (5) to arrive at the similarity transformation (9). Likewise, we can obtain a formal proof of (10). If we pre-multiply (9) by **P** or post-multiply (10) by **P**, we obtain (8).

4. Cycles and Characteristic Equations

A state of the network $\mathbf{Y}_t$ is called cyclic if

$$\mathbf{Y}_{t+m} \equiv \mathbf{T}^m\ \mathbf{Y}_t = \mathbf{Y}_t \qquad \text{for some m} \tag{22a}$$

Otherwise it is called transient when

$$\mathbf{Y}_{t+m} \equiv \mathbf{T}^m\ \mathbf{Y}_t \neq \mathbf{Y}_t \qquad \text{for all m} > 0 \tag{22b}$$

A set of cyclic states that map into one another are called a cycle. A state into which no state maps is called a first state, i.e., a first state does not serve as a next state for any current state, and hence the row corresponding to it in the transition matrix **T** is an all-0 row. A set of transient states that can be reached from a first state are called a transient chain. Note that a state is transient if it is a first state or it can be reached only from a first state or from some other transient states.

By analogy, to characteristic polynomials of real matrices [8,18], Cull [3] introduced the concept of the binary characteristic polynomial of a binary matrix **T**, defined as det($\mathbf{T}+x\mathbf{I}$) where the operations are carried out in the binary field. This polynomial will have $(x)^k$ as a factor, where k is the number of transient states. This can be seen by considering the rows of ($\mathbf{T}+x\mathbf{I}$) corresponding to first states. Each of these rows will have an (x) on the diagonal as its only non-zero entry. Expanding the determinant in terms of each of these rows, we obtain a factor of (x) for each of these rows and a reduced determinant. Each state that can be reached only from the eliminated first states will have in the reduced determinant an (x) on the diagonal as the only non-zero entry in its row. Expanding in terms of these rows and continuing the process we obtain the desired result. Having eliminated transient states, we can rearrange the rows and perform the same column operations so that the various cycles form blocks in the matrix. These rearrangements (which might produce a sign change in the real field) do not affect the characteristic polynomial over GF(2). The determinant of a matrix consisting of such blocks is the product of the determinants of these blocks. Hence, the characteristic polynomial of the cycle states is the product of the characteristic polynomials of the cycles. The characteristic polynomial of a cycle of length r will take the form of an $r \times r$ determinant:

$$\det \begin{bmatrix} x & & & \| & & & & 1 \\ 1 & x & & \| & & & & \\ & 1 & x & \| & & & & \\ & & 1 & \| & & & & \\ = & = & = & = & = & = & = & = \\ & & & \| & & x & & \\ & & & \| & & 1 & x & \\ & & & \| & & & 1 & x \end{bmatrix} = (x^r + 1).$$

Thus the characteristic polynomial of $\mathbf{T}$ becomes:

$$\det\,(\mathbf{T} + x\mathbf{I}) = x^k\,(x^{r_1} + 1)\,.\,.\,.\,(x^{r_i} + 1) \tag{23}$$

where k is the number of transient states and the subscripted r's are the lengths of the various cycles. We note that the eigenvalues are the roots of (23). In GF(2), the only possible eigenvalues are $x = 1$ (corresponding to one-eigenvectors that represent cycles) and $x = 0$ (corresponding to zero-eigenvectors that represent transient chains.)

Cull [3] observed that the factorization of the binary characteristic polynomial is not always unique since:

$$x^r + 1 = (\,x + 1\,)^r \tag{24}$$

when r is a power of 2. We note that (20) is true since

$$(x+1)^r = \sum_{k=0}^{r} \binom{r}{k} x^k = 1 + x^r + \sum_{k=1}^{r-1} \binom{r}{k} x^k \tag{25}$$

and $\binom{r}{k}$ { for $r = 2^m$, $1 \le k \le (r\text{-}1)$}is an even number that reduces to 0 in GF(2). This can be ascertained by viewing rows $r = 2^m$ in Pascal's triangle [19] since such rows consist solely of even entries except for their extreme entries, which are both equal to 1. We note also that the form (23) can always be reached provided the construction described in the previous paragraph is followed exactly. If a factor $(x+1)^{2^{\mathrm{m}}}$ is obtained, it should not be reduced according to (24) to a factor $(x^{2^{\mathrm{m}}}+1)$.

We note that the above considerations in terms of the state vector **Y** and the transition matrix **T** are still valid if the alternative state vector **X** and the function matrix **A** are used instead. Cull [3] asserted that the matrices **T** and **A** have the same characteristic polynomial. However, he did not offer a satisfactory proof for this assertion. A correct proof can go as follows and depends on the use of (7) and (10) together with the fact that the determinant of a product of several matrices is the product of their determinants [14]:

$$\begin{aligned}\det(\mathbf{A} + x\mathbf{I}\,) &= \det(\mathbf{PTP} + x\mathbf{I})\\ &= \det(\mathbf{PTP}+ x\mathbf{PP})\\ &= \det\,(\mathbf{P}\,(\mathbf{T}+x\mathbf{I})\,\mathbf{P})\\ &= \det \mathbf{P}\;\; \det\,(\mathbf{T}+x\mathbf{I}) \det \mathbf{P}\\ &= \det\,(\mathbf{T}+x\mathbf{I}) \det \mathbf{P} \det \mathbf{P}\\ &= \det\,(\mathbf{T}+x\mathbf{I}) \det \mathbf{PP}\\ &= \det\,(\mathbf{T}+x\mathbf{I}) \det \mathbf{I}\\ &= \det\,(\mathbf{T} + x\mathbf{I})\end{aligned} \tag{26}$$

Eq. (26) extends to the finite case a theorem that is well established in the continuous case; that two similar matrices share the same characteristic equation. The Cayley-Hamilton Theorem [8] tells us that any matrix satisfies its own characteristic equation; that is,

$$\mathbf{A}^{\mathrm{k}}\,(\mathbf{A}^{\mathrm{r}_1} + \mathbf{I}) \ldots (\mathbf{A}^{\mathrm{r}_i} + \mathbf{I}) = \mathbf{0} \tag{27}$$

and further, that any polynomial of a lower degree that **A** satisfies must divide the characteristic polynomial. Cull [3] asserted that there is an equation of lowest degree that **A** satisfies; which he called the minimal equation of **A**. Such an equation is typically given by the form

$$\mathbf{A}^{\mathrm{j}} (\mathbf{A}^{\mathrm{r}_\mathrm{h}}+ \mathbf{I}) \ldots (\mathbf{A}^{\mathrm{r}_\mathrm{g}}+ \mathbf{I}) = \mathbf{0} \tag{28}$$

Since the matrix polynomial is identically zero it must map every state vector $\mathbf{X}_\mathrm{t}$ to zero. Thus any state must be taken to a cycle and taken around the cycle an integral number of times. Thus in the minimal equation, j is the length of the longest transient chain and $r_h \ldots r_g$ are distinct cycle lengths such that all other cycle lengths divide one of the r's with no remainder. We now demonstrate the above concepts via the following illustrative examples.

Example 1 (revisited)

Using Tables 4 and 5, we find

$$\det (\mathbf{T} + x\mathbf{I}) = \det (\mathbf{A}+ x\mathbf{I}) = x\,(x^4 + 1)\,(x^3 + 1) \tag{29}$$

In this case the minimal equation is identical to the characteristic equation, *i.e.,*

$$\mathbf{A}\,(\mathbf{A}^4 + \mathbf{I})\,(\mathbf{A}^3 + \mathbf{I}) = \mathbf{0} \tag{30}$$

These facts tell us that the system has a transient chain of length 1, a cycle of length 4, and a cycle of length 3. These conclusions are readily verified by referring to the state diagram (Fig. 2 or Fig. 3).

Example 2 (revisited)

Using Tables 6 and 7, we find

$$\det (\mathbf{T} + x\mathbf{I}) = \det (\mathbf{A}+ x\mathbf{I}) = x^3(x^5 + 1) \tag{31}$$

and hence the system has 3 transient states and a cycles of length 5. Here, the minimal equation is not identical to the characteristic equation, but is given by

$$\mathbf{A}^2\,(\mathbf{A}^5 + \mathbf{I}) = \mathbf{0} \tag{32}$$

which means that the longest transient chain is of length 2 (in agreement with Fig. 6).

Example 3 (revisited)

Using tables 8 and 9, we find:

$$\det (\mathbf{T} + x\mathbf{I}) = \det (\mathbf{A}+ x\mathbf{I}) = x^6(x + 1)^2 \tag{33}$$

and hence the system has 6 transient states and two cycles of length 1. The factor $(x + 1)^2$ should not be replaced by $(x^2 + 1)$ which represents a single cycle of length 2. Here, the minimal equation is not identical to the characteristic equation, but is given by

$$\mathbf{A}^2\,(\mathbf{A} + \mathbf{I})^2 = \mathbf{0} \tag{34}$$

which means that the longest transient chain is of length 2 (in agreement with Fig. 9).

5. The Use of Eigenvectors

5.1 *One-Eigenvectors and Cycles*

The cyclic behavior of a synchronous switching network can be obtained from the **X**-type one-eigenvectors of the function matrix **A** or the **Y**-type one-eigenvectors of the transition matrix **T**. Cull [3] argued that the set of states in a cycle C maps onto itself, that is,

$$\mathbf{A}\sum_{C}\mathbf{X}=\sum_{C}\mathbf{X} \tag{35}$$

where $\sum_{C}\mathbf{X}$ is the sum (modulo 2) of the vectors representing the states on a cycle C. So the sum of the states on a cycle is a one-eigenvector of the matrix **A**. Eq. (35) can be stated for two specific cycles C_1 and C_2 as:

$$\mathbf{A}\sum_{C_1}\mathbf{X}=\sum_{C_1}\mathbf{X} \tag{36}$$

and

$$\mathbf{A}\sum_{C_2}\mathbf{X}=\sum_{C_2}\mathbf{X} \tag{37}$$

The sum (modulo 2) of (36) and (37) is

$$\mathbf{A}(\sum_{C_1}\mathbf{X}+\sum_{C_2}\mathbf{X})= \sum_{C_1}\mathbf{X}+\sum_{C_2}\mathbf{X} \tag{38}$$

Eq. (38) can be generalized to state that the sum of the states on any set of cycles is also a one-eigenvector of **A**. Relations (35)-(38) still hold when **A** is replaced by **T** and **X** is replaced by **Y**. Now let us consider any one-eigenvector of **A**:

$$\mathbf{AZ}=\mathbf{Z} \tag{39}$$

Since the set of vectors representing the states of the network form a basis of the 2^n dimensional vector space,

$$\mathbf{Z}=\sum_i a_i \mathbf{X}_i \tag{40}$$

where the $\mathbf{X}_i$'s are the 2^n vectors representing states of the network and each a_i is either 0 or 1, and can be seen to equal $(\mathbf{PZ})_i$, i.e., the ith component of **PZ**. This expansion of **Z** contains none of the $\mathbf{X}_i$'s that represent transient states since

$$\mathbf{A}^{2^n}\mathbf{Z}=\mathbf{Z} \tag{41}$$

and no transient state can occur after 2^n steps. Thus **Z** contains only cyclic states. If **Z** contains any state on a cycle it must contain each cyclic state that maps to this state. Continuing this argument we see that if **Z** contains any state on a cycle it contains all the states on that cycle. Thus any one eigenvector of **A** is composed of the sum of the states on a set of cycles. This allows us to conclude that the number of independent one-eigenvectors of **A** is the number of cycles exhibited by the behavior of the network represented by **A**. These considerations also hold for the transition matrix **T**, since if:

$$\mathbf{AZ}=\mathbf{Z} \tag{42}$$

$$\mathbf{PTPZ}=\mathbf{Z} \tag{43}$$

or

$$\mathbf{T}(\mathbf{PZ}) = (\mathbf{PZ}) \tag{44}$$

The intersection $\mathbf{Z}_1 \wedge \mathbf{Z}_2$ of two vectors, $\mathbf{Z}_1$ and $\mathbf{Z}_2$ is defined as the vector $\mathbf{Z}_3$ that represents the set of states that $\mathbf{Z}_1$ and $\mathbf{Z}_2$ have in common.

If :

$$\mathbf{Z}_1=\sum_i a_i^{(1)} \mathbf{X}_i \tag{45}$$

$$\mathbf{Z}_2=\sum_i a_i^{(2)} \mathbf{X}_i \tag{46}$$

then

$$\mathbf{Z}_1 \wedge \mathbf{Z}_2 = \sum_i (a_i^{(1)} a_i^{(2)}) \mathbf{X}_i \tag{47}$$

Vectors representing distinct cycles have the property that:

$$\mathbf{C}_1 \wedge \mathbf{C}_2 = \mathbf{C}_1' \wedge \mathbf{C}_2' = \mathbf{0} \tag{48}$$

where $\mathbf{C}_1 = \sum_{C_1} \mathbf{X}$, $\mathbf{C}_2 = \sum_{C_2} \mathbf{X}$, $\mathbf{C}_1' = \sum_{C_1} \mathbf{Y}$, and $\mathbf{C}_2' = \sum_{C_2} \mathbf{Y}$. So if one knows that there are k cycles and k disjoint one-eigenvectors can be obtained, the cycles, their lengths, and the exact sets of states that form the cycle can be determined [3]. We demonstrate some of the above ideas via the following example.

Example 1 (revisited)

The system in this example has the state diagram shown in Fig. 2 or Fig. 3, and hence has two cycles. The first cycle C_1 is:

$100 \to 110 \to 111 \to 100,$

which has three states represented by the three vectors

$$\mathbf{X} = \begin{bmatrix}1\\1\\0\\0\\0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}1\\1\\1\\1\\0\\0\\0\\0\end{bmatrix}, \text{and} \begin{bmatrix}1\\1\\1\\1\\1\\1\\1\\1\end{bmatrix}, \text{or} \quad \mathbf{Y} = \begin{bmatrix}0\\1\\0\\0\\0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\\0\\0\\0\\0\end{bmatrix}, \text{and} \begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\1\end{bmatrix} \tag{49}$$

referenced in the basis $\mathbf{B}_3 = [1 \quad x_1 \quad x_2 \quad x_1x_2 \quad x_3 \quad x_1x_3 \quad x_2x_3 \quad x_1x_2x_3]^T$

and hence, the sum of the vectors representing the states on the cycle C_1 is:

$$\mathbf{C}_1 = \sum_{C_1} \mathbf{X} = [1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1]^T \text{ or} \tag{50a}$$

$$\mathbf{C}_1' = \sum_{C_1} \mathbf{Y} = [0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1]^T \tag{50b}$$

which can be seen to satisfy $\mathbf{AC}_1 = \mathbf{C}_1$, $\mathbf{T}\mathbf{C}_1' = \mathbf{C}_1'$. The second cycle C_2 is

$010 \to 001 \to 101 \to 011 \to 010,$

which has four states represented by the vectors

$$\mathbf{X}=\begin{bmatrix}1\\0\\1\\0\\0\\0\\0\\0\end{bmatrix},\begin{bmatrix}1\\0\\0\\0\\1\\0\\0\\0\end{bmatrix},\begin{bmatrix}1\\1\\0\\0\\1\\1\\0\\0\end{bmatrix}\text{, and}\begin{bmatrix}1\\0\\1\\0\\1\\0\\1\\0\end{bmatrix}\text{, or}\quad \mathbf{Y}=\begin{bmatrix}0\\0\\1\\0\\0\\0\\0\\0\end{bmatrix},\begin{bmatrix}0\\0\\0\\0\\1\\0\\0\\0\end{bmatrix},\begin{bmatrix}0\\0\\0\\0\\0\\1\\0\\0\end{bmatrix}\text{, and}\begin{bmatrix}0\\0\\0\\0\\0\\0\\1\\0\end{bmatrix} \qquad (51)$$

and hence, the sum of the vectors representing the states on the cycle C_2 is:

$$\mathbf{C}_2=\sum_{C_2}\mathbf{X}=\begin{bmatrix}0 & 1 & 0 & 0 & 1 & 1 & 1 & 0\end{bmatrix}^T\text{, or} \qquad (52a)$$

$$\mathbf{C}_2'=\sum_{C_2}\mathbf{Y}=\begin{bmatrix}0 & 0 & 1 & 0 & 1 & 1 & 1 & 0\end{bmatrix}^T \qquad (52b)$$

which can also be seen to satisfy $\mathbf{AC}_2=\mathbf{C}_2$, $\mathbf{T}\mathbf{C}_2'=\mathbf{C}_2'$. The sum of the states of the two cycles C_1 and C_2:

$$\mathbf{C}_1+\mathbf{C}_2=(\sum_{C_1}\mathbf{X}+\sum_{C_2}\mathbf{X})=\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}^T\text{, or} \qquad (53a)$$

$$\mathbf{C}_1'+\mathbf{C}_2'=(\sum_{C_1}\mathbf{Y}+\sum_{C_2}\mathbf{Y})=\begin{bmatrix}0 & 1 & 1 & 1 & 1 & 1 & 1 & 1\end{bmatrix}^T \qquad (53b)$$

Note that the 3 vectors in (49) and the 4 vectors in (51) together with the vector

$$\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}^T \qquad (54)$$

representing the transient state 000, constitute a set of 8 vectors that form a basis of the 2^3-dimensional vector space. The basis formed by the **Y** vectors is a permutation of the identity matrix **I**, while that formed by the **X** vectors is a vector-permutation of the state matrix **P**. That is why it is easier to visualize (48) for the $\mathbf{C}'$ vectors than for the **C** vectors.

5.2 *Zero-Eigenvectors and Transient Chains*

The states of a transient chain come to an end at a state $\mathbf{X}_i$ whose next state $\mathbf{AX}_i$ belongs to some cycle C. This next state serves also as the next state $\mathbf{AX}_j$ of some state $\mathbf{X}_j$ belonging to C. Therefore, while $\mathbf{X}_i \neq \mathbf{X}_j$, we have:

$$\mathbf{AX}_i = \mathbf{AX}_j \tag{55}$$

or

$$\mathbf{A}\ (\mathbf{X}_i + \mathbf{X}_j) = \mathbf{0} \tag{56}$$

If we sum (modulo 2) relations of the form (56) for several transient chains, we obtain a relation of the form

$$\mathbf{AZ} = \mathbf{0} \tag{57}$$

where

$$\mathbf{Z} = \mathbf{X}_{(1)} + \mathbf{X}_{(2)} + ... + \mathbf{X}_{(2k)} \tag{58}$$

is a typical zero-eigenvector, represented as the sum (modulo 2) of an even number of state vectors. Note that the vectors $\mathbf{X}_i$ are distinct for different transient chains, while the vector $\mathbf{X}_j$ is possibly shared by some chains, in which case duplicate instances of $\mathbf{X}_j$ cancel, two at a time, ($\mathbf{X}_j$+$\mathbf{X}_j$= $\mathbf{0}$). Therefore, an even number 2k of state vectors remain in (58), where k is the number of chains involved minus the number of $\mathbf{X}_j$ cancellations. There are as many independent zero-eigenvectors as there are chains. Each of these independent zero-eigenvectors is of the form ($\mathbf{X}_i$+$\mathbf{X}_j$), where the $\mathbf{X}_i$'s are independent. Other zero-eigenvectors are not independent, since they are sums (modulo 2) of some of the independent ones. If **Z** is a zero-eigenvector of **A**, then **PZ** is a zero-eigenvector for **T**, since (57) implies that

$$\mathbf{PTPZ} = \mathbf{0},$$

$$\mathbf{PPTPZ} = \mathbf{P0},$$

$$\mathbf{T(PZ) = 0} \tag{59}$$

However, it is more useful for **T** to use a row zero-eigenvector $\mathbf{Z}^{\mathrm{T}}$, since

$$\mathbf{Z}^{\mathrm{T}}\mathbf{T} = \mathbf{0}^{\mathrm{T}} \tag{60}$$

will have an independent solution for each row of **T** that is composed entirely of zeroes. Such a row will in fact correspond to a first state of a transient chain. If $\mathbf{Z}^{\mathrm{T}}$ is a row zero-eigenvector of **T**, then $\mathbf{Z}^{\mathrm{T}}\mathbf{P}$ is a row zero-eigenvector of **A**, since (60) implies

$$\mathbf{Z}^{\mathrm{T}}\mathbf{PAP} = \mathbf{0}^{\mathrm{T}}$$

$$\mathbf{Z}^{\mathrm{T}}\mathbf{PAPP = 0}^{\mathrm{T}}\mathbf{P},$$

$$(\mathbf{Z}^{\mathrm{T}}\mathbf{P})\mathbf{A}=\mathbf{0}^{\mathrm{T}} \tag{61}$$

We demonstrate some of the above ideas via the following example.

Example 3 (revisited)

The state diagram in Fig. 9 has three transient chains. For each chain, Table 10 shows the end state $\mathbf{X}_i$, the state $\mathbf{X}_j$ sharing the same next state with $\mathbf{X}_i$, and the zero-eigenvector **Z** for matrix **A** corresponding to the chain. The three zero-eigenvectors **Z** in Table 10 can be verified to satisfy (57) and are independent. These three vectors constitute a basis for the zero-eigenspace of **A**.

Table 10. The Three Transient Chains of Example 3.

Chain	$\mathbf{X}_i$	$\mathbf{X}_j$ (always state $x_1x_2x_3$)	$\mathbf{Z} = \mathbf{X}_i + \mathbf{X}_j$
$x_3 \to x_1x_2 \to \ldots$	$[1\ 1\ 1\ 1\ 0\ 0\ 0\ 0]^T$	$[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^T$	$[0\ 0\ 0\ 0\ 1\ 1\ 1\ 1]^T$
$x_2 \to x_1x_3 \to \ldots$	$[1\ 1\ 0\ 0\ 1\ 1\ 0\ 0]^T$	$[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^T$	$[0\ 0\ 1\ 1\ 0\ 0\ 1\ 1]^T$
$x_1 \to x_2x_3 \to \ldots$	$[1\ 0\ 1\ 0\ 1\ 0\ 1\ 0]^T$	$[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^T$	$[0\ 1\ 0\ 1\ 0\ 1\ 0\ 1]^T$

Table 8 demonstrates that the transition matrix **T** has three rows composed entirely of zeroes, which correspond to the three first states x_1, x_2, and x_3 of the transient chains in Fig. 9. Therefore, the row zero-eigenspace of **T** is spanned by the three independent row vectors

$$\mathbf{Z}_1^T = \begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix},$$

$$\mathbf{Z}_2^T = \begin{bmatrix}0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\end{bmatrix},$$

$$\mathbf{Z}_3^T = \begin{bmatrix}0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\end{bmatrix},$$

which are **Y**-type representations of the first states.

6. Conclusions

This paper presents a tutorial exposition of the classical matrix techniques used in the linear analysis of synchronous switching networks. The current work parallels but dramatically extends and updates the seminal work of Cull [3]. The analysis is based on the use of the simplest of the finite or Galois fields, namely, GF(2).

For future work, we propose the extension of the concepts and methods developed herein to higher-order finite or Galois fields GF(p^r). There are situations in which the use of higher-order fields is possibly more compact, convenient, or enlightening. We also note that Cull [3] preferred finite fields to Boolean algebras. There is an equivalence between Boolean algebras and the so-called Boolean rings with unit [12]. The simplest finite field GF(2) used by Cull [3] and used herein coincides with the 2-element Boolean ring. Therefore, it might be useful to explore recent research findings [20,21], so as to utilize higher-order Boolean rings also in the current analysis.

We also hope to consider possibilities of dispensing with the redundancies in the current representations. Though both the function matrix **A** and the transient matrix **T** are $2^n \times 2^n$ binary matrices, each of them is equivalent to only $n \times 2^n$ bits of information. The matrix **A** is uniquely determined by only n of its 2^n-element rows. Similarly, the matrix T is uniquely determined by specifying (in each of its 2^n columns) the location of a single 1 element among 2^n elements. Such a specification requires n bits only. We believe that the removal of redundancies from the current matrix representation can lead to a systematic derivation of scalar representations of synchronous switching networks. Such a systematic derivation could replace the ad hoc derivations of scalar equations or reduced scalar equations given in Ref. [9] and [10], respectively.

Another possible area of future investigation is the use of spectral techniques [22] in solving the transition or difference equations over GF(2) arising in the study of synchronous switching networks. Such an approach is promising due to the analogy between the current discrete case and the continuous case. Needless to say, spectral or transform-domain techniques are very powerful in the solution of differential equations arising in the classical dynamic system theory.

Finally, we would like to investigate the relation between the current state matrix **P**, and the transformation matrix that relates the vector representations of a general n-variable switching function in the operational domain (minterm expansion) and in the function domain (Reed-Muller expansion) [23].

References

[1] **Kauffman**, **S. A.**, Metabolic Stability and Epigenesis in Randomly Connected Nets, *J. Theor. Biol.*, **22:**437-467 (1969).

[2] **MeCulloch**, **W.S.** and **Pitts**, **W.H.**, A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bull. Math. Biophys.*, **5:** 115-133 (1943).

[3] **Cull**, **P.**, Linear Analysis of Switching Nets, *Kybernetik*, **8** (1): 31-39 (1971).

[4] **Lin**, **S.** and **Costello. D. J.**, *Error Control Coding*, Second Edition, Prentice Hall, Englewood Cliffs, NJ (2004).

[5] **Garrett**, **P.**, *The Mathematics of Coding Theory*, Pearson Prentice Hall, Upper Saddle River, NJ (2004).

[6] **Muroga**, **S.**, *Switching Theory and Logic Design*, Wiley, New York (1979).

[7] **Brown**, **F. M.**, *Boolean Reasoning: The logic of Boolean Equations*, Kluwer Academic Publishers, Boston (1990).

[8] **Antsaklis**, **J. P.** and **Michel**, **A. N.**, *Linear Systems*, Birkhauser, Boston (2005).

[9] **Heidel**, **J.**, **Maloney**, **J.**, **Farrow**, **C.** and **Rogers**, **J.**, Finding Cycles in Synchronous Boolean Networks with Applications to Biochemical Systems, *Int. J. Bifurcat. Chaos.*, **13** (3): 535-552 (2003).

[10] **Farrow**, **C.**, **Heidel**, **J.**, **Maloney**, **J.** and **Rogers**, **J.**, Scalar Equations for Synchronous Boolean Networks with Biological Applications. *IEEE Transactions on Neural Networks.*, **15** (2): 348-354 (2004).

[11] **Zhao**, **Q.**, A remark on "Scalar Equations for Synchronous Boolean Networks with Biological Applications," *IEEE Transactions on Neural Networks*., **16** (6): 1715-1716 (2005).

[12] **Rudeanu**, **S.**, *Boolean Functions and Equations*, North Holland Publishing Company, Amsterdam, The Netherlands (1974).

[13] **Rushdi**, **A. M.**, Using Variable-Entered Karnaugh Maps to Solve Boolean Equations, *International Journal of Computer Mathematics*, **78** (1): 23-38 (2001).

[14] **Rushdi**, **A. M.**, Efficient Solution of Boolean Equations Using Variable-Entered Karnaugh Maps, *Journal of King Abdulaziz University: Engineering Sciences*, **15** (1): 105-121 (2004).

[15] **Fraleigh**, **J. B.**, *A First Course in Abstract Algebra*, Seventh Edition, Addison Wesley, Reading, MA (2002).

[16] **Lee**, **S. C.**, *Modern Switching Theory and Digital Design*, Prentice Hall, Englewood Cliffs, NJ (1978).

[17] **Trivedi**, **K. S.**, *Probability and Statistics with Reliability, Queuring and Computer Science Applications* (Second Edition), Prentice Hall, Englewood Cliffs, NJ (2002).

[18] **Kreyszig**, **E.**, *Advanced Engineering Mathematics*, Ninth Edition, Wiley, New York (2006).

[19] **Edwards**, **A. W. F.**, *Pascal's Arithmetical Triangle: The Story of a Mathematical Idea*, Wiley, New York (2002).

[20] **Ekin**, **O.**, **Foldes, S.**, **Hammer, P. L.** and **Hellerstein**, **L.**, Equational Characterization of Boolean Function Classes, *Discrete Mathematics*, **211**: 27-51 (2000).

[21] **Couceiro**, **M.** and **Foldes**, **S.**, Definability of Boolean Function Classes by Linear Equations over GF(2), *Discrete Applied Mathematics,* **142**: 29-34 (2004).

[22] **Hurst**, **S. L.**, **Miller**, **D. M.** and **Muzio**, **J. C.**, *Spectral Techniques in Digital Logic*, Academic Press, London (1985).

[23] **Quintana**, **J. M.** and **Avedillo**, **M. J.**, Reed-Muller Descriptions of Symmetric Functions, *Proceedings of the 2001 IEEE International Symposium on Circuits and Systems (ISCAS 2001)*, **4**: 682-685 (2001).

حول التحليل الخطي لشبكات التبديل المتزامنة

علي محمد علي رشدي وسلطان عبيد سعد العتيبي

قسم الهندسة الكهربائية وهندسة الحاسبات ، جامعة الملك عبدالعزيز

جدة، المملكة العربية السعودية

المستخلص. يتم تقديم عرض تفصيلي لبحث سابق رائد عن التحليل الخطي لشبكات التبديل المتزامنة. هذا التحليل، الذي يعتمد على استعمال الحقل المحدود أو حقل غالوا ح غ (٢)، يشبه التحليل الخطي للنظم المتصلة، وله تطبيقات هامة في علميّ الجينات والكيمياء الحيوية. يتم تمثيل شبكة التبديل بمصفوفة اقتران أو مصفوفة انتقال، وهما مصفوفتان مرتبطتان بتحويل تشابه تمثله مصفوفة للحالات. وقد أدى استخدامنا لترتيب جديد معاود لمفاتيح أو مؤشرات هاتين المصفوفتين إلى الكشف عن العديد من الملامح والخصائص الجديدة والمهمة. وقد لاحظنا أن مصفوفة الحالات لا تعتمد على أية شبكة معينة، وإنما تعتمد فقط على عدد العقد (الرؤوس) في الشبكة، ومن ثم قدمنا تفسيراً مبتكراً لهذه المصفوفة بدلالة المفهوم الحديث للاحتواء الحرفي لمضروب منطقي بواسطة مضروب آخر. وقد كشف ذلك عن وجود بنية معاودة لمصفوفة الحالات مما أدى إلى البرهنة على كونها تساوي معكوسها. وتجري دراسة السلوك الذاتي لشبكات التبديل بدلالة المعادلات المميزة والمتجهات المميزة للمصفوفتين سالفتي الذكر. وبوجه عام، يجري إثراء الأفكار التقليدية بمفاهيم ومصطلحات حديثة، مع دعمها ببراهين صحيحة وتوضيحها بأمثلة تعليمية تفصيلية.